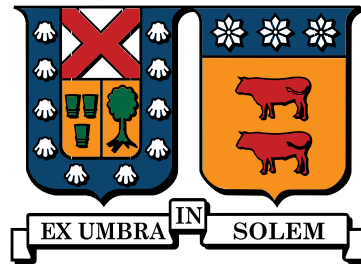


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



SISTEMA CENTRALIZADO DE
RECOLECCIÓN AUTOMÁTICA DE
CONSUMO DE SERVICIOS BÁSICOS
CON ACCESO VÍA WEB

SEBASTIÁN ANDRÉS APABLAZA BENAVIDES

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELECTRÓNICO MENCIÓN COMPUTADORES

Profesor guía:

Ph.D. Agustín J. González V.

Profesor Correferente:

Ph.D. Marcos D. Zuñiga B.

Mayo 2016

Sistema centralizado de recolección automática de consumo de servicios básicos con acceso vía WEB

Memoria para optar al título de Ingeniero Civil Electrónico, mención

Computadores

Sebastián Andrés Apablaza Benavides

Profesor Guía: Ph.D. Agustín J. González V.

Mayo 2016

Resumen

El objetivo principal de este trabajo es proponer una alternativa para la implementación de un sistema de medición inteligente, por medio de un prototipo vertical que se adapte a la realidad chilena. El trabajo se centra en la distribución eléctrica, ya que ésta permite incorporar los requerimientos de la nueva ley 20.571, la cual regula la generación domiciliaria de electricidad mediante el uso de energías renovables no convencionales. Este sistema proporciona información a las empresas distribuidoras sobre flujos de electricidad con gran granularidad en tiempo real y permite a los usuarios revisar sus consumos a tiempo, mejorando la transparencia en la facturación.

El análisis efectuado en este trabajo pretende sentar las bases para una alternativa de implementación de un sistema de medición de consumos eléctricos capaz de soportar las mediciones de consumo de los distintos servicios básicos simultáneamente, realizando pocas modificaciones al prototipo actual, así como implementar otras funcionalidades tales como tarifas diferenciadas con el fin de incentivar consumos en horarios de baja demanda.

El sistema desarrollado se compone por un servidor y un ordenador de placa única conectados a través de una red Ethernet, los cuales transmiten y almacenan los datos de consumo provenientes de un medidor de electricidad Itron de comunicación unidireccional.

Palabras claves: medición inteligente, consumo domiciliario, redes eléctricas

Centralized system of automatic collection of consumption of basic services accessible via WEB

Thesis to earn the degree of Civil Engineer in Electronics, in the master of
Computer Sciences.

Sebastián Andrés Apablaza Benavides

Guide Professor: Ph.D. Agustín J. González V.

May 2016

Abstract

The main objective of this paper is to propose an alternative for the application of a smart metering system by means of a vertical prototype, adaptable to the Chilean's situation. The project is focused on electric distribution, since it allows the incorporation of the new law requirements, which regulate the generation of household electricity by the use of non-conventional renewable energies. Moreover, The system provides information to the electric utilities regarding electricity flows with considerable granularity in real time. In addition, it allows users to check their power consumption forecasts on time, consequently the veracity of the billing improves.

The analysis conducted in this paper intends to establish the basis to provide a new alternative for the application of an electric consumption metering system, able to withstand the consumption measurements of several basic services simultaneously by making some modifications to the current prototype, as well as implementing other features such as differentiated fees with the objective of boosting user's electricity consumption in off-peak hours.

The system is based on a server and a single-board computer, connected by Ethernet. They transmit and store the consumption data that comes from an Itron unidirectional electricity meter.

Keywords: smart metering, Household consumption, Electric Grid

Índice general

Resumen	I
Abstract	II
Índice de figuras	V
Índice de tablas	VII
1. Introducción	1
1.1. Descripción del problema	2
1.1.1. Problemas de las empresas distribuidoras	2
1.1.2. Problemas de los clientes	3
1.2. Objetivos	6
2. Estado del Arte	7
2.1. Tecnologías de comunicación	8
2.1.1. Zigbee	8
2.1.2. Redes Celulares	9
2.1.3. ERT	9
2.1.4. PLC	10
2.2. Tipos de arquitecturas del Sistema	11
3. Diseño del sistema de adquisición automático de mediciones	15
3.1. Arquitectura	15
3.2. Descripción de los módulos	16
3.2.1. Medición	17
3.2.2. Adquisición	17
3.2.3. Transmisión	18
3.2.4. Almacenamiento	18

3.2.5. Presentación	19
4. Implementación del sistema de adquisición automático de mediciones	21
4.1. Módulo Medición	21
4.2. Módulo Adquisición	23
4.3. Módulo Transmisión	27
4.4. Módulo Almacenamiento	28
4.5. Módulo Presentación	34
5. Verificación y Evaluación	37
5.1. Puesta en marcha del sistema	37
5.2. Pruebas de estabilidad y desempeño del sistema	41
6. Conclusiones y trabajo futuro	47
Bibliografía	51
A. Códigos fuente	53
B. Medidores compatibles	67
C. Licencia Apache 2.0	69

Índice de figuras

1.1. Distribución por tipo de consumo	4
1.2. Opinión de Confiabilidad y precisión del medidor	4
1.3. Distribución de reclamos por tipo	5
1.4. Tipo de resolución	5
1.5. Distribución de responsabilidades	6
2.1. Topología de malla	11
2.2. Topología jerárquica	12
2.3. Conexión Directa	13
3.1. Modelo de capas SWAN	16
3.2. Esquema del sistema	16
4.1. Medidor inteligente Itron	21
4.2. Componentes del concentrador	24
4.3. Diagrama de flujo de la obtención de la medición	26
4.4. Diagrama de flujo concentrador	27
4.5. Esquema de recepción de datos	29
4.6. Diagrama relacional de la base de datos	32
4.7. Diagrama de flujo para aviso de adulteración	34
4.8. Diagrama de flujo Módulo Presentación	35
4.9. Esquema de archivos PHP	35
5.1. Página registro de usuarios	38
5.2. Inicio servicio RTL TCP	38
5.3. Inicio decodificador de mensaje	39
5.4. Revisión de los mensajes generados	39
5.5. Servidor receptor de mensajes	39
5.6. Envío de mensajes al servidor	39

5.7. Inserción de mediciones a la BD	40
5.8. Correo de aviso de adulteración	40
5.9. Despliegue de consumos	41
5.10. Velocidad de transmisión	42
5.11. Tiempo de ejecución	43
5.12. Estadísticas de sistema del servidor	44

Índice de tablas

2.1. Características de tecnologías celulares	9
4.1. Características de comunicación del medidor	22
4.2. Descripción de un paquete SCM	23
4.3. Características Raspberry Pi modelo B	24
4.4. Particiones de la tarjeta SD	25

1. Introducción

El tema de investigación nace a partir del desafío propuesto por la empresa Dimension Data, el cual se titula «¿Cómo implementar proyectos de medición inteligente de luz, agua o gas en Chile bajo un modelo económicamente sustentable?», en el marco del programa piloto de Memorias Multidisciplinarias.

Este estudio se desarrolla para el caso particular del consumo eléctrico por dos razones principales: En primer lugar cuando los países migran hacia una red de consumo inteligente de servicios básicos, el primer servicio actualizado es principalmente la distribución eléctrica, lo que conlleva a una tecnología más madura y por consiguiente más validada. En segundo lugar la entrada en vigencia de la nueva Ley 20.571, conocida como Ley de Generación Distribuida[7], regula el pago a las inyecciones al sistema de distribución eléctrica realizada por medios pequeños de generación renovables. Esto ofrece un incentivo extra a favor de la actualización de la red de medición eléctrica ya que los equipos actuales no permiten la medición de flujo de energía bidireccional, esto debido a que la ley 20.571 exige a las empresas contabilizar la inyección y en la actualidad esto se realiza instalando dos medidores distintos, uno para el consumo y el segundo para la inyección.

Como punto relevante cabe mencionar la tendencia mundial de transición hacia un consumo responsable de energía, este tipo de tecnologías permite la incorporación de políticas que incentiven conductas de consumo acorde a la realidad de nuestro país haciendo al cliente un participante activo en el consumo sustentable de energía. Esta tecnología permite a empoderar a los clientes al entregar herramientas para mantenerlo informado de su consumo en tiempo real.

Actualmente en el mundo se encuentran distintas etapas de implementación de medición inteligente, la instalación de mayor envergadura se encuentra en Italia donde toda la base de clientes de la principal empresa distribuidora de electricidad fueron actualizados desde los medidores convencionales a medidores inteligentes,

mientras que en otros países se encuentran instalaciones en sus principales estados o provincias como es el caso de Australia y Canadá entre otros. En el caso particular de Chile se han implementado diversos pilotos con distintos alcances y clientes objetivos, de los que se destacan la implementación de medición inteligente de consumo eléctrico en el sector de Ciudad Empresarial, Región Metropolitana, por ‘Smartcity Santiago’[4], SOCOEPA con 590 puntos residenciales de zonas remotas y 50 puntos comerciales de suministro trifásico[3] y por último la implementación de 540 medidores de agua para grandes clientes en zonas de difícil acceso por Aguas Andinas[5].

Finalmente un aspecto fundamental a considerar es la recepción que la incorporación de este tipo de tecnología por parte de los usuarios finales. Alrededor del mundo existen asociaciones ‘anti-smartmeters’ las cuales afirman que se «...Transmite información detallada sobre sus patrones de consumo, emiten radiación electromagnética que puede producir cáncer y dañar el ADN...»[11]

1.1. Descripción del problema

Las principales problemáticas pueden ser separadas en dos grupos: El primero se puede definir como los que enfrentan las empresas distribuidoras durante el proceso de distribución y facturación. Mientras que el segundo grupo corresponde a los problemas que afectan a los clientes o usuarios de consumo eléctrico domiciliario.

1.1.1. Problemas de las empresas distribuidoras

Los principales problemas de las empresas distribuidoras se dan por las llamadas «pérdidas no técnicas» que se refieren al mal uso del servicio, así como también a errores administrativos propios del negocio, las cuales corresponden al 1,9 % del total de la compra de energía a las generadoras, que suma alrededor de \$15.200 millones al año[6]. Particularmente los dos tipos más relevantes para este análisis son las «Pérdidas no técnicas administrativas» y las «Pérdidas no técnicas fraudulentas».

Pérdidas no técnicas administrativas: Se producen en el proceso de instalación

de medidores en nuevas localidades sin informar a tiempo o correctamente al área comercial de la empresa para su ingreso al sistema comercial. Esto implica que esos consumos no sean facturados y que este consumo no se tenga que pagar.

Pérdidas no técnicas fraudulentas: Estos son principalmente dos, externos al medidor; en donde los usuarios se cuelgan a la red eléctrica haciendo puentes desde el tendido eléctrico. Esto puede ser en paralelo al medidor eléctrico para disminuir su consumo o directamente desconectar el medidor y realizar la conexión al poste, esto produce lecturas sin consumo. Otra pérdida se presenta con la intervención al medidor los que se refieren a cualquier manipulación por parte del cliente con el objetivo de disminuir las mediciones de su consumo. Estas manipulaciones pueden ser sellos violados, discos intervenidos, bobinas abiertas, etc.

Estos problemas descritos anteriormente con la tecnología actual requieren que las empresas incurran en grandes esfuerzos y dinero tanto en la prevención como en fiscalización.

1.1.2. Problemas de los clientes

En Chile actualmente existen 34 empresas de distribución eléctrica, donde el consumo preponderante proviene de los clientes residenciales. Para un mejor entendimiento del rol que cumplen estos clientes es conveniente desglosar el consumo en los distintos tipos de clientes como muestra la Figura 1.1[13]. De donde es evidente que el consumo domiciliario ocupa una gran cuota de mercado, por lo que la opinión de los clientes pasa a ser una medida de calidad importante utilizada por la Superintendencia de electricidad y combustibles al momento de generar el «Ranking de Calidad de Servicio de las Empresas de Distribución de Electricidad».

Para este Ranking el 50 % de la evaluación proviene de la opinión de los clientes, a través de encuestas y reclamos registrados. En particular para este trabajo resultan importantes dos ítemes de dicho documento, una de las preguntas correspondiente

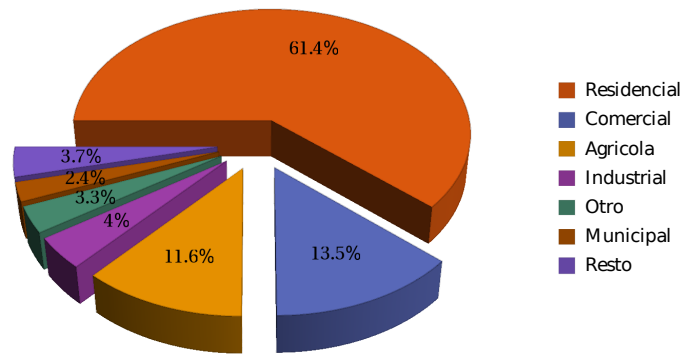


Figura 1.1: Distribución por tipo de consumo

a la encuesta: «En general ¿Qué nota le pone a la precisión y confiabilidad de su medidor?», cuya respuesta se muestra en la Figura 1.2

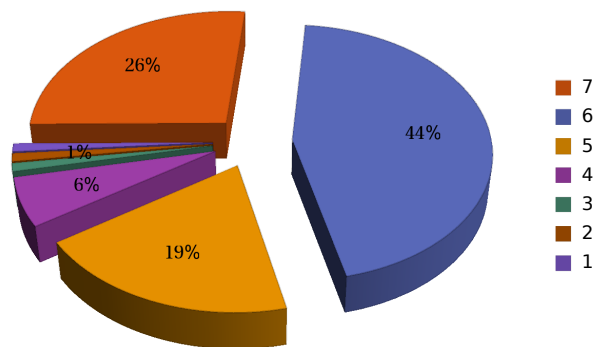


Figura 1.2: Opinión de Confiabilidad y precisión del medidor

La distribución de reclamos por tipo en la Figura 1.3 donde la mayor cantidad de reclamos por parte de los usuarios tiene que ver específicamente con la Medición y Facturación y Facturación. Si la opinión de Confiabilidad y precisión del medidor se contrasta con la estadística de «tipo de resolución» de la Figura 1.4 en donde el 57 % de los reclamos de los clientes se resuelven desfavorablemente para el usuario, esto puede indicar una desinformación o falta de claridad sobre su consumo.

