

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA DE INGENIERIA ELÉCTRICA



**Medidor inalámbrico de consumo de energía  
eléctrica de bajo costo**

PRESENTADO POR:

**JONATHAN ALBERTO ZALDAÑA**

PARA OPTAR AL TITULO DE:

**INGENIERO ELECTRICISTA**

CIUDAD UNIVERSITARIA, DICIEMBRE 2011

**UNIVERSIDAD DE EL SALVADOR**

**RECTOR :**

**ING. MARIO ROBERTO NIETO LOVO**

**SECRETARIA GENERAL :**

**DRA. ANA LETICIA ZAVALA DE AMAYA**

**FACULTAD DE INGENIERIA Y ARQUITECTURA**

**DECANO :**

**ING. FRANCISCO ANTONIO ALARCÓN SANDOVAL**

**SECRETARIO :**

**ING. JULIO ALBERTO PORTILLO**

**ESCUELA DE INGENIERIA ELÉCTRICA**

**DIRECTOR :**

**ING. JOSÉ WILBER CALDERÓN URRUTIA**

UNIVERSIDAD DE EL SALVADOR  
FACULTAD DE INGENIERIA Y ARQUITECTURA  
ESCUELA DE INGENIERIA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

**INGENIERO ELECTRICISTA**

Título :

**Medidor inalámbrico de consumo de energía  
eléctrica de bajo costo**

Presentado por :

**JONATHAN ALBERTO ZALDAÑA**

Trabajo de Graduación Aprobado por:

Docente Director :

**Dr. CARLOS EUGENIO MARTÍNEZ CRUZ**

San Salvador, diciembre 2011

Trabajo de Graduación Aprobado por:

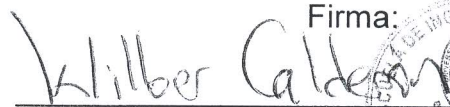

Docente Director :

**DR. CARLOS EUGENIO MARTÍNEZ CRUZ**

## ACTA DE CONSTANCIA DE NOTA Y DEFENSA FINAL

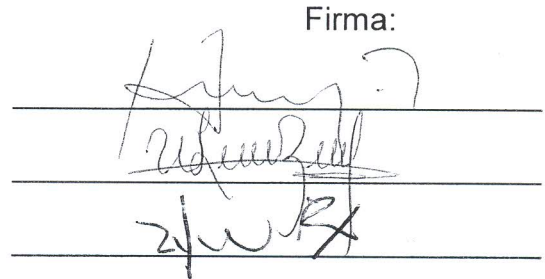
En esta fecha, 24 de noviembre de 2011, en la Sala de Lectura de la EIE, a las 4:00 horas, en presencia de las siguientes autoridades de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

1. Ing. José Wilber Calderón Urrutia  
Director Interino

Firma:   


Y, con el Honorable Jurado de Evaluación integrado por las personas siguientes:

- 1- Ing. Hugo Miguel Colato Rodríguez
- 2- Ing. Walter Leopoldo Zelaya Chicas
- 3- Ing. José Lito Rodríguez Lizama

Firma: 

Se efectuó la defensa final reglamentaria del Trabajo de Graduación:

Medidor Inalámbrico de consumo de energía eléctrica de bajo costo.

A cargo del los Bachiller:

Br. Jonathan Alberto Zaldaña

Habiendo obtenido el presente Trabajo una nota final, global de: 9.0

( NUEVE PUNTO CERO )

## **Agradecimientos**

A Dios y a Jesucristo, por estar con migo todos los días de mi vida, por todas las bendiciones derramadas en mi vida, por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos.

A mi abuela, Marta; mi madre, Isabel; mis hermanos, Mauricio y Gabriela; mis tíos; mi tía, Ana que en paz descanse. Gracias a todos por apoyarme y animarme siempre a seguir adelante.

A mis amigos que me apoyaron, que compartieron sus alegrías, sus tristezas y su confianza conmigo. Gracias por todo su apoyo sin ustedes esto no hubiese sido posible gracias.

A mi asesor, por ser parte de este esfuerzo, gracias por el apoyo brindado a lo largo de todo el proyecto de graduación.

A AES El Salvador, por la donación de los materiales que fueron utilizados en este proyecto de graduación.

A todos los que de una manera u otra me ayudaron a concluir esta etapa de mi vida muchas gracias a todos.

Jonathan Alberto Zaldaña.

Para Marta Garcia, mi abuela.

## Resumen

Para medir el consumo de energía eléctrica se diseñó y construyó un medidor tipo *Smart Meter*. Uno de los objetivos principales de este trabajo fue que el diseño y construcción del medidor resulte de bajo costo. No se pretende la construcción de un medidor con la exactitud de un *Smart Meter*, ya que esto encarecería la construcción del medidor, pero si utilizar la forma de comunicación inalámbrica que muchos *Smart Meter* utilizan. Muchos medidores *Smart Meter* que están a la venta en el mercado utilizan sistemas inalámbricos para poder enviar las mediciones de energía eléctrica. La ventaja de utilizar una comunicación inalámbrica y en especial la WIFI es la posibilidad de monitorear las mediciones en tiempo real. Este monitoreo en tiempo real permite analizar el estado de la carga.

Las mediciones se realizan a través de un sistema el cual mide la potencia y energía eléctrica consumida en la red eléctrica. Luego se envían las mediciones por medio de una red WIFI a un servidor web. Éste utiliza un router personal con el sistema operativo OpenWrt, el cual le da la potencia de un sistema embebido en un router personal.

El trabajo de graduación se divide en tres partes principales. La primera consiste en un circuito que mide la potencia eléctrica. Éste se compone de un microcontrolador PIC16F876A, un reloj de tiempo real el DS1307 y un transductor de corriente AC a voltaje DC (Pinza de corriente). La otra parte del diseño es el sistema para el envío de las mediciones por medio de una red WIFI. Éste está basado en la idea del medidor Flukso, el cual utiliza un router personal como sistema embebido. En nuestro caso se utilizó el router DIR-300. A este se le instaló el sistema Operativo OpenWrt compilación de Flukso. También se le programó aplicaciones que realizan las funciones para el envío de las mediciones vía WIFI y la comunicación del router con el circuito medidor de potencia. Esta comunicación se realizó por medio del puerto UART.

La tercera parte consistió en el diseño de un servidor web. El servidor grafica las mediciones donde el usuario podrá monitorear sus consumo energético. Se diseñó un sitio web utilizando el modelo en 3 capas. Para la programación de este se utilizó: PHP, MYSQL y RRDTOLL. Las gráficas que se podrán visualizar en el sitio web serán: gráficas de potencia versus tiempo o energía versus tiempo. Para esto se utilizó el programa RRTOOL. Los usuarios del servidor web por un lado serán las personas que adquieran un medidor las cuales iniciaran cesión en este por medio de la creación de una cuenta, por otro lado los medidores los cuales enviaran las mediaciones obtenidas por medio de una red WIFI la cual tenga acceso al servidor web.

# Objetivos

## **Objetivo general:**

Construir un medidor inalámbrico de consumo de energía eléctrica de bajo costo que opere en el entorno de red social.

## **Objetivos específicos:**

1. Diseñar un medidor de energía de bajo costo basado en un router WIFI.
2. Construir el medidor de energía utilizando el router DIR-300 del fabricante DLINK.
3. Montar un servidor web en el cual se puedan guardar los datos de consumo de energía.
4. Montar una pequeña red de por lo menos cinco medidores de consumo de energía en el campus de la Universidad.

# Contenido:

1. Introducción.....	17
2. Medidor de potencia eléctrica Flukso .....	21
2.1 Configuración e instalación del medidor Flukso.....	23
2.2 Gráficas de las Mediciones en el servidor web .....	24
2.3 Análisis del hardware y del software de Flukso.....	26
2.4 Comunicación Microcontrolador-Router.....	38
2.5 Ventajas y desventajas del medidor Flukso .....	39
2.5.1 Ventajas de Flukso .....	39
2.5.2 Desventajas de Flukso.....	40
3. Diseño y construcción del medidor inalámbrico de consumo de energía eléctrica de bajo costo..	42
3.1 Diseño del sistema para el envío de las mediciones vía WIFI.....	43
3.2 Diseño del circuito medidor de potencia y energía eléctrica.....	45
3.2.1 Microcontrolador PIC16F876A .....	46
3.2.2 Reloj de tiempo real DS1307 .....	48
3.2.2.1 Características del DS1307.....	49
3.2.2.2 Diagrama de conexión del DS1307.....	49
3.2.3 Referencia de voltaje REF195 .....	51
3.2.4 Pinza de corriente y el módulo ADC del PIC16F876A.....	53
3.3 Circuito medidor de potencia eléctrica.....	55
3.4 Algoritmos y rutinas utilizadas en el PIC16F876A .....	57
3.5 Construcción del circuito .....	61
3.6 Acoplamiento del circuito medidor de potencia con el DIR-300 .....	64
4. Modificación del script flukso.lua .....	67
4.1 Lectura de las mediciones por el puerto UART .....	67
4.2 Envío de las mediciones hacia servidor web por medio de XML-RPC .....	70
4.3 LUA Socket.....	71
4.4 Modificación del script flukso.lua para el envío de las mediciones a el servidor Web. ....	72
5. Diseño e implementación del servidor web .....	74
5.1 Arquitectura cliente/servidor en tres 3 capas.....	75
5.2 Diseño de la capa de datos .....	83

5.2.1 Información de los usuarios en tablas de MYSQL.....	83
6. Línea futura del Medidor Inalámbrico de Consumo de Energía Eléctrica de Bajo Costo.....	86
6.1 Uso del router DRAGINO.....	86
6.2 Topología de red Mesh en los medidores de potencia.....	88
7. Conclusiones.....	89
Anexo A.....	90
A. Instalación del firmware OpenWrt al router.....	90
A.2 Procedimiento para la instalación si el DIR-300 posee el software del fabricante.....	91
A.3 Pasos para la instalación de OpenWrt si el DIR-300 tiene instalado DD-WRT u OpenWrt.....	100
A.4 Instalación del paquete Flukso.....	101
Anexo B.....	103
B. Creación de las tablas en MYSQL.....	103
Anexo C.....	105
C. Gráficos con RRDTool.....	105
C.1 Gráficos obtenidos con RRDTool.....	107
C.2 Bases de datos en RRDTool.....	108
Anexo D.....	111
D. Pruebas realizadas al sistema.....	111
D.1 Pruebas de comunicación inalámbrica WIFI.....	111
D.2 Análisis de las graficas de potencia versus tiempo.....	115
D.3 Comparación de las mediciones tomadas con el Medidor y el analizador de carga FLUK.....	116
Anexo E.....	118
E. Topología de red.....	118
Anexo F.....	120
F. Costo de construcción del medidor.....	120
Referencias.....	122

## Lista de Figuras:

Figura 2.1: Medidor Flukso.....	21
Figura 2.2: Medidor Flukso con todos sus accesorios. ....	22
Figura 2.3 : Pinza de Corrientes o Transductor de Corriente AC/DC.....	22
Figura 2.4: Interfaz web del medidor Flukso. ....	23
Figura 2.5: Conexión del Medidor Flukso al tablero eléctrico por medio de la pinza de corriente.....	24
Figura 2.6 : Pagina web del servidor Flukso donde se visualizan las mediciones. ....	25
Figura 2.7: Diagrama de Bloques del sistema del medidor Flukso.....	26
Figura 2.8 : Interior del medidor Flukso donde se observa el sensor medidor de potencia.....	27
Figura 2.9: Sensor medidor de Potencia Flukso V 1.2. ....	28
Figura 2.10: WAP-2102 de Abocon.....	29
Figura 2.11 : conexión del medidor Flukso con la PC.....	30
Figura 2.12 : Conexión al Flukso por medio de ssh. ....	31
Figura 2.13 : Sistema de Directorios en OpenWrt compilación para Flukso.....	31
Figura 2.14 : Directorios en OpenWrt.....	32
Figura 2.15 : Directorio WWW el cual contiene los archivos HTML y CGI que utiliza un servidor web. ....	32
Figura 2.16 : Ejecución del intérprete o maquina virtual de LUA en el Flukso. ....	33
Figura 2.17 : Directorio en donde se encuentran alojados los script de Flukso en OpenWrt. ....	33
Figura 2.18 : Fichero /etc/init.d/ donde se encuentran los script que inician o detienen los demonios en Linux. ....	34
Figura 2.19 : Archivo de configuración del programa ntpclient.....	35
Figura 2.20 : conexión con un terminal remoto al medidor Flukso, ejecución del comando date.....	36
Figura 2.21 : Directorio /etc/config/ en donde se alojan las configuraciones de las aplicaciones en OpenWrt.....	36
Figura 2.22 : contenido del archivo de configuración flukso. ....	37
Figura 2.23 : estructura de los archivos de configuración en OpenWrt.....	37
Figura 2.24 : Consola de comandos en donde se ejecuta el comando cat en el fichero especial ttyS0. ....	38
Figura 3.1 : Metodología de diseño para el medidor de energía WIFI de bajo costo. ....	42
Figura 3.2 : Medidor Shark con comunicación WIFI [23].....	44
Figura 3.3: Diagrama de Bloques del Sistema: Medidor de energía, red eléctrica y servidor web.....	46
Figura 3.4 : Diagrama de distribución de pines del PIC16F876A.....	47
Figura 3.5 : PIC16F876A encapsulado tipo SOIC. ....	48
Figura 3.6 : Diagrama de pines del DS1307.....	49
Figura 3.7 : Circuito típico de conexión el DS1307.....	50
Figura 3.8: Configuración de pines en un encapsulado SOIC-REF195. ....	52
Figura 3.9 : Conexión típica de la referencia de voltaje REF-195.....	52
Figura 3.10 : Conexión de la pinza de corriente al módulo ADC.....	54
Figura 3.11 : Circuito medidor de potencia versión 1.3.....	56

Figura 3.12 : Diagrama de Flujo del programa principal.....	59
Figura 3.13 : Subrutinas a atender cuando ocurren las interrupciones por TMR1 y EXT_RBO.....	60
Figura 3.14: (a) Diseño del PCB; (b) Construcción del PCB. ....	61
Figura 3.15 : (a) Microscopio; (b) estación de soldadura necesaria para poder realizar la soldadura de los componentes SMD. ....	63
Figura 3.16 : componentes SMD y circuito impreso. ....	63
Figura 3.17: (a) Cara del PCB donde se muestra el PIC16F876A; (b) Cara del PCB donde se soldó la batería de respaldo.....	64
Figura 3.18 : (a) Router DIR-300; (b) ubicación del puerto UART en el DIR-300.....	64
Figura 3.19 : (a )Acoplamiento del circuito con el DIR-300; (b)Conexión de los cables para la pinza; (c) Pines que conectan los puertos UART; (d) Conectores de la pinza y fuente de alimentación.....	65
Figura 3.20 : (a) conexión de la pinza de corriente; (b) Conector para la pinza de corriente.....	66
Figura 3.21 : Medidor Inalámbrico de Consumo de Energía Eléctrica de Bajo Costo. ....	66
Figura 4.1 : segmento de cogido del script flukso.lua. ....	68
Figura 4.2 : Continuación del segmento de cogido del script flukso.lua. ....	69
Figura 4.3 : Esquema de comunicación con XML-RPC.....	71
Figura 4.4 : segmento de código para el envío de las mediciones de energía hacia el servidor web en formato XML-RPC. ....	72
Figura 4.5 : segmento de código para el envío de las mediciones de potencia hacia el servidor web en formato XML-RPC. ....	72
Figura 4.6 : Lógica de la variable Tabla donde se guardan las mediciones de potencia.....	73
Figura 5.1 : Arquitectura del servidor web. ....	75
Figura 5.2 : Modelo 3 capas. ....	76
Figura 5.3 : Archivos HTML de la capa de presentación.....	76
Figura 5.4 : Inicio de sesión index.html: ....	77
Figura 5.5 : creación de un nuevo usuario imagen del navegador web Chrome. ....	77
Figura 5.6 : Relación del Key y el ID. ....	78
Figura 5.7 : Contenido del directorio cgi-bin.....	78
Figura 5.8 : flujo del negocio de los usuarios. ....	79
Figura 5.9 : graficas en el servidor web vistas desde el navegador Chrome. ....	80
Figura 5.10 : Ejecución del procedimiento remoto AddWatts.cgi con XML-RPC. ....	80
Figura 5.11 : Procedimiento AddWatts.cgi.....	81
Figura 5.12 : script cargar.sh; este guarda las mediciones en la base de datos de RRDTOOL. ....	82
Figura 5.13 : forma de uso de cargar.sh.....	82
Figura 5.14: Tabla meter, esta contiene la información de los medidores. ....	83
Figura 5.15: Estructura de la tabla usuarios. ....	84
Figura 5.16: Estructura de la tabla usuario_x_meter. ....	84
Figura 6.1: Router Dragino. ....	86
Figura 6.2: Conectores para entradas analógicas. ....	87
Figura 6.3: circuitos del router Dragino. ....	87
Figura 6.4 : Medidores conectados en una red Mesh. ....	88

Figura A.1: Router WAP-2102 el cual utiliza Flukso.....	90
Figura A.2 : Router D-LINK DIR-300. ....	91
Figura A.3: Archivo redboot.bat. ....	91
Figura A.4 : archivo redboot1.bat.....	92
Figura A.5: ventana donde se fijara una IP. ....	93
Figura A.6 : servidor de TFTP para Windows.....	94
Figura A.7: Configuración del servidor.....	94
Figura A.8 : Conexión del Router a la PC.....	95
Figura A.9: Respuesta al ejecutar redboot.bat.....	95
Figura A.10: respuesta al ejecutar el paso 11. ....	96
Figura A.11: Ejecución del comando load. ....	96
Figura A.12: Ejecución del comando go. ....	96
Figura A.13: (a) Fijación e la IP estática según se muestra; (b) Abrir una sesión telnet con PUTTY.	97
Figura A.14: consola del router con el prom DD-WRT.....	97
Figura A.15: Comandos para cargar un nuevo firmware.....	98
Figura A.16: Comandos para cargar un nuevo firmware en este caso OpenWrt. ....	99
Figura A.17: interfaz de comunicación web. ....	100
Figura A.18: Consola del router.....	101
Figura A.19: Fijación de una IP estática. ....	102
Figura A.20: Instalación del paquete Flukso. ....	102
Figura A.21: arranque del demonio flukso. ....	102
Figura B.1: Instalación de mysql. ....	103
Figura B.2: configuración de la calve de administrador de mysql. ....	103
Figura B.3: Inicio de mysql server.....	103
Figura B.4: Creación de la base de datos.....	104
Figura B.5: creación de la tabla usuarios.....	104
Figura B.6: Creación de la tabla meter.....	104
Figura B.7: Creación de la tabla usuario_x_meter. ....	104
Figura C.1: Contenido del script graficar.sh. ....	105
Figura C.2: Ejecución del script graficar.sh.....	106
Figura C.3: Graficas Potencia Vs tiempo.....	107
Figura C.4: grafica Energía acumulada Vs tiempo.....	108
Figura C.5: Contenido del script Create_rrd.sh, este crea las bases de datos RRDTOOL. ....	109
Figura C.6: Uso del script create_rrd.sh.....	110
Figura D.1 : (a); (b); (c) y (d) Muestran la instalación del router BULLET 2 en modo AP en el techo de la EIE.....	111
Figura D.2: Creación de la red de medidores alrededor de la EIE.....	112
Figura D.3 : (a) Pinza de corriente; (b) instalación de la Pinza de corriente; (c) Instalación de la pinza en el tablero eléctrico principal de la EIE; (d) Instalación de la pinza de corriente en una fase del tablero.....	113

Figura D.4: (a) Instalación del medidor en el tablero del laboratorio de comunicaciones; (b) Instalación del medidor1 en el edificio C; (b) instalación de la pinza en el tablero del edificio C; (d) instalación del medidor3 en la acometida de la caseta de de la entrada vehicular de la EIE. ....	114
Figura D.5: (a) Graficas obtenidas con las mediciones del medidor1; (b) Graficas obtenidas con el medidor3.....	114
Figura D.6: grafica de potencia versus tiempo obtenidas por el uso de una lavadora. ....	115
Figura D.7: (a) Analizar de carga FLUK; (b) Conexión del analizador a un tablero eléctrico; (c) Tablero eléctrico; (d)Conexión de las pinza del medidor y el analizador FLUK en la misma fase. ...	116
Figura E.1 : topología de red de los medidores.....	118
Figura E.2: Contenido del archivo de configuración wireless.....	119

## Lista de Tablas:

Tabla 2.1: Descripción del hardware del WAP-2102. ....	29
Tabla 2.2 : Mensaje recibido por el puerto UART. ....	39
Tabla 3.1 : Comparación del Router DIR-300 y el WAP-2102. ....	44
Tabla 3.2 : Descripción del PIC16F876A.....	47
Tabla 3.3 : Características del REF-195. ....	51
Tabla 3.4 : Formato de las mediciones a enviar por el puerto UART. ....	58
Tabla D.1: Comparación de las mediciones obtenidas con los medidores construidos y el analizador de carga FLUK. ....	117
Tabla F.1: detalle del costo unitario de los medidores diseñados y construidos. ....	120

## Acrónimos

OPENWRT: OpenWrt es una distribución de Linux basada en firmware usada para dispositivos empotrados tales como routers inalámbricos.

DD-WRT: es un firmware usada para dispositivos empotrados tales como routers inalámbricos.

WIFI: es una marca de la Wi-Fi Alliance, la organización comercial que adopta, prueba y certifica que los equipos cumplen los estándares 802.11 relacionados a redes inalámbricas de área local.

GNU: El proyecto GNU fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre: el sistema GNU.

UART: son las siglas de "Universal Asynchronous Receiver-Transmitter" (en español, Transmisor-Receptor Asíncrono Universal). Éste controla los puertos y dispositivos serie.

RS-232: es una interfaz que designa una norma para el intercambio serie de datos binarios entre un DTE (Equipo terminal de datos) y un DCE (Equipo de Comunicación de datos), aunque existen otras en las que también se utiliza la interfaz RS-232.

BAUDIOS: El baudio es una unidad de medida, usada en telecomunicaciones, que representa el número de símbolos transmitidos por segundo en una red analógica.

PIC: los PIC son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc.

SMD: Traducido al español: Componentes de soldadura superficial.

AC: Corriente alterna.

DC: Corriente directa.

W: Vatios, unidad de potencia eléctrica.

Wh: Vatios Hora, unidad de energía eléctrica.

AP: Punto de acceso, por sus siglas en inglés: Wireless Access Point

STA: modo estación cliente de un router WIFI, por sus siglas en inglés: station/client mode.

EIE: Escuela de ingeniería eléctrica de la Universidad de El Salvador

# 1. Introducción

Los medidores de energía eléctrica existentes en el mercado miden la potencia activa de la red eléctrica. La exactitud de esta medición depende del fabricante del medidor o del uso que se le dará a esta medición. El principio fundamental de la potencia activa es la multiplicación de las señales de corriente y voltaje de la red eléctrica. Ésta se realiza en los circuitos de los medidores. Un ejemplo de circuito integrado que mide la potencia eléctrica es el ADE7757 [1] de *Analog Devices*. Este circuito integrado multiplica las señales de corriente y voltaje en un multiplicador analógico en su interior. De esta manera se obtiene la potencia activa real, la cual se envía por un puerto el cual genera un pulso digital de frecuencia proporcional a la potencia activa medida. A continuación se detalla la formulación matemática para obtener la potencia activa:

$$\text{Voltaje instantáneo: } v(t) = \sqrt{2} * V_{rms} * \sin(2\pi ft) \quad (\text{Ecuación 1.1})$$

$$\text{Corriente instantánea: } i(t) = \sqrt{2} * V_{rms} * \sin(2\pi ft - \theta) \quad (\text{Ecuación 1.2})$$

$\theta$  representa el desplazamiento de la señal de corriente respecto a la de voltaje.

$$\text{Potencia instantánea: } P(t) = v(t) * i(t) \quad (\text{Ecuación 1.3})$$

Sustituyendo 1.1 y 1.2 en 1.3 se obtiene:

$$P(t) = \sqrt{2} * V_{rms} * \sin(2\pi ft) * \sqrt{2} * V_{rms} * \sin(2\pi ft - \theta) \quad (\text{Ecuación 1.4})$$

En donde las raíces de dos se pueden simplificar:

$$P(t) = 2 * V_{rms} * I_{rms} * \sin(2\pi ft) * \sin(2\pi ft + \theta) \quad (\text{Ecuación 1.5})$$

Utilizando la propiedad trigonométrica de los senos la cual se muestra a continuación:

$$\sin(x) * \sin(y) = \frac{1}{2} * [\cos(x - y) - \cos(x + y)] \quad (\text{Ecuación 1.6})$$

Se obtiene:

$$P(t) = \frac{1}{2} * 2 * V_{rms} * I_{rms} * [\cos(2\pi ft - 2\pi ft + \theta) - \cos(2\pi ft + 2\pi ft + \theta)] \quad (\text{Ecuación 1.7})$$

Simplificando la expresión anterior:

$$P(t) = V_{rms} * I_{rms} * \cos(\theta) - V_{rms} * I_{rms} * \cos(4\pi ft + \theta) \quad (\text{Ecuación 1.8})$$

Esta Ecuación muestra que la potencia instantánea  $P(t)$  posee una componente de DC y una componente en la frecuencia del segundo armónico. Se observa que la componente de DC es la potencia activa. Para obtener esta componente se necesita eliminar la componente de alta frecuencia. Ésta componente puede eliminarse por medio de un filtro paso bajo con una frecuencia de corte por debajo de los 60Hz que es la frecuencia de la red eléctrica. La potencia activa se calcula utilizando la formula siguiente:

$$\text{Potencia activa: } P = V_{rms} * I_{rms} * \cos(\theta) \quad (\text{Ecuación 1.9})$$

Donde:

$V_{rms}$  : representa el voltaje RMS.

$I_{rms}$  : representa la corriente RMS.

$\cos(\theta)$  : representa el factor de potencia.

Todos los circuitos integrados dedicados a la medición de la potencia activa necesitan que a sus entradas se les proporcione tanto la señal de voltaje como la señal de corriente. Las magnitudes de estas señales a su vez son adecuadas al nivel de tensión que soportan las entrada de estos circuitos para lo cual se utilizan transformadores reductores de voltaje y transformadores de corriente además de instrumentación electrónica para un adecuamiento correcto de las señales. El adecuamiento de las señales de voltaje y corriente lleva a la utilización de electrónica robusta en los circuitos de los medidores de energía, conduciendo al encarecimiento de estos por el uso de tales componentes.

El medidor inalámbrico de consumo de energía eléctrica de bajo costo no mide con precisión la potencia activa de la red eléctrica. Lo que se medirá será la magnitud de la corriente de la red eléctrica. Esta magnitud se medirá con el transductor de corriente AC a voltaje DC (Pinza de corriente). Con la magnitud de la corriente digitalizada esta se multiplicará por el voltaje del nodo donde se pretende medir el consumo energético, que en este caso será un voltaje de 120V AC. Esta multiplicación se realizará en las rutinas de un microcontrolador PIC16F876A [2], en donde los 120V AC serán representados en un registro del microcontrolador, para obtener la aproximación de la potencia activa como se muestra en la Ecuación 1.10:

$$\text{Potencia} = 120 * ADC \quad (\text{Ecuación 1.10})$$

Donde:  $ADC$  representa el valor de la corriente eléctrica.

Las mediciones se realizaran con el *circuito medidor de potencia v1.3* diseñado y utilizado en el *medidor inalámbrico de consumo de energía eléctrica de bajo Costo* este utiliza la lógica de la ecuación 1.10 para obtener un aproximado de la potencia activa de la ecuación 1.9.

Como se mencionó entre las desventajas de utilizar la lógica de la ecuación 1.10 es que no se toma en cuenta el factor de potencia lo cual nos proporciona en si un aproximado de la potencia activa que según la ecuación 1.10 esta será más próxima al valor real si el voltaje del nodo a medir es el que se programe en el registro del PIC16F876A. Agregado a esto el voltaje 120 VAC entregado a los hogares por la distribuidora eléctrica, no es exactamente este valor sino que varía dependiendo de la zona y de la calidad de la energía entregada por la distribuidora y por el transformador de distribución.

También podemos recurrir a utilizar un registro el cual contenga el factor de potencia de determinada carga, pero este tendría que cambiarse según la carga que se esté midiendo y así minimizar la diferencia entre la potencia activa de la ecuación 1.9 y el aproximado de la ecuación 1.10.

La utilización de la Ecuación 1.10 en el diseño del *circuito medidor de potencia v1.3* nos proporciona las siguientes ventajas:

1. Diseño más económico ya que solo se utiliza un transductor de corriente AC a voltaje DC, dejando de lado la pinza de corriente AC y el transformador reductor de voltaje que utilizan otros circuitos medidores de potencia (ADE7757).
2. Placa del circuito más reducida ya que se utilizaran menos componentes electrónicos, electrónica menos robusta.

El consumo energético se define en vatios hora (Wh), hasta este punto hemos obtenido la potencia de la red eléctrica (Ecuación 1.10) pero falta definir como se medirá la energía, dicho consumo se obtendrá como se detalla a continuación:

$$1Wh = 1W * 1h \quad \text{(Ecuación 1.11)}$$

La Ecuación 1.11 se puede describir como el consumo de una carga de un vatio en una hora, Dicho de otra forma en una hora cierta carga tuvo una demanda promedio de 1W. Ahora el circuito medidor de potencia y energía versión 1.3 es capaz, hasta el momento según lo descrito, de medir la potencia de la carga. En base a esto podemos plantear la ecuación siguiente:

$$EnergiaWh = Potencia * Tiempo \quad \text{(Ecuación 1.12)}$$

Como lo que se requiere es cuantificar la energía consumida esto se puede lograr sumando los Wh que se están consumiendo y guardando en un registro del PIC16F877A, dicho registro aumentara una unidad por cada Wh consumido según la ecuación 1.11 este registro el cual contará los Wh tendrá que ser aumentado en el tiempo en segundos que

una potencia eléctrica demanda consume 1 Wh, todo esto se puede resumir en el algoritmo siguiente:

De la ecuación 1.11 y 1.12 se pueden definir la unidad a cuantificar:

$$EnergiaWh = 1Wh = 1W * 1h \quad (\text{Ecuación 1.13})$$

En donde 1h = 3600 s sustituyendo en 1.12 se obtiene:

$$EnergiaWh = 1Wh = 1W * 3600s \quad (\text{Ecuación 1.14})$$

Sustituyendo la ecuación 1.14 en 1.12 se obtiene

$$1W * 3600s = Potencia [W] * Tiempo[s] \quad (\text{Ecuación 1.15})$$

Despejando el tiempo que es la única incógnita de la ecuación 1.15 y además la variable que nos interesa obtenemos:

$$Tiempo = 1W * \frac{3600[s]}{Potencia [w]} \quad (\text{Ecuación 1.16})$$

Lo que nos proporciona la ecuación 1.16 es el Tiempo en segundos que una Potencia eléctrica demandada por una carga se tarda en consumir un Vatio hora (Wh), para lo cual podremos utilizar los módulos temporizadores del PIC16F877A para conseguir esta temporización, al cumplirse la temporización esto indicará que se ha consumido un 1Wh y esto se podrá cuantificar con un registro en el PIC16F876A.

## 2. Medidor de potencia eléctrica Flukso

Flukso [3] es un medidor de energía eléctrica de nivel residencial, basado en un router. Flukso toma mediciones de la red eléctrica de un hogar y las envía a un servidor web a través de internet por medio de una red WIFI. Se realiza así una telemedición de la energía eléctrica consumida en un hogar. Flukso es una aplicación de software y hardware libre bajo licencia GNU. Esta aplicación utiliza un sitio web donde se descarga las mediciones de potencia y energía eléctrica en intervalos de tiempo variable. Estos tiempos pueden ser: hora, día, semana y año. La interfaz web es utilizada por usuarios que han adquirido un medidor Flukso. Para acceder a esta se debe crear un usuario. En la Figura 2.1 se muestra una imagen de este dispositivo:



Figura 2.1: Medidor Flukso.

En la Figura 2.1 se puede apreciar el medidor Flukso. Éste está construido a partir de un router *WAP-2102* [4]. En la figura se pueden apreciar las dimensiones del dispositivo como se observa es bastante pequeño (7.5 cm de largo por 6 cm de ancho). También se puede apreciar la antena WIFI del medidor.

En la Figura 2.2 se muestra las partes del medidor Flukso. Éste utiliza una pinza de corriente la cual es básicamente un transductor de corriente AC a DC, con una resolución de 1AC/100mVDC. Se observa la fuente de voltaje del dispositivo y el cable RJ-45 el cual se utiliza para conectar el medidor a la PC por medio del puerto Ethernet. En la Figura 2.3 se

puede apreciar la Pinza de corriente (Transductor de Corriente AC a Voltaje DC) que utiliza este medidor.



Figura 2.2: Medidor Flukso con todos sus accesorios.

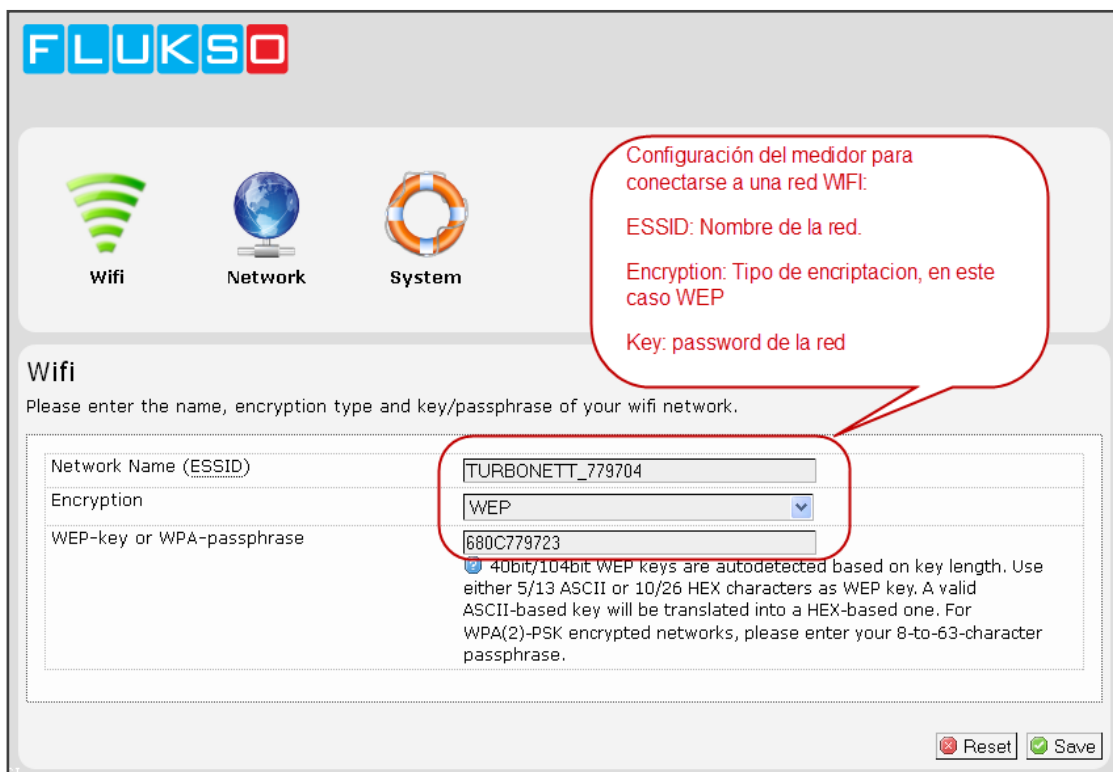


Figura 2.3 : Pinza de Corrientes o Transductor de Corriente AC/DC.

## 2.1 Configuración e instalación del medidor Flukso

Flukso envía las mediciones a un servidor web. Se requiere que el medidor tenga acceso a internet, a través de una red WIFI. El medidor se conecta a una red WIFI en modo AP. Así se envían las mediciones hacia el servidor web donde se podrá monitorizar el consumo energético, donde esté conectado el medidor.

El medidor se puede configurar a través de su interfaz web, a la cual podemos tener acceso si conectamos el medidor a la PC por medio de un cable RJ45. Utilizando un navegador web accedemos a la dirección: 192.168.255.1 (esta es la dirección IP estática para la configuración del medidor Flukso). La interfaz web de configuración del medidor se muestra en la Figura 2.4:



The screenshot shows the Flukso web interface with three main sections: 'Wifi', 'Network', and 'System'. The 'Wifi' section is active and contains a form for configuring a WiFi network. A red callout box points to the form fields with the following text: 'Configuración del medidor para conectarse a una red WIFI: ESSID: Nombre de la red. Encryption: Tipo de encriptacion, en este caso WEP Key: password de la red'. The form fields are: 'Network Name (ESSID)' with the value 'TURBONETT\_779704', 'Encryption' set to 'WEP', and 'WEP-key or WPA-passphrase' with the value '680C779723'. Below the form is a note: '40bit/104bit WEP keys are autodetected based on key length. Use either 5/13 ASCII or 10/26 HEX characters as WEP key. A valid ASCII-based key will be translated into a HEX-based one. For WPA(2)-PSK encrypted networks, please enter your 8-to-63-character passphrase.' At the bottom right are 'Reset' and 'Save' buttons.

Figura 2.4: Interfaz web del medidor Flukso.

En la Figura 2.4 se muestra la interfaz de configuración, los parámetros a configurar son el nombre de la red a conectarse (Network Name o ESSID) la encriptación y el password si lo posee. El medidor se conectara a un punto de acceso AP. Vale la pena resaltar que este viene con figurado para trabajar en modo STA (modo cliente de los router inalámbricos en dónde este se comportan como tarjetas de red.)