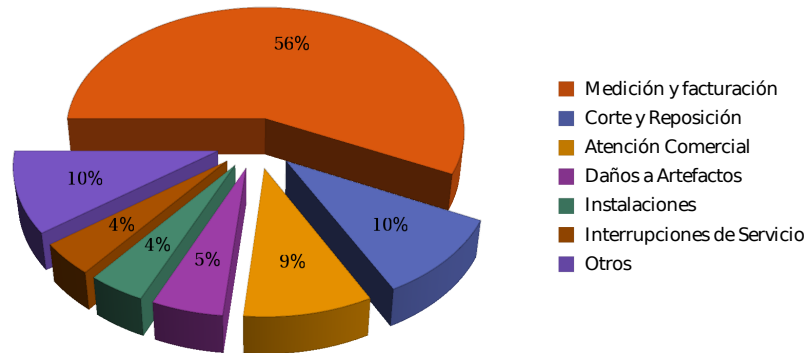


Figura 1.3: Distribución de reclamos por tipo

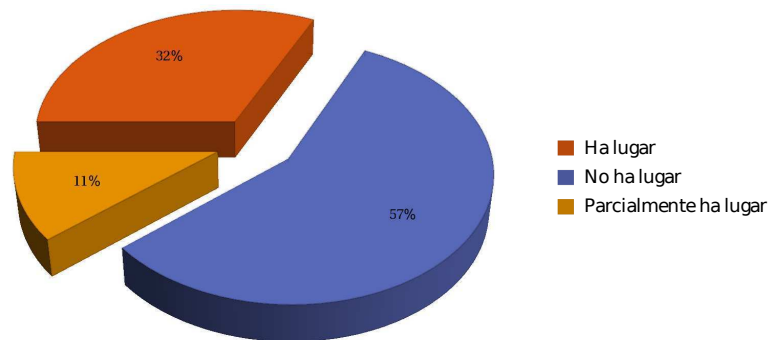


Figura 1.4: Tipo de resolución

1.2. Objetivos

El objetivo general del presente trabajo de título es desarrollar un sistema capaz de recolectar, transmitir y almacenar los datos de los usuarios para su posterior análisis y despliegue a través de la WEB para el usuario final así como para la empresa realice la facturación a través de los datos almacenados en el servidor. El desarrollo de este sistema se propone en forma de un prototipo vertical para demostrar la viabilidad de una implementación a gran escala de esta tecnología. Se desarrollan todas las partes del sistema sin incluir todas las funcionalidades que puede llegar a tener el sistema en etapa de producción.

El sistema debe realizar las mediciones de consumo desde las casas de los usuarios, transmitirlos a un servidor o almacenarlos localmente en el caso de no tener conexión entre el concentrador con el servidor y desplegarlos a través de una plataforma WEB para que los usuarios puedan observar su consumo. Además debe avisar oportunamente ante la detección de cualquier adulteración realizada a los medidores.

El presente trabajo se realiza en conjunto con Andrés Silva, estudiante de Ingeniería Civil Telemática[12]. La distribución de responsabilidades se resume en la Figura 1.5, donde el nombre de cada integrante se encuentra bajo el tema de su responsabilidad exclusiva, en tanto los temas desarrollados en conjunto poseen el rótulo «Compartida».

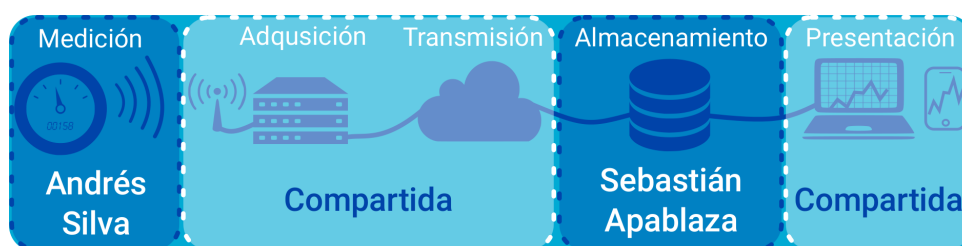


Figura 1.5: Distribución de responsabilidades

2. Estado del Arte

Las tecnologías de medición se dividen en medición automática AMR (Automatic Meter Reading) e Infraestructura de medición avanzada AMI (Advanced Metering Infrastructure). La medición automática se define como la tecnología que recolecta los consumos de energía y transfiere esa información del medidor de la residencia a la empresa distribuidora a través de una comunicación unidireccional. La lectura de estos consumos se realizan de dos maneras distintas, a través de un operario que transita con un dispositivo que recolecta la información de manera inalámbrica automáticamente desde los medidores que se encuentran dentro del rango de transmisión de estos. El recorrido se puede realizar a pie o a través de un vehículo. El otro método de lectura es a través de una estación fija que se instala dentro del rango de transmisión de los medidores que retransmite la información hasta la empresa distribuidora. Mientras que la medición inteligente es una modernización de la lectura automática dado que no solo miden cuánta electricidad es utilizada sino que también en qué horarios del día y otros datos extra como alertas de manipulación. Además están diseñados para transmitir las tarifas e información desde la compañía distribuidora hacia el cliente.

El componente que comparten todos los tipos de medidores es la conectividad entre los distintos elementos del sistema, los medios físicos por los que transmite la información y el protocolo utilizado. Precisamente, es en este último componente en que se basa esta investigación debido a que la comunicación es independiente al tipo de servicio que se quiera medir y puede ser implementado indistintamente para cualquiera de los tres tipos de consumos básicos por lo que resulta beneficioso para cualquiera de las empresas de servicios básicos. A continuación se detallan los tipos de conexión, tecnologías y arquitecturas utilizadas en los sistemas AMR y AMI.

2.1. Tecnologías de comunicación

Existen distintas tecnologías disponibles para la comunicación las que se dividen en dos principales medios de transmisión, cableado e inalámbrico. En ciertos casos, la tecnología inalámbrica posee ciertas ventajas por sobre las tecnologías cableadas, como lo son el bajo costo en infraestructura y la conexión de áreas con difícil acceso. Por otro lado, las tecnologías cableadas no tienen problemas de interferencia y son generalmente más confiables. Los estándares de comunicación más utilizados para la comunicación desde el medidor se enumeran y detallan a continuación.

2.1.1. Zigbee

Es un conjunto de protocolos de alto nivel de comunicación inalámbrica de bajo consumo, basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal 'WPAN'. Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías. En la medición inteligente se utiliza este tipo de conexión para configuraciones descentralizadas donde cada medidor se comunica con otro que tenga a su alcance. Esto permite redundancia ya que la red se organiza automáticamente en rutas de menos costo que llevan a un medidor con conexión a Internet desde donde se transmiten los datos de consumo como se explica más adelante en el esquema de la Figura 2.1. Lo cual resulta en ventajas ya que no requiere infraestructura extra para enviar los datos así como una fácil escalabilidad de la implementación. La principal desventaja de esta tecnología es que utiliza el espectro libre de 2.4 GHz por lo que debe competir por el espectro con otras tecnologías como lo son WiFi, Bluetooth y el ruido generado por hornos microonda con fugas por ejemplo.

2.1.2. Redes Celulares

Estas redes poseen beneficios para la comunicación entre medidores y el servidor donde se almacenan los datos. La infraestructura ya se encuentra instalada, es robusta y segura lo que permite ahorrar costos de instalación así como tiempo de implementación. Particularmente para el servicio de medición de consumos no son necesarias altas tasas de transferencia por lo que la capacidad celular instalada actualmente permite trabajar holgadamente en condiciones normales. Se trata además de un sistema ampliamente difundido, con cobertura en todas las ciudades del país enfocada a una gran cantidad de clientes con la capacidad de soportar protocolos de transferencia de datos. Sin embargo este sistema presenta problemas de congestión y conectividad en situaciones de alta demanda. Al ser un servicio administrado por una empresa externa no se tiene control sobre la calidad, disponibilidad o costos asociados al uso de la conexión. Las tecnologías celulares útiles son aquellas que permiten el tráfico de datos, esto es desde la tecnología 2G en adelante, en particular GPRS, EDGE y 3G. En la Tabla 2.1 se presentan las principales características de las tecnologías antes mencionadas.

Tabla 2.1: Características de tecnologías celulares
fuente: European Telecommunications Standards Institute

Tecnología	GPRS	EDGE	IMT-2000 (3G)
Frecuencia de operación	0.8/ 1.7/ 1.8 GHz	0.8/ 1.7/ 1.8 GHz	1.7/ 2.1 GHz
Velocidad de Transmisión	Hasta 115 Kbps	Hasta 384 Kbps	Hasta 7.2 Mbps
Rango de Cobertura	Hasta 5 Km	Hasta 5 Km	Hasta 5 Km

2.1.3. ERT

El protocolo de radio ‘Encoder Receiver Transmitter’ (ERT)[1] desarrollado por Itron para la lectura automática de medidores. Utiliza modulación OOK transmitida en la banda de los 900-920 MHz mediante codificación Manchester,

además de utilizar salto en frecuencia para disminuir la probabilidad de interferencia con otros dispositivos. Una de las ventajas de este protocolo es su adopción por parte de otras empresas, en parte debido a que la patente pasó a dominio público al haber transcurrido más de 20 años desde su publicación. Además está diseñado para la recolección automática de la información de consumo por medio de un receptor que se instala dentro de un auto el cual circula por la calle recopilando los datos de consumo mientras transita. Las ventaja de este método es que requiere poca infraestructura, solo un receptor que se encuentre al alcance de los medidores. La principal desventaja radica en que el proceso de recolección de datos de consumo es esporádico y no almacena todos los datos que emite el medidor.

2.1.4. PLC

‘Power Line Communication’ (PLC) es una tecnología que transmite la información por el tendido eléctrico de baja tensión, con la ventaja de la cobertura ya existente de redes eléctricas sin necesidad de desplegar nuevos cables dedicados. La información se transmite por todos los cables similar al ‘Broadcast’ de una red IP y llegan a un equipo llamado concentrador que usualmente se encuentra junto a los transformadores de baja tensión, los que reciben las mediciones de todos los medidores a su alcance para luego transmitir los datos al servidor de la empresa. Este tipo de conexión se utiliza en medidores de electricidad así como de agua y gas por medio de módulos PLC que se instalan en medidores. Sin embargo, como las líneas eléctricas no fueron diseñadas para transmitir datos la comunicación puede ser compleja y ruidosa, lo que causa interferencia y da como resultado comunicaciones inestables. Los principales factores de interferencia son las grandes variaciones en las impedancias de carga que producen cambios en el factor de potencia de la línea dinámicamente, atenuación selectiva en frecuencias portadoras producida por aparatos eléctricos encendidos al azar y ruido en la línea producto de fuentes conmutadas. A pesar de que existen métodos para reducir estas interferencias, estos resultan complejos de implementar ya que se debe analizar las características de cada línea de

transmisión a corregir.

El principal estándar de PLC para redes inteligentes es PRIME, basado en Acceso Múltiple por División de Frecuencias Ortogonales (OFDM) y Modulación por desplazamiento de fase (BPSK, DQPSK y D8PSK) como modulación. Las frecuencias de transmisión pueden ser por la banda A de CENELEC (42-89KHz) con velocidades de transmisión entre 5.4 y 128.6 Kbps dependiendo del tipo de modulación, y frecuencias de hasta 471 KHz en bandas de ARIB y FCC.

2.2. Tipos de arquitecturas del Sistema

La forma en que se organizan los medidores para enviar los datos hasta un servidor centralizado depende de la tecnología de comunicación que se utilice. Existen tres tipos de arquitecturas en las que se pueden disponer las redes de medición inteligente: Topología jerárquica o en árbol, topología de malla y conexión directa o punto a punto.

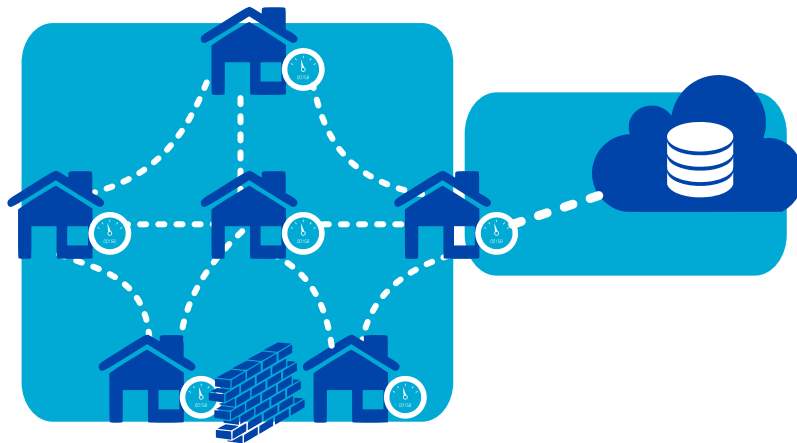


Figura 2.1: Topología de malla

La topología de malla -como se muestra en la Figura 2.1- es capaz de ajustarse dinámicamente ante interferencias externas, por lo tanto posee redundancia en los enlaces, si falla un nodo la comunicación no se ve afectada. En esta arquitectura uno de los nodos asume el rol de enrutador y es el que recibe todos los datos de la red para transmitirlos a través de Internet a fin de ser almacenados. Como desventajas se encuentra la complejidad que puede alcanzar la red así como el cálculo de tamaño

ideal de nodos que compone cada red. Al ser una arquitectura dinámica requiere calcular las rutas más eficientes y su posterior almacenamiento lo que encarece los medidores al requerir hardware más potente que otras arquitecturas.

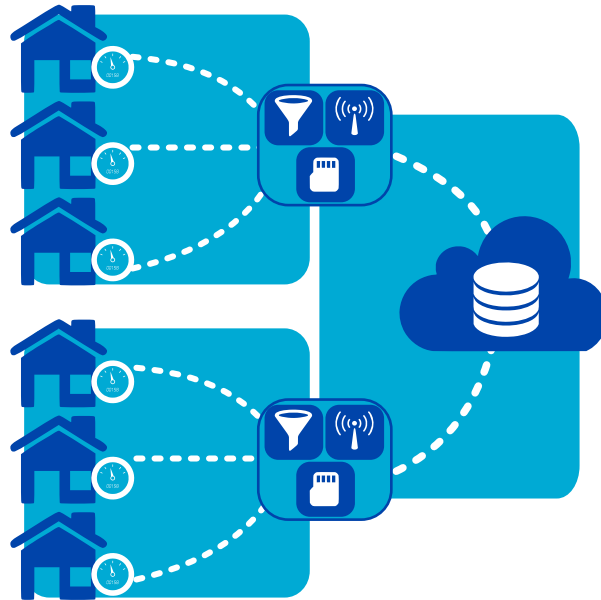


Figura 2.2: Topología jerárquica

La arquitectura jerárquica de la Figura 2.2 posee un equipo específico, generalmente llamado coordinador o concentrador, encargado de recibir la información de cada nodo y transmitirla hacia el servidor. Este dispositivo se encuentra en la raíz del esquema. Como ventajas se tiene que la carga de procesamiento se la lleva el coordinador en la raíz del sistema, lo que permite que los nodos requieran menos capacidad de procesamiento reduciendo costos, además, es una arquitectura fácil de escalar, mantener y diseñar. Por otro lado, una de sus principales desventajas es que al poseer un concentrador encargado de conectar un gran número de nodos, en caso de falla de este, se pierden todos los datos de los nodos atendidos.

La conexión directa es la más simple de las tres arquitecturas, para este caso, como se representa en la Figura 2.3, cada medidor reporta sus datos al servidor directamente a través de Internet. Como ventaja este método presenta su simpleza, la falla de un nodo no afecta en nada el funcionamiento del resto de los medidores. Como la conexión es directa con el servidor no se requiere un equipo con gran



Figura 2.3: Conexión Directa

capacidad de procesamiento lo que permite equipos de bajo costo. La desventaja principal es que se requiere de un enlace punto a punto para cada medidor lo que aumenta los costos si se utiliza una conexión de datos a través de la red celular.

En general cualquier tecnología de comunicación como las descritas en la Sección 2.1 pueden ser desplegadas en las arquitecturas jerárquicas y de malla. Como caso particular se encuentra la red celular que utiliza la infraestructura existente y no es posible realizar conexiones directas entre equipos lo que la limita a ser utilizada en el último tipo de arquitectura o para proveer el acceso a Internet al concentrador o nodo enrutador en las topologías jerárquicas y de malla respectivamente.

3. Diseño del sistema de adquisición automático de mediciones

En el presente capítulo se desglosa el sistema a implementar en módulos funcionales a fin de analizar en detalle los requerimientos específicos que debe cumplir cada uno de estos para el correcto funcionamiento del sistema en su totalidad. Se deben tomar en consideración los objetivos planteados en la Sección 1.2.

3.1. Arquitectura

La arquitectura escogida es del tipo árbol explicada en la Sección 2.2, con un concentrador que atiende a varios medidores. La principal razón para la elección de esta arquitectura, es debido al amplio espectro de aparatos disponibles en el comercio que operan en este tipo de arquitectura, además de ser el sistema con menor costo en los equipos. Para el caso de la arquitectura tipo malla los medidores requieren mayor capacidad de procesamiento, así como un firmware más avanzado para manejar los enrutamientos dinámicos necesarios, mientras que para una arquitectura de conexión directa el costo por servicios de telefonía móvil aumentan significativamente al aumentar el número de medidores.

La medición inteligente, cualquiera sea su arquitectura, se puede separar por capas como lo muestra la Figura 3.1 publicada por el foro “The Smart Water Networks Forum”[2]. Este diagrama es un símil del modelo de interconexión OSI que representa el flujo de información desde el medidor hasta el análisis de datos.

Para el presente trabajo se abarca hasta la capa 4 de almacenamiento y visualización ya que la 5ta guarda relación con minería de datos ya almacenados y a nivel de prototipo funcional no es una etapa relevante.



Figura 3.1: Modelo de capas SWAN

3.2. Descripción de los módulos

A partir del esquema de la Figura 3.1 se definen los módulos que forman parte del sistema y se llega al esquema que se muestra en la Figura 3.2. Para cada uno de los módulos definidos en la figura se detallan características y requerimientos necesarios que debe cumplir cada uno de estos a fin de tener un sistema funcional desde el punto de vista técnico y regulatorio.

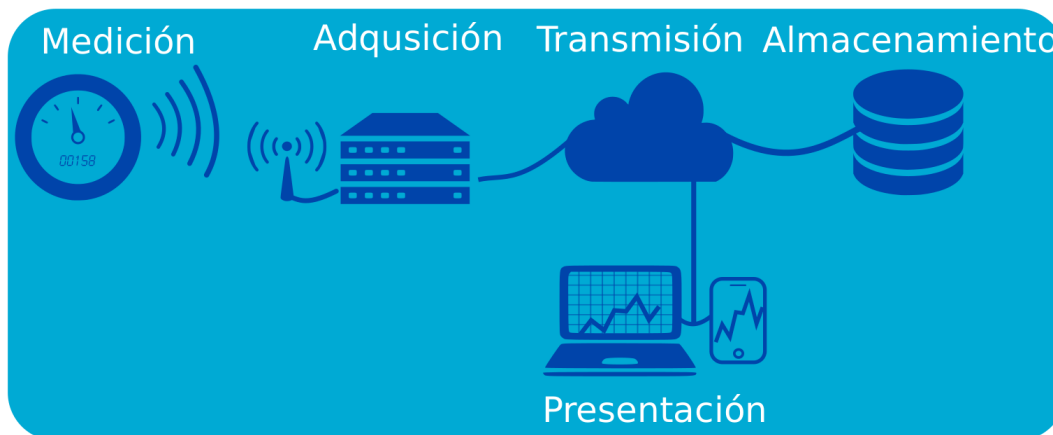


Figura 3.2: Esquema del sistema

3.2.1. Medición

Este módulo está compuesto por el medidor de consumo eléctrico y de ser necesario el dispositivo que provea la conexión hasta el módulo de adquisición. Su función es medir el consumo eléctrico en la residencia del cliente, registrando el consumo e inyección que se genere y transmitir los datos hacia el módulo de adquisición. Las características que debe cumplir son:

- El medidor debe cumplir con la protección física que lo proteja de los elementos a la intemperie así como proteger a las personas que tengan acceso a él, evitando descargas eléctricas.
- La comunicación que se utilice debe cumplir con las disposiciones legales de Chile respecto al medio que se utilice para transmitir, la máxima potencia emisible y características eléctricas del medidor en sí.
- La información que se transmite debe contener el consumo y otros datos relevantes, como un indicador de adulteración del equipo.
- El medidor debe ser capaz de medir consumo eléctrico en ambos sentidos (inyección a la red) para cumplir con la ley 20.571[7]

3.2.2. Adquisición

El módulo de adquisición es el responsable de recibir los datos de todos los medidores que tenga enlazados y reenviar los datos a un sistema centralizado encargado de almacenar el consumo de todos los medidores. Este equipo se conoce comúnmente como concentrador.

El cual debe cumplir con los siguientes requerimientos:

- Debe cumplir con el estándar de comunicación del medidor para poder recibir los datos.
- Requiere autonomía en caso de una falla eléctrica para no perder datos de desde medidores que sigan en funcionamiento.

- Debe poseer memoria de almacenamiento en caso de perder la conexión con el servidor para su descarga manual o reenviarlos al momento que se restablece la conexión
- Debe permitir distintos medios de comunicación con el servidor para incorporar redundancia o flexibilidad a la transmisión de datos.

3.2.3. Transmisión

Este módulo se encarga del transporte de datos desde el concentrador hasta el servidor donde se almacenan. El medio de conexión puede ser a través de un ISP por medio de conexión cableada, móvil como 3g, wifi, etc. O a través de conexión directa punto a punto propiedad de la misma empresa distribuidora de electricidad. Dentro de los requisitos para este módulo se encuentran:

- El ancho de banda de este enlace debe ser capaz de transportar los datos generados sin saturar la línea para evitar pérdidas de datos.
- Si la conexión es provista por un ISP se deben tomar en cuenta los horarios peak de conexión así como la saturación de líneas en caso de situaciones de emergencia.
- Se debe tener en cuenta la seguridad del medio de transmisión a fin de evitar intervenciones en la línea por parte de terceros.

3.2.4. Almacenamiento

Este módulo es el encargado de recibir los datos desde todos los concentradores desplegados, guardarlos de manera organizada y tenerlos disponibles para su posterior análisis, visualización o facturación. Generalmente se utiliza un computador de propósito específico para cumplir esta función dado que debe mantener una alta disponibilidad. Otros requerimientos que debe poseer son:

- Debe ser capaz de almacenar grandes cantidades de datos que se generan en una implementación de cientos de medidores.

- Los datos almacenados deben protegerse ya que se guarda información privada de los usuarios.
- Debe poseer mecanismos que permitan informar de manera oportuna las manipulaciones indebidas por parte de los usuarios
- Se requiere alguna forma de informar sobre cualquier adulteración de los medidor automáticamente sin tener que revisar el registro de cada medidor

3.2.5. Presentación

Este módulo es el último en la cadena de módulos debido a que los datos de consumo deben pasar por cada uno de los anteriores para llegar a este. Es el encargado de ordenar, categorizar y desplegar la información de consumo al usuario de manera simple, dinámica y segura, al mismo protegiendo su información privada. Como características necesarias para este módulo se encuentran:

- La información debe presentarse de manera fácil de entender, como en un gráfico de consumo en el tiempo, así como la monetización del consumo actual.
- Cada usuario debe tener acceso solo a sus datos y no a los de cualquier cliente.
- Como consecuencia del punto anterior, el cliente debe encontrarse registrado en el sistema antes de revisar sus consumos.

4. Implementación del sistema de adquisición automático de mediciones

En este capítulo se desarrolla un sistema funcional a partir de los requerimientos expresados en la Sección 3.2 y la forma en que se configura cada uno de estos para su correcto funcionamiento como sistema integrado. Además se definen cuales de las tecnologías disponibles actualmente detalladas en el Capítulo 2 son las que mejor se adaptan a las necesidades y limitaciones propios del sistema propuesto.

4.1. Módulo Medición



Figura 4.1: Medidor inteligente Itron

El principal componente de este módulo se trata del medidor de consumo eléctrico. Se decide utilizar un medidor comercial en lugar de desarrollarlo desde cero debido a que se trata de un equipo que debe cumplir con los requerimientos

descritos en la Subsección 3.2.1. Por lo tanto el desarrollo de un medidor resulta en una tarea poco provechosa ya que para poder ser instalado y utilizado requeriría todas las certificaciones de seguridad, precisión y radioemisión entre otras.

El medidor utilizado es un Itron Centron C1SR que se muestra en la Figura 4.1 es un medidor unidireccional por lo tanto el flujo de información es en un solo sentido, solo transmite datos.

La comunicación se realiza a través del protocolo ERT[1] el que provee el consumo de energía en kWh, número de identificación del medidor, indicadores de manipulación, tipo de medidor y chequeo de error en cada transmisión. En la Tabla 4.2 se detalla la estructura de los paquetes SCM que son los mensajes en sí que se transmiten en el protocolo ERT.

Tabla 4.1: Características de comunicación del medidor

Tipo de modulación	OOK
Frecuencia de operación	910-920 MHz
Potencia de transmisión	0.75 mW

En la Tabla 4.1 se muestran las principales características de comunicación. La frecuencia de transmisión así como el intervalo entre mensajes son aleatorios para minimizar la interferencia de otros dispositivos. En promedio la transmisión se realiza tres veces por segundo.

Una característica importante acerca de este medidor es que permite «Net Metering», por lo que cumple con la ley 20.571[7]. Otra característica importante de este medidor es la detección de adulteración «tamper detection»[8]. Posee la capacidad de detectar dos tipos de manipulación, por medio de un sensor de péndulo detecta cualquier tipo de movimiento o sacudida del medidor, mientras que el segundo tipo de detección de adulteración está activo solo si la opción de «Net Metering» está desactivada y detecta la inversión de terminales.

Tabla 4.2: Descripción de un paquete SCM

Campo	Largo (bits)	Descripción
Bit de sincronía	1	
Preámbulo	20	
ERT ID Bit MS bits	2	Los dos bits más significativos del número de serie del medidor
Reservado	1	
Tamper físico	2	
Tipo de ERT	4	Es el tipo de servicio (agua, gas, electricidad)
Tamper codificador	2	
Datos de consumo	24	La lectura de consumo del medidor
ERT ID LS bits	24	El resto de bits del número de serie
Checksum	16	Código BHC con generador polinomial

4.2. Módulo Adquisición

En la etapa de adquisición se utiliza generalmente un módulo listo, llamado concentrador que permite la recepción de datos desde distintos medidores que se encuentren dentro del rango de cobertura. Existen dificultades respecto utilizar un equipo comercial:

- Estos equipos están diseñados y programados para interactuar con plataformas de propiedad de la misma empresa fabricante, esto significa que además de contar con el concentrador se debe adquirir el software para alojar estos datos que es de propiedad del proveedor. Lo que resulta ineficiente si se desea utilizar una plataforma propia para alojar estos datos.
- El precio de un concentrador es entre 50 y 100 veces un medidor domiciliario, por lo que escapa al alcance práctico de esta tesis.
- El espectro de medidores que abarca es acotado, por lo que limita los medidores que se pueden implementar.

Debido a todo esto, se toma la decisión de integrar una solución de código abierto un concentrador. Como primer paso se requiere un receptor que sea compatible con el protocolo ERT. Esto se logra a través de un Dongle USB

llamado RTL-SDR, que es una radio definida por software basada en el chip RTL2832U. Este hardware en conjunción con el software desarrollado por Douglas Hall[9] permite la recepción de los mensajes por parte del medidor.