Después de configurar el medidor para que se conecte a la red WIFI, la cual podría ser la red WIFI que sirve internet en nuestro hogar, procedemos a conectar el medidor (por medio de la pinza de corriente) en el tablero eléctrico como se muestra en la Figura 2.5.

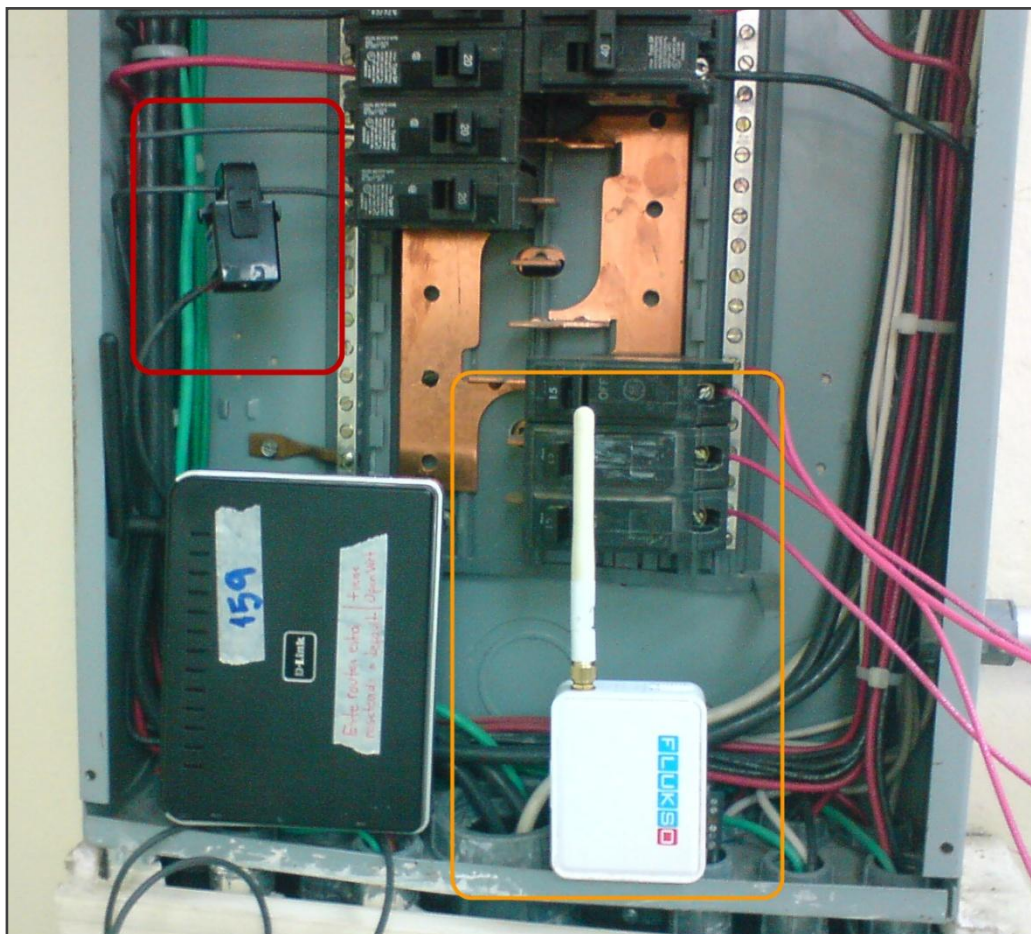


Figura 2.5: Conexión del Medidor Flukso al tablero eléctrico por medio de la pinza de corriente.

En la Figura 2.5 el recuadro de color Rojo muestra la pinza de corriente. Esta se ha conectado a un circuito del tablero, el cual posee aproximadamente 120 V AC. La pinza de corriente de este medidor soporta un máximo de 50 A AC.

## 2.2 Gráficas de las Mediciones en el servidor web

Después de configurar e instalar el medidor este se conecta a internet y envía las mediciones hacia el servidor web [logger.flukso.net](http://logger.flukso.net). Las mediciones se pueden visualizar en la dirección: [www.flukso.net](http://www.flukso.net). Al servidor se accede con nuestro usuario y password. En la Figura 2.6 se muestra un ejemplo donde se puede seleccionar las gráficas de potencia versus tiempo del punto de medición donde se ha conectado el medidor de potencia eléctrica Flukso.

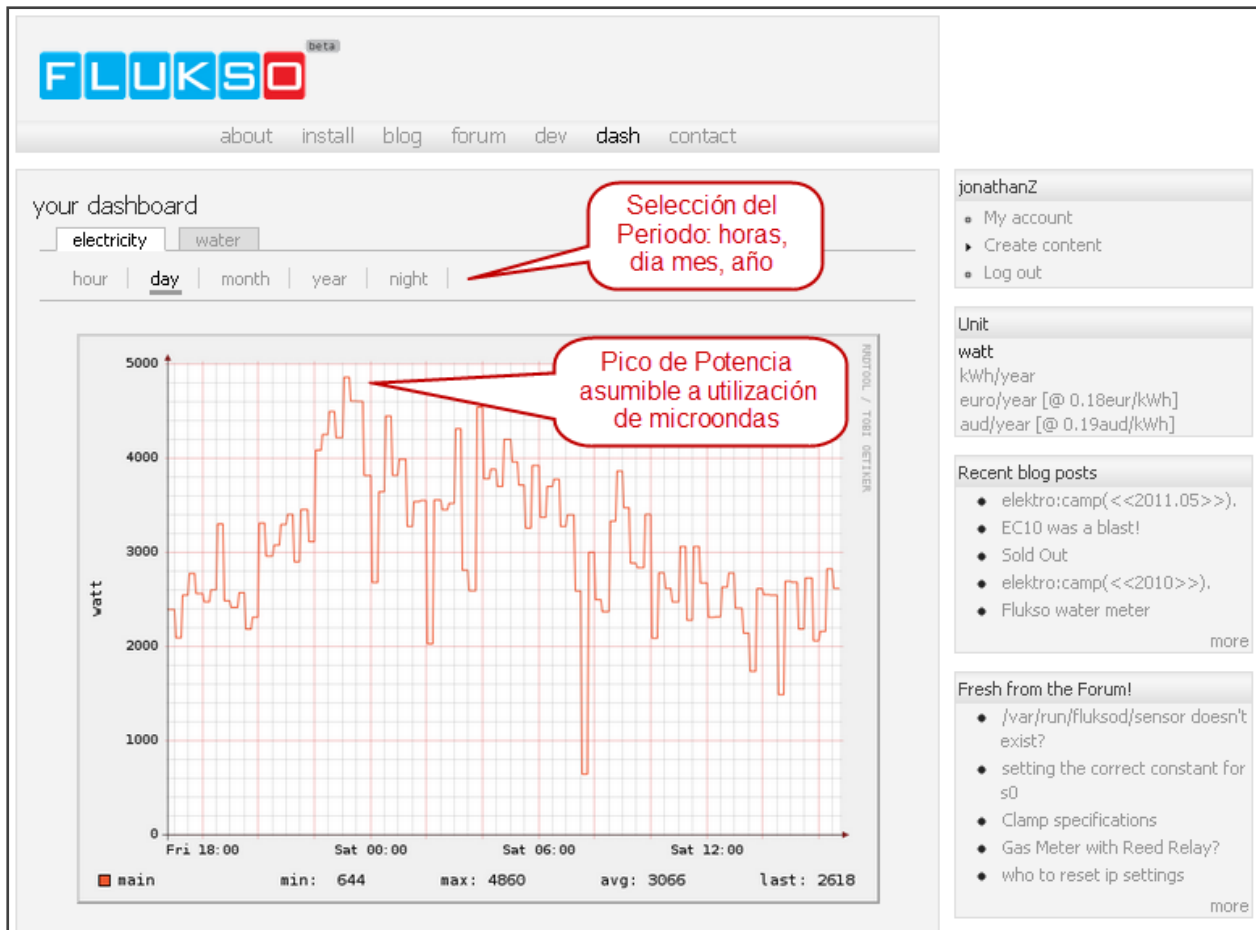


Figura 2.6 : Pagina web del servidor Flukso donde se visualizan las mediciones.

En la Figura 2.6 se puede observar los botones de selección del tiempo de las gráficas. Se puede obtener mediciones de: la última hora, ultimo día, última semana, y mediciones de un año completo. Las gráficas muestran: el valor máximo de potencia, el valor mínimo de potencia y el promedio de la potencia eléctrica. El tipo de gráfica que se observa en la Figura 2.6 es generada por un software GNU el cual se llama RRDTOOL [5]. En este programa se puede crear bases de datos. El contenido de estas se puede graficar. También, con esta herramienta se puede obtener el promedio de los datos en un intervalo de tiempo especificado.

Las gráficas potencia versus tiempo son un estimado del valor real. El medidor Flukso solo muestrea la corriente de la red eléctrica, fijando el voltaje en una variable en el microcontrolador del sistema. Por tanto, la gráfica mostrada en la Figura 2.6 muestra una aproximación de la potencia eléctrica en el punto de medición. Pero, vale la pena resaltar que la filosofía del medidor Flukso es: Poder Ahorrar energía eléctrica monitoreando el consumo en nuestro hogar. Esto se puede realizar a partir de las gráficas mostradas. El usuario puede observar los picos de potencia y la demanda que está teniendo en su hogar.

Con esta información se esperaría que el usuario mejorara su hábito de consumo de energía eléctrica. Un ejemplo de esto se puede realizar con la Figura 2.6 en donde se muestra un pico de potencia el cual se debe a la utilización de un microondas. Se espera que con dicha información el usuario pueda moderar el tiempo de uso de dicho microondas.

### 2.3 Análisis del hardware y del software de Flukso

Como se mencionó anteriormente el medidor de energía eléctrica Flukso está basado en un router (WAP-2102). Utilizado como sistema embebido, a este se le conecta por medio del puerto UART (UART son las siglas de "Universal Asynchronous Receiver-Transmitter" en español, Transmisor-Receptor Asíncrono Universal, este controla los puertos y dispositivos serie bajo el protocolo RS-232.) un circuito que es capaz de estimar la potencia eléctrica.

En la Figura 2.7 se muestra el diagrama de bloques del sistema del medidor Flukso. El sistema en general se puede dividir en 4 subsistemas cada uno de ellos con una función específica. El subsistema 1 es la red eléctrica en sí, que en este caso puede ser la red eléctrica de un hogar. Ésta tiene varias fases a 120V AC. Los sistemas 2 y 3 representan el medidor. El sistema 2 corresponde al circuito el cual muestrea la corriente de la red eléctrica por medio de un micro controlador ATMEGA168 [6]. El sistema 3 representa el router (WAP-2102) o sistema embebido el cual se comunica con el microcontrolador del sistema 2, por medio del puerto UART del router. Las funciones del sistema 3 (router) son la comunicación con el circuito que mide la potencia (sistema 2) y el envío de estas mediciones hacia el servidor web por medio de una red WIFI. Finalmente el sistema 4 representa el servidor web en donde se pueden graficar las mediciones proporcionadas por el medidor de energía eléctrica Flukso.

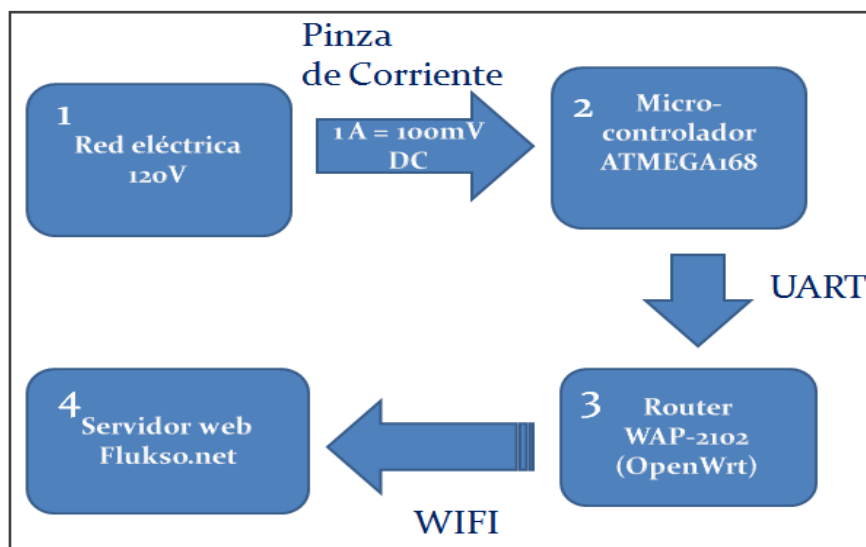


Figura 2.7: Diagrama de Bloques del sistema del medidor Flukso.

En la Figura 2.8 se muestra el interior del dispositivo Flukso, se resalta el sensor medidor de potencia (sistema 2 de la Figura 2.7). La Figura 2.8 también muestra un jumper en donde se conecta la pinza de corriente.

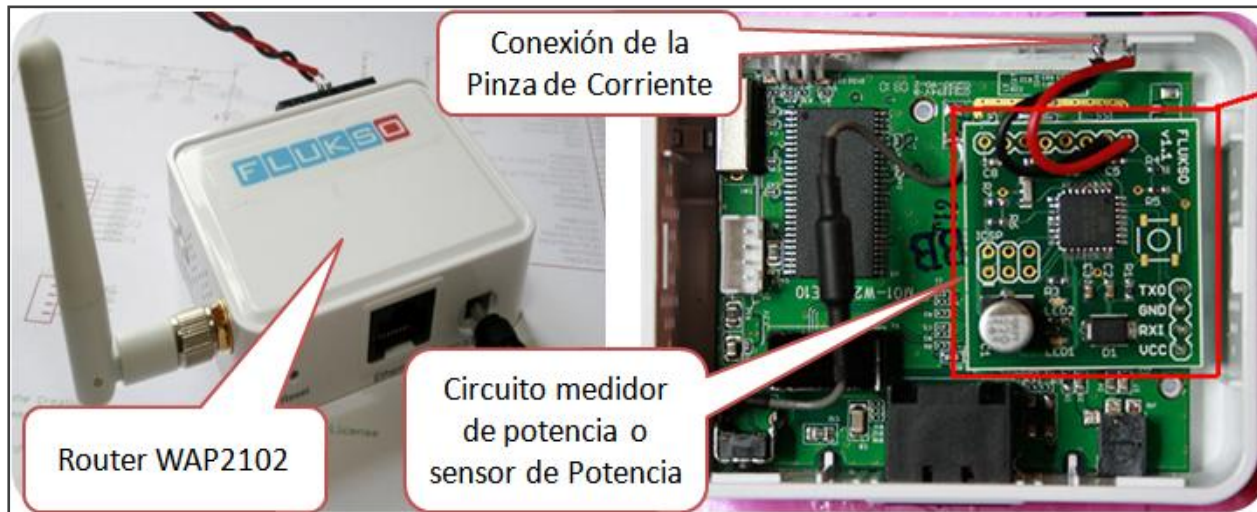


Figura 2.8 : Interior del medidor Flukso donde se observa el sensor medidor de potencia.

Como ya se mencionó, en la Figura 2.8 el recuadro rojo en marca el sensor medidor de potencia. Este no forma parte del router en sí. El router utilizado es el WAP-2102 de Abocon [7]. Dicho router tiene instalado OpenWrt [8], con una versión compilada por el fabricante de Flukso. La comunicación del sensor medidor de potencia y el router se realiza a través del puerto UART.

En la Figura 2.9 se presenta el esquemático del circuito que mide la potencia en el medidor Flukso. Como se mencionó anteriormente, el circuito utiliza el microcontrolador ATMEGA168 el cual muestrea en una de sus entradas analógicas el voltaje que proporciona un transductor de corriente AC a voltaje DC (Pinza de corriente). Como se observaba el circuito básicamente realiza una conversión analógica digital de la corriente de la red eléctrica luego esta conversión es multiplicada en el firmware del microcontrolador por el valor de voltaje en el punto a medición, que en este caso es de 120V AC y así se obtiene un estimado de la potencia activa.

El circuito que se encarga de las mediciones de potencia es bastante sencillo. Se basa fundamentalmente en el microcontrolador ANTEGA168 y unos pocos componentes pasivos (resistencias y condensadores). Por otro lado solo se está tomando en cuenta la corriente de la red eléctrica y no se muestrea el voltaje. Además la pinza de corriente es un transductor de corriente AC a voltaje DC resultando así en una aproximación de la potencia eléctrica de la red.

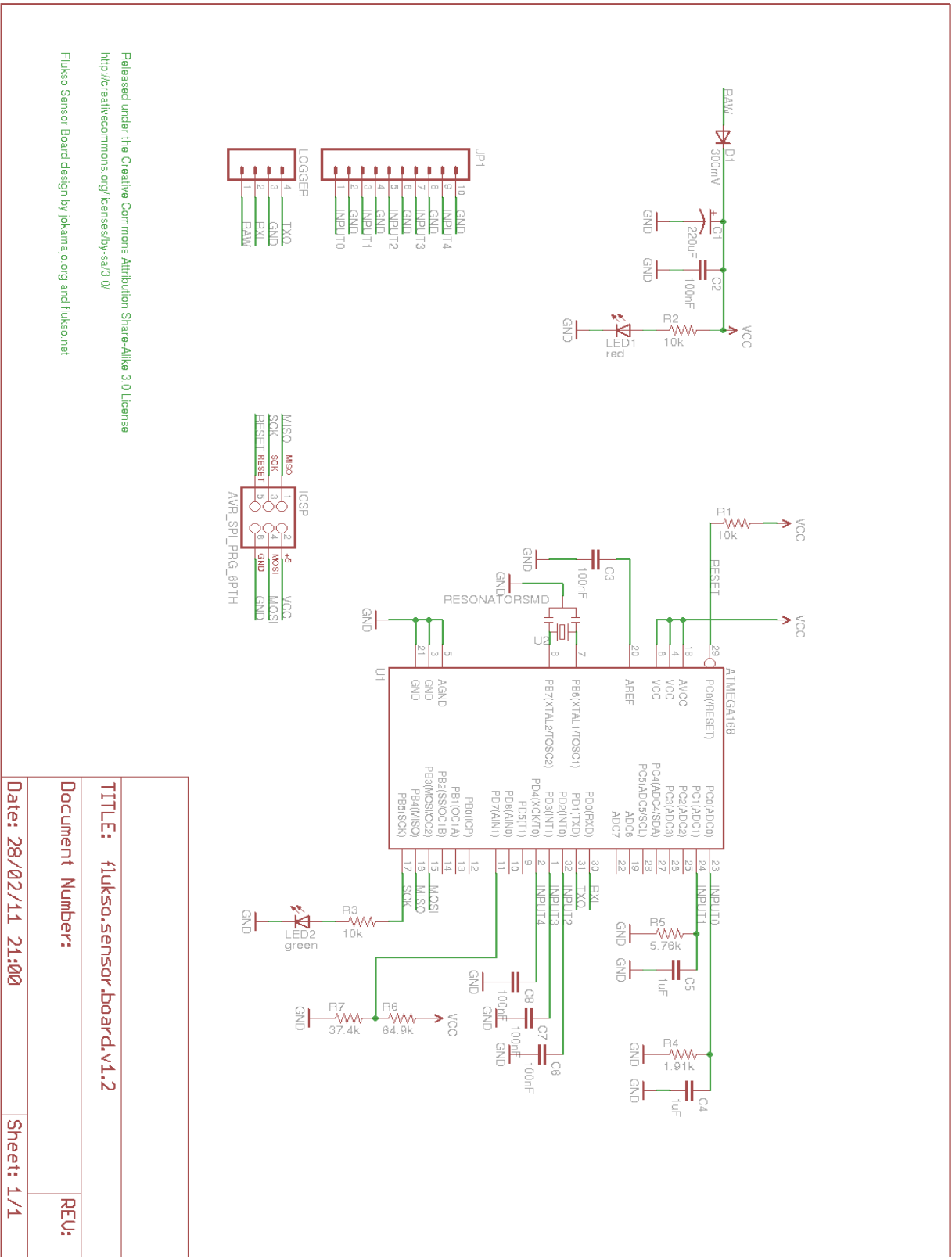


Figura 2.9: Sensor medidor de Potencia Fluxo V 1.2.

Released under the Creative Commons Attribution Share-Alike 3.0 License  
<http://creativecommons.org/licenses/by-sa/3.0/>  
 Fluxo Sensor Board design by jokamajo.org and fluxo.net

TITLE: fluxo-sensor.board.v1.2	
Document Number:	REV:
Date: 28/02/11 21:00	Sheet: 1/1

Hablar del software del Flukso es hablar del sistema operativo OpenWrt y los diferentes programas que se ejecutan en él. El router originalmente no poseía el sistema operativo OpenWrt. Fue el fabricante del Flukso quien instaló este sistema operativo. Así se conseguirá la potencia de un sistema embebido con un sistema operativo Linux. Esto resulta económicamente más barato que adquirir una tarjeta embebida WIFI [9]. La idea de utilizar un router personal como sistema embebido no es nueva. Ésta idea ha sido utilizada en otros proyectos como lo son: *La Fonera* [10], la cual es router WiFi 802.11b/g basado en Linux 2.4, con un firmware OpenWrt; El *mesh-potato* [11] este es un dispositivo basado en un router el cual utiliza OpenWrt y su función principal es ofrecer telefonía IP e Internet a bajo costo, en zonas donde el acceso es remoto y no existe cobertura de estos servicios o es demasiado caro. El hardware y software de ambos proyectos es bajo licencia GNU.

El medidor Flukso al igual que los proyectos descritos utiliza OpenWrt, para poder apreciar la capacidad de este sistema operativo instalado en el router del medidor Flukso se muestra a continuación la capacidad del hardware del *WAP-2102*:



Figura 2.10: WAP-2102 de Abocon.

Router	CPU	Ram	Flash	Network	USB	Serial	JTag
<b>WAP2102</b>	Atheros AR2317@182MHz	16MB	4MB	1	No	Yes	No

Tabla 2.1: Descripción del hardware del WAP-2102.

La Tabla 2.1 muestra la capacidad del router WAP-2102. Este posee 4MB de memoria Flash y es en esta donde se instala el sistema operativo OpenWrt. La capacidad de la CPU es de 182 MHz. Se muestra en la Tabla 2.1 que este router posee como periféricos un puerto

serial por donde se comunicará el circuito que mide la potencia (según la figura 2.7). Posee un puerto Ethernet y comunicación vía WIFI. Para comenzar con el análisis del software conectamos el medidor a la PC por medio del cable RJ-45, como se muestra en la Figura 2.11:



Figura 2.11 : conexión del medidor Flukso con la PC.

Posterior a la conexión del Flukso con la PC, asumiendo que la PC utiliza como sistema operativo Linux en este caso Ubuntu 11.10, abrimos un terminal del sistema y realizaremos una conexión *ssh*<sup>1</sup> (Secure Shell, en español: intérprete de órdenes segura). Ingresamos como usuario *root* en el terminal de Ubuntu y realizamos la conexión *ssh* a la dirección 192.168.255.1 que es la IP estática definida en el medidor Flukso para realizar configuraciones al mismo.

En la Figura 2.12 se muestra la imagen del terminal donde se observan los comandos utilizados para la conexión *ssh* hacia el Flukso, recordemos que al ser el router un sistema embebido este no posee periféricos (teclado, monitor). Por ello para ingresar y visualizar el sistema operativo Linux que se ejecuta en el Flukso accedemos a este de una manera remota por medio de la red Ethernet de éste. Así se utilizan los periféricos de la PC para realizar la programación y configuración que se requiera en el Flukso.

---

<sup>1</sup> SSH (Secure SHell, en español: intérprete de órdenes segura) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos.



```

/#: Es la raíz del sistema de directorios. Aquí se monta la partición
principal Linux EXT
/etc: Contiene los archivos de configuración de la mayoría de los programas
/home/: Contiene los archivos personales de los usuarios
/bin/: Contiene comandos básicos y muchos programas
/dev/: Contiene archivos simbólicos que representan partes del hardware,
tales como discos duros, memoria...
/mnt/: Contiene subdirectorios donde se montan (se enlaza con) otras
particiones de disco duro, CDROMs, etc.
/tmp/: Ficheros temporales o de recursos de programas
/usr/: Programas y librerías instalados con la distribución
/usr/local/: Programas y librerías instalados por el administrador
/sbin/: Comandos administrativos
/lib/: Librerías varias y módulos del kernel
/var/: Datos varios como archivos de log (registro de actividad) de
programas, bases de datos, contenidos del servidor web, copias de
seguridad...
/proc/: Información temporal sobre los procesos del sistema
/www/: Páginas web y CGI.

```

Figura 2.14 : Directorios en OpenWrt.

Un directorio muy interesante es el *www* este directorio en un sistema Linux contiene ficheros web o páginas web *html*. Este mismo directorio contiene el subdirectorio *cgi-bin* el cual es utilizado para almacenar las CGI<sup>2</sup>. Normalmente el directorio *www* lo utiliza el servidor web para contener las páginas web y las CGI. El medidor Flukso utiliza el servidor web *uhttp*.

Como se mostró en la Figura 2.4 para poder configurar el Flukso accedemos a la interfaz web de éste, por medio de la red LAN. Los archivos fuente de esta interfaz web se encuentran en el directorio *www*. El contenido de este directorio en el Flukso se muestra en la Figura 2.15, donde podemos acceder al directorio por medio del comando *cd* y visualizar su contenido con el comando *ls*.

```

root@flukso:/# cd /www/
root@flukso:/www# ls
cgi-bin      index.html  jquery      logos        luci-static  resources
root@flukso:/www#

```

Figura 2.15 : Directorio WWW el cual contiene los archivos HTML y CGI que utiliza un servidor web.

No se pretende en esta sección describir todos los programas que se ejecutan en el sistema operativo de Flukso, sino más bien describir los más importantes. OpenWrt es un sistema operativo genérico para router. Al ser un sistema operativo abierto queda en la creatividad

---

<sup>2</sup> Interfaz de entrada común (en inglés Common Gateway Interface, abreviado CGI) es una importante tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web.

del programador y desarrollador poder utilizar el sistema para cualquier proyecto, fue este el caso con el proyecto Flukso. Flukso desarrolló aplicaciones para que el router pudiera comunicarse a través puerto serie con un circuito el cual estima la potencia de una red eléctrica.

El programa *flukso.lua* es una aplicación daemon<sup>3</sup> desarrollada en un lenguaje de programación LUA [12]. Éste lenguaje de programación es del tipo interpretado como Perl o Phyton. El programa *flukso.lua* en si es un script que se interpreta y ejecuta por medio de la maquina virtual de LUA. La versión de la maquina virtual de LUA que utiliza Flukso se muestra en la Figura 2.16. Se observa que esta es la versión *Lua 5.1.4*. La mayoría de programas que utiliza OpenWrt están programados en LUA. Éste es el intérprete entre las aplicaciones y el código ensamblador del chipset Atheros AR2317 del router [13].

```
root@flukso:/# lua
Lua 5.1.4 Copyright (C) 1994-2008 Lua.org, PUC-Rio (double int32)
>
```

Figura 2.16 : Ejecución del intérprete o maquina virtual de LUA en el Flukso.

El script *flukso.lua* es el encargado de la comunicación por el puerto serie UART (RS-232) del router hacia el circuito que mide la potencia eléctrica. También este script es el encargado de el envío de las mediciones por medio de la red WIFI hacia Internet.

```
root@flukso:/# cd /usr/share/lua/flukso/
root@flukso:/usr/share/lua/flukso# ls
auth.lua      dbg.lua      heartbeat.lua
data.lua      flukso.lua  restful.lua
root@flukso:/usr/share/lua/flukso#
```

Figura 2.17 : Directorio en donde se encuentran alojados los script de Flukso en OpenWrt.

Los script programados para el medidor Flukso se encuentran alojados en dirección: */usr/share/lua/Flukso/*. Como se puede apreciar en la Figura 2.17 es en este directorio en donde se alojan los programas credos por el desarrollador del Flukso.

El código fuente del script *flukso.lua* se puede encontrar en el CD de anexo en */Flukso/programas/*. Como se mencionó anteriormente el script *flukso.lua* es un demonio el cual se ejecuta en un segundo plano, esperando las mediciones que son enviadas por el

---

<sup>3</sup> Un demonio, daemon o dæmon (de sus siglas en inglés Disk And Execution Monitor), es un tipo especial de proceso informático no interactivo, es decir, que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario. Este tipo de programas se ejecutan de forma continua (infinita), vale decir, que aunque se intente cerrar o matar el proceso, este continuará en ejecución o se reiniciará automáticamente. Todo esto sin intervención de terceros y sin dependencia de consola alguna.

puerto serie UART del circuito sensor de potencia eléctrica. Los programas que ejecutamos como demonios pueden estar ubicados en cualquier fichero del sistema, pero tienen un punto en común: todos utilizan un script para ser iniciados o detenidos, estos scripts se encuentran en el directorio: */etc/init.d/*. El script que inicia o detiene el demonio *flukso.lua* se encuentra en el fichero mencionado como se puede apreciar en la Figura 2.18. En este caso el script posee el mismo nombre que el demonio: *flukso*.

```
root@flukso:/#  
root@flukso:/# cd /etc/init.d/  
root@flukso:/etc/init.d# ls  
avahi-daemon      done                luci_dhcp_migrate  umount  
batmand           dropbear           network            usb  
boot              firewall           rcS                watchdog  
cron              flukso             sysctl  
dnsmasq          led                uhttpd  
root@flukso:/etc/init.d#
```

Figura 2.18 : Fichero */etc/init.d/* donde se encuentran los script que inician o detienen los demonios en Linux.

Además del demonio *flukso* como se observa en la Figura 2.18, existen otros demonios que utiliza OpenWrt. Uno de estos demonios es el *uhttpd* el cual es un servidor web el cual se describirá a continuación.

OpenWrt 10.03 utiliza el servidor web *uhttpd* [14]. Éste contiene las interfaces web o paginas HTML utilizadas en OpenWrt en el Directorio */www/*. El servidor web *uhttpd* o servidor HTTP (Hypertext Transfer Protocol o HTTP, en español protocolo de transferencia de hipertexto) es capaz de ejecutar CGI programadas en LUA las cuales son almacenadas en el directorio */www/cgi-bin/*. La interfaz de configuración web de Flukso la cual se muestra en la Figura 2.4, es el resultado de ejecutar en el servidor web *uhttpd* la CGI que se encuentra en el directorio */cgi-bin/wifi*. La ejecución de la CGI se realiza por medio de un navegador web en la dirección *192.168.255.1/cgi-bin/wifi*.

El router que utiliza Flukso no posee en su hardware un reloj de tiempo real. Cuando el sistema operativo OpenWrt se reinicia pierde la configuración del reloj. El reloj o el tiempo en el sistema operativo OpenWrt se establece por medio de el programa *ntpclient* [15]. Este se encarga de la sincronización del reloj del sistema operativo realizando una petición a un servidor de NTP (Network Time Protocol). NTP es un protocolo de Internet para sincronizar los relojes de los sistemas informáticos a través de ruteo de paquetes en redes.

El *ntpclient* realiza una petición al servidor de NTP el cual está definido en el archivo de configuración que se encuentra en el fichero: */etc/config/ntpclient*, el contenido del archivo de configuración se muestra a continuación en la Figura 2.19:

```

config ntpserver
    option hostname '0.openwrt.pool.ntp.org'
    option port     '123'

config ntpserver
    option hostname '1.openwrt.pool.ntp.org'
    option port     '123'

config ntpserver
    option hostname '2.openwrt.pool.ntp.org'
    option port     '123'

config ntpserver
    option hostname '3.openwrt.pool.ntp.org'
    option port     '123'

config ntpdrift
    option freq     '0'

config ntpclient
    option interval 600
    #option count   10
    #option interface wan

```

Figura 2.19: Archivo de configuración del programa *ntpclient*.

En el archivo de configuración del *ntpclient* se puede definir el servidor NTP con el cual se sincronizará la hora. Para el ejemplo de la Figura 2.19 es el servidor: *0.openwrt.pool.ntp.org*. La sincronización de la hora se realiza en intervalos de 600 segundos, como se muestra en los campos del archivo de configuración. La hora se sincroniza en formato UTC<sup>4</sup>. El *ntpclient* es un daemon al igual que los programas *flukso.lua* y el *uhttpd*, estos se ejecutan en un segundo plano en el sistema operativo.

Es importante recalcar que para que la sincronización de la hora del sistema operativo OpenWrt del Flukso se realice, éste debe de poseer acceso a internet para poder realizar dicha sincronización con el servidor NTP. Por tanto, si el medidor Flukso es conectado y este no posee acceso a internet, cuándo el sistema operativo OpenWrt inicia este intentará sincronizar la hora pero no se podrá sincronizar, ya que no se podrá comunicar con el servidor de NTP.

Para comprobar que la hora no se sincroniza cuando no se tiene acceso a internet se conecto el medidor Flukso sin acceso a internet. Se ejecutó el comando *date* en el terminal remoto como se puede apreciar en la Figura 2.20 la respuesta a este comando es la hora actual del sistema operativo en este caso la hora es: las cero horas del 1 de enero de

---

<sup>4</sup> El tiempo universal coordinado, o UTC, en español, es el tiempo de la zona horaria de referencia respecto a la cual se calculan todas las otras zonas del mundo.

1970(Tiempo Unix: se define como la cantidad de segundos transcurridos desde la medianoche UTC del 1 de enero de 1970, sin contar segundos intercalares).

```
root@flukso:/#  
root@flukso:/# date  
Thu Jan 1 00:00:05 UTC 1970  
root@flukso:/#
```

Figura 2.20 : conexión con un terminal remoto al medidor Flukso, ejecución del comando date.

El tiempo en un medidor tipo Smart-Meter, es una variable muy importante [16]. Este tipo de medidores de consumo de energía eléctrica, identifican el tiempo que se realizan las mediciones. Este es el caso del medidor Flukso el cual envía cada 5 minutos las mediciones de potencia eléctrica hacia el servidor web. Estas mediciones poseen cada una su respectiva marca de tiempo para poder ser graficadas en el servidor web según la Figura 2.6. Si el medidor Flukso no poseyera una hora adecuada este enviaría las mediciones con marcas de tiempo erróneas. No sería posible identificar un evento ocurrido a determinada hora en la red eléctrica donde se encuentra conectado el medidor. Es por ello que es de vital importancia tener una buena sincronización del reloj del sistema operativo. El medidor Flukso no posee un reloj de tiempo real en su hardware. Por tanto siempre depende de estar conectado a internet para la sincronización del reloj.

Los archivos de configuración para los principales programas, incluyendo los descritos en este capítulo se encuentran en el directorio `/etc/config/` el contenido de este directorio se muestra en la Figura 2.21:

```
root@flukso:/# cd /etc/config/  
root@flukso:/etc/config# ls  
batmand      firewall     network     timeserver  watth  
dhcp         flukso      ntpclient   ucitrack    watts  
dropbear    luci        system      uhttpd      wireless  
root@flukso:/etc/config#
```

Figura 2.21 : Directorio `/etc/config/` en donde se alojan las configuraciones de las aplicaciones en OpenWrt.

Los archivos contenidos en este directorio son simples archivos de texto plano. Éstos siguen una estructura de finida en OpenWrt según el modelo UCI [17]. En estos se guardan las configuraciones del sistema y de las principales aplicaciones. Algunos archivos de configuración que se encuentran en este directorio son:

- Network: Configuraciones de las interfaces de red.
- Wireless: configuraciones de topologías de red como por ejemplo: punto acceso.

- Ntpclient: Configuraciones de los servidores NTP con los cuales se sincronizara la hora.
- Uhttpd: configuración del servidor web, configuraciones de puertos.
- Flukso: en este archivo se encuentran las configuraciones de el medidor Flukso, como lo son configuraciones del ID (identificador del dispositivo), configuración de servidor web al cual se le enviarian las mediciones etc.

El contenido del archivo de configuración del medidor Flukso (archivo flukso según la Figura 2.21) se muestra a continuación en la Figura 2.22:

```

config flukso main
    option port /dev/ttyS0
    option home logger.flukso.net
    option homeVersion 1
    option homeInterval 300
    option homeEnable 1
    option localDir /tmp/sensor
    option localVersion 1.0
    option localInterval 0
    option localEnable 1
    option debug 0
config sensor 1
    option id 2d4f76edc4dcf577fa0a49b833513bfe
    option input analog
    list port 1
    option voltage 120
    option current 50

```

Figura 2.22 : contenido del archivo de configuración flukso.

En el contenido del archivo *flukso* se encuentran las configuraciones del servidor web al cual se enviarian las mediciones. En este caso es el campo: *option home logger.flukso.net*, así también es importante resaltar el campo de configuración del ID del dispositivo: *option id 27f8cbad4babe36ecc67e8179f6fc289*. El archivo de configuración *flukso* como también los otros archivos de configuración siguen la estructura siguiente:

```

config 'name' 'flukso'
    option 'configuracion1' 'valor'
    option 'configuracion2' 'Valor'
    option 'configuracion3' 'Valor'

```

Figura 2.23 : estructura de los archivos de configuración en OpenWrt.

Hay que observar primero la diferencia entre *config* y *option*. *config* se refiere a cada una de las entradas que tiene el fichero y *option* son cada uno de los campos de los que está

compuesta cada entrada del fichero. Cada campo option posee su respectivo campo de configuración (configuracion1) y el valor de la configuración (valor). OpenWrt posee herramientas para la manipulación de estos archivos tanto desde scripts LUA como desde consola de comandos. Ésto hace muy versátil la escritura y borrado de los campos de estos archivos, tanto para las aplicaciones como para el usuario desde la consola de comandos.