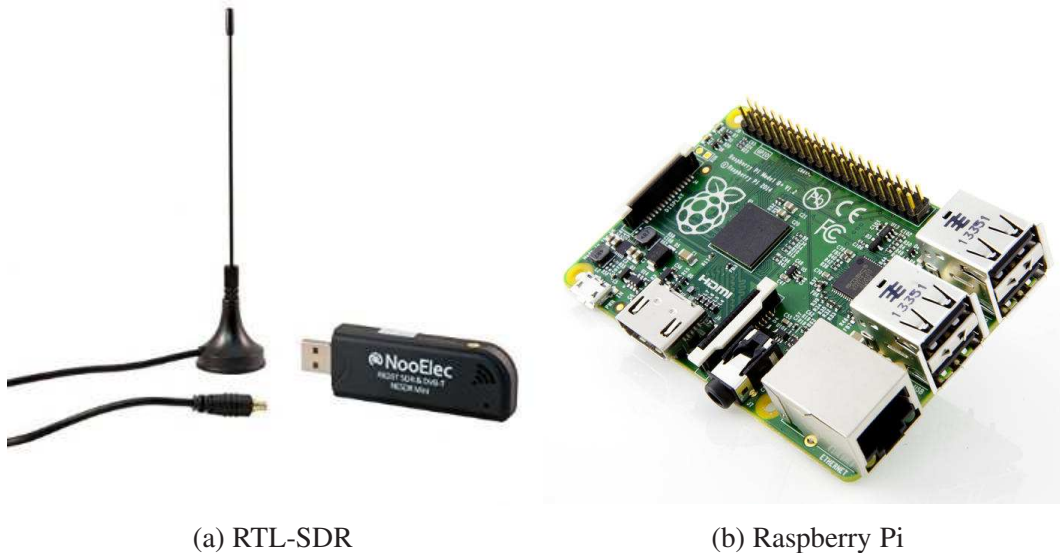


Figura 4.2: Componentes del concentrador

El software, programado en lenguaje GO[14], funciona sobre sistemas operativos Windows y Linux por lo que se opta por un Raspberry Pi como opción económica, de bajo consumo que ofrece una gran versatilidad debido a las interfaces USB y Ethernet que posee junto con un sistema operativo basado en Linux. Las principales características del Raspberry Pi modelo B utilizado se presentan en la Tabla 4.3.

Tabla 4.3: Características Raspberry Pi modelo B

Chip	Broadcom BCM2835
Procesador	ARMv6 single core
Velocidad del procesador	700 MHz
Voltaje y Corriente	600mA a 5V
Memoria Ram	512 MB SDRAM a 400 MHz
GPIO	26
USB 2.0	2
Ethernet	10/100Mbps

Para la instalación del sistema operativo en la Raspberry Pi se utiliza Fedora ARM al que se reemplaza el kernel AMR por el nativo Raspberry para

Tabla 4.4: Particiones de la tarjeta SD

Partición	Tamaño	Descripción
boot	128MB	Debe ser la primera partición y ser formateada en FAT32 la que contendrá el kernel.
swap	256MB	Se debe formatear para 'Linux swap'.
root	Resto	Se le asigna el resto de espacio de la tarjeta con formato EXT4

posteriormente grabarlo en una tarjeta SD, la cual en su estructura más básica debe contener las particiones 'root', 'boot' y 'swap'. Estas particiones se realizan utilizando 'fdisk' en linux siguiendo las recomendaciones indicadas en la página WEB de fedoraproject.org, las que se resumen la Tabla 4.4

Una vez instalado Fedora en la Raspberry Pi, se instalan los programas necesarios para utilizar el dongle USB. En primer lugar se descarga e instala 'golang' desde los repositorios de Fedora, el cual es una suite de desarrollo para el lenguaje de programación go. Posteriormente se instala 'rtlamr'.

```
go get github.com/bemasher/rtlamr
```

El que produce un archivo binario en '/go/bin/rtlamr'. Por último se instala 'rtlsdr' por medio de los repositorios de Fedora, el que provee utilidades de software definido por radio para Dongles basados en el chip Realtek RTL2832.

Para comenzar con la adquisición de datos se debe iniciar el programa rtl_tcp desde un terminal, el cual es un servidor de radioespectro que obtiene la modulación física a partir de la antena USB y deja los datos disponibles en un puerto tcp. Luego desde otro terminal en paralelo se inicia 'rtlamr'.

```
~/go/bin/./rtlamr -format=json -logfile=amrlog.log \
-msgtype=scm -unique=true
```

Al programa se le agregan los parámetros '-format=json' que permite que los datos se almacenen en formato JSON, '-logfile' que permite guardar en un archivo la información en lugar de mostrarla por pantalla, '-msgtype=scm' filtra los mensajes SCM de los IDM o R900 si los hubiera, '-unique=true' el cual permite escribir las

mediciones solo si se produce algún cambio en la medición de otro modo descarta el mensaje, esto es útil para evitar sobrepoblar la base de datos con información idéntica. En la Figura 4.3 se detalla el proceso de obtención del dato a través de ambos programas ‘rtl_tcp’ y ‘rtlamr’.

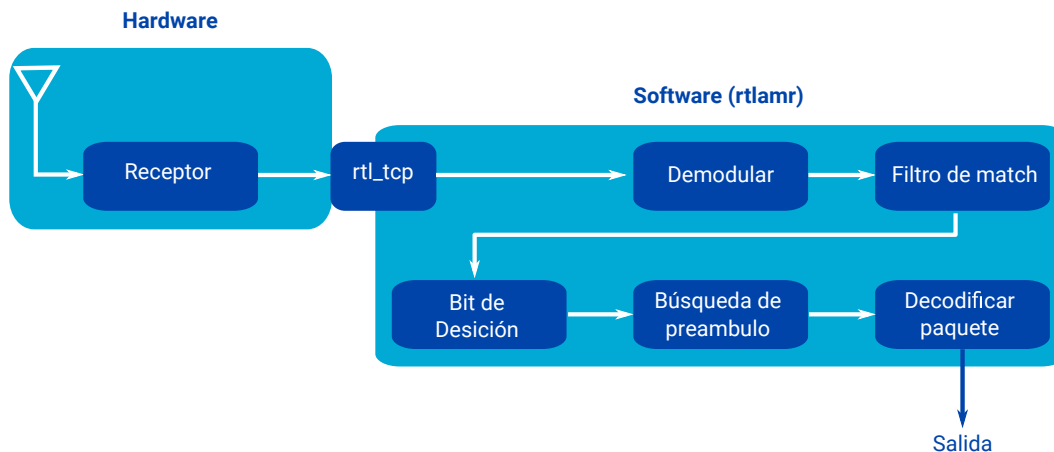


Figura 4.3: Diagrama de flujo de la obtención de la medición

Los datos almacenados en el archivo de texto ‘armlog.log’ deben ser leídos, filtrados y enviados hasta el servidor para ser almacenados. Para realizar estas tareas se elige Python como lenguaje de programación de estas funciones debido a múltiples razones:

- Al ser un lenguaje interpretado permite utilizarlo en múltiples arquitecturas sin realizar mayores cambios.
- Posee una amplia disponibilidad de funciones para realizar tareas comunes como reconocimiento de cadenas caracteres.
- El interprete Python se encuentra instalado por defecto en todas las distribuciones Linux modernas.
- Posee una amplia documentación en línea así como preguntas frecuentes y ejemplos útiles.

Se crea un archivo ‘main.py’ que al ejecutarlo revisa si existe el archivo ‘armlog.log’, de existir envían los datos al servidor y elimina las mediciones almacenadas localmente que son enviadas satisfactoriamente. El envío de

mediciones se realiza a través de de una conexión HTTP por método POST hacia el servidor. Se utiliza el método POST como buena práctica para el envío de este tipo de datos y además permite un tamaño arbitrario de datos a enviar. El funcionamiento de 'main.py' se presenta en la Figura 4.4

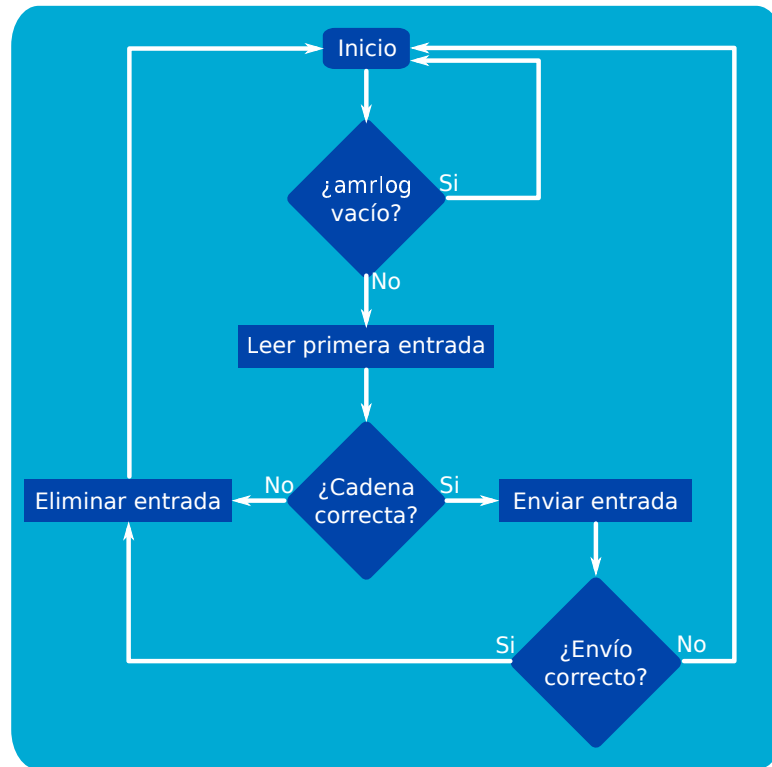


Figura 4.4: Diagrama de flujo concentrador

4.3. Módulo Transmisión

Para los alcances de esta tesis se utiliza una conexión a través de la interfaz Ethernet del Raspberry Pi que pasa a través de un Router hacia el Servidor. Las otras alternativas de conexión que pueden ser aplicadas son conexión a Internet por medio de un Dongle USB 3G. Como la Raspberry Pi posee dos puertos USB esta alternativa es fácilmente implementable.

Otro método disponible es por medio de Wi-Fi a través de una conexión punto a punto por medio de antenas direccionales si se tienen ambas antenas en línea vista.

Existen muchas alternativas distintas para lograr la comunicación entre componentes, las que pueden ser intercambiadas o utilizadas paralelamente a fin

de lograr robustez. Se elige Ethernet como conexión entre adquisición y almacenamiento fin de probar de manera exhaustiva el comportamiento de los demás componentes del sistema.

4.4. Módulo Almacenamiento

Este módulo se compone de un servidor encargado de recibir los datos desde los concentradores, reestructurar los mensajes SCM que recibe para luego almacenarlos en una base de datos.

El sistema operativo utilizado para este servidor es CentOS 6.7 debido que ofrece una base sólida y predecible sobre la cual construir, junto con amplios recursos para desarrollar, probar, liberar y mantener. además de eficiencia en el consumo de recursos.

La instalación de CentOS se realiza en una máquina virtual con la interfaz de red en modo bridge para que posea su propia dirección IP y sea visible desde una red local. A esta máquina virtual se le asignan dos núcleos de procesador a 2.5 GHz con 1024 MB de memoria RAM. Luego de la instalación del sistema operativo se instalan los programas necesarios para cumplir con los requerimientos descritos en el diseño. Se utilizan los siguientes programas:

Apache: Servidor WEB con PHP para manejar la capa de negocios. Se elige Apache por ser uno de los servidores WEB más utilizados en el mundo con una extensa documentación y un desarrollo muy maduro, junto con su arquitectura modular que permite instalar solo las funcionalidades necesarias a fin de lograr un mejor desempeño y menos carga en el servidor.

MySQL: Se elige MySQL como base de datos por su amplia adopción a nivel mundial y por consiguiente gran cantidad de documentación. Otra razón para su utilización es el software de desarrollo que posee 'MySQL Workbench' que permite el diseño y mantenimiento de la base de datos de manera remota y segura así como herramientas que facilitan su administración.

Python: Se utiliza el lenguaje interpretado que viene instalado por defecto en la

distribución de CentOS 6.7 (Python 2.6) para proveer el servicio que se muestra en el esquema de la Figura 4.5 para recibir los datos desde el concentrador y almacenarlos en la base de datos.

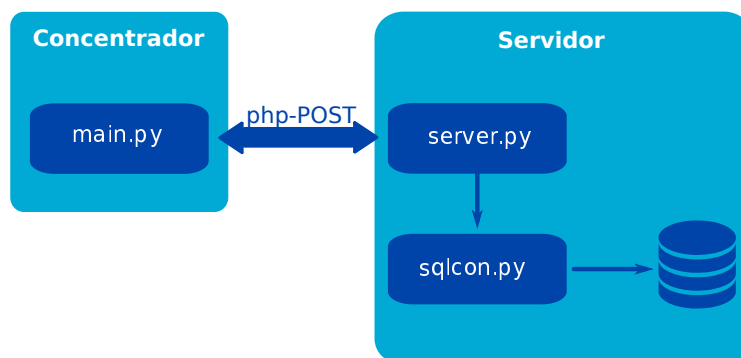


Figura 4.5: Esquema de recepción de datos

La recepción de los datos se realiza según el esquema de la Figura 4.5 en el lado del servidor, 'server.py' levanta un servidor HTTP en la dirección IP del servidor con puerto 8000. Al recibir la cadena de datos desde el concentrador se realiza un análisis y separación de los datos en una estructura JSON para insertar la información a la base de datos por medio de 'sqlcon.py' que recibe los datos como argumento, crea una conexión con la base de datos, inserta los valores de medición y cierra la conexión, para lograr la conexión con la base de datos se debe instalar el módulo MySQL-python.i386 desde el repositorio de CentOS. El detalle de los códigos python del servidor se encuentran en el Apéndice A de los anexos.

Para iniciar el servidor Python se utiliza el programa 'screen'[15] desde un terminal en CentOS lo que permite mantener activa la sesión aún cuando la conexión sea interrumpida, esto es útil ya que permite retomar la sesión por medio de un terminal remoto por ejemplo por medio de un túnel SSH, además permite mantener múltiples terminales disponibles desde una misma sesión SSH.

```
screen python server.py
```

El motor de base de datos, servidor Apache y los módulos necesarios para la comunicación entre ellos se instalan desde los repositorios de CentOS por medio del comando:

```
yum install mysql-server.i686 httpd.i686 php56w.i386\  
php56w-pdo.i386 php-mysql.i686 php56w-pdo.i386 -y
```

Una vez realizada la instalación es conveniente iniciar estos servicios al momento de iniciar el sistema, lo que se realiza por medio del comando:

```
chkconfig httpd on && chkconfig mysqld on
```

Es recomendado iniciar el motor de base de datos con el instalador seguro, el que a través de la línea de comandos permite definir contraseña para el usuario 'root', deshabilitar el acceso para el usuario root desde fuera de la red local, remover las cuentas de usuario anónimos y eliminar la base de datos de prueba que viene implementada por defecto.

```
mysql_secure_installation
```

Con el motor de base de datos instalado se procede a implementar la base de datos que debe contener la información recibida desde los medidores así como información de los clientes y de los concentradores a fin de generar reportes, facturas y otra información que se pueda deducir desde una base de datos relacional. En la Figura 4.6 se muestra el esquema de la base de datos generada a partir de los datos que entregan los medidores y la información que contiene una boleta de electricidad.

Es importante destacar que la contraseña de los usuarios se almacena cifrada para cumplir con las buenas prácticas de seguridad. Por lo que los usuarios no pueden ser registrados directamente desde la base de datos, en cambio la inserción de un nuevo usuario debe hacerse a través de una página WEB PHP en donde la clave ingresada en el registro es codificada a través del algoritmo 'bcrypt' que da como resultado una cadena de 60 caracteres de longitud. Para agregarle seguridad extra a esta página, se utiliza la característica de apache conocida como 'archivo de configuración distribuida' (.htaccess) lo que permite restringir el acceso por medio de contraseña para una página en particular sin modificar la configuración global de las demás páginas servidas.

Para realizar esto es necesario configurar apache para utilizar este archivo de configuración.

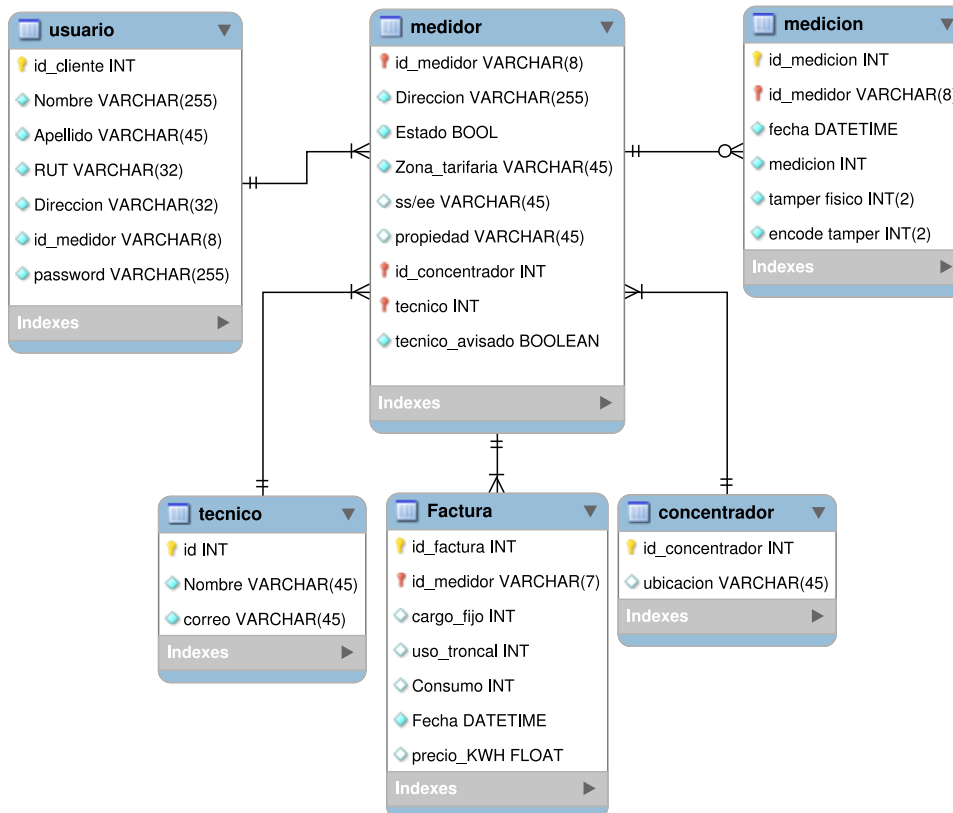


Figura 4.6: Diagrama relacional de la base de datos

En primer lugar se permite el uso de archivos de configuración locales desde la configuración principal de apache, en este caso utilizando el editor de texto ‘vim’.

```
vim /etc/httpd/conf/httpd.conf
```

y modificar en la siguiente sección ‘AllowOverride’ de ‘none’ a ‘All’

```
<Directory /var/www/html>
    Options Indexes FollowSymLinks
    AllowOverride All
    Order allow,deny
    allow from all
</Directory>
```

Realizada la modificación se guarda el archivo y reinicia apache.

```
service httpd restart
```

Posteriormente se crea un archivo con los usuarios y contraseñas los que permitirán el ingreso a la página WEB que crea nuevos clientes en la base de datos.

```
htpasswd -c /etc/httpd/.htpasswd nombre_usuario
```

Finalmente en la carpeta donde se encuentran las páginas WEB se debe crear un archivo ‘.htaccess’ que restringe el acceso a la página que inserta nuevos usuarios a la base de datos. Dentro de este archivo se agregan las siguientes líneas que indican la ruta al deposito de claves y determina que archivo afecta

```
AuthName "Acceso Restringido"  
AuthType Basic  
AuthUserFile /etc/httpd/.htpasswd  
<Files "password.php">  
Require valid-user  
</Files>
```

En el archivo ‘password.php’ se agrega un formulario HTML el cual inserta al usuario a la base de datos.

Otro requerimiento importante del servidor es verificar si los medidores instalados han sufrido algún tipo de manipulación indebida por parte del usuario. Los datos de adulteración vienen incluidos en la información que se ingresa a la base de datos pero es necesario filtrar la información a fin de determinar que técnico es responsable de la revisión de cada medidor.

En la Figura 4.7 se presenta un diagrama de flujo donde se observa la lógica que debe seguir el programa responsable de monitorear el estado de los medidores instalados. En la tabla donde se almacenan los medidores existe una entrada que corresponde al técnico responsable de cada medidor, así como un indicador booleano que indica si el técnico ya fue avisado a fin de no saturar su correo con información sobre la misma adulteración. Este código está escrito en python y se mantiene en ejecución constante. Al momento de detectar una adulteración desde la tabla donde se almacenan los datos de medición, ubica desde la tabla de

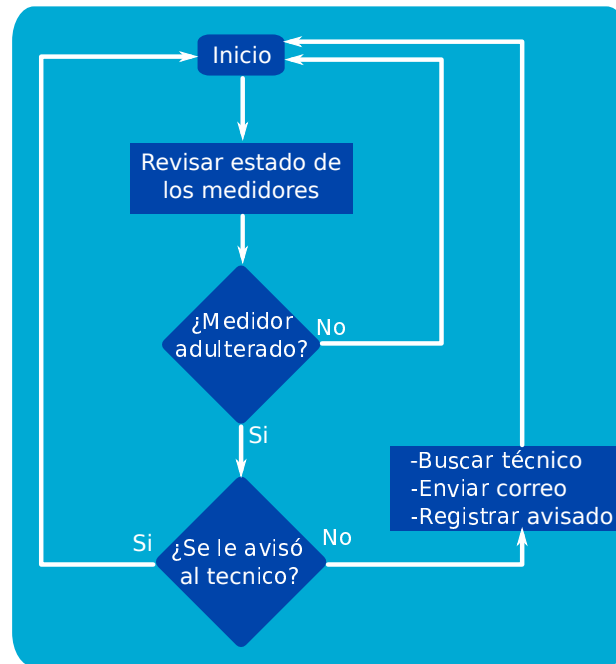


Figura 4.7: Diagrama de flujo para aviso de adulteración

medidores al técnico responsable así como el indicador de aviso, si el indicador de aviso no ha sido modificado se le envía un correo al técnico responsable y modifica el indicador para registrar que el técnico ya ha sido informado del problema. El correo se envía a través del protocolo SMTP por lo que solo requiere un servidor de correo que utilice SMTP para enviar las notificaciones, para las pruebas realizadas se utiliza la plataforma de correos de Google.

4.5. Módulo Presentación

En este módulo se implementa en el mismo computador que el módulo de almacenamiento y es el encargado de recuperar la información desde la base de datos, filtrar los consumos de cada usuario y mostrarlos en una página WEB en forma de gráfico de área. El diseño de las páginas que componen este módulo se ajustan al diagrama de flujo presentado en la Figura 4.8.

La página de inicio 'login.php' se compone principalmente de un formulario HTML que envía los datos ingresados a través de POST hacia la página 'login_submit.php' la cual se conecta a la base de datos a través de "PHP Data Objects" (PDO) en lugar de las extensiones clásicas como MySQL o MySQLi.

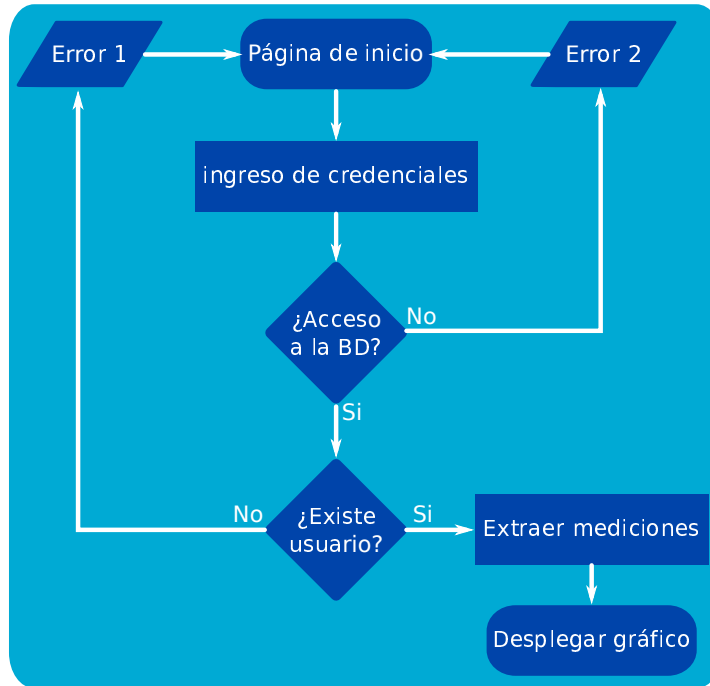


Figura 4.8: Diagrama de flujo Módulo Presentación

Esto permite cambiar el motor de bases de datos sin alterar el código cambiando solo la sintaxis específica de las consultas para cada motor de base de datos. Esta extensión soporta hasta el momento 10 motores de bases de datos distintos entre los que se destacan ‘Microsoft SQL Server’, ‘MySQL’, ‘Oracle Database’, ‘PostgreSQL’ y ‘SQLite’ entre otras. Permite además la utilización de consultas preparadas, esto es consultas SQL precompiladas que pueden ser ejecutadas múltiples veces con la ventaja añadida que evita los ataques de inyección SQL.

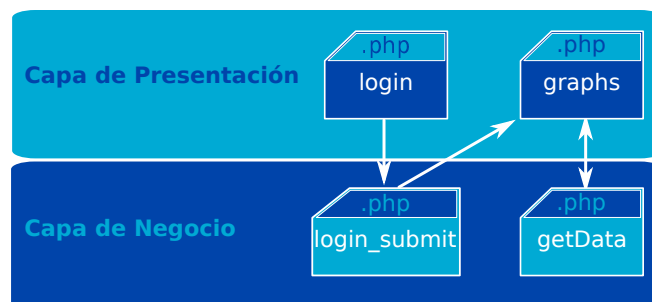


Figura 4.9: Esquema de archivos PHP

Una vez realizada la conexión a la base de datos, se busca por el usuario ingresado y comprueba la contraseña por medio de la aplicación y comparación del Hash almacenado. De no existir el usuario o la contraseña ingresada es

incorrecta se regresa a la página de inicio con un mensaje de error. El paso siguiente corresponde a crear una sesión PHP que se puede definir como una variable 'superglobal' la cual está disponible en cualquier parte del script a través de todas las páginas. En esta variable se guarda el ID del medidor para consultar más adelante en otra página por las mediciones que se graficarán. Posteriormente se redirige la conexión hacia 'graphs.php', compuesta por un código 'javascript' diseñado por Google que genera un gráfico de área. La página 'graphs.php' posee además un selector que permite elegir el mes de consumo que se desea observar. Los datos utilizados para generar el gráfico se obtienen de la base de datos por medio de la página 'getData.php' que a través de una serie de consultas SQL a la base de datos filtra los datos por id del medidor y rango de fechas elegidas en la página anterior. Esta información recuperada desde la base de datos es codificada en forma de JSON para finalmente ser recibida por el código javascript que la muestra en una página WEB. En la Figura 4.9 se muestra un esquema del flujo de información entre los archivos PHP.

5. Verificación y Evaluación

A continuación se procede a verificar el correcto funcionamiento del sistema desde la adquisición por parte del medidor, la correcta recepción de datos desde el concentrador, la inserción de la información a la base de datos sin problemas, la autenticación de los usuarios y el despliegue de los gráficos solicitados por el usuario. Además de comprobar el mecanismo que avisa al técnico correspondiente cuando se detecta una adulteración.

Una vez verificado el correcto funcionamiento de cada componente del sistema se realizan pruebas más exhaustivas a cada módulo con el objetivo de conocer las limitaciones de estos y realizar una evaluación de su desempeño.

5.1. Puesta en marcha del sistema

En primer lugar se debe poblar manualmente algunas tablas de la base de datos. Se crea una entrada en la tabla 'concentrador' con un ID y una ubicación, luego la tabla 'tecnico' con un nombre y correo de contacto, la tabla 'medidor' con el ID del medidor utilizado y el ID del técnico creado en el paso anterior, la tabla 'factura' con el ID del medidor y un valor para los KWh, finalmente se crea un usuario por medio de la página WEB 'password.php' rellenando el formulario que se muestra en la Figura 5.1

Luego continúa el concentrador, que después de conectar el Dongle USB se ejecutan los programas necesarios para la recepción y manejo de los mensajes de consumo. El primer programa que debe ser ejecutado es 'rtl_tcp', se debe revisar la salida estándar del programa para confirmar que reconoce correctamente el dispositivo USB e inicia el servicio en la dirección IP local 127.0.0.1 con el puerto por defecto 1234

La salida del programa debe ser similar a la que se muestra en la Figura 5.2. Esto confirma el correcto funcionamiento del primer programa.

A screenshot of a web browser window. The address bar shows the URL `http://172.16.0.143/password.php`. The page content is a registration form with the following fields: "Nombre", "Apellido", "RUT", "Dirección", "ID Medidor", and "Contraseña". Each field is represented by a rectangular input box. Below the fields is a button labeled "Registrar".