## 2.4 Comunicación Microcontrolador-Router

Como se mencionó anteriormente el medidor de energía eléctrica Flukso envía las mediciones por medio de WIFI. Las mediciones de la potencia de la red eléctrica son realizadas por el circuito medidor de potencia eléctrica o sensor de potencia. Este se comunica con el router por medio del puerto UART.

En un sistema Linux los periféricos son vistos como archivos especiales de dispositivos. Cada periférico aloja su fichero especial en el directorio */dev/*. El sistema operativo del Flukso trabaja de la misma manera. En este directorio se encuentra el archivo del dispositivo del puerto serie del router (WAP-2102) *ttyS0* [18]. Si el sistema poseyera dos puertos seriales el segundo se denominaría *ttyS1*. El router que utiliza Flukso posee un puerto serie y el archivo especial del dispositivo se encuentra en el directorio */dev/ttyS0*. Al analizar el medidor Flukso se puede observar que el sensor de potencia envía las mediciones tomadas cada segundo. Esto se puede corroborar si revisamos el archivo especial del dispositivo serie. La Figura 2.24 muestra el resultado de la ejecución del comando *cat /dev/ttyS0*. La respuesta de este comando es la impresión de las mediciones de potencia y energía eléctrica en la consola. Estas llegan al puerto serie UART del router, enviados por el sensor de potencia. Como se puede apreciar poseen un formato que se describe en la Tabla 2.2.

```
root@flukso:/# cat /dev/ttyS0
pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
pls 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000500
pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
pwr 2d4f76edc4dcf577fa0a49b8335c3bfe:0000000300
^C
root@flukso:/#
```

Figura 2.24 : Consola de comandos en donde se ejecuta el comando cat en el fichero especial *ttyS0*.

<b>pwr</b>	<b>2d4f76edc4dcf577fa0a49b833513bfe</b>	<b>:0000000000</b>	<b>0A</b>
<b>pls</b>	2d4f76edc4dcf577fa0a49b833513bfe	:0000000000	0A
<b>Cadena Inicial, la cual distingue el mensaje.</b>	Identificador del medidor, este es necesario para que el servidor web reconozca que medidor envía las mediciones.	En esta parte del mensaje irán las mediciones de potencia o energía obtenidas.	El mensaje termina con un avance de línea.

Tabla 2.2 : Mensaje recibido por el puerto UART.

Según la Tabla 2.2, los mensajes enviados por el sensor de potencia hacia el router poseen el formato donde las cadenas iniciadas con *pwr* representan las mediciones de potencia eléctrica en vatios (W). Por otra parte las cadenas que inician con *pls* representan las mediciones de energía eléctrica consumida en vatios hora (Wh). Como se puede observar de la Tabla 2.2 el sensor de potencia envía el identificador del dispositivo ID el cual también está definido en los archivo de configuración del router.

## 2.5 Ventajas y desventajas del medidor Flukso

A lo largo de este capítulo hemos descrito el funcionamiento del medidor de energía eléctrica Flukso. También se ha analizado tanto el hardware como el software de éste. El medidor Flukso no es un medidor de energía eléctrica preciso. Pues éste no toma en cuenta la señal de voltaje en el punto de medición. Sin embargo, Flukso no fue diseñado para tomar una medición precisa de la potencia activa. Flukso realiza mediciones aproximadas del consumo energético por medio del análisis de las gráficas de potencia versus tiempo que se obtienen en el servidor web. Las gráficas de la estimación de la potencia activa que se obtienen con este medidor se pueden apreciar en la Figura 2.6. Luego de la descripción de este medidor se puede denotar las ventajas y desventajas de este para poder así realizar un análisis de la factibilidad de la construcción de este medidor y tomar en cuenta esto para el diseño del *Medidor Inalámbrico de Consumo de Energía Eléctrica de Bajo Costo* que es el objetivo principal de este trabajo.

### 2.5.1 Ventajas de Flukso

Las principales ventajas que se han encontrado en el diseño medidor Flukso son:

1. La utilización del router como sistema embebido para el envío de las mediciones por medio de la red WIFI.
2. Utilización del sistema operativo Linux OpenWrt 10.03 lo cual hace que la programación de aplicaciones sea más versátil.
3. la utilización de los puertos de comunicación del router.

4. Espacio reducido del sensor medidor de energía o circuito que mide la potencia sistema 2 de la Figura 2.7 y circuito de la Figura 2.9.
5. Diseño más económico por no poseer electrónica robusta, ya que solo se utiliza un microcontrolador y pocos elementos pasivos.
6. Voltaje reducido del sensor medidor de potencia 3.3V.
7. Fácil instalación por no tomar en cuenta la señal de voltaje en el punto de medición, y solo utilizar una pinza de corriente. La factibilidad radica en que el usuario no tendrá que colocar puntas de medición de voltaje, sino que solo instalará la pinza de corriente.
8. Las gráficas en el servidor web resultan una manera de ahorrar energía eléctrica, por la fácil identificación de eventos en el sistema eléctrico por medio de la interpretación de estas gráficas.
9. por utilizar un router con el sistema operativo OpenWrt como sistema para el envío de las mediciones vía WIFI, se puede realizar otras topologías de red inalámbricas (WIFI) con este medidor (por defecto el medidor viene configurado en modo cliente o *STA* para ser conectado en punto de acceso *AP*).

### **2.5.2 Desventajas de Flukso.**

Las desventajas encontradas en el análisis del medidor de energía eléctrica Flukso son:

1. Mediciones inexactas por no tomar en cuenta el voltaje ni la corriente AC, el medidor Flukso utiliza una pinza de corriente que es un transductor de corriente AC a Voltaje DC.
2. Por fijar un valor de voltaje y utilizar un transductor de corriente AC/DC (pinza de corriente) no se toma en cuenta el factor de potencia.
3. Solo se puede conectar a nodos de un solo valor de voltaje, debido a que el valor de voltaje es programado en el microcontrolador en donde se multiplica por el estimado de la corriente obtenido por la pinza de corriente.
4. El medidor Flukso visto como sistema integrado no posee un reloj de tiempo real, ya que ni el circuito que mide la potencia ni el router poseen un reloj de tiempo real en hardware. Esto hace que al desconectar la fuente de alimentación de voltaje DC o reiniciar el medidor este pierda la hora sincronizada,

5. La dependencia de que el medidor tenga siempre acceso a internet (para la sincronización de la hora y el envío de las mediciones al servidor web) se vuelve una desventaja si se quisiera utilizar en una red local en donde esta no posea acceso a internet.
6. Muchos medidores de este tipo poseen un registro de las mediciones tomadas, al cual se puede acceder de forma local sin tener que ingresar el servidor web, en este caso por no poseer un registro de las mediciones en la memoria del medidor este no podrá ser conectado en determinado hogar que no posea acceso a internet o si este ha perdido el acceso temporalmente de este servicio.

### 3. Diseño y construcción del medidor inalámbrico de consumo de energía eléctrica de bajo costo

En el capítulo anterior se analizó el medidor de potencia y energía eléctrica Flukso. Éste fue tomado como base de diseño en dos trabajos previos [19] [20]. En esos trabajos se tomó como base de diseño la lógica que utiliza el medidor Flukso. Tomando como base esta lógica se diseñaron y construyeron dos prototipos de medidores de energía eléctrica. Para cada uno de estos medidores se diseñaron dos circuitos que estimaban la potencia y energía eléctrica: *Circuito medidor de potencia V1.1* y *circuito medidor de potencia V1.2*, respectivamente.

Tomando como base el medidor Flukso se desarrolló un nuevo diseño, que mejorara algunas de sus desventajas. Para comenzar a abordar el nuevo diseño fue necesario plantear una metodología de trabajo. La metodología que se utilizó se puede resumir en el esquema de la Figura 3.1:

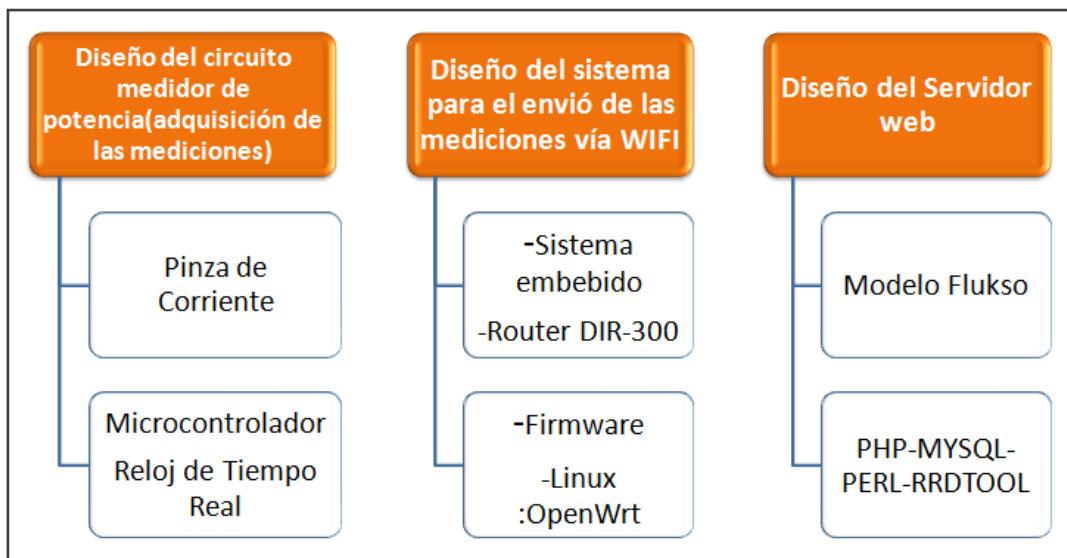


Figura 3.1 : Metodología de diseño para el medidor de energía WIFI de bajo costo.

El diseño del medidor se dividió en tres partes. Primero está el *Diseño Circuito del medidor de Potencia* cuya lógica utiliza la pinza de corriente del medidor Flukso. Para obtener la corriente de la carga ésta se muestreá con un microcontrolador (PIC16F876A). Se mejoró parte de las desventajas del medidor Flukso, incorporando un reloj de tiempo real. Este reloj de tiempo real vendrá a eliminar la dependencia de la sincronización de la hora con el servidor NTP.

La segunda parte considerada fue el sistema inalámbrico, que envía las mediciones por medio de una red WIFI. Esta parte se basó en la idea que utiliza Flukso de utilizar un router

con el sistema operativo OpenWrt. Esta idea es la clave para reducir el costo del medidor. Ya que el precio de un router es mucho menor que la adquisición de una tarjeta embebida con comunicación WIFI. El router con el sistema operativo OpenWrt es el núcleo del medidor Flukso que utiliza el router WAP-2102. El circuito medidor de potencia está instalado dentro del router según la Figura 2.8. En el diseño del medidor del presente trabajo se utilizó el router DIR-300. Éste fue utilizado en los dos trabajos mencionados anteriormente. Más adelante se detallara como fue que se llegó a la decisión de utilizar este router. Pero, adelantándonos un poco podemos mencionar que el WAP-2102 que utiliza Flukso cuesta en promedio \$60 USD. Además este router no está a la venta en el país.

La tercera parte trata sobre el diseño del servidor web que permite almacenar los datos. Se utilizaron las herramientas que utiliza Flukso las cuales son: PHP [21], PERL [22], RRDTOOL y MYSQL [23].

En los siguientes capítulos se detallará el diseño de las tres partes del *medidor inalámbrico de consumo de energía eléctrica de bajo costo*. Se culminara con la construcción del medidor y las pruebas realizadas a los medidores construidos. Pero vale la pena resaltar que primero se detallará el diseño del sistema WIFI ya que como se mencionó este es el núcleo del medidor. Puede decirse que es la parte más importante del medidor por tener esta alguna cualidad de una PC. Se puede denominar este sistema el cerebro del medidor por las capacidades que se adquieren cuando se utiliza el sistema operativo OpenWrt.

### **3.1 Diseño del sistema para el envío de las mediciones vía WIFI**

Muchos medidores *Smart Meters* utilizan sistemas inalámbricos para poder enviar las mediciones de energía eléctrica. Este monitoreo en tiempo real permite analizar el estado de la carga que el medidor este censando.

El medidor de la Figura 3.2 utiliza la comunicación WIFI. Este medidor es capaz de obtener los parámetros de red tanto consumo energético como también parámetros de calidad de energía. No se tomó como referencia este medidor para nuestro diseño. Únicamente se resalta la importancia de la comunicación WIFI, la cual es utilizada por la mayoría de medidores de este tipo. La comunicación WIFI abre la posibilidad de utilizar topologías de red (red de datos) y la comunicación de los medidores por esta.

El diseño del medidor que se plantea en este trabajo está pensado para ser utilizado a nivel residencial. Como ya se mencionó este no tendrá la precisión de medición de los parámetros de red que tiene un *Smart Meters*. Pero, contara con las ventajas de la comunicación inalámbrica, en este caso la WIFI.

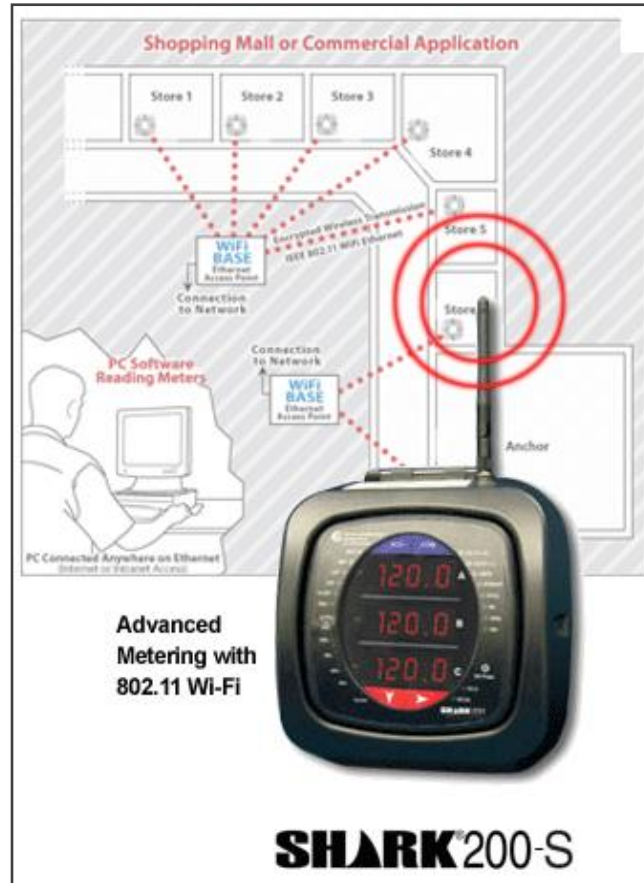


Figura 3.2 : Medidor Shark con comunicación WIFI [23].

Como se mostró el medidor Flukso utiliza para el envío de las mediciones vía WIFI el router WAP-2102. Pero, lo que le da al router la posibilidad de ser utilizado de esta forma es la instalación del sistema operativo OpenWrt.

Al igual que como se hizo en los dos trabajos anteriores [19] [20] aquí se utilizara el router DIR-300. A continuación se muestran las características del hardware del DIR-300 comparadas con las del WAP-2102:

Router	CPU	Ram	Flash	Network	USB	Serial	JTag
WAP2102	Atheros AR2317@182MHz	16MB	4MB	1	No	Yes	No
DIR-300	Atheros AR2317@182MHz	16MB	4MB	4x1	No	Yes	Yes

Tabla 3.1 : Comparación del Router DIR-300 y el WAP-2102.

Como se puede apreciar en la Tabla 3.1 las características del hardware de los router son similares. La más importa de ellas es el CPU el cual como se muestra es el mismo para ambos routers.

Como ya se ha decidido utilizar el router DIR-300 y el sistema operativo OpenWrt se instaló este sistema según los pasos mostrados en el Anexo A.

### **3.2 Diseño del circuito medidor de potencia y energía eléctrica**

Como se dijo anteriormente se utiliza como modelo el circuito medidor de potencia del medidor Flukso (Figura 2.9). El diseño propuesto introduce la utilización de un reloj de tiempo real o RTC (Real-Time Clock).

El circuito a diseñar será capaz de:

1. Muestreo de la corriente utilizando un transductor de corriente AC a voltaje DC (pinza de corriente que utiliza Flukso).
2. Con la corriente digitalizada esta se multiplicará por el voltaje ideal en una red monofásica en este caso 120V, para obtener así un estimado de la potencia activa.
3. Comunicarse con el RTC para la sincronización de la hora.
4. Comunicarse con el router DIR-300 por medio del puerto UART de ambos, el DIR-300 tendrá instalado el firmware OpenWrt compilación de Flukso.

Las cuatro funciones que realiza el circuito nos llevan a la utilización de un microcontrolador. En los dos trabajos anteriores se utilizó un microcontrolador de la familia PIC16F87X. El nuevo diseño utiliza el PIC16F876A. Éste microcontrolador digitaliza (por medio del módulo ADC [24]) la corriente y la multiplicará por el voltaje 120 V. con ello se obtiene la potencia estimada. La multiplicación se realiza en el microcontrolador. Otra de las funciones de este microcontrolador será el envío de las mediciones de potencia y energía eléctrica hacia el router por medio del puerto UART. La comunicación con el RTC se realiza por medio del puerto I2C [25] del PIC16F876A. Para poder realizar las funciones requeridas se programará las rutinas necesarias en el PIC16F876A.

El diagrama de bloques mostrado en la Figura 2.7, muestra todas las partes del sistema de medición que utiliza el medidor Flukso. El esquema propuesto en nuestro diseño se muestra continuación en la Figura 3.3:

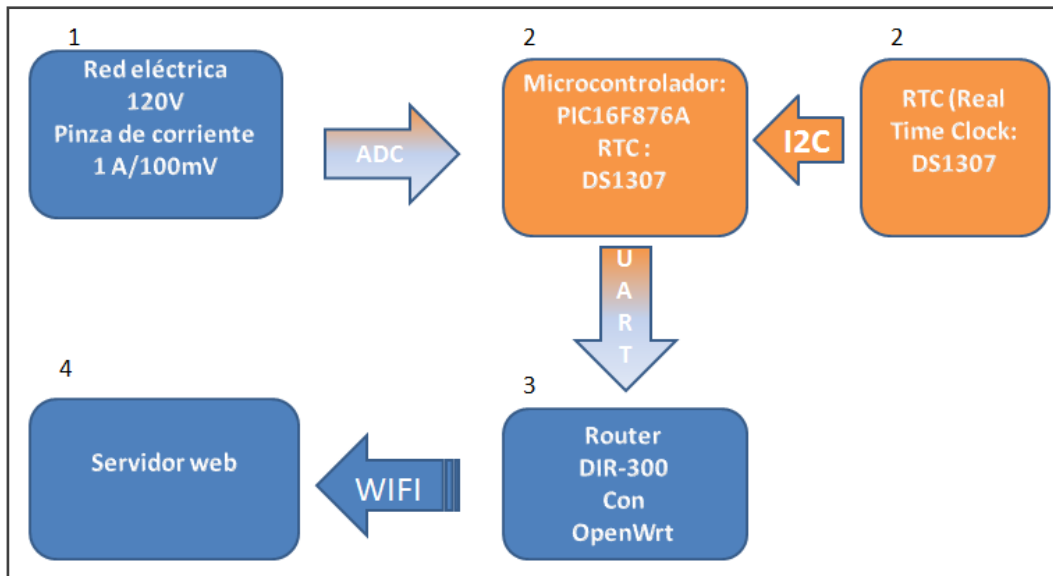


Figura 3.3: Diagrama de Bloques del Sistema: Medidor de energía, red eléctrica y servidor web.

El bloque 1 de la Figura 3.3 representa la red eléctrica y la pinza de corriente. El bloque 2 representa el circuito que mide la potencia y energía eléctrica. El bloque 3 representa el sistema inalámbrico WIFI. A continuación se describen las principales características de los componentes a utilizar.

### 3.2.1 Microcontrolador PIC16F876A

El circuito medidor de potencia debe comunicarse por el puerto serie UART (hacia el router) y el puerto I2C (con el reloj de tiempo real). Además de poder digitalizar el voltaje de la pinza de corriente (voltaje proporcional a la corriente por ser la pinza de corriente un transductor AC a DC). Se seleccionó el microcontrolador PIC16F876A por poseer éste los puertos de comunicación (USART e I2C) y un convertidor analógico digital. Éste pertenece a la familia de gama media de de la serie de microcontrolador PIC16F87X de Microchip [26]. También se puede agregar a esto que el precio de estos microcontroladores resulta bastante accesible para los propósitos y requerimiento del diseño. A continuación en la Tabla 3.2 se resumen las características técnicas de este.

LA comunicación serie asíncrona utiliza la codificación RS-232 para el transporte de los datos. Es muy común en los router la utilización de este puerto serial. En el PIC16F876A, éste se encuentra en los pines 18 y 17 que son RX y TX respectivamente. Como se puede apreciar en el diagrama de distribución de pines del PIC16F876A en la Figura 3.4.

Característica	Descripción
Familia	Embedded - Microcontrollers
Serie	PIC® 16F87XA
Tamaño de la palabra de la CPU	8-Bit
Velocidad	20MHz
Conectividad	I <sup>2</sup> C, SPI, UART/USART
Periféricos	Brown-out Detect/Reset, POR, PWM, WDT
Numero I/O (Entradas y salidas)	22
Tamaño de la memoria de programa	14KB (8K x 14)
Tipo de memoria de programa	FLASH
EEPROM	256 x 8 B
RAM	368 x 8 B
Fuente de voltaje (Vcc/Vdd)	4 V ~ 5.5 V
ADC(Convertidor analógico digital)	A/D 5 x 10 Bit
Temperatura de Operación	-40°C ~ 85°C
Encapsulado	28-SOIC (0.295", 7.50 mm Ancho)

Tabla 3.2 : Descripción del PIC16F876A.

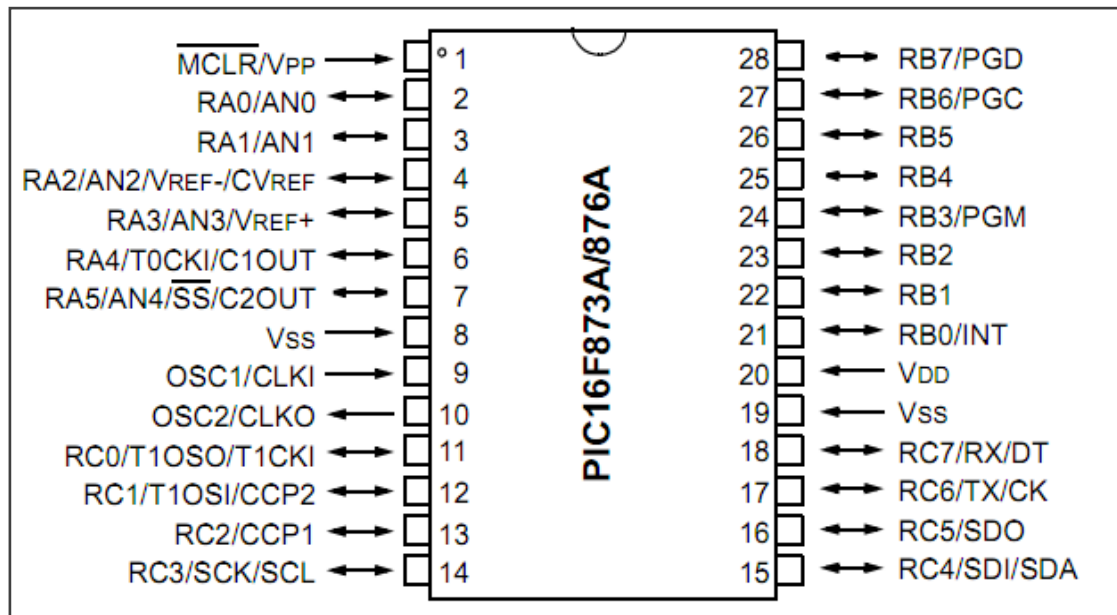


Figura 3.4 : Diagrama de distribución de pines del PIC16F876A

El puerto de comunicación I2C utiliza los pines 15 y 14. Otra de las características de este microcontrolador es que posee un convertidor analógico digital *ADC*, el cual posee 5 canales multiplexados es decir que podemos utilizar 5 pines como entrada analógicas del

ADC. Es de importancia resaltar que el encapsulado de este microcontrolador es tipo SOIC<sup>5</sup> el cual pertenece a los encapsulados *SMT* (La tecnología de montaje superficial, más conocida por sus siglas en inglés *SMT Surface Mount Technology* es el método de construcción de dispositivos electrónicos más utilizado actualmente).

El resto de componentes del circuito medidor de potencia son también tipo *SMT* (tanto el RTC como los componentes pasivos a utilizar resistencias y condensadores) esto se eligió de esta manera para poder construir un circuito lo más pequeño posible para que este pueda ser integrado dentro del router DIR-300. La Figura 3.5 se puede apreciar el encapsulado del PIC16F876A tipo SOIC:



Figura 3.5 : PIC16F876A encapsulado tipo SOIC.

### 3.2.2 Reloj de tiempo real DS1307

El diseño propuesto en este trabajo de graduación incorpora un reloj de tiempo real, que permite guardar la hora exacta. La sincronización temporal se mantiene aún después de desconectar la fuente de alimentación del sistema. También, se evita la dependencia de sincronizar la hora con el *ntpclient*.

El circuito integrado Maxim/Dallas [27] DS1307 [28] es un reloj de tiempo real exacto. Automáticamente mantiene el tiempo y la fecha actual, incluyendo compensación para meses con menos de 31 días y saltos de año. La dirección y los datos son transferidos por el puerto I2C. También el reloj opera en formato de 24 horas o en formato de 12 horas AM/PM.

Los relojes calendario no suelen ser el elemento principal en los dispositivos electrónicos, sino un aditamento que es preferible que opere independientemente, incluso con su propia

---

<sup>5</sup> SOIC: es un acrónimo de Small-Outline Integrated Circuit, y ocupa entre un 30% y 50% menos que un circuito equivalente en cápsula DIP (Dual in-line package por sus siglas en inglés, es una forma de encapsulamiento común en la construcción de circuitos integrados. La forma consiste en un bloque con dos hileras paralelas de pines, la cantidad de éstos depende de cada circuit).

fuentes de alimentación. Por ejemplo, el procesador de una computadora podrá ser muy potente pero aun así el reloj del sistema va en otro chip y usa su propia batería. Justificada la necesidad de los RTCs, veremos que existe una casi incalculable cantidad de RTCs (de diversas marcas, de diversas características). En este trabajo de graduación utiliza el circuito integrado DS1307 de la empresa DALLAS Semiconductors.

### 3.2.2.1 Características del DS1307

El DS1307 es uno de los RTCs I2C más fáciles de usar y sus principales características son:

1. Computa los segundos, minutos, horas, días de la semana, días del mes, meses y años (de 2000 hasta 2099).
2. Aparte de los registros de hora y fecha del RTC, ofrece una SRAM de 56 bytes que se podrían usar como RAM extendida del microcontrolador.
3. Posee el pin SQW/OUT, en donde se puede obtener señal cuadrada de frecuencia programable.
4. Alimentación alterna usando una batería. En ausencia o deficiencia de la alimentación principal de Vcc, el DS1307 pasa automáticamente a alimentarse de la batería.
5. Soporta el protocolo I2C en Standard Mode (máxima frecuencia de reloj de 100 kHz)

### 3.2.2.2 Diagrama de conexión del DS1307

El diagrama de pines del DS1307 se muestra a continuación:

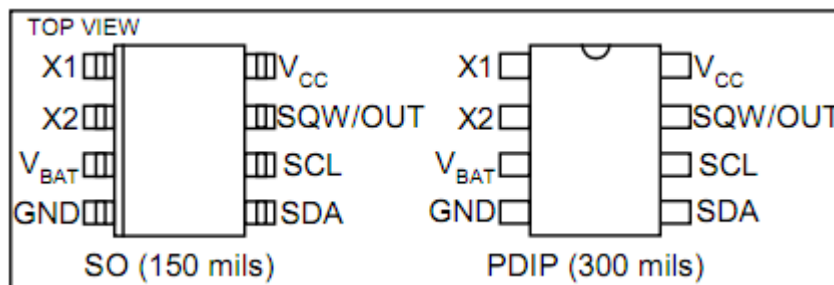


Figura 3.6 : Diagrama de pines del DS1307.

Se muestra en la Figura 3.6 el diagrama de distribución de pines del DS1307. Se utiliza la distribución de pines para un encapsulado tipo SOIC y no la PDIP. Como ya se dijo

anteriormente se busca reducir las dimensiones del circuito. A continuación se resume la descripción de sus pines:

1. SDA y SCL. Pines de interface I2C.
2. Vcc y GND. Pines de alimentación. Vcc es típicamente de 5 V.
3. X1 y X2. Pines para conectar un cristal de cuarzo estándar externo de 32.768 KHz. Los capacitores para estabilizar el circuito oscilador se incluyen internamente.
4. Vbat. Pin para conectar opcionalmente una batería de 2.0 a 3.5 V. Normalmente el DS1307 operará con su fuente del pin Vcc. En ausencia de dicha tensión o cuando su nivel caiga por debajo de Vbat, el DS1307 empezará a trabajar con la batería.
5. SQW/OUT. El DS1307 puede sacar una onda cuadrada de cuatro frecuencias: 1 Hz, 4.096 kHz, 8.192 kHz ó 32.768 kHz. Se configura con el registro de control. Es un pin de drenador abierto y por tanto necesitará de una pull up si se usa.

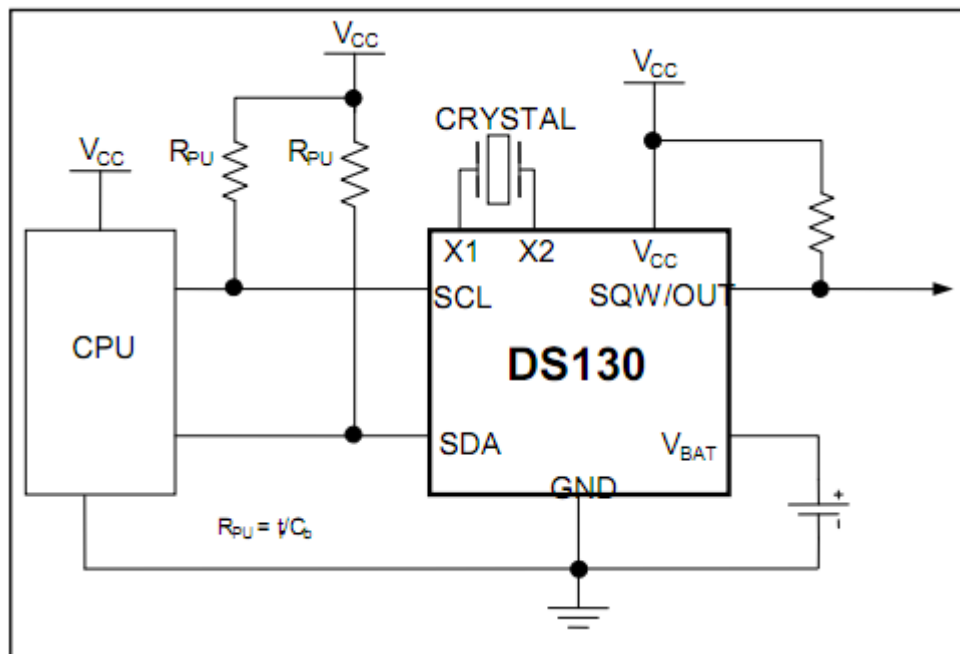


Figura 3.7 : Circuito típico de conexión el DS1307.

La Figura 3.7 muestra la conexión típica del circuito DS1307. Puede observar que se utiliza una batería (VBAT) externa, de tal manera que cuando se desconecta la alimentación Vcc no se pierda la configuración de la hora establecida.

La configuración y programación de la hora y fecha del RTC DS1307 se realiza por medio del puerto I2C. Se configuran los registros de la memoria SRAM del DS1307, los registros

configurar se pueden encontrar en [29]. Como se muestra en la Figura 3.7 el microcontrolador PIC16F876A se comunica por el puerto I2C con el DS1307.

### 3.2.3 Referencia de voltaje REF195

El circuito medidor de potencia necesita alimentación de 5V DC. Éste voltaje de alimentación se obtuvo de la fuente de alimentación del DIR-300. Sin embargo, se determinaron variaciones en el nivel de voltaje de la fuente, que oscila entre 5.2 a 5.6 voltios DC. Se determinó que las oscilaciones eran la causa de errores en las mediciones muestreadas.

Como solución al problema de inestabilidad de voltaje de la fuente de alimentación del DIR-300 se incluyó una referencia de voltaje. Este circuito proporciona un voltaje DC estable. La referencia de voltaje que se utilizó fue la REF-195 [30] de *Analog Devices* [31]. Sus características principales se muestran a continuación:

Característica	Descripción
Familia	PMIC - Voltage Reference
Series	REF195
Tipo de Referencia	Series, Precision
Voltage - Output	5V
Tolerancia	±5mV
Coefficiente de Temperatura	10ppm/°C
Voltage - Input	5.1 V ~ 15 V
Número de canales	1
Corriente - Output	30mA
Temperatura de Operación	-40°C ~ 85°C
Package / Case	8-SOIC (0.154", 3.90mm Width)

Tabla 3.3 : Características del REF-195.

El circuito integrado REF-195 es una referencia de precisión, con una tolerancia de ±5mV. La selección de esta referencia de voltaje tomó en cuenta la máxima corriente que está puede proporcionar (la cual es de 30mA). En los cálculos siguientes se demuestra que los circuitos integrados a utilizar no sobrepasan esta corriente:

$$PIC16F876A: \text{Corriente promedio a } 20MHz = 15m A \quad (\text{Ecuación 3.1.})$$

$$DS1307: \text{Corriente de operacion} = 1.5m A \quad (\text{Ecuación 3.2.})$$

$$\text{Corriente demandada al REF195} = 1.5m A + 15mA + X_{comp} \quad (\text{Ecuación 3.3.})$$

$$X_{comp} = -1.5m - 15mA + 30mA = 13.5 mA \quad (\text{Ecuación 3.4.})$$

Donde  $X_{comp}$  representa la corriente que no se demanda al REF-195. En las ecuaciones 3.1 y 3.2 se muestra la corriente demanda por los IC a utilizar. La suma de estas es de 16.5mA. Esa cantidad no sobrepasa los 30mA que proporciona el REF-195, al sustituir la ecuación 3.3 la corriente que proporciona el REF-195 y despejando esta para  $X_{comp}$  obtenemos la ecuación 3.4 la cual nos muestra la corriente que sobra de la utilización de los IC y la cual se utilizara para la alimentación de otros componentes en el circuito.

La hoja técnica de REF-195 muestra la distribución de pines [30]. La Figura 3.8 muestra la distribución para un encapsulado tipo SOIC.

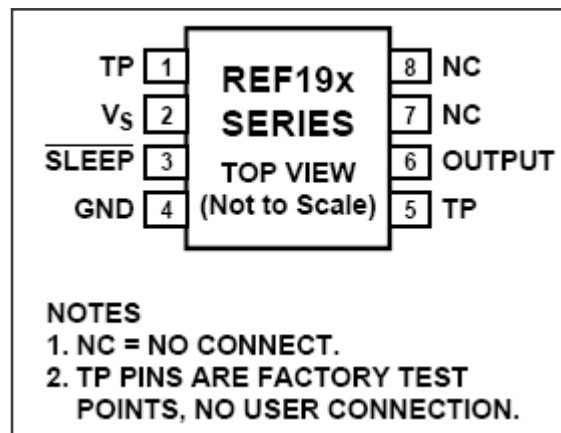


Figura 3.8: Configuración de pines en un encapsulado SOIC-REF195.

En la Figura 3.9 se muestra la conexión típica de la referencia de voltaje REF-195. La conexión típica solo se utiliza 3 y 4 capacitores.

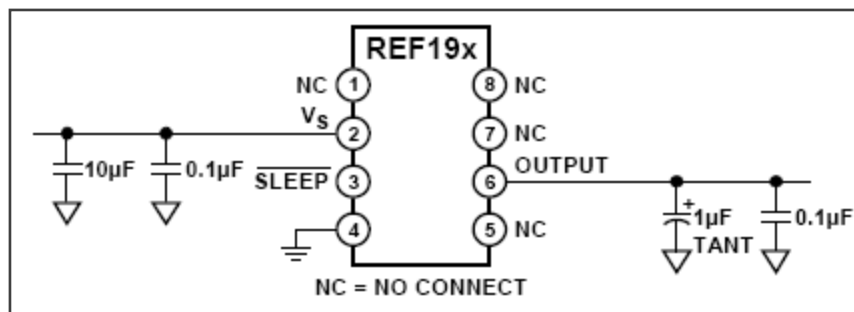


Figura 3.9 : Conexión típica de la referencia de voltaje REF-195.

La introducción de la REF-195 en el nuevo diseño proporcionó la estabilidad requerida. Además, se utilizó como referencia en el convertidor analógico digital del PIC16F876A.

### 3.2.4 Pinza de corriente y el módulo ADC del PIC16F876A

La pinza de corriente o transductor de corriente a voltaje que se utilizó tiene una resolución de (1A AC)/ (100 mV DC). El módulo ADC del PIC16F876A es un convertidor de analógico a digital de 10 bits de resolución. La resolución que tiene cada bit procedente de la conversión tiene un valor que es función de la tensión de referencia  $V_{ref}$  de acuerdo con la ecuación siguiente:

$$Resolucion = \frac{(V_{REF+} - V_{REF-})}{1024} \quad (\text{Ecuación 3.5})$$

Este diseño solo utilizó voltajes de referencia positivos +5 VDC (proporcionado por el REF-195). El voltaje de referencia negativo corresponde a 0 V. De lo anterior se obtiene una resolución de:

$$Resolucion = \frac{5_{VDC}}{1024} \approx 4.88 \text{ mV}_{DC} \quad (\text{Ecuación 3.6})$$

La pinza de corriente (Transductor de corriente AC a voltaje de DC) es de 50 Amperios, con una resolución de 1A/100mV. Por tanto, el máximo voltaje que se obtendrá con esta pinza será de 5VDC (100mV multiplicado por 50) lo cual es el voltaje de operación del micro controlador.

Ahora falta definir como se interpretará el valor de voltaje proporcionado por la pinza de corriente en el módulo ADC del microcontrolador lo cual se muestra a continuación:

Trabajaremos con una resolución de 1A = 20 (ya digitalizado en el ADC del PIC16F876A) entonces para un valor Digitalizado de ADC = 1LSB (bit menos significativo, LSB o Least Significant Bit, en sus siglas en inglés) se obtendrá una medición de corriente de:

$$1\text{LSB} = \frac{1 \text{ A}}{20} = 0.05 \text{ A} \quad (\text{Ecuación 3.7})$$

Por tanto cada paso o resolución del conversor ADC del PIC16F876A representara 0.05 Amperios y trabajando en unidades de amperios cada 20 pasos del conversor ADC representará una unidad de amperios AC, obteniendo una resolución mínima de 0.05 A AC.

Como la salida de la pinza de corriente es 1AC = 100mVDC, y se requiere esto sea interpretado en el ADC como 20 en decimal. Se requiere hacer un ajuste porque con una resolución de 4.88mVDC para que la conversión en el ADC sea igual 20 se necesitaría que a su entrada llegue una tensión de 97.6 mV DC como se detalla a continuación:

$$ADC = 20(decimal) = 20 * 4.88 mV_{DC} = 97.6 mV_{DC} \text{ (Ecuación 3.8)}$$

Por tanto habrá que introducir una ganancia  $G$  entre la salida de la pinza de corriente y el ADC del microcontrolador de:

$$97.6 mV = 100mV G \text{ (Ecuación 3.9)}$$

$$G = \frac{97.6}{100} = 0.976 \text{ (Ecuación 3.10)}$$

Esta ganancia se puede obtener con un par de resistores realizando un divisor de tensión como se muestra en la Figura 3.10:

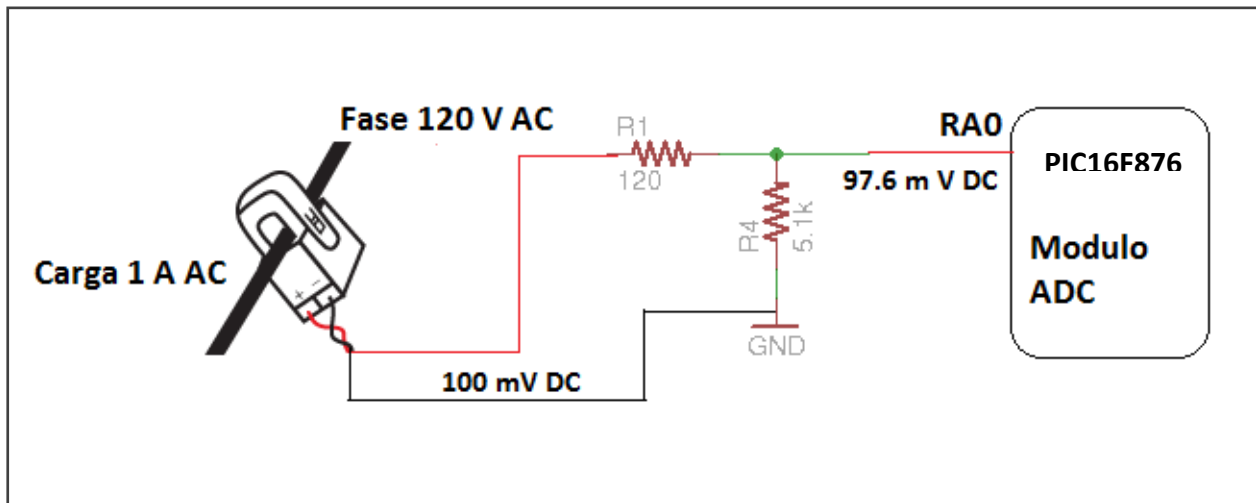


Figura 3.10 : Conexión de la pinza de corriente al módulo ADC.

Con los valores de resistores que se observa en la Figura 3.10 se puede obtener la ganancia de 0.976 y así obtener la resolución requerida. También, se observa la conexión de la pinza de corriente (transductor de corriente AC a voltaje DC) a un nodo con 120 AC con una carga de 1 A AC. La pinza proporciona un voltaje de 100 mV DC. A la entrada del conversor ADC (RA0 entrada 1 el conversor ADC) un voltaje de 97.6 mV DC. Para esta carga el valor digitalizado correspondería a:

$$ADC = \frac{97.6 mV_{DC}}{4.88mV_{DC}} = 20 \text{ (Ecuación 3.11)}$$

Entonces como la carga esta calibrada para obtener idealmente  $120 W = 120 VAC * 1 AC$  para obtener la potencia en el microcontrolador habrá que utilizar un registro con un valor fijo según la ecuación 3.5:

$$REGISTRO\_PIC = \frac{120}{20} = 6 \text{ (Ecuación 3.12)}$$

Y así obtener la potencia activa aproximada:

$$Potencia = ADC * REGISTRO\_PIC = 120 W \text{ (Ecuación 3.13)}$$

La multiplicación de estas variables se realizarán en las rutinas de programación del microcontrolador del PIC16F876A en capítulos posteriores se tratara brevemente las rutinas encargadas de obtener dicha potencia.

### 3.3 Circuito medidor de potencia eléctrica

En Figura 3.11 se muestra el diagrama esquemático del circuito medidor de potencia diseñado. El circuito esquemático muestra la conexión de los IC vistos en la sección 3.2, a continuación se describen las partes del circuito según la Figura 3.11:

#### Conectores:

Según la Figura 5.10 la pinza de corriente se conectan en el conector *PINZAS*. A su vez se conectan con las entradas analógicas *A0* y *A1* del convertidor analógico digital del PIC16F876A. El diseño permite conexión de dos pinzas de corriente aunque se programara el micro controlador para la utilización de una.

*PROGRAMER*: este conector será utilizado para la programación del microcontrolador y este está conectado a los pines: *PGD*, *PGC*, *MCLR*, *VDD* y *GND* los por los cuales se realiza la programación del PIC16F876A

*USART*: este conector esta unido a los pines del puerto serie UART del micro controlador los cuales son: *TX*, *RX* y *GND* este conector se unirá con el puerto UART del router DIR-300 para enviar las mediciones tomadas por el circuito medidor de potencia.

*PULSES\_FOR\_ADE*: este conector esta unido al pin CCP1 [32] y se ha contemplado en el diseño para que en proyectos futuros poder conectar a este un IC como el ADE7757 el cual como se describió anteriormente este envía un pulso digital con una frecuencia proporcional a la potencia activa, para lo cual habría que cambiar el programa del microcontrolador.

#### Regulador de voltaje 5V:

En el circuito de la Figura 3.11, se utiliza la REF-195 como se planteo en la sección 3.2.3. Esta referencia de voltaje nos proporcionara 5V DC para la alimentación de de los IC del circuito medidor de potencia, el voltaje de entrada de esta referencia de voltaje se tomara de la fuente de alimentación del DIR-300. El esquema de conexión utilizado para la REF195 es igual al utilizado en la Figura 3.9 la cual es la conexión típica de este dispositivo.

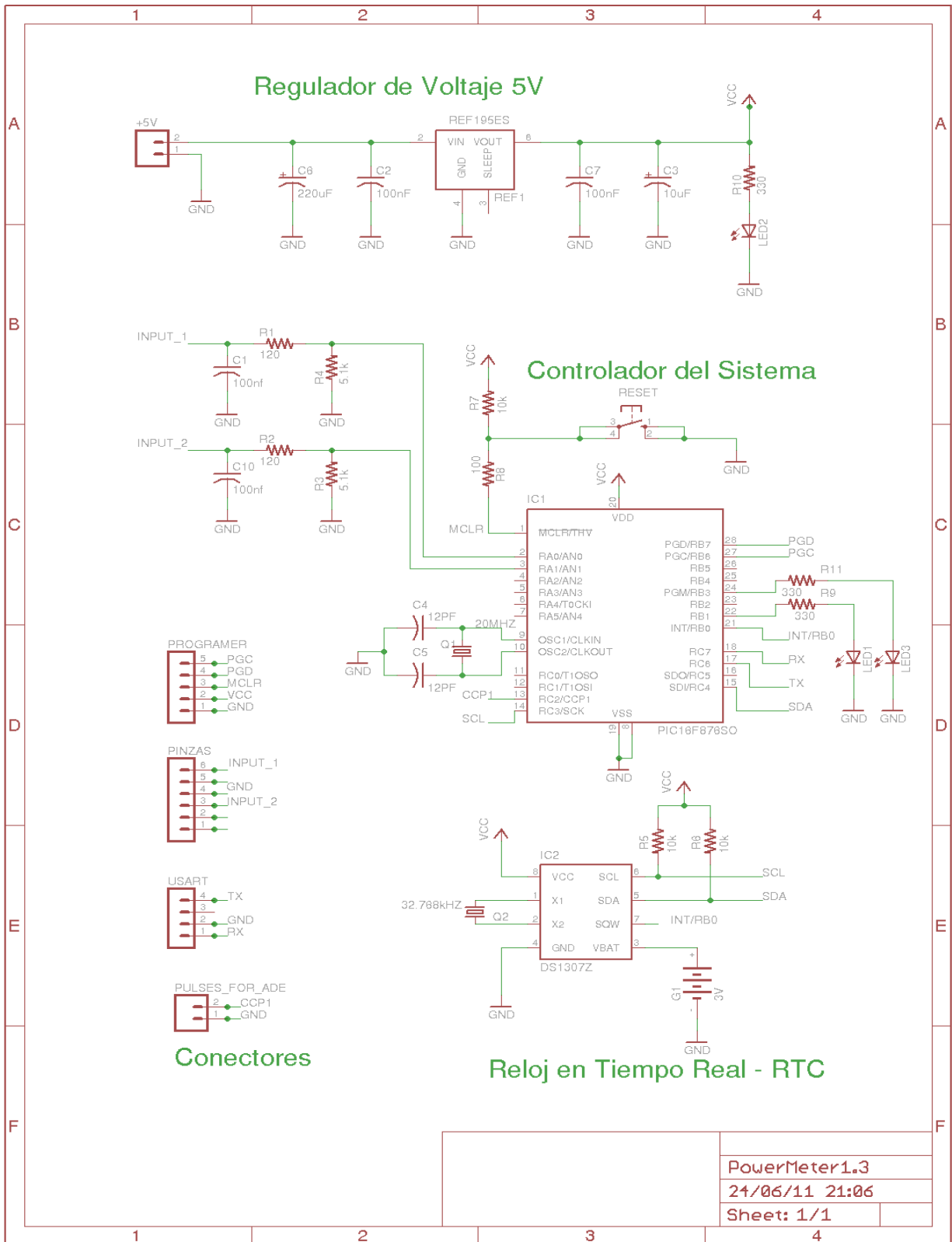


Figura 3.11 : Circuito medidor de potencia versión 1.3

## Reloj de tiempo real RTC:

Como se muestra en la Figura 3.11 el reloj de tiempo real DS1307 se comunicara con el PIC16F876A por medio del puerto de comunicación *I2C*. Los pines del *I2C* en el PIC16F876A son *SDA* y *SCL*. La batería de respaldo del DS1307 se conecta en el pin 3 *Vbat*. Se utiliza una batería de 3V según las especificaciones del DS1307.

## Controlador del sistema:

El controlador del sistema está compuesto básicamente por el PIC16F876A. El controlador se encarga de la comunicación con el RTC para obtener la hora y fecha para sincronizar está en el sistema operativo OpenWrt del DIR-300. También, el PIC16F876A muestrea la corriente procedente de la pinza de corriente. Otra función del controlador del sistema es enviar por el puerto serie UART las mediciones tomadas, la hora y fecha hacia el DIR-300 (cuando sea necesario). Todas las funciones que realiza el PIC16F876A son programadas, configurando los puertos y recursos a utilizar. Por medio de la programación de rutina y algoritmos para realizar las tareas descritas.

### 3.4 Algoritmos y rutinas utilizadas en el PIC16F876A

Para obtener la potencia activa aproximada utilizamos la ecuación 3.13. Por otro lado para obtener la energía eléctrica se utilizara el algoritmo planteado la ecuación 1.16. Esta ecuación nos proporciona el tiempo que una carga se tarda en consumir un 1 Wh. Con la utilización de la ecuación 1.16 y los módulos temporizadores del PIC16F876A se consiguió estimar la energía eléctrica. No se pretende explicar en si todo el código fuente del programa pero este código puede encontrarse en el CD de anexo en */PIC16F876/programas/CSS*.

La Tabla 3.4 indica el formato de las mediciones, a enviarse por el puerto UART del PIC16F876A. El formato se basa en la idea de Flukso, con la diferencia de que se agrega una cadena en donde se trasmitirá la hora y fecha. Como se muestra en la Tabla 3.4 cada cadena a enviar comienza con un comando este comando distingue las cadenas según su tipo potencia, energía y tiempo. La frecuencia con la que se trasmitirán estas cadenas son: *1 s*, *600 s* y *al iniciar el sistema* respectivamente para *pwr*, *pls* y *tim*.

En las cadenas que inician con el comando *pwr* y *pls* se trasmitirá la potencia y la energía eléctrica respectivamente. En la cadena que comienza con el comando *tim*, se enviara la fecha esta posee un formato el cual se detalla a continuación: según la Tabla 3.4 para la cadena *tim* se tiene una medición de *1110011057* lo que corresponde a la fecha: *2011-10-01* y la hora *10:57 am*, la fecha se trasmitirá en el formato *YY:MM:DD* días mes y año.

Comando	ID	Medición	Fin	Tipo
<b>pwr</b>	2d4f76edc4dcf577fa0a49b833513bfe	:0000000120	0A	Potencia W
<b>pls</b>	2d4f76edc4dcf577fa0a49b833513bfe	:0000000300	0A	Energía Wh
<b>tim</b>	2d4f76edc4dcf577fa0a49b833513bfe	:1110011057	0A	Tiempo
<b>Cadena Inicial la cual distingue el mensaje.</b>	Identificador del medidor, este es necesario para que el servidor web reconozca que medidor envía las mediciones.	En esta parte irán las mediciones de potencia o energía obtenidas	El mensaje termina con un avance de línea.	

Tabla 3.4 : Formato de las mediciones a enviar por el puerto UART.

Las cadenas mostradas en la Tabla 3.4 poseen en su totalidad 48 caracteres (según formato de Flukso) y se enviarán por el puerto UART a 4800 baudios para lo cual se configurará el puerto UART del PIC16F876A según [33].

Las rutinas que se grabarán en el PIC16F876A nos permitirán estimar la potencia activa según la Ecuación 3.13 y la energía eléctrica consumida según la ecuación 1.16. Para posteriormente enviar estas por el puerto UART hacia el router DIR-300 en el formato descrito en la Tabla 3.4. La lógica planteada en los diagramas de flujo que se mostraron a continuación toman en cuenta que el circuito a diseñar será capaz de:

1. Muestreo de la corriente utilizando un transductor de corriente AC a voltaje DC (pinza de corriente que utiliza Flukso).
2. Con la corriente digitalizada esta se multiplicará por el voltaje ideal en una red monofásica en este caso 120V, para obtener así un estimado de la potencia activa (Ecuación 3.13).
3. Obtener la energía eléctrica consumida por la carga utilizando la ecuación 1.16.
4. Comunicarse con el RTC para la sincronización de la hora (por medio del puerto I2C).
5. Comunicarse con el router DIR-300 por medio del puerto UART de ambos en el formato descrito en la Tabla 3.4.

En la Figura 3.12 se presenta el diagrama de flujo del programa hacer gravado en el PIC16F876A el cual cumple los requisitos planteados anteriormente así como también este utiliza la ecuación 3.13 y 1.16 para obtener la estimación de la potencia y energía eléctrica respectivamente. EL diagrama de flujo de la Figura 3.12 se muestra la secuencia de configuración de los recursos a utilizar del PIC16F876A (UART, ADC, I2C y TMR1), posterior a esto se tiene un retardo de 20 segundos para esperar que el DIR-300 se estabilice y OpenWrt inicie correctamente. Luego de esto se lee la hora y fecha en el

DS1307 por medio del I2C y se envía esta al DIR-300 por medio del puerto UART en el formato establecido en la Tabla 3.4. Se puede observar la referencia marcada como 1 la cual representa el ciclo infinito en donde el programa estará utilizando la Ecuación 1.16 para obtener el retardo necesario para aumentar el registro de la Energía eléctrica.

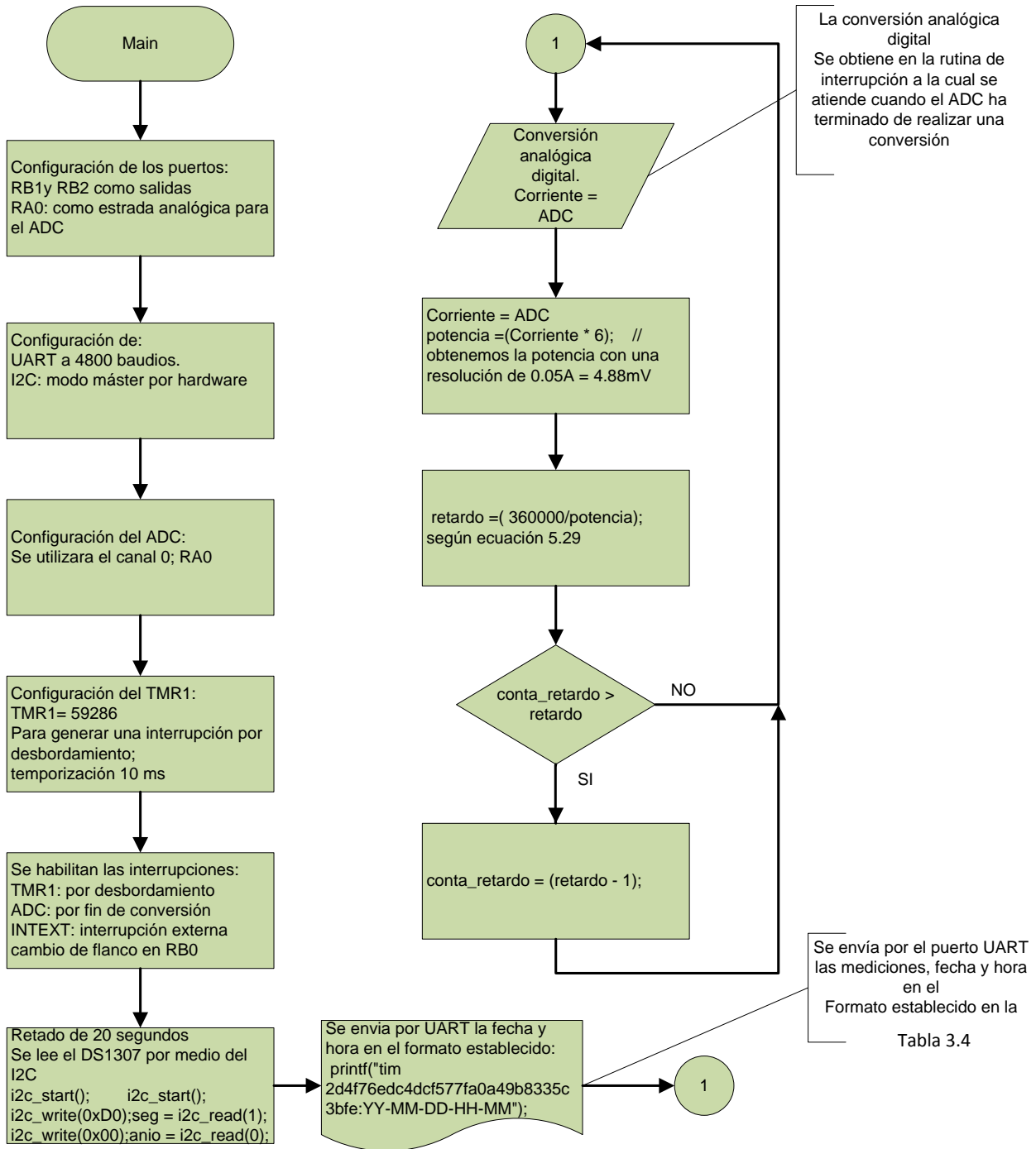


Figura 3.12 : Diagrama de Flujo del programa principal.

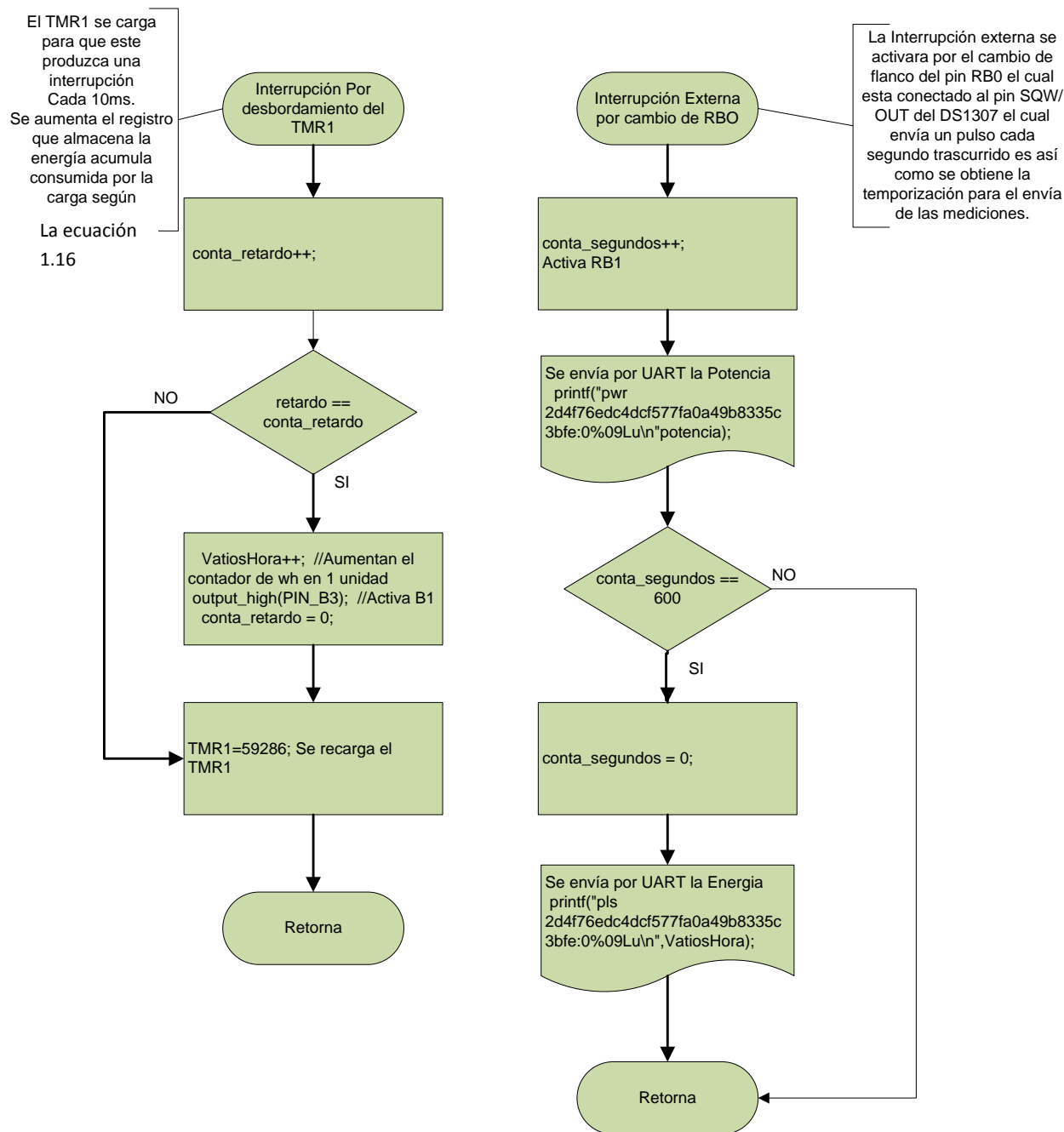


Figura 3.13 : Subrutinas a atender cuando ocurren las interrupciones por TMR1 y EXT\_RBO

Las subrutinas que se muestran en la Figura 3.13 son ejecutadas al desbordarse el TMR1 y cuándo el Pin RBO cambia de flanco este cambio de flanco lo realiza el DS1307 al transcurrir 1 segundo en esta interrupción se envía por el UART la potencia eléctrica cada segundo y la energía eléctrica cada 5 minutos o 600 segundos. El registro utilizado para almacenar la energía eléctrica es aumentado cada vez que la carga conectada ha consumido 1 Wh. Esto se realiza por medio de la Ecuación 1.16 y la rutina de interrupción por desbordamiento del

TMR1 en donde se compara el retardo obtenido con la ecuación 1.16 y la temporización que se lleva con el TMR1 en este caso según el diagrama de flujo de la Figura 3.13 corresponde a la variable *conta\_retardo* la cual aumenta con cada desborde el TMR1, lo cual se da cada 10ms.

Los diagramas de Flujo de la Figura 3.12 y 3.13 serán traducidos a código fuente para ser grabados en el microcontrolador. En este proyecto de graduación hemos utilizado el lenguaje de alto nivel CCS [34] para programar las rutinas descritas. Se ha elegido el lenguaje CCS para programar el microcontrolador por ser este C y la flexibilidad que este otorga a la hora de programar los algoritmos y rutinas requeridas dejando de lado la programación en el Ensamblador del microcontrolador. El CCS utiliza un compilador del mismo nombre el cual traduce el lenguaje C del CCS al ensamblador del PIC16F876A. En el CD de anexo en */PIC16F876A/programas/CSS*, se encuentra el programa en CCS para el circuito medidor de potencia que utiliza el *medidor inalámbrico de consumo de energía eléctrica de bajo costo*.

### 3.5 Construcción del circuito

En la Figura 3.11 se mostró el circuito esquemático del medidor de potencia. El esquema fue diseñado con en el programa Eagle [35]. Con el mismo programa se pueden diseñar las pistas del circuito impresos o PCB<sup>6</sup>. Los componentes utilizados fueron de soldadura superficial SMD (dispositivos de montaje superficial, o por sus siglas en inglés, SMD: Surface Mount Device). La utilización de componentes SMD proporciona las siguientes ventajas: menor consumo de potencia y un tamaño reducido del PCB. En la Figura 3.14 se muestra el diseño del PCB y la construcción del mismo:

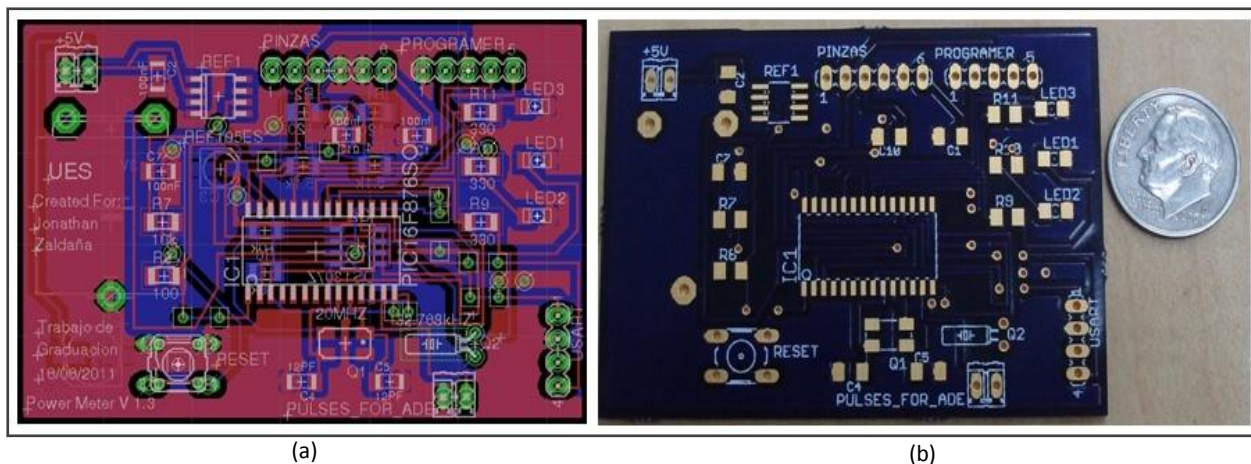


Figura 3.14: (a) Diseño del PCB; (b) Construcción del PCB.

<sup>6</sup> En electrónica, un circuito impreso o PCB (del inglés printed circuit board), es un medio para sostener mecánicamente y conectar eléctricamente componentes electrónicos, a través de rutas o pistas de material conductor, grabados en hojas de cobre laminadas sobre un sustrato no conductor, comúnmente baquelita o fibra de vidrio.

En la Figura 3.14 se muestra el diseño y construcción del circuito impreso del medidor. El diseño en Eagle del PCB se muestra en la Figura 3.14 (a) y la construcción del mismo en la Figura 3.14 (b). Las características de este son:

1. Placa de Fibra de vidrio.
2. Dimensiones del PCB: 6.5 por 4.5 centímetros.
3. Pistas en ambas caras de la placa de fibra de vidrio.
4. Unión de las pistas con remaches.
5. Pistas con protección anti soldadura.
6. Serigrafías de número de los componentes y ubicación de los componentes.

El PCB fue construido por una empresa especializada en estos procesos. A la empresa PCB-MEXICO [36], se le proporcionó en Eagle. Se mandó a construir 5 PCB. El precio de construcción de estos fue de 35\$ USD por unidad. El precio fue bastante caro. La explicación radica en que al construir un PCB por primera vez se utilizan materiales para realizar las maquetas de diseño. Si se mandase a construir otros PCB, el precio se reduciría en un 60% y. Por economía de escala si se mandara a construir 100 unidades el precio se reduciría en un 88%. Los archivos de Eagle del diseño del PCB se pueden encontrar en el CD de anexo en: */Eagle/*.

La soldadura de los componentes SMD en los PCB se realizó utilizando las herramientas de la Figura 8.2. Se utilizó la estación de soldadura que se muestra en la figura por poseer este un sistema para la regulación de la temperatura del caudín.

Al realizar la soldadura de los componentes es necesario asegurarse que la temperatura del caudín este por debajo de los 350° centígrados. Para soldar los circuitos integrados se utiliza una temperatura de 230° Centígrados. Como se muestra en la Figura 3.15 (b), la estación de soldadura que se utilizó posee un sistema de regulación de temperatura por medio de un sistema electrónico.

En la Figura 3.16 se muestra el PCB y los componentes a soldar. Los circuitos integrados utilizados tipo SOIC. Las dimensiones de las resistencias los diodos led son de 3.20 mm por 1.60 mm. Las dimensiones de los encapsulados de los componentes utilizados se configuran en el programa Eagle. El detalle de las dimensiones de todos los componentes utilizados en el PCB se pueden encontrar en el CD de anexo en: */Eagle/*.



(a)

(b)

Figura 3.15 : (a) Microscopio; (b) estación de soldadura necesaria para poder realizar la soldadura de los componentes SMD.

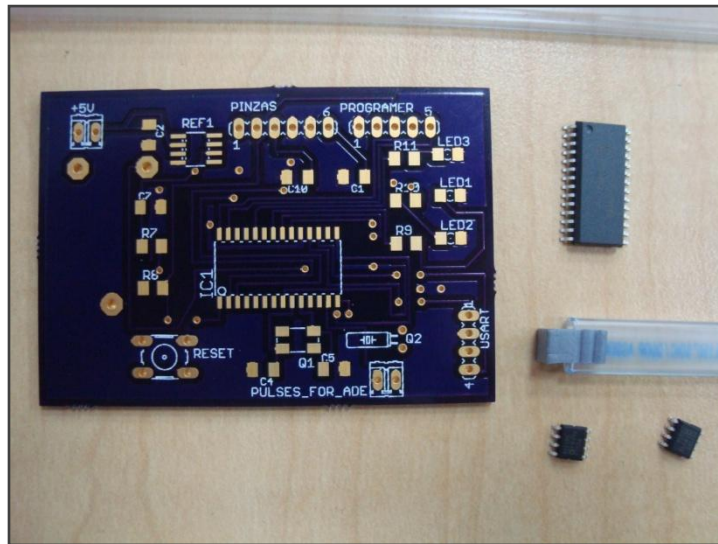


Figura 3.16 : componentes SMD y circuito impreso.

Con la ayuda de las herramientas mencionadas se realizó la soldadura de los componentes en los PCB. En la Figura 3.17 se muestra las dos caras del PCB con los componentes soldados en estas:

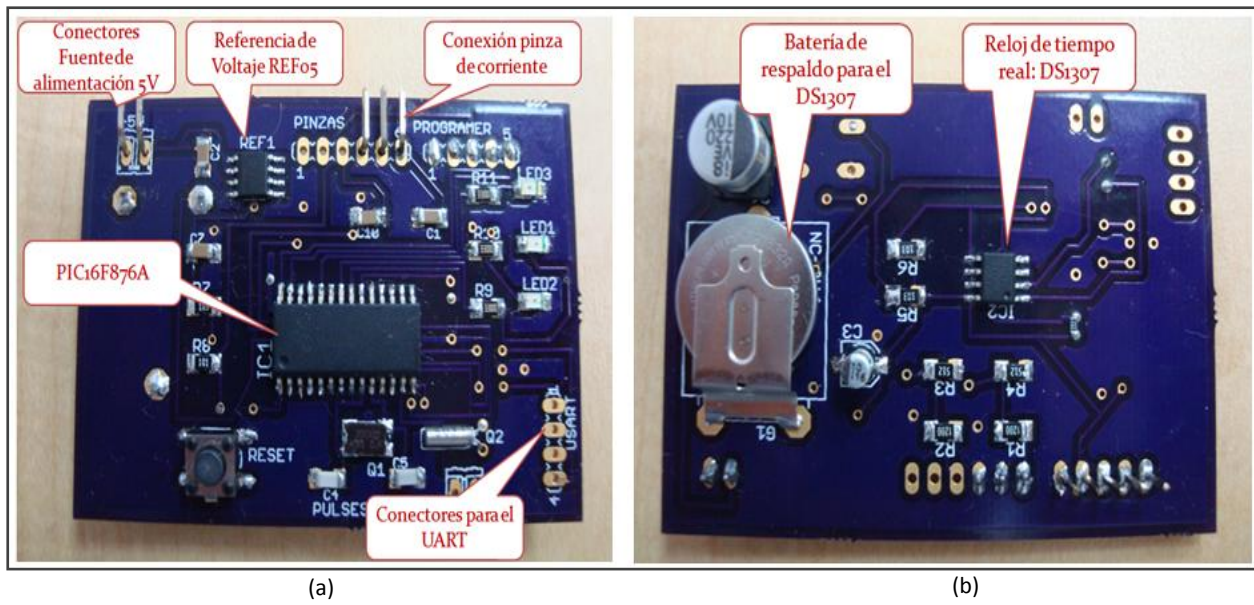


Figura 3.17: (a) Cara del PCB donde se muestra el PIC16F876A; (b) Cara del PCB donde se soldó la batería de respaldo.

Como se puede apreciar en la Figura 3.17 la soldadura realizada es de muy buena calidad. Se pueden apreciar los conectores utilizados para la fuente de alimentación. También se observa la conexión de la pinza de corriente. En la Figura 3.17 (b) se puede apreciar la batería de 3V. Esta sirve de respaldo al DS1307 cuando se desconecta la fuente de alimentación, así este no pierde la hora configurada.

### 3.6 Acoplamiento del circuito medidor de potencia con el DIR-300

Con el circuito construido se procede a acoplar al router.

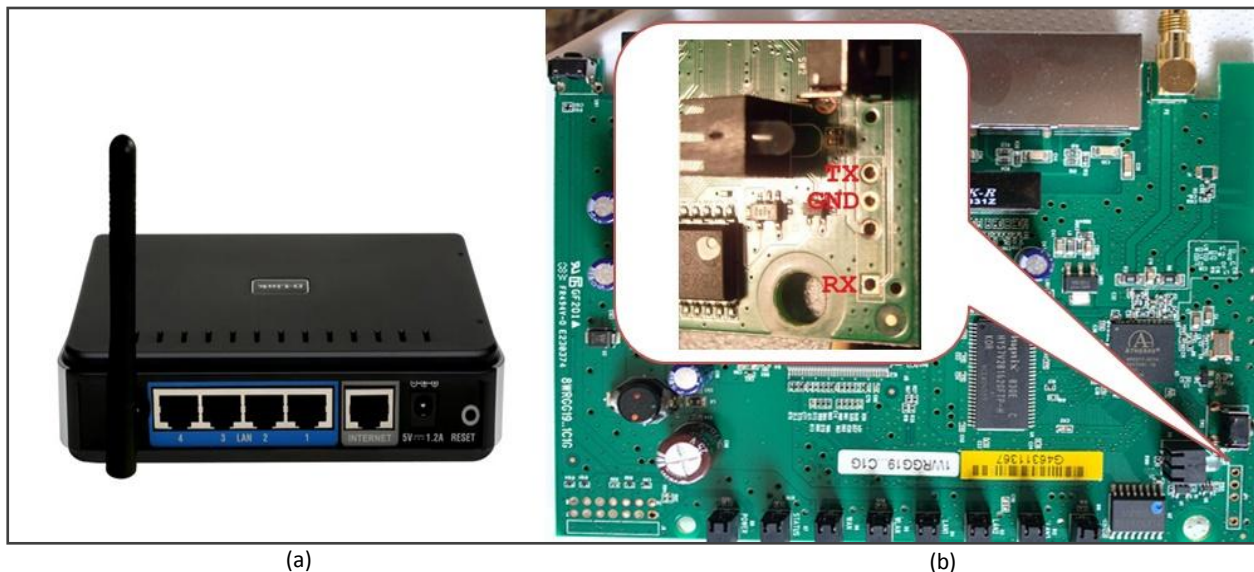


Figura 3.18 : (a) Router DIR-300; (b) ubicación del puerto UART en el DIR-300.