Figura 5.1: Página registro de usuarios

```
[sebastian@pez ~]$ rtl_tcp
Found 1 device(s):
 0: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Detached kernel driver
Found Rafael Micro R820T tuner
Tuned to 1000000000 Hz.
listening...
Use the device argument 'rtl_tcp=127.0.0.1:1234' in OsmoSDR (gr-osmosdr) source
to receive samples in GRC and control rtl_tcp parameters (frequency, gain, ...).
```

Figura 5.2: Inicio servicio RTL TCP

Luego se debe iniciar 'rtlamr' el cual se conecta al servicio generado por 'rtl_tcp' con los parámetros definidos en la Sección 4.2 y se muestra en Figura 5.3. Al utilizar los parámetros ahí definidos el programa no entrega información por la salida estándar, por lo que es necesario revisar el archivo generado manualmente para comprobar la correcta recepción de los datos. Esto se realiza luego de energizar el medidor y por medio del comando 'tail', se deben observar las cadenas JSON añadirse al final del archivo periódicamente de la forma que se muestra en la Figura 5.4.

Posterior a esto, se debe levantar el servicio encargado de recibir los datos en el servidor a través de protocolo HTTP. Se inicia en el servidor 'server.py' el cual queda en espera de las conexiones entrantes.

```
[sebastian@pez bin]$ ./rtlamr -format=json -logfile=amrlog.log -msgtype=scm -unique=true
```

Figura 5.3: Inicio decodificador de mensaje

```
[sebastian@pez Escritorio]$ tail -F amrlog.log
{"Time": "2016-04-01T12:24:59.02505069-03:00", "Offset": 0, "Length": 35840, "Message": {"ID": 60243827, "Type": 4, "TamperPhy": 0, "TamperEnc": 3, "Consumption": 31160, "ChecksumVal": 40837}}
{"Time": "2016-04-01T12:24:59.249563039-03:00", "Offset": 0, "Length": 35840, "Message": {"ID": 60243827, "Type": 4, "TamperPhy": 0, "TamperEnc": 3, "Consumption": 31160, "ChecksumVal": 40837}}
{"Time": "2016-04-01T12:24:59.473489595-03:00", "Offset": 0, "Length": 35840, "Message": {"ID": 60243827, "Type": 4, "TamperPhy": 0, "TamperEnc": 3, "Consumption": 31160, "ChecksumVal": 40837}}
```

Figura 5.4: Revisión de los mensajes generados

```
[root@centos home]# python server.py
{"Time": "2016-05-01T13:25:40.894914347-03:00", "Offset": 0, "Length": 35840, "Message": {"ID": 60243828, "Type": 4, "TamperPhy": 0, "TamperEnc": 3, "Consumption": 31167, "ChecksumVal": 40837}}
pez - - [10/Apr/2016 21:14:50] "POST /testurl HTTP/1.1" 200 -
```

Figura 5.5: Servidor receptor de mensajes

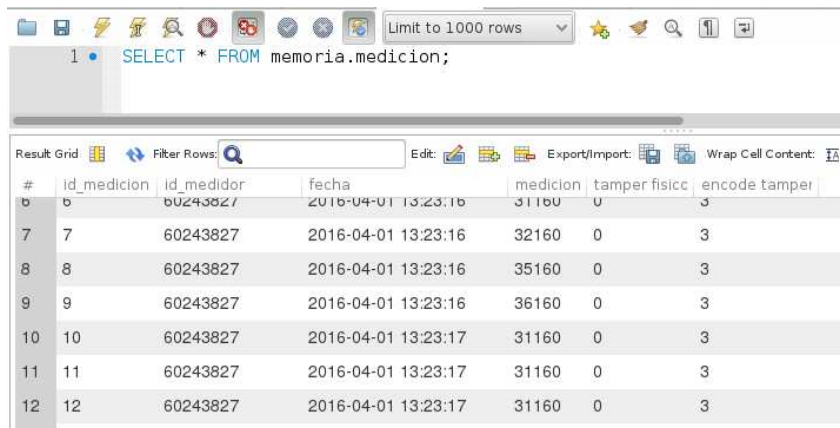
Con todo lo anterior en funcionamiento, el último paso para el concentrador es poner en marcha el código Python que monitorea los cambios en archivo ‘amrlog.log’ en busca de nuevas entradas que son enviadas al servicio ‘server.py’ levantado en el paso anterior. En la Figura 5.6 se comprueba que ‘main.py’ en el concentrador detecta los cambios al archivo que se monitorea y los envía a través de HTTP hacia ‘server.py’ el cual recibe el dato correctamente a través del método POST de HTTP con el estado 200 ‘ok’ como se observa en la Figura 5.5

```
[sebastian@pez concentrador]$ python main.py
watching logfile /home/sebastian/Escritorio/amrlog.log
{"Time": "2016-05-01T13:25:10.894914347-03:00", "Offset": 0, "Length": 35840, "Message": {"ID": 60243828, "Type": 4, "TamperPhy": 0, "TamperEnc": 3, "Consumption": 31167, "ChecksumVal": 40837}}
<httplib.HTTPResponse instance at 0x7f2eb2995098>
```

Figura 5.6: Envío de mensajes al servidor

Para comprobar que los mensajes se insertan correctamente a la base de datos, por medio de la aplicación ‘MySQL Workbenck’ se revisa la tabla ‘medicion’

verificando que los datos fueron insertados como se muestra en la Figura 5.7



The screenshot shows a database query tool interface. At the top, there is a toolbar with various icons and a dropdown menu set to 'Limit to 1000 rows'. Below the toolbar, a SQL query is entered: `SELECT * FROM memoria.medicion;`. The main area displays a 'Result Grid' with the following data:

#	id_medicion	id_medidor	fecha	medicion	tamper fisicc	encode tamper
6	6	60243827	2016-04-01 13:23:16	31160	0	3
7	7	60243827	2016-04-01 13:23:16	32160	0	3
8	8	60243827	2016-04-01 13:23:16	35160	0	3
9	9	60243827	2016-04-01 13:23:16	36160	0	3
10	10	60243827	2016-04-01 13:23:17	31160	0	3
11	11	60243827	2016-04-01 13:23:17	31160	0	3
12	12	60243827	2016-04-01 13:23:17	31160	0	3

Figura 5.7: Inserción de mediciones a la BD

Posteriormente, se revisa el correcto despliegue de datos a través de la plataforma WEB utilizando la cuenta de usuario creada en los pasos anteriores. El resultado se aprecia en la captura de pantalla que se muestra en la Figura 5.9.

Finalmente, se verifica que se generen los avisos de adulteración, para esto se modifica una entrada desde la base de datos aumentando el contador de 'tamper' mientras el servicio encargado de detectar manipulaciones se encuentra levantado. El correo resultante debe indicar el ID del medidor y la dirección donde se encuentra como muestra la Figura 5.8.



Figura 5.8: Correo de aviso de adulteración

Con este resultado se comprueba que el sistema cumple con los requerimientos listados en el Capítulo 3. Los datos transmitidos por el medidor son recibidos por el concentrador, filtrados y enviados al servidor para ser procesados e insertados en la base de datos para su posterior despliegue en la plataforma WEB para la visualización de consumos por parte del usuario, además de la capacidad de avisar por medio de correo electrónico la adulteración de cualquier medidor al técnico correspondiente.

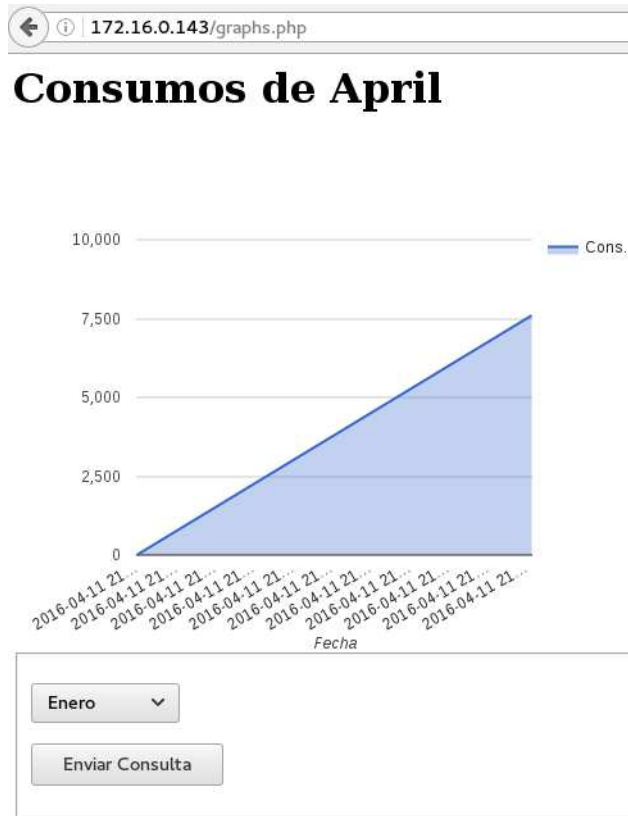


Figura 5.9: Despliegue de consumos

5.2. Pruebas de estabilidad y desempeño del sistema

Las pruebas que se realizan apuntan a determinar las capacidades de distintas partes del sistema cuando se encuentran bajo estrés, esto puede ser altas cargas o situaciones anormales como problemas de conectividad y principalmente el número de medidores que se puede atender por cada concentrador.

Alcance de transmisión del medidor

Esta prueba realizada por Andrés Silva, uno de los integrantes del proyecto, tiene por objetivo determinar la máxima distancia que puede existir entre el medidor y concentrador a fin de recibir las mediciones consistentemente. Para lograr un escenario más realista se realizaron pruebas en interior como en exterior, de estas se obtiene que el máximo alcance del medidor en un ambiente urbano como lo es una calle rodeada por casas es de hasta 110 metros, el detalle de la

metodología utilizada se encuentra en la memoria de titulación realizada por Andrés[12]

Cantidad de medidores que puede atender

Se refiere a la capacidad del Dongle USB en recibir simultáneamente desde múltiples medidores. Al no contar con múltiples medidores es necesario recurrir a la información publicada sobre el software que interactúa con la antena USB donde informa que mide por lo menos 300 medidores de manera continua e intermitentemente desde otros 600[9].

Ancho de banda al enviar datos

Es necesario determinar el ancho de banda que se utiliza al enviar datos desde el concentrador hasta el servidor. Esta información es necesaria para determinar el mínimo ancho de banda necesario para enviar los datos de manera correcta sin saturar la conexión y la cantidad máxima de mediciones por segundo que puede recibir a fin que la tasa de envío sea mayor a la tasa de lecturas desde los medidores.

Para probar esto se pobla el archivo ‘armlog.log’ con 2MB de mediciones, lo que corresponde aproximadamente a 11400 mediciones. Luego por medio del programa ‘iftop’ se monitorea el tráfico por la interfaz de red que une al concentrador con el servidor, los cuales se encuentran conectados a través de cable Ethernet que se observa en la Figura 5.10 la velocidad de transmisión es en promedio 2.4KB/s, en total la conexión necesaria para enviar datos de manera correcta es de 39.6Kb/s.

TX:	cum:	242KB	peak:	21,8Kb	rates:	21,8Kb	19,2Kb	18,8Kb
RX:		247KB		34,6Kb		34,6Kb	20,4Kb	18,6Kb
TOTAL:		488KB		56,4Kb		56,4Kb	39,6Kb	37,4Kb

Figura 5.10: Velocidad de transmisión

Para determinar la tasa de transferencia en ‘mediciones por segundo’ se utiliza el comando en Linux ‘time -v’ el cual entrega información sobre tiempo de ejecución. Para realizar esta prueba se modifica el código ‘main.py’ de modo que envíe solo diez mediciones cambiando el bucle ‘while’ por un bucle ‘for’ a fin de obtener

el tiempo total en realizar diez envíos. El resultado se observa en la Figura 5.11. En promedio le toma al programa 239 milisegundos enviar una medición hasta el servidor.

```
Command being timed: "python main.py"
User time (seconds): 0.36
System time (seconds): 1.09
Percent of CPU this job got: 60%
Elapsed (wall clock) time (h:mm:ss or m:ss): 0:02.39
```

Figura 5.11: Tiempo de ejecución

Por lo tanto en promedio puede enviar cuatro mediciones por segundo. Lo que corresponde a la frecuencia con que transmite un medidor.

Capacidad de almacenamiento local

Durante el funcionamiento normal del concentrador los datos enviados al servidor correctamente son eliminados, en el caso de perder conexión con el servidor el archivo ‘armlog.log’ continua almacenando las mediciones nuevas que se reciben así como las mediciones anteriores que no pudieron ser enviadas al servidor. La importancia de esto es determinar la tasa en que crece el archivo hasta ocupar completamente la memoria en la tarjeta SD del Raspberry Pi a fin de determinar los tamaños necesarios para esta tarjeta así como los requisitos de disponibilidad de la conexión.

Cada entrada el archivo ‘armlog.log’ tiene un tamaño de 176 Bytes y el medidor es capaz de transmitir tres mensajes por segundo¹. Se considera que el concentrador debe atender 50 medidores, por lo que el máximo teórico de mensajes que se deben almacenar son 26.4KBytes por segundo. A esta tasa se logra llenar una tarjeta SD con 8GBytes dedicados al almacenamiento en 84 horas o 3.5 días.

¹El consumo promedio mensual de electricidad de un hogar chileno de cuatro personas es aproximadamente de 220 kWh[10]. Como el software de recepción guarda en el archivo solo cuando se produce un cambio en el medidor esto se traduce en aproximadamente 8 mensajes diarios dado que la granularidad del medidor es KWH

Carga en el servidor

El siguiente componente importante del sistema es el servicio encargado de recibir los datos desde el concentrador, desempaquetarlos e insertarlos en la base de datos. Es importante determinar la eficiencia del servicio en utilización de recursos del sistema así como determinar que no tenga fugas de memoria que puedan conducir a un mal funcionamiento del servidor.

Para esto se analizan las estadísticas del servidor por medio de la aplicación 'htop' mientras se realiza en envío de datos desde el concentrador. Mientras se mantiene en funcionamiento se observa el porcentaje de CPU que tiene asignado así como el porcentaje de memoria RAM en busca de cambios significativos en el tiempo en busca de posibles problemas. Como se observa en la Figura 5.12 la ocupación del procesador oscila entre 1.9 y 2.4 % para un concentrador conectado. Al aumentar a cuatro el número de concentradores se observa un aumento a 7.5 % de uso de CPU manteniendo en todas las pruebas una ocupación de memoria RAM de un 0.7 % durante dos horas de prueba.