En la Figura 3.18 (b) se muestra la ubicación del puerto UART en el router DIR-300. También se muestra la distribución de los pines de este puerto. En el diseño del PCB del circuito medidor de potencia se tomó en cuenta que este tendría que acoplarse en el puerto UART del DIR-300. Las dimensiones de la ubicación de los pines del puerto UART del PCB se muestran en la Figura 3.17 (a). En la Figura 3.19 se muestra el acoplamiento del circuito medidor de potencia con el DIR-300, el circuito se acopla perfectamente en el puerto UART del DIR-300, no se tiene problemas cuando se acopla la tapa del DIR-300. En la Figura 3.19 (c) se puede apreciar que la conexión de los puertos UART se realizó por medio de pines los cuales unen ambos puertos. La Figura 3.19 (d) resalta la conexión de la pinza de corriente, esta se realizó por medio de conectores tipo DIP. La conexión de la fuente de alimentación del DIR-300 y el circuito, se realizó por medio de conectores DIP.

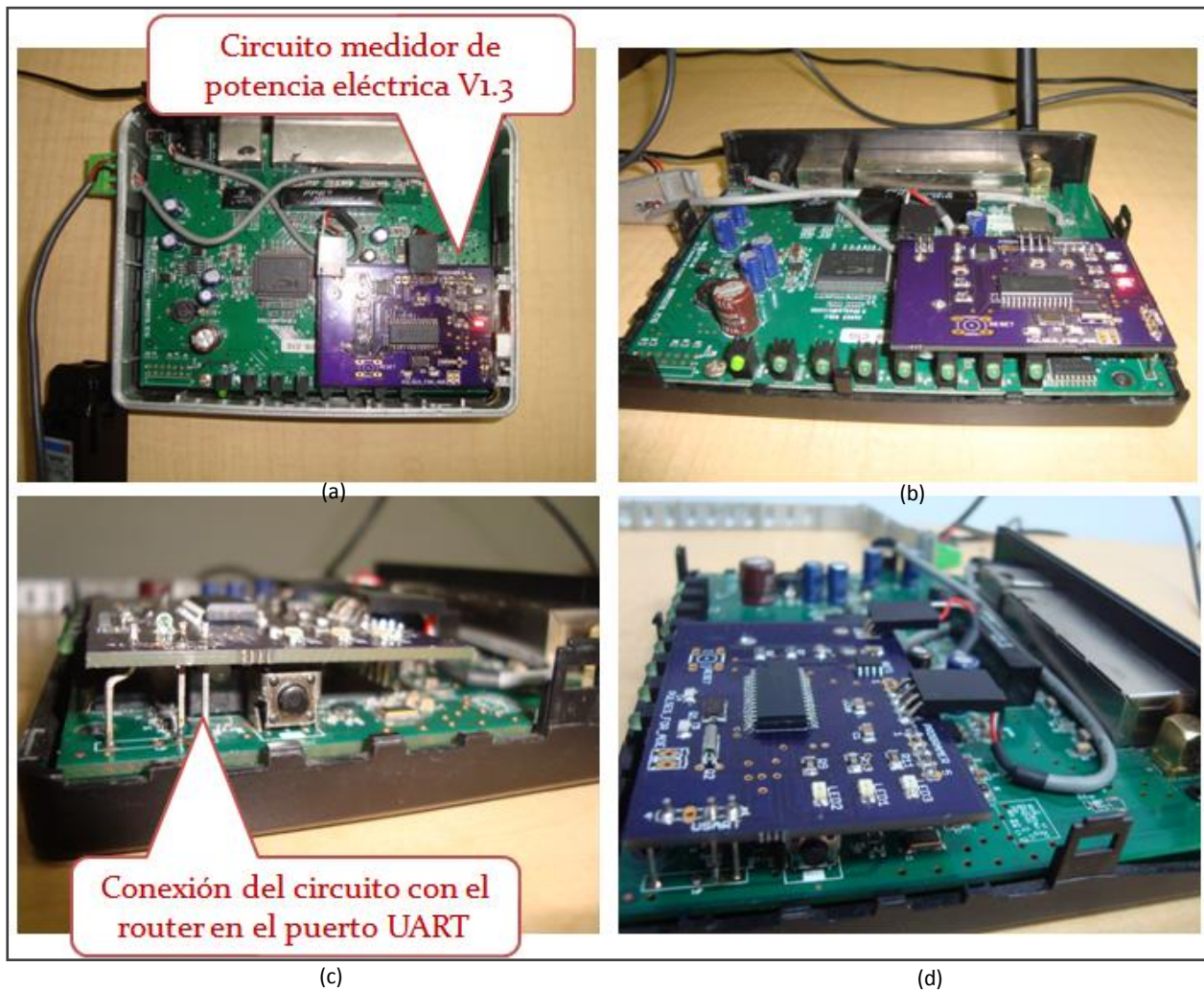


Figura 3.19 : (a) Acoplamiento del circuito con el DIR-300; (b) Conexión de los cables para la pinza; (c) Pines que conectan los puertos UART; (d) Conectores de la pinza y fuente de alimentación.

Como se puede apreciar en la Figura 3.19, el acoplamiento del DIR-300 y el circuito se realizó exitosamente. Si se quisiera retirar el circuito del router se tendría que desconectar los conectores DIP los cuales son de muy fácil manipulación y desoldar los pines en puerto

UART del circuito. Para poder conectar la pinza de corriente se instaló un conector en la esquina superior izquierda del router DIR-300. Como se puede apreciar en la Figura 3.20 (a) este conector hará que la instalación de la pinza con el medidor sea sencilla.

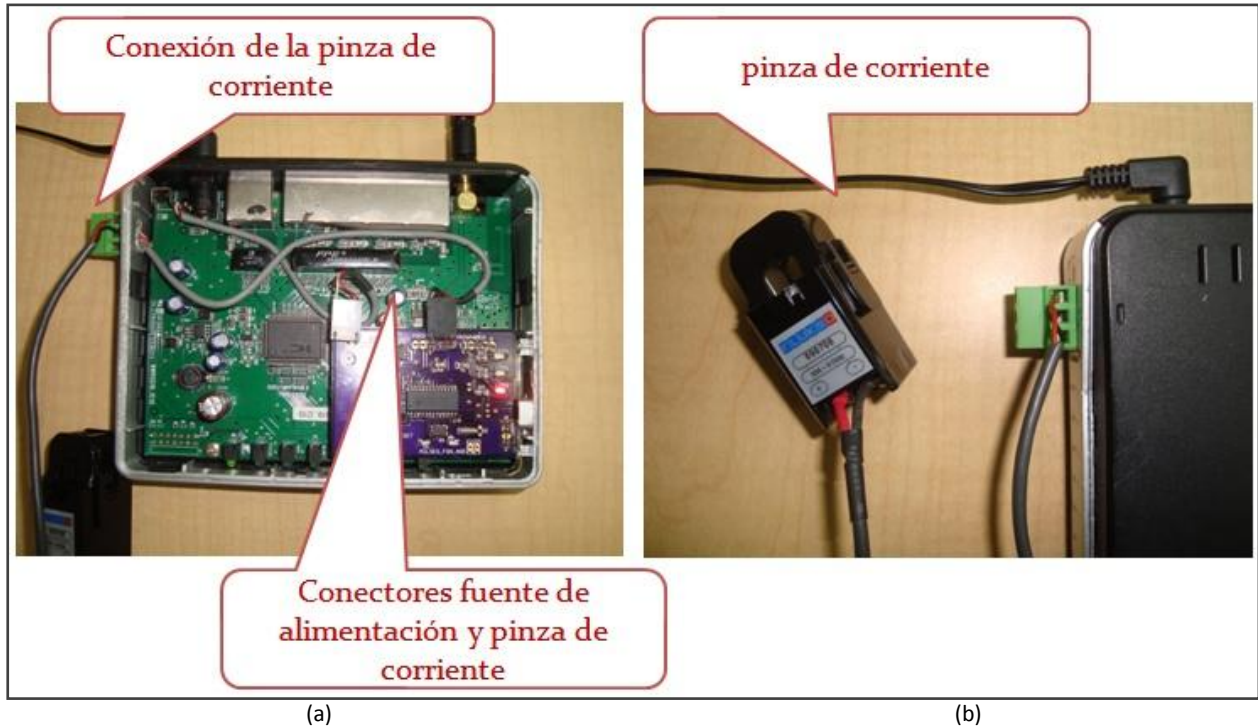


Figura 3.20 : (a) conexión de la pinza de corriente; (b) Conector para la pinza de corriente.



Figura 3.21 : Medidor Inalámbrico de Consumo de Energía Eléctrica de Bajo Costo.

La Figura 3.21 muestra el circuito y la pinza completamente integrados en el router DIR-300.

## 4. Modificación del script *flukso.lua*

El circuito medidor de potencia envía las mediciones hacia el router DIR-300. Estas mediciones serán enviadas a través del puerto serie UART del PIC16F876A hacia el puerto UART del DIR-300. La Tabla 3.4 muestra el formato que llevarán las mediciones. Las cadenas de caracteres son interpretadas en el sistema operativo del router. Flukso utiliza un demonio llamado *flukso.lua*, su función es leer el puerto serie UART del router. También el programa *flukso.lua* se encarga de enviar las mediciones hacia el servidor web.

Se modificó el programa *flukso.lua* agregando líneas de código para que el router acepte las mediciones en el nuevo formato y se puedan enviar las mediciones hacia un servidor web.

### 4.1 Lectura de las mediciones por el puerto UART

Para poder comprender mejor las modificaciones realizadas al programa *flukso.lua*, a continuación en la Figura 4.1 y 4.2 se muestra un segmento del código. Se resalta en color azul las modificaciones realizadas, en negro los comentarios y el código original. También es de importancia resaltar que el código que se muestra corresponde de un segmento del programa *flukso.lua* el cual se muestra en el CD de anexo en */Flukso/programas/*

En las líneas de código 1-3 se utiliza el procedimiento *io.lines(/dev/ttyS0)* encargado de leer el puerto UART. Este acepta como parámetro de entrada la dirección del fichero especial de dispositivo del puerto serie */dev/ttyS0*. Tal como se muestra en el segmento de código, las mediciones o la fecha y hora enviadas por el circuito medidor de potencia son almacenadas en la variable *line* (línea 1). En la línea 2 se efectúa la operación de dividir la cadena enviada y guarda en las variables: *command* el comando (el cual puede ser *pwr*, *pls* o *tim*), en *meter* el ID del medidor y en *value* en valor de la medición potencia, energía o tiempo (fecha y hora).

La instrucción *for* de la línea 1 realiza un ciclo *for* ínfimo. En el cual el demonio *flukso.lua* leerá el puerto serie y enviará las mediciones hacia servidor web. El script se divide en 3 bloques (líneas 9 y 57 de la Figura 4.1 y línea 98 de la Figura 4.2) estos bloques son intrusiones *if* en donde se consulta el comando enviado por el circuito. El primer *if* del bloque se encuentra en la línea 9 aquí se consulta si el comando enviado es *pls*. Consulta si la cadena enviada contiene la energía de ser así las líneas siguientes a este acumulan en una base de datos la energía eléctrica medida. Posterior a esto se envía la medición de energía hacia el servidor web.

```

1.for line in io.lines(/dev/ttyS0) do --líneas de flukso, se lee el puerto UART
2.  local command, meter, value, msec = line:match(pattern) --command:
3.  value = tonumber(value or '0') --meter se almacena el ID
4.  msec = tonumber(msec or '0') --value: se almacena la medición o el tiempo si
5.  -- el comando es tim
6.  local length = line:len() --longitud de la cadena en este caso es 48
7.  -- caracteres
8.
9. if command == 'pls' and (length == 47 or length == 58) then -- entra en el if si
10.  -- el comando es pls
11.  flash()
12.--#####Si el medidor de potencia envía por UART el comando pls(Energía)###
13.--#####como comando se guarda la medición #####
14.
15.  cursoruci = uci.cursor() --cursor Uci para direccionar los archivos uci
16.
17.
18.  Energy1 = cursoruci:get("watth","Energy","data60") -- copiamos de la base
19.  --de datos la energía acumulada
20.  if logica == 1 then
21.  value2 = value - value1
22.  Energy = value2 + Energy1
23.  end
24.  if logica == 0 then
25.  logica = 1
26.  Energy = Energy1
27.  end
28.  value1= value
29.
30.  cursoruci:set("watth", "Energy", "data60", Energy) --se guarda la medición en
31.  -- la base en este caso Energía
32.  cursoruci:commit("watth") --Se almacena la medición en el
33.  -- formato [tiempo1,medicion1 ..tiempo10,medicion10]
34.  hora = os.date("%H:%M",os.time() - 21600) --obtenemos la hora del sistema
35.  -- en el formato H:M:D se resta 21600
36.  cursoruci:set("watth", "date", "data60", hora) --para pasar de UTC a la hora
37.  --local; se guarda la fecha de la medición
39.  cursoruci:commit("watth")
40.
41.  Device ="27f8cbad4babe36ecc67e8179f6fc287" --ID del medidor de potencia
42.  --se envían las mediciones almacenadas en Tabla[],por el protocolo XMLRPC
43.
44.  local ok, res = xmlrpc.http.call ("http://192.168.1.180/cgi-bin/AddEnergy.cgi
45.  ",sample.AddEnergy,Device,os.time(),Energy)
46.
47.--#####
48.  posix.syslog(30, 'received pulse from ' .. port .. ': ' .. line:sub(5))
49.
50.  if homeEnable == 1 then coroutine.resume(e_child, meter, os.time(), value) end
51.
52.  -- pls includes a msec timestamp so report to p_child as well
53.  if length == 58 and localEnable == 1 then
54.  coroutine.resume(p_child, meter, os.time(), value, msec)
54.  end
56.
57. elseif command == 'pwr' and length == 47 then -- si el comando enviado es
58.  --pwr(potencia)

```

Figura 4.1 : segmento de código del script flukso.lua.

```

59.--##### Si el medidor de potencia envía por UART el comando pwr como #####
60.--##### comando se guarda la medición #####
61.
62.   potencia = potencia + value           -- Acumulamos la potencia medida
63.   i = i + 1
64.   if i == 60 then                       -- si ay 60 mediciones(1 medición cada 1s)
65.       -- guardamos el dato
66.   potencia = potencia / i               --Promedio de las 60 mediciones
67.   Tabla[IndiceTabla]= os.time()         --Se guarda el tiempo en que fue tomada
68.       --la medición
69.   Tabla[IndiceTabla+1]= potencia        --Se almacena la medición en el
70.       --formato [tiempo1,medicion1 ..tiempo10,medicion10]
71.   IndiceTabla = IndiceTabla + 2        --aumenta el índice de la tabla para
72.       --formar pares [tiempo, medición..]
73.   i = 0
74.--##### En esta parte del programa se envían las mediciones#####
75.--##### a el servidor web #####
76.
77.   if IndiceTabla == 11 then             -- si el índice de tabla es 11 se han
78.       --almacenado 5 mediciones(5 minutos)
79.   IndiceTabla = 1
80.
81.   Device ="27f8cbad4babe36ecc67e8179f6fc287" --ID del medidor de potencia
82.       --se envían las mediciones
83.       -- almacenadas en Tabla[],por el protocolo XMLRPC
84.
85.   local ok, res = xmlrpc.http.call ("http://192.168.1.180/cgi-bin/AddWatts.cgi
86.       ",sample.AddWatts,Device,Tabla[1],Tabla[2],Tabla[3],
87.       Tabla[4],Tabla[5],Tabla[6],Tabla[7],Tabla[8],Tabla[9],Tabla[10])
88.
89.-- En el servidor se ejecutara el proceso AddWatts el cual guardara la medición
90.   end
91.   else
92.   end
93.--##### Si el medidor de potencia envía por UART el comando#####
94.--##### tim se saetea la hora y fecha en el sistema #####
95. if localEnable == 1 then coroutine.resume(p_child, meter, os.time(), value) end
96. -- limas de flukso
97.
98. elseif command == 'tim' then           -- si el comando es tim entra al if
99.   posix.syslog(31, 'received message from ' .. port .. ': ' .. line:sub(5))
100.   os.execute("date -u 20"..value)       -- se setea la hora y fecha con el
101.       -- comando date -u "YY-MM-DD HH:MM"
102.--#####
103.   else                                   -- error
104.   posix.syslog(27, 'input error on ' .. port .. ': ' .. line)
105.   end

```

Figura 4.2 : Continuación del segmento de código del script flukso.lua.

El siguiente bloque *if* se encuentra en la línea 57, aquí se consulta si el comando enviado es *pwr*. De ser así se acumula la potencia según la línea 62 (*potencia = potencia + value*). Luego se obtienen 5 mediciones (1 por minuto) las cuales se guardan en una variable con su respectiva marca de tiempo. El código que realiza esta acción se encuentra en las líneas 66-71.

Las mediciones de potencia eléctrica y el tiempo en que fueron tomadas estas se almacenan en una variable de tipo tabla llamada *Tabla[IndiceTabla]* en el orden siguiente:

*Tabla[tiempo1,medicion1,tiempo2,medicion2 ... tiempo5,medicion5]*; en este formato se almacenaran las 5 mediciones.

Para poder obtener el tiempo en que son tomadas las mediciones se utiliza la función *os.time()* de la línea 67 y se obtienen específicamente con las instrucción: *Tabla[IndiceTabla]= os.time()*. En donde el procediendo *os.time()* obtiene el tiempo(hora y fecha) del sistema operativo en formato *Tiempo-Unix*<sup>7</sup>. Se utiliza el formato *tiempo Unix* por porque en el servidor web el programa RRDTOLL requiere este formato.

El último bloque *if* se encuentra en línea 98, aquí se consulta si el comando enviado es *tim* de ser así la variable *value* contiene la hora y fecha obtenidas del DS1307. Con la hora y fecha en la variable *value* se procede a realizar la sincronización del tiempo en el sistema operativo. El código de la línea 100 (*os.execute("date -u 20"..value)* ) utiliza el procedimiento *os.execute()* el cual ejecuta los comandos del Shell, en este caso ejecuta el comendo *date* para fijar la hora y fecha al DIR-300.

## 4.2 Envío de las mediciones hacia servidor web por medio de XML-RPC

En la Figura 4.1 y 4.2 se mostro las modificaciones del programa *flukso.lua*. Estas contemplan el envío de las mediciones a el servidor web en el formato XML-RPC [37]. A continuación se detallara como se realiza esto.

XML-RPC Se trata de una especificación que permite enviar datos por medio de HTTP en un formato XML. También se pueden realizar llamadas de procedimiento remotos a través de Internet. Está llamadas a procedimientos remotos se realiza a través de HTTP como el transporte y XML como la codificación. XML-RPC está diseñado para ser tan simple como sea posible, permitiendo al mismo tiempo las estructuras de datos complejas para ser transmitida, procesada y devuelta. En la Figura 4.3 se muestra la lógica que utiliza XML-RPC para trasportar los datos.

La Figura 4.3 muestra el flujo de los datos cuando se utiliza XML-RPC. Los datos a enviar se codifican en XML según la especificación [38]. Posteriormente se trasportan por medio de HTTP. Además de los datos se envía el nombre del procedimiento remoto a ejecutar. En nuestro caso se envía el nombre del procedimiento que está alojado en el servidor web, el cual guardara las mediciones de potencia y energía eléctrica obtenidas.

---

<sup>7</sup> Tiempo Unix: es un sistema para la descripción de instantes de tiempo: se define como la cantidad de segundos transcurridos desde la medianoche UTC del 1 de enero de 1970.

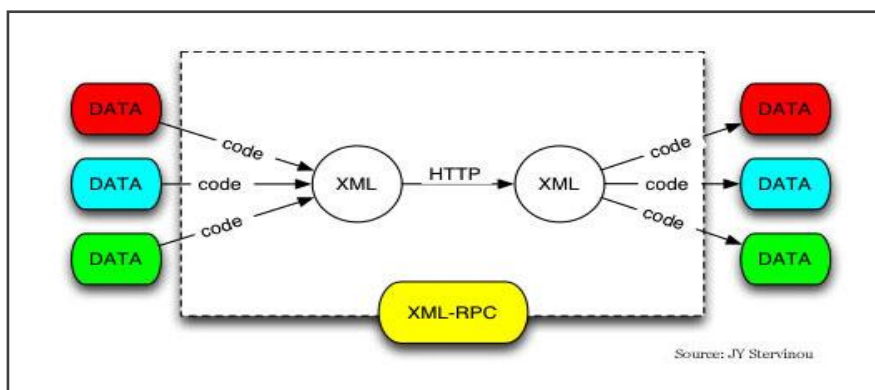


Figura 4.3 : Esquema de comunicación con XML-RPC.

Una de las ventajas de utilizar XML-RPC es que la comunicación se puede realizar en sistemas operativos diferentes y en entornos diferentes como lo son PERL, LUA, PHP, JAVA etc. Los lenguajes de programación mencionados poseen librerías XML-RPC para hacer más flexible la programación de aplicaciones que utilicen llamadas a procedimientos remotos. El entorno o lenguaje de programación utilizado por el cliente y el servidor pueden ser diferentes. Por ejemplo se puede realizar la comunicación entre un cliente el cual utilice PERL y un servidor que utilice PHP.

Para programar las aplicaciones de comunicación entre el circuito que mide la potencia y el router se utiliza el lenguaje LUA. En concreto se modifica el demonio *flukso.lua*, para que el router DIR-300 envíe las mediciones en el formato XML-RPC. Los medidores serán clientes los cuales enviaran las mediciones y ejecutarán el procedimiento en el servidor web el cual guardara las mediciones en una base de datos.

### 4.3 LUA Socket

Lua Socket [39] es una librería de LUA. Se compone de dos partes: un núcleo C que proporciona soporte para los protocolos TCP y UDP y un conjunto de módulos de LUA que añaden soporte para las aplicaciones que tienen que ver con la Internet.

En nuestro caso utilizamos la librería *lua-socket*. Ésta viene en el paquete *luasocket\_2.0.2-3\_atheros.ipk*. Y está instalado en la compilación de OpenWrt de Flukso, que se está utilizando. También se utiliza y la librería *xmlrpc.http* que viene en el paquete: *liblua-xmlrpc\_1.0x-1\_atheros.ipk*. Al igual que *lua-socket*, viene instalada en la compilación de OpenWrt que estamos utilizando. También, se pueden encontrar estas librerías en los repositorios de OpenWrt [40].

## 4.4 Modificación del script *flukso.lua* para el envío de las mediciones a el servidor Web.

Con las librerías *Lua-Socket* y *xmlrpc.http* instaladas en el DIR-300 se realizan las modificaciones necesarias al script *flukso.lua*. El segmento de código del script *flukso.lua*, que representó en la sección 4.1 contiene las modificaciones realizadas para el envío de las mediciones por medio de XML-RPC en LUA.

En las líneas 44-45 del segmento de código *flukso.lua* se encuentra el uso de la función *xmlrpc.http.call* en modo cliente, el detalle del uso de esta función se encuentra en [41]. A continuación en la Figura 4.4 se muestra el segmento de código:

```
local ok, res =  
xmlrpc.http.call ("http://192.168.1.180/cgi-bin/AddEnergy.cgi", sample.AddEnergy  
, Device, os.time(), Energy)
```

Figura 4.4 : segmento de código para el envío de las mediciones de energía hacia el servidor web en formato XML-RPC.

Esta función ejecuta un procedimiento remoto alojado en la dirección: *http://192.168.1.180/cgi-bin/AddEnergy.cgi* (dirección del servidor web directorio */cgi-bin/*). La dirección del servidor web donde se envían las mediciones y donde se ejecuta el procedimiento *AddEnergy.cgi*. Éste ultimo guarda las mediciones de energía eléctrica en una base de datos. Los datos que se envían por XML-RPC son el ID del dispositivo (línea 41), el tiempo obtenido con la función *os.time()* y la energía acumulada. Esta última se encuentra en la variable *Energy* (líneas 18-28 del segmento de código *flukso.lua*). Al ejecutarse esta función se realiza la llamada al procedimiento remoto *AddEnergy.cgi* el cual guarda las mediciones enviados por el medidor con el ID correspondiente.

El envío de las mediciones de potencia eléctrica se realiza de manera similar a la energía, el segmento de código que realiza esta acción se encuentra en las líneas 85-87:

```
local ok, res = xmlrpc.http.call ("http://192.168.1.180/cgi-  
bin/AddWatts.cgi", sample.AddWatts, Device, Tabla[1], Tabla[2], Tabla[3], Tabla[4],  
Tabla[5], Tabla[6], Tabla[7], Tabla[8], Tabla[9], Tabla[10])
```

Figura 4.5 : segmento de código para el envío de las mediciones de potencia hacia el servidor web en formato XML-RPC.

En este caso la función *xmlrpc.http.call* tiene como parámetro la dirección del servidor web, que es la misma que para la energía. El procediendo *AddWatts.cgi* guarda la potencia eléctrica en la base de datos del servidor web. La otra diferencia en el uso de la función *xmlrpc.http.call* son los datos a enviados. En este caso corresponden a las mediciones de

potencia tomadas durante 5 minutos. Estas mediciones son almacenadas en la variable *Tabla* en el formato:

```
Tabla[tiempo1,medicion1,tiempo2,medicion2 ... tiempo5,medicion5];  
Tabla: es una variable tipo Table similar a un arreglo en el lenguaje C
```

Figura 4.6 : Lógica de la variable *Tabla* donde se guardan las mediciones de potencia.

En la Figura 4.4 cada campo par (2, 4, 6, 8 y 10) de este arreglo corresponde a el promedio de las mediciones de potencia tomadas durante 1 minuto. El PIC16F876A esta programado para enviar las mediciones obtenidas en intervalos de un segundo, de estas se saca un promedio de 60 mediciones (líneas 62-66 de la Figura 4.2). Así obtener una medición por minuto. Como se muestra en la Figura 4.4 se guarda en la variable *Tabla* un total 5 mediciones de potencia. También Figura 4.4 Los campos impares del arreglo *Tabla* (3, 5, 7 y 9) corresponden al tiempo en que fueron tomadas las mediciones con la función *os.time()* (línea 67 de la Figura 4.2). De esta manera se envían las 5 mediciones de potencia hacia el servidor web en el formato XML-RPC.

En este capítulo se detalló la programación realizada para que el DIR-300 pueda leer las mediciones enviadas por el circuito medidor de potencia. También se mostro la programación para que el Dir-300 envíe de las mediciones en el formato XML-RPC. Resta ahora detallar la arquitectura del servidor web quien recibe estas mediciones y las guarda en una base de datos.

## 5. Diseño e implementación del servidor web

En los dos trabajos anteriores [19] [20], se utilizó el servidor web del medidor Flukso para poder visualizar las mediciones obtenidas. Se diseño e implemento un servidor web con herramientas GNU (siguiendo parte de la arquitectura que utiliza Flukso en su servidor web).

El servidor tiene las funciones siguientes.

1. Se utiliza el servidor *apache 2* [42] en una plataforma Linux.
2. Contiene las bases de datos de los medidores y usuarios asociados a los medidores, para esto utilizaremos las bases de datos en MYSQL.
3. Contiene las bases de datos en donde se guardaran las mediciones de energía y potencia eléctrica enviadas por los medidores, para realizar estas bases se utiliza la herramienta RRDTOOL [5].
4. Se dibujan las mediciones de potencia y energía eléctrica, para lo cual se utilizó la herramienta RRDTOOL.
5. Contiene las CGI necesarias para realizar las funciones de: ingresar usuario, crear un nuevo usuario, generar y visualizar las graficas de las mediciones, estas CGI serán programadas en PHP y comandos del SHELL de Linux.
6. Contiene los procedimientos remotos, cuyo acceso se realiza por medio de XML-RPC. Éste se utiliza en modo servidor y se implemento utilizando el leguaje de programación PERL.

En la Figura 5.1 se muestra el esquema de la arquitectura utilizada en el servidor web. Como se puede apreciar este tendrá como clientes a los usuarios que visualizaran las mediciones y a los medidores, que envían las mediciones de potencia y energía.

El esquema de la Figura 5.1 muestra a los medidores conectados a un AP, el cual les dará el acceso a internet. Las mediciones se envían al servidor web por medio de XML-RPC. La arquitectura utilizada para el servidor web es la de 3 capas. Éstas son: la capa de presentación, la capa de negocio y la capa de datos. En cada capa del servidor web se utilizaron las herramientas mostradas en los recuadros de la Figura 5.1. La capa de presentación utiliza páginas en HTML. La capa de negocio utiliza CGI en PHP, PERL y CGI programadas con comandos del Shell de Linux. La capa de datos utiliza las herramientas MYSQL y RRDTOOL para almacenar los datos. A continuación describiremos el diseño de capa del servidor web:

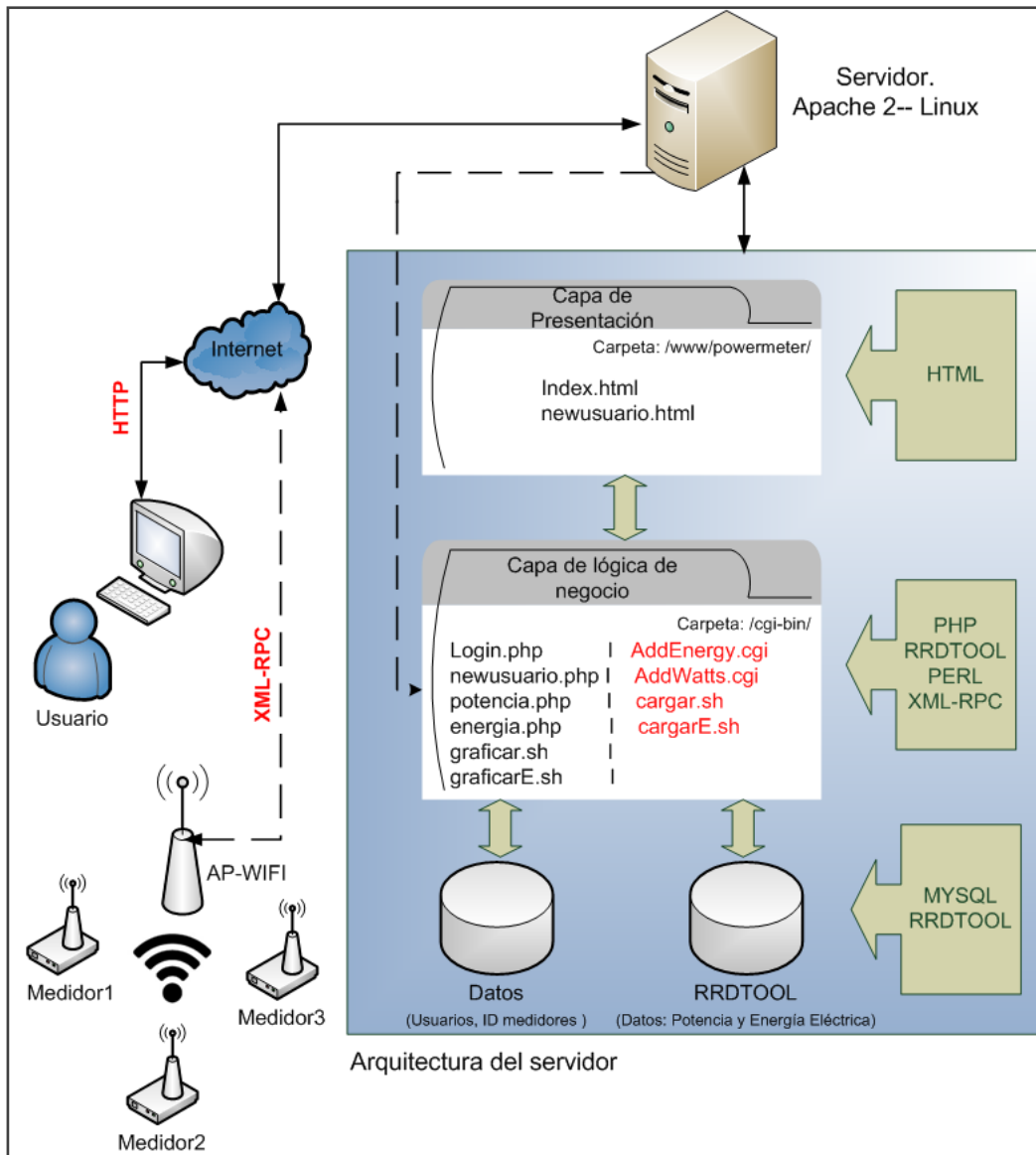


Figura 5.1 : Arquitectura del servidor web.

### 5.1 Arquitectura cliente/servidor en tres 3 capas.

Lo que se conoce como arquitectura en capas es en realidad un estilo de programación donde el objetivo principal es separar los diferentes aspectos del desarrollo, tales como las cuestiones de presentación, lógica de negocio, mecanismos de almacenamiento, etc. Este modelo nos permite tener un sitio web ordenado y más seguro ya que le usuario solo tiene acceso a los niveles de presentación en nuestro caso serán el acceso al sitio web y la visualización de las graficas.

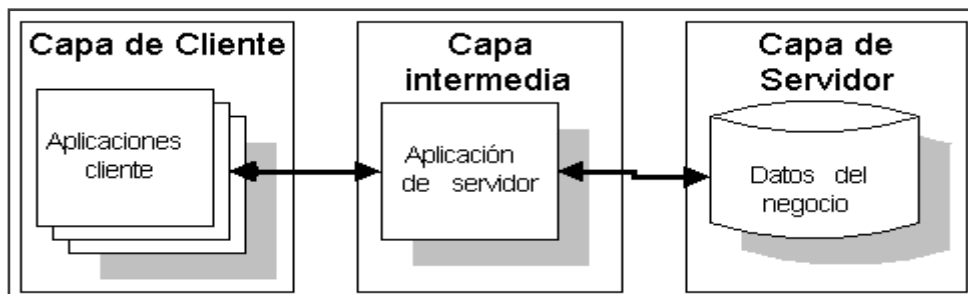


Figura 5.2 : Modelo 3 capas.

En la Figura 5.2 se muestra la arquitectura en 3 capas, la capa de cliente o capa de presentación típicamente son las hojas en HTML o PHP en donde los clientes acceden a los sitios web llenando los formularios necesarios para acceder a este. La capa intermedia o capa de negocio contendrá las CGI necesarias para el negocio de los clientes en el servidor web, entendiéndose por negocio: la creación de un usuario, el inicio de sesión de un usuario y visualización de las graficas (potencia vs tiempo y Energía acumulada vs tiempo). La capa de datos (capa de servidor según figura 5.1) en este caso contendrá las bases de datos en Mysql en donde se crearan las tablas que guardaran los datos de los usuarios, y los ID de los medidores. Los usuarios se relacionaran con un ID de medidor, para poder identificar el medidor de el usuario que accede al sitio web. Como se muestra en la Figura 5.1 los clientes tendrán acceso a la capa de presentación y para llegar a los datos tendrán que ingresar por medio de la capa intermedia o de negocio.

Es de importancia resaltar que los medidores ejecutan los procedimientos remotos para guardar las mediciones, estos procedimientos pertenecen a la capa 2 o capa de negocio por lo tanto los medidores tendrán acceso directo a la capa de negocio o intermedia por medio de XML-RPC, ya que los medidores no requieren de una capa de presentación para interactuar con el servidor web.

La capa de presentación contiene las páginas en HTML para el negocio del cliente con la capa lógica de negocio, en este caso se programaron dos páginas en HTML las cuales se encuentran contenidas en el directorio `/www/powermeter/` del sistema operativo Ubuntu 10.11 como se muestra en la Figura 5.3:

```
Directorio /www/powermeter/
Index.html
newusuario.html
```

Figura 5.3 : Archivos HTML de la capa de presentación.

Estos archivos se encuentran en el CD de anexo en `/servidor/www/powermeter/`, y su ejecución en el servidor se muestra a continuación:

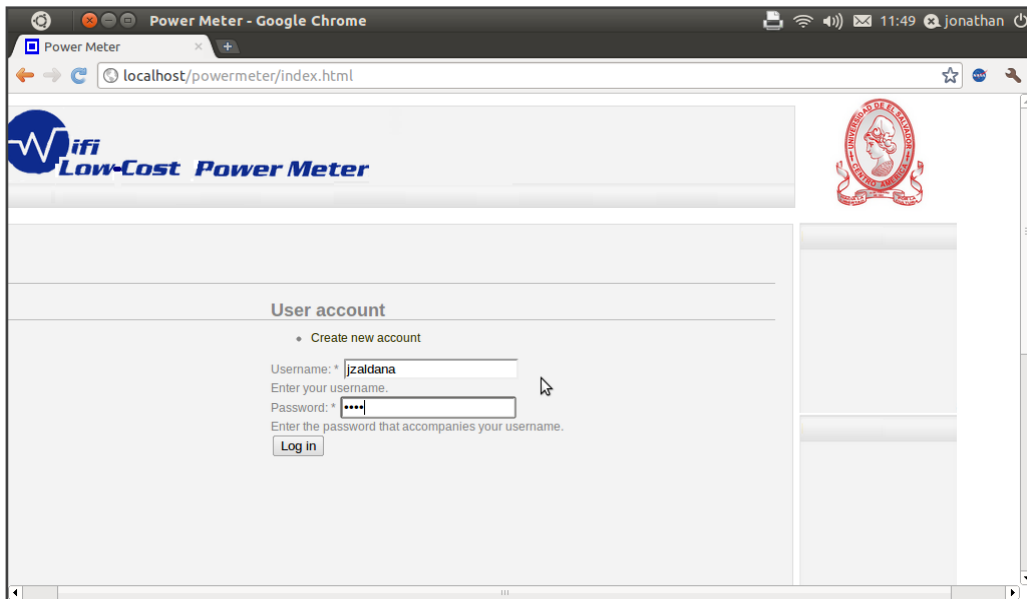


Figura 5.4 : Inicio de sesión index.html:

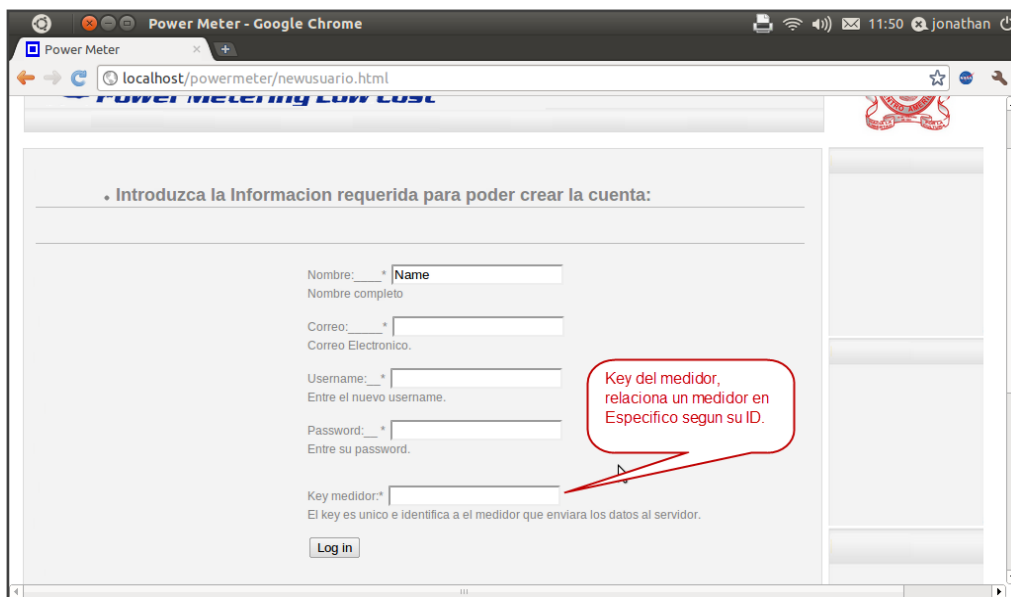


Figura 5.5 : creación de un nuevo usuario imagen del navegador web Chrome.

La Figura 5.4 muestra la sesión de inicio. El usuario tiene que llenar el formulario con el nombre de usuario y password y así poder pasar a la capa de negocio. Si aun no se está dado de alta se debe acceder al vínculo de creación de usuario. Se debe llenar el formulario que se muestra en la Figura 5.5. Los campos requeridos son: los datos del usuario y la llave de acceso (*key*) del medidor, el *key* relaciona un ID de medidor como se muestra continuación en la Figura 5.6:

id	estado	key	device	tipo
1	S	1234567891	27f8cbad4babe36ecc67e8179f6fc287	1

Figura 5.6 : Relación del Key y el ID.

El *key* es una cadena de 10 caracteres numéricos, relacionada con el identificador (ID) del medidor. Los ID de los medidores son los mismos que se mostraron en la Tabla 3.4. La idea de usar un *Key* es que si un usuario adquiere un medidor, éste tendrá un *key* para poder registrarse. También, el *key* relaciona la cuenta de un usuario con un medidor específico. De esta manera el usuario visualiza las gráficas de las mediciones obtenidas con su medidor.

La capa de lógica de negocio o capa intermedia según la Figura 5.1 esta utiliza CGI escritas en PHP y en PERL estas CGI se encuentran en el directorio */cgi-bin/*. El contenido de este directorio se muestra en la Figura 5.7:

```

Directorio /cgi-bin/
Login.php          AddEnergy.cgi
newusuario.php    AddWatts.cgi
potencia.php      cargar.sh
energia.php       cargarE.sh
graficar.sh
graficarE.sh
/ graficas_Potencia/
/ rrd_database/

```

Figura 5.7 : Contenido del directorio cgi-bin.

Las CGI marcadas en color rojo son los procedimientos remotos que los medidores ejecutan por medio de XML-RPC. Estas CGI se pueden encontrar en el CD de anexo en: */servidor/cgi-bin/*.

La capa de negocio se puede dividir en dos partes el negocio de los usuarios y el negocio de los medidores.

Para ingresar a la capa lógica de negocio, los usuarios tienen que llenar los formularios provistos en la capa de presentación Figuras 5.4 y 5.5. Al acceder un usuario a la capa de negocio podrá visualizar las gráficas de potencia y energía eléctrica. Ésto lo realiza las CGI *potencia.php* y *energia.php*. A su vez la CGI *potencia.php* ejecuta la CGI *graficar.sh*, la cual realiza la gráfica de las mediciones. La CGI *energia.php* ejecuta a *graficarE.sh*. La CGI

*graficar.sh* está programada con comandos Shell de Linux. Pero, para generar los gráficos ésta utiliza el programa RRDTOOL. En el Anexo C se describe la utilización de RRDTOOL.

A continuación se describe el flujo que llevan los datos en la arquitectura de 3 capas. En la Figura 5.8 se muestra el flujo de los datos al ingresar un usuario a la página web.

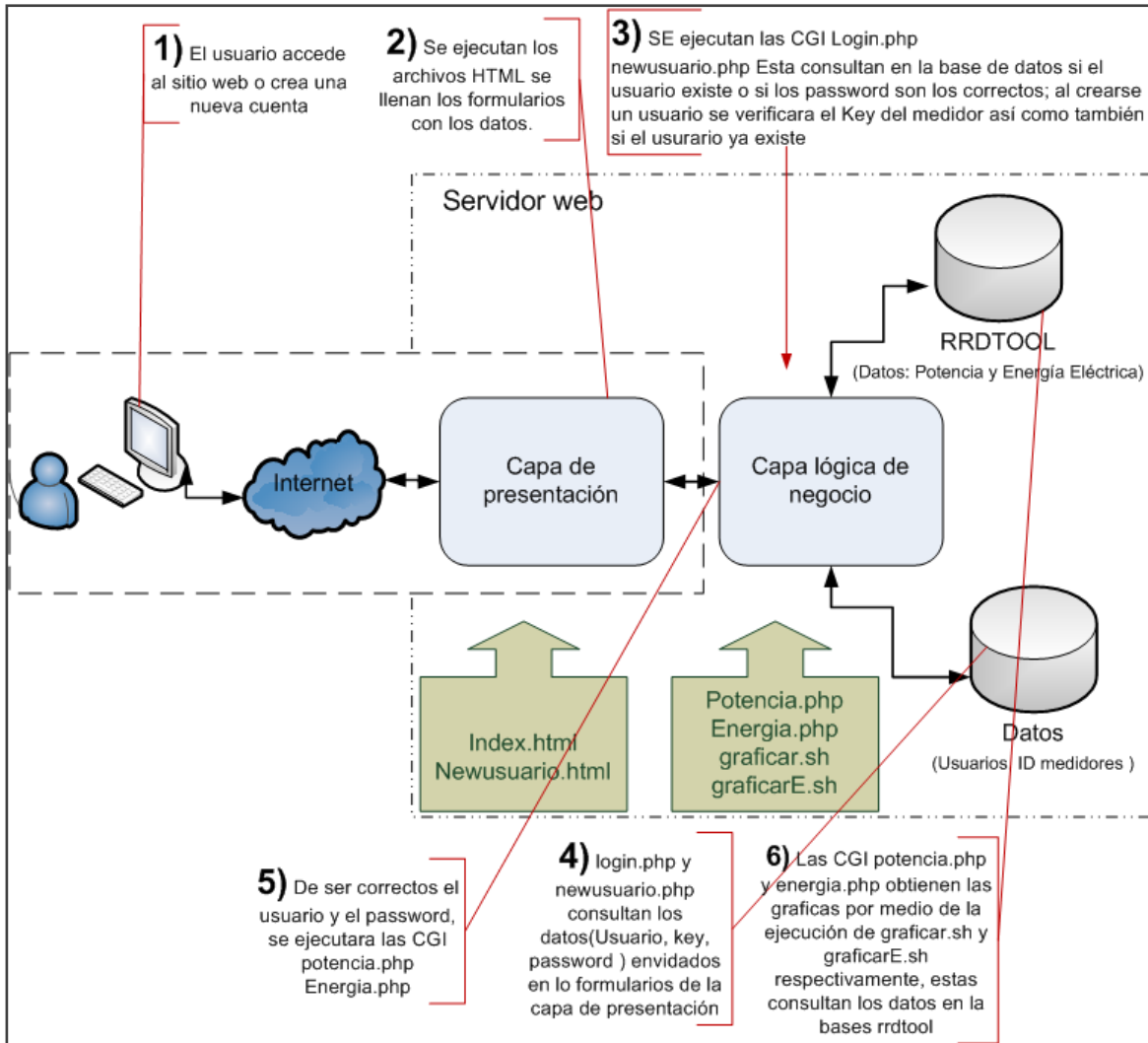


Figura 5.8 : flujo del negocio de los usuarios.

En las figuras 5.4 y 5.5 se mostraron las imágenes del inicio de sesión desde un navegador web en este se puede introducir el usuario y el password y así poder ingresar al sitio web según el flujo del esquema de la Figura 5.8. De ser correctos el usuario y el password (pasos del flujo 1-4) se enviara la respuesta al navegador web del usuario y este podrá visualizar las graficas (ejecución de las CGI *potencia.php* o *energía.php*. En la Figura 5.9 se muestra la respuesta en el navegador web del cliente:

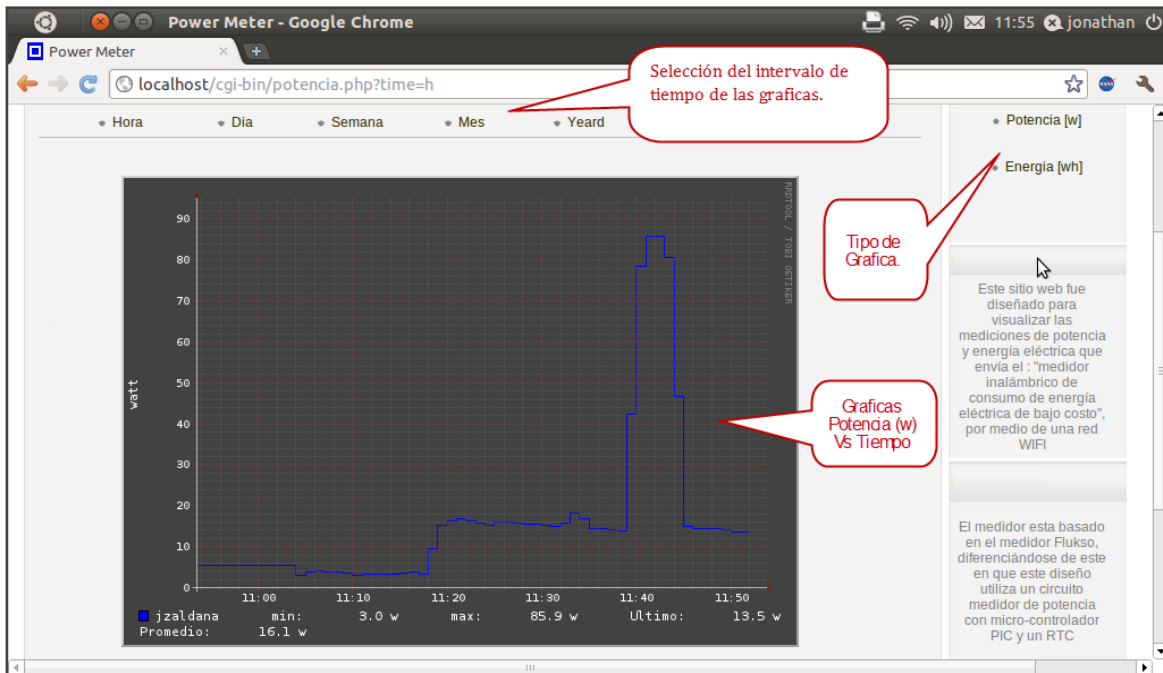


Figura 5.9 : graficas en el servidor web vistas desde el navegador Chrome.

En la Figura 5.9 se muestra la pagina web generada al ejecutarse la CGI *potencia.php*, esta ejecuta el script *grafica.sh* el cual genera la grafica en RRDTOOL que se muestra en la Figura 5.9. En esta página web también se puede elegir el intervalo de tiempo de presentación de las graficas así como el tipo de grafica: potencia en watt Vs tiempo y Energía en Watt Hora Vs tiempo. Como se puede apreciar esta graficas son similares a las obtenidas en el servidor web del medidor Flukso Figura 2.6.

Los medidores enviaran los datos por medio de XML-RPC, como se detallo en la sección 4.2, estos ejecutarán los procesos *AddEnergy.cgi* y *AddWatts.cgi* estos procesos corresponden a la capa lógica de negocio. Por tanto los medidores serán los únicos clientes del servidor web que podrán interactuar con la capa lógica de negocio directamente esto es así porque los medidores no necesitan una presentación grafica en un navegador web para poder enviar las mediciones. Esta es una de las ventajas de utilizar XML-RPC.

Recordando del capítulo 4 sección 4.2 el segmento de código en LUA que envía las mediciones de potencia eléctrica por medio del formato XML-RPC como se muestra a continuación en la Figura 5.10:

```
local ok, res = xmlrpc.http.call ("http://192.168.1.180/cgi-bin/AddWatts.cgi", sample.AddWatts, Device, Tabla[1], Tabla[2], Tabla[3], Tabla[4], Tabla[5], Tabla[6], Tabla[7], Tabla[8], Tabla[9], Tabla[10])
```

Figura 5.10 : Ejecución del procedimiento remoto AddWatts.cgi con XML-RPC.

En el segmento de código que se muestra en la Figura 5.10, muestra resaltado en azul el procedimiento a ejecutar en el servidor web para guardar las mediciones. El procedimiento *AddWatts.cgi* pertenece a la capa lógica de negocio y está programado en el lenguaje PERL. En donde se utilizó la librería *Frontier::RPC2* la cual nos permitirá realizar el servicio XML-RPC en el servidor web. Los pasos para instalar esta librería de PERL se pueden encontrar en [43].

El procedimiento *AddWatts.cgi* se muestra a continuación (el procedimiento *AddEnergy.cgi* es similar) en La Figura 5.11:

```
#!/usr/bin/perl -w

use strict;
use Frontier::RPC2;

sub AddWatts {
    my ($Base_rrd1, $x, $y) = @_ ;

    my @table = @_ ;
    my $Base_rrd = $table[0];
    my $i;
    my $comp;
    my $Tiempo;
    my $Potencia;

    for($i=1; $i<=10; $i++){

        $comp = ($i % 2);
        if($comp != 0 ){
            $Tiempo = $table[$i];
            $Potencia = $table[$i+1];
        }

        system("./cargar.sh $Base_rrd $Tiempo $Potencia");

    }

    return {'Tiempo' => $x, 'Potencia' => $y, 'BASE'=> $Base_rrd};
}

process_cgi_call({'sample.AddWatts' => \&AddWatts});
```

Figura 5.11 : Procedimiento *AddWatts.cgi*

El procedimiento *AddWatts.cgi* Utiliza la librería *Frontier::RPC2* como se menciona anteriormente con esta hemos programado el servicio XML-RPC para guardar las mediciones de potencia y energía eléctrica. Esta CGI a su vez utiliza la función *AddWatts* en donde se guardan las mediciones en la base de datos de RRDTOOL, esto se realiza en el

segmento de código: `system("./cargar.sh $Base_rrd $Tiempo $Potencia");` esta función ejecuta los comandos del Shell de Linux y ejecuta el script `caragar.sh` el cual se muestra a continuación en la Figura 5.12:

```
#!/usr/bin/env bash

if [ "$#" != "3" ]; then
    echo "Usage: $0 <nameRRD> <Unix Time> <Power> "
    echo
    echo "Example: $0 all 1302369817 300"
    exit
fi;

NAME=$1
DB=$2
OUTPUT=$3

RRDTOOL update /cgi-bin/rrd_database/$NAME.rrd $DB:$OUTPUT
```

Figura 5.12 : script `cargar.sh`; este guarda las mediciones en la base de datos de RRDTOOL.

El script `cargar.sh` (`cargarE.sh` en el caso de la energía) utiliza la fusión de `RRDTOOL update` [44] esta se utiliza para guardar los datos en las bases de RRDTOOL. Los datos que se introducen con esta función son el tiempo en el formato Tiempo Unix y las mediciones ya sea potencia o energía (según el script mostrado la base RRDTOOL se encuentra en el directorio `/cgi-bin/rrd_database/` ). A Continuación en la Figura 5.13 se muestra la forma de uso del script `caragar.sh`:

```
jonathan@ubuntu:$. ./cargar.sh 27f8cbad4babe36ecc67e8179f6fc287.rrd Unixtime
potencia
```

Figura 5.13 : forma de uso de `cargar.sh`

El script `cargar.sh` guarda las mediciones y el tiempo en que fueron tomadas en la base de datos `RRDTOOL`: `27f8cbad4babe36ecc67e8179f6fc287.rrd`. El nombre de esta base de datos como corresponde a el ID del medidor que está enviando las mediciones. De esta manera las mediciones enviadas al servidor web por el medidor con el ID `27f8cbad4babe36ecc67e8179f6fc287` se guardan en la base correspondiente a este: `27f8cbad4babe36ecc67e8179f6fc287.rrd`. De esta manera se concluye el diseño de capa lógica de negocio a continuación se detalla el diseño de la capa de datos.

## 5.2 Diseño de la capa de datos

Al igual que la capa lógica de negocio la capa de datos interactúa tanto con peticiones de los usuarios como también con las peticiones de los medidores. En la arquitectura del servidor web mostrada en la Figura 5.1 se puede observar que la capa de datos esta en el último nivel y que solo las CGI y scripts de la capa de negocio pueden consultar los datos que contiene esta.

La capa de negocio se dividió en 2 partes: el negocio del cliente y el negocio de los medidores, también la capa de datos sigue esta misma lógica ya que los datos de los usuarios se guardaran en bases de datos en MYSQL y las mediciones enviadas se guardaran en las bases circulares de RRDTOOL.

### 5.2.1 Información de los usuarios en tablas de MYSQL

Para guardar los datos de los usuarios que accederán al servidor web se creó una base de datos llamada *base*. En esta se creó las tablas que contienen la información de los usuarios así como la información de los medidores. Los pasos para realizar la instalación de MYSQL y la creación de la base de datos y las tablas que contendrá esta se pueden encontrar en el Anexo B.

Para identificar los medidores que pueden enviar mediciones se creó la tabla *meter* como se muestra a continuación en la Figura 5.14:

Tabla meter:

id	estado	key	device	tipo
1	S	1234567891	27f8cbad4babe36ecc67e8179f6fc287	1
2	S	1234567892	37f8cbad4babe36ecc67e8179f6fc287	1
3	S	1234567893	47f8cbad4babe36ecc67e8179f6fc287	1
4	S	1234567894	57f8cbad4babe36ecc67e8179f6fc287	1
5	S	1234567895	67f8cbad4babe36ecc67e8179f6fc287	1
6	S	1234567896	77f8cbad4babe36ecc67e8179f6fc287	1
7	N	1234567897	87f8cbad4babe36ecc67e8179f6fc287	1
8	N	1234567898	97f8cbad4babe36ecc67e8179f6fc287	1

Figura 5.14: Tabla meter, esta contiene la información de los medidores.

La tabla de MYSQL *meter* posee las columnas: *id*, *estado*, *key*, *device* y *tipo*. La columna *device* contiene los ID de los medidores que pueden enviar datos al servidor web cada ID se relaciona con un *key* único. Si un usuario quiere crear una cuenta en el servidor web este deberá de poseer un medidor el cual contendrá un *key* y así se podrá relacionar el medidor del usuario con el ID.

Para guardar la información de los usuarios se creó la tabla usuarios como se muestra a continuación en la Figura 5.15:

```

Tabla usuarios:
+-----+-----+-----+-----+
| id | nombre | mail | username | password |
+-----+-----+-----+-----+
| 1 | medidor1 | medidor3@yahoo.com | medidor1 | root1 |
| 2 | medidor2 | medidor3@yahoo.com | medidor2 | root2 |
| 3 | medidor3 | medidor3@yahoo.com | medidor3 | root3 |
| 4 | medidor4 | medidor1@yahoo.com | medidor4 | root4 |
| 5 | medidor5 | medidor2@yahoo.com | medidor5 | root5 |
| 6 | medidor6 | medidor3@yahoo.com | medidor6 | root6 |
+-----+-----+-----+-----+

```

Figura 5.15: Estructura de la tabla usuarios.

En la tabla *usuarios* se guardaran: los nombres de los usuarios, el correo electrónico, el username y el password.

Cuando se crea un usuario se deberá de llenar el formulario de la Figura 5.5, En donde se muestran los campos que requiere la tabla *usuarios*. Además de la información del usuarios se requiere que este introduzca el *key* del medidor que el usuario posee y así relacionar la información del medidor que se encuentra en la tabla *meters* con el usuario creado (todo esto ocurrirá si el *key* introducido es válido). Para relacionar los usuarios con un medidor en específico se creó la tabla *usuario\_x\_meter* la cual se muestra a continuación en la Figura 5.16:

```

Tabla usuario_X_meter
+-----+-----+-----+-----+
| id_rel | id_usuario | id_meter | fecha_compra |
+-----+-----+-----+-----+
| 1 | 1 | 1 | 2011-10-12 |
| 2 | 2 | 2 | 2011-10-12 |
| 3 | 3 | 3 | 2011-10-12 |
| 4 | 4 | 4 | 2011-06-20 |
| 5 | 5 | 5 | 2011-10-11 |
| 6 | 6 | 6 | 2011-06-21 |
+-----+-----+-----+-----+

```

Figura 5.16: Estructura de la tabla usuario\_x\_meter.

La tabla *usuario\_x\_meter* relaciona las tablas *usuarios* y *meters* por medio de la llave principal de la tabla (*id* de tabla). La relación que se obtiene con la tabla *usuario\_x\_meter* es de uno a uno, dicho de otra manera un usuario con *id\_usuario* (llave principal del usuario

en la en la fila de la tabla usuarios) puede poseer un único medidor con *id\_meter* (llave principal del medidor en la en la fila de la tabla medidores).

Toda la lógica que se ha planteado para las tablas mostradas se realiza en las CGI de la capa de negocio, para la creación de un nuevo usuario la lógica se encuentra en la CGI *newusuario.php*.

La creación de las bases de datos en RRDTOOL y como se obtienen la graficas de los datos de estas se detalla en el Anexo C.

A lo largo del presente capitulo se ha mostrado el diseño de la arquitectura del servidor web según la figura 5.1, como hemos visto este solo utiliza en su progresión tecnologías GNU como los son: PHP, PERL, RRDTOOL, APACHE, MYSQL y LINUX. Esto se realizo así para seguir con la línea de bajo costo del *medidor inalámbrico de consumo de energía eléctrica de bajo costo*, el cual se diseño en el presente trabajo de graduación. El servidor web forma parte del sistema del medidor englobando así no solo el diseño y construcción del medidor sino también el diseño del servidor web.

## 6. Línea futura del Medidor Inalámbrico de Consumo de Energía Eléctrica de Bajo Costo.

Se propone la utilización de un circuito medidor de potencia trifásico, pero siguiendo la misma lógica de utilizar los transductores de corriente (pinzas de corriente) que se ha utilizado en el presente trabajo.

### 6.1 Uso del router DRAGINO

Los routers DIR-300 que se utilizaron como sistemas embebido para el envío de las mediciones vía WIFI fueron dejados de fabricar por DLINK. En un proyecto futuro se puede utilizar el router DRAGINO [45], este router presenta las ventajas de ser de hardware y software libre en la Figura 6.1 se muestra una imagen de este router:



Figura 6.1: Router Dragino.

El diseño del router DRAGINO, surgió por la necesidad de los diseñadores del medidor Flukso de utilizar un router el cual fuese open hardware y así ya no utilizar el WAP-2102(propiedad de ABOCON). Otra de las ventajas de utilizar este router es que este ya posee los conectores para las pinzas de corriente como se muestra en la Figura 6.2

En la Figura 6.3 se muestran los circuitos del router DRAGINO, se puede apreciar el puerto UART de éste. En la misma ubicación del Puerto UART se encuentra el puerto SPI estos tienen una mejor accesibilidad que en el DIR-300. Se muestra también en la Figura 6.3 el

sensor medidor de potencia que utiliza la versión 2 del medidor Flukso el cual salió al mercado el mes de Octubre del 2011 [3].



Figura 6.2: Conectores para entradas analógicas.



Figura 6.3: circuitos del router Dragino.

## 6.2 Topología de red Mesh en los medidores de potencia

Básicamente son redes con topología de infraestructura, pero que permiten unirse a la red a dispositivos que, a pesar de estar fuera del rango de cobertura de los AP o nodos, están dentro del rango de cobertura de algún dispositivo WIFI que directamente o indirectamente esté dentro del rango de cobertura del AP; también permiten que los dispositivos WIFI se comuniquen, independientemente del AP, entre sí, es decir, los dispositivos que actúan como emisores pueden no mandar directamente sus paquetes al AP sino que pueden pasárselos a otros dispositivos WIFI para que lleguen a su destino, que caracteriza de las redes con topología ad-hoc.

La creación de una red Mesh con los medidores de potencia nos daría la ventaja de tener el apoyo de los medidores conectados en la red, si un medidor de la red se desconecta o deja de funcionar el que estaba conectado a este buscaría otra ruta para el envío de las mediciones en la Figura 11.4 se muestra un esquema de red Mesh con los medidores:

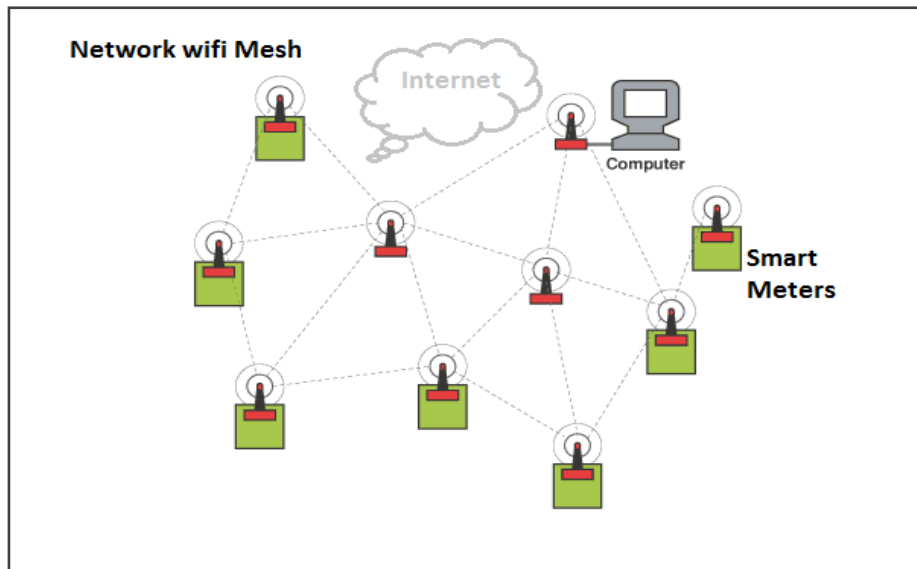


Figura 6.4 : Medidores conectados en una red Mesh

La configuración de red Mesh en OpenWrt es posible mayor información para realizar esto lo podemos encontrar en [46].

## 7. Conclusiones

- El medidor de energía diseñado y construido a lo largo del presente trabajo de graduación se ha sido diseñado para que su construcción resulte de bajo costo. Esto se puede comprobar con los resultados en la Tabla F.1 la cual resume el análisis económico de este medidor, resultando el precio de construcción de este en \$120.93 USD. Según la Tabla F.1 si se construyesen 100 PCB el costo se reduciría en \$89.10 USD. En comparación a muchos medidores de este tipo que se encuentran en el mercado el construido en el presente trabajo resulta más económico.
- La instalación de OpenWrt en el router DIR-300 le otorga a este las cualidades de un sistema embebido en un equipo de menor precio como lo es DIR-300. Otra de las ventajas que otorga utilizar el sistema operativo Linux OpenWrt es la flexibilidad para programar aplicaciones en el lenguaje LUA. Como se realizó con el programa *flukso.lua*.
- Las mediciones de potencia eléctrica obtenidas con el *circuito medidor de potencia* representa una aproximación de la potencia activa de la red eléctrica. Por no poseer este un muestreo del voltaje de la red eléctrica no se toma en cuenta el factor de potencia ni las variaciones de amplitud que posea la señal de voltaje. Al compararse el medidor construido con un analizador de carga FLUK se estima un error de 6.3% en las mediciones.
- Las graficas de potencia versus tiempo y energía versus tiempo, obtenidas en el servidor web, son una herramienta muy útil para poder monitorear el consumo energético. Esto se puede realizar a partir de las gráficas obtenidas en el servidor web. El usuario puede observar los picos de potencia y la demanda que está teniendo en su hogar. Con esta información se esperaría que el usuario mejorara su hábito de consumo de energía eléctrica.

## Anexo A

### A. Instalación del firmware OpenWrt al router

El medidor de energía Flukso utiliza para el envío de los datos a través de la red WIFI el router WAP-2102, Figura A.1. Al cual se le instaló el sistema operativo OpenWrt 10.03, este sistema operativo posee un núcleo Linux. Para lograr la comunicación del microcontrolador con el router, el diseñador del Flukso creó un paquete el cual posee los programas necesarios para la comunicación a través del puerto UART y el envío de las mediciones al servidor web.

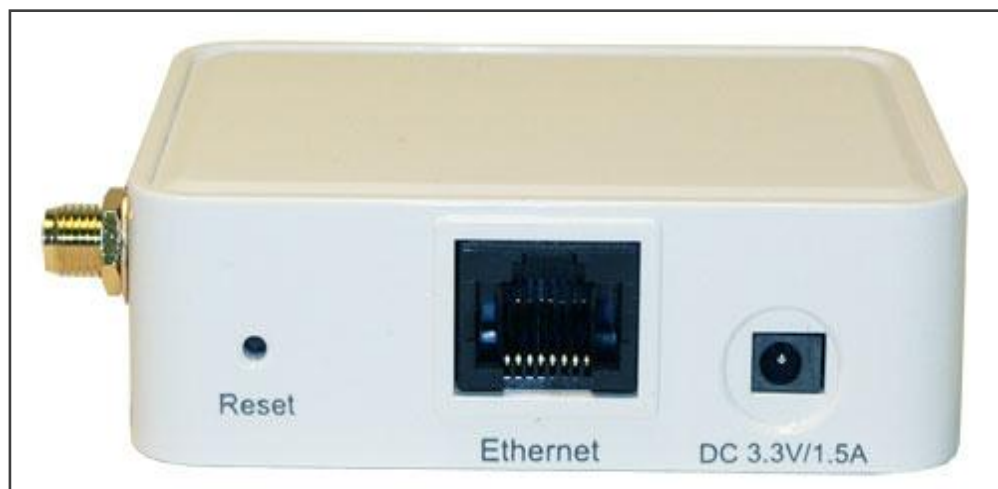


Figura A.1: Router WAP-2102 el cual utiliza Flukso.

Los archivos binarios de OpenWrt 10.03 Flukso pueden ser descargados de la página web del creador [47]. Así como también se puede descargar de esta misma web el paquete *flukso\_1.0-1\_mips.ipk*, es de importancia destacar que todo el software del medidor de energía Flukso está bajo licencia GNU.

El router WAP-2102 posee un procesador Atheros-AR2317 y una capacidad de memoria flash para el software de 4MB, este router no se encuentra a la venta en la región es por ello que para el diseño de nuestro medidor de potencia se optó por utilizar el router D-LINK DIR-300 el cual resulta más económico que el WAP-2102, el DIR-300 posee el mismo procesador y la misma capacidad que el WAP-2102 por tanto no habrá problema al instalarle el sistema operativo OpenWrt, en la Figura A.2 se muestra una imagen del router DIR-300.



Figura A.2 : Router D-LINK DIR-300.

## A.2 Procedimiento para la instalación si el DIR-300 posee el software del fabricante

A continuación se detallaran todos los pasos necesarios para la instalación del firmware al DIR-300 algunos pasos para esta instalación fueron tomados de [48]. La instalación será realizada a través del sistema operativo Windows en donde deben estar instalados previamente los programas *SolarWinds TFTP Server* y *putty.exe*. También es necesario crear los script *redboot.bat* y *redboot1.bat*. A continuación en la Figura A.3 se muestra el contenido del script *redboot.bat*:

```
#redboot.bat
rem echo off
:start
rem -----
ping 192.168.1.1 -n 1 -w 1 > NULL
IF ERRORLEVEL 1 goto start
rem putty
rem -----
break
putty.exe telnet://192.168.1.1:9000 -m redboot.txt
exit
```

Figura A.3: Archivo redboot.bat.

El script *redboot1.bat* es similar al script *redboot.bat* la única diferencia de ellos es la dirección IP que utilizan a continuación en la Figura A.4 se muestra dicho script:

```

#redboot1.bat
rem echo off
:start
rem -----
ping 192.168.2551.1 -n 1 -w 1 > NULL
IF ERRORLEVEL 1 goto start
rem putty
rem -----
break
putty.exe telnet://192.168.255.1:9000 -m redboot1.txt
exit

```

Figura A.4 : archivo redboot1.bat.

Los script antes mencionados son necesarios para la conexión de la PC con el Router por medio de *putty* utilizando una sesión *telnet* en donde se accederá al redboot<sup>8</sup> del Router para poder modificar o instalar un nuevo sistema operativo al router.

El procedimiento mostrado a continuación supone que al DIR-300 no se le ha instalado un nuevo firmware, sino que posee el del fabricante.

- 1) Instalación de los programas SolarWinds TFTP Server y putty.exe (este programa es ejecutable y no requiere de instalación).
- 2) Guardar el script *redboot.bat* en la raíz que en nuestro caso es C:\ y a la vez crear un archivo *redboot.txt* que contendrá sola la siguiente línea: ^C.
- 3) Descargar los archivos binarios de la compilación de OpenWrt que utiliza Flukso estos se pueden descargar de la siguiente dirección: <http://www.flukso.net/files/openwrt/FLUKSO-1-0/FLUKSO-1-0-bin.tar.gz>

Luego de descargar el archivo se procederá a descomprimirlo.

4) Cuando se instaló el programa *SolarWinds TFTP Server* se creó en la raíz C:\ un directorio llamado *TFTP-Root* en este directorio se colocaran los archivos binarios como sigue:

- Luego de descomprimir *FLUKSO-1-0-bin.tar.gz* se genera un directorio llamado *bin* el cual contiene los archivos binarios de OpenWrt, se copiaran los siguientes archivos al directorio *TFTP-root*:
  - A) *openwrt-atheros-ubnt2-squashfs*
  - B) *openwrt-atheros-vmlinux.lzma*

---

<sup>8</sup> RedBoot: es un código abierto de aplicaciones que utiliza en tiempo real del sistema operativo en la Capa de abstracción de hardware para proporcionar arranque firmware para sistemas embebidos.

- Descargar *ap61.ram* colocarlo en el directorio *TFTP-root* el archivo se descargara de: <http://www.dd-wrt.com/site/support/other-downloads>
- Descargar *ap61.rom* colocarlo en el directorio *TFTP-root* el archivo se descargara de: <http://www.dd-wrt.com/site/support/other-downloads>

6) Desactivar el antivirus, Desactivar el firewall de Windows (panel de control). Si la PC tiene conectividad inalámbrica ésta debe de ser deshabilitada.

7) Fijar la IP 192.168.20.80 estática ala PC como se detalla a continuación en la Figura A.5:

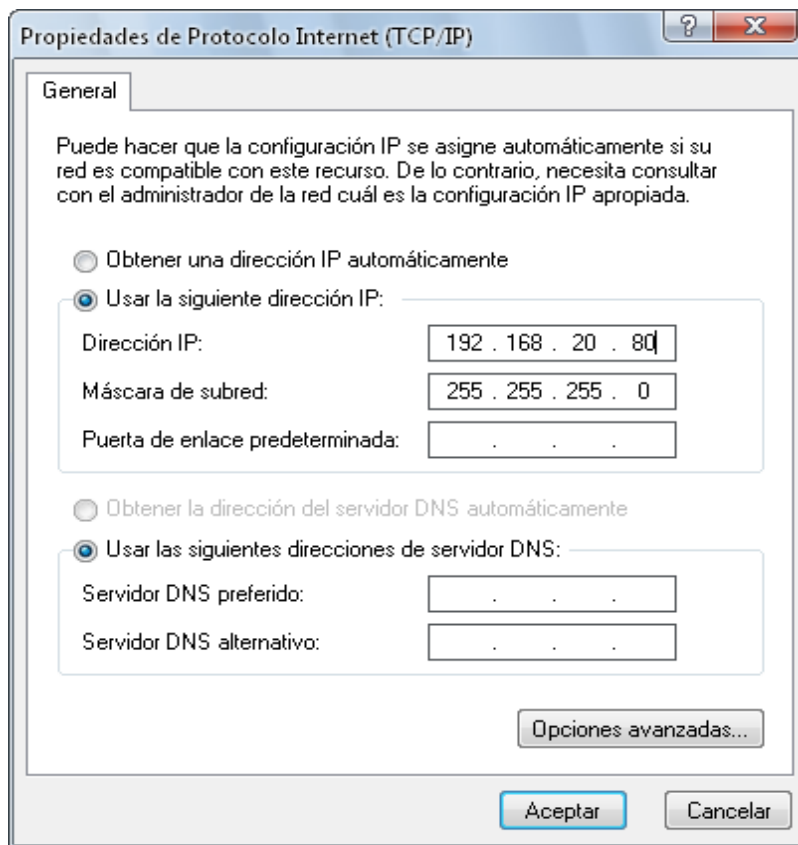


Figura A.5: ventana donde se fijara una IP.