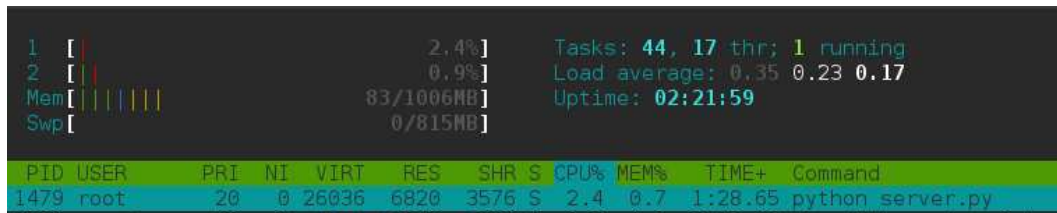


Figura 5.12: Estadísticas de sistema del servidor

Esto representa un 1.8 % promedio de ocupación de CPU por cada concentrador conectado. Estos porcentajes son en base a una máquina virtual de dos procesadores a 2.5 MHz, con 1024 MB de memoria RAM.

Consultas en la base de datos

Se deben revisar los tiempos que toma realizar las principales consultas SQL realizadas por la interfaz WEB en el servidor, debido a que es posible saturar el servidor si las consultas SQL no están lo suficientemente optimizadas para soportar la carga requerida.

Para realizar esta medición se debe tomar en cuenta que la tabla que almacena las mediciones es la que posee mayor cantidad de entradas que todas las demás tablas juntas. Por lo que las consultas SQL realizadas a esta tabla son las que requerirán más tiempo de ejecución. Se comienza poblando la tabla con mediciones con 77000 entradas. Para luego realizar las consultas SQL directamente en el motor de base de datos para obtener el tiempo total de la consulta como perfiles de cada consulta que desglosan el tiempo que le toma cada uno de los pasos que componen la consulta. Al realizar todas las consultas SQL para revisar su tiempo de ejecución se encuentran tres que poseen tiempos de ejecución de tres ordenes de magnitud mayores que el resto. La consulta que más tarda es la encargada de obtener los datos que se despliegan en el gráfico WEB. Debe consultar el valor kWh actual desde otra tabla y restar la primera medición del mes a todas las demás mediciones para empezar el conteo desde cero cada mes. Este proceso toma 237 milisegundos, en comparación seleccionar todas las mediciones almacenadas toma 182 milisegundos. Finalmente la consulta encargada de revisar si existe alguna adulteración del medidor toma 200 milisegundos.

6. Conclusiones y trabajo futuro

A partir del trabajo expuesto se confirma que el sistema desarrollado cumple bajo condiciones de laboratorio con los objetivos propuestos de recolectar, transmitir y almacenar los datos de los usuarios, además de desplegar dicha información para los usuarios así como avisar al personal responsable sobre la manipulación de los medidores. El sistema es capaz de transportar el dato desde el medidor hasta un servidor de manera confiable a pesar de las posibles pérdidas de conexión. Aún cuando el sistema está pensado para consumo eléctrico, el protocolo de mensajes ERT se utiliza en medidores de otros servicios básicos, los que se detallan en el Apéndice B. Esto permite utilizar esta misma infraestructura indistintamente del servicio que se desea medir. Esta flexibilidad permite también que las mediciones sean integradas a cualquier plataforma que pueda poseer la empresa distribuidora de servicios básicos como son los sistemas de planificación de recursos empresariales (ERP) ya que los datos no son dependientes de un software específico que limite su exportación.

De las pruebas realizadas se determinan aspectos destacables del sistema. El concentrador es capaz de atender una cantidad razonable de medidores de manera satisfactoria considerando que los datos se reciben solo cuando ocurre algún cambio en la medición o parámetro de ‘tamper’. Se toma en cuenta que el consumo promedio mensual de electricidad de un hogar chileno de cuatro personas es aproximadamente 220 kWh [10]. Esto quiere decir que cada medidor registrará en promedio ocho mediciones diarias, por lo que cada concentrador puede abarcar 30 medidores con facilidad si es ubicado en una posición estratégica que permita una línea vista con pocos obstáculos entre el concentrador y los medidores.

Como debilidades limitaciones se tiene el código que lee los datos del archivo ‘armlog.log’ posee un ‘overhead’ grande y no resulta muy eficiente ya que luego de cada lectura se requiere reescribir el archivo completo. Por otro lado los medidores utilizados poseen comunicación unidireccional en consecuencia no es

posible realizar todas las características de una red de medición inteligente como los cortes de servicio remoto o implementar modalidades como consumo prepagado.

La adopción de esta tecnología permite implementar nuevas modalidades de consumo que no son posibles en el sistema actual, como tarifas diferenciadas por horarios de consumo lo que puede incentivar el uso de electricidad en momentos de baja demanda, además el sistema es más seguro ya que si bien ninguna tecnología es invulnerable, se requiere una persona con más conocimientos técnicos en electrónica e informática para intentar vulnerar el sistema además de los recursos necesarios pasan de ser un destornillador y una lima a una antena capaz de inyectar mensajes propios o bien desarmar el medidor y modificar la memoria interna.

La medición inteligente tiene por objetivo aumentar la eficiencia de las empresas tanto distribuidoras como generadoras, permitiendo un mayor control de sus procesos al conocer de manera precisa los consumos de los usuarios finales. A su vez, esta tecnología le permite al usuario un empoderamiento sobre sus consumos ya que puede tomar acciones correctivas al instante sin tener que esperar a fin de mes para ver el resultado.

Finalmente la buena recepción de estas tecnologías por parte del usuario juega un rol preponderante por lo que resulta imprescindible antes de la adopción de este tipo de tecnologías realizar campañas de educación a la gente así como tratar con extremo cuidado la información sensible de los consumos individuales de los usuarios.

Este sistema al tratar con información que se utiliza para realizar los cobros a los clientes, la principal prioridad es la seguridad e integridad de los datos por lo que el trabajo futuro se debe centrar en mejorar o incorporar medidas que prevengan intrusiones al sistema. Con esto en cuenta, la primera medida que debe ser añadida es un certificado SSL a fin de permitir una conexión cifrada entre el concentrador y el equipo que aloja los datos, así como la conexión de los usuarios al sistema. Otra característica necesaria a futuro es la disponibilidad de los servicios que entrega el computador centralizado. Se deben implementar medidas que permitan configurar

los servicios en alta disponibilidad, así como respaldos de los datos en equipos en ubicaciones diferentes para mayor seguridad.

Bibliografía

- [1] P. Robinson, M. Ouellette, and L. Schmidt, “Automatic meter reading and control system”, US, 4.315.251 Febrero 9 1982.
- [2] “The Smart Water Networks Forum”, [en línea] (<http://www.swan-forum.com/>), [consulta: 30 agosto 2015].
- [3] “The Smart Grid Alliance for the Americas, Chilean pilot.” [en línea], (<http://www.nreca.coop/what-we-do/international-programs/smart-grid-initiative/smart-grid-alliance/pilot-projects/chile/>), [consulta: 6 de agosto 2015].
- [4] “Smartcity Santiago” [en línea], (<http://www.smartcitysantiago.cl/>), [consulta: 6 agosto 2015].
- [5] “Aqualogy Revenue Management, Caso aguas andinas”, [en línea], (<http://www.smartmetering.aqualogy.net/es/casos/19/aguas-andinas-gran-santiago—chile->), [consulta: agosto 2015].
- [6] M. Bustamante Moltedo. “Rediseño del proceso de control de pérdidas de energía eléctrica: ’transformador de distribución como eje articulador en la gestión de las pérdidas de energía’ ”, Tesis (Ing. Civil Industrial), Universidad de Chile, Departamento de Ingeniería Industrial 2009.
- [7] Ley 20.571, “Regula el pago de las tarifas eléctricas de las generadoras residenciales” Diario Oficial de la República de Chile, Santiago, Marzo 2012.
- [8] D. Johnson, M. Wiebe, “Tamper detection device for utility meter”, US, 5.086.292, Febrero 1992.
- [9] Douglas Hall, “An rtl-sdr receiver for Itron ERT compatible smart meters operating in the 900MHz ISM band”, [en línea], (<https://github.com/bemasher/rtlamr>), [consulta: septiembre 2015].

- [10] “Chile renueva sus energías”, [en línea], [⟨http://www.chilerenuevaenergias.cl/index.php?option=comk2&view=item&layout=item&id=10⟩](http://www.chilerenuevaenergias.cl/index.php?option=comk2&view=item&layout=item&id=10), [consulta: 30 enero 2016].
- [11] “Stop smart meters!”, [en línea], [⟨http://stopsmartmeters.org/how-you-can-stop-smart-meters/⟩](http://stopsmartmeters.org/how-you-can-stop-smart-meters/), [consulta: febrero 2016].
- [12] A. Silva Torres “Sistema centralizado de recolección automática de consumo de servicios básicos con acceso vía WEB”, Tesis (Ing. Civil Telemática), Valparaíso. UTFSM, Departamento de Ingeniería Telemática 2016.
- [13] Superintendencia de Electricidad y Combustibles, [en línea], “Ranking calidad del servicio eléctrico 2014”, [⟨http://www.sec.cl/portal/page?pageid=33,3435546&dad=portal&schema=PORTAL⟩](http://www.sec.cl/portal/page?pageid=33,3435546&dad=portal&schema=PORTAL), [consulta: 12 diciembre 2015].
- [14] The GO Programming Language [en línea] [⟨https://golang.org/⟩](https://golang.org/) [consulta: 01 junio 2016]
- [15] Screen Manual [en línea] <https://www.gnu.org/software/screen/manual/screen.html> [consulta: 01 junio 2016]

A. Códigos fuente

A continuación de presentan los códigos utilizados en el proyecto

Concentrador

main.py:

```
1 import re
2 import time
3 import httpLib
4 import socket
5 import subprocess
6
7
8 def get_lock(process_name):
9     global lock_socket
10    lock_socket = socket.socket(socket.AF_UNIX, socket.SOCK_DGRAM)
11    try:
12        lock_socket.bind('\0' + process_name)
13        print 'I got the lock'
14    except socket.error:
15        print 'lock exists'
16        sys.exit()
17
18 get_lock('data_sender')
19
20 PATH = "armlog.log"
21 PATTERN = re.compile('\.{173}\}')
22 SERVER="172.16.0.155:8000"
23 while True:
24
25     with open(PATH, "rb") as infile:
26         for line in infile:
```

```

27         reading = PATTERN.findall(line)
28         if reading:
29             try:
30                 conn = httplib.HTTPConnection(SERVER)
31                 conn.request("POST", "/testurl", reading[0])
32                 conn.getresponse()
33                 conn.close()
34                 subprocess.call("ex -sc 'ld|x' /home/sebastian
/armlog.log", shell=True)
35             except socket.error:
36                 print "coudn't connect"
37                 continue
38         else:
39             subprocess.call("ex -sc 'ld|x' /home/sebastian/
armlog.log", shell=True)
40             print "error line"
41             continue

```

Python Servidor

Los códigos a continuación son los encargados de recibir las mediciones desde el concentrador.

server.py:

```

1  __author__ = "Sebastian Apablaza, Andres Silva"
2  __copyright__ = "Copyright 2016"
3  __license__ = "Apache 2.0"
4  __version__ = "0.1"
5  __email__ = "sebastian.apablaza@alumnos.usm.cl, andres.
        silvat@alumnos.usm.cl"
6  __status__ = "Development"
7
8  import json
9  import sqlcon
10 from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer
11
12 ADDR = '172.16.0.143'

```

```

13 PORT = 8000
14 IP = "127.0.0.1"
15 USER = "concentrador"
16 PASSWRD = "concentradorA1"
17 DB = "memoria"
18 connection = sqlcon.Connect(IP, USER, PASSWRD, DB)
19
20 class RequestHandler(BaseHTTPRequestHandler):
21     def do_POST(self):
22         length = int(self.headers['Content-length'])
23         reading = (self.rfile.read(length))
24         print reading
25         if reading:
26             data = json.loads(reading)
27             data["Time"] = data["Time"].split(".")[0].replace("T",
28 " ")
29             sqlcon.SendData(connection, (data["Message"])["ID"],
30 data["Time"], (data["Message"])["Consumption"],(data["Message"]
31 )["TamperPhy"],(data["Message"])["TamperEnc"])
32             self.send_response(200, "OK")
33             self.end_headers()
34
35 httpd = HTTPServer((ADDR, PORT), RequestHandler)
36 httpd.serve_forever()
37 sqlcon.Close(connection)

```

sqlcon.py:

```

1 __author__ = "Sebastian Apablaza, Andres Silva"
2 __copyright__ = "Copyright 2016"
3 __license__ = "Apache 2.0"
4 __version__ = "0.3"
5 __email__ = "sebastian.apablaza@alumnos.usm.cl, andres.
6 silvat@alumnos.usm.cl"
7
8 __status__ = "Development"
9
10 import MySQLdb
11
12 def Connect(ip, user, passwd, db):

```

```

11     con = MySQLdb.connect(ip ,user , passwd ,db)
12     return con
13
14 def SendData (connection , measure_id , datetime , measure , phy ,
15     encoded):
16     cursor = connection.cursor()
17     cursor.execute("""INSERT INTO 'memoria'. 'medicion' ( '
18     id_medidor ', 'fecha ', 'medicion ', 'tamper fisico ', 'encode
19     tamper ') VALUES (%s, %s, %s, %s, %s)""" , (measure_id , datetime ,
20     measure ,phy , encoded) )
21     return

```

Los siguientes códigos corresponden al servicio que verifica la adulteración de los medidores.