La máscara de subred se configurara según la Figura A.5.

8) Ejecutar *SolarWinds TFTP Server*:

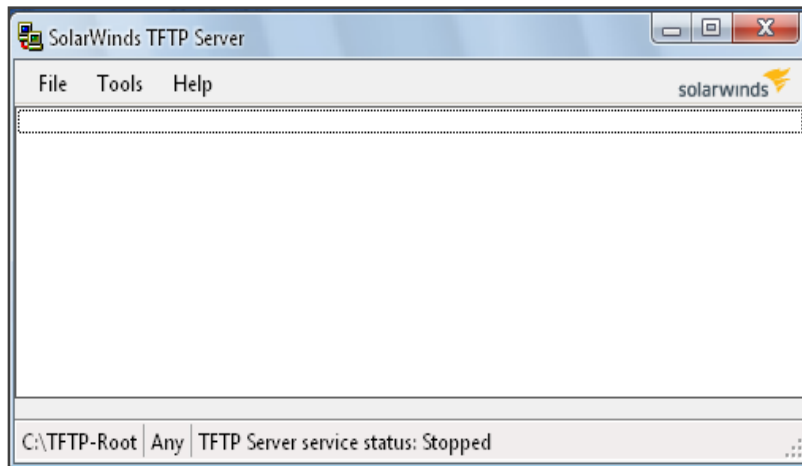


Figura A.6 : servidor de TFTP para Windows.

Accedemos al menú *File configuración*, como se muestra en la Figura A.7, presionamos el botón start el cual inicia el servidor de *TFTP* (son las siglas de Trivial file transfer Protocol; Protocolo de transferencia de archivos trivial.), salimos del menú presionando el botón OK.

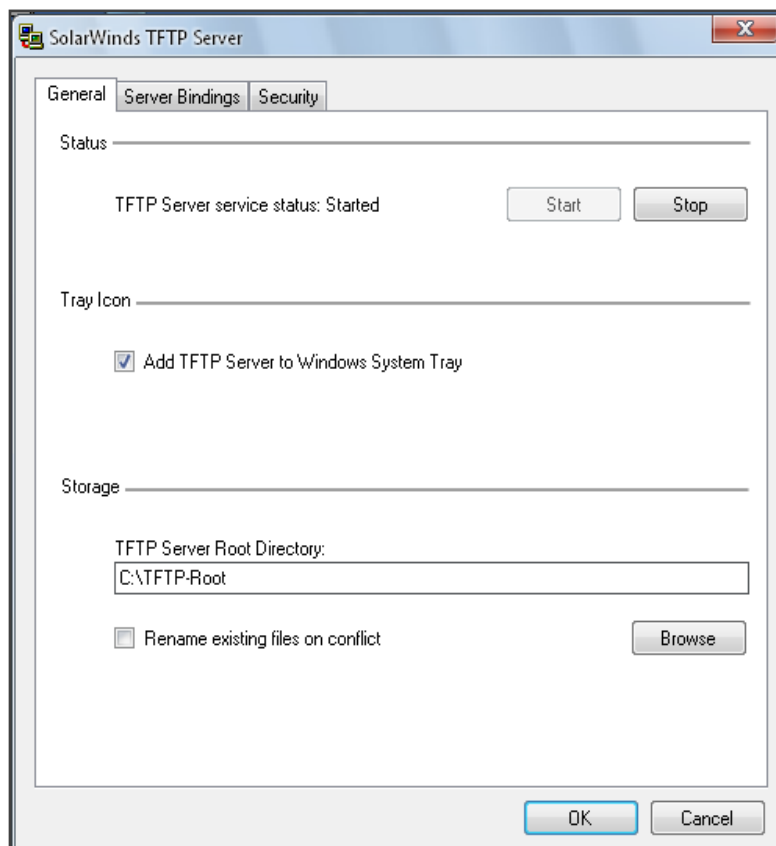


Figura A.7: Configuración del servidor.

9) Conectar el cable UTP al puerto WAN del DIR-300 y al puerto Ethernet de la PC, como se muestra en la Figura A.8:

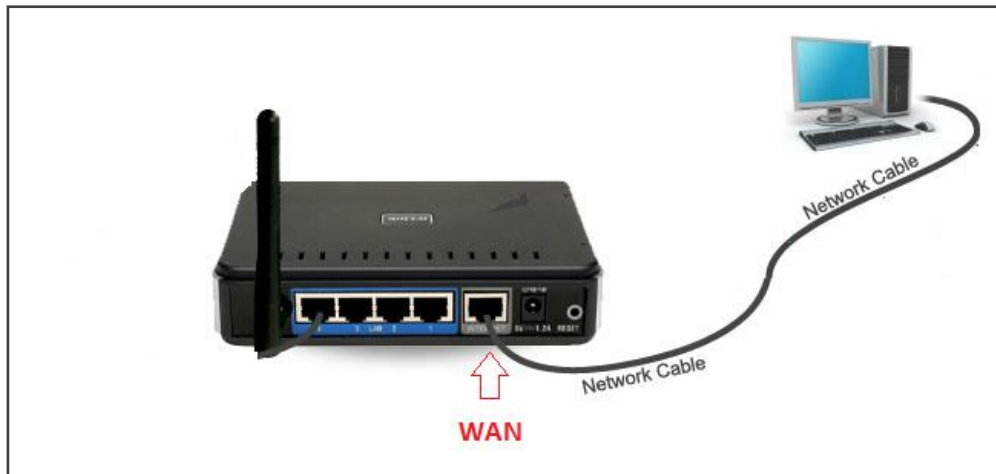


Figura A.8 : Conexión del Router a la PC.

10) Ejecutamos una consola MSDOS en Windows, nos posicionamos en la raíz C:\ y ejecutamos el script "redboot.bat" como se observa en la Figura A.9:

```
ca: C:\WINDOWS\system32\cmd.exe - redboot.bat
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Administrador>cd ..
C:\Documents and Settings>cd ..
C:\>redboot.bat
C:\>rem echo off
C:\>rem -----
C:\>ping 192.168.1.1 -n 1 -w 1 1>NULL
C:\>IF ERRORLEVEL 1 goto start
C:\>rem -----
C:\>ping 192.168.1.1 -n 1 -w 1 1>NULL
C:\>IF ERRORLEVEL 1 goto start
C:\>rem -----
C:\>ping 192.168.1.1 -n 1 -w 1 1>NULL
C:\>IF ERRORLEVEL 1 goto start
C:\>rem -----
C:\>ping 192.168.1.1 -n 1 -w 1 1>NULL
```

Figura A.9: Respuesta al ejecutar redboot.bat.

En la Figura A.9 se observa que el script se queda ejecutando hasta obtener una respuesta a través de ping 192.168.1.1 que es la IP del router.

**11)** Desconectar la fuente de alimentación del router, presionamos el botón de reset del router lo mantenemos presionado y conectamos la fuente de alimentación al router sin dejar de presionar el botón reset aproximadamente 30s luego de esto si los pasos anteriores fueron ejecutados correctamente se iniciará una sesión telnet a través de *putty.exe* con la respuesta siguiente:

```
RedBoot>
RedBoot>
RedBoot>
```

Figura A.10: respuesta al ejecutar el paso 11.

**12)** Cargar el archivo *ap61ram* para actualizar el nuevo *redboot*, lo cual nos permitirá poder cambiar el firmware, digitar en la consola *RedBoot* el siguiente comando, también se muestra en la Figura A.11, la respuesta al digitar el comando *load*.

```
RedBoot>
RedBoot> load ap61.ram

Using default protocol (TFTP)

Entry point: 0x800410bc, address range: 0x80041000-0x800680d8
```

Figura A.11: Ejecución del comando *load*.

Luego ejecute el comando *go* como se muestra en la Figura A.12:

```
RedBoot> go
RedBoot>
RedBoot>
RedBoot>
```

Figura A.12: Ejecución del comando *go*.

El router debería cargar el nuevo *bootloader* y reiniciar automáticamente después de unos minutos. En el caso de permanecer la consola de telnet todavía activa presionamos *Enter* y debería desaparecer en un par de segundos.

**13)** Fijar nuevamente la IP a 192.168.1.2 estática a la PC como se detalla a continuación en la Figura A.13 (a):

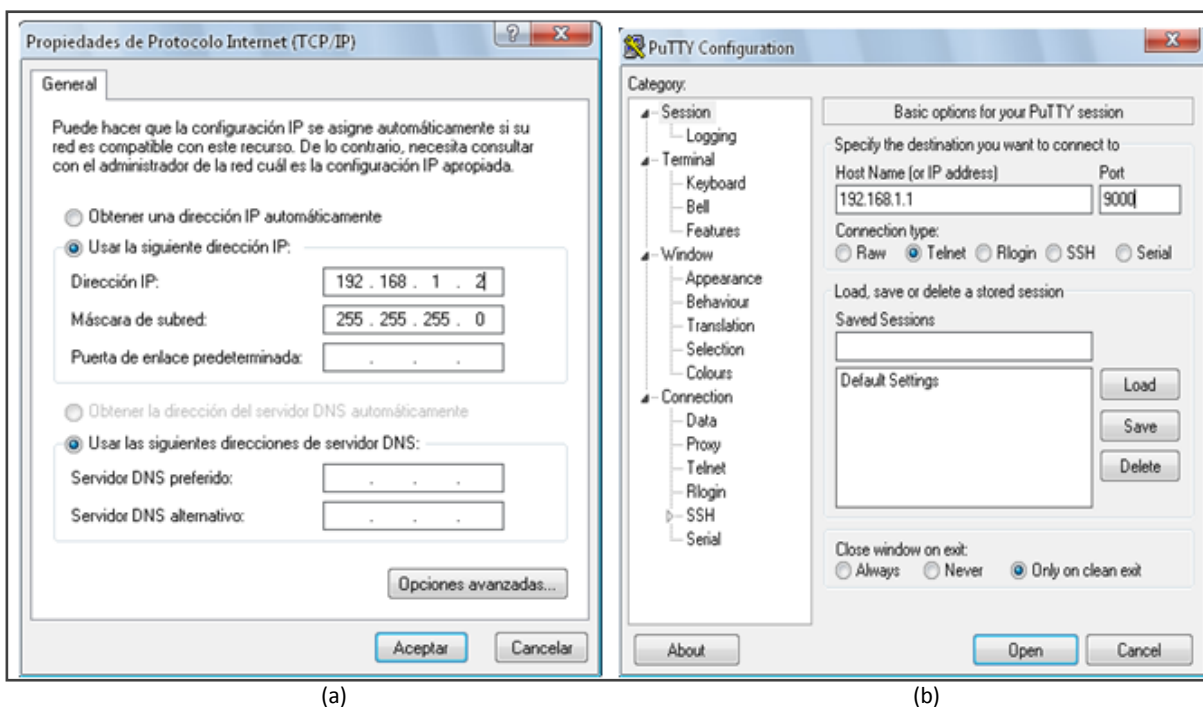


Figura A.13: (a) Fijación e la IP estática según se muestra; (b) Abrir una sesión telnet con PUTTY.

**14)** Ejecute *putty.exe* y inicie una sesión telnet a la dirección 192.168.1.1:9000 como se muestra en la Figura A.13 (b):

La respuesta a la acción anterior se muestra a continuación en la Figura A.14:

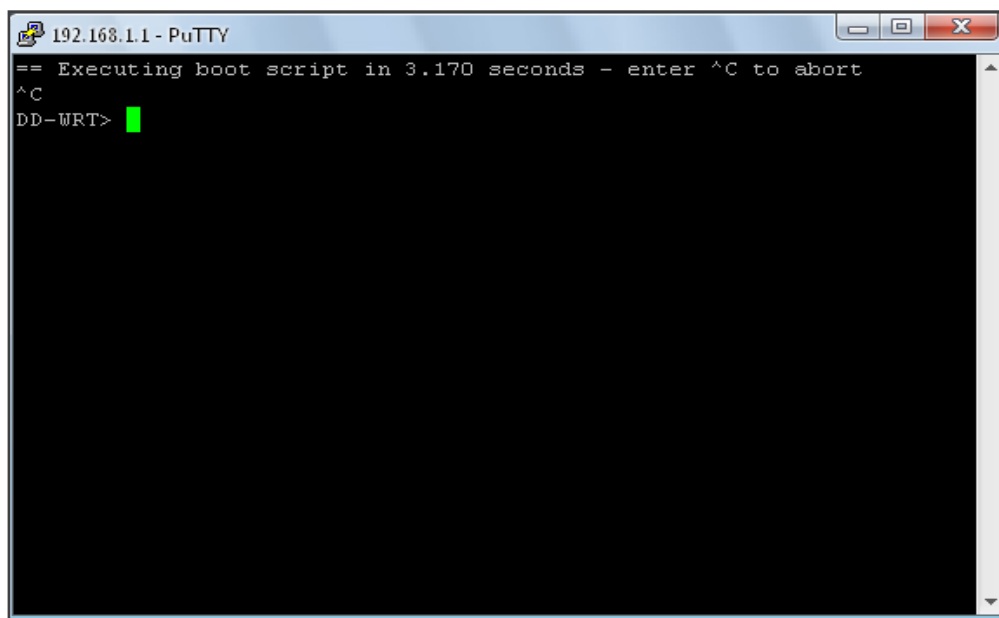


Figura A.14: consola del router con el prom DD-WRT.

La Figura A.14 muestra la consola DD-WRT desde aquí se realizara la instalación del OpenWrt al DIR-300 la serie de comandos para dicha instalación se muestra a continuación en los pasos siguientes.

**15)** Igual que en el paso 11, seguimos digitando los siguientes comandos. En la Figura A.15 se muestra los comandos seguidos de las respuestas de estos.

```
DD-WRT> fconfig -i
Initialize non-volatile configuration - continue (y/n)? y
Run script at boot: false
Use BOOTP for network configuration: true
Default server IP address:
Console baud rate: 9600
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false
Update RedBoot non-volatile configuration - continue (y/n)? y
... Erase from 0xbffe0000-0xbfff0000: .
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
DD-WRT> fis init
About to initialize [format] FLASH image system - continue (y/n)? y
... Erase from 0xbffe0000-0xbfff0000: .
... Program from 0x807f0000-0x80800000 at 0xbffe0000: .
DD-WRT> ip_address -h 192.168.1.2
Default server: 192.168.1.2
DD-WRT> load -r -b %{FREEMEMLO} ap61.rom
Using default protocol (TFTP)
Raw file loaded 0x80080000-0x800a8717, assumed entry at 0x80080000
DD-WRT> fis create -l 0x30000 -e 0xbfc00000 RedBoot
An image named 'RedBoot' exists - continue (y/n)? y
... Erase from 0xbfc00000-0xbfc30000: ...
... Program from 0x80080000-0x800a8718 at 0xbfc00000: ...
... Erase from 0xbffe0000-0xbfff0000: .
... Program from 0x807f0000-0x80800000 at 0xbffe0000: .
DD-WRT> reset
```

Figura A.15: Comandos para cargar un nuevo firmware.

Luego del reset se cerrara la consola telnet esperar 30s y seguir el siguiente paso.

**16)** Luego procederemos nuevamente el paso 14, se reabrirá la consola telnet luego seguir con el paso 17.

**17)** En la Figura A.16 se muestra la serie de comandos en la que cargaremos e instalaremos los archivos binarios de OpenWrt.

```

DD-WRT> ip_address -h 192.168.1.2
Default server: 192.168.1.2
DD-WRT> fis init
About to initialize [format] FLASH image system - continue (y/n)? y
*** Initialize FLASH Image System
... Erase from 0xbffe0000-0xbfff0000: .
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
DD-WRT> load -r -b %{FREEMEMLO} openwrt-atheros-vmlinux.lzma
Using default protocol (TFTP)
Raw file loaded 0x80040800-0x801007ff, assumed entry at 0x80040800
DD-WRT> fis create -e 0x80041000 -r 0x80041000 vmlinux.bin.17
... Erase from 0xbfc30000-0xbfcf0000: .....
... Program from 0x80040800-0x80100800 at 0xbfc30000: .....
... Erase from 0xbffe0000-0xbfff0000: .
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
DD-WRT> load -r -b %{FREEMEMLO} openwrt-atheros-root.squashfs
Using default protocol (TFTP)
Raw file loaded 0x80040800-0x802207ff, assumed entry at 0x80040800
DD-WRT> fis create rootfs
... Erase from 0xbfcf0000-0xbffe0000:
.....
... Program from 0x80040800-0x80220800 at 0xbfcf0000:
.....
... Erase from 0xbffe0000-0xbfff0000: .
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
DD-WRT> fconfig -d
Run script at boot: true
Boot script:
Enter script, terminate with empty line
>> fis load -l vmlinux.bin.17
>> exec
>>
Boot script timeout (1000ms resolution): 5
Use BOOTP for network configuration: false
Gateway IP address: 192.168.1.2
Local IP address: 192.168.1.1
Local IP address mask: 255.255.255.0
Default server IP address: 192.168.1.2
Console baud rate: 4800
GDB connection port: 9000
Force console for special debug messages: false
Network debug at boot time: false
Update RedBoot non-volatile configuration - continue (y/n)? y
... Erase from 0xbffe0000-0xbfff0000: .
... Program from 0x80ff0000-0x81000000 at 0xbffe0000: .
DD-WRT> reset
DD-WRT>

```

Figura A.16: Comandos para cargar un nuevo firmware en este caso OpenWrt.

Es de importancia destacar que la velocidad en baudios de la consola se programó a un valor de 4800 baudios ya que esta es la velocidad a la cual trabajara el puerto UART del microcontrolador a utilizar en la comunicación hacia el router vía UART.

Luego de instalar OpenWrt reiniciamos el router desde el navegador web accedemos a la dirección 192.168.255.1 (dirección configurada en la instalación) y aparecerá la interfaz de comunicación LUCI que se muestra en la Figura A.17:



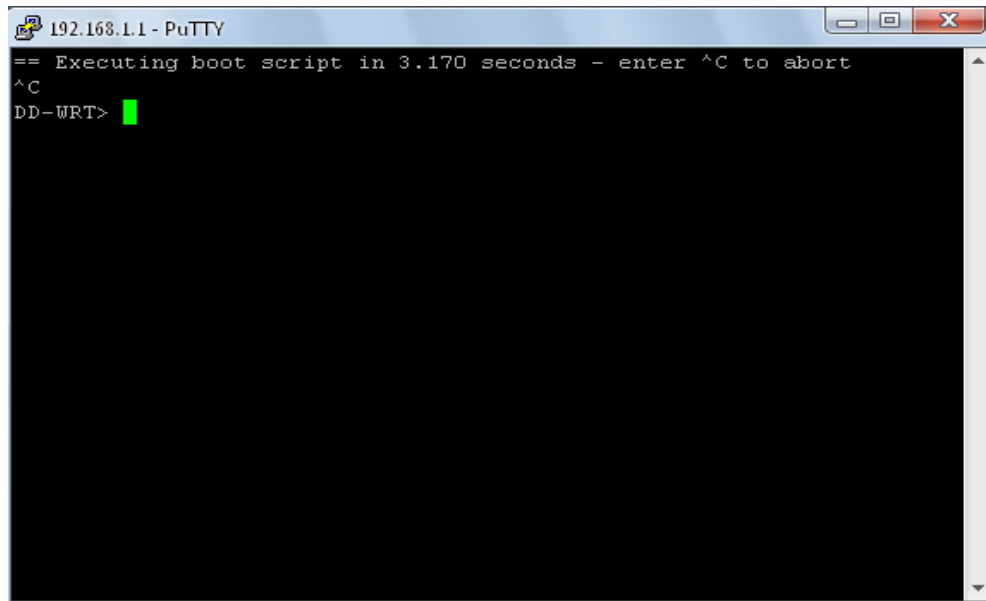
Figura A.17: interfaz de comunicación web.

### A.3 Pasos para la instalación de OpenWrt si el DIR-300 tiene instalado DD-WRT u OpenWrt

Si el router ya posee una instalación OpenWrt o DD-WRT se procederá a la instalación del firmware de Flukso como sigue en los siguientes pasos:

- 1) Repetir pasó 1 anterior.
- 2) Guardar el script *redboot1.bat* en la raíz que en nuestro caso es C:\ y a la vez crear un archivo *redboot.txt* que contendrá sola la siguiente línea: ^C.
- 3) Repetir los pasos 3, 4 y 6.
- 4) ejecutar el paso 13.

5) Desconectar la fuente de alimentación del router y ejecutar el script *redboot1.bat*, presionamos el botón de reset del router lo mantenemos presionado y conectamos la fuente de alimentación al router sin dejar de presionar el botón reset aproximadamente 30s luego de esto si los pasos anteriores fueron ejecutados correctamente se iniciara una sesión telnet a través de putty.exe con la respuesta siguiente:



```
192.168.1.1 - PuTTY
== Executing boot script in 3.170 seconds - enter ^C to abort
^C
DD-WRT>
```

Figura A.18: Consola del router.

Los paso 12 y 15 no se ejecutan porque los archivos *ap61.rom* y *ap61.ram* ya están instalados.

6) Repetir el paso 17.

#### A.4 Instalación del paquete Flukso

Una vez instalado el firmware al router el sistema contiene OpenWrt con la compilación de Flukso, hace falta instalar un paquete el cual contiene los programas necesarios para la comunicación del router a través del puerto UART y para el envío de la información al servidor en paquete se puede descargar de la siguiente dirección:

[http://www.flukso.net/files/openwrt/FLUKSO-1-0/bin/packages/mips/flukso\\_1.0-1\\_mips.ipk](http://www.flukso.net/files/openwrt/FLUKSO-1-0/bin/packages/mips/flukso_1.0-1_mips.ipk)

La instalación del paquete será realizada en un sistema operativo Linux, se procederá a conectar el router al puerto Ethernet de la PC, luego de esto se abrirá una consola de comandos como usuario *root* y se digitaran los comandos mostrados en la Figura A.19:

```
root>ifconfig eth0 192.168.255.2/24
root>ping 192.168.255.1
```

Figura A.19: Fijación de una IP estática.

En la serie de comandos anteriores fijamos una IP a la PC, luego de esto hacemos ping para asegurarnos de que el router esté conectado correctamente.

A continuación en la Figura A.20 se muestra la serie de comando para instalar el paquete, los requisitos para realizar estos comandos son: que la PC tenga acceso a internet y que la comunicación con el router haiga sido exitosa.

```
root>ssh root@192.168.255.1 "opkg install
'http://www.flukso.net/files/openwrt/FLUKSO-1-
0/bin/packages/mips/flukso_1.0-1_mips.ipk'"
root@192.168.255.1's password:
```

Figura A.20: Instalación del paquete Flukso.

En el comando anterior se muestra la respuesta de este si esto aparece los pasos han sido realizados correctamente sino es así se tendría que regresar a los comando anteriores. Según se muestra en la respuesta nos pide el password del router el cual es por defecto root, luego de introducir el password el paquete será instalado.

Al instalarse correctamente el paquete procederemos a darle start al demonio de Flukso como se muestra a continuación:

```
root>ssh root@192.168.255.1 "/etc/init.d/flukso enable; /etc/init.d/flukso
start"
root@192.168.255.1's password:
```

Figura A.21: arranque del demonio flukso.

Introducimos el password y listo el router ahora está habilitado para poder recibir las mediciones de la potencia a través del puerto UART y luego mandarlos al servidor.

## Anexo B

### B. Creación de las tablas en MYSQL

A continuación se detallara la instalación de MYSQL y la creación de las tablas utilizadas en el servidor web. En las tablas usuarios y meter se guarda la información de los usuarios y los medidores respectivamente.

En la Figura B.1 se muestran los comandos para la instalación e MYSQL, en este caso desde un terminal en Ubuntu.

```
jonathan@ubuntu:~$ sudo su
[sudo] password for jonathan:
root@ubuntu:/# aptitude install mysql-server
root@ubuntu:/# aptitude install mysql-client
```

Figura B.1: Instalación de mysql.

Luego de instalar MYSQL configuramos la contraseña del administrador como se muestra en la Figura B.2:

```
root@ubuntu:/# usr/bin/mysqladmin -u root password clavenueva
```

Figura B.2: configuración de la clave de administrador de mysql.

Una vez asignada una contraseña al administrador, nos conectamos con el servidor MYSQL de la manera siguiente:

```
root@ubuntu:/home/jonathan# mysql -h localhost -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 35
Server version: 5.1.58-1ubuntu1 (Ubuntu)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.
mysql>
```

Figura B.3: Inicio de mysql server.

En la Figura B.3, *localhost* es la dirección del host y *root* el nombre de usuario. A continuación se introduce el password de la base de datos.

Luego de la instalación y configuración de MYSQL, nos conectamos con el servidor MYSQL. Creamos una base de datos, en la Figura B.4 se muestra la creación de la base de datos que contendrá las tablas.

```
mysql>CREATE DATABASE BASE;
```

Figura B.4: Creación de la base de datos.

A continuación se describe la creación de las tablas en la base de datos *base*. En la Figura B.5 se muestran los comandos para la creación de la tabla usuarios. Esta contendrá la información de los usuarios.

```
mysql> USE BASE;
mysql>CREATE TABLE `USUARIOS` ( `ID` INT(10) UNSIGNED NOT NULL
AUTO_INCREMENT, `NOMBRE` VARCHAR(255) NOT NULL DEFAULT '', `MAIL`
VARCHAR(255) NOT NULL DEFAULT '', `USERNAME` VARCHAR(255), `PASSWORD`
VARCHAR(255), PRIMARY KEY (`ID`) ) ENGINE=INNODB DEFAULT CHARSET=LATIN1;
```

Figura B.5: creación de la tabla usuarios.

La creación de la tabla *meter* se muestra en la Figura B.6, esta tabla contendrá la información de los medidores como se mostró en la Figura 5.14.

```
mysql> USE BASE;
mysql> CREATE TABLE `METER` ( `ID` INT(10) UNSIGNED NOT NULL PRIMARY KEY,
`KEY` INT(10) UNSIGNED NOT NULL, `DEVICE` VARCHAR(255) NOT NULL, TIPO
INT(1), DESCRIPCION VARCHAR(1000)) ENGINE = INNODB DEFAULT CHARSET =LATIN1;
```

Figura B.6: Creación de la tabla meter.

La tabla *usuario\_x\_meter* relacionara los usuarios con los medidores los comando para la creación de esta tabla se muestran en la Figura B.7. La descripción de tabla *usuario\_x\_meter* se mostró en la Figura 5.16.

```
mysql> USE BASE;
mysql> CREATE TABLE `USUARIO_X_METER`(`ID_REL` INT(10) UNSIGNED NOT NULL,
ID_USUARIO INT(10) UNSIGNED NOT NULL, ID_METER INT(10) UNSIGNED NOT NULL,
FECHA_COMPRA DATE, UNIQUE KEY `ID_USUARIO_X_METER`
(`ID_REL`,`ID_USUARIO`,`ID_METER`), CONSTRAINT `F_REL_X_METER` FOREIGN KEY
(`ID_METER`) REFERENCES `METER` (`ID`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `F_REL_X_USUARIO` FOREIGN KEY (`ID_USUARIO`) REFERENCES
`USUARIOS` (`ID`) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE = INNODB
DEFAULT CHARSET = LATIN1;
mysql> ALTER TABLE USUARIO_X_METER ADD UNIQUE(ID_METER);
```

Figura B.7: Creación de la tabla usuario\_x\_meter.

## Anexo C

### C. Gráficos con RRDTOLL

RRDTOOL (*Round Robin Databases*; bases de datos circulares) es un sistema que permite almacenar y representar datos en intervalos temporales (ancho de banda, temperatura etc.). Guarda los datos en una base de datos que no crece en el tiempo. Permite crear gráficas para representar los datos. Una base de datos circular va a contener siempre la misma cantidad de datos. Éstas funcionan bajo el principio de sobrescribir sobre datos antiguos. Se puede describir este tipo de bases de datos como un círculo en el que se van a ir colocando datos. Si empezamos en un punto colocando datos, cuando hayamos dado una vuelta a todo el círculo, habremos llegado al inicio del círculo, y es ahí cuando empezaremos a sobre escribir los datos que recopilamos al principio.

RRDTOOL es una herramienta la cual nos permite realizar gráficos temporales tomado como datos de los gráficos los almacenados en las bases Round Robin del mismo, en la sección siguiente describiremos como se generaron las bases de datos RRDTOOL para cada medidor. Con esta herramienta podremos generar las gráficas de potencia vs tiempo y energía vs tiempo. Los tiempos de visualización de las graficas variaran de la misma manera que el medidor Flukso. A continuación en la Figura C.1 se muestra el script *graficar.sh* el cual utiliza RRDTOOL para generar las graficas:

```
#!/usr/bin/env bash
if [ "$#" != "4" ]; then
    echo "Usage: $0 <rrd-database> <output-filename> <Time>"
    echo
    echo "Example: $0 test.rrd output.png h"
    exit
fi;
DB="/cgi-bin/rrd_database/$1"
OUTPUT="/cgi-bin/graficas_Potencia/$2"
FECHA=$3
USUARIO=$4
RRDTOOL graph $OUTPUT -a PNG --color BACK#424242 --color CANVAS#424242 --
color FONT#FFFFFF --color AXIS#FFFFFF --vertical-label="watt" --full-size-
mode --width 597 --height 424 --alt-autoscale-max --lower-limit=0 --start -
1$FECHA \
    DEF:watt=$DB:watt:AVERAGE \
    DEF:wattmin=$DB:watt:MIN \
    DEF:wattmax=$DB:watt:MAX \
    LINE1:watt#0000FF:$USUARIO \
    'GPRINT:watt:MIN:min\: %9.11f w' \
    'GPRINT:watt:MAX:max\: %9.11f w' \
    'GPRINT:watt:LAST:Ultimo\: %9.11f w' \
    'GPRINT:watt:AVERAGE:Promedio\: %9.11f w' \
    > /dev/null
```

Figura C.1: Contenido del script *graficar.sh*.

El script *graficar.sh* como se puede apreciar utiliza los comandos del shell para generar las graficas el comando RRDTOOL que genera las graficas se llama RRDTOOL *graph* y la forma de uso se puede encontrar en [49].

Al ejecutar el script *graficar.sh* con los parámetros requeridos este genera la grafica de la base RRDTOOL como se muestra a continuación:

```
jonathan@ubuntu:$. /graficar.sh 27f8cbad4babe36ecc67e8179f6fc287.rrd  
27f8cbad4babe36ecc67e8179f6fc287.png h
```

Figura C.2: Ejecución del script *graficar.sh*.

Los parámetros de entrada que utiliza el script *graficar.sh* son:

1. El nombre de la base de datos que para nuestra conveniencia las hemos llamado con el nombre de los ID de los dispositivos: *27f8cbad4babe36ecc67e8179f6fc287.rrd* (rrd es la extensión de las bases de datos RRDTOOL) esto nos ayudara a relacionar y guardar correctamente las mediciones enviadas por los medidores (la creación de esta base de datos se mostrara en la sección siguiente).
2. El nombre de la grafica con la extensión PNG la cual será la extensión de la imagen generada la cual plasmara la grafica con las mediciones, para cada medidor las graficas generadas tendrán el nombre del ID del dispositivo en este caso: *27f8cbad4babe36ecc67e8179f6fc287.png*.
3. El tercer parámetro que requiere el script es el intervalo de tiempo para el cual queremos generar la grafica en este caso como se muestra el parámetro es *h* en este caso se generara una grafica con las mediciones tomadas hace una hora las otras opciones para generar otros intervalos de tiempo son: D, W, e Y para generar graficas de el ultimo día, última semana y ultimo año respectivamente.
4. Luego de ejecutar en la consola se generará la grafica correspondiente al intervalo de tiempo requerido para el medidor con el ID introducido y está se almacenara en el directorio de salida: */cgi-bin / graficas\_Potencia/*.

Vale la pena resaltar que, cómo se describió en la figura 5.1 y 5.8 la capa lógica de negocio es la única que interactúa con la capa de datos en este caso los datos almacenados en las bases RRDTOOL, quien generara las grafica serán las CGI *potencia.php* y *energía.php* las cuales ejecutan los script *graficar.sh* y *graficarE.sh*.

## C.1 Gráficos obtenidos con RRDTOOL

En la Figura C.3 se muestra con mayor detalle la grafica obtenida con la herramienta RRDTOOL esta representa la potencia eléctrica enviada por los medidor en intervalos de minutos (recordar que se envían 5 mediciones cada 5 minutos) en esta grafica generada con RRDTOOL también se anexa en la parte inferior de la imagen: la mínima, máxima, promedio y ultima potencia obtenida en el intervalo de tiempo. Esta grafica generada con RRDTOOL, fue generada para un intervalo de tiempo de una hora, se graficaran las mediciones guardadas en la base de datos RRDTOOL de hace una hora exactamente como se puede apreciar en la Figura C.3.

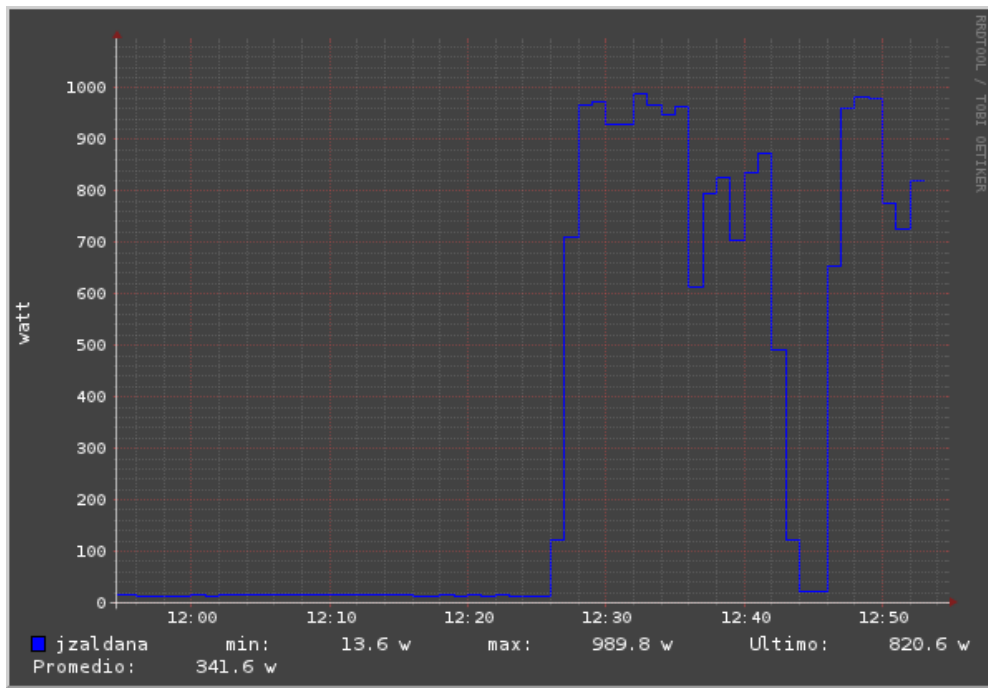


Figura C.3: Graficas Potencia Vs tiempo

En la Figura C.4 se muestra la grafica de Energía Vs tiempo. Esta representa el acumulado de la energía vs el intervalo de tiempo elegido.

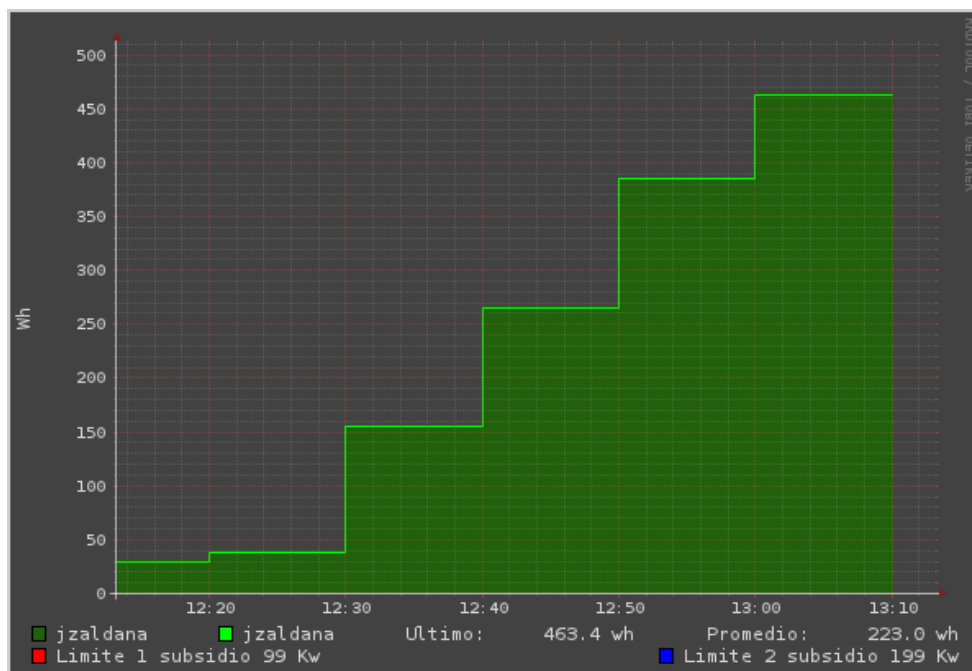


Figura C.4: grafica Energía acumulada Vs tiempo.

## C.2 Bases de datos en RRDTOOL

Se en la sección C que para graficar las mediciones por medio de RRDTOOL este extraía los datos a visualizar en las graficas de las bases de datos RRDTOOL. Para relacionar una base de datos en RRDTOOL con un medidor en específico se utilizo como nombre de la base de datos el ID del medidor y así se podrá asociar un medidor a una base RRDTOOL (27f8cbad4babe36ecc67e8179f6fc287.rrd). De esta manera cada medidor registrado en la tabla *meters* poseerá una base RRDTOOL con el nombre del ID del medidor.

La creación de la bases de datos en RRDTOOL se realiza con la función *create* [50], para poder crear una base de datos más fácil se ha programado un script el cual se ejecuta en la consola de comandos con un solo parámetro, el cual es el nombre de la base de datos RRDTOOL que queremos crear.

A continuación en la Figura C.5 se muestra el contenido del script *create\_rrd* el cual crea una base de datos RRDTOOL (el script *create\_rrd* se puede encontrar en el CD de anexos en: */servidor/RRDTOOL/*):

```

#!/usr/bin/env bash

if [ "$#" != "1" ]; then
    echo "Usage: $0 <database>"
    echo
    echo "Creates an empty rrd database for use with flukso data.
Initializes the start time to 30 days ago."
    echo
    echo "Example: $0 test.rrd"
    exit
fi;

PAT="/cgi-bin/rrd_database/$1"

START=`php -r "echo time() - 60*60*24*30;"`
RRDTOOL create $PAT --start $START -s60
DS:watt:GAUGE:32140800:U:U
RRA:AVERAGE:0.5:1:3200
RRA:AVERAGE:0.5:6:3200
RRA:AVERAGE:0.5:36:3200
RRA:AVERAGE:0.5:144:3200
RRA:AVERAGE:0.5:1008:3200
RRA:AVERAGE:0.5:4320:3200
RRA:AVERAGE:0.5:52560:3200
RRA:AVERAGE:0.5:525600:3200
RRA:MIN:0.5:1:3200
RRA:MIN:0.5:6:3200
RRA:MIN:0.5:36:3200
RRA:MIN:0.5:144:3200
RRA:MIN:0.5:1008:3200
RRA:MIN:0.5:4320:3200
RRA:MIN:0.5:52560:3200
RRA:MIN:0.5:525600:3200
RRA:MAX:0.5:1:3200
RRA:MAX:0.5:6:3200
RRA:MAX:0.5:36:3200
RRA:MAX:0.5:144:3200
RRA:MAX:0.5:1008:3200
RRA:MAX:0.5:4320:3200
RRA:MAX:0.5:52560:3200
RRA:MAX:0.5:525600:3200

```

Figura C.5: Contenido del script Create\_rrd.sh, este crea las bases de datos RRDTOOL.

El script *create\_rrd.sh* crea una base de datos la cual promediara los datos introducidos a esta en minutos. Por tanto si se introducen en un minuto 3 mediciones estos 3 valores se promediaran y se introducirán a la base de datos, todo esto lo genera RRDTOOL automáticamente según los parámetros que se muestran en el script *create\_rrd.sh*. A continuación en la Figura C.6 se muestra la forma de uso del script:

```
jonathan@ubuntu:$. ./create_rrd.sh 27f8cbad4babe36ecc67e8179f6fc287.rrd
```

Figura C.6: Uso del script `create_rrd.sh`.

La ejecución del script crea una base de datos RRDTOOL con el nombre de `27f8cbad4babe36ecc67e8179f6fc287.rrd` y esta se guardara en el directorio `/cgi-bin/rrd_database/`.

Para guardar las mediciones de energía eléctrica se crearan bases en RRDTOOL similar a las mostradas para la potencia eléctrica salvo que en este caso se creara con el script `createE_rrd.sh` el cual se encuentra en el CD de anexo en: `/servidor/RRDTOOL/`. La diferencia de este script es que crea una base de datos RRDTOOL la cual promediara los datos introducidos en intervalos de 10 minutos y que la base creada se guardara en el directorio `/cgi-bin/rrd_database/Energy/`. El nombre de la base de datos para guarda la energía eléctrica enviada por los medidores será el mismo que para la potencia (nombre igual que el ID del medidor).

## Anexo D

### D. Pruebas realizadas al sistema

Una vez contruidos los medidores se procedió a realizar algunas pruebas. Las pruebas a realizar se han dividido en tres las cuales son:

1. Pruebas de comunicación inalámbrica WIFI de los medidores conectados a un AP.
2. Análisis de gráficas de potencia versus tiempo.
3. Comparación de las mediciones tomadas con los medidores contruidos, y el analizador de carga FLUK.

#### D.1 Pruebas de comunicación inalámbrica WIFI

Para realizar la comunicación inalámbrica se utilizo el router *Bullet* [51]. El router fue instalado en el techo de la escuela de ingeniería eléctrica como se muestra en la Figura D.1:



Figura D.1 : (a); (b); (c) y (d) Muestran la instalación del router BULLET 2 en modo AP en el techo de la EIE.

Luego de instalar el router BULLET, se procedió a instalar los medidores. Se instalaron 4 medidores, uno en el tablero secundario del laboratorio de comunicaciones y el resto alrededor de la EIE como se muestra en la Figura D.2:



Figura D.2: Creación de la red de medidores alrededor de la EIE.

El medidor 1 se instaló en el tablero eléctrico del edificio C, que está ubicado a una distancia de 52 m. El medidor 2 se colocó en el patio de la EIE. Éste se instaló sin la pinza de corriente. El medidor 3 se colocó en la caseta de la entrada vehicular de de ingeniería eléctrica ubicado a una distancia de 75 m, este instalado en una acometida.

La red AP creada con el router BULLET se configuró con la *ESSID powermeter* y el password igual a *1111111111*. Por tanto, los 4 medidores tendrán que tener configurado estos parámetros.

En la Figura D.3 se muestra la forma de instalar la pinza de corriente en un tablero eléctrico. Como se puede observar, la instalación del medidor es similar a la instalación del medidor Flukso. El medidor solo utiliza una pinza de corriente de núcleo partido. Si el medidor tomara muestras de voltaje la instalación de este sería más difícil para un usuario final.



Figura D.3 : (a) Pinza de corriente; (b) instalación de la Pinza de corriente; (c) Instalación de la pinza en el tablero eléctrico principal de la EIE; (d) Instalación de la pinza de corriente en una fase del tablero.

En la Figura D.4 se muestra el detalle de la instalación de los medidores en los puntos definidos en la Figura D.2. En la Figura D.4 (b) y (c) muestran el tablero principal del edificio C de la facultad de ingeniería. En este Tablero se instaló el medidor1. En la Figura D.4 (d) se muestra la instalación del medidor en la acometida de la caseta de la entrada vehicular de la EIE. Los medidores envían las mediciones hacia el servidor web el cual está instalado en una PC. Esta PC está conectada con el router BULLET 2 por tanto forma parte de la red al igual que los medidores.

En la Figura D.5 (a) y (b) se muestra las graficas de las mediciones obtenidas con el medidor1 y el medidor2. Se puede concluir que la comunicación con el servidor fue exitosa. El análisis de las graficas obtenidas en el servidor se mostrara en la siguiente sección.

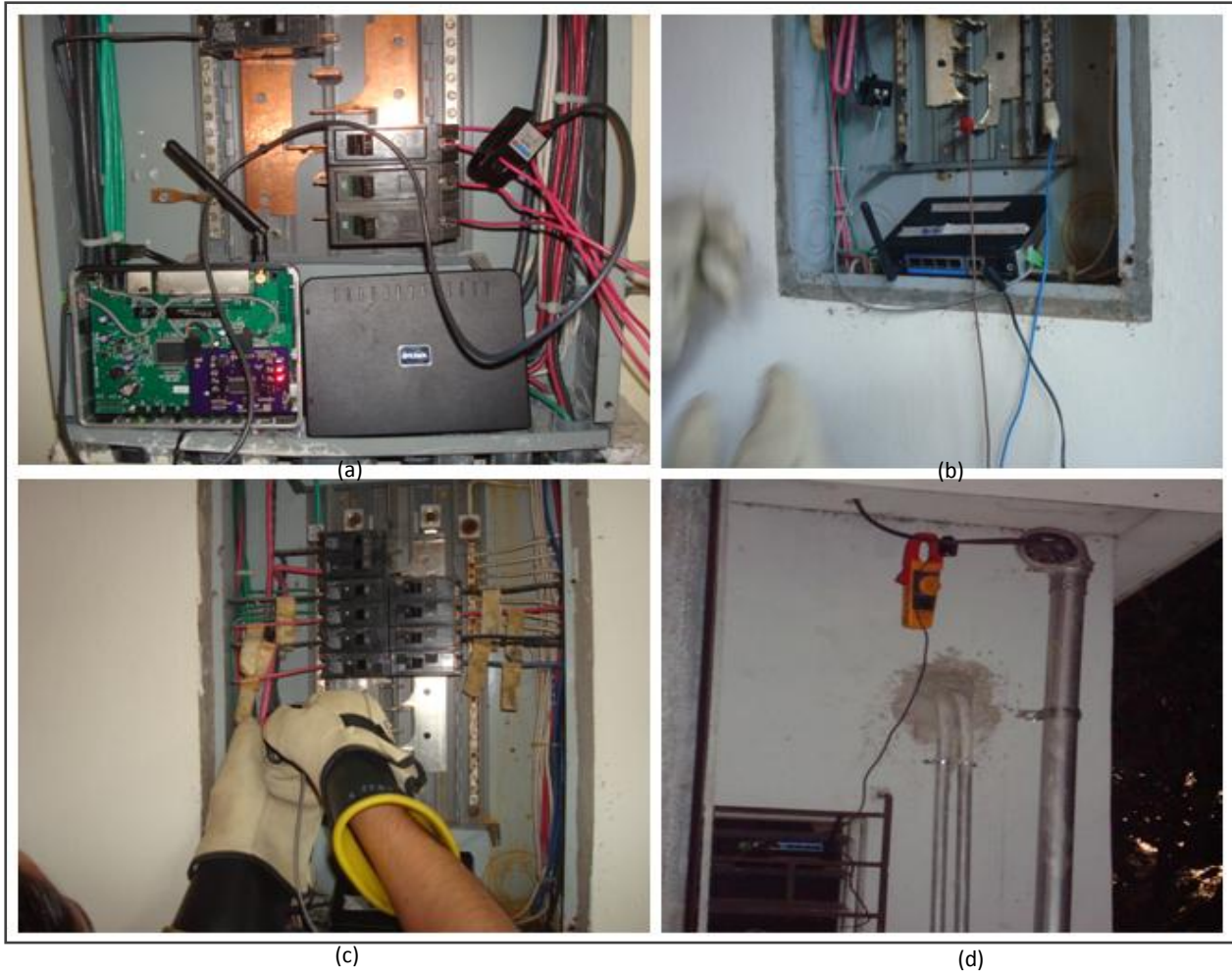


Figura D.4: (a) Instalación del medidor en el tablero del laboratorio de comunicaciones; (b) Instalación del medidor1 en el edificio C; (b) instalación de la pinza en el tablero del edificio C; (d) instalación del medidor3 en la acometida de la caseta de de la entrada vehicular de la EIE.

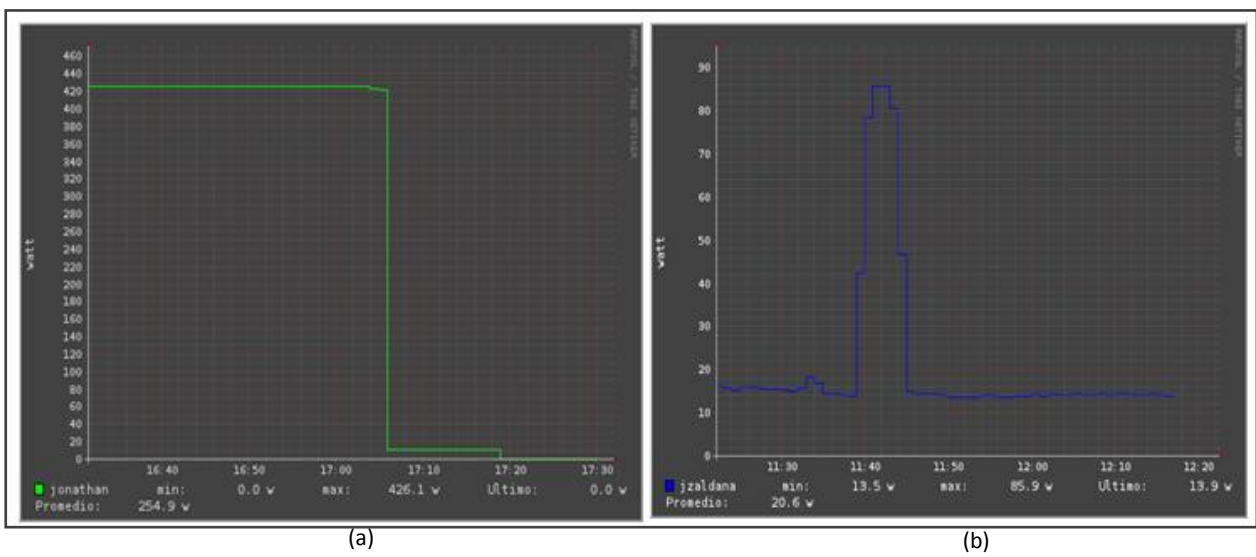


Figura D.5: (a) Graficas obtenidas con las mediciones del medidor1; (b) Graficas obtenidas con el medidor3.

## D.2 Análisis de las graficas de potencia versus tiempo

Una de las finalidades de este proyecto es la monitorización en tiempo real del consumo energético. Esto permitirá tomar decisiones que le lleven a un mejor hábito de consumo energético. Poder Ahorrar energía eléctrica monitoreando el consumo en nuestro hogar, esta es la filosofía de los medidores construidos. El usuario puede observar los picos de potencia y la máxima demanda que está teniendo en su hogar. Con ello puede regular su consumo de electricidad. Se realizó una prueba colocando un medidor en el tablero principal de un apartamento. A continuación se muestra la grafica de potencia eléctrica obtenida con el medidor:

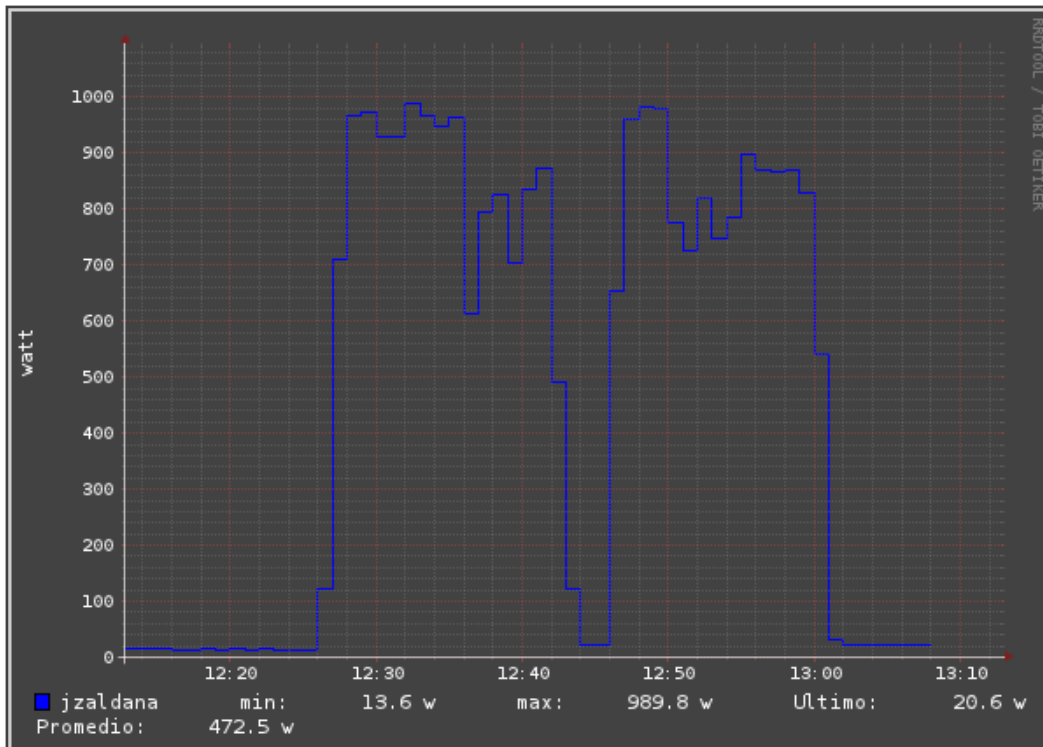


Figura D.6: grafica de potencia versus tiempo obtenidas por el uso de una lavadora.

Para poder visualizar el consumo energético se accede al sitio web. Una vez iniciada la sesión damos clic en el vinculo *hora* el cual se muestra en la Figura D.6. Con esto generamos una gráfica de las mediciones obtenidas una hora atrás de la hora actual. En la Figura D.6 se aprecia la grafica de las mediciones de potencia que está registrando el medidor. Como se muestra en la grafica, en el inicio del pico de voltaje y el final de este. La grafica nos proporciona la máxima potencia demandada por una lavadora utilizada en el momento de las pruebas. Dicha potencia es de 989 W. Se registro un promedio de la ultima hora de 472 W, como se puede apreciar el consumo de esta lavadora es bastante alto. Con la visualización de la graficas en el servidor web podemos regular el ciclo de lavado de la

lavadora para que en este solo genere un enjuague y un centrifugado además de esto podemos ver que la demanda promedio de esta lavadora es bastante alta y se podría tomar la decisión de comprar otra lavadora o mejorar el uso de esta.

### D.3 Comparación de las mediciones tomadas con el Medidor y el analizador de carga FLUK.

Los medidores de potencia que existen el mercado toman muestras de corriente y voltaje de la red eléctrica para obtener la potencia activa. En nuestro caso el medidor solo toma muestras de la magnitud de la corriente de la red eléctrica, fijando en una variable del microcontrolador PIC16F876A el voltaje nominal. El diseño se realizó de esta manera porque no se busco obtener una medición exacta de la potencia activa sino más bien un aproximado de está. Para poder cuantificar el error de las mediciones obtenidas con los medidores construidos, se conecto éste a una carga de prueba junto con el analizador de carga FLUK. Véase la Figura D.7:



Figura D.7: (a) Analizar de carga FLUK; (b) Conexión del analizador a un tablero eléctrico; (c) Tablero eléctrico; (d) Conexión de las pinza del medidor y el analizador FLUK en la misma fase.

Se conectaron los medidores como se muestra en la Figura D.7. Se promediaron muestras de potencia eléctrica en intervalos de 5 minutos durante 30 minutos los resultados de estas pruebas se muestran a continuación en la Tabla D.1:

Tiempo (minutos)	Potencia obtenida con el medidor construido (watts)	Potencia obtenida con el medidor FLUK (watts)	Error %
5	612	601.03	2%
10	360	355.71	1%
15	342	356.00	-4%
20	348	356.00	-2%
25	612	601.04	2%
30	608	600.07	1%

Tabla D.1: Comparación de las mediciones obtenidas con los medidores construidos y el analizador de carga FLUK.

Los resultados de la Tabla D.1 fueron obtenidos variando la carga del tablero donde se conectaron los medidores. Tomando como referencia el medidor FLUK se obtuvo el error en las mediciones del medidor construido, como se muestra en la Tabla D.1.

## Anexo E

### E. Topología de red

Con la topología de red nos referimos a la forma en que se conectarán los medidores construidos con una red WIFI, los routers que proveen internet en un hogar por lo general están configurados para trabajar como AP punto de acceso inalámbrico (WAP o AP por sus siglas en inglés: Wireless Access Point). Los medidores están configurados para que se puedan conectar en una red WIFI AP, esto se logra configurando el nombre de la red WIFI e introduciendo el password de esta. Esto se realiza por medio de la interfaz de configuración WIFI de los medidores a la cual tenemos acceso si conectamos el router a la PC y por medio de un navegador web accedemos a la dirección *192.168.255.1* como se detalla en la sección 2.1.1 del capítulo 2.

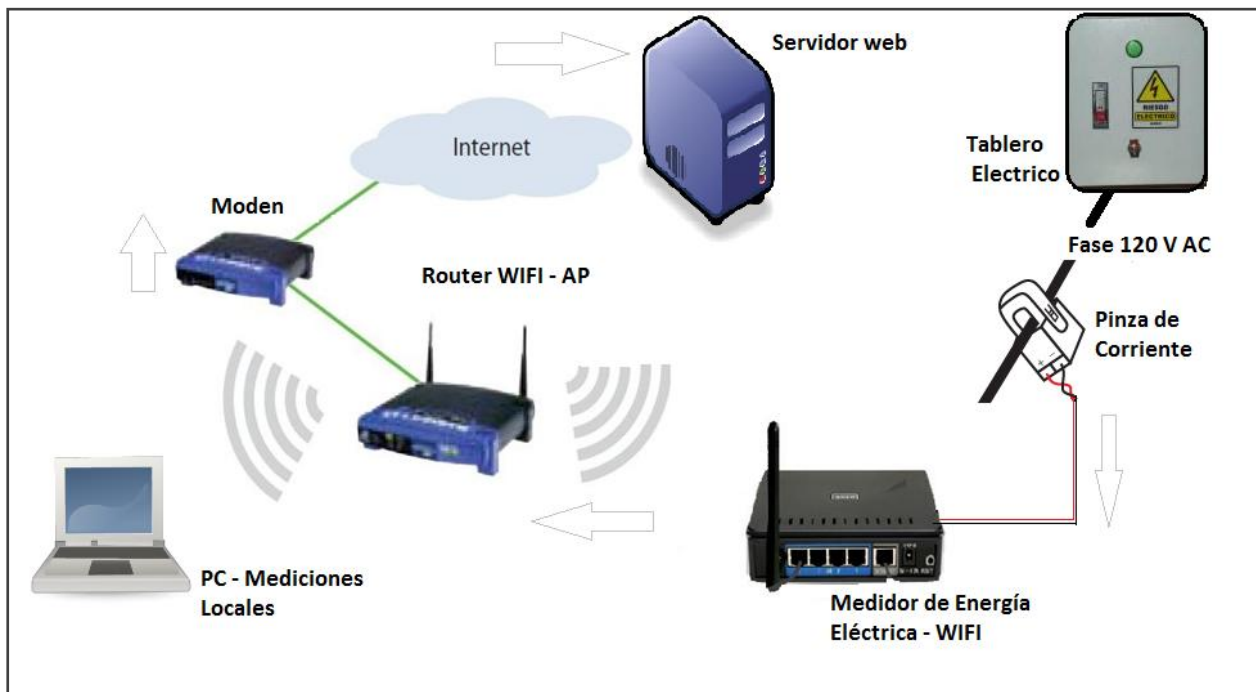


Figura E.1 : topología de red de los medidores.

En la figura 9.1 se muestra un esquema de la topología de red esta muestra la conexión de los medidores a un tablero eléctrico (conexión por medio de la pinza de corriente), luego el medidor se conecta a la red WIFI que sirve el router en modo AP el cual posee acceso a internet (o a la red LAN donde esté conectado el servidor web). Como se puede apreciar la topología red es sencilla y no requiere de un mayor conocimiento de redes inalámbricas para poder conectar inalámbricamente un medidor en una red AP.

Si accedemos por medio de una sesión *ssh* al medidor como se detallo en el capítulo 2 (recordar que todas las configuraciones del medidor Flukso son iguales los medidores construidos ya que se está utilizando el mismo sistema operativo OpenWrt) e ingresamos al directorio */etc/config/* podemos encontrar en este el archivo de configuración de las redes inalámbricas, este archivo llamado *wireless* se muestra a continuación:

```
Wireless:
config wifi-device  wifi0
    option type      atheros
    option channel   auto
    option mode      11bg
    option diversity 0
    option txantenna 1
    option rxantenna 1
    option disabled  0

config wifi-iface
    option device    wifi0
    option network   wan
    option mode      sta
    option nosbeacon 1
    option hidden    0
    option ssid      zwaluw
    option encryption wep
    option key       4ae56b7820f6b6b3cba78da46b
```

Figura E.2: Contenido del archivo de configuración wireless.

Como se puede observar en el contenido del archivo ene este se configura la interfaz *wifi0* del medidor, la opción *mode* como se muestra está configurada en STA este modo los router con OpenWrt actúan como clientes inalámbricos es decir se comportan como la tarjeta de red de una PC el detalle de las demás opciones se pueden encontrar en [46].

## Anexo F

### F. Costo de construcción del medidor

Para realizar el análisis del costo unitario de los medidores construidos sea tomado dos escenarios en función de la cantidad de circuitos impresos a adquirir ya que con el fabricante: <http://www.pcb-mexico.com/> la fabricación 6 PCB cuesta \$ 35.67 USD cada uno, y con este mismo la fabricación de 100 unidades cuesta \$3.93 USD cada uno. En base a la variación del precio de los PCB se realizó el análisis presentado en la Tabla 8.1:

Componentes	Tipo/Valor	PU (\$ USD)	Cantidad (\$ USD)	Total(6 PCB) (\$ USD)	Total (100 PCB) (\$ USD)
Router DIR-300	DIR-300	25.00	1	25.00	25.00
PCB	doble cara, con antisloder	35.67	1	35.67	3.93
Pinzas de Corriente	Flukso J&D	28.00	1	28.00	28.00
IC Referencia de 5V	REF195	6.45	1	6.45	6.45
IC Microcontrolador	PIC16F876A	8.80	1	8.80	8.80
IC Real Time Clock	DS1307	5.32	1	5.32	5.32
Batería	3V	2.76	1	2.76	2.76
Cristal XT	20MHZ	2.18	1	2.18	2.18
Cristal XT	3.768KHZ	0.60	1	0.60	0.60
Capacitor	10uF	0.80	1	0.80	0.80
Capacitor	220uF	0.80	1	0.80	0.80
Capacitor	12pF	0.60	2	1.20	1.20
Capacitor	0.1uF	0.16	4	0.64	0.64
Resistencia	120 $\Omega$	0.07	2	0.14	0.14
Resistencia	330 $\Omega$	0.40	3	1.20	1.20
Resistencia	5.1k $\Omega$	0.05	2	0.10	0.10
Resistencia	100 $\Omega$	0.05	1	0.05	0.05
Resistencia	10k $\Omega$	0.05	1	0.05	0.05
Led	Rojo	0.39	3	1.17	1.17
				120.93	89.19

Tabla F.1: detalle del costo unitario de los medidores diseñados y construidos.

En la Tabla F.1 se muestra el detalle de los componentes que utiliza el *medidor inalámbrico de consumo de energía eléctrica de Bajo Costo*, los componentes electrónicos (exceptuando la pinza de corriente, Los PCB y el router DIR-300) fueron cotizados y comprados en la

tienda virtual: <http://www.digikey.com/>, y los precios reflejados en tabla 8.1 incluyen costo de envío y tramites de aduana.

Como se muestra en la Tabla 8.1 el coste de fabricación del medidor si se adquieren 6 PCB es de \$120.93 USD por cada medidor, este es el costo de los medidores construidos en el presente trabajo ya que solo se adquirió 6 PCB, pero como se muestra en la Tabla 8.1 por economía de escala si se hubiesen adquirido 100 PCB el precio de los medidores seria de \$89.19 USD.

Los costos unitarios mostrados en la Tabla 8.1 para los medidores no incluyen, el costo de fabricación y el costo de propiedad intelectual.

## Referencias

- [1] Analog Devices, “Hoja Técnica del ADE7757”. [En línea]. Disponible:  
[http://www.analog.com/static/imported-files/data\\_sheets/ADE7757.pdf](http://www.analog.com/static/imported-files/data_sheets/ADE7757.pdf)
- [2] Microchip, “Hoja técnica Pic16f876A” [En línea]. Disponible:  
<http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf>
- [3] Flukso, “Flukso web principal”. [En línea]. Disponible:  
<http://www.flukso.net>.
- [4] Abocon, “Información técnica del WAP-2102”. [En línea]. Disponible:  
[http://www.abocom.com.tw/download\\_show.aspx?ptype=66395a7ea68b3cb8a3e548690eb0df8a&id=44D4E3D6036181B74C2BE1576589A958](http://www.abocom.com.tw/download_show.aspx?ptype=66395a7ea68b3cb8a3e548690eb0df8a&id=44D4E3D6036181B74C2BE1576589A958).
- [5] RRDTOOL, “About RRDtool”. [En Línea]. Disponible:  
<http://www.mrtg.org/rrdtool>.
- [6] Atmel, “Hoja técnica del ATMEGA168”. [En línea]. Disponible:  
[http://www.atmel.com/dyn/resources/prod\\_documents/doc2545.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf)
- [7] Abocon, “Información técnica del WAP-2102”. [En línea]. Disponible:  
[http://www.abocom.com.tw/download\\_show.aspx?ptype=66395a7ea68b3cb8a3e548690eb0df8a&id=44D4E3D6036181B74C2BE1576589A958](http://www.abocom.com.tw/download_show.aspx?ptype=66395a7ea68b3cb8a3e548690eb0df8a&id=44D4E3D6036181B74C2BE1576589A958)
- [8] OpenWrt Wireless Freedom, “OpenWrt web principal”. [En línea]. Disponible:  
<https://openwrt.org/>
- [9] Digi Connect, “Pagina principal”. [En línea]. Disponible:  
<http://www.digi.com/es/products/embeddedolutions/digiconnectwiem#overview>
- [10] FON, “FON web principal”. [En línea]. Disponible:  
<http://www.fon.com/es>
- [11] VillageTelco, “Mesh Potato web principal”. [En línea]. Disponible:  
<http://villagetelco.org/mesh-potato/>
- [12] LUA. “The Programming Language Lua”. [En línea]. Disponible:  
<http://www.lua.org/>
- [13] Qualcomm Atheros, “Technology WLAN” [En línea]. Disponible:  
<http://www.qca.qualcomm.com/technology/technology.php?nav1=47>

- [14] OpenWrt Wireless Freedom, "Configuración del servidor web UHTTPD". [En línea]. Disponible: <http://wiki.openwrt.org/doc/uci/uhttpd>
- [15] OpenWrt Wireless Freedom, "Configuración del cliente NTP". [En línea]. Disponible: <http://wiki.openwrt.org/doc/uci/ntpclient>
- [16] FC. Casellas, GV. Velasco, FG. Guinjoan, RP. Piqué, "El Concepto de Smart Metering en el nuevo escenario de distribución eléctrica. Universidad Politécnica de Catalunya. [En línea] Disponible: <http://upcommons.upc.edu/e-prints/bitstream/2117/9066/1/5025.pdf>
- [17] OpenWrt Wireless Freedom, "El sistema UCI". [En línea]. Disponible: <http://wiki.openwrt.org/doc/uci>
- [18] vanemery "Puerto Serie" ". [En línea]. Disponible: <http://www.vanemery.com/Linux/Serial/serial-console.html>
- [19] J.A Zaldaña, "Construcción de un medidor de energía eléctrica inalámbrico WIFI I", Proyecto terminal, Universidad de El Salvador, San Salvador, El Salvador, 2010.
- [20] J.A Zaldaña, "Construcción de un medidor de energía eléctrica inalámbrico WIFI II", Proyecto terminal, Universidad de El Salvador, San Salvador, El Salvador, 2010.
- [21] PHP: Hypertext Preprocessor, "PHP web principal". [En línea]. Disponible: <http://www.php.net/>
- [22] The Perl Programming Language, "Perl web principal". [En línea]. Disponible: <http://www.perl.org/>
- [23] MySQL, "MySQL web principal". [En línea]. Disponible: <http://www.mysql.com/>
- [24] J.M.A. Usategui, S R. Yesa, y I.A. Martínez, "El Conversor A/D" en Microcontroladores PIC diseño práctico de aplicaciones segunda parte.3ra ed. MC-Gran-Hill, 2000, pp. 131.
- [25] J.M.A. Usategui, S R. Yesa, y I.A. Martínez, "MÓDULO MSSP TRABAJANDO EN MODO I2C" en Microcontroladores PIC diseño práctico de aplicaciones segunda parte.3ra ed. MC-Gran-Hill, 2000, pp. 159.
- [26] Microchip, "Microchip provides Microcontroller and Analog Semiconductors". [En línea]. Disponible: <http://www.microchip.com>
- [27] Maxim Innovation Delivered, "Maxim web principal". [En línea]. Disponible: <http://www.maxim-ic.com/>

- [28] Maxim, "Hoja técnica DS1307". [En línea]. Disponible:  
<http://www.datasheetcatalog.org/datasheet/maxim/DS1307.pdf>
- [29] Cursomicros.com, "Registros del DS1307". [En línea]. Disponible:  
[http://www.cursomicros.com/control\\_i2c/c02\\_03\\_ds1307\\_regs.html](http://www.cursomicros.com/control_i2c/c02_03_ds1307_regs.html)
- [30] Analog Devices Inc, "Hoja técnica REF-195". [En línea]. Disponible:  
[http://www.analog.com/static/imported-files/Data\\_Sheets/REF19XSERIES.pdf](http://www.analog.com/static/imported-files/Data_Sheets/REF19XSERIES.pdf)
- [31] Analog Devices Inc, "Semiconductors and Signal Processing ICs". [En línea]. Disponible:  
<http://www.analog.com/en/index.html>.
- [32] J.M.A. Usategui, S R. Yesa, y I.A. Martínez, "MÓDULO DE CAPTURA COMPARACIÓN Y MODULACIÓN DE ANCHURA DE PULSOS" en Microcontroladores PIC diseño práctico de aplicaciones segunda parte.3ra ed. MC-Gran-Hill, 2000, pp. 105.
- [33] J.M.A. Usategui, S R. Yesa, y I.A. Martínez, "GENERADOR DE BAUDIOS" en Microcontroladores PIC diseño práctico de aplicaciones segunda parte.3ra ed. MC-Gran-Hill, 2000, pp. 194.
- [34] CCS, "CCS web principal". [En línea]. Disponible:  
<http://www.ccsinfo.com/>.
- [35] CadSoft, "Eagle PCB Software de diseño". [En línea]. Disponible:  
<http://www.cadsoftusa.com/eagle-pcb-design-software/?language=en>
- [36] Servicios Tecnológicos Tres Ríos, "PCB Fabricación de tarjetas de circuito impreso, Servicios Tecnológicos Tres Ríos. México D.F.". [En línea]. Disponible: [www.pcb-mexico.com](http://www.pcb-mexico.com)
- [37] XML-RPC," XML-RPC web principal". [En línea]. Disponible: <http://www.xmlrpc.com>.
- [38] XML-RPC,"XML-RPC Especificaciones". [En línea]. Disponible:  
<http://xmlrpc.scripting.com/spec.html>
- [39]LUA Socket, "LUA Socket información web". [En línea]. Disponible:  
<http://w3.impa.br/~diego/software/luasocket/>
- [40] OpenWrt Wireless Freedom, "BackFire paquetes". [En línea]. Disponible:  
<http://backfire.openwrt.org/10.03/atheros/packages/>.
- [41]LUA Socket, "LUA Socket información web". [En línea]. Disponible:  
<http://w3.impa.br/~diego/software/luasocket/>

[42] Apache Software Foundation, “Apache Software web principal”. [En línea]. Disponible: <http://www.apache.org/>

[43] XML-RPC HOWTO, “XML-RPC con Perl”. [En línea]. Disponible: <http://tldp.org/HOWTO/XML-RPC-HOWTO/xmlrpc-howto-perl.html>

[44] RRDTOOL, “Sinapsis RRDTOOL”. [En línea]. Disponible: <http://oss.oetiker.ch/RRDTOOL/doc/RRDTOOL.en.html>

[45] Dragino, “Dragino web principal”. [En línea]. Disponible: <http://www.dragino.com/>

[46] OpenWrt Wireless Freedom, “Wireless configuration”. [En línea]. Disponible: <http://wiki.openwrt.org/doc/uci/wireless>.

[47] Flukso, “Archivos binarios de OpenWrt”. [En línea]. Disponible: <http://www.flukso.net/files/openwrt/FLUKSO-1-0/FLUKSO-1-0-bin.tar.gz>.

[48] Village telco, “Instalación de OpenWrt al DIR-300” “. [En línea]. Disponible: [Fhttp://wiki.villagetelco.org/index.php?title=Interesting\\_Atheros\\_SoC\\_based\\_low-cost\\_device\\_with\\_4\\_port\\_switch](http://wiki.villagetelco.org/index.php?title=Interesting_Atheros_SoC_based_low-cost_device_with_4_port_switch).

[49] RRDTOOL, “Uso de rrdgraph”. [En línea]. Disponible: <http://oss.oetiker.ch/rrdtool/doc/rrdgraph.en.html>

[50] RRDTOOL, “uso de rrdcreate”. [En línea]. Disponible: <http://oss.oetiker.ch/rrdtool/doc/rrdcreate.en.html>.

[51] “Hoja técnica del bullet2”. [En línea]. Disponible: [http://ubnt.com/downloads/b2\\_datasheet.pdf](http://ubnt.com/downloads/b2_datasheet.pdf)