tamper.py:

```
1 __author__ = "Sebastian Apablaza"
2 __copyright__ = "Copyright 2016"
3 __license__ = "Apache 2.0"
4 __version__ = "0.3"
5 __email__ = "sebastian.apablaza@alumnos.usm.cl"
6 __status__ = "Development"
7
8 import sqlcon
9 import mail
10 import time
11
12 ip = "127.0.0.1"
13 user = "concentrador"
14 passwd = "concentradorA1"
15 db = "memoria"
16 mailfrom = "prueba@gmail.com"
17 mailpasswd = "password"
18
19 while True:
20     conn = sqlcon.Connect(ip, user, passwd, db)
21     TamperedList = sqlcon.SearchTamper(conn)
22     if TamperedList:
23         for row in TamperedList:
24             info = sqlcon.GetMail(conn, row[0])
25             if not info[1]:
26                 message = "id: "+info[2)+"\nDireccion "+info[3]
27                 state = mail.send_email(mailfrom, mailpasswd, info
[0], 'medidor adulterado', message)
28                 if state:
29                     sqlcon.Notified(conn, info[2])
30                 else:
31                     continue
32             else:
33                 continue
```

```
34 sqlcon.Close(conn)
35 time.sleep(5)
```

mail.py:

```
1 __author__ = "Sebastian Apablaza"
2 __copyright__ = "Copyright 2016"
3 __license__ = "Apache 2.0"
4 __version__ = "0.3"
5 __email__ = "sebastian.apablaza@alumnos.usm.cl"
6 __status__ = "Development"
7
8 import smtplib
9
10 def send_email(user, pwd, recipient, subject, body):
11
12     gmail_user = user
13     gmail_pwd = pwd
14     FROM = user
15     TO = recipient if type(recipient) is list else [recipient]
16     SUBJECT = subject
17     TEXT = body
18
19     message = """\From: %s\nTo: %s\nSubject: %s\n\n%s""" % (FROM,
20     ", ".join(TO), SUBJECT, TEXT)
21
22     try:
23         server = smtplib.SMTP("smtp.gmail.com", 587)
24         server.ehlo()
25         server.starttls()
26         server.login(gmail_user, gmail_pwd)
27         server.sendmail(FROM, TO, message)
28         server.close()
29         return 1
30
31     except:
32         return 0
```

PHP servidor

Estos códigos son los que permiten agregar usuarios a la base de datos, el control de ingreso de usuario y mostrar los consumos por medio de gráficos.

login.php:

```
1 <html>
2 <head>
3 <title>Consumo centralizado</title>
4 </head>
5 <body>
6 <h2>Autenticacion</h2>
7 <form action="login_submit.php" method="post">
8 <fieldset>
9 <p>
10 <label for="username">Usuario (rut)</label>
11 <input type="text" id="username" name="username" value=""
    maxlength="20" />
12 </p>
13 <p>
14 <label for="password">Contraseña</label>
15 <input type="password" id="password" name="password" value=""
    maxlength="20" />
16 </p>
17 <p>
18 <input type="submit" value="Ingresar" />
19 </p>
20 </fieldset>
21 </form>
22 <H4>
23 <?php
24 if(isset($_GET['error'])) {
25     if($_GET['error'] == 1){
26         echo "usuario o contraseña no validos";
27     }
28     elseif ($_GET['error'] == 2) {
29         echo "Error al conectarse al servidor";
```

```

30 }
31 }
32 ?>
33 </H4>
34 </body>
35 </html>

```

login_submit.php:

```

1 <?php
2 $username = $_POST[ 'username' ];
3 $password = $_POST[ 'password' ];
4 $mysql_hostname = 'localhost';
5 $mysql_username = 'login';
6 $mysql_password = 'concentradorA1';
7 $mysql_dbname = 'memoria';
8     try{
9         $dbh = new PDO("mysql:host=$mysql_hostname;dbname=$mysql_dbname"
10             , $mysql_username , $mysql_password);
11     }
12     catch(PDOException $ex){
13         echo "Error!<br>";
14         echo $ex;
15         header( 'Location: http://172.16.0.143/login.php?error=2' );
16     }
17 $sth= $dbh->prepare("SELECT * FROM usuario WHERE RUT = :username")
18     ;
19 $sth->bindParam( ':username' , $username);
20 $sth->execute();
21 $row = $sth->fetch();
22 $dbh = null;
23 $hash = password_hash($row[ 'password' ], PASSWORD_DEFAULT);
24 if ( password_verify($password , $row[ 'password' ])) {
25     session_start();
26     $_SESSION[ 'user' ] = $username;
27     $_SESSION[ 'id' ] = $row[ 'id_medidor' ];
28     header( 'Location: http://172.16.0.143/graphs.php' );
29 }
30 else {

```

```

29 header('Location: http://172.16.0.143/login.php?error=1');
30 }
31 ?>

```

graphs.php:

```

1 <html>
2 <head>
3   <title>Grafico de consumo</title>
4 </head>
5 <?php
6   session_start();
7   if(isset($_POST['month'])){
8     $mo = $_POST['month'];
9     $_SESSION['di'] = date("Y-m-d H:i:s", mktime(00, 00, 00, $mo,
10      01, 2016));
11    $_SESSION['df'] = date("Y-m-d H:i:s", mktime(00, 00, 00, $mo+1,
12      01, 2016));
13    $_SESSION['month'] = date('F', mktime(00, 00, 00, $mo, 01, 2016)
14      );
15    echo "<H1>Consumos de " . $_SESSION['month'] . "</H1>";
16  }
17  else{
18    $_SESSION['di'] = date("Y-m-d H:i:s");
19    $_SESSION['df'] = date("Y-m-d H:i:s");
20    $_SESSION['month'] = date('F');
21    echo "<H3>Consumos de " . date('F') . "</H3>";
22  }
23 ?>
24 <script type="text/javascript" src="https://www.gstatic.com/charts
25   /loader.js"></script>
26 <script type="text/javascript" src="//ajax.googleapis.com/ajax/
27   libs/jquery/1.10.2/jquery.min.js"></script>
28 <script type="text/javascript">
29   google.charts.load('current', {'packages':['corechart']});
30   google.charts.setOnLoadCallback(drawChart);
31   function drawChart() {
32     var jsonData = $.ajax({
33       url: "getData.php",

```

```

29     dataType: "json",
30     async: false
31   }).responseText;
32   var data = new google.visualization.DataTable(jsonData);
33   var options = {
34     title: '',
35     hAxis: {title: 'Fecha', titleTextStyle: {color: '#333'}},
36     vAxis: {minValue: 0}
37   };
38   var chart = new google.visualization.AreaChart(document.
39     getElementById('chart_div'));
40   chart.draw(data, options);
41   </script>
42 </head>
43 <body>
44   <div id="chart_div" style="width: 500px; height: 400px;"></div
45   >
46 </body>
47 <fieldset>
48 <p>
49 <form action="" method="post">
50 <select name="month">
51   <option value='01'>Enero </option>
52   <option value='02'>Febrero </option>
53   <option value='03'>Marzo </option>
54   <option value='04'>Abril </option>
55   <option value='05'>Mayo </option>
56   <option value='06'>Junio </option>
57   <option value='07'>Julio </option>
58   <option value='08'>Agosto </option>
59   <option value='09'>Septiembre </option>
60   <option value='10'>Octubre </option>
61   <option value='11'>Noviembre </option>
62   <option value='12'>Diciembre </option>
63 </select>
64 </p>

```

```

64 <p>
65 <input type="submit" name="Consultar" />
66 </p>
67 </form>
68 </fieldset >
69 </html>

```

getData.php:

```

1 <?php
2 session_start();
3 $id = $_SESSION['id'];
4 $mysql_hostname = 'localhost';
5 $mysql_username = 'login';
6 $mysql_password = 'concentradorA1';
7 $mysql_dbname = 'memoria';
8 try{
9     $dbh = new PDO("mysql:host=$mysql_hostname;dbname=$mysql_dbname"
10         , $mysql_username , $mysql_password);
11     $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
12 }
13 catch(PDOException $ex){
14     echo "Error!<br>";
15     echo $ex;
16 }
17 $array['cols'][] = array('label'=>'fecha', 'type' => 'string');
18 $array['cols'][] = array('label'=>'Consumo CLP', 'type' => 'number'
19     );
20 $sth= $dbh->prepare("SELECT fecha , ROUND((medicion - (SELECT
21     medicion FROM memoria.medicion WHERE fecha BETWEEN :init_d AND
22     :end_d ORDER BY fecha ASC LIMIT 1))*(SELECT precio_KWH FROM
23     memoria.Factura),0) AS 'medicion' FROM medicion WHERE
24     id_medidor = :id AND fecha BETWEEN :init_d AND :end_d", array(
25     PDO::ATTR_CURSOR => PDO::CURSOR_SCROLL));
26 $sth->bindParam(':id', $id);
27 $sth->bindParam(':init_d', $_SESSION['di']);
28 $sth->bindParam(':end_d', $_SESSION['df']);
29 $sth->execute();
30 while ($row = $sth->fetch(PDO::FETCH_NUM, PDO::FETCH_ORI_NEXT)) {

```

```

24 $array[ 'rows' ][] = array( 'c' => array( array( 'v'=>$row[0] ),
    array( 'v'=> (INT)$row[1]) ) );
25 }
26 echo json_encode( $array );
27 ?>

```

password.php:

```

1 <?php
2 if ( isset( $_POST[ 'name' ] ) ) {
3 $hash = password_hash( $_POST[ 'password' ], PASSWORD_DEFAULT );
4 $mysql_hostname = 'localhost';
5 $mysql_username = 'login';
6 $mysql_password = 'concentradorA1';
7 $mysql_dbname = 'memoria';
8 try {
9 $dbh = new PDO( "mysql:host=$mysql_hostname;dbname=$mysql_dbname"
    , $mysql_username , $mysql_password );
10 }
11 catch ( PDOException $ex ) {
12 echo "Error!<br>";
13 echo $ex;
14 }
15 $sth= $dbh->prepare( "INSERT INTO usuario (Nombre, Apellido, RUT,
    Direccion, id_medidor, password) VALUES (?, ?, ?, ?, ?, ?)");
16 $sth->bindParam( 1, $_POST[ 'name' ] );
17 $sth->bindParam( 2, $_POST[ 'lastname' ] );
18 $sth->bindParam( 3, $_POST[ 'RUT' ] );
19 $sth->bindParam( 4, $_POST[ 'addr' ] );
20 $sth->bindParam( 5, $_POST[ 'id_meter' ] );
21 $sth->bindParam( 6, $hash );
22 $sth->execute();
23 echo "Cliente " . $_POST[ 'name' ] . " creado <br>";
24 }
25 ?>
26 <html>
27 <form action="" method="post">
28 <fieldset>
29 <p>

```

```
30 <label for="name">Nombre</label>
31 <input type="text" id="name" name="name" value="" maxlength="20"
    />
32 </p>
33 <p>
34 <label for="lastname">Apellido</label>
35 <input type="text" id="lastname" name="lastname" value=""
    maxlength="20" />
36 </p>
37 <p>
38 <label for="RUT">RUT</label>
39 <input type="text" id="RUT" name="RUT" value="" maxlength="20" />
40 </p>
41 <p>
42 <label for="addr">Direccion</label>
43 <input type="text" id="addr" name="addr" value="" maxlength="20"/>
44 </p>
45 <p>
46 <label for="id_meter">ID Medidor</label>
47 <input type="text" id="id_meter" name="id_meter" value=""
    maxlength="20"/>
48 </p>
49 <p>
50 <label for="password">Contrasena</label>
51 <input type="password" id="password" name="password" value=""
    maxlength="20"/>
52 </p>
53 <p>
54 <input type="submit" value="Registrar" />
55 </p>
56 </fieldset >
57 </form>
58 </html>
```


B. Medidores compatibles

Manufacturer	Model Name	Commodity	Lower (MHz)	Upper (MHz)
Itron	40EN	Electric		
Itron	40ER	Electric		
Itron	40ER+AC0-1	Electric		
Itron	40ES	Electric		
Itron	41ER+AC0-1	Electric		
Itron	45EN+AC0-1	Electric		
Itron	45ER+AC0-1	Electric		
Itron	45ES+AC0-1	Electric	910	920
Itron	50ESS	Electric	910	920
Itron	51ESS	Electric	910	920
Itron	52ESS	Electric	910	920
Itron	53ESS	Electric	902	928
Itron	54ESS	Electric		
Itron	54ESS+AC0-HP	Electric		
Itron	55ESS	Electric		
Itron	55ESS+AC0-HP	Electric		
Itron	56ESS	Electric		
Itron	56ESS+AC0-HP	Electric		
Itron	57ESS+AC0-HP	Electric		
Itron	AMI4	Electric	902.2	927.8
Itron	AMI6	Electric	902.2	927.8
Itron	AMI6	Electric	909.6	921.8
Itron	C1A+AC0-2	Electric	910	920
Itron	C1A+AC0-2	Electric	917.6	917.6
Itron	C1A+AC0-3	Electric	909	922
Itron	C2SOD	Electric	902.2	927.8
Itron	C3A+AC0-1H	Electric	909	922
Itron	C3A1L	Electric	909	922
Itron	CVSO+AC0-B	Electric	902.2	927.8

Itron	R300	Electric		
Itron	R300+AC0-HP	Electric		
Itron	R300C	Electric		
Itron	R300CD	Electric		
Itron	R300CD+AC0-HP	Electric		
Itron	R300CD3+AC0-HP	Electric		
Itron	R300S	Electric		
Itron	R300S2	Electric	909.3	918.4
Itron	R300SD	Electric		
Itron	R300SD3	Electric		
Itron	100G	Gas	903	928
Itron	100GDLAN	Gas	908	926.8
Itron	100T	Gas	903	926.9
Itron	25G +AC0- 1 foot	Gas		
Itron	25G +AC0- 2 foot	Gas		
Itron	40G	Gas		
Itron	40GB	Gas		
Itron	100W	Water	903	927
Itron	100WP	Water		
Itron	40W	Water		
Itron	50W	Water		
Itron	60W	Water	910	919.8
Itron	60WP	Water		
Itron	80W+AC0-i	Water	910	920
Landis+-Gyr	AirPoint Series 41	Electric	910	920
Landis+-Gyr	AirPoint Focus	Electric	913672	916138
Landis+-Gyr	AirPoint I+AC0-210	Electric	913672	916138
Landis+-Gyr	AirPoint iCon	Electric	913.75	916.25
Landis+-Gyr	HP AirPoint	Electric	909586	921773
Landis+-Gyr	HP AirPoint	Electric	909586	921773
Landis+-Gyr	HP AirPoint I+AC0-210	Electric	909586	921773
Schlumberger	CENTRON OOK RF	Electric	917.58	917.58

C. Licencia Apache 2.0

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

”License” shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

”Licensor” shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

”Legal Entity” shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, control” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50 %) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

”You”(or ”Your”) shall mean an individual or Legal Entity exercising permissions granted by this License.

”Source” form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

”Object” form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

”Work” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

”Derivative Works” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original

work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.”

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution

incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

a. You must give any other recipients of the Work or Derivative Works a copy of this License; and b. You must cause any modified files to carry prominent notices stating that You changed the files; and c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and d. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the

Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of

any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS