

*Escuela Politécnica
Del Ejército*

Tesis de Grado

*Un Sistema de
Monitoreo de Energía
Eléctrica mediante
Internet*

Denyss Estévez Montalvo

CAPÍTULO 1

INTRODUCCIÓN

1.1. Objetivos

1.1.1. Objetivo General

Desarrollar un sistema que permita el monitoreo de energía eléctrica en ambientes industriales para reducir los costos operacionales de la empresa y hacerla más competitiva en el mercado.

1.1.2. Objetivos Específicos

- Reducir los costos operacionales de la empresa mediante un sistema de calidad que permita el monitoreo del flujo de energía, minimización de costos de licencias, utilizando clientes browsers y programas de libre distribución en la medida de lo posible.
- Implementar una barrera tecnológica ante la competencia, utilizando software y hardware de punta.
- Capturar los datos de los parámetros de energía mediante el uso de tarjetas de adquisición de datos y permitir recuperación de los mismos para elaborar estadísticas.
- Diseñar un software que permita monitorear las variables desde cualquier computador mediante un browser en la Intranet local o en el Internet.
- Analizar de herramientas con respecto al costo beneficio

1.2. Metodología - Norma IEEE 830

El estándar fue creado en 1985, año en que se tenía como estándar la metodología estructurada de diseño, luego en el año 1992 se hicieron las correcciones necesarias para acoplar esta norma al diseño orientado a objetos que es el estándar actual. A continuación se muestra la norma en su diseño original.

1. Introducción
 - 1.1. Propósito
 - 1.2. Alcance
 - 1.3. definiciones, acrónimos, abreviaturas
 - 1.4. Referencias
 - 1.5. Apreciación General
2. descripción General
 - 2.1. Perspectiva del producto
 - 2.2. Funciones del producto
 - 2.3. Características del usuario
 - 2.4. Limitaciones

- 2.5. Asunciones y dependencias
- 3. Especificación de requerimientos
 - 3.1. Interfaces externas
 - 3.1.1. Interfaces de usuario
 - 3.1.2. Interfaces de hardware
 - 3.1.3. Interfaces de software
 - 3.1.4. Interfaces de comunicaciones
 - 3.2. Funciones
 - 3.2.1. Flujos de información
 - 3.2.2. descripción de los procesos
 - 3.2.3. Diccionario de datos
 - 3.3. Requerimientos de desempeño
 - 3.4. Requerimientos lógicos de base de datos
 - 3.5. Limitaciones de diseño
 - 3.6. Atributos del sistema de software
 - 3.6.1. Confiabilidad
 - 3.6.2. Disponibilidad
 - 3.6.3. Seguridad
 - 3.6.4. Mantenibilidad
 - 3.6.5. Portabilidad

Seguidamente se muestra las correcciones que se deben hacer a la norma con la metodología estructurada para que se convierta a la orientada a objetos

Tabla 1.1 – Equivalencias de la Norma IEEE 830 estructurada a objetos

Información necesitada	Representación de la Información
Cuales son las funciones del sistema?	Casos de uso
Cuales son los atributos del sistema?	descripción textual
Cuales son los roles y responsabilidades de las clases importantes del sistema?	Diagramas de clases Diagramas de interacción Diagramas de transición de estados
la arquitectura propuesta soportará el producto?	Prototipos ejecutables

1.3. Planteamiento del Problema

Mediante el estudio y las estadísticas realizadas a nivel nacional en las industrias se ha podido observar que el 95% de éstas no cuentan con un sistema de monitoreo de la energía que consume su maquinaria y, por lo tanto no saben si la misma es de buena calidad o está afectando el tiempo de vida útil de sus equipos y maquinaria.

Una energía eléctrica de mala calidad puede causar averías y daños en algunos casos irreparables a los equipos eléctricos y electrónicos así como también a la maquinaria industrial, especialmente a los que poseen motores y que son muy costosos.

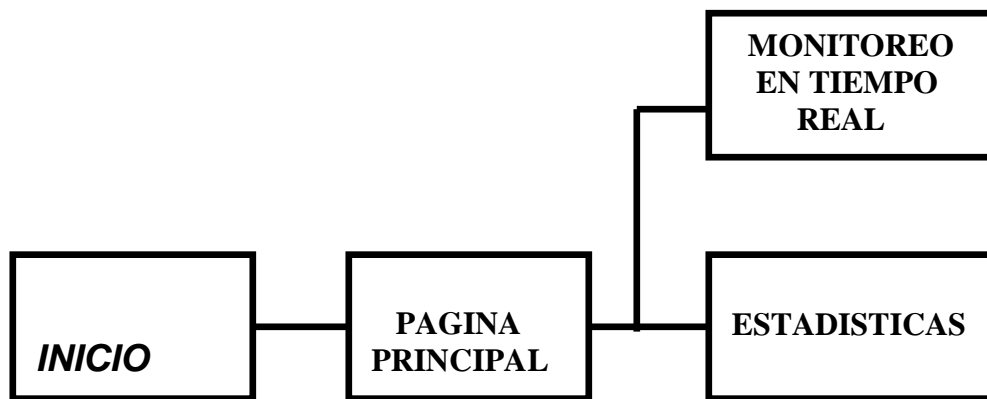
Algunas industrias grandes, así como las medianas y pequeñas no disponen del capital suficiente para invertir en equipos de propósito específico para monitoreo de energía eléctrica; por lo tanto se hace necesario encontrar una forma alternativa y económica para permitir a estas industrias monitorear la calidad de energía eléctrica que están consumiendo y poder tomar los correctivos necesarios para proteger la maquinaria y extender su tiempo de vida útil.

1.4. Justificación y Alcance

Como alcance dado a este proyecto se realizará una medición de los valores de los parámetros por los que se cuantifica la energía eléctrica (voltaje, corriente, factor de potencia), visualización de los mismos y cálculos de otros importantes que se obtienen de los primeros (potencias, potencias reactivas, energías, energías reactivas).

La aplicación realizará un monitoreo en forma constante del flujo de energía, para obtener datos históricos y estadísticos, los cuales permitirán con mayor facilidad tomar decisiones de tipo operativo y administrativo dentro de la empresa.

Al incursionar en Internet, se debe incorporar parámetros que permitirán operar con mayor seguridad. Por lo que se utilizará todos los mecanismos de hardware y software disponibles y que principalmente estén al alcance económico de las empresas.



1.5. Enfoque de Investigación

La realización del presente trabajo cubre varios campos tecnológicos del conocimiento y una infinidad de campos industriales de aplicación; en los campos del conocimiento se tiene la Electrónica y por supuesto los Sistemas e Informática y, entre los campos de aplicación industrial de este sistema se mencionan algunas industrias como:

- Alimenticia
- Automotriz
- Química
- Maderera

- Metalmecánica
- Siderúrgica
- Farmacéutica
- Plástica
- Conservera
- Licorera

y otras más que se verán beneficiadas de la implantación de la investigación en cuestión.

Se sabe ahora que no sólo el país sino el mundo entero está sufriendo una recesión económica bastante grave, y se sabe también que una de las maneras más acertadas de que las empresas e industrias puedan sobrevivir en el agresivo mundo de la libre competencia es la reducción de sus costos operativos.

Las industrias que se han mencionado anteriormente tienen un elevado consumo de energía eléctrica de tipo industrial y sobre la cual no tienen un apropiado control ni monitoreo y por lo tanto un desembolso económico desconocido, por esa razón y algunas más que se mencionan más adelante se ha visto que la realización de la presente investigación puede generar grandes ahorros para la empresa. La utilización de esta energía genera un costo operativo fijo que el empresario lo traslada a los productos que genera su fábrica; pero no sabe realmente que es lo que está pagando porque no conoce la calidad de la energía eléctrica ni cómo ésta puede afectar sobre el tiempo de vida útil de su maquinaria. Tampoco sabe si está aprovechando realmente todo lo que paga o existe un desperdicio de la misma, ni qué acciones o medidas puede tomar para hacer el mejor uso de la misma.

1.6. Descripción de la Problemática

Actualmente algunas industrias poseen sus medidores de consumo de energía eléctrica pero estos no indican la calidad de la misma, ni recogen datos para poder crear estadísticas de uso, ni realizan cálculos para poder obtener ciertos parámetros importantes dentro de los términos de la energía eléctrica. Algunas industrias bastante grandes poseen ciertos equipos que realizan este tipo de tareas antes indicadas pero sus costos son excesivamente altos, de tal forma que llega a ser prohibitivo para las pequeñas y medianas industrias. Además estos equipos son de propósito específico, es decir que no se puede reprogramarlos para que realicen otras tareas adicionales o diferentes de las ya establecidas.

Por todas las razones antes mencionadas el presente proyecto propone una manera más económica, flexible, estable, escalable, portable, segura y de alta calidad de realizar estas tareas y muchas más sobre una planta industrial de cualquier tamaño.

Para la realización de este proyecto se considera pertinente que se debe tener conocimientos sobre energía eléctrica industrial, como medirla, que puede afectarla; además ciertos conocimientos sobre monitoreo de señales, acondicionamiento y tratamiento de de las mismas.

Una vez resuelta la parte eléctrica y electrónica se necesitarán conocimientos sobre sistemas informáticos; estos deben versar sobre aplicaciones distribuidas, aplicaciones y seguridades en Internet, redes de datos, plataformas Windows y Linux entre lo más importante a mencionar.

1.7. Bosquejo de Solución y Desarrollo

El proyecto tienen su gran importancia debido a que busca la reducción de los costos de implantación del mismo utilizando software de libre distribución, tarjetas de adquisición de datos de multipropósito, un computador estándar compatible como herramienta de proceso, visualización y monitoreo y, finalmente un software de cliente que sea muy sencillo e intuitivo de manejar de tal forma que la empresa ahorre también costos en la capacitación de los usuarios.

Este proyecto tiene como idea principal hacer uso de tarjetas multipropósito de adquisición de datos que se puede colocar en un computador estándar y configurarlas con mucha facilidad para que el sistema sea flexible, escalable y estable; luego hacer uso de una interfaz de cliente muy sencilla de manejar en la que se pueda implementar ciertos mecanismos de seguridad, que sea libre en su costo de licencias de uso de software, multiplataforma para que el empresario pueda aprovechar la infraestructura ya existente en su planta industrial para de esta forma crear un sistema económico, portable, seguro y de alta calidad.

El sistema se encargará de recopilar la información de la energía eléctrica de la red pública a la que está conectada la planta industrial así como de otros puntos dentro de la misma, el proceso de la información obtenida, el almacenamiento de la misma para poder obtener datos históricos y estadísticas que se usarán para la toma de decisiones en la empresa con respecto a su consumo y como afecta a los costos operativos, mejor aprovechamiento de la energía, creación de planes de mantenimiento e inversión de la maquinaria que usa esta energía ya que con el tiempo y el uso continuo sufre un desgaste más grande y diferente de la maquinaria que se usa por horas o en una oficina. Finalmente la utilización de una interfase en un explorador de Internet que permitirá tener un ahorro en licencias de uso de software de cliente, que sea multiplataforma y permita realizar un monitoreo en forma remota, la capacitación de los usuarios sea más económica y sencilla.

CAPÍTULO 2

MARCO TEÓRICO

2.1. Energía Eléctrica

2.1.1. Sistemas Trifásicos

Los generadores trifásicos contienen tres frentes sinusoidales de tensión de igual frecuencia, pero desfasados 120° unos con otros. Esto se realiza ubicando tres bobinas separadas 120° eléctricos en un mismo rotor. Por lo general, las amplitudes de las tres fases también son iguales. En este caso se dice que el generador está equilibrado. Al girar en el sentido contrario a las agujas del reloj, los laterales A, B y C de las bobinas pasan bajo los polos en el orden A-B-C-A-B-C... La polaridad de la tensión se invierte en cada cambio de polo. Suponiendo que la forma del polo y la correspondiente densidad de flujo magnético son tales, que las tensiones inducidas son sinusoidales, el resultado en las tres bobinas es el que se ha representado en la Figura 2.1 donde las tres bobinas están igualmente distribuidas alrededor de la circunferencia del rotor, es decir, las bobinas están desplazadas unas de otras en 120° mecánicos.

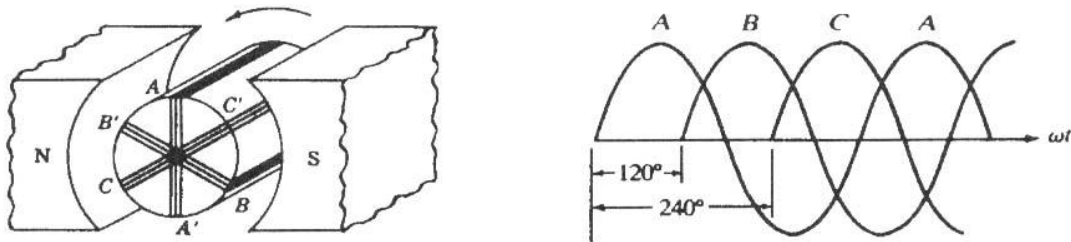


Figura 2.1. Generador trifásico con sus correspondientes tensiones inducidas

La tensión B está 120° eléctricos retrasada respecto a la A, y la C retrasa 240° . Esta distribución se conoce como secuencia ABC. Cambiando el sentido de giro se obtendría A-C-B-A-C-B, que se denomina secuencia BCA. Es importante distinguir la secuencia de un sistema trifásico porque una confusión en la determinación de la secuencia provocaría que se realicen cálculos erróneos.

Las tensiones equilibradas de la secuencia ABC, en los dominios fasorial y temporal, se dan a continuación: las tres primeras corresponden al fasorial y las tres siguientes al temporal.

$$v_{an}(t) = (V_p \sqrt{2}) \cos \omega t$$

$$v_{bn}(t) = (V_p \sqrt{2}) \cos(\omega t - 120^\circ)$$

$$v_{cn}(t) = (V_p \sqrt{2}) \cos(\omega t - 240^\circ)$$

$$V_{an} = V_p \angle 0^\circ$$

$$V_{bn} = V_p \angle -120^\circ$$

$$V_{cn} = V_p \angle -240^\circ$$

El diagrama fasorial de las tensiones se ha representado en la Figura 2.2.

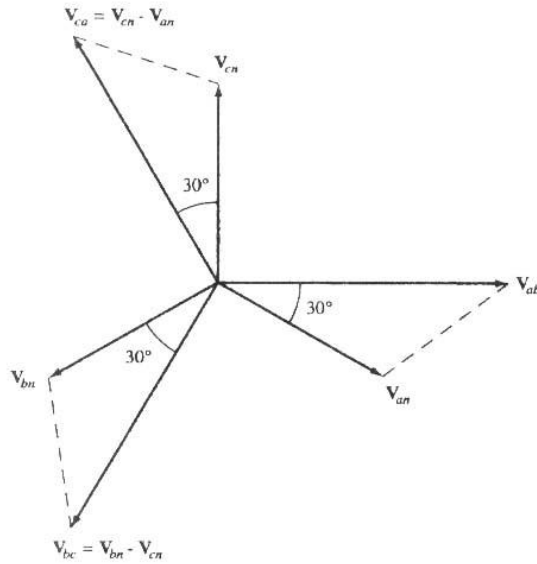


Figura 2.2. Diagrama fasorial de las tensiones de la secuencia ABC

2.1.2. Fasores de las Tensiones

La selección del ángulo de fase de una de las tensiones en el sistema trifásico fija los ángulos de todas las demás. Esto es equivalente a fijar el instante inicial $t = 0$ en el eje horizontal de la Figura 2.2, lo que se puede hacer en forma arbitraria. En el presente trabajo no es necesario escoger ningún ángulo de fase, ya que esto no aporta ningún beneficio. Las ondas son obtenidas directamente o bien son calculadas a partir de una onda desfasándola 120° en adelante o en retraso y por lo mismo solamente es necesario conocer la secuencia del sistema.

La tensión entre fases V_L es $\sqrt{3}$ veces la tensión entre cada fase y el neutro. Todas las tensiones de la secuencia ABC y BCA respectivamente, se han representado en la Figura 2.3. Esos fasores de tensión reflejan los valores máximos. En el sistema trifásico a cuatro hilos y 480 V, de uso general para cargas industriales, y en el sistema de 208 V, común en edificios comerciales, los valores citados son los eficaces.

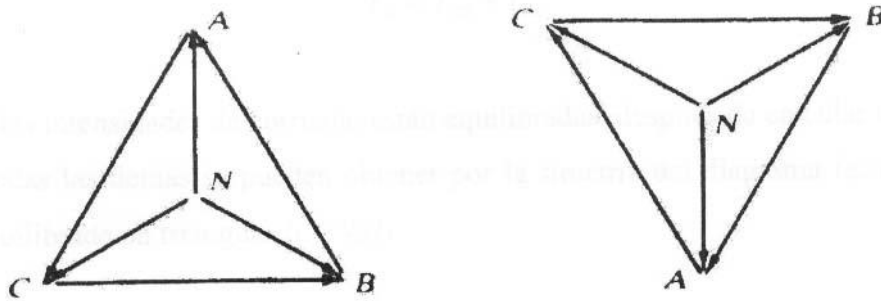


Figura 2.3. Tensiones de la secuencia ABC y BCA respectivamente

2.1.3. Circuito Monofásico Equivalente de Cargas Trifásicas Equilibradas

La figura 2.4 (a) muestra una carga equilibrada conectada en Y. En muchos casos, por ejemplo en cálculos de potencia, solamente se necesita el módulo I_L de las tres intensidades de línea. Esta se puede obtener a partir del circuito monofásico equivalente, figura 2.4 (b) que representa una fase del sistema original, en la que se da arbitrariamente a la tensión fase- neutro ángulo de fase cero. El método es aplicable a cargas conectadas en triángulo si la carga se sustituye por su equivalente en estrella.

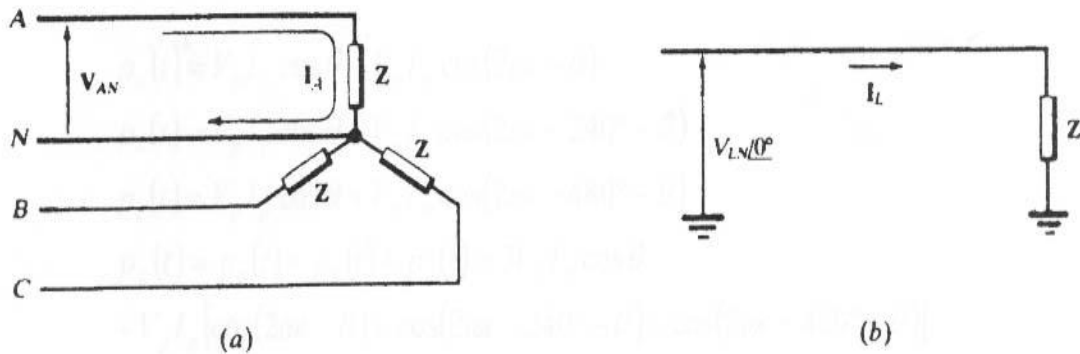


Figura 2.4. Carga equilibrada conectada en Y con circuito monofásico equivalente

En la mayor parte de los sistemas se realiza una medición monofásica debido a que dichos sistemas son equilibrados y por lo tanto se requiere solamente de una onda de corriente.

2.1.4. Potencia Trifásica

Las potencias suministradas por las tres fases de un generador equilibrado a tres impedancias idénticas con argumento θ son:

$$\begin{aligned}
 p_a(t) &= V_p I_p \cos \theta + V_p I_p \cos(2\omega t - \theta) \\
 p_b(t) &= V_p I_p \cos \theta + V_p I_p \cos(2\omega t - 240^\circ - \theta) \\
 p_c(t) &= V_p I_p \cos \theta + V_p I_p \cos(2\omega t - 480^\circ - \theta) \\
 p_T(t) &= p_a(t) + p_b(t) + p_c(t) = 3V_p I_p \cos \theta \\
 &+ V_p I_p [\cos(2\omega t - \theta) + \cos(2\omega t - 240^\circ - \theta) + \cos(2\omega t - 480^\circ - \theta)]
 \end{aligned}$$

Pero $\cos(2\omega t - \theta) + \cos(2\omega t - 240^\circ - \theta) + \cos(2\omega t - 480^\circ - \theta) = 0$ para cualquier t .
Entonces,

$$p_T(t) = 3V_p I_p \cos \theta = P$$

La potencia instantánea es igual a la potencia media total. Se puede escribir en función de la tensión de línea V_L y de la intensidad de línea I_L , quedando

En el sistema en triángulo, $V_L = V_f$ e $I_L = \sqrt{3}I_p$. Entonces, $P = \sqrt{3} V_L I_L \cos \theta$.

En el sistema en estrella, $V_L = \sqrt{3}V_f$ e $I_L = I_p$. Entonces, $P = \sqrt{3} V_L I_L \cos \theta$.

La expresión $\sqrt{3} V_L I_L \cos\theta$ da la potencia en un sistema trifásico equilibrado, independientemente del sistema de conexión. El factor de potencia del sistema trifásico es $\cos\theta$. La tensión de línea V_L en los sistemas industriales es siempre conocida. Si la carga está equilibrada, la potencia total se puede calcular a partir de la intensidad de línea y el factor de potencia.

En resumen, potencia activa, reactiva, aparente y factor de potencia en un sistema trifásico son:

$$P = \sqrt{3} V_L I_L \cos \theta \quad Q = \sqrt{3} V_L I_L \sin \theta \quad S = \sqrt{3} V_L I_L \quad \text{fp} = \frac{P}{S}$$

Todas las tensiones y las intensidades de corriente indicadas son valores eficaces.

2.1.5. Medida de Potencia y Método de los Dos Vatímetros

Un vatímetro de CA tiene una bobina de tensión y otra de corriente y responde al producto de la tensión eficaz, la corriente eficaz y el coseno del desfase entre ambos fasores. Por tanto, en la Figura 2.5 el vatímetro indicará la potencia activa suministrada al circuito pasivo.

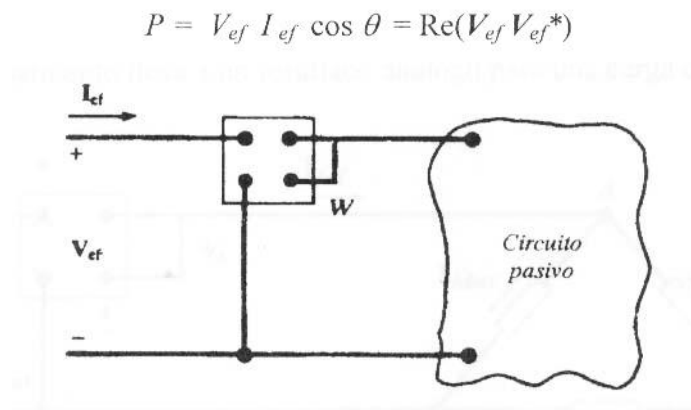


Figura 2.5. Vatímetro midiendo la potencia activa suministrado al circuito pasivo

Dos vatímetros conectados en dos de las líneas de un sistema trifásico a tres hilos indicarán correctamente la potencia trifásica total mediante la suma de las dos lecturas. Un vatímetro intentará medir negativo si el desfase entre la tensión y la corriente es mayor de

90°. En este caso, se deben invertir las conexiones de la bobina de intensidad y la medida realizada se debe tratar como negativa en la suma. En el caso del presente trabajo se obtuvieron ondas que tenían desfase mayores a 90° por lo que fue necesario revisar que los transformadores de corriente estuvieran bien conectados determinándose en la mayoría de los casos que presentaban este problema que los transformadores de corriente se encontraban conectados al revés. En la Figura 2.6 los vatímetros están instalados en las líneas A y C, con las bobinas de tensión conectadas en la línea B. Sus lecturas serán:

$$W_A = \text{Re}(V_{AB\text{ef}} I_{A\text{ef}}^*) = \text{Re}(V_{AB\text{ef}} I_{AB\text{ef}}^*) + \text{Re}(V_{AB\text{ef}} I_{AC\text{ef}}^*)$$

$$W_C = \text{Re}(V_{CB\text{ef}} I_{C\text{ef}}^*) = \text{Re}(V_{CB\text{ef}} I_{CA\text{ef}}^*) + \text{Re}(V_{CB\text{ef}} I_{CB\text{ef}}^*)$$

en las que se han usado las leyes de Kirchhoff para las corrientes $I_A = I_{AB} + I_{AC}$ e $I_C = I_{CA} + I_{CB}$, con el fin de sustituir las intensidades de línea por las de fase. El primer sumando en W_A es P_{AB} , la potencia activa que consume la carga entre las fases AB; asimismo, el segundo sumando en W_C es P_{CB} . Sumando ambas ecuaciones y combinando los sumandos intermedios se llega a

$$W_A + W_C = P_{AB} + \text{Re}[(V_{AB\text{ef}} - V_{CB\text{ef}}) I_{AC\text{ef}}^*] + P_{CB} = P_{AB} + P_{AC} + P_{CB}$$

ya que según, la ley de Kirchhoff para las tensiones, $V_{AB} - V_{CB} = V_{AC}$.

El mismo razonamiento lleva a un resultado análogo para una carga conectada en Y.

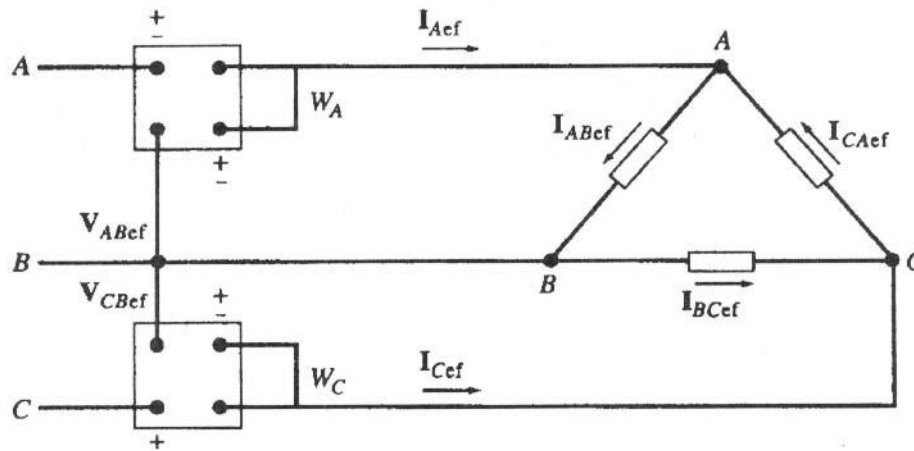


Figura 2.6. Medida de potencia utilizando dos vatímetros

El algoritmo de medición de potencia trifásica utilizado en el presente proyecto, se basa en el método de los dos vatímetros y en el cálculo de la potencia instantánea. Lo que se hace es multiplicar punto a punto los valores de corriente y de voltaje en un ciclo de onda de manera de tener un arreglo de valores instantáneos de potencia, los mismos que

sirven para obtener el valor de la potencia, sacando el valor medio de los mismos. Para obtener la potencia media se utilizan las siguientes fórmulas:

$$p(t) = v(t)i(t)$$

$$p(t) = vab(t)ia(t) + vcb(t)ic(t)$$

$$P = med(p(t))$$

2.2. Aplicaciones Distribuidas

2.2.1. Conceptos de Aplicaciones Distribuidas

2.2.1.1. Arquitectura Centralizada

- Múltiples terminales accediendo como esclavos
- Servidor dimensionado para soportar todo
- Impresoras o Pc's emulando terminal
- Un servidor central con datos y procesos
- Los puntos remotos también se conectan a través de comunicaciones como terminal

Ventajas

- Alta disponibilidad
- Control total del computador central
- Concentración de personal técnico

Desventajas

- Precio alto del servidor
- Alta dependencia de comunicaciones
- Pésimas interfaces
- Arquitecturas propietarias

2.2.1.2. Arquitectura Distribuida

- Interconecta a lugares con recursos para recibir, enviar y procesar datos
- Equipos en varias marcas
- Cada nodo puede tener sistema operativo distinto pero uno los enlaza
- El usuario no conoce la ubicación de los recursos
- Multiproceso
- Uso compartido de recursos
- Redundancia de recursos ante fallas
- El usuario trabaja en una terminal local inteligente
- Los recursos que no estén disponibles se los toma de otro lado

- Interconexión a través de redes
- Información estratégica en el computador central
- Otra información menos importante se distribuye

Ventajas

- Buen tiempo de respuesta
- Tecnología en todos los departamentos
- Entornos gráficos y amigables

Desventajas

- Flujo alto de información en la red (relativo)
- Si no hay integración puede existir inconsistencia en los datos

2.2.2. Cliente - Servidor

Cliente servidor es un modelo de desarrollo de sistema donde las transacciones se dividen en procesos que cooperan entre si.

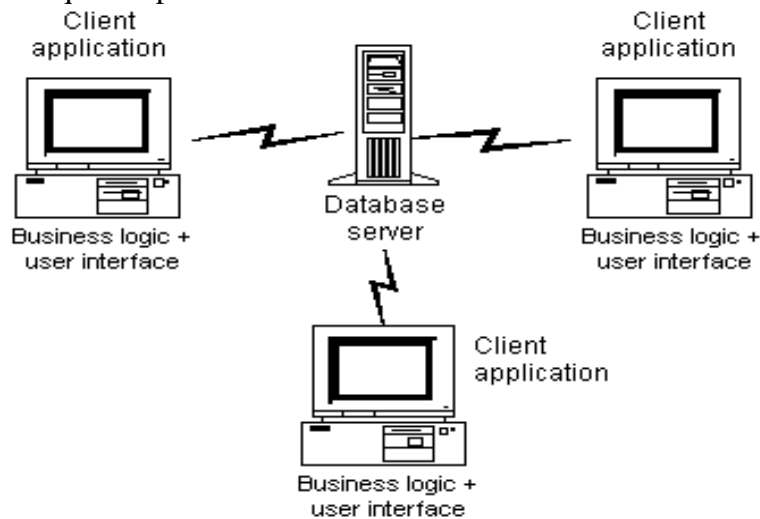


Figura 2.7. Esquema de aplicaciones cliente - servidor

2.2.2.1. Características

- Los elementos son cliente, servidor y comunicaciones
- El cliente : Presenta interface, captura datos, valida, genera consultas e informes
- El servidor: Devuelve resultados, maneja interbloqueos, recuperación a fallas, integridad, etc
- Ejemplos de servidores: impresión, archivos, bases de datos, correo, etc
- 2 o mas procesos separados trabajan juntos para cumplir una tarea

- El proceso cliente solicita al proceso servidor la ejecución de alguna acción
- Los procesos pueden o no estar en una máquina física
- La respuesta ante una solicitud solo está a cargo del servidor de base de datos y no del cliente
- El servidor devuelve solo la información solicitada
- Se asegura la integridad de los datos
- Front end en lenguaje de cuarta generación (4GL) (PowerBuilder, Oracle, Visual Basic, Visual C++, Visual Fox, Java, etc)
- Back end, cualquier base de datos (Sybase, Oracle, SQL Server, Infomix, etc)
- Conexión ODBC, OLEDB (ADO), JDBC
- En cliente solo conoce la interface para comunicación con el servidor
- Uso de procedimientos almacenados
- Plataformas diferentes

2.2.2.2. Razones para Implementar Cliente Servidor

- Cambios estructurales y organizativos
- Respuesta a la dinámica del mercado.
- Cambios en los procesos de negocio.
- Necesidad de sistemas fáciles de usar
- La relación precio - rendimiento de las estaciones de trabajo y de los servidores.
- Presencia en el mercado de herramientas mas eficientes para toma de decisiones

2.2.2.3. Ventajas de Cliente Servidor

- Aumento de la productividad.- facilidades al usuario, interfaces amigables, independencia entre front-end y back-end, apertura a sistemas abiertos, orientación a objetos
- Menores costos de operación.- hardware barato, recursos compartidos, integración de varios fabricantes, ciertas funciones de servidor van a máquinas baratas
- Mejora del rendimiento de la red.- menor tráfico, control de ordenadores aislados, uso de distintos medios de comunicación

2.2.2.4. Desventajas de Cliente Servidor

- Falta de control centralizado
 - Componentes no centralizados
 - Lógica de negocio en el cliente
 - Difícil mantenimiento
 - Cambios triviales complejos

- Diversas tecnologías
- Falta de seguridad
 - Difícil control en acceso a datos
 - Clientes accedendo a algoritmos sensibles
- Sobrecarga en el cliente
 - La lógica del negocio está presente en el cliente
 - Costos de hardware más altos en el cliente

2.3. Descripción de Metodología

2.3.1 Metodologías

2.3.1.1. Conceptos de Orientación a Objetos

Significa que el sistema se organiza como una colección de objetos que interactúan entre sí y que contienen tanto estructuras de datos como un comportamiento. Esto se opone a la programación convencional, en la cual las estructuras de datos y el comportamiento solamente están relacionadas de forma débil, ya que estos se enfocan principalmente a las funciones.

Objeto.- Los objetos son las cosas físicas y conceptuales que se encuentran en el universo alrededor de nosotros. Hardware, software, documentos, seres humanos, son todos ejemplos de objetos.

Características de los Objetos

Identidad. Los datos están cuantificados en entidades discretas y distinguibles denominadas objetos. Ejemplo: una televisión, una bicicleta, un árbol. Los objetos pueden ser concretos, como un archivo en un sistema de archivos, o bien conceptuales como la política de planificación en un sistema operativo con multiprocesos. Cada objeto posee su propia identidad inherente. En otras palabras: dos objetos serán distintos aun cuando los valores de todos sus atributos (tales como el nombre y el tamaño) sean idénticos.

Clasificación. Significa que los objetos con la misma estructura de datos (atributos) y comportamiento (operaciones) se reúnen para formar una clase. La selección de clases es arbitraria y depende de la aplicación.

- Objetos: Bicicleta de montaña, Bicicleta de carreras, Bicicleta de niños
- Clase: Bicicleta
- Atributos: Tamaño del cuadro, tamaño de rueda, material, marca, velocidad
- Operaciones: mover, reparar, cambiar velocidad
- Objetos: Triángulo, Cuadrado, Octágono
- Clase: Polígonos
- Atributos: vértices, color del borde, color del interior
- Operaciones: dibujar, borrar, mover

Polimorfismo. Significa que una misma operación puede comportarse de modos distintos en distintas clases. La operación mover por ejemplo, se puede comportar de modo distinto en las clases Ventana y Pieza de ajedrez. Una operación es una acción o una transformación que se lleva a cabo o que se aplica a un objeto. Justificar a la derecha, visualizar y mover son ejemplos de operaciones. Una implementación específica de una operación por parte de una cierta clase es lo que se denomina método. Dado que los operadores orientados a objetos son polimórficos es posible que haya más de un método que lo implemente.

En el mundo real una operación es simplemente, una abstracción de comportamiento análogo entre distintas clases de objetos. Cada objeto "sabe" llevar a cabo sus propias operaciones. Sin embargo, en un lenguaje orientado a objetos es este el que selecciona automáticamente el método correcto para implementar una operación basándose en el nombre de la operación y en la clase del objeto que está siendo afectado. El usuario de una operación no necesita ser consciente del número de métodos que existen para implementar una cierta operación polimórfica. Se pueden añadir nuevas clases sin modificar el código existente, siempre y cuando se proporcionen métodos para todas las operaciones aplicables a las nuevas clases.

Herencia.- Es compartir atributos y operaciones entre clases tomando como base una relación jerárquica. En términos generales se puede definir una clase que después se irá refinando sucesivamente para producir subclases. Todas las subclases poseen o heredan, todas y cada una de las propiedades de su superclase y añaden, además, sus propiedades exclusivas. No es necesario repetir las propiedades de las superclases en cada subclase. Por ejemplo: Ventana de desplazamiento y ventana fija son subclases de ventana. Ambas subclases heredan las propiedades de ventana tales como una región visible de la pantalla. La ventana de desplazamiento añade una barra de desplazamiento y un ascensor. La capacidad de sacar factor común a las propiedades de varias clases en una superclase común y de heredar las propiedades de la superclase puede reducir muchísimo la repetición en el diseño y en los programas siendo una de las ventajas principales de un sistema orientado a objetos.

2.3.1.2. Qué es una Metodología

Conjunto de métodos empleados para el desarrollo de sistemas automatizados. Una metodología completa es algo más que una notación, un proceso, y herramientas. Además de una "notación, de un proceso, y de herramientas," estas "metodologías completas" proporcionan:

- Guías para estimar costos,
- Manejo del proyecto en las tareas y entregas,
- Medidas y métricas,
- Formas definidas y dirección en las entregas de la construcción,
- Políticas y procedimientos para garantizar la calidad del software,
- Descripciones de los roles y programas de entrenamiento detallados,
- Ejemplos totalmente trabajados,
- Ejercicios de entrenamiento,

- Técnicas para adaptar el método, y
- Técnicas definidas

2.3.2. Descripción de la Metodología a Utilizar

La metodología a ser utilizada en este proyecto será la definida por James Rumbaugh, denominada Object Modeling Technique (OMT), la misma que será descrita a continuación.

2.3.2.1. Introducción

Existen muchas aproximaciones de desarrollo de software que utilizan modelos orientado a objetos, pero que no tienen todos los soportes para desarrollo de aplicaciones de base de datos. Algunas aproximaciones carecen de suficientes abstracciones y tienen un bajo relacionamiento para detalles de implementación.

Otros métodos de programación orientados a objetos, ponen un escaso énfasis en la estructura de datos y constantes, que son muy importantes para aplicaciones de base de datos.

OMT pone énfasis en la importancia del modelo y uso de modelo para lograr una abstracción, en el cual el análisis está enfocado en el mundo real para un nivel de diseño, también pone detalles particulares para modelado de recursos de la computadora. Esta Tecnología puede ser aplicada en varios aspectos de implementación incluyendo archivos, base de datos relacionales, base de datos orientados a objetos. OMT está construido alrededor de descripciones de estructura de datos, constantes, sistemas para procesos de transacciones.

Desde que la comunidad de programación orientada a objetos tuvo la noción de incorporar el pensamiento de que los objetos son entidades coherentes con identidad, estado y conducta, estos objetos pueden ser organizados por sus similitudes y sus diferencias, puestas en uso en herencia y polimorfismo.

Desde el modelado de información, tuvo que ser adoptada la noción de entidades que son conectadas con entidad relación, los modelos de relación son declarativos, imperativos.

OMT pone énfasis en especificaciones declarativas de la información, para capturar limpiamente los requerimientos, especificaciones imperativas para poder descender prematuramente en el diseño, declaraciones que permiten optimizar los estados, además provee un soporte declarativo para una directa implementación de DBMS.

2.3.2.2. Proceso de Desarrollo de OMT

Los pasos para desarrollar diseño OMT son:

Conceptualización.- El desarrollo empieza con el análisis de la empresa o negocio, o de cómo los usuarios conciben el sistema y formulan sus requerimientos.

La conceptualización es a menudo una observación crítica de los procesos de la empresa, y su impacto económico.

En esta etapa se debe tener en cuenta las siguientes preguntas:

- ¿Cuál es la aplicación?
- ¿Que problemas tendrán que ser resueltos?
- ¿Dónde será usado el sistema?
- ¿Cuándo será requerido el sistema?
- ¿Para qué es necesario el sistema?

2.3.2.3. Análisis

Los requerimientos formados durante la conceptualización son revisados y analizados para la construcción del modelo real. La meta del análisis es especificar las necesidades que deben ser satisfechas.

Pueden existir diversas fuentes de información que pueden servir para el análisis, puede existir un lenguaje formal para describir el problema. Algunas veces los expertos del dominio pueden proveer escenarios, storyboards y casos de uso para un nuevo sistema.

Aquí es donde se determina el modelo de objeto, se hace una tentativa de clases eliminar las clases irrelevantes, las posibles asociaciones entre las clases, luego se hace la refinación de asociaciones eliminando las redundantes o las que no tienen relevancia, posteriormente se hace una tentativa de atributos de objetos y enlaces.

Una vez obtenido los objetos del sistema se hace un refinamiento del modelo, posteriormente se busca un nivel de abstracción para modelar subsistemas, para buscar un sistema tangible y sólido.

Una vez desarrollado el modelo se introduce la noción de transacción, es una forma de modelar procesos o describir cambio de datos, movimiento de datos; una vez definido el movimiento de datos se define un diccionario de datos de todas las entidades modeladas.

2.3.2.4. Diseño del Sistema

El diseño tiene un alto nivel estratégico y decisión para resolver los problemas.

Los problemas grandes se deben ver desde el punto de análisis y diseño, este sistema se divide en subsistemas, a su vez este subsistema puede ser dividido en otros subsistemas de manera que puedan ser manejados y cada componente pueda ser comprensible.

En esta etapa se deben crear estrategias, formular una arquitectura para el sistema y las políticas que deben guiarla además de un detalle del diseño. Debiendo tener en cuenta los siguientes aspectos:

- Divisar una arquitectura
- Elegir una implementación para un control externo
- Si se usa base de datos elegir el paradigma de administración de base de datos
- Determinar oportunidades para la reutilización
- Elegir estrategia para interacción de datos
- Elegir una forma de identificar los objetos
- Detallar el diseño

Durante el diseño del sistema se debe hacer un cuadro de estrategias y decisiones arquitecturales, tener una idea más precisa de clases y métodos individuales. Adicionalmente se puede mejorar el modelo de diseño para mejorar la implementación.

Se deben considerar los siguientes pasos:

- Uso de transformaciones para simplificar y optimizar el modelo de objetos desde el análisis.
- Elaborar un modelo de objeto
- Elaborar un modelo funcional
- Evaluar la calidad del diseño del modelo
- Implementación

El diseño es trasladado a un lenguaje de programación actual y código de base de datos. Este paso puede ser aplicado y considerado durante el análisis y diseño para mejorar el performance del sistema.

2.3.2.5. Mantenimiento

La documentación del desarrollo y seguimiento de los modelos a través del código facilita el posterior mantenimiento

La metodología OMT soporta múltiples estilos de desarrollo. Se puede usar OMT para conseguir un alto performance en la fase de análisis y diseño e implementación con una estricta secuencia de pasos, también adopta una estrategia de desarrollo iterativa.

Primeramente se desarrolla el núcleo del sistema, se analiza, diseña, implementa, y crea el código fuente.

La idea principal de OMT es la conceptualización de una entidad que permita manejar atributos y asociaciones y sus transformaciones, transacciones, para modelar esos atributos y entidades orientado a objetos, y la interacción con la base de datos.

OMT pone énfasis en la importancia del modelo y uso de modelo para lograr una abstracción, en el cual el análisis está enfocado en el mundo real para un nivel de diseño,

también pone detalles particulares para modelado de recursos de la computadora. Esta tecnología puede ser aplicada en varios aspectos de implementación incluyendo archivos, base de datos relacionales, base de datos orientados a objetos. OMT esta construido alrededor de descripciones de estructura de datos, constantes, sistemas para procesos de transacciones.

2.4. Estudio de Seguridades en Internet e Intranet

2.4.1. El Internet

El Internet no es más que una herramienta que sabiendo como funciona se le puede sacar mucho provecho

2.4.1.1. Internet en un Principio

En 1969 el departamento de defensa de los estados unidos creó la ARPA (Agencia para Proyectos Avanzados de Investigación. El departamento de defensa aspiraba crear una red de comunicación de tal manera que si una parte de la misma sufría un colapso total, los mensajes pudieran encontrar el camino hasta su destino de cualquier manera. El resultado fue ARPAnet.

En 1983, mas que nada debido a razones pragmáticas, ARPAnet se dividió en dos sistemas diferentes llamados ARPAnet y MILENET. La primera fue puesta a disposición de los ciudadanos para uso civiles, y MILENET fue reservado para uso militar. Las redes se conectaron de tal manera que los usuarios pudieran intercambiar información; esto acabó por conocerse como Internet.

Con el paso del tiempo comenzaron a surgir otras redes como BITNET y CSNET. Al principio se trabajaba con redes totalmente independientes, usadas con propósitos educativos o de investigación, pero más adelante se conectaron con Internet para poder compartir información fácilmente entre organizaciones.

Uno de los avances más importantes de Internet tuvo lugar en 1986, cuando NFS (Fundación Nacional de la Ciencia) de los Estados Unidos creo NSFNET con el propósito de conectar varias supercomputadoras de gran velocidad a lo largo del país, principalmente con fines de investigación. ARPAnet fue desmantelada y NSFNET se convirtió en el principal conducto de Internet.

2.4.1.2. Internet en la Actualidad

Internet no es más que una red de grandes servidores en configuración de cliente servidor, quiero decir con esto que se envía una petición al servidor que es respondida por éste y son vistos e interpretados en la propia interfase cliente. Por lo que Internet es una red WAN (world area network)

Ya a estas alturas todo el mundo ha oído o sabe navegar en Internet como comúnmente se le dice. Este es tan variado y ofrece tantos servicios que la comunicación no es ni será la misma nunca jamás gracias a este servicio. Uno de los ejemplos más comunes es el del E-mail o correo electrónico. Una de las redes más grandes del mundo es la Microsoft.

La razón es simple, las mayorías de computadoras personales utilizan su sistema operativo, Windows.

2.4.1.3. Internet en el Futuro

Aunque ya en el presente el comercio por Internet es algo común se esperará más de éste en el futuro al igual que las videoconferencias, música, juegos, educación, televisión, imágenes, radio y muchas cosas más.

2.4.2. ¿CÓMO FUNCIONA INTERNET?

Todas las computadoras ya poseen módem (de no tenerlo no es problema ni costoso instalarle uno) y en todos los países hay servicio a Internet por lo tanto no hay que ser un experto para tener acceso a Internet sino recursos, que es lamentable aunque justo ya que todo servicio de debe pagar.

Cada equipo de cómputo que participa en Internet recibe el nombre de computadora host. Algunos host sirven el contenido o aplicaciones por lo que se les denomina servidores. Otras computadoras como la que estoy utilizando yo (clientes) consumen el contenido o la información ofrecida por los servidores. A esta relación de le denomina cliente servidor.

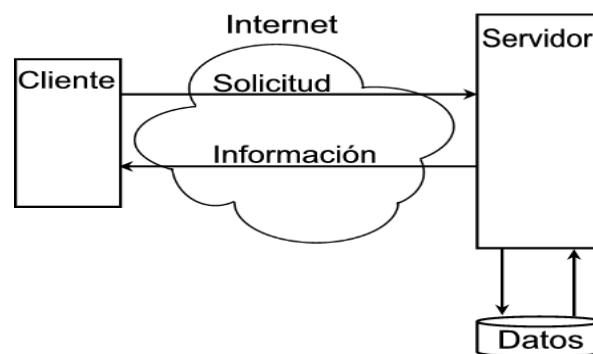


Figura 2.8. Esquema de una aplicación en internet

La misma tecnología funciona con respecto a programas que se ejecutan en computadoras conectadas a Internet. Cada programa servidor necesita un programa cliente que hable el mismo lenguaje.

Cada computadora necesita una configuración correcta para conectar a Internet, esto es un protocolo, un cliente y un adaptador. En este caso el cliente estaría dependiendo de

las personas que vayan a utilizar el equipo, el adaptador del dispositivo que esté utilizando en su computadora y el protocolo que será el más usado, el TCP/IP.

El protocolo TCP/IP no es más que un protocolo para poder comunicarse con otras computadoras. Este protocolo norma la manera en que la computadora host se comunica al definir como se empaquetan los datos y como se encuentra el camino hasta la computadora remota. Este protocolo esta formado por dos protocolos diferentes, unidos y acoplados. Cada persona tiene una contraseña dada por el proveedor de Internet que es única también.

IP: Computadoras especiales llamadas enrutadoras, usan un protocolo de Internet para mover bits de información a través de Internet. Cada paquete de información cuenta con la dirección IP tanto de la computadora que lo envió como de la que recibe el paquete. Una dirección IP es un número de identificación único de la computadora tal como es reconocida por las demás computadoras en Internet. Las direcciones IP constan de cuatro números separados por puntos como muestra el cuadro de abajo.

Pero estos números no hay que ponerlos obligatoriamente porque inmediatamente se conecta a Internet se le asigna uno automáticamente a su computadora.

TCP: El protocolo de control de transmisión define la manera en que la información será separada en paquetes y enviada a través de Internet, se asegura también de que cada paquete se recombine el orden correcto y también los revisa para localizar errores. Como en cada computadora que participa en Internet se le asigna una dirección IP una persona ordinaria no podría recordar tantos números de cada computadora, entonces por lo que hay una manera más fácil de hacerlo. Por el DOMINIO. Que es el nombre para un host determinado. Como por ejemplo la dirección IP de Yahoo (64. 128. 48. 24), pero su nombre dominio es yahoo.com. Lo cual también tiene una estructura URL (Localizador Uniforme de Recursos) que es el método estándar que emplea para especificar la ubicación de los recursos de Internet.

Los tres primeros números indican la red a la que pertenece el ordenador, y el último sirve para diferenciar el ordenador de los otros que "cuelguen" de la misma red. Los nombres de dominio consisten en dos o más palabras separadas por puntos. WWW (World Wide Web) es una recopilación masiva de documentos estáticos e interactivos vinculados entre sí. Se utiliza un navegador Web para visualizar esas páginas Web los cuales se encuentran en cientos de miles de servidores alrededor del mundo.

En el caso de la figura de arriba se esta usando un navegador de Windows, el Internet Explorer versión 5.0. Este explorador es muy común en las computadoras por la razón de que éste viene con el paquete de Windows. De ahí el problema de monopolio que tiene esta empresa multimillonaria.

Los dominios del primer nivel son más específicos como COM, NET, EDU, lo cual indica el tipo de organización que es incluso el país donde se encuentra. Se muestran unos ejemplos.

Tabla 2.1. Tipos de extensiones de sitios WEB en Internet

Nombre	Descripción
Com	Organizaciones comerciales y con fines de lucro
Net	Organizaciones diversas y sin fines de lucro
Edu	Universidad que imparten carreras de cuarto años
Gov	Agencias del gobierno federal
Do	Republica dominicana
Uk	Reino unido

Los servidores de nombre de dominio traducen estos nombres en direcciones IP. El concepto de proveedor de servicios de Internet refiere una empresa que proporciona conectividad con Internet. La labor de estas empresas es de mantener una gran red que se conecta directamente con Internet. La computadora que utiliza la gente común como usted y yo establece una conexión de red con el proveedor de servicios a través de un protocolo de conexión y de un módem. A esto se le llama conexión PPP.

La diferencia entre una conexión directa con una PPP es que la PPP es más lenta que la directa y también que es temporal o hasta que el cliente siga pagando el servicio. Todas las páginas de Internet tienen un formato, este es el HTML que es un lenguaje que especifica como se ve un documento en Internet. Es conocido como lenguaje de etiquetas o de marcado de hipertexto. Esto quiere decir que las páginas de Internet se pueden modificar. Algunas sí pero estas tienen derecho de autor como cualquier libro por lo tanto es contra la ley hacerlo. Editores como el FRONTPAGE y el COMPOSER son uno de los más usados para modificar o hacer páginas de Internet. Hay muchos otros lenguajes de programación que en combinación del HTML hacen que las páginas de Internet sean más eficientes, rápidas y llamativas. Ejemplos de estos programas son JAVA, FLASH, MACROMEDIA, etc.

2.4.3. Servicios Por Internet

2.4.3.1. Correo Electrónico (E-Mail)

Existen algunos servicios por Internet pero solo se va a explicar este servicio puesto que está relacionado con la aplicación a realizarse. Es tal vez el principal servicio de Internet, y sin duda el de mayor importancia histórica. Cada persona que está conectada cuenta con un "buzón electrónico" personal, simbolizado en una dirección de correo.

El correo electrónico sirve para enviar y recibir mensajes a otros usuarios, y por eso no hay nunca dos nombres iguales. La primera parte de una dirección identifica habitualmente a la persona y la segunda a la empresa u organización para la que trabaja, o al proveedor de Internet a través del que recibe la información.

Por este medio se pueden enviar texto, gráficos, hojas de cálculo, algunos programas ejecutables (dependiendo de la política del proveedor y del espacio que este le dé para el correo), etc.

Este medio de comunicación es ya muy común en la mayoría de las empresas por no decir todas y usado por la mayoría de las personas.

La mayoría de los BROWSER o buscadores como YAHOO, INFOSEEK, ALTAVISTA, etc., ofrecen servicios de correo gratuito y estos pueden durar mientras usted quiera. Hay personas que no tiene la facilidad del servicio de Internet y tienen el correo electrónico el cual accesa desde la universidad, desde la PC de un amigo, etc. También estos browser ofrecen chat, messenger, tarjetas electrónicas, compras, noticias y un sin fin de servicios e información.

Cada vez que se accede a una página de Internet la computadora guarda información del browser o navegador Web con el fin de restaurar sus preferencias personales la próxima vez que se visite ese sitio. A esa pieza de información se le llama COOKIES.

2.4.4. Seguridades En Internet

2.4.4.1. Secure Socket Layer (SSL)

El protocolo SSL es un sistema diseñado y propuesto por Netscape Communications Corporation. Se encuentra en la pila OSI entre los niveles de TCP/IP y de los protocolos [HTTP](#), FTP, SMTP, etc. Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, típicamente el RC4 o IDEA, y cifrando la clave de sesión de RC4 o IDEA mediante un algoritmo de cifrado de clave pública, típicamente el RSA. La clave de sesión es la que se utiliza para cifrar los datos que vienen del y van al servidor seguro. Se genera una clave de sesión distinta para cada transacción, lo cual permite que aunque sea reventada por un atacante en una transacción dada, no sirva para descifrar futuras transacciones. MD5 se usa como algoritmo de hash.

Proporciona cifrado de datos, autenticación de servidores, integridad de mensajes y, opcionalmente, autenticación de cliente para conexiones TCP/IP.

Cuando el cliente pide al servidor seguro una comunicación segura, el servidor abre un puerto cifrado, gestionado por un software llamado Protocolo SSL Record, situado encima de TCP. Será el software de alto nivel, Protocolo SSL Handshake, quien utilice el Protocolo SSL Record y el puerto abierto para comunicarse de forma segura con el cliente.

2.4.4.1.1. El Protocolo SSL Handshake

Durante el protocolo SSL Handshake, el cliente y el servidor intercambian una serie de mensajes para negociar las mejoras de seguridad. Este protocolo sigue las siguientes seis fases (de manera muy resumida):

La fase Hola, usada para ponerse de acuerdo sobre el conjunto de algoritmos para mantener la intimidad y para la autenticación.

La fase de intercambio de claves, en la que intercambia información sobre las claves, de modo que al final ambas partes comparten una clave maestra.

La fase de producción de clave de sesión, que será la usada para cifrar los datos intercambiados.

La fase de verificación del servidor, presente sólo cuando se usa RSA como algoritmo de intercambio de claves, y sirve para que el cliente autentique al servidor.

La fase de autenticación del cliente, en la que el servidor solicita al cliente un certificado X.509 (si es necesaria la autenticación de cliente).

Por último, la fase de fin, que indica que ya se puede comenzar la sesión segura.

2.4.4.1.2. El Protocolo SSL Record

El Protocolo SSL Record especifica la forma de encapsular los datos transmitidos y recibidos.

La porción de datos del protocolo tiene tres componentes:

- MAC-DATA, el código de autenticación del mensaje.
- ACTUAL-DATA, los datos de aplicación a transmitir.
- PADDING-DATA, los datos requeridos para rellenar el mensaje cuando se usa cifrado en bloque.

2.4.4.2. Correo Seguro

No hay nada más fácil que leer los correos de otras personas, ya que viajan desnudos por la Red, valga la siguiente analogía. Un correo electrónico normal es como una tarjeta postal sin sobre, que puede leer todo el que tenga interés. Por consiguiente, la mejor manera de preservar la intimidad en los mensajes de correo electrónico es recurrir a la criptografía. Por medio de potentes técnicas criptográficas, el contenido del mensaje puede ser enviado cifrado, permitiendo así que sólo el destinatario legítimo del correo sea capaz de leerlo. Con este mecanismo se garantiza la confidencialidad del correo.

Sin embargo, los modernos sistemas de seguridad del correo, como PGP y otros, no se limitan a cifrar el contenido de los mensajes intercambiados, sino que también añaden otros servicios, como la integridad, que garantiza que el contenido del mensaje no ha sido alterado por el camino; la autenticación, que asegura la identidad del remitente del correo, de manera que se puede estar seguros de que fue escrito por quien lo envió y no ha sido falsificado; y el no repudio, que protege frente a que posteriormente el que envió el correo (o lo recibió de nosotros) alegue posteriormente no haberlo enviado (o recibido si era el destinatario). Estos últimos servicios se prestan mediante las firmas digitales.

2.4.4.3. Firewall

Un firewall en Internet es un sistema o grupo de sistemas que impone una política de seguridad entre la organización de red privada y el Internet. El firewall determina cual de los servicios de red pueden ser accedidos dentro de esta por los que están fuera, es decir quien puede entrar para utilizar los recursos de red pertenecientes a la organización. Para que un firewall sea efectivo, todo tráfico de información a través del Internet deberá pasar a través del mismo donde podrá ser inspeccionada la información. El firewall podrá únicamente autorizar el paso del tráfico, y el mismo podrá ser inmune a la penetración de la información. Desafortunadamente, este sistema no puede ofrecer protección alguna una vez que el agresor lo traspasa o permanece en torno a este.

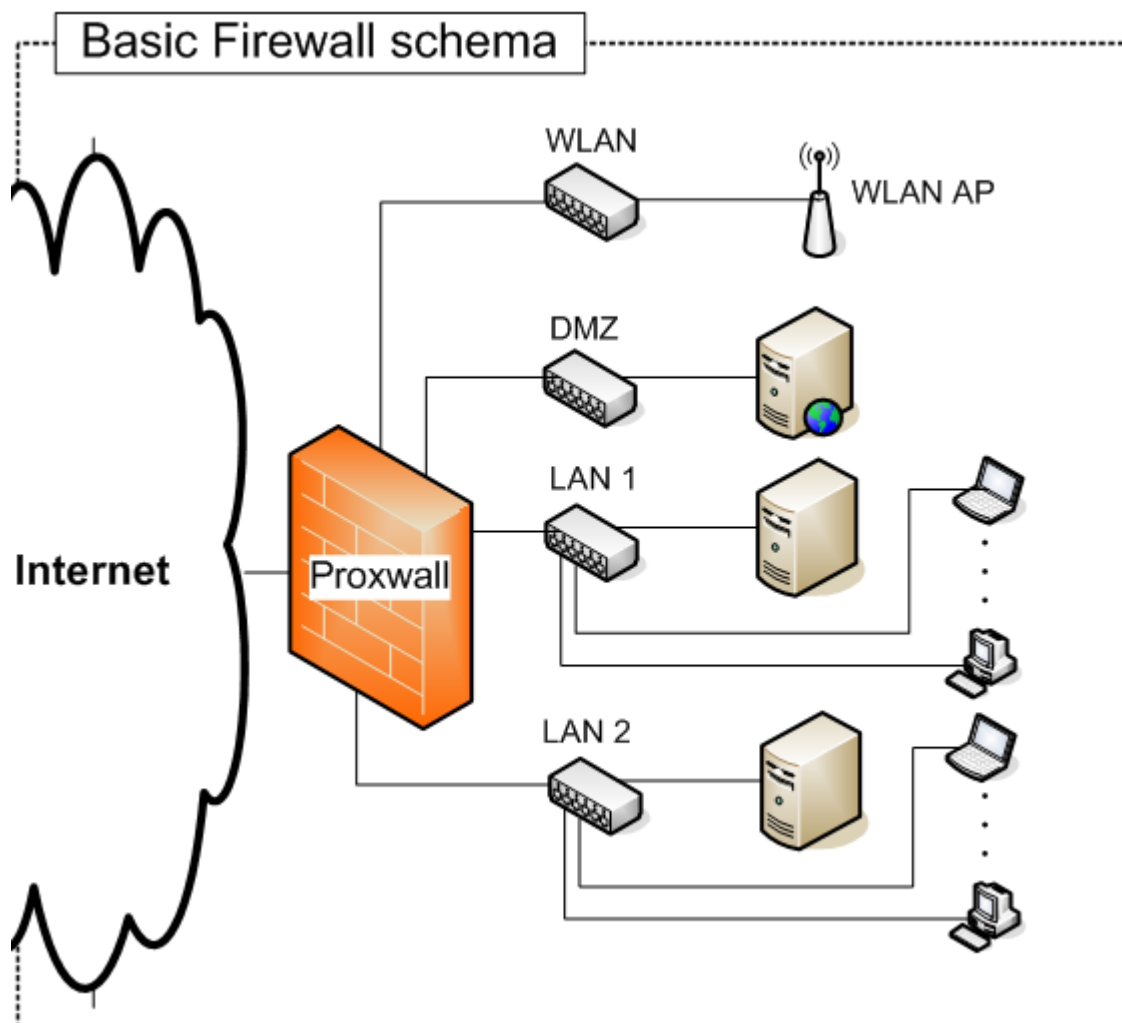


Figura 2.9. Esquema de funcionamiento de un firewall en el internet

Esto es importante, ya que se debe notar que un firewall de Internet no es justamente un ruteador, un servidor de defensa, o una combinación de elementos que proveen

seguridad para la red. El firewall es parte de una política de seguridad completa que crea un perímetro de defensa diseñada para proteger las fuentes de información. Esta política de seguridad podrá incluir publicaciones con las guías de ayuda donde se informe a los usuarios de sus responsabilidades, normas de acceso a la red, política de servicios en la red, política de autenticidad en acceso remoto o local a usuarios propios de la red, normas de dial-in y dial-out, reglas de encriptación de datos y discos, normas de protección de virus, y entrenamiento. Todos los puntos potenciales de ataque en la red podrán ser protegidos con el mismo nivel de seguridad. Un firewall de Internet sin una política de seguridad comprensiva es como poner una puerta de acero en una tienda.

2.4.4.3.1. Beneficios de un Firewall en Internet

Los firewalls en Internet administran los accesos posibles del Internet a la red privada. Sin un firewall, cada uno de los servidores propios del sistema se exponen al ataque de otros servidores en el Internet. Esto significa que la seguridad en la red privada depende de la "Dureza" con que cada uno de los servidores cuenta y es únicamente seguro tanto como la seguridad en la fragilidad posible del sistema.

El firewall permite al administrador de la red definir un "choke point" (embudo), manteniendo al margen los usuarios no-autorizados (tal como hackers, crackers, vándalos, y espías) fuera de la red, prohibiendo potencialmente la entrada o salida al vulnerar los servicios de la red, y proporcionar la protección para varios tipos de ataques posibles. Uno de los beneficios clave de un firewall en Internet es que ayuda a simplificar los trabajos de administración, una vez que se consolida la seguridad en el sistema firewall, es mejor que distribuirla en cada uno de los servidores que integran la red privada.

El firewall ofrece un punto donde la seguridad puede ser monitoreada y si aparece alguna actividad sospechosa, éste generará una alarma ante la posibilidad de que ocurra un ataque o suceda algún problema en el tránsito de los datos. Esto se podrá notar al acceder la organización al Internet, la pregunta general es "si" pero "cuando" ocurrirá el ataque. Esto es extremadamente importante para que el administrador audite y lleve una bitácora del tráfico significativo a través del firewall. También, si el administrador de la red toma el tiempo para responder una alarma y examina regularmente los registros de base. Esto es innecesario para el firewall, desde que el administrador de red desconoce si ha sido exitosamente atacado!

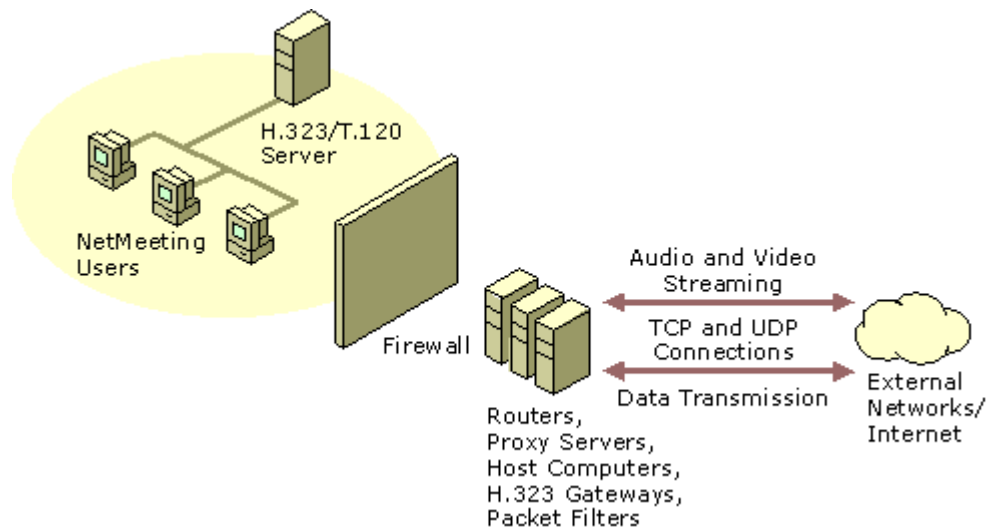


Figura 2.10. Red privada conectada a la Internet mediante un firewall

El firewall realiza las siguientes funciones:

- Concentra la seguridad
- Centraliza los accesos
- Genera alarmas de seguridad
- Traduce direcciones (NAT)
- Monitorea y registra el uso de Servicios de WWW y FTP

Con el paso de algunos años, el Internet ha experimentado una crisis en las direcciones, logrando que el direccionamiento IP sea menos generoso en los recursos que proporciona. Por este medio se organizan las compañías conectadas al Internet, debido a esto hoy no es posible obtener suficientes registros de direcciones IP para responder a la población de usuarios en demanda de los servicios. Un firewall es un lugar lógico para desplegar un Traductor de Direcciones de Red (NAT), esto puede ayudar aliviando el espacio de direccionamiento acortando y eliminando lo necesario para re-enumerar cuando la organización cambie del Proveedor de Servicios de Internet (ISP's).

Un firewall de Internet es el punto perfecto para auditar o registrar el uso del Internet. Esto permite al administrador de red justificar el gasto que implica la conexión al Internet, localizando con precisión los cuellos de botella potenciales del ancho de banda, y promueve el método de cargo a los departamentos dentro del modelo de finanzas de la organización.

Un firewall de Internet ofrece un punto de reunión para la organización. Si una de sus metas es proporcionar y entregar servicios información a consumidores, el firewall de Internet es ideal para desplegar servidores WWW y FTP.

Finalmente, el firewall puede presentar los problemas que genera un punto de falla simple. Enfatizando, si este punto de falla se presenta en la conexión al Internet, aun así la red interna de la organización puede seguir operando, únicamente el acceso al Internet está perdido.

La preocupación principal del administrador de red, son los múltiples accesos al Internet, que se pueden registrar con un monitor y un firewall en cada punto de acceso que posee la organización hacia el Internet. Estos dos puntos de acceso significan dos puntos potenciales de ataque a la red interna que tendrán que ser monitoreados regularmente.

2.4.4.3.2. Limitaciones de un Firewall

Un firewall no puede protegerse contra aquellos ataques que se efectúen fuera de su punto de operación.

Por ejemplo, si existe una conexión dial-out sin restricciones que permita entrar a la red protegida, el usuario puede hacer una conexión SLIP o PPP al Internet. Los usuarios con sentido común suelen "irritarse" cuando se requiere una autenticación adicional requerida por un Firewall Proxy Server (FPS) lo cual se puede ser provocado por un sistema de seguridad circunvecino que esta incluido en una conexión directa SLIP o PPP del ISP.

Este tipo de conexiones derivan la seguridad provista por un firewall construido cuidadosamente, creando una puerta de ataque. Los usuarios pueden estar consientes de que este tipo de conexiones no son permitidas como parte de integral de la arquitectura de la seguridad en la organización.

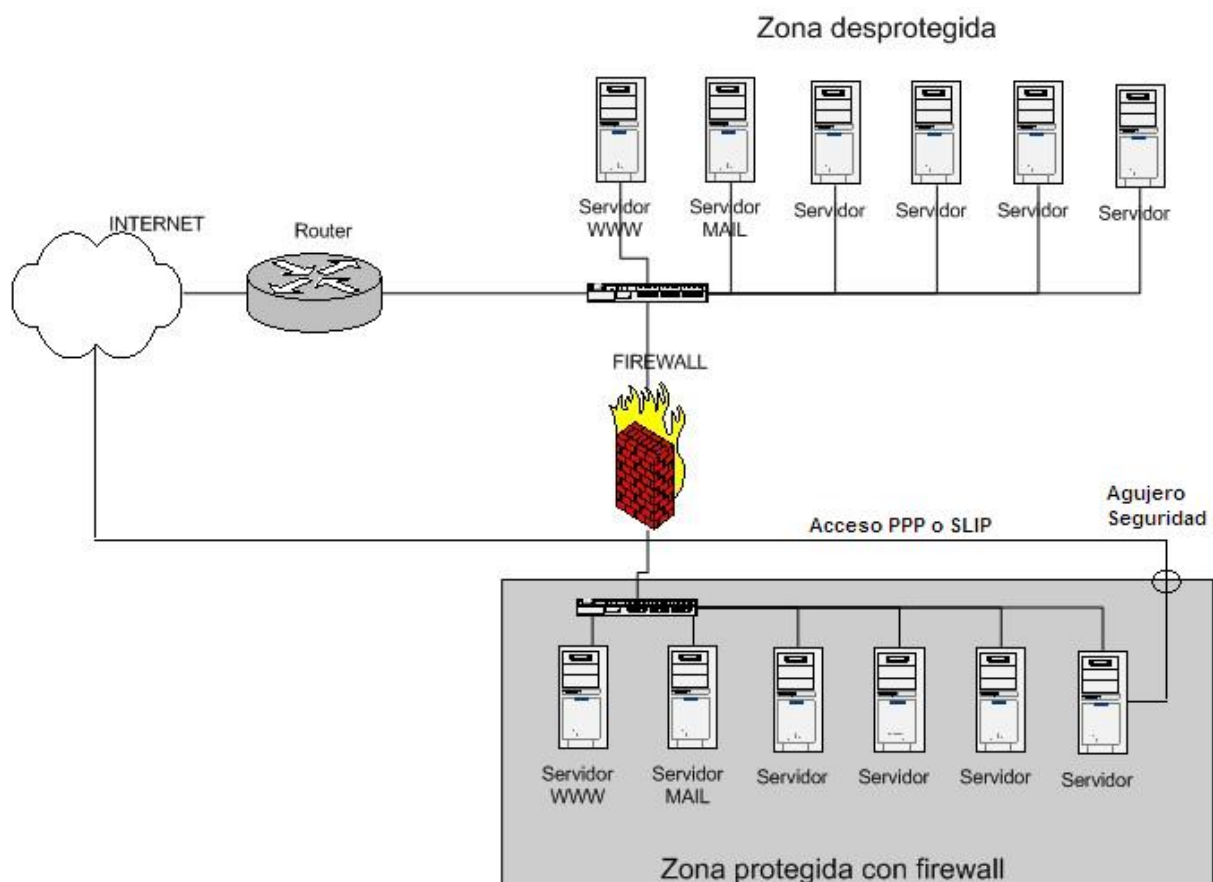


Figura 2.11. Fallas de seguridad en la red privada por mal uso de políticas

El firewall no puede protegerse de las amenazas a que esta sometido por traidores o usuarios inconscientes. El firewall no puede prohibir que los traidores o espías corporativos copien datos sensitivos en disquettes o tarjetas PCMCIA y substraigan estas del edificio.

El firewall no puede proteger contra los ataques de la "Ingeniería Social", por ejemplo un Hacker que pretende ser un supervisor o un nuevo empleado despistado, persuade al menos sofisticado de los usuarios a que le permita usar la contraseña al servidor del corporativo o que le permita el acceso "temporal" a la red.

Para controlar estas situaciones, los empleados deberían ser educados acerca de los varios tipos de ataque social que pueden suceder, y a cambiar sus contraseñas si es necesario periódicamente.

El firewall no puede protegerse contra los ataques posibles a la red interna por virus informativos a través de archivos y software. Obtenidos del Internet por sistemas operativos al momento de comprimir o descomprimir archivos binarios, el firewall de Internet no puede contar con un sistema preciso de SCAN para cada tipo de virus que se puedan presentar en los archivos que pasan a través de el.

La solución real está en que la organización debe ser consciente en instalar software anti-viral en cada despacho para protegerse de los virus que llegan por medio de disquettes o cualquier otra fuente.

Finalmente, el firewall de Internet no puede protegerse contra los ataques posibles en la transferencia de datos, estos ocurren cuando aparentente datos inocuos son enviados o copiados a un servidor interno y son ejecutados despachando un ataque.

Por ejemplo, una transferencia de datos podría causar que un servidor modificara los archivos relacionados a la seguridad haciendo mas fácil el acceso de un intruso al sistema.

Como se puede ver, el desempeño de los servidores Proxy en un servidor de defensa es un excelente medio de prohibición a las conexiones directas por agentes externos y reduce las amenazas posibles por los ataques con transferencia de datos.

2.4.4.4. Técnicas Criptográficas

No existe un único mecanismo capaz de proveer todos los servicios anteriormente citados, pero la mayoría de ellos hacen uso de técnicas criptográficas basadas en el cifrado de la información. Los más importantes son los siguientes:

Intercambio de autenticación.- corrobora que una entidad, ya sea origen o destino de la información, es la deseada, por ejemplo, A envía un número aleatorio cifrado con la clave pública de B, B lo descifra con su clave privada y se lo reenvía a A, demostrando así

que es quien pretende ser. Por supuesto, hay que ser cuidadoso a la hora de diseñar estos protocolos, ya que existen ataques para desbaratarlos.

Cifrado.- garantiza que la información no es inteligible para individuos, entidades o procesos no autorizados (confidencialidad). Consiste en transformar un texto en claro mediante un proceso de cifrado en un texto cifrado, gracias a una información secreta o clave de cifrado. Cuando se emplea la misma clave en las operaciones de cifrado y descifrado, se dice que el criptosistema es simétrico. Estos sistemas son mucho más rápidos que los de clave pública, resultando apropiados para funciones de cifrado de grandes volúmenes de datos. Se pueden dividir en dos categorías: cifradores de bloque, que cifran los datos en bloques de tamaño fijo (típicamente bloques de 64 bits), y cifradores en flujo, que trabajan sobre flujos continuos de bits. Cuando se utiliza una pareja de claves para separar los procesos de cifrado y descifrado, se dice que el criptosistema es asimétrico o de clave pública. Una clave, la privada, se mantiene secreta, mientras que la segunda clave, la pública, puede ser conocida por todos. De forma general, las claves públicas se utilizan para cifrar y las privadas, para descifrar. El sistema tiene la propiedad de que a partir del conocimiento de la clave pública no es posible determinar la clave privada. Los criptosistemas de clave pública, aunque más lentos que los simétricos, resultan adecuados para las funciones de autenticación, distribución de claves y firmas digitales.

Integridad de datos.- este mecanismo implica el cifrado de una cadena comprimida de datos a transmitir, llamada generalmente valor de comprobación de integridad (Integrity Check Value o ICV). Este mensaje se envía al receptor junto con los datos ordinarios. El receptor repite la compresión y el cifrado posterior de los datos y compara el resultado obtenido con el que le llega, para verificar que los datos no han sido modificados.

Firma digital.- este mecanismo implica el cifrado, por medio de la clave secreta del emisor, de una cadena comprimida de datos que se va a transferir. La firma digital se envía junto con los datos ordinarios. Este mensaje se procesa en el receptor, para verificar su integridad. Juega un papel esencial en el servicio de no repudio.

Control de acceso.- esfuerzo para que sólo aquellos usuarios autorizados accedan a los recursos del sistema o a la red, como por ejemplo mediante las contraseñas de acceso. Tráfico de relleno: consiste en enviar tráfico espurio junto con los datos válidos para que el atacante no sepa si se está enviando información, ni qué cantidad de datos útiles se está transmitiendo.

Control de encaminamiento.- permite enviar determinada información por determinadas zonas consideradas clasificadas. Asimismo posibilita solicitar otras rutas, en caso que se detecten persistentes violaciones de integridad en una ruta determinada. Unicidad: consiste en añadir a los datos un número de secuencia, la fecha y hora, un número aleatorio, o alguna combinación de los anteriores, que se incluyen en la firma digital o integridad de datos. De esta forma se evitan amenazas como la reactuación o resecuenciación de mensajes.

Los mecanismos básicos pueden agruparse de varias formas para proporcionar los servicios previamente mencionados. Conviene resaltar que los mecanismos poseen tres componentes principales:

- Una información secreta, como claves y contraseñas, conocidas por las entidades autorizadas.
- Un conjunto de algoritmos, para llevar a cabo el cifrado, descifrado, hash y generación de números aleatorios.
- Un conjunto de procedimientos, que definen cómo se usarán los algoritmos, quién envía qué a quién y cuándo.

Asimismo es importante notar que los sistemas de seguridad requieren una gestión de seguridad. La gestión comprende dos campos bien amplios:

- Seguridad en la generación, localización y distribución de la información secreta, de modo que sólo pueda ser accedida por aquellas entidades autorizadas.
- La política de los servicios y mecanismos de seguridad para detectar infracciones de seguridad y emprender acciones correctivas.

2.4.4.4.1. Firmas de Comprobación Aleatoria (Hash).

Las funciones de comprobación aleatoria son similares a las de cifrado (de hecho, algunas de ellas son funciones de cifrado con ligeras modificaciones). La mayoría de estas funciones toma un bloque de datos y lo somete reiteradamente a una sencilla función de desordenación (scrambling) para alterar sus elementos. Si esta operación se repite un cierto número de veces, no existe forma práctica conocida de predecir el resultado. Es imposible modificar un documento de un modo determinado y estar seguro de que la función de comprobación aleatoria producirá el mismo resultado.

Este tipo de firma utiliza una función de comprobación aleatoria criptográficamente segura, como Message digest 5 (MD-5) o Secure Hash Algorithm (SHA), para producir un valor de comprobación aleatoria a partir de un archivo. El procedimiento de comprobación aleatoria encadena su clave secreta. El destinatario también tiene una copia de la clave secreta y la utiliza para evaluar la firma.

2.4.4.4.2. Algoritmo DSS

En este algoritmo existen dos claves para cada persona. Una de ellas crea la firma y se mantiene secreta. La otra, la clave pública, verifica la firma.

El DSS fue desarrollado por el U.S. National Institute of Standards and Technology (NIST) con la colaboración de la National Security Agency (NSA). Sólo están obligadas a utilizarlo las compañías que mantienen negocios con el gobierno americano, y muchas prefieren no hacerlo porque es un sistema exclusivamente de firma. El NIST eligió esta solución limitada, porque el gobierno de EE.UU. pretende desalentar el uso de cualquier

software de cifrado que cercene su capacidad para fisgonear en asuntos ajenos. El software que sólo proporciona autenticación, como el DSS, puede exportarse libremente en los productos, mientras que el software que emplea RSA para cifrado general está sometido a severas restricciones.

2.4.4.4.3. Algoritmo RSA

Las firmas RSA son, con mucho, las más populares, gracias en parte al marketing agresivo, la política de patentes y el desarrollo a largo plazo que ha adoptado RSA Data Security. Esta empresa controla muchas de las patentes más importantes en este campo, y aunque ha tenido que hacer frente a numerosos litigios, RSA ha sabido utilizar su posición para consolidarse definitivamente como líder. Su software y sus bibliotecas se encuentran en el núcleo de muchos productos, y la compañía sigue contando entre sus filas con algunos de los criptógrafos más reconocidos.

RSA Data Security fue la firma encargada de integrar el software de firma digital con el sistema operativo Macintosh mucho antes de que floreciera el Web, por ejemplo. También añadió a PowerTalk (el software cooperativo de Apple) applets de firma de Arrastrar y soltar que permitían que cualquier usuario incrustara una firma digital en un formulario electrónico con sólo arrastrarlo hasta el icono. RSA Data Security ha otorgado licencia sobre sus patentes a los principales vendedores de sistemas operativos, incluidos Microsoft, IBM, Sun y Digital, y cada uno de ellos ha incorporado prestaciones similares a sus líneas de productos, aunque sin alcanzar el mismo nivel de integración. A diferencia de DSS, RSA también puede utilizarse para cifrar datos y proporcionar seguridad, además de autenticidad.

2.4.4.4.4. Seguridad del SET

Teóricamente, el elevado nivel de seguridad del SET está a prueba de bombas. Con una clave pública RSA de 1.024 bits, la seguridad de su implantación intrachip (onchip) parece total, o al menos muy superior a los sistemas burlados hasta ahora. Thian Yee Chua, presidente de Gemplus Japón, avanza que "durante el segundo semestre de 1997, Gemplus/MasterCard piensa evaluar este tipo de cifrado de tarjetas en Japón y Taiwán".

El cifrado basado en software suele ser más rápido que el RSA. El Data Encryption Algorithm (DEA) que emplean Netscape y otros vendedores para el comercio Internet también es significativamente más rápido que los algoritmos con base RSA. Sin embargo, este algoritmo es fácil de forzar, como ha demostrado recientemente Greenberg, de la University of California-Berkeley. Un código de cifrado DEA de 56 bits, que actualmente está en fase de desarrollo, será 65.000 veces más difícil de desarmar.

CAPÍTULO 3

METODOLOGÍA Y HERRAMIENTAS,

ANÁLISIS Y DISEÑO DE SISMONERIN

3.1. Herramientas y Equipos de Medición

3.1.1. Análisis de la Situación Actual

Debido a que este es un proyecto de carácter general no se puede realizar ninguna referencia específica hacia algún tipo de planta o estándar, pero para efectos de visualización y mejor comprensión de la problemática se realizará una división en 4 tipos de plantas industriales, que son los siguientes:

- Pequeñas, que poseen de 1 a 10 puntos de medición
- Medianas, que poseen de 11 a 50 puntos de medición
- Grandes, que poseen de 51 a 200 puntos de medición
- Extragrandes, que poseen más de 200 puntos de medición

En la actualidad existen unos pequeños medidores de potencia para sistemas trifásicos por un precio alrededor de USD. 300, pero estos no tienen ninguna posibilidad de comunicación con otros dispositivos, por lo tanto se tiene el problema de que un supervisor debería recorrer toda la planta constantemente para obtener los datos de potencia de cada uno de los puntos de carga tomando en cuenta que la memoria de almacenamiento de estos medidores no es muy grande y puede ocurrir una pérdida de datos importante mientras aumenta la cantidad de puntos a monitorear y la distancia entre estos. Se tiene un medidor de potencia trifásica más completo que tiene conexiones para RS-232 y RS-485 en forma opcional con su software propio y que puede guardar de 80.000 a 320.000 medidas con un costo alrededor de USD. 600 para cada punto de carga trifásica.

Aunque este es un medidor con capacidades de conexión de tipo serial se tiene que tomar en cuenta que en el computador donde se realiza el registro de las medidas deben existir varios puertos seriales para poder conectar varios de estos medidores, además el software que viene con el medidor necesitaría que sean del mismo tipo para poder registrar a varios de ellos porque de otro modo se necesitaría un software diferente para cada uno; finalmente se necesitaría desarrollar un software que recoja las medidas de todos los equipos de medición y esto implica un costo adicional. Realizando un análisis completo se observa que los costos totales llegan a ser más altos que el sistema que se está proponiendo mediante una tarjeta de adquisición de datos y un computador.

En la tabla 3.1 se mostrarán los costos en forma ilustrativa de acuerdo al número de puntos de medición en una planta industrial desde una pequeña hasta una grande, que son las que pueden existir con más frecuencia en el país.

Tabla 3.1. Tabla comparativa de costos del método tradicional de medición

Puntos de Medición	Método Tradicional
6 puntos	Medidores USD. 300 \$ 1,800.00
16 puntos	Medidores USD. 600 Computador USD. 1.000 \$ 10,600.00
40 puntos	Medidores USD. 600 Computador USD. 1.000 \$ 25,000.00
85 puntos	Medidores USD. 600 Computador USD. 1.000 \$ 52,000.00

Los medidores antes mencionados son las opciones más económicas disponibles en el mercado, sin embargo en muchos de estos casos se necesita realizar mediciones y análisis más complejos que implican equipos más costosos. Por todas las razones antes expuestas muchas empresas no tienen sistemas para monitoreo del uso de la energía eléctrica para sus máquinas y motores, finalmente tienen que pagar costos más elevados por multas del mal uso de la energía eléctrica o por reparaciones y adquisiciones de maquinaria nueva.

3.1.2. Estudio de las Perturbaciones Eléctricas vs. el Tiempo de Vida de las Máquinas

Primero se debe saber que debido a que la corriente eléctrica que se dispone para uso industrial y residencial ha sido generada en una forma oscilante y que por lo tanto la frecuencia a la que se recibe la energía no es la única sino la fundamental, es decir la que tiene más potencia de todas las frecuencias dentro de la señal.

Pero la energía eléctrica contiene otras frecuencias que se llaman armónicas y cada una tiene sus propias características. Para poder medir todas las armónicas de una onda de energía eléctrica del tendido público se necesitaría por lo menos 13 medidores de potencia para cada línea de carga. Cuando una o varias de estas armónicas están fuera de los rangos permitidos ocasionan daños a la maquinaria aunque imperceptibles pero se van acumulando.

Para poder transportar la energía eléctrica trifásica es necesario 3 conductores debido a que una onda de voltaje está desfasada 120° de la siguiente y de la anterior, por lo tanto puede suceder la ruptura de un solo conductor y no de toda la línea de transmisión y es lo que se conoce como caída de una fase. Frente a esta falla de suministro eléctrico pueden suceder muchas cosas como la destrucción casi inmediata de un motor por el desbalance de voltaje y corriente de la línea o un grave deterioro de la estructura interna que normalmente no es visible debido a que se encuentra dentro de la maquinaria y para esto se debería desarmar la misma.

Estos problemas deterioran generalmente el cable de las bobinas de los motores y maquinaria ocasionando un mayor consumo de energía que se representa en forma directa en aumento de costos de operación y, además que ya se ha reducido en un 10% a 20% el

tiempo de vida de las mismas porque no están operando en las condiciones nominales dadas por los fabricantes. Generalmente esto no es muy apreciado porque los tiempos de vida medios antes de fallas (MTBF) son bastantes altos en este tipo de maquinaria y de cierta forma casi despreciables.

Otro tipo de problema que suele suceder es la caída de tensión en toda la red de distribución eléctrica y los picos de tensión cuando se vuelve a la tensión nominal, generalmente esto no causa problemas a la maquinaria, pero puede causar paros de producción y/o daños a la materia prima causando pérdidas a la empresa. Casi la totalidad de la maquinaria industrial actual está diseñada para que soporte ciertos picos altos y caídas de tensión durante períodos muy cortos de tiempo, pero si estas fallas son dadas en forma repetitiva también causan un deterioro progresivo de la maquinaria.

3.1.3. Estudio de los Equipos que Permitan que el Proyecto sea Económico y Atractivo

Una vez realizada esta división, se procederá también a realizar una división de las tarjetas de adquisición según su número de entradas/salidas y la frecuencia de muestreo de las mismas.

- Pequeñas de USD. 350 a USD. 1000 con 16 entradas analógicas y frecuencias hasta 200.000 muestras por segundo
- Medianas de USD. 1000 hasta USD. 3000 con 64 entradas analógicas y frecuencias hasta 1.250.000 muestras por segundo
- Grandes desde USD. 3000 en adelante con 64 entradas analógicas y frecuencias hasta 5.000.000 muestras por segundo
- Propósito específico desde USD. 3500 con sólo 4 entradas analógicas y frecuencias desde 10.000.000 muestras por segundo en adelante

Para tener una idea más clara de como los costos aumentan mediante el método tradicional y el método propuesto se creará 2 escenarios de una industria pequeña en crecimiento. En el primer escenario mediante el método tradicional se supone que una pequeña industria tiene 6 puntos de medición y se usan los medidores pequeños de USD. 300, luego la planta crece y los puntos de medición aumentan a 16 y ya se hace necesario tener un monitoreo centralizado de los puntos de carga, por lo tanto se hace un cambio a los medidores de USD. 600 con el respectivo cableado y la adquisición de un computador que monitoreará los puntos de carga y almacenará la información.

En el segundo escenario mediante el método propuesto en esta tesis se tiene un costo inicial elevado, pero es muy flexible al momento de aumentar los puntos de medición y es casi imperceptible. En la tabla 3.2 se muestra un análisis comparativo de los costos que implica la implementación de un sistema de monitoreo mediante el método tradicional y el método de monitoreo basado en PC propuesto en la tesis.

En lo referente al software de monitoreo se podría prescindir de ese rubro también debido a que se puede desarrollar los controladores para la tarjeta DAQ y por supuesto la interfase de monitoreo, sin embargo no es tan recomendable porque este desarrollo tomaría

un tiempo considerable que representa el costo o más en tiempo que se debe pagar a un programador para que realice el trabajo.

Tabla 3.2 – Tabla comparativa de costos entre el método tradicional y el basado en PC

Puntos de Medición	Método Tradicional	Método Propuesto
6 puntos	Medidores USD. 300 \$ 1,800.00	Tarjeta DAQ USD. 700 Computador USD. 1.000 SW Monitoreo USD. 2.195 Programación USD. 500 \$ 4,395.00
16 puntos	Medidores USD. 600 Computador USD. 1.000 \$ 10,600.00	Tarjeta DAQ USD. 2.000 Computador USD. 1.000 SW Monitoreo USD. 2.195 Programación USD. 1000 \$ 6,195.00
40 puntos	Medidores USD. 600 Computador USD. 1.000 \$ 25,000.00	Tarjeta DAQ USD. 4.000 Computador USD. 1.000 SW Monitoreo USD. 2.195 Programación USD. 2000 \$ 9,195.00
85 puntos	Medidores USD. 600 Computador USD. 1.000 \$ 52,000.00	Tarjetas DAQ USD. 6.000 Computador USD. 1.000 SW Monitoreo USD. 2.195 Programación USD. 2000 \$ 11,195.00

Con el sistema propuesto de monitoreo basado en PC lo único que se necesita para realizar análisis más complejos es implementar la programación adecuada en el software sin tener que realizar nuevas adquisiciones de equipos, de tal manera que la implementación para estos análisis requerirá solamente de unos días de programación debido a que la infraestructura principal ya se encuentra montada. Como se observa en la tabla 3.2 con 16 puntos de medición existe un ahorro del 41,5%, para 40 puntos de medición existe un ahorro del 63,2% y para 85 puntos de medición existe un ahorro del 78,5%, de esta manera se ejemplifica claramente que a más puntos de medición mayor es el ahorro que puede tener la empresa.

Tabla 3.3 – Tabla comparativa de costos entre el método tradicional y el basado en PC con Acondicionadores de señales

Puntos de Medición	Método Tradicional	Método Propuesto
16 puntos	Medidores USD. 600 Computador USD. 1.000 \$ 10,600.00	Sistema de Monitoreo 1 Acondicionador \$ 7,195.00
40 puntos	Medidores USD. 600 Computador USD. 1.000 \$ 25,000.00	Sistema de Monitoreo 2 Acondicionadores \$ 11,195.00
85 puntos	Medidores USD. 600 Computador USD. 1.000 \$ 52,000.00	Sistema de Monitoreo 4 Acondicionadores \$ 15,195.00

En algunos casos se necesita que las medidas sean de alta precisión y confiabilidad y además se necesita filtrar el ruido electromagnético que se genera por los motores trifásicos en funcionamiento, para esto se debe adicionar cierto hardware que realiza un acondicionamiento de las señales eléctricas. Un equipo de acondicionamiento de señales analógicas de 32 canales cuesta USD. 1.000, lo cual no representa un costo muy representativo dado el ahorro que ya se tiene al implementar un sistema de monitoreo basado en PC. Para visualizar de mejor forma los costos totales se ha realizado la tabla 3.3 donde se reflejan los valores, pues para el caso del monitoreo de 16 puntos aún se tiene un ahorro del 32,1%, para 40 puntos de medición el ahorro es de 55,2% y para 85 puntos de medición el ahorro es del 70,8% demostrándose así una vez más el ahorro de costos que se puede obtener con un sistema de monitoreo propuesto en la presente tesis.

3.2. Selección de Plataformas Operativas y de Aplicación

Cada uno de los sistemas operativos mencionados a continuación tiene sus fortalezas y sus debilidades, los se va a analizar para seleccionar cual plataforma es la más indicada para el sistema.

3.2.1. Estudio de Plataformas Microsoft

Con NT el problema es que se puede manejar el servidor solo desde otra máquina Windows NT que incluya una Consola de Administración Microsoft (MMC) el cual proporciona un explorer como estilo de interfase. Así de esta manera cuando carga la (MMC) carga también el módulo de IIS.

Existe una versión HTML de IIS pero se puede tomar ventaja con el Internet Explorer para manejar las capacidades administrativas en forma remota. Pero el IIE es el único componente que puede ser manejado por un Web Browser y no puede ser remplazado por otro.

NT viene con una versión de Internet Information Server (IIS) que tan solo es un código Beta, pero la última versión es la mas codiciada por todas las personas puesto que incluye el mejor modelo de programación para servidor web como son los ASP.

NT soporta la mayoría de las herramientas que Linux tiene, el inconveniente es su costo ya que es muy elevado.

También hay que analizar las necesidades, puesto que si solo es para un grupo de trabajo pequeño NT es demasiado pero si se habla de utilizar las herramientas de desarrollo para web existentes en el mercado NT es la mejor opción.

En cuanto a instalación se refiere NT es un sistema demasiado largo y complicado puesto que hay que instalar todo manualmente y siendo un proceso tan largo es una instalación muy tediosa. Por otro lado cuando se presentan problemas solo existen cajas de diálogos que solucionan los problemas de forma muy genérica que no ayudan en mucho.

GUI ha cambiado desde Windows 3.1 a Windows 95 en una forma drástica y de esta a Windows 2000 muy poco pero hacia Windows XP existe bastante cambio. Estas dos últimas versiones permiten GUI. Para Windows GUI es un componente esencial del sistema operativo

Para la plataforma Windows se ha seleccionado los siguientes paquetes con sus costos correspondientes los que se han investigados de las siguientes páginas de Internet:

<http://www.microsoft.com/windows2000/advancedserver/howtobuy/pricing/default.asp>

<http://www.microsoft.com/isaserver/howtobuy/pricing/default.asp>

<http://www.microsoft.com/sql/howtobuy/pricing/default.asp>

<http://www.microsoft.com/exchange/howtobuy/pricing/default.asp>

Tabla 3.4. Tabla de costos de la solución desarrollada mediante la plataforma Microsoft

Sistema Operativo

	Precio	Proveedor	Comentarios
Microsoft Windows 2000 Advanced Server	\$3,999.00	Microsoft	Viene con 25 licencias de acceso cliente, \$67 por cada una

Software e Desarrollo

	Precio	Proveedor	Comentarios
Microsoft ISA Standard Server 2000	\$1,499.00	Microsoft	Licencia de procesador. Incluye servidor firewall y proxy.
Microsoft SQL Server 2000	\$4,999.00	Microsoft	Licencia de procesador.

Microsoft Exchange Server 2000	\$1,299.00	Microsoft	Viene con 5 licencias de acceso cliente, \$67 por cada una.
LabVIEW 6i para Windows	\$2,195.00	National Instruments	

3.2.2. Estudio de Plataformas Linux

Linux toma gran ventaja en cuanto a la manejabilidad puesto que la administración es hecha al estilo Unix, lo cual significa que maneja la máquina desde cualquier ordenador remotamente. El problema es que no existe suficiente información de ayuda por lo que hace al proceso mas laborioso que en el caso de NT.

Permitiendo de esta manera que el otro extremo pueda ser accesado por cualquier máquina con cliente Telnet.

Linux permite obtener un servidor Web en forma libre es decir sin costo por cada copia del Software que sea obtiene Desafortunadamente siempre viene la versión vieja pero esto no es un problema ya que mediante Internet se puede hacer la actualización y obtener así la ultima versión que incluye scripts para configurar, compilar e instalar el mismo, haciendo de esto un proceso simple y breve.

El servidor web de Linux incluye soporte para todas las aplicaciones CGI, Per, Tcl/Tk y compiladores de C/ C++, PHP, incluso contiene un proyecto que permite al servidor correr Java Servlets. A pesar de NT tener las mejores herramientas Linux ofrece mejor flexibilidad en cuanto se trata de una plataforma web server

Completamente diferente a NT la instalación de Linux es muy simple, puesto que hay que tomar decisiones acerca de cómo particionar el disco, incluye una herramienta que permite simplificar la partición del disco llamada Disk Druid

Linux típicamente provee 2 tipos de GUI: KDE y Gnome. Aquí el GUI es opcional y en algunas ocasiones permite realizar un control remoto y una administración remota de una computadora en una forma más natural que lo que hace Windows.

Para la plataforma Linux se ha seleccionado los siguientes paquetes los mismos que hacen referencia sus costos en la siguiente página web:

<http://www.mandriva.com>

Tabla 3.5. Tabla de costos de la solución desarrollada mediante la plataforma Linux

Sistema Operativo

	Precio	Comentarios
Mandrake Linux 10.0	\$100.00	Apropiado para estaciones de trabajo o servidores.

Software de Desarrollo

	Precio	Comentarios
Apache (Web server)	Incluido en la distribución de Linux o descarga gratuita	Eficiente y extensible servidor web, usado un 59% de servidor web en Internet.
Squid (Proxy server)	Incluido en la distribución de Linux o descarga gratuita	Alto rendimiento web y cache en el servidor proxy.
MySQL (Database)	Incluido en la distribución de Linux o descarga gratuita	MySQL es resistente para la siguiente generación Object Relational DBMS
LabVIEW 6i para Linux	\$2,195.00	Extrañamente es más caro en la plataforma Linux y no ofrece todas las funcionalidades que la versión Windows

3.2.3. Análisis Comparativo

Se comparará las dos plataformas para ver cual es la que mas beneficios brinda al proyecto, y se apoyará mucho en el aspecto económico.

Tabla 3.6. Tabla de comparación de costos entre las plataformas Microsoft y Linux

	Windows	Linux
Software		
Software de Plataforma	\$3,999.00	\$150.00
Software de Desarrollo	\$9,992.00	\$2,195.00
Software de Productividad	\$479.00	\$0.00
Total de Costos de Software.	\$14,470.00	\$2,345.00

Costos Operativos		
Salarios del personal (2 personas)	\$700.00	\$700.00
Conexión a Internet	\$100.00	\$100.00
Total Costos Operacionales	\$800.00	\$800.00

Costos Totales	\$15,270.00	\$3,145.00
-----------------------	--------------------	-------------------

En conclusión con Linux se ahorra \$12,125.00 que es el 79.40% del costo total.

3.2.4. Selección de Plataforma Operativa y de Desarrollo

Puesto que las dos plataformas brindan los mismos servicios y Linux es la más económica y se ha optado por ésta plataforma ya que al ser de menor costo es más accesible para las empresas ya sean medianas o grandes.

Otra razón por la que se ha optado por esta plataforma es que cualquier actualización que se necesite para el software se realiza a través de Internet y no es necesario adquirir nuevas licencias, lo que permite a futuro poder ampliar las aplicaciones de acuerdo las necesidades de la empresa.

3.3. Especificación de Requerimientos Mediante la Norma IEEE 830

3.3.1. Introducción

3.3.1.1. Propósito

El propósito de este sistema es monitorear la energía eléctrica en ambientes industriales, capturar información relevante que permita realizar análisis y estadísticas para optimizar el uso de energía eléctrica y poder monitorear la planta a través de la web.

3.3.1.2. Alcance

Este sistema permitirá realizar un monitoreo de la energía eléctrica en la planta industrial, pero no permite un control completo sobre la misma. El control que se hará es a nivel estadístico y administrativo, es decir que mediante la información obtenida del monitoreo se tomarán decisiones gerenciales para un mejor uso y explotación de este recurso.

3.3.1.3. Definiciones, Acrónimos y Abreviaturas

3.3.1.3.1. Acrónimos

- SISMONERIN.- Sistema de Monitoreo de Energía Eléctrica en ambientes industriales a través de la web
- RDBMS.- Relational Data Base Management System (Sistema de administración de bases de datos relacionales)
- LAN.- Local Area Network (Red de Área Local)
- WAN.- Wide Area Network (Red de Área Extendida)
- Mbps.- Megabits por segundo
- DAQ.- Data Acquisition (Adquisición de Datos)

3.3.1.3.2. Abreviaciones

- SW.- Software
- HW.- Hardware

3.3.1.4. Referencias

La información para la implementación de este sistema se tomará de tutoriales de adquisición de datos, medición de energía y métodos prácticos de medición de energía eléctrica; así como también de métodos prácticos y tips según Microsoft, Oracle, Sybase, Linux Journal y otros de implementación de aplicaciones distribuidas.

3.3.1.5. Apreciación General

El sistema debe ser flexible y portable que permita monitorear la energía eléctrica en ambientes industriales, así como también capturar información relevante que facilite realizar análisis y estadísticas para tomar decisiones gerenciales en cuanto a una optimización del uso de energía eléctrica y además que permita monitorear de una forma centralizada una o más plantas industriales debido a que su interfase se basa en web. Se planea tener una independencia de la plataforma operativa.

3.3.2. Descripción General

3.3.2.1. Perspectiva del Producto

El sistema a realizarse tendrá las siguientes características:

- Reducción de costos por utilizar productos de libre distribución
- Independencia de plataformas ya que desde cualquier browser se puede visualizar la planta
- Monitoreo centralizado, pues permite monitorear una o varias plantas en un solo computador

3.3.2.2. Funciones del Producto

Las funciones que tendrá el sistema son:

- Conectar las tarjetas acondicionadoras de señal a las máquinas de la planta
- Tomar los datos de la planta para ser analizados mediante las tarjetas de adquisición
- A través del monitoreo detectar y evitar los posibles problemas en la planta causados por las fallas de las líneas de energía

3.3.2.3. Características del Usuario

➤ **Administrador.-** Este usuario no tiene restricciones, tiene acceso a todas las secciones y funciones del sistema.

➤ **Operador Medio.-** Este es un usuario de nivel medio que puede realizar un monitoreo intermedio de la planta, no pudiendo acceder a todos los niveles de seguridad del sistema.

➤ **Operador Básico:** Este usuario solo puede realizar un monitoreo básico y general de la planta y no puede detener el sistema.

3.3.2.4. Limitaciones

➤ **Políticas de la Empresa.-** como políticas se considera que la empresa que utilice este sistema es pequeña a mediana y que además tenga una apertura hacia el software libre.

➤ **Limitaciones del Hardware.-** las limitaciones de hardware se dan de acuerdo al número de ranuras de expansión PCI que posea el computador sobre el cual se colocan las tarjetas de adquisición, por lo cual se recomienda que se utilice una placa madre de tipo full ATX que posea de 6 a 7 ranuras de expansión para que se pueda extender el sistema cuando sea necesario.

➤ **Interfaces con otras aplicaciones.-** al momento no existen interfaces con otras aplicaciones, sin embargo, los datos están abiertos para que puedan interactuar con otras aplicaciones como las financieras para que se integre todo en una visión global de la empresa.

➤ **Operaciones Paralelas.-** este sistema puede trabajar en paralelo con equipos de propósito específico, sin embargo no es recomendable, pues crea una carga adicional y la medición no es precisa.

➤ **Funciones de Auditoria.-** el sistema realizará funciones de auditoria básica como el ingreso y salida de un usuario hacia el sistema.

➤ **Funciones de Control.-** el sistema realizará un chequeo de control básico donde verificará la existencia y disponibilidad hacia el mismo de las tarjetas de adquisición de datos; registrando su estado en el momento de inicialización de la aplicación, así como también cuando exista un cambio de estado de algún dispositivo de adquisición.

➤ **Protocolos de Comunicación.-** como protocolos de comunicación se utilizará el TCP/IP estándar para comunicación entre la base datos, la interfase de servidor y la de cliente.

- **Requisitos de Fiabilidad.-** la fiabilidad de los datos que obtenga la aplicación depende solamente de la calidad de tarjetas de adquisición que se coloquen para el funcionamiento del sistema.
- **Criticidad de la Aplicación.-** la aplicación no es de misión crítica ya que hace un monitoreo de la energía eléctrica que proviene de forma externa. El almacenamiento de datos se lo hace cada 5 minutos obteniendo los máximos y mínimos de las mediciones, por lo tanto el nivel de criticidad de la misma es bajo.
- **Consideraciones de Seguridad.-** la aplicación realizará sólo el monitoreo de la calidad y continuidad de la energía eléctrica suministrada, por lo tanto se empleará un esquema básico de seguridad que consiste de autenticación de usuario, de otro modo se necesitará un servidor adicional que se encargue de proveer dichos servicios.

3.3.2.5. Asunciones y Dependencias

Se asume que las personas que van a manejar el sistema tienen los conocimientos suficientes para poder manejar un browser de internet, además de conocimientos acerca del funcionamiento de los equipos utilizados como la tarjeta de adquisición de datos entre otros.

3.3.3. Especificación de Requerimientos

3.3.3.1. Interfases Externas

3.3.3.1.1. Interfases de Usuario

Para los usuarios comunes se encuentran las interfaces web que dependiendo del grado de seguridad indicado anteriormente podrán utilizar los usuarios en sus diferentes categorías. En el sistema de servidor se encontrará la interfase en ventanas de la plataforma operativa; aquí se realizarán las tareas administrativas como creación, eliminación y modificación de usuarios, áreas, circuitos y puntos de medición.

3.3.3.1.2. Interfases de Hardware

En el computador principal que se conecta a la planta industrial se utilizarán los siguientes equipos:

- Tarjeta de adquisición de datos en el computador
- Acondicionadores de señales conectados a la tarjeta de adquisición
- Transformadores de voltaje y corriente en cada punto.

3.3.3.1.3. Interfases de Software

En el presente sistema se utilizarán algunas interfases de software para poder comunicar las aplicaciones de servidor y de cliente con las interfases de hardware y los almacenamientos de datos.

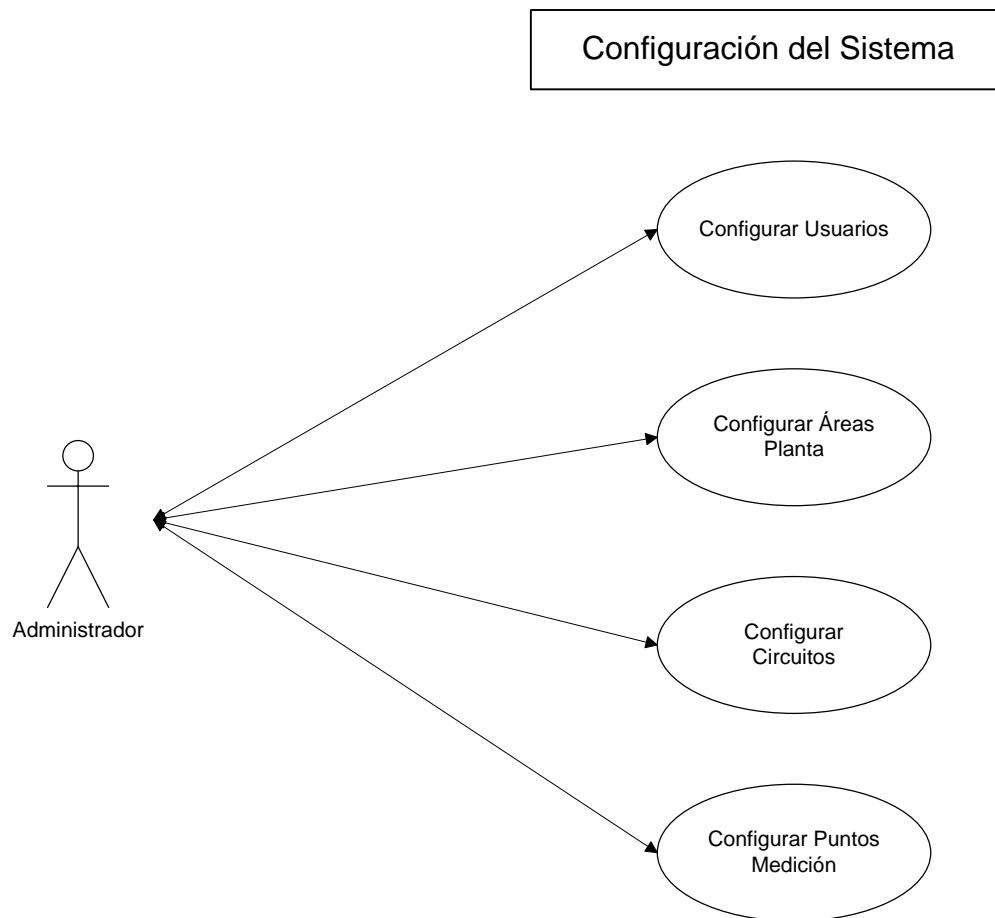
- Controladores para la comunicación con la tarjeta DAQ
- Controladores para comunicarse con la base de datos

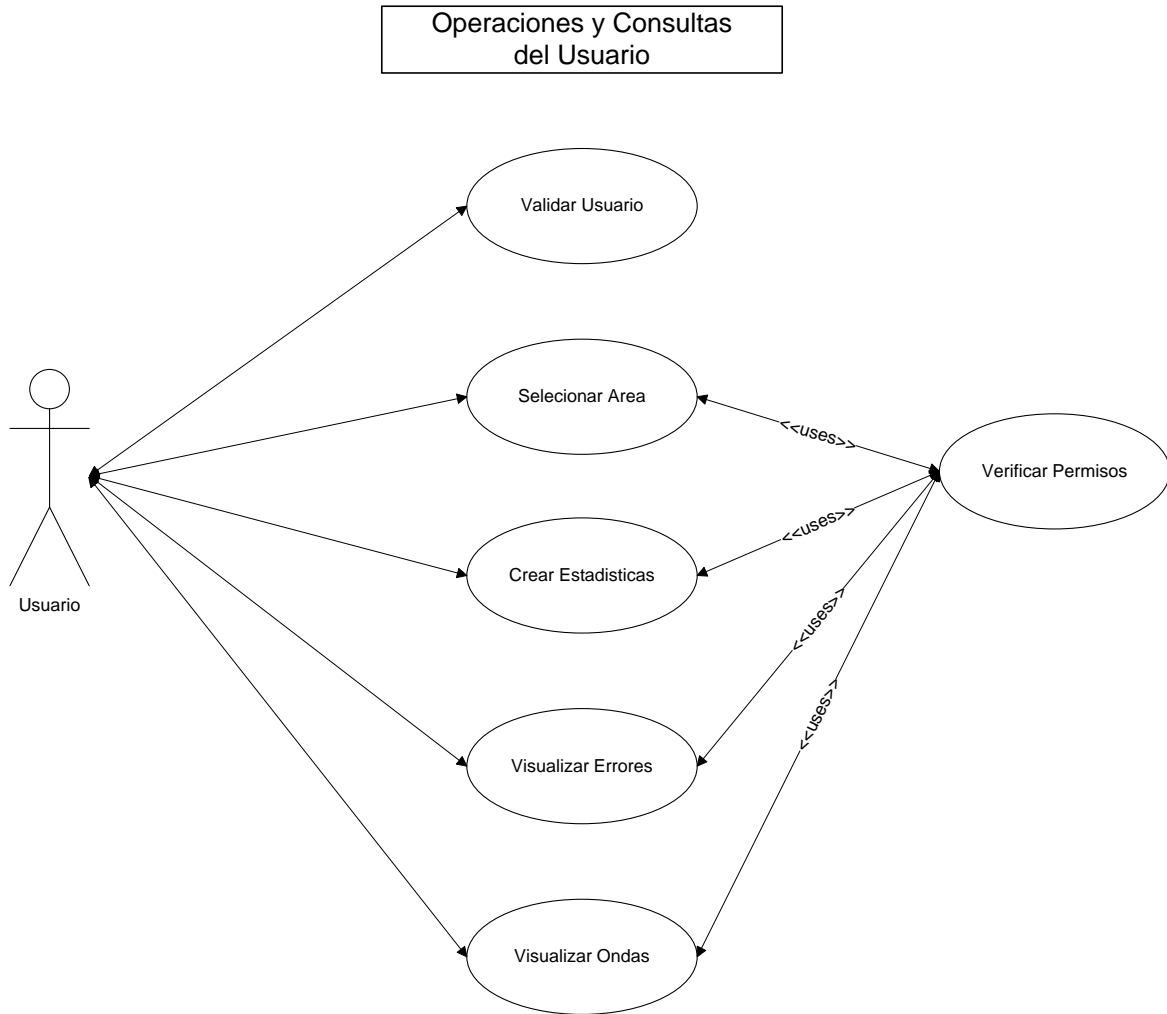
3.3.3.1.4. Interfases de Comunicaciones

Cualquier tipo de interfase de LAN o WAN que el usuario disponga y sea ésta cableada o inalámbrica.

3.3.3.2. Funciones

3.3.3.2.1. Casos de Uso



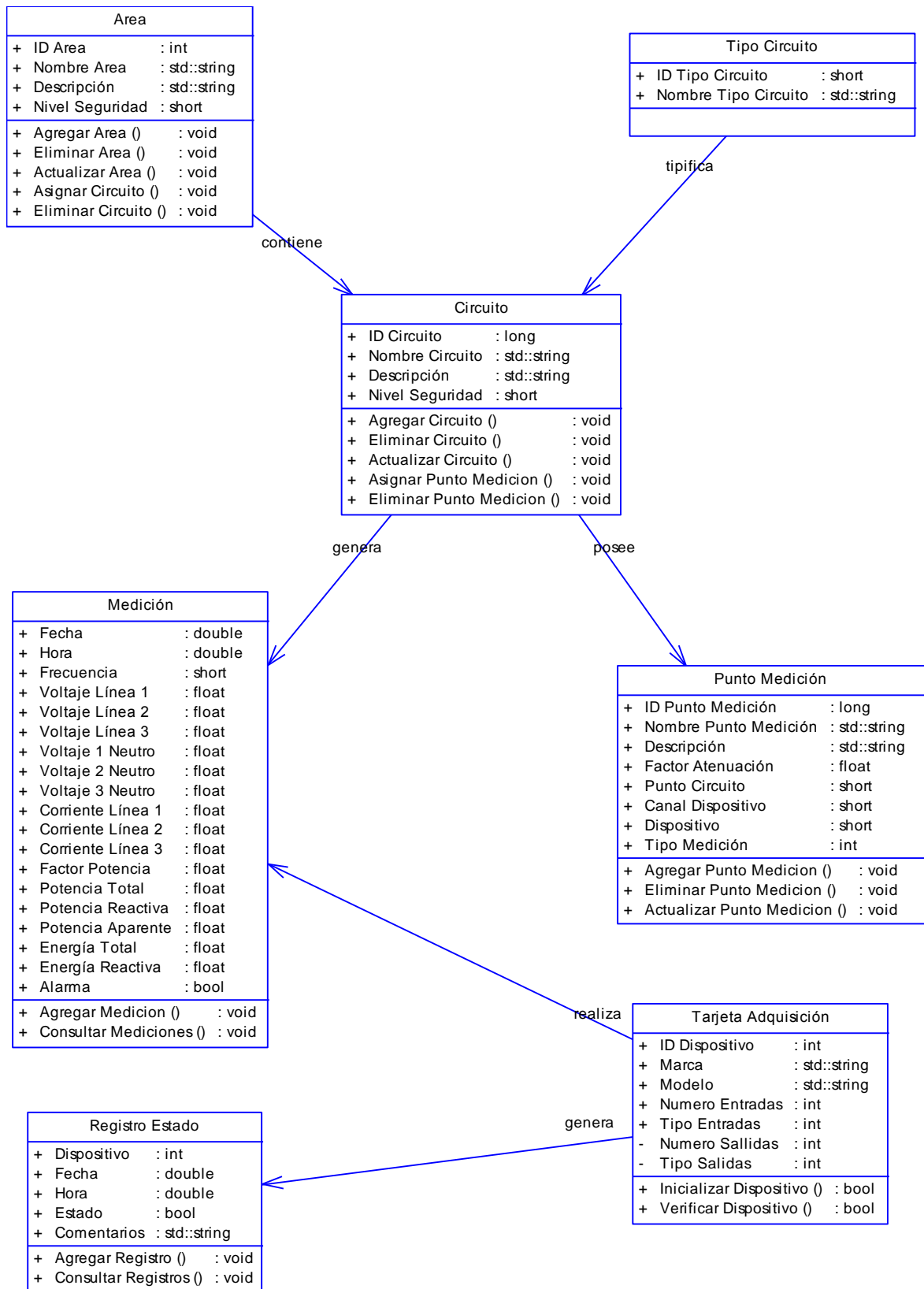


3.3.3.2. Descripción Textual

El administrador es el encargado de crear a los demás usuarios del sistema con sus respectivos permisos y tiene todos los derechos sobre el mismo.

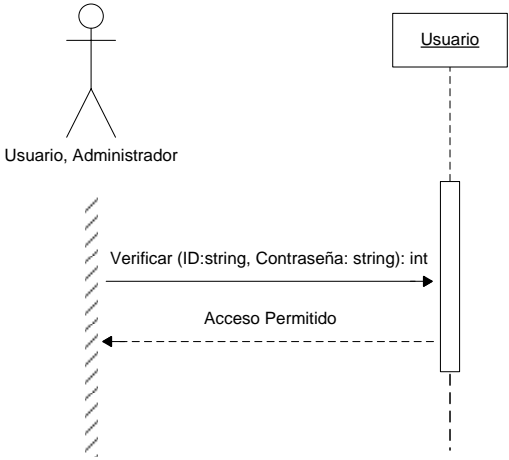
El usuario se considera desde un nivel de seguridad de 1 a 9, el cual no puede hacer cambios en la configuración del sistema bajo ningún aspecto ni puede cambiar los objetos de visualización que no sean permitidos mostrar.

3.3.3.2.3. Diagramas de Clases

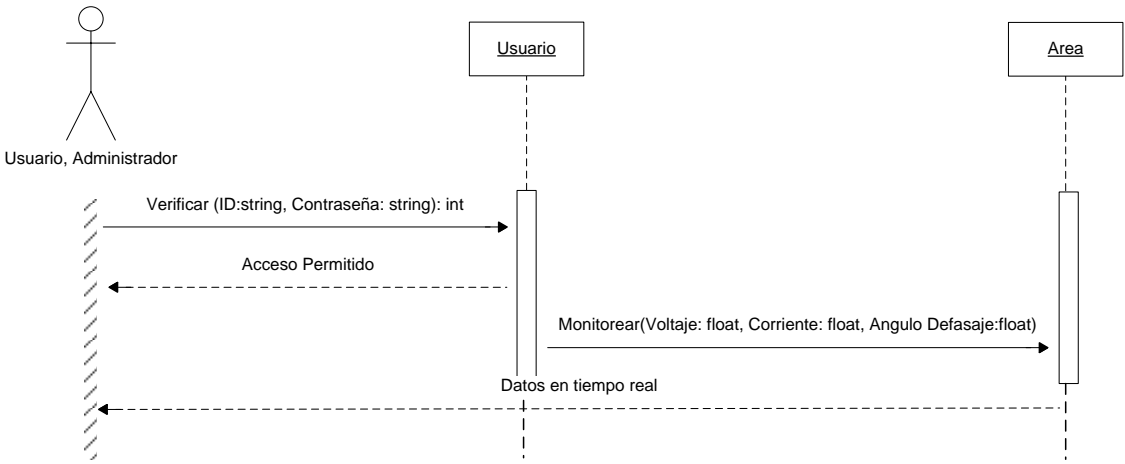


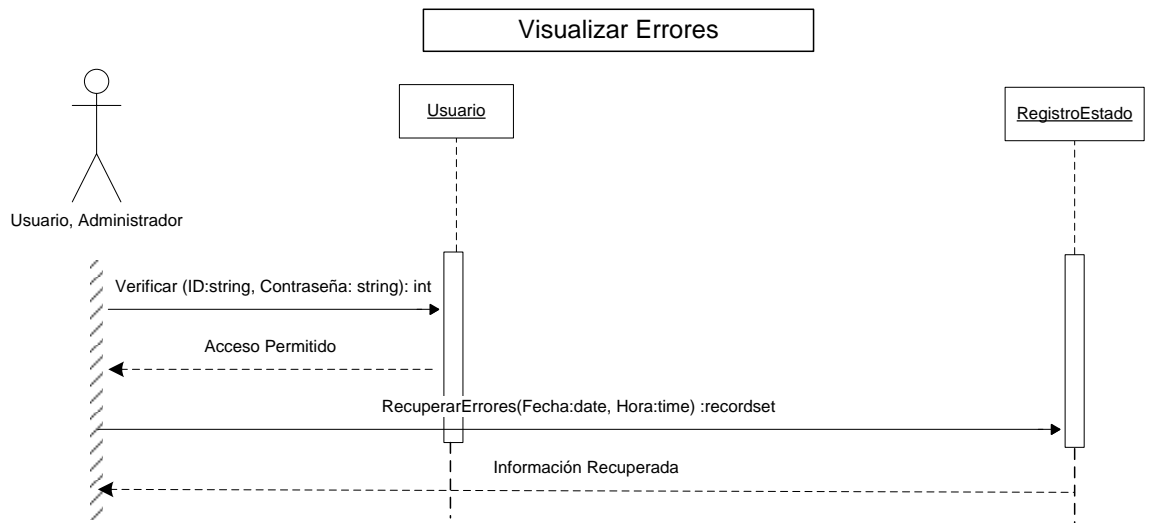
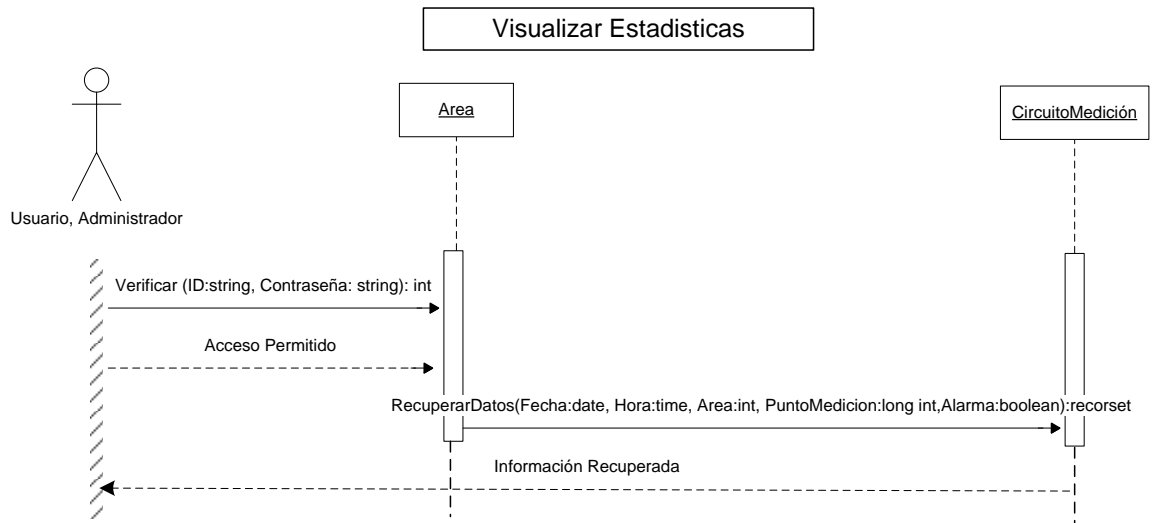
3.3.3.2.4. Diagramas de Interacción

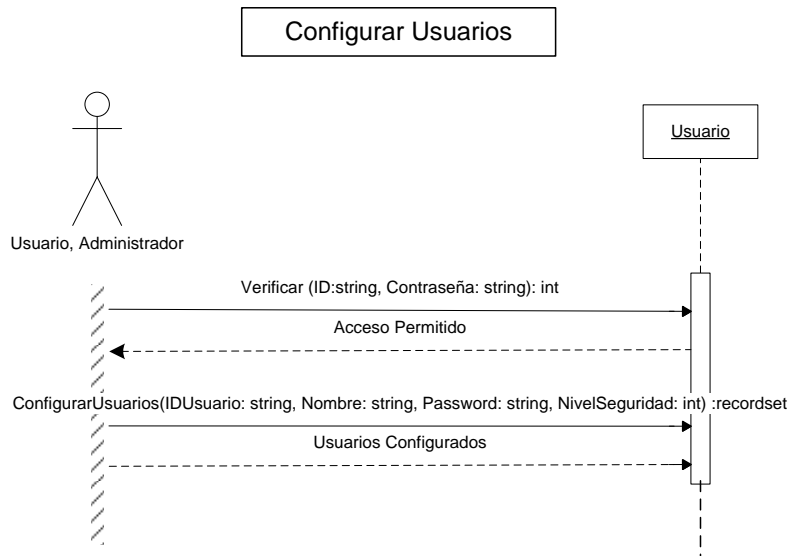
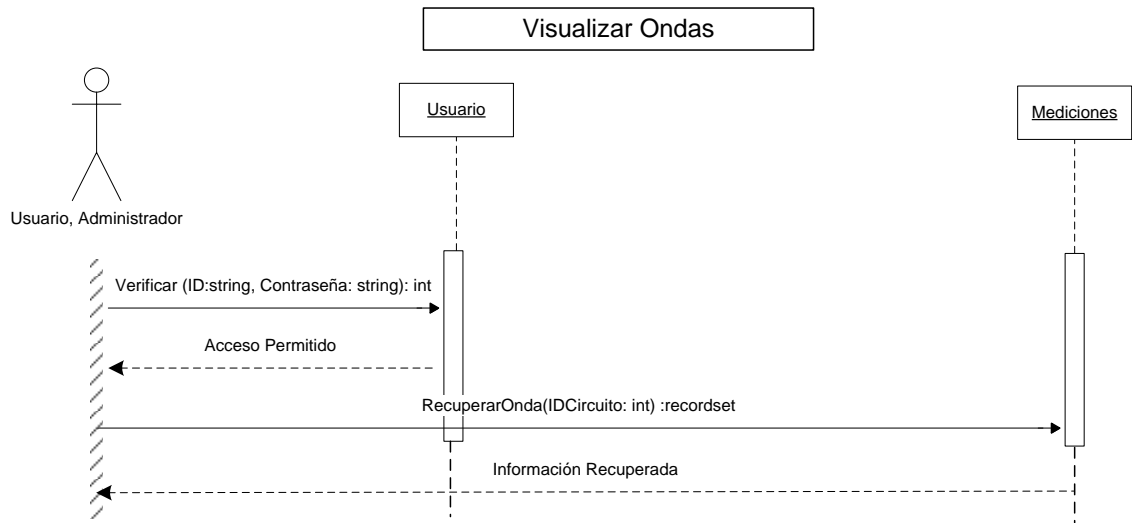
Acceder al Sistema

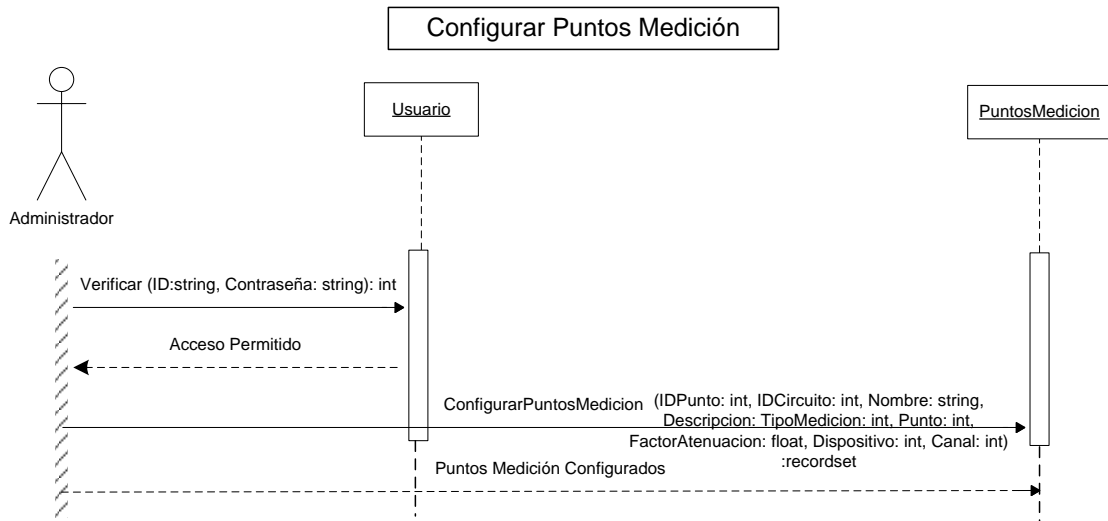
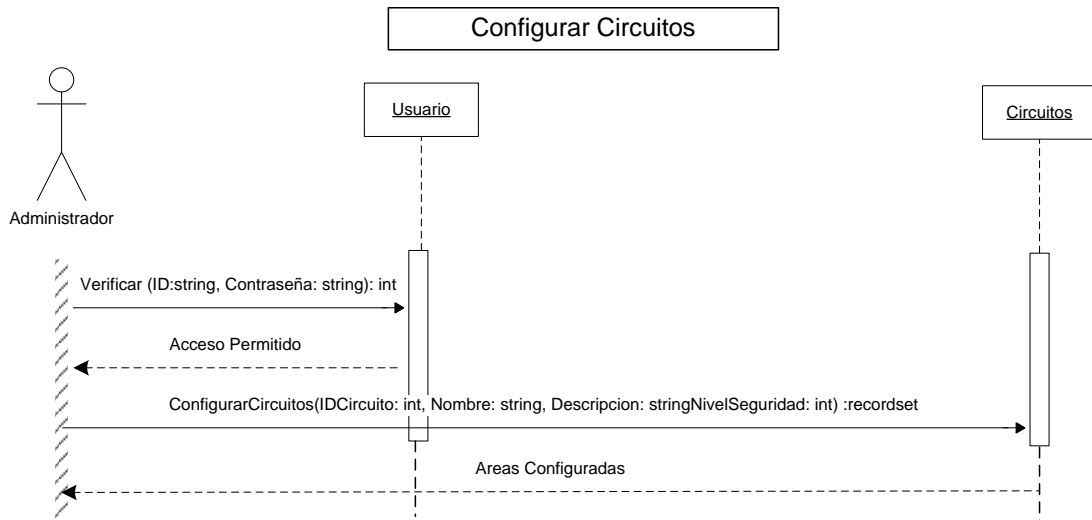
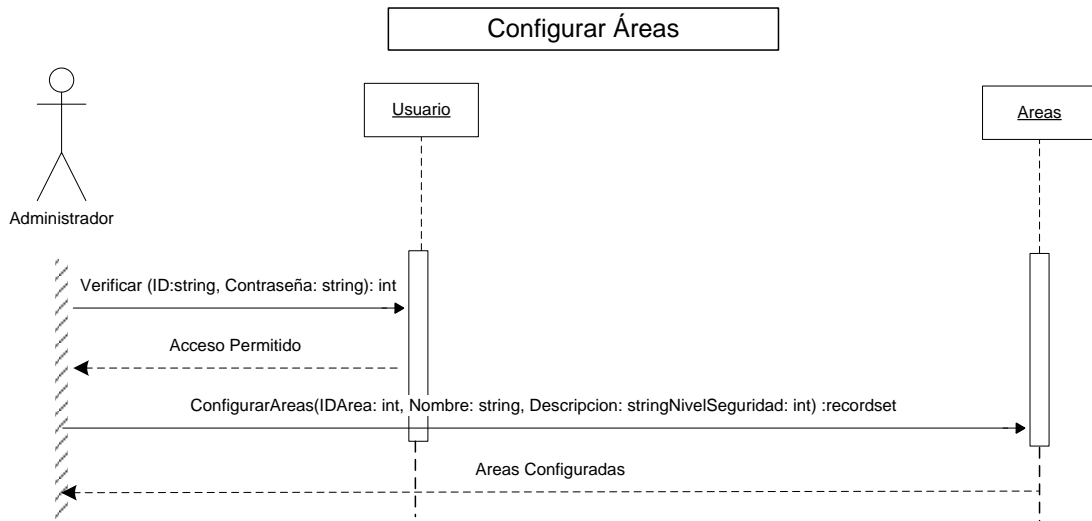


Seleccionar Area de Monitoreo



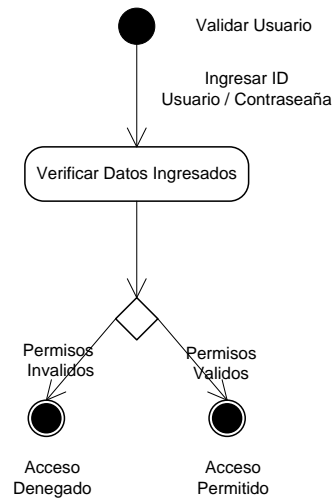




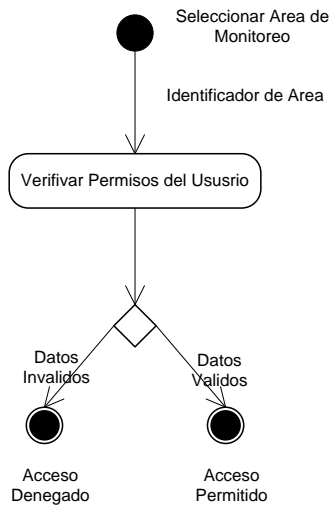


3.3.3.2.5. Diagramas de Transición de Estados

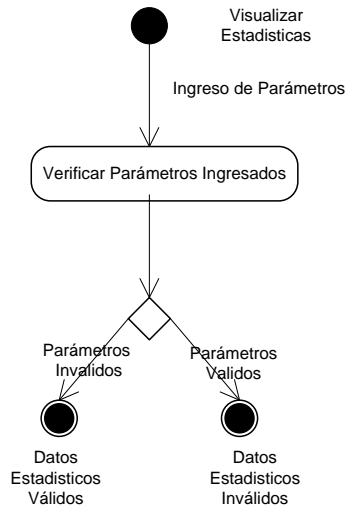
Validar Usuario



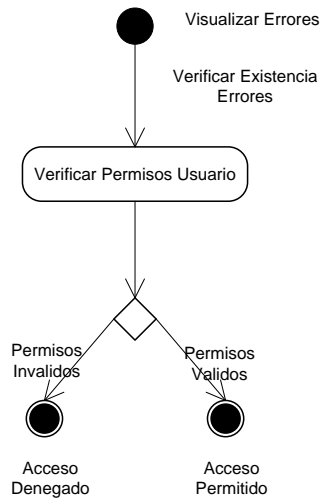
Seleccionar Area de Monitoreo



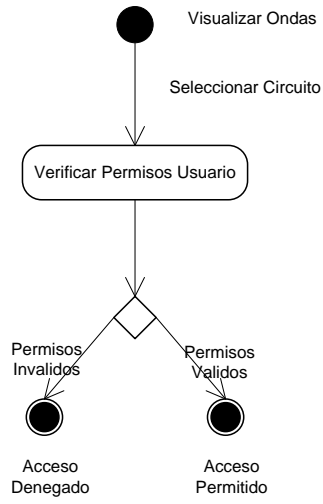
Visualizar Estadísticas



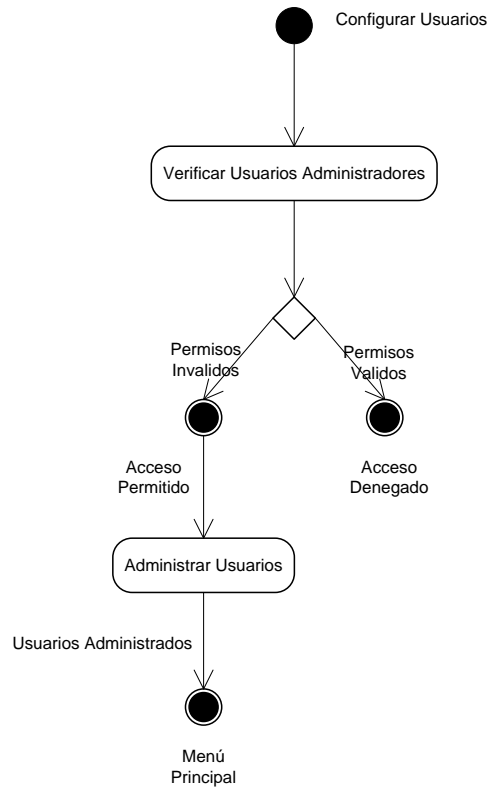
Visualizar Errores



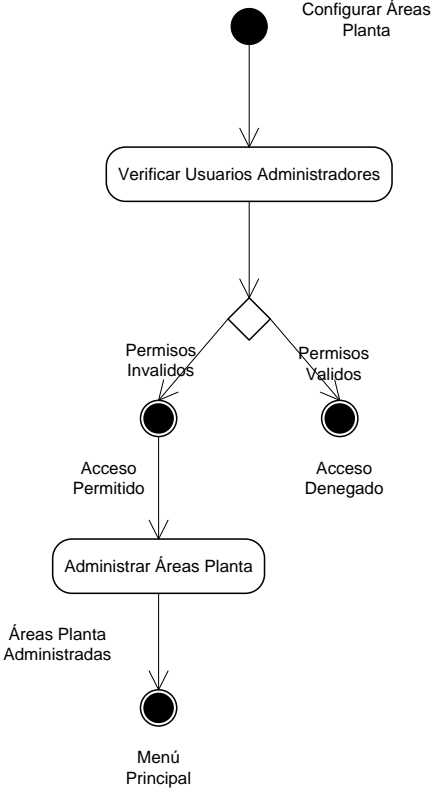
Visualizar Ondas



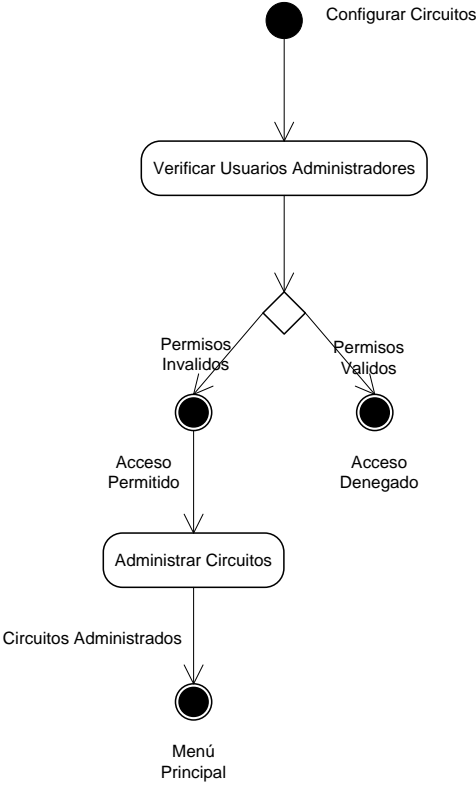
Configurar Usuarios

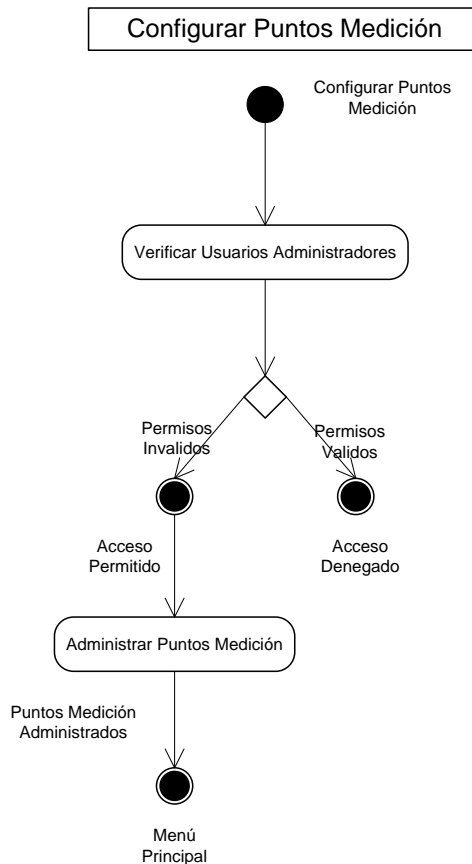


Configurar Áreas Planta



Configurar Circuitos





3.3.3.2.6. Prototipos Ejecutables

Para este sistema no se pueden definir prototipos ejecutables ya que se necesita toda la funcionalidad completa e integrada; además el lenguaje de programación que se utiliza es de tipo gráfico sin asistentes y por lo tanto no permite la creación de prototipos.

3.3.3.3. Requerimientos de Desempeño

El sistema en cuestión debe funcionar como una aplicación de 3 capas e implementarse mediante una intranet, ya que los clientes serán browsers de Internet.

La capa inferior la soportará el motor del servidor de bases de datos de MySQL versión 4.21, ya que se ha propuesto utilizar software con licencia GPL. La capa media funcionará con el servidor web Advanced Extranet Server mediante Apache 2.0.5, ya que la aplicación de servidor de adquisición de datos se encuentra sobre Mandrake Linux 10.0. Finalmente la capa superior se encuentra implementada como una intranet que posee páginas con contenido estático y dinámico mediante páginas de servidor PHP y JavaScript.

El sistema debe proveer un gran desempeño en velocidad de respuesta en la parte que realizará el monitoreo en tiempo real de las ondas de voltaje y corriente de cada uno de

los circuitos en la planta; sin embargo puede ser aceptable un pequeño retraso de respuesta en las páginas que obtienen los datos estadísticos de la base de datos y tiempo real del monitoreo de circuitos hasta realizar la comunicación.

3.3.3.4. Limitaciones de Diseño

El sistema se limitará a realizar un monitoreo de la energía eléctrica que provenga de la red pública o de los generadores que provea la empresa. Algunas máquinas y motores trifásicos pueden necesitar especial atención cuando cae algunas de las fases que alimentan a los mismos, sin embargo esto será parte de otro sistema.

El presente sólo registrará estadísticamente los valores RMS y acumulativos de cada uno de los circuitos conectados a las tarjetas de adquisición de datos.

3.3.3.5. Atributos del Sistema de Software

3.3.3.5.1. Confiabilidad

El sistema a desarrollarse presentará una alta confiabilidad debido a que el lenguaje básico sobre el cual está desarrollado LabVIEW es C++. Se realiza la conexión a la base de datos mediante componentes escritos en C nativo y compilados como librerías compartidas.

El servidor de aplicaciones se implementará mediante el servidor Apache 2.0.5 debido a que está integrado con la plataforma operativa de Mandrake Linux y tiene también su amplia trayectoria en el mercado.

Con referencia a los cálculos matemáticos que se realizarán en la aplicación del servidor se pueden garantizar también una alta confiabilidad debido a que LabVIEW es el líder mundial actual en el área de instrumentación virtual.

3.3.3.5.2. Disponibilidad

El sistema contará con una alta disponibilidad siempre y cuando se implementare mediante capas bien marcadas y distribuidas en distintos servidores de la siguiente manera:

- Servidor de aplicación de adquisición de datos
- Servidor de base de datos
- Servidor de Internet y comunicaciones en tiempo real

De esta manera se garantiza que cada equipo se especialice en las actividades específicas que debe realizar y de esta manera brindar una alta disponibilidad al sistema.

3.3.3.5.3. Seguridad

Para la seguridad del sistema se implementará el sistema básico de nombre de usuario y contraseña. Los usuarios para la parte basada en GUI y basada en entorno web serán los mismos, además en web se utilizará el esquema de tiempo límite para la sesión de usuario.

Existen otros esquemas de seguridad como la capa de sockets seguros y otros protocolos de seguridad y ciframiento, sin embargo no se hará uso de los mismos porque requiere una infraestructura más avanzada para el soporte y funcionamiento.

3.3.3.5.4. Mantenibilidad

Con respecto a la mantenibilidad se puede notar que es muy alta debido a que la aplicación ha sido diseñada para que sea muy flexible con respecto a posibilidades de expansión, pues lo único que se necesita es añadir tarjetas de adquisición de datos al computador y configurarlas en el sistema del servidor aumentando así la capacidad de puntos de medición.

Todos los cambios que se realicen en la configuración de los puntos de medición se reflejarán automáticamente en las páginas web de estadísticas y monitoreo, sin embargo, para la aplicación en GUI se necesita reiniciar la misma para que cargue la nueva configuración creando así un sistema muy fácil de realizar mantenimiento.

3.3.3.5.5. Portabilidad

En cuanto a la portabilidad se requiere que el sistema pueda funcionar sobre cualquier plataforma operativa de uso comercial como es el caso de Windows, MacOS, Linux y Solaris en lo que se refiere a la aplicación de servidor. La aplicación de cliente debe funcionar sobre cualquier browser de Internet y por lo tanto debe poder visualizarse virtualmente sobre cualquier plataforma operativa existente.

CAPÍTULO 4

CONSTRUCCIÓN DE SISMONERIN

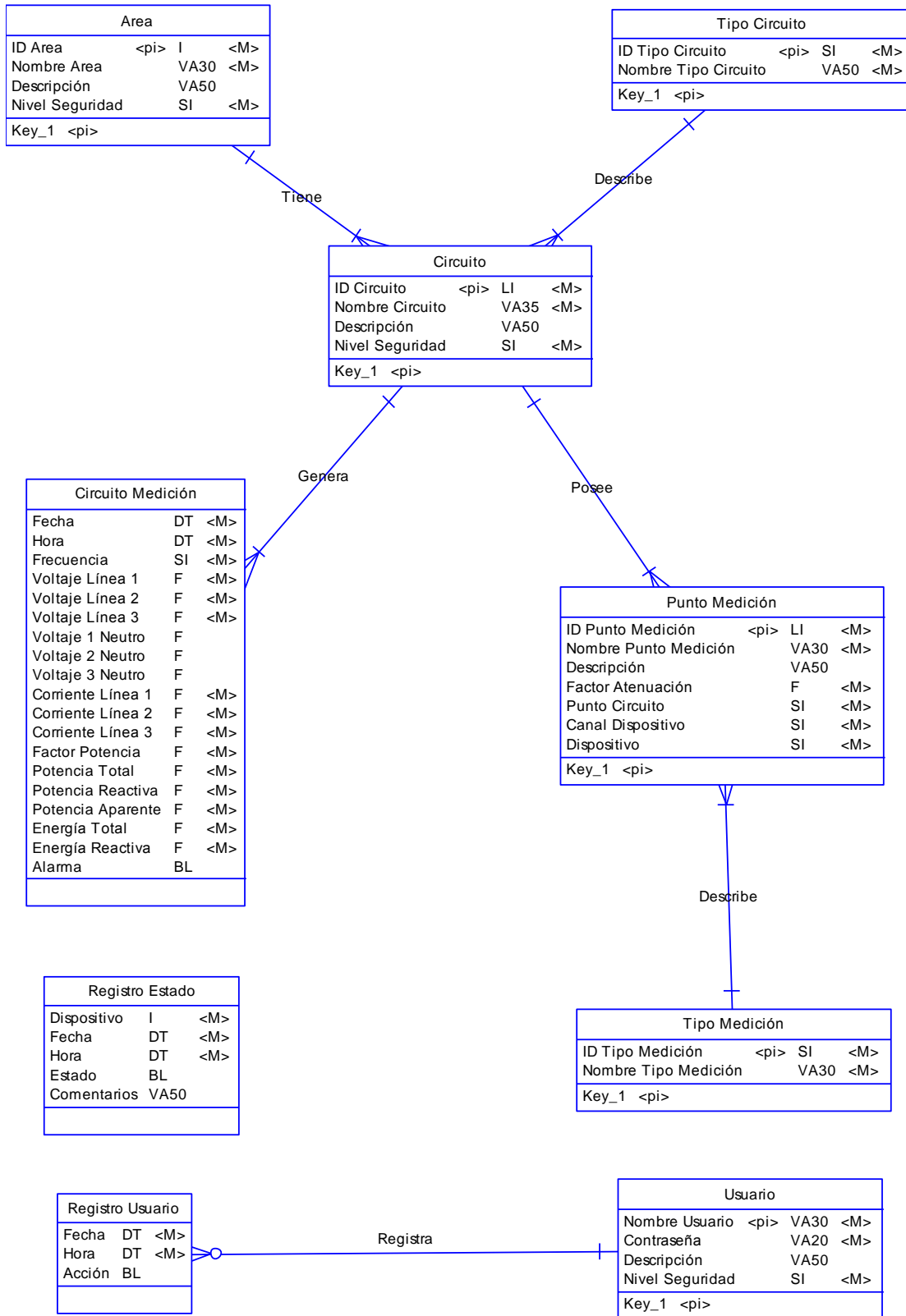
4.1. Construcción de la Aplicación

Para la construcción de la aplicación se utilizarán herramientas que no son orientadas a objetos como lo es la DBMS y el lenguaje de programación de la aplicación de adquisición de datos, para lo cual se debe realizar una transformación del modelo de objetos a los modelos entidad – relación donde existe un modelo lógico o conceptual que es el equivalente al modelo de objetos y el modelo físico que depende de la DBMS sobre la cual se implemente por los tipos de datos que dispone cada una.

Una vez que se tiene el modelo físico se procede a generar el script de creación de la base de datos y luego éste se ejecuta sobre la interfase de la DBMS para realizar consultas y poder ejecutar aquí el bloque de instrucciones.

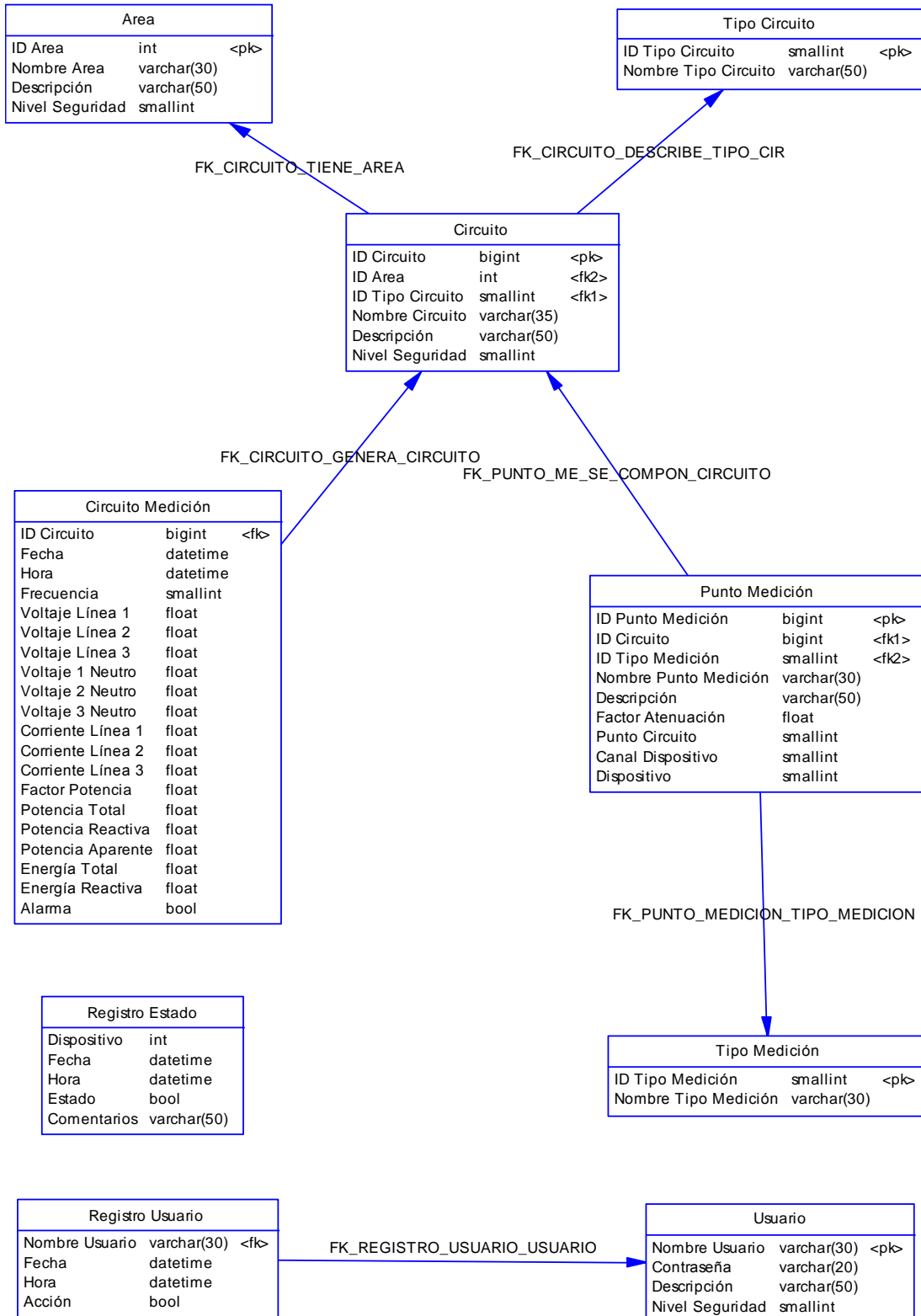
4.1.1. Modelo Conceptual

MODELO CONCEPTUAL



4.2.2. Modelo Físico

MODELO FÍSICO



4.2.3. Construcción de Tablas

Para la implementación de la base de datos se ha optado por utilizar MySQL debido a que es una de las DBMS con licencia GPL más conocida y explotada en la actualidad y, que además funciona sobre una gran variedad de plataformas operativas. Para la creación de las tablas se ha utilizado el lenguaje SQL estándar y aplicado mediante un script indicado a continuación.

```
/*=====*/
/* DBMS name:   monitoreo           */
/* Created on:  2005/07/01 1:57:12 AM */
/*=====*/

drop index DESCRIBE_FK on Circuito;

drop index TIENE_FK on Circuito;

drop index GENERA_FK on CircuitoMedicion;

drop index SE_COMPONE_FK on PuntoMedicion;

drop index Index_1 on RegistroUsuario;

drop table if exists Area;

drop table if exists Circuito;

drop table if exists CircuitoMedicion;

drop table if exists PuntoMedicion;

drop table if exists RegistroEstado;

drop table if exists RegistroUsuario;

drop table if exists TipoCircuito;

drop table if exists TipoMedicion;

drop table if exists Usuario;

/*=====*/
/* Table: Area           */
/*=====*/
create table Area
(
```

```

areIDArea          int          not null,
areNombreArea      varchar(30)    not null,
areDescripcion     varchar(50),
areNivelSeguridad  smallint    not null,
primary key (areIDArea)
);

/*=====*/
/* Table: Circuito */
/*=====*/
create table Circuito
(
  crcIDCircuito     bigint      not null,
  areIDArea         int         not null,
  tctIDTipoCircuito smallint    not null,
  crcNombreCircuito varchar(35)  not null,
  crcDescripcion    varchar(50),
  crcNivelSeguridad smallint    not null,
  primary key (crcIDCircuito)
);

/*=====*/
/* Index: DESCRIBE_FK */
/*=====*/
create index DESCRIBE_FK on Circuito
(
  tctIDTipoCircuito
);

/*=====*/
/* Index: TIENE_FK */
/*=====*/
create index TIENE_FK on Circuito
(
  areIDArea
);

/*=====*/
/* Table: CircuitoMedicion */
/*=====*/
create table CircuitoMedicion
(
  crcIDCircuito     bigint      not null,
  cmdFecha          date        not null,
  cmdHora           time        not null,
  cmdFrecuencia     smallint    not null,
  cmdVoltaje1      float       not null,

```

```

cmdVoltaje2          float          not null,
cmdVoltaje3          float          not null,
cmdVoltaje10         float,
cmdVoltaje20         float,
cmdVoltaje30         float,
cmdCorriente1        float          not null,
cmdCorriente2        float          not null,
cmdCorriente3        float          not null,
cmdFactorPotencia    float          not null,
cmdPotencia          float          not null,
cmdPotenciaReactiva  float          not null,
cmdPotenciaAparente float          not null,
cmdEnergia           float          not null,
cmdEnergiaReactiva   float          not null,
cmdAlarma            bool
);

/*=====*/
/* Index: GENERA_FK */
/*=====*/
create index GENERA_FK on CircuitoMedicion
(
  crcIDCircuito
);

/*=====*/
/* Table: PuntoMedicion */
/*=====*/
create table PuntoMedicion
(
  pmdIDPuntoMedicion  bigint          not null,
  crcIDCircuito        bigint          not null,
  tmdIDTipoMedicion   smallint        not null,
  pmdNombrePuntoMedicion varchar(30)  not null,
  pmdDescripcion      varchar(50),
  pmdFactorAtenuacion float          not null,
  pmdPuntoCircuito    smallint        not null,
  pmdCanal             smallint        not null,
  pmdDispositivo      smallint        not null,
  primary key (pmdIDPuntoMedicion)
);

/*=====*/
/* Index: "DESCRIBE_FK" */
/*=====*/
create index DESCRIBE_FK
(

```

```

    tmdIDTipoMedicion
);

/*=====*/
/* Index: SE_COMPONE_FK */
/*=====*/
create index SE_COMPONE_FK on PuntoMedicion
(
    crcIDCircuito
);

/*=====*/
/* Table: RegistroEstado */
/*=====*/
create table RegistroEstado
(
    rgdDispositivo      int          not null,
    rgdFecha            date         not null,
    rgdHora              time        not null,
    rgdEstado           bool,
    rgdComentarios      varchar(50)
);

/*=====*/
/* Table: RegistroUsuario */
/*=====*/
create table RegistroUsuario
(
    usrNombreUsuario    varchar(30)  not null,
    rguFecha            date         not null,
    rguHora              time        not null,
    rguAccion           bool
);

/*=====*/
/* Index: Index_1 */
/*=====*/
create index Index_1 on RegistroUsuario
(
    usrNombreUsuario
);

/*=====*/
/* Table: TipoCircuito */
/*=====*/
create table TipoCircuito
(

```

```

    tctIDTipoCircuito      smallint      not null,
    tctNombreTipoCircuito  varchar(50)      not null,
    primary key (tctIDTipoCircuito)
);

/*=====*/
/* Table: TipoMedicion */
/*=====*/
create table TipoMedicion
(
    tmdIDTipoMedicion      smallint      not null,
    tmdNombreTipoMedicion  varchar(30)      not null,
    primary key (tmdIDTipoMedicion)
);

/*=====*/
/* Table: Usuario */
/*=====*/
create table Usuario
(
    usrNombreUsuario      varchar(30)      not null,
    usrContrasena        varchar(20)      not null,
    usrDescripcion        varchar(50),
    usrNivelSeguridad     smallint      not null,
    primary key (usrNombreUsuario)
);

alter table Circuito add constraint FK_CIRCUITO_DESCRIBE_TIPO_CIR foreign key
(tctIDTipoCircuito)
references TipoCircuito (tctIDTipoCircuito) on delete restrict on update restrict;

alter table Circuito add constraint FK_CIRCUITO_TIENE_AREA foreign key (areIDArea)
references Area (areIDArea) on delete restrict on update restrict;

alter table CircuitoMedicion add constraint FK_CIRCUITO_GENERA_CIRCUITO
foreign key (crcIDCircuito)
references Circuito (crcIDCircuito) on delete restrict on update restrict;

alter table PuntoMedicion add constraint FK_PUNTO_MEDICION_TIPO_MEDICION
foreign key (tmdIDTipoMedicion)
references TipoMedicion (tmdIDTipoMedicion) on delete restrict on update restrict;

alter table PuntoMedicion add constraint FK_PUNTO_ME_SE_COMPON_CIRCUITO
foreign key (crcIDCircuito)
references Circuito (crcIDCircuito) on delete restrict on update restrict;

```

alter table RegistroUsuario add constraint FK_REGISTRO_USUARIO_USUARIO foreign key (usrNombreUsuario) references Usuario (usrNombreUsuario) on delete restrict on update restrict;

4.2.4. Construcción de Interfaces

4.2.4.1. Interfaces de Aplicación Servidora

Para la construcción de las interfaces se utilizó LabVIEW 6i para la aplicación de servidor; el cual proporciona una gran facilidad en cuanto a controles visuales tridimensionales y específicos para la instrumentación, sin embargo de que posee una programación simplificada por ser gráfica puede llegar a ser muy extensa y laboriosa en aplicaciones medianas a grandes. Además, se utilizó lenguaje C para los componentes de conexión a la base de datos y hacia las tarjetas de adquisición de datos A continuación se muestra las figuras de cada de las interfaces que el usuario utilizará en la aplicación de servidor.



Figura 4.1 – Cuadro de Dialogo de Ingreso y Confirmación del Password del Administrador

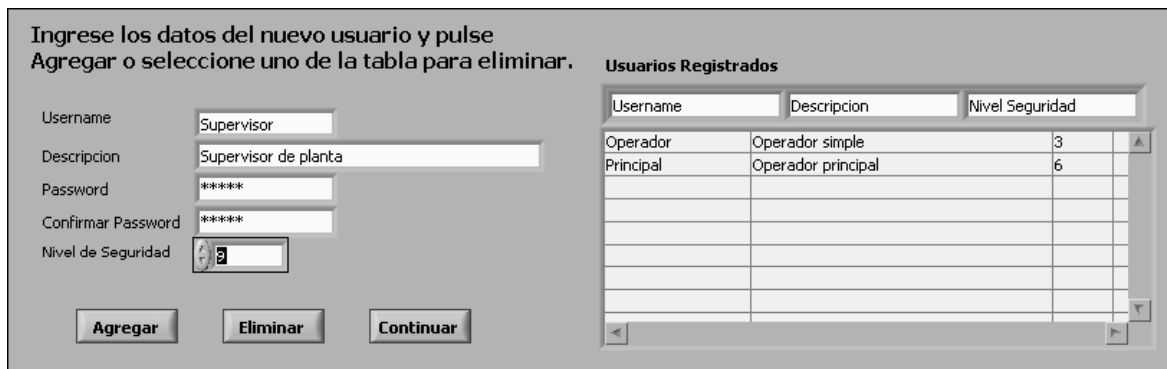


Figura 4.2 – Pantalla de Configuración de Usuarios

Ingrese los datos de la nueva área de monitoreo y pulse Agregar o seleccione uno de la tabla para eliminar.

Área:
 Descripción:
 Nivel de Seguridad:

Áreas Registradas

ID Área	Área	Descripción	Nivel Seguridad
1	Calderos	Calderos para vapor	8
3	Fermentacion	Fermentacion de ceb	6
5	Gases	Inyeccion de agua y	5
4	Lavado	Lavado y esterilizaor	7
2	Maquinas	Maquinas para embol	1

Figura 4.3 – Pantalla de Configuración de Áreas

Ingrese los datos del nuevo circuito de monitoreo y pulse Agregar o seleccione uno de la tabla para eliminar.

Tipo Circuito:
 Área:
 Circuito:
 Descripción:
 Nivel de Seguridad:

Circuitos Registrados

ID Circuito	Tipo Circuito	Área	Circuito	Descripción	Nivel Seguridad
1	Trifasico 3 Hilos	Calderos	Caldero 1	Circuito 1 del area d	9
2	Trifasico 4 Hilos	Calderos	Caldero 2	Circuito de Caldero 2	9
3	Trifasico Balancead	Calderos	Caldero 3	Circuito 3 del area i	8
4	Trifasico 3 Hilos	Fermentacion	Fermentacion 1	Circuito 1 del area f	7
5	Trifasico Balancead	Fermentacion	Fermentacion 2	Circuito 2 del area d	8
6	Trifasico 3 Hilos	Gases	Gases Livianos	Circuitos de gases li	8
7	Trifasico 4 Hilos	Gases	Gases pesados	Circuito de gases p	9

Figura 4.4 – Pantalla de Configuración de Circuitos

Ingrese los datos del nuevo punto de medición y pulse Agregar o seleccione uno de la tabla para eliminar.

Circuito:
 Punto Medicion:
 Descripción:
 Tipo Medicion:
 Factor Atenuacion:
 Punto Circuito:
 Canal:
 Dispositivo:

Puntos de Medición Registrados

ID Punto	Circuito	Punto Medicion	Descripción	Tipo Medicion	Factor Atenuacion	Punto Circuito	Canal	Dispositivo
5	Caldero 2	Punto 1 - Caldero 2	Punto 1 - Caldero 2	Corriente	5	1	4	3
6	Caldero 2	Punto 1 - Caldero 2	Punto 1 - Caldero 2	Voltaje	5	1	5	3
7	Fermentacion 1	Punto 1 - Fermentac	Punto 1 - Fermentac	Corriente	10	1	0	1
8	Fermentacion 1	Punto 1 - Fermentac	Punto 1 - Fermentac	Voltaje	10	1	1	1
9	Fermentacion 1	Punto 2 - Fermentac	Punto 2 - Fermentac	Voltaje	0	2	0	1
10	Fermentacion 1	Punto 2 - Fermentac	Punto 2 - Fermentac	Corriente	0	2	2	4
11	Fermentacion 1	Punto 3 - Fermentac	Punto 3 - Fermentac	Corriente	12	3	4	1
12	Fermentacion 1	Punto 3 - Fermentac	Punto 3 - Fermentac	Voltaje	12	3	5	1
13	Fermentacion 1	Punto 1 - Lavado Co	Punto 1 - Lavado Co	Corriente	3	1	0	2
14	Lavado Completo	Punto 1 - Lavado Co	Punto 1 - Lavado Co	Voltaje	0	1	1	9
15	Lavado Completo	Punto 2 - Lavado Co	Punto 2 - Lavado Co	Corriente	0	2	2	2
16	Lavado Completo	Punto 2 - Lavado Co	Punto 2 - Lavado Co	Voltaje	0	2	3	0
17	Lavado Completo	Punto 2 - Lavado Co	Punto 2 - Lavado Co	Voltaje	0	2	3	0

Figura 4.5 – Pantalla de Configuración de Puntos de Medición

Login.vi

Digite el Username y Password para ingresar

UserID:

Password:

Figura 4.6 – Pantalla del VI de Verificación de Usuarios



Figura 4.7 – Pantalla Principal con los Botones para Configuraciones

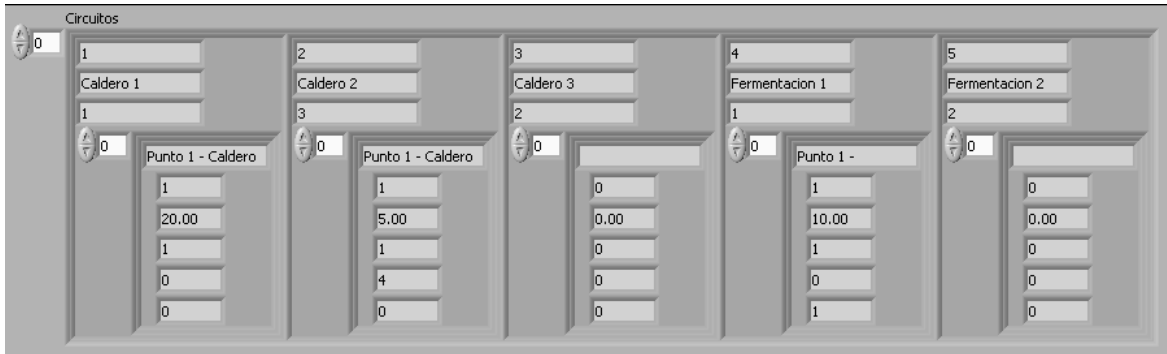


Figura 4.8 – Interfase de Usuario, Arreglo de Clusters de Circuito

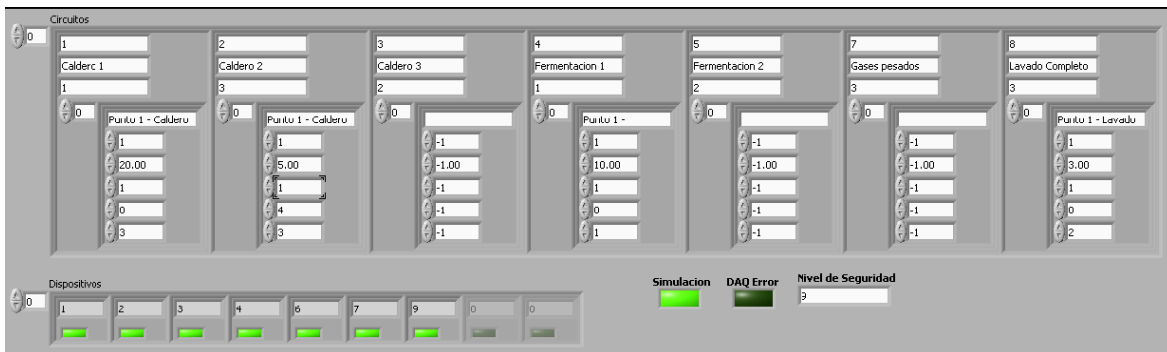


Figura 4.9 – Interfase de Usuario, Verificación de Tarjetas DAQ

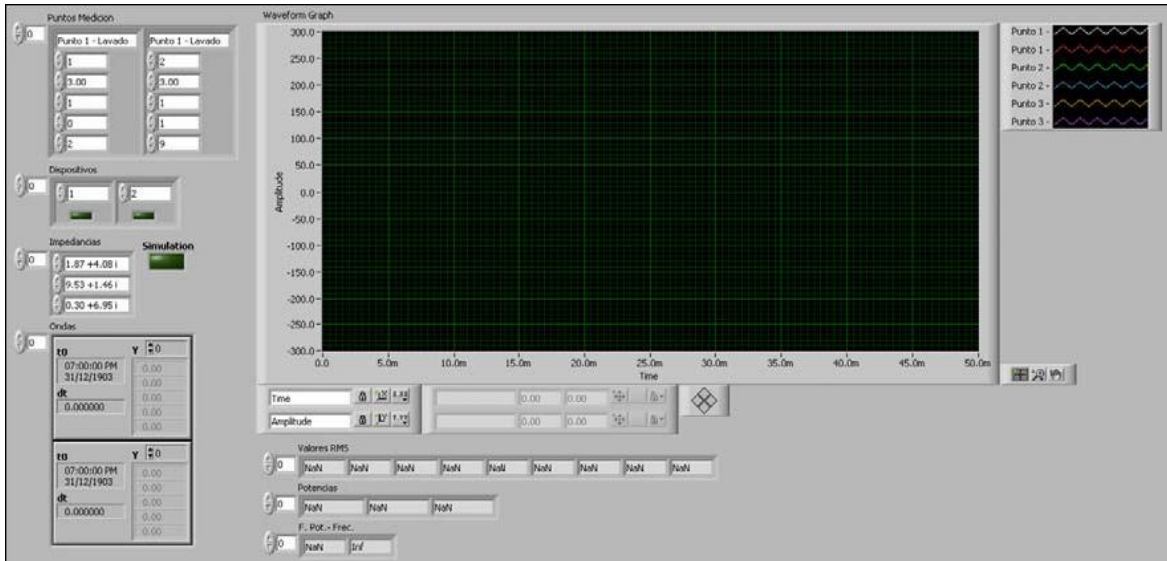


Figura 4.10 – Panel Frontal del VI de Circuito Trifásico de 4 Hilos en Adquisición

En la figura 4.01 se visualiza la interfase que analiza y monitorea un circuito trifásico de 4 hilos. Para los casos de circuitos trifásicos de 3 hilos y 3 hilos balanceados se utilizará la misma interfase, los cambios dentro del proceso y cálculos son invisibles pues el resultado es el mismo.

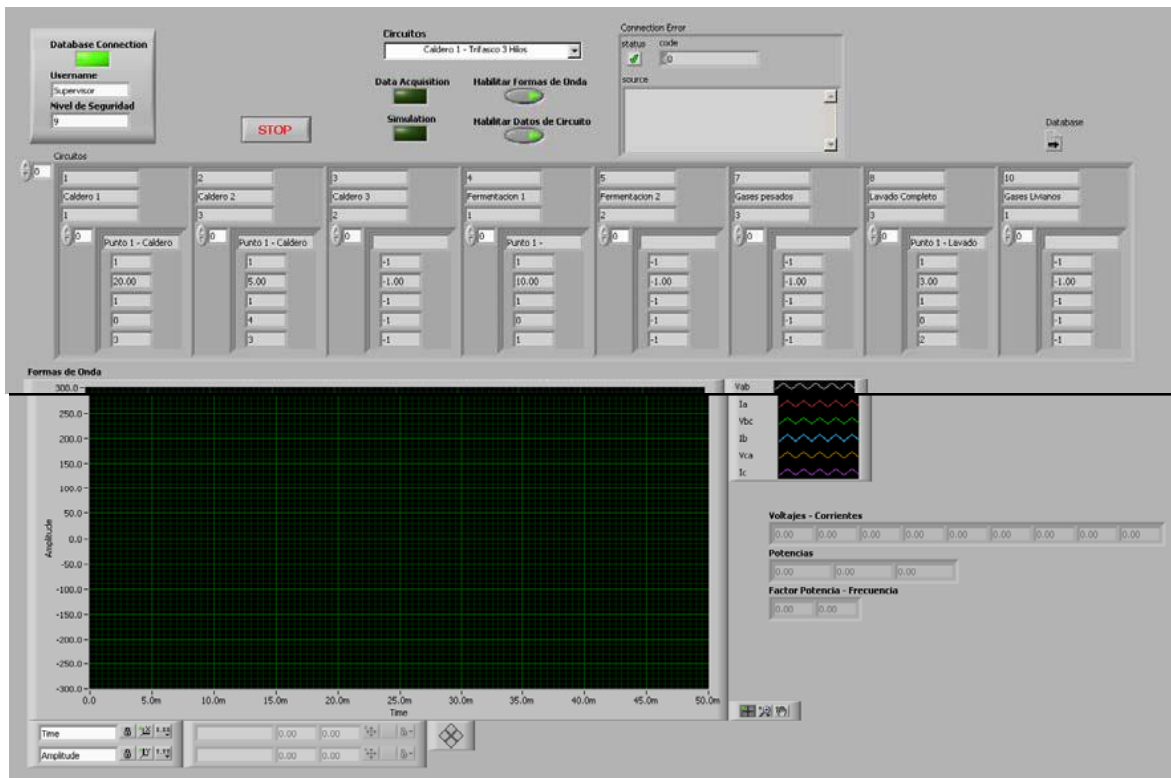


Figura 4.11 – Panel Principal de la Aplicación de Monitoreo

4.2.4.2. Interfaces de Aplicación Cliente en Web

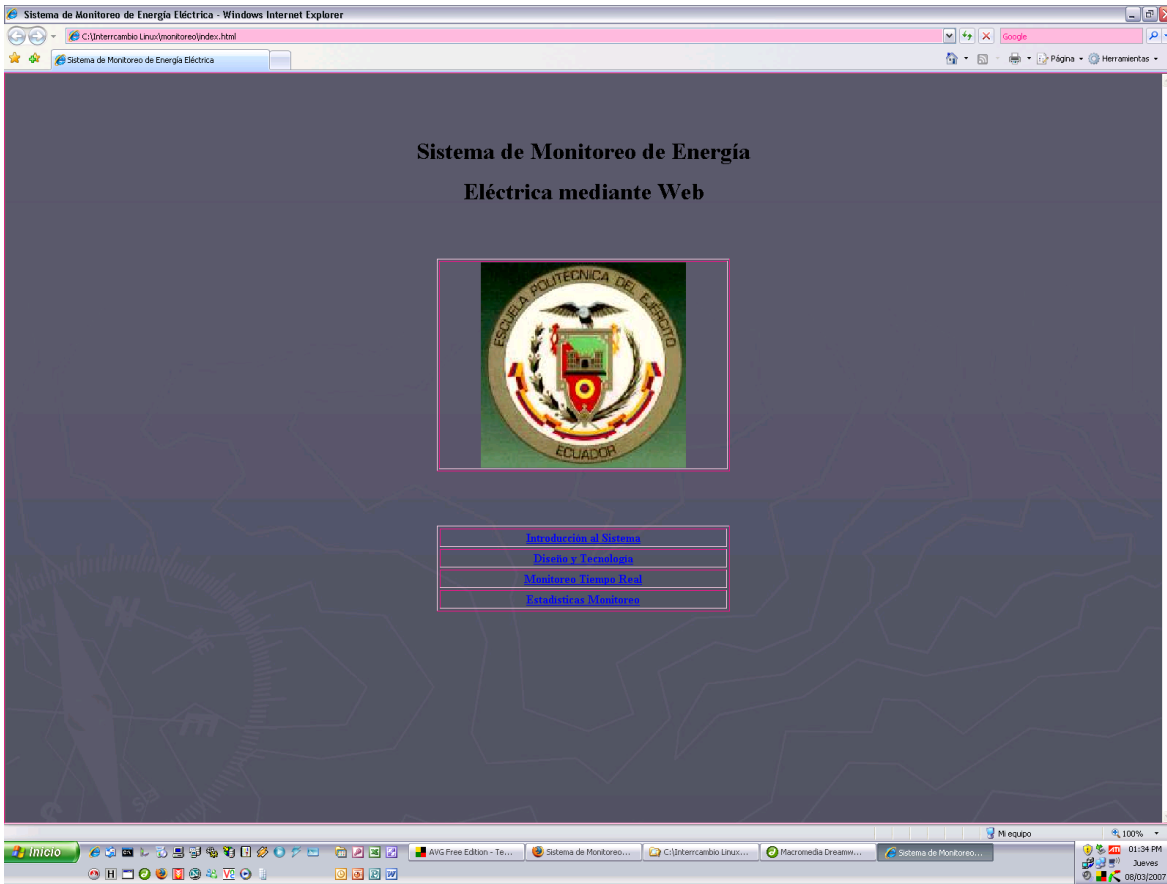


Figura 4.12 – Página Principal de la Aplicación de Cliente

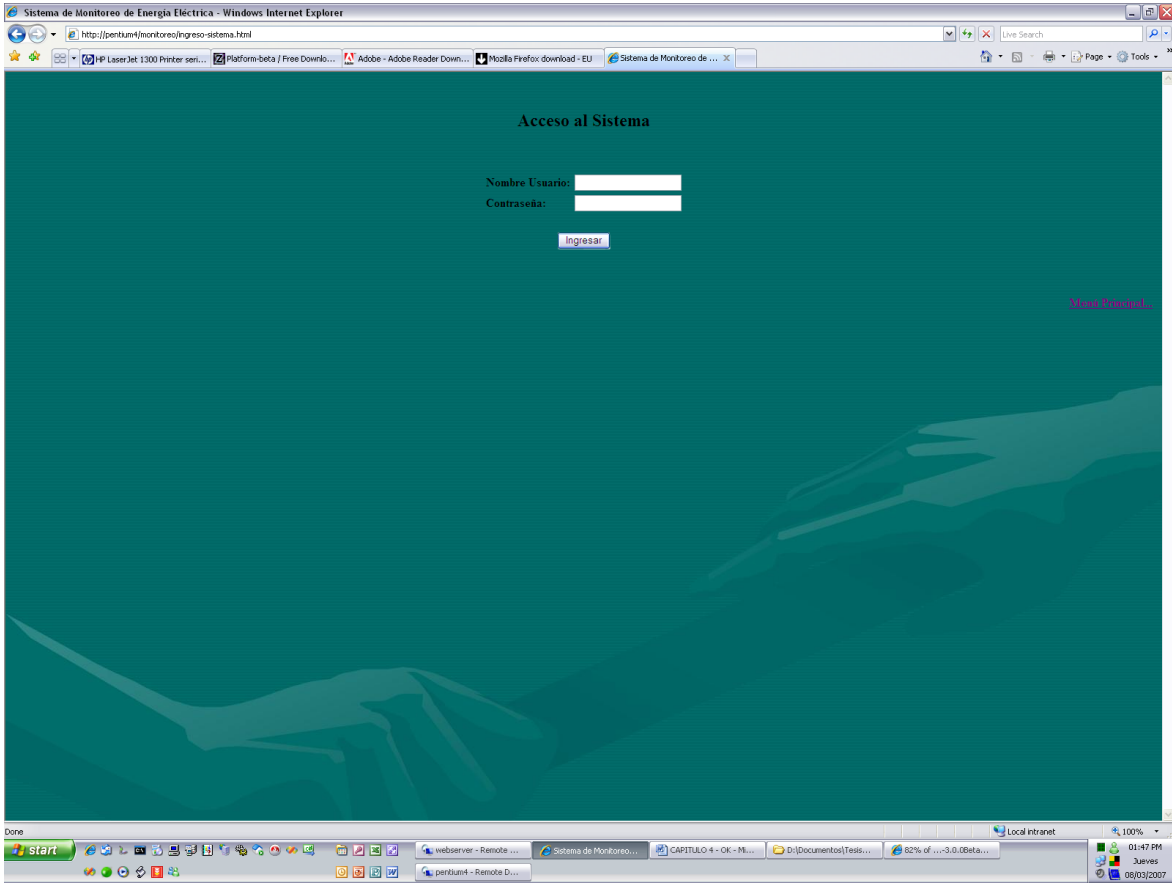


Figura 4.14 – Página Web para Verificación de Usuario

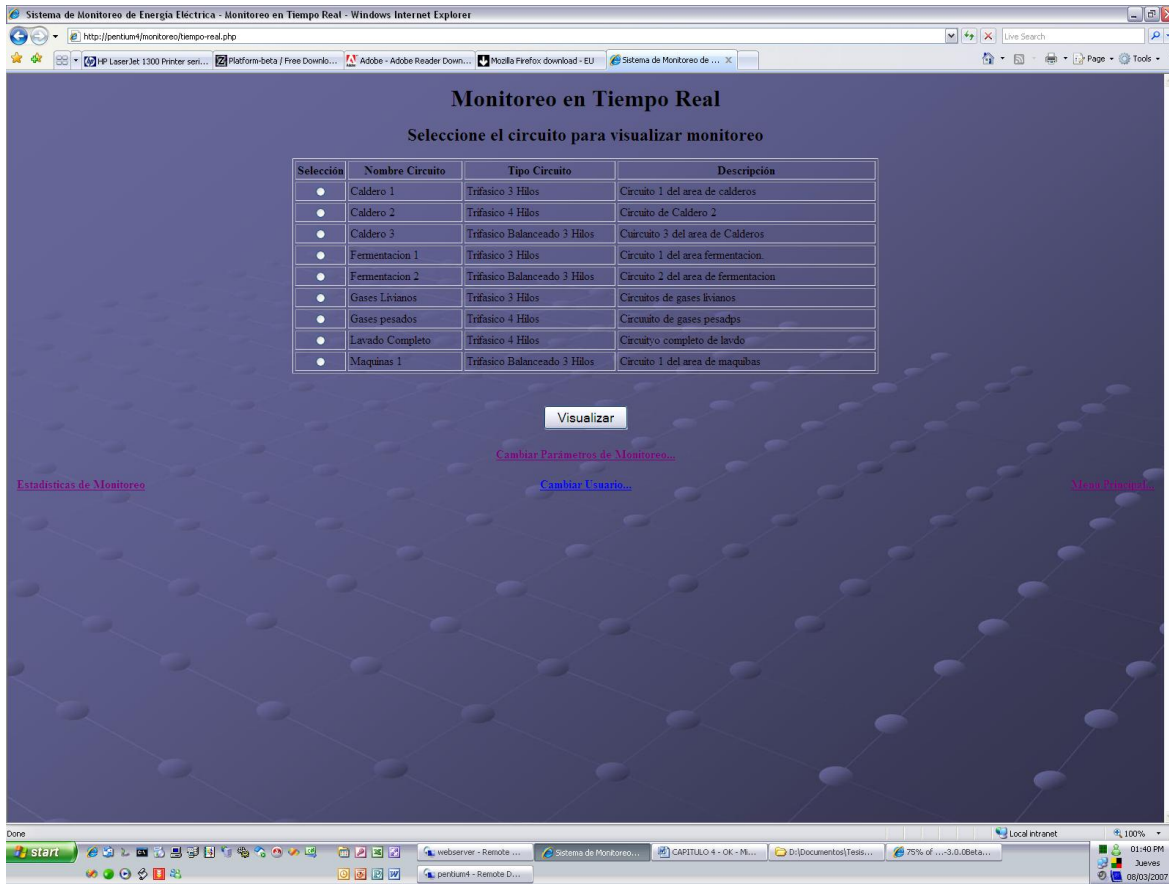


Figura 4.15 – Página Web de Mediciones en Tiempo Real para Seleccionar el Circuito

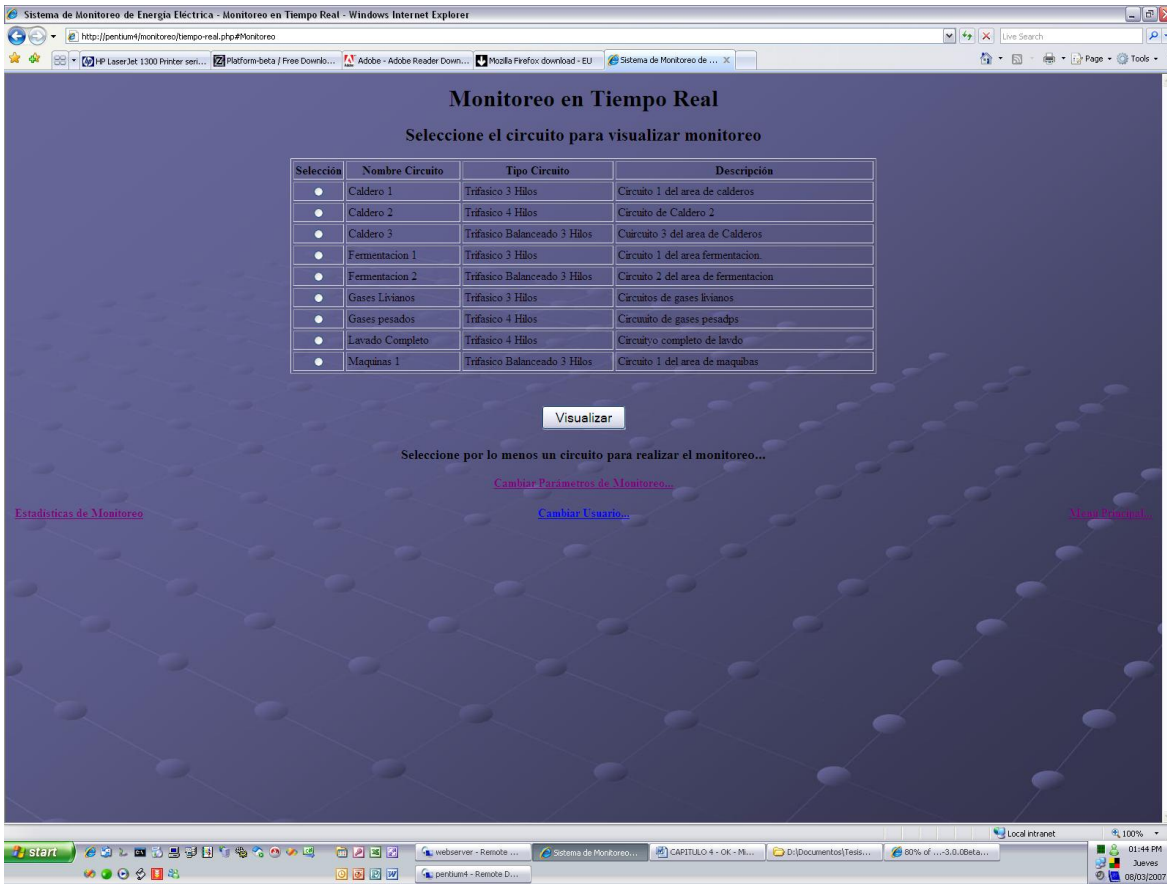


Figura 4.16 – Página Web de Mediciones en Tiempo Real, mensaje de circuito no seleccionado para visualización

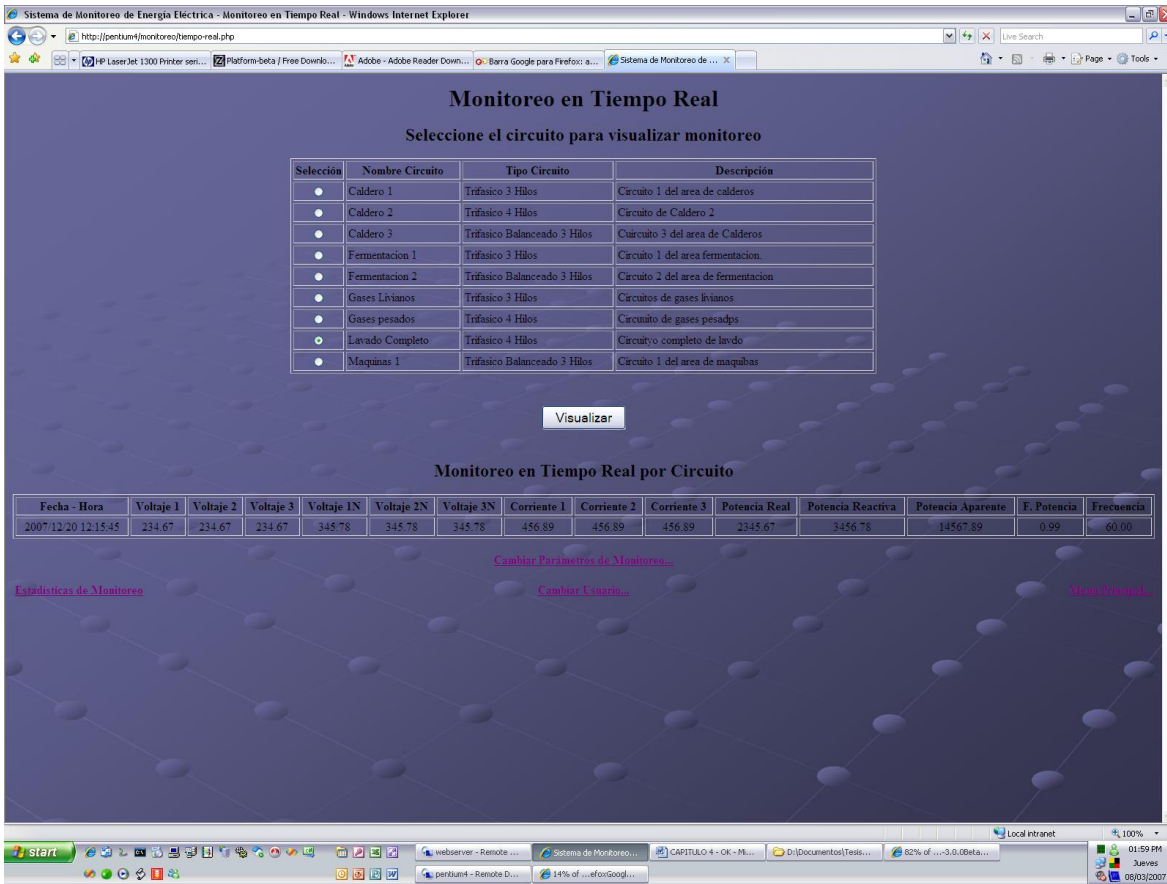


Figura 4.17 – Página Web de Mediciones en Tiempo Real con Datos de un Circuito

Sistema de Monitoreo de Energía Eléctrica - Estadísticas Monitoreo - Windows Internet Explorer

http://pentium4/monitoreo/estadisticas-monitoreo.php

Seleccione el circuito para visualizar estadísticas

Selección	Nombre Circuito	Tipo Circuito	Descripción
<input type="checkbox"/>	Caldero 1	Trifásico 3 Hilos	Circuito 1 del area de calderos
<input type="checkbox"/>	Caldero 2	Trifásico 4 Hilos	Circuito de Caldero 2
<input type="checkbox"/>	Caldero 3	Trifásico Balanceado 3 Hilos	Circuito 3 del area de Calderos
<input type="checkbox"/>	Fermentacion 1	Trifásico 3 Hilos	Circuito 1 del area fermentacion.
<input type="checkbox"/>	Fermentacion 2	Trifásico Balanceado 3 Hilos	Circuito 2 del area de fermentacion
<input type="checkbox"/>	Gases Livianos	Trifásico 3 Hilos	Circuitos de gases livianos
<input type="checkbox"/>	Gases pesados	Trifásico 4 Hilos	Circuito de gases pesados
<input type="checkbox"/>	Lavado Completo	Trifásico 4 Hilos	Circuito completo de lavado
<input type="checkbox"/>	Maquinas 1	Trifásico Balanceado 3 Hilos	Circuito 1 del area de maquinas

Fecha Inicial: Fecha Final:

[Monitoreo en Tiempo Real](#) [Cambiar Usuario...](#) [Menú Principal...](#)

Done

start

webserver - Remote ... Sistema de Monitoreo... CAPITULO 4 - CK - M... D:\Documentos\Tesis... 62% of ...-3.0.0beta... Local Intranet 100% 01:49 PM Jueves 08/03/2007

Figura 4.18 – Página Web de Datos Históricos, selección de circuito y fechas de búsqueda



Figura 4.19 – Página Web de Datos Históricos, verificación de ingreso de datos de búsqueda

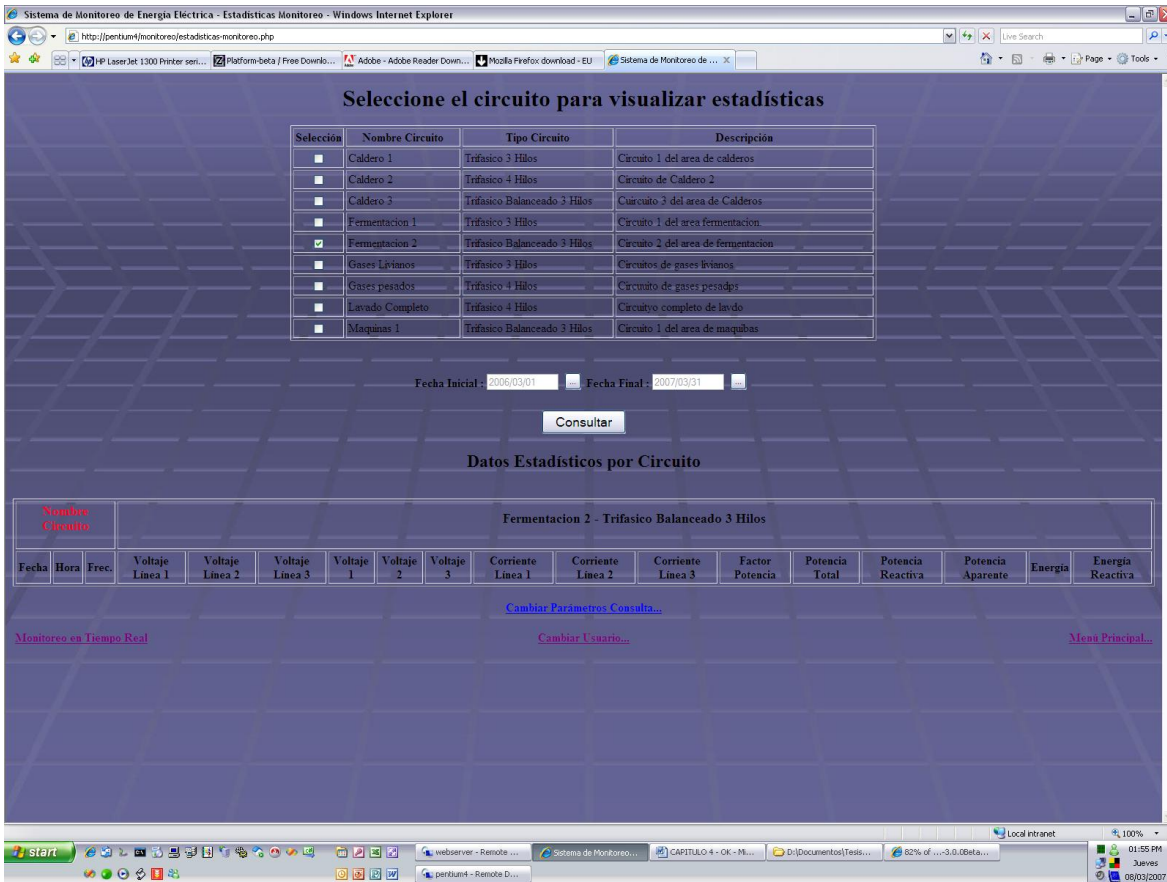


Figura 4.20 – Página Web de Datos Históricos – Datos Consultados en la DBMS

4.2.5. Construcción de Procedimientos

Para la construcción de procedimientos o funciones se implementaron mediante instrumentos virtuales; los cuales poseen sus parámetros de entrada y salida. Además, es importante mencionar que LabVIEW 6i no posee soporte directo para operaciones con bases de datos, por lo tanto fue necesario implementar funciones que realicen cada una de estas tareas de selección, inserción, eliminación y actualización mediante la utilización de llamadas a funciones de librerías externas compartidas que fueron desarrolladas en C y compiladas utilizando KDevelop 3.3.

4.2.5.1. Procedimientos en LabVIEW 6i – Aplicación Monitoreo

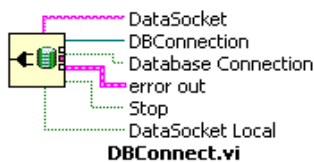


Figura 4.21 – VI de Conexión a la DBMS (Sintaxis)

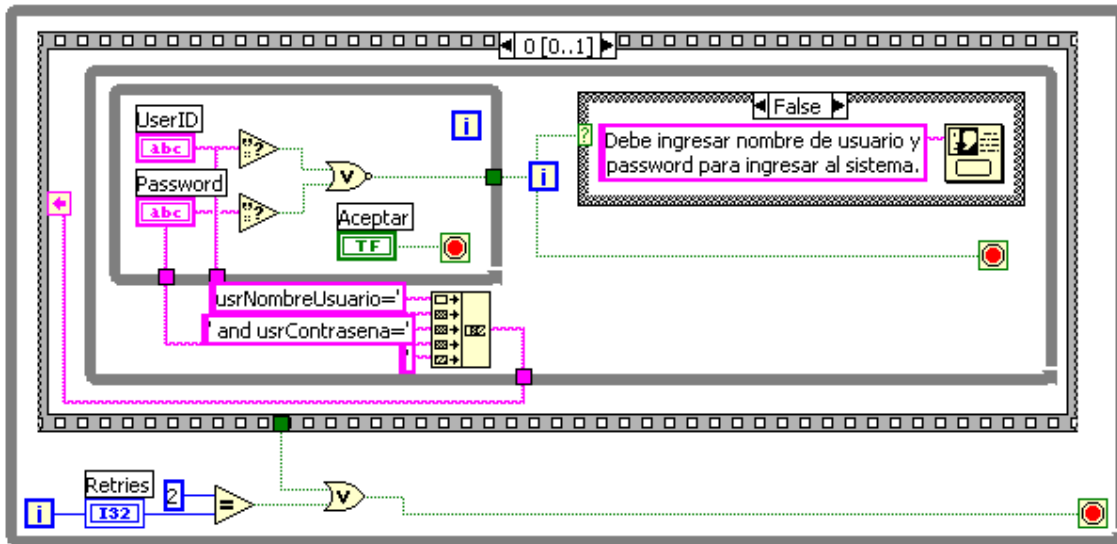


Figura 4.22 – Programación del VI de Verificación de Usuarios

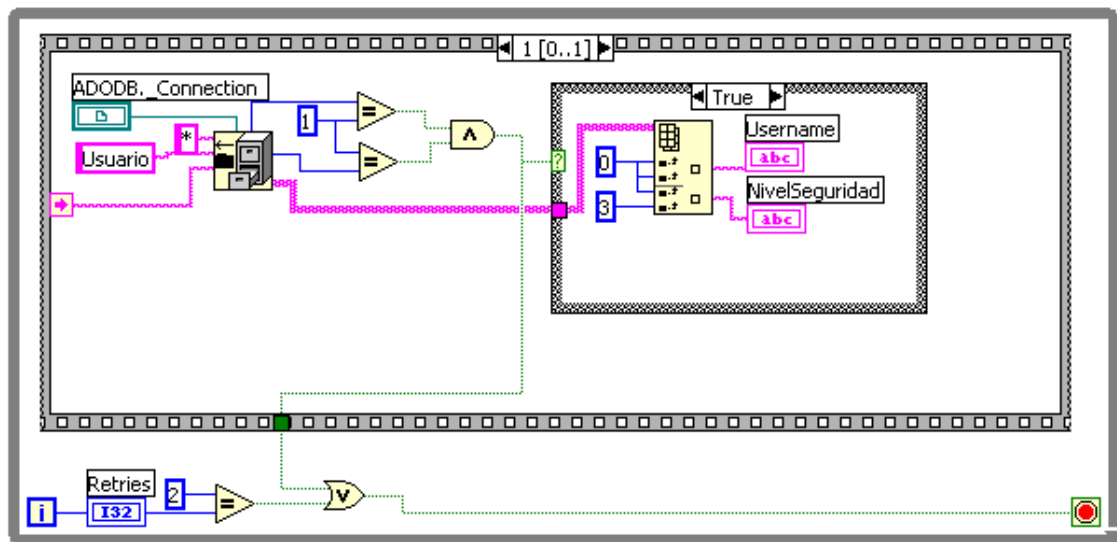


Figura 4.23 – Continuación Programación del VI de Verificación de Usuarios

Para la ejecución de las operaciones de conteo, selección, inserción, eliminación y actualización con los registros de la base de datos se ha tenido que crear también instrumentos virtuales (funciones) que las implementen. A continuación se muestra las funciones.

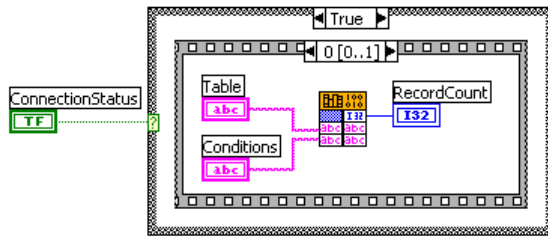


Figura 4.24 – Código para Conteo de Registros

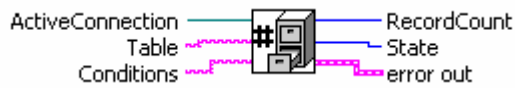


Figura 4.25 – VI para Conteo de Registros (Sintaxis)

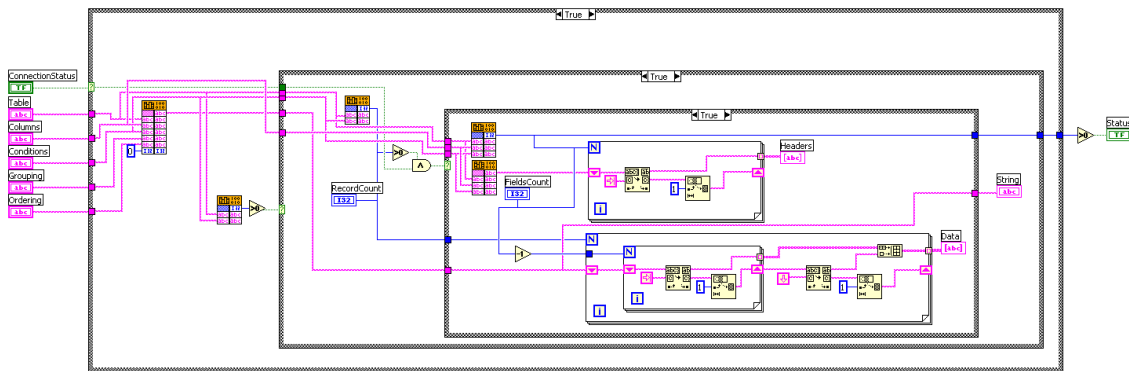


Figura 4.26 – Código para Selección de Columnas y Registros

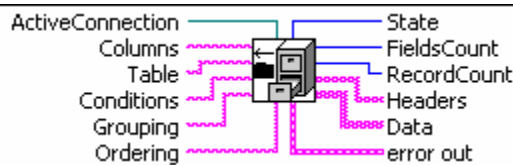


Figura 4.27 – VI para Selección de Columnas y Registros (Sintaxis)

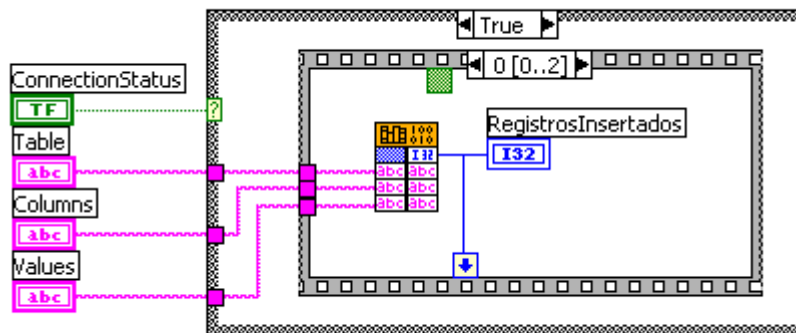


Figura 4.28 – Código para Inserción de Registros

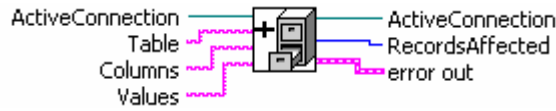


Figura 4.29 – VI para Inserción de Registros (Sintaxis)

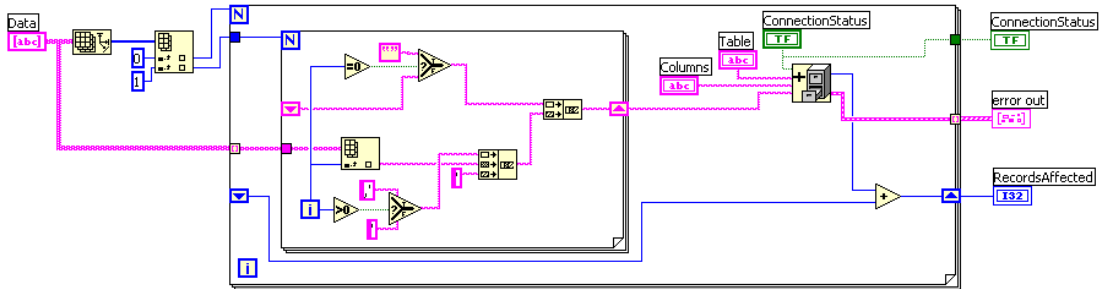


Figura 4.30 – Código para Inserción de Matriz de Registros

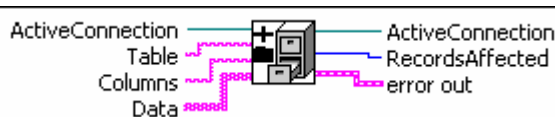


Figura 4.31 – VI para Inserción de Matriz de Registros (Sintaxis)

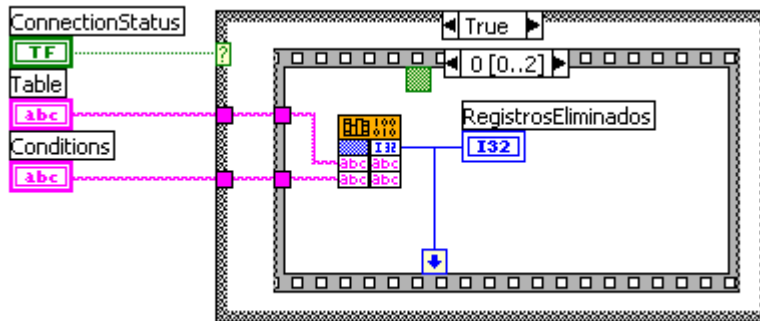


Figura 4.32 – Código para Eliminación de Registros

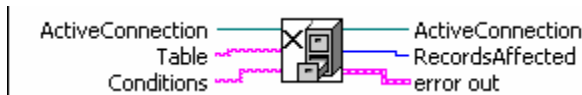


Figura 4.33 – VI para Eliminación de Registros (Sintaxis)

A continuación se muestra el código de programación para las interfaces de usuario donde se realiza la administración de información de usuarios, áreas de planta, circuitos trifásicos y puntos de medición de cada circuito.

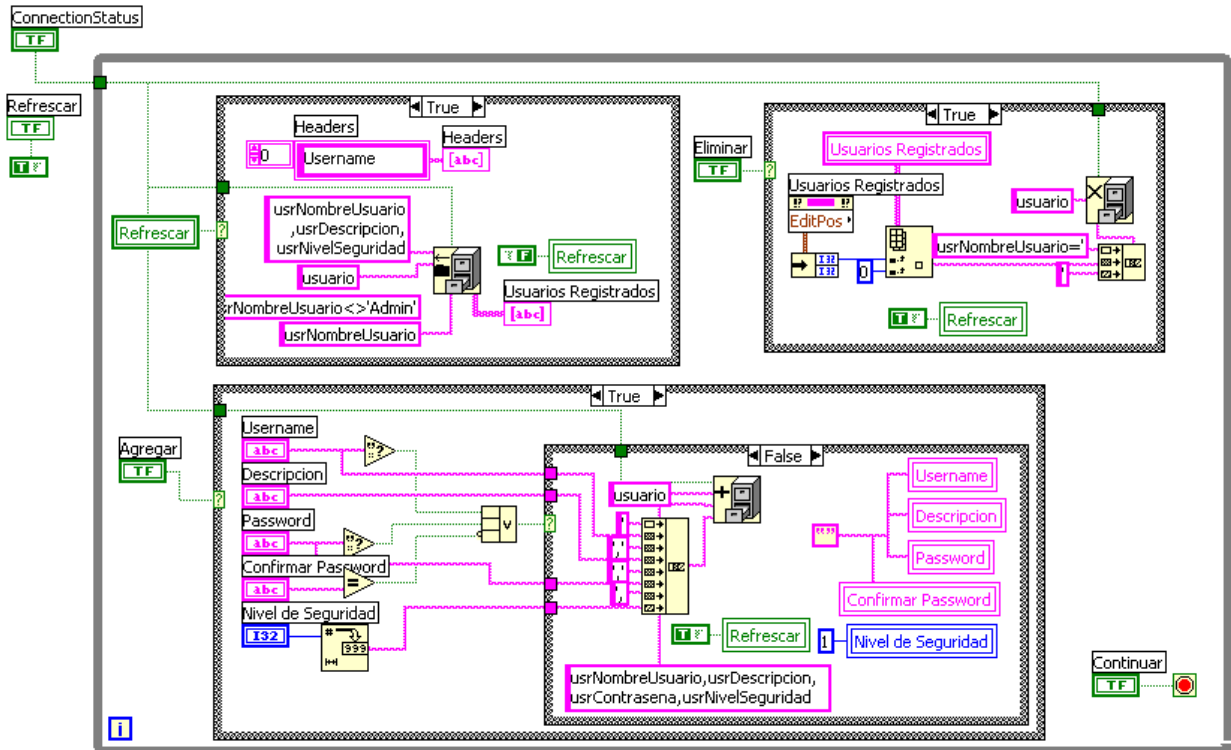


Figura 4.34 – Programación de Interfase de Configuración de Usuarios

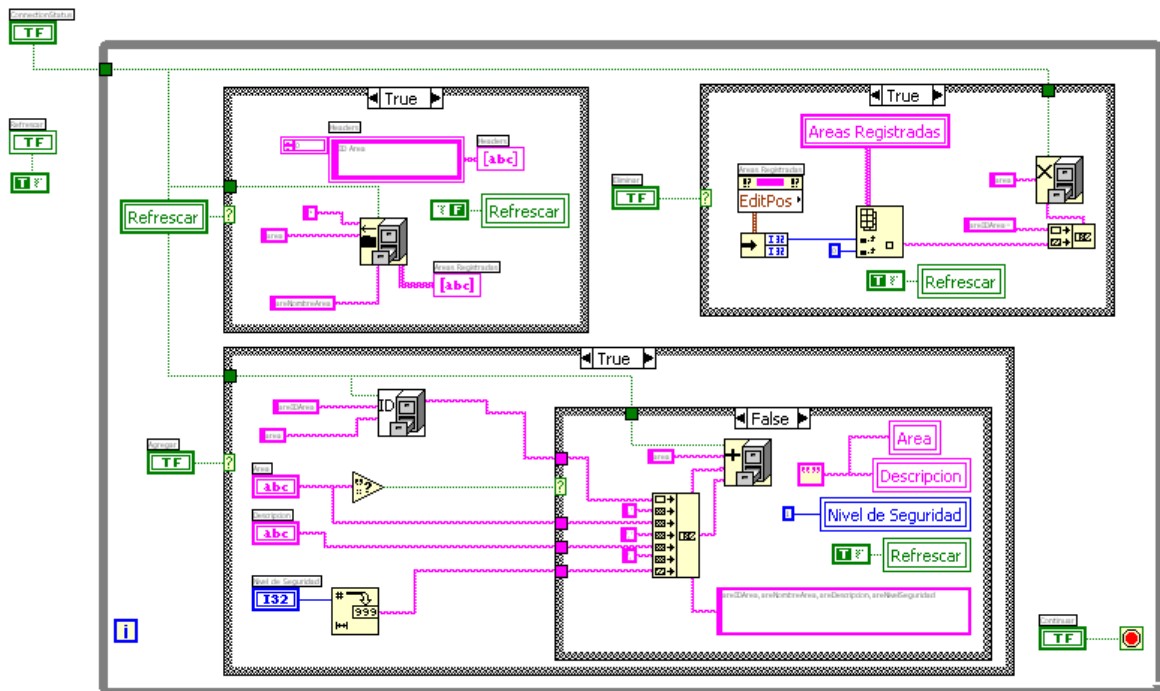


Figura 4.35 – Programación de Interfase de Configuración de Áreas

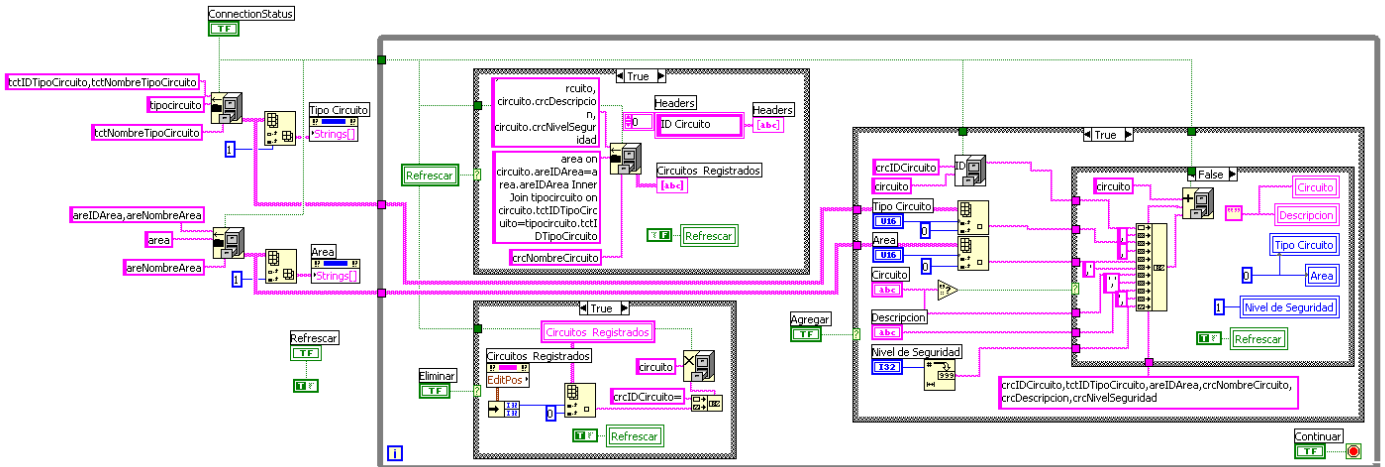


Figura 4.36 – Programación de Interfase de Configuración de Circuitos

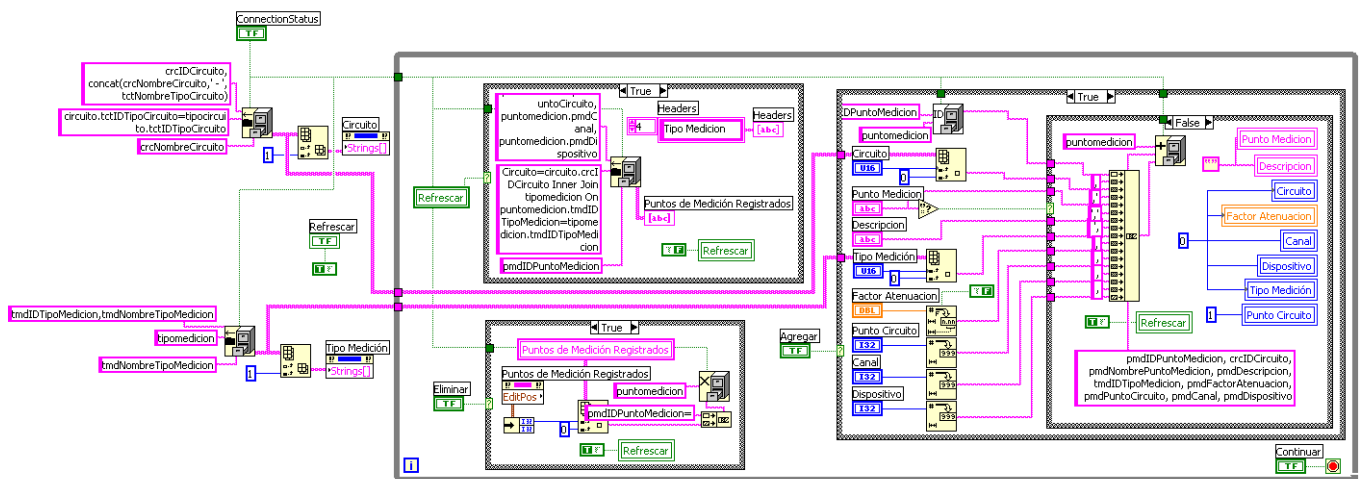


Figura 4.37 – Programación de Interfase de Configuración de Puntos de Medición

A continuación se muestra el código de verificación de los usuarios con el conteo de intentos y sus restricciones.

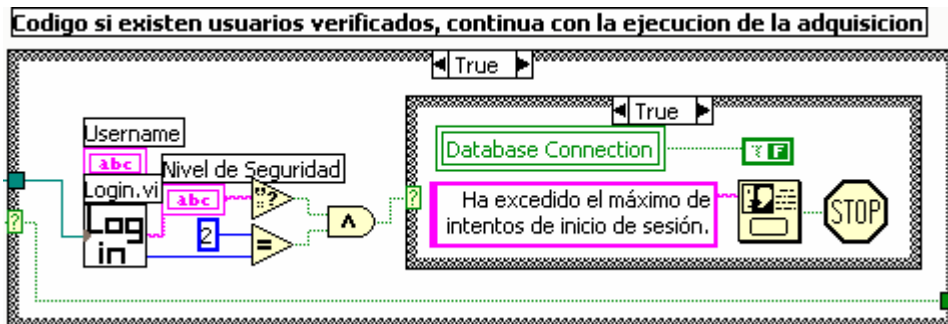


Figura 4.38 – Programación para Obtener los Tres Intentos de Inicio de Sesión

En un lenguaje escrito sea estructurado o de objetos, lo único necesario para utilizar una función es llamarla y pasarle los parámetros que necesita. Se verá la forma tan singular en que funciona este lenguaje cuando hace una llamada a un instrumento virtual (función).

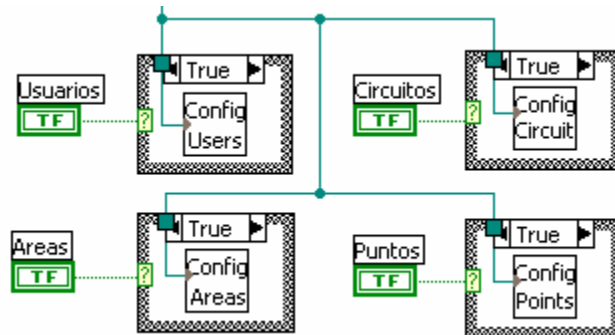


Figura 4.39 – Programación para Botones de llamada a Instrumentos Virtuales

Como se mencionó anteriormente, la aplicación necesita poseer su esquema de seguridad y sus formas de realizar auditorias; para lo cual se implementó los permisos de administración para que los cambios de configuración sólo los pueda realizar un Administrador. Además se implementó un registro de los inicios de sesión de usuario para saber que usuario entra o sale y en que momento..

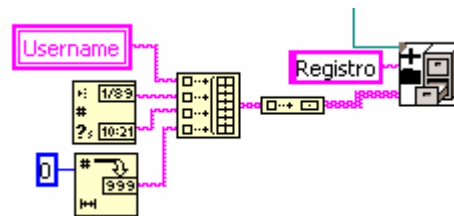


Figura 4.40 – Programación de Inserción de Registros de Auditoría

Una vez que se realizó la verificación de usuario y la inserción de registros para auditoria se proceda a la carga inicial de información y esto es la información de configuración de cada circuito.

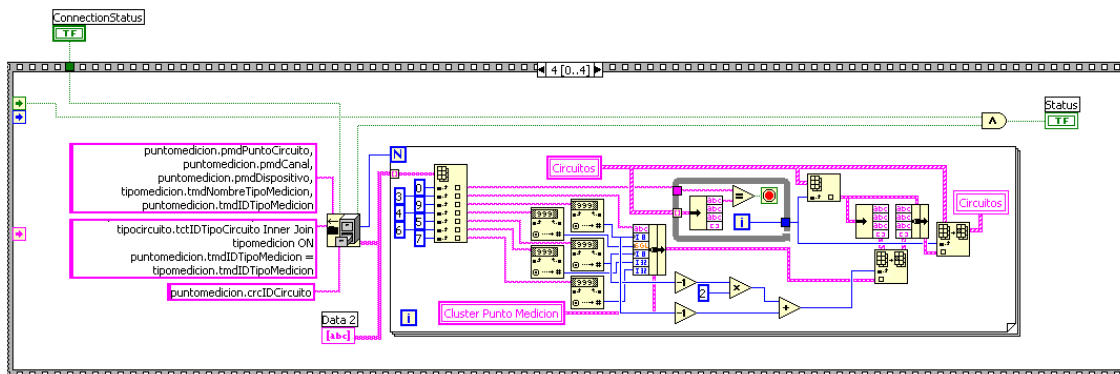


Figura 4.41 – Programación para Cargar los Circuitos con sus Puntos de Medición

Una vez que se han cargado todos los circuitos configurados, el sistema procede a realizar una verificación del estado de las tarjetas de adquisición de datos y a realizar un registro de su estado en la base de datos.

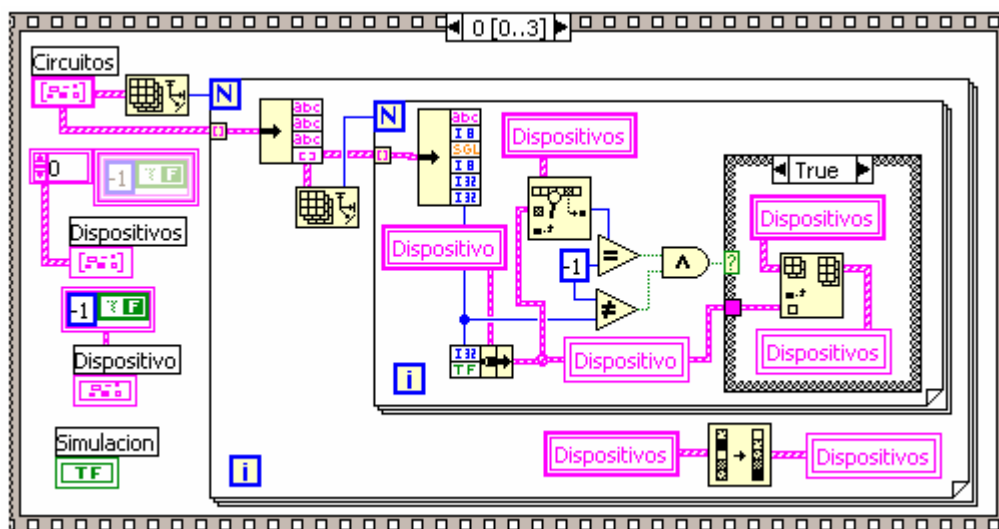


Figura 4.42 – Programación para Verificación del Estado de Tarjetas DAQ

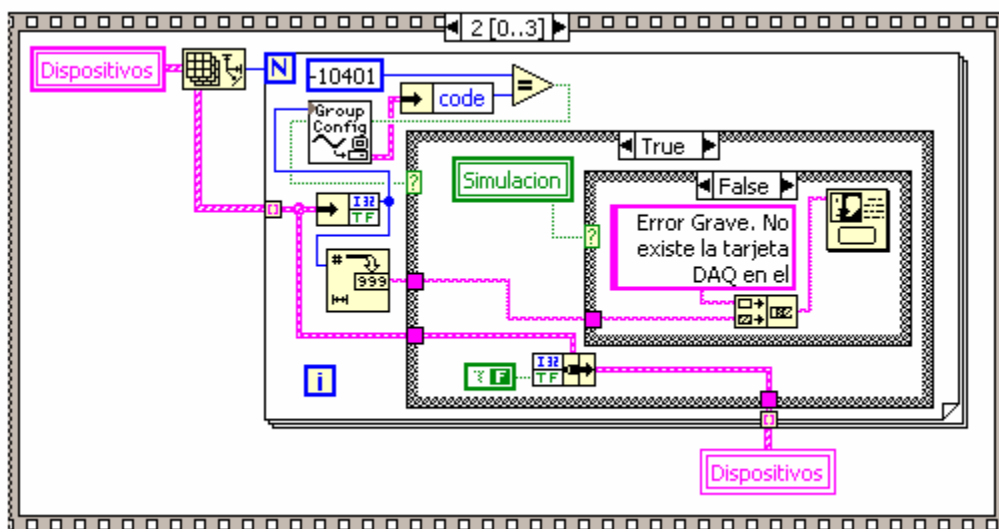


Figura 4.43 – Programación para Creación de Matriz de Estado de Tarjetas DAQ

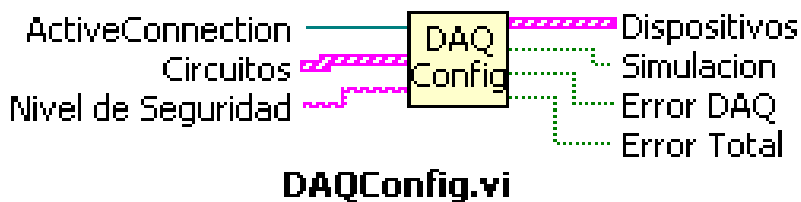


Figura 4.44 – VI para Creación de Matriz de Estado de Tarjetas DAQ (Sintaxis)

Una vez obtenido el estado de las tarjetas DAQ se avisa al usuario cuando han existido errores en las mismas, mediante una luz parpadeante roja. Luego se procede a realizar la adquisición de datos o simulación propiamente dicha, para esto se pasa a una función una matriz de los circuitos configurados y otra matriz de los estados de las tarjetas de adquisición de datos.

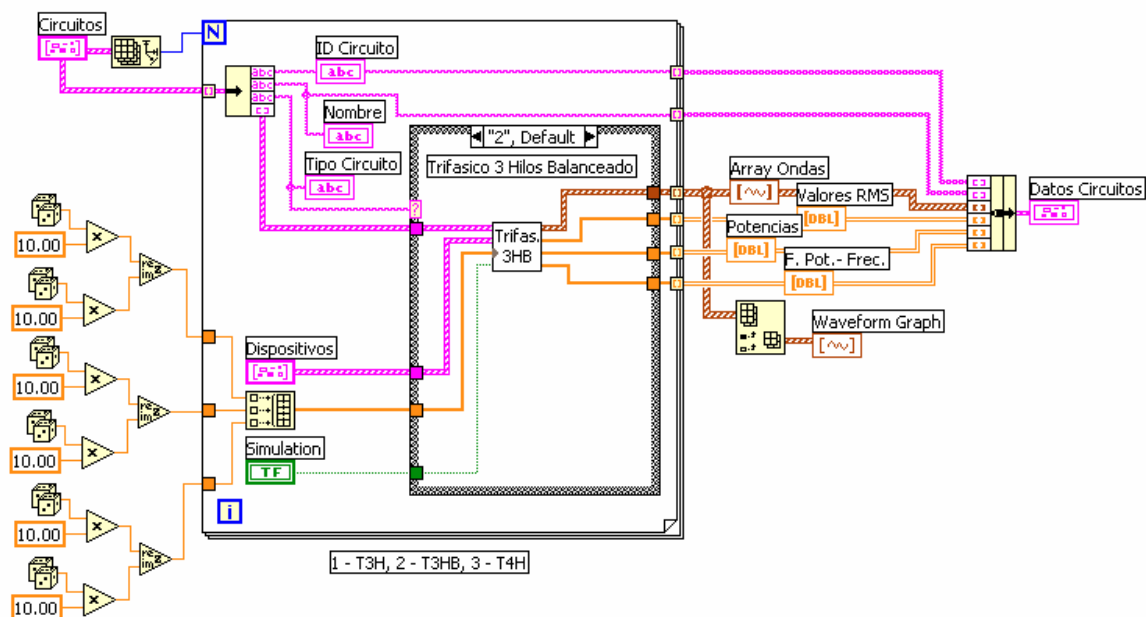


Figura 4.45 – Programación para la Adquisición de Datos de cada Circuito

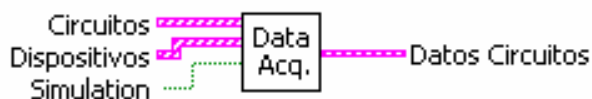


Figura 4.46 – VI para Adquisición de Datos (Sintaxis)

Una vez tomados algunos datos principales se procede a la adquisición de datos propiamente dicha, eligiendo en cada caso el tipo de análisis que se realizará de acuerdo al tipo de circuito que vaya ser monitoreado; a continuación se muestran las funciones para cada tipo de circuito.

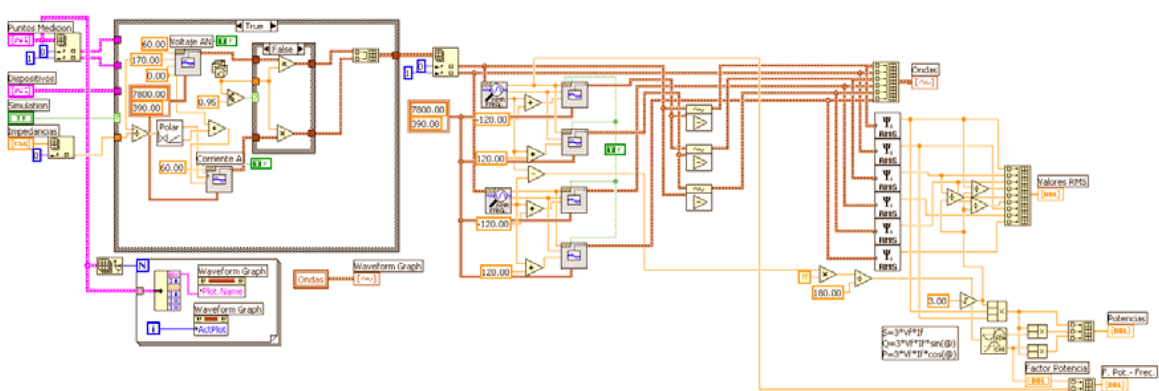


Figura 4.47 – Programación para Circuito Trifásico Balanceado de 3 Hilos en Simulación

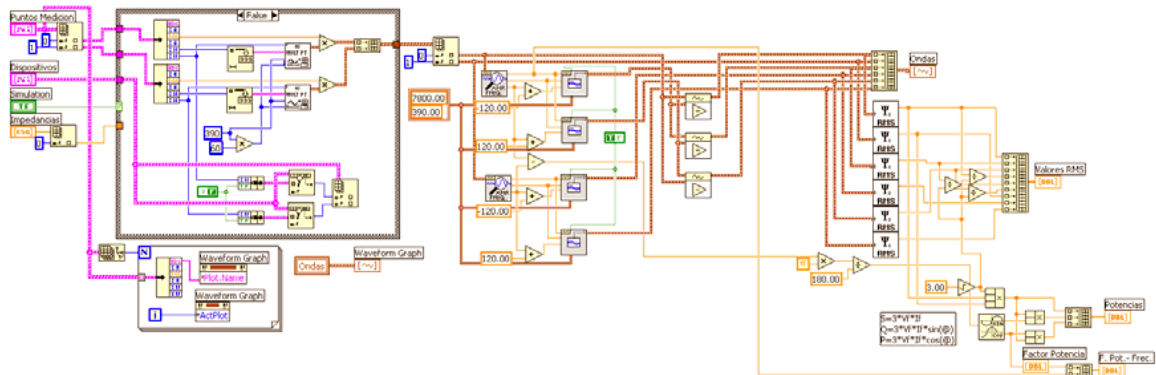


Figura 4.48 – Programación para Circuito Trifásico Balanceado de 3 Hilos en Adquisición de Datos

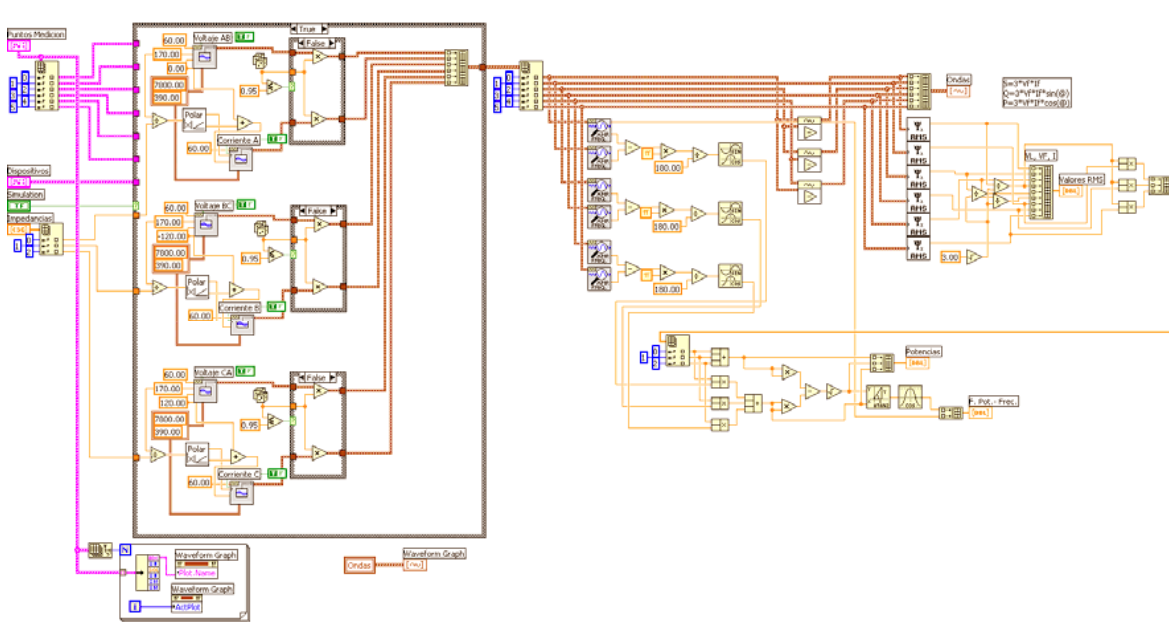
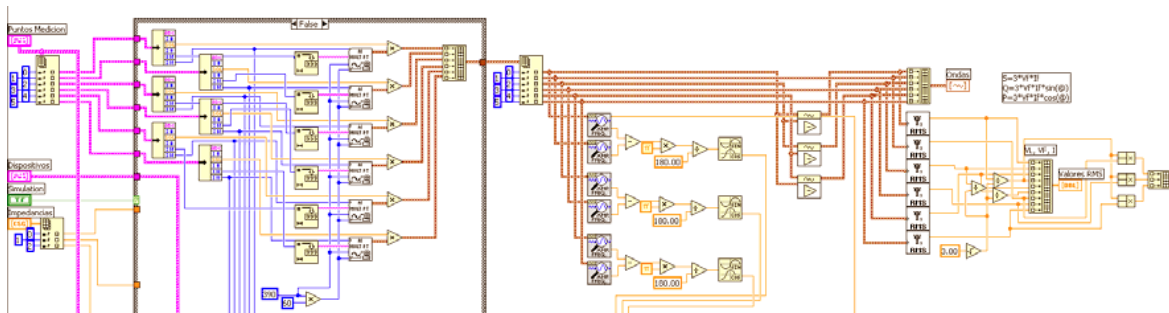


Figura 4.49 – Programación para Circuito Trifásico de 4 Hilos en Simulación



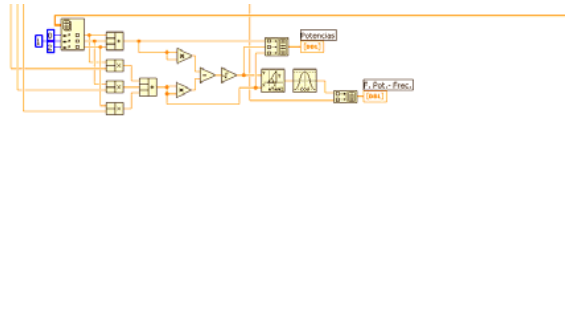
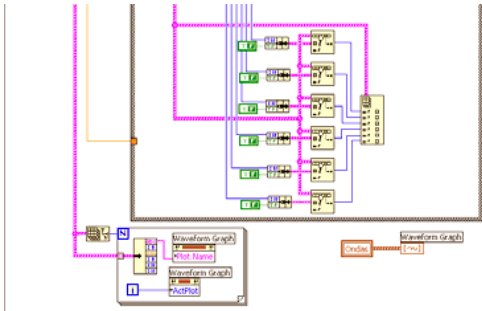


Figura 4.50 – Programación para Circuito Trifásico de 4 Hilos en Adquisición de Datos

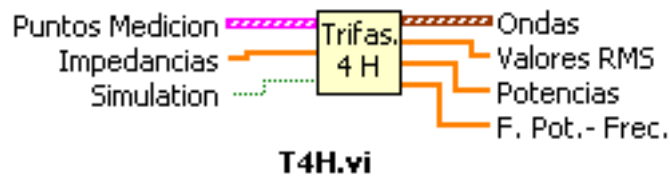


Figura 4.51 – VI para Circuito Trifásico de 4 Hilos (Sintaxis)

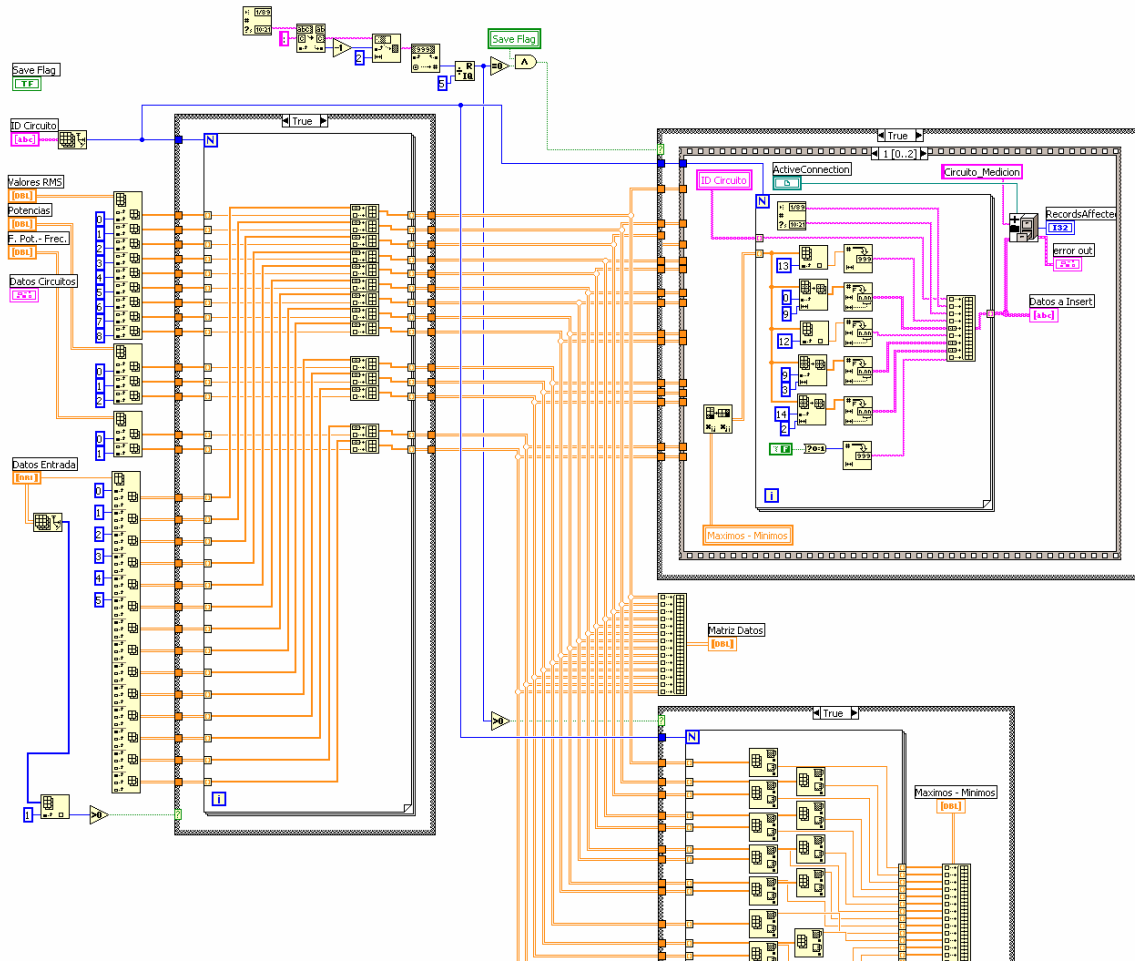


Figura 4.52 – Programación para Obtención de Máximos y Mínimos de los Datos

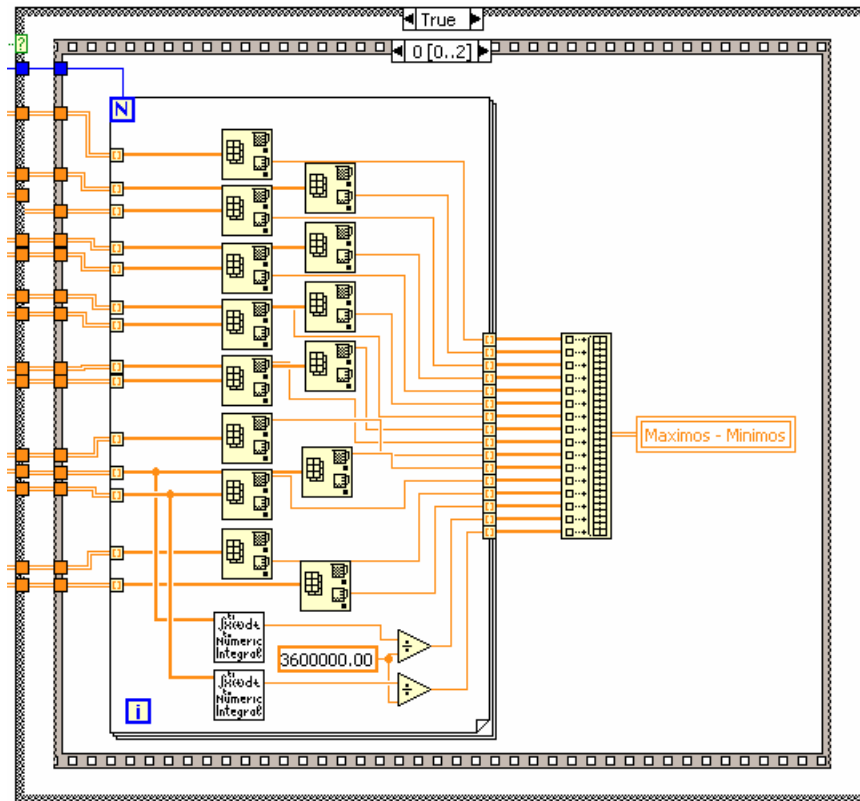


Figura 4.53 – Obtención de Máximos y Mínimos Finales y Cálculo de Energías Totales

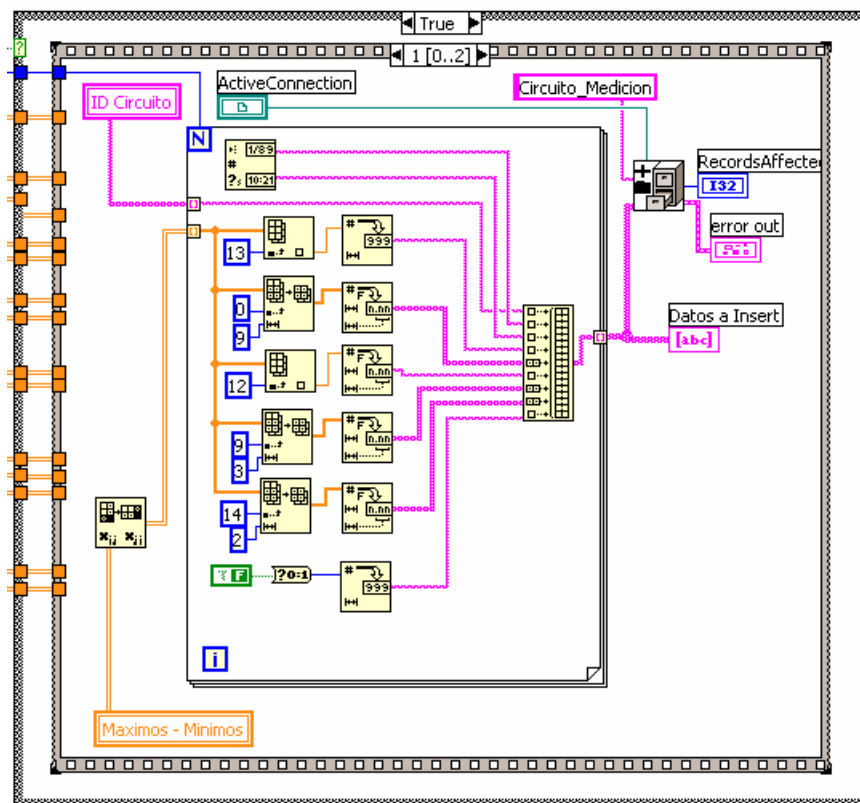


Figura 4.54 – Programación para Almacenamiento de Datos Críticos

Una vez realizada la adquisición de datos se procede a realizar la obtención de máximos y mínimos críticos y finalmente el almacenamiento en la base de datos. A continuación el código que realiza todas esas funciones.

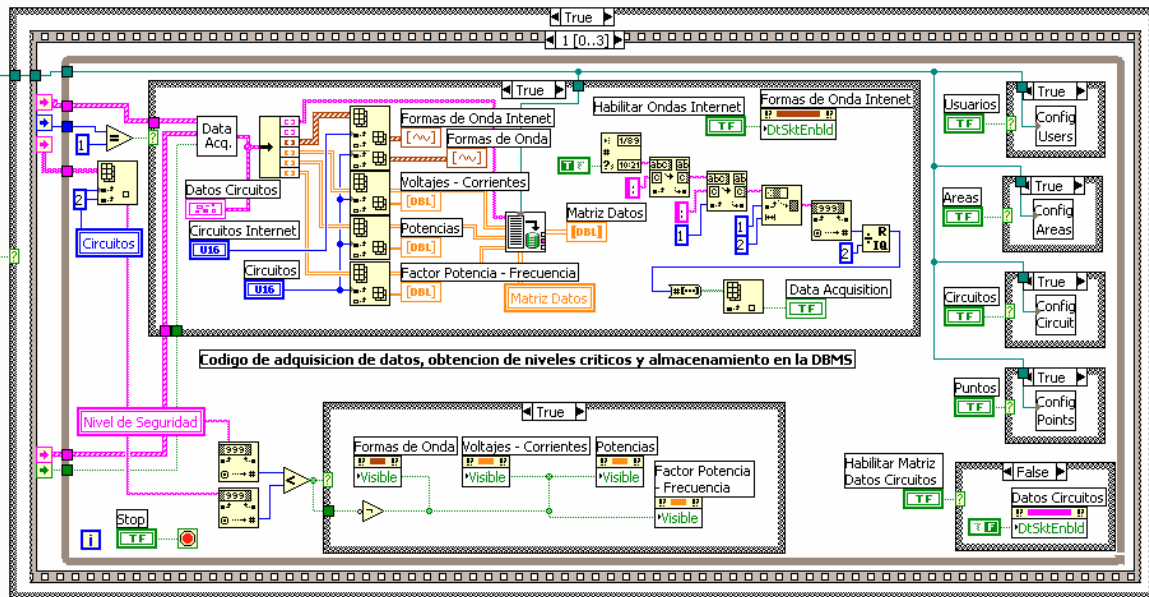


Figura 4.55 – Programación Completa del Ciclo de Adquisición de Datos y Almacenamiento

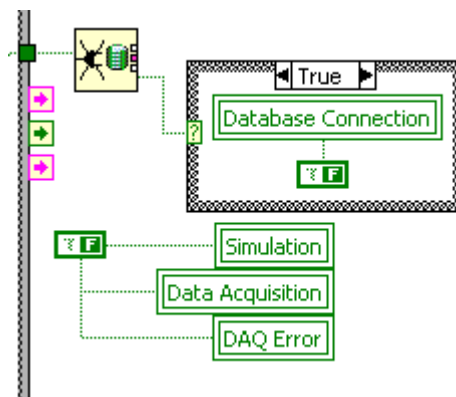


Figura 4.56 – Desconexión de la DBMS

4.2.5.2. Procedimientos en C – Controladores de Dispositivos DAQ

Debido a que no existen controladores para las tarjetas de adquisición de datos proveídos por National Instruments para la plataforma Linux se ha tenido que recurrir a aplicaciones de terceros. Para este proyecto se ha utilizado un conjunto de interfaces y controladores en forma de módulos de kernel que ha desarrollado el proyecto Comedi en lenguaje C, también como un esfuerzo y apoyo a la comunidad de software libre. El proyecto Comedi que significa, interfaces de dispositivos para control y mediciones sobre Linux; se dedica al desarrollo de controladores, herramientas y librerías para adquisición de datos en código abierto.

El proyecto consta de tres componentes principales:

- **Comedi**, es una colección de controladores para una variedad de tarjetas comunes de adquisición de datos.
- **Comedilib**, es una librería de espacio de usuario que provee una interfase amigable para el desarrollador para los dispositivos Comedi.
- **Kcomedilib**, es un módulo de kernel de Linux (distribuido con Comedi) que provee la misma interfase que Comedilib en el espacio de kernel.

Entre las características que posee el proyecto Comedi se puede destacar las siguientes:

- Soporte de tiempo real integrado para la mayoría de hardware
- Librería de alto nivel (comedilib)
- Independencia de dispositivos a nivel de aplicación
- Funciona con los kernels de Linux 2.0, 2.2, 2.4 y 2.6

Comedi Project soporta dispositivos de los siguientes fabricantes:

- ADDI-DATA GMBH
- Advantech
- ADLink
- Analog Devices
- Data Translation
- Intelligent Instrumentation
- IOtech
- Incite Technology Limited
- Keithley Metrabyte
- [Measurement Computing](#) (formerly ComputerBoards)
- National Instruments
- Quanser Consulting
- [SuperLogics](#) (formerly Quatech)

De esta manera se puede ver que existe una amplia gama de dispositivos soportados por el proyecto Comedi, lo cual permite que se pueda considerar otras soluciones para implementación de instrumentación virtual que pertenezcan al software libre.

Para más información puede referirse a su sitio web: <http://www.comedi.org>.

4.2.5.3. Procedimientos en C – Conexión Base de Datos MySQL

Debido a que LabVIEW 6i no tiene integrado el soporte para conexión y manejo de base de datos fue necesario desarrollar estos componentes como librerías compartidas para que LabVIEW pueda realizar las llamadas de manera adecuada y pueda tener funcionalidad.

KDevelop en su versión 3.3 viene integrado con Mandrake Linux 10.0, razón por la cual se hizo uso de esta herramienta para la creación de las librerías compartidas. KDevelop es un IDE (entorno de desarrollo integrado) gratuito que puede ser instalado sobre el manejador de ventanas KDE que soporta el desarrollo de casi todos los lenguajes soportados por Linux, entre los cuales se tiene:

- ADA
- C
- C++
- Fortran
- Haskell
- Java
- Phyton
- Perl
- Pascal
- Ruby

Cuando se realiza la compilación de la librería compartida necesaria para tener acceso a MySQL desde LabVIEW deben fijarse un par de opciones porque de otro modo no enlazará correctamente la librería. Debe utilizarse como referencia la librería básica de MySQL (libmysql.so) y adicionalmente se debe usar la librería de compresión (z.so). debido a que el servidor de MySQL entrega los datos comprimidos al cliente y este los descomprime para poder trabajar con los mismos.

LabVIEW no permite el uso de librería estáticas pese a que el código es el mismo, sólo puede trabajar con librerías dinámicas; utiliza llamadas a funciones de librería.

A continuación se muestra el código en C para la librería de acceso a datos para LabVIEW 6i, además se encuentran un par de funciones adicionales en otra librería ya que el lenguaje C estándar no contiene rutinas de transformación de tipos de datos numéricos a cadenas de texto.

Archivo: librería_mysql.h

```
#ifndef _LIBRERIA_MYSQL_H
#define _LIBRERIA_MYSQL_H

unsigned int VerificarConexion(char* Servidor, char* Usuario, char*
Contrasena);
unsigned int ConectarBaseDatos(char* Servidor, char* Usuario, char*
Contrasena);
void DesconectarBaseDatos();
const char* MensajeError();
int SeleccionarBaseDatos(char* BaseDatos);
char* NombresCampos(char* Tabla, char* Campos, char* Filtro);
long ContarRegistros(char* Tabla, char* Filtro);
int ContarCampos(char* Tabla, char* Campos, char* Filtro);
long RegistrosAfectados();
long Insertar(char* Tabla, char* Campos, char* Valores);
```

```

long Eliminar(char* Tabla, char* Filtro);
long Actualizar(char* Tabla, char* Valores, char* Filtro);
char* Seleccionar(char* Tabla, char* Campos, char* Filtro, char*
Agrupacion, char* Orden, int Registros);
char* NuevoID(char* Tabla, char* Campo, char* Filtro);
char* EjecutarComando(char* Comando);
char* VersionCliente();
char* VersionServidor();

#endif

```

Archivo: librería_mysql.c

```

#include <stdio.h>
#include <stdlib.h>
#include "/usr/include/mysql/mysql.h"
#include "conversiones.h"

MYSQL *myServerConnect;
char strError[4096], strConsulta[4096];
long lRegistrosAfectados;

unsigned int VerificarConexion(char* Servidor, char* Usuario, char*
Contrasena)
{
    MYSQL* myTestConnect;
    unsigned int iError;

    myTestConnect = mysql_init(NULL);

    if (myTestConnect)
    {
        if (!mysql_real_connect(myTestConnect, Servidor, Usuario, Contrasena,
NULL, 0, NULL, 0))
        {
            strcpy(strError, mysql_error(myTestConnect));

            return mysql_errno(myTestConnect);
        }
        else
        {
            strcpy(strError, "El Servidor de Base de Datos esta funcionando
correctamente.");
            iError = mysql_errno(myTestConnect);
            mysql_close(myTestConnect);

            return iError;
        }
    }
    else
    {
        strcpy(strError, "No existe suficiente memoria para crear el objeto de
conexion a la base de datos.");

        return -1;
    }
}

```

```

}

unsigned int ConectarBaseDatos(char* Servidor, char* Usuario, char*
Contrasena)
{
    myServerConnect = mysql_init(NULL);
    unsigned int iError;

    if (myServerConnect)
    {
        if (!mysql_real_connect(myServerConnect, Servidor, Usuario,
Contrasena, NULL, 0, NULL, 0))
        {
            strcpy(strError, mysql_error(myServerConnect));

            return mysql_errno(myServerConnect);
        }
        else
        {
            strcpy(strError, "Conexion Exitosa al Servidor de Base de Datos.");
            iError = mysql_errno(myServerConnect);

            return iError;
        }
    }
    else
    {
        strcpy(strError, "No existe suficiente memoria para crear el objeto de
conexion a la base de datos.");

        return -1;
    }
}

void DesconectarBaseDatos()
{
    mysql_close(myServerConnect);
}

const char* MensajeError()
{
    return strError;
}

int SeleccionarBaseDatos(char* BaseDatos)
{
    int iError;

    if (myServerConnect)
    {
        iError = mysql_select_db(myServerConnect, BaseDatos);

        if (iError == 0)
            strcpy(strError, "Base de Datos encontrada.");
        else
            strcpy(strError, mysql_error(myServerConnect));
    }
}

```

```

        return iError;
    }
    else
        return -1;
}

char* NombresCampos(char *Tabla, char* Campos, char* Filtro)
{
    int iError, iColumnas, iContador, iTotalEspacio;
    MYSQL_RES *resRegistros;
    MYSQL_FIELD *fldCampos;
    char* strCampos = NULL;

    if (!Tabla && !Filtro)
        return NULL;
    else
    {
        strcpy(strConsulta, "Select ");
        if (strlen(Campos) > 0)
            strcat(strConsulta, Campos);
        else
            strcat(strConsulta, "*");
        strcat(strConsulta, " from ");
        strcat(strConsulta, Tabla);

        if (strlen(Filtro) > 0)
        {
            strcat(strConsulta, " where ");
            strcat(strConsulta, Filtro);
        }
        strcat(strConsulta, " limit 1");
        iError = mysql_real_query(myServerConnect, strConsulta,
            strlen(strConsulta));

        if (iError == 0)
        {
            resRegistros = mysql_store_result(myServerConnect);
            if (resRegistros)
            {
                iColumnas = mysql_num_fields(resRegistros);
                fldCampos = mysql_fetch_fields(resRegistros);
                iTotalEspacio = 1;

                for (iContador = 0; iContador < iColumnas; iContador++)
                {
                    iTotalEspacio += strlen(fldCampos[iContador].name) + 1;
                }
                strCampos = (char*)malloc(iTotalEspacio * sizeof(char));
                strcpy(strCampos, "");

                for (iContador = 0; iContador < iColumnas; iContador++)
                {
                    strcat(strCampos, fldCampos[iContador].name);
                    if (iContador < iColumnas - 1)
                        strcat(strCampos, "\t");
                    else
                        strcat(strCampos, "\n");
                }
            }
        }
    }
}

```

```

    }

    mysql_free_result(resRegistros);
    strcpy(strError, mysql_error(myServerConnect));

    return strCampos;
}
else
{
    strcpy(strError, mysql_error(myServerConnect));

    return NULL;
}
}
else
{
    strcpy(strError, mysql_error(myServerConnect));

    return NULL;
}
}
}

long ContarRegistros(char* Tabla, char* Filtro)
{
    int iError;
    long lRegistros;
    char* strNumero = NULL;
    MYSQL_RES *resRegistros;

    if (!Tabla && !Filtro)
        return -1;
    else
    {
        strcpy(strConsulta, "Select * from ");
        strcat(strConsulta, Tabla);

        if (strlen(Filtro) > 0)
        {
            strcat(strConsulta, " where ");
            strcat(strConsulta, Filtro);
        }
        iError = mysql_real_query(myServerConnect, strConsulta,
            strlen(strConsulta));

        if (!iError)
        {
            resRegistros = mysql_store_result(myServerConnect);

            if (resRegistros)
            {
                lRegistros = (long)mysql_num_rows(resRegistros);
                mysql_free_result(resRegistros);
                strNumero = ltoa(lRegistros);
                strcpy(strError, strNumero);
                strcat(strError, " Registros Encontrados");
            }
        }
    }
}

```



```

        return lRegistros;
    }
    else
    {
        strcpy(strError, mysql_error(myServerConnect));

        return -1;
    }
}
else
{
    strcpy(strError, mysql_error(myServerConnect));

    return -1;
}
}
}

int ContarCampos(char* Tabla, char* Campos, char* Filtro)
{
    int iError, iColumnas;
    char* strNumero;
    MYSQL_RES *resRegistros;

    if (!Tabla && !Filtro)
        return -1;
    else
    {
        strcpy(strConsulta, "Select ");
        if (strlen(Campos) > 0)
            strcat(strConsulta, Campos);
        else
            strcat(strConsulta, "*");
        strcat(strConsulta, " from ");
        strcat(strConsulta, Tabla);

        if (strlen(Filtro) > 0)
        {
            strcat(strConsulta, " where ");
            strcat(strConsulta, Filtro);
        }
        strcat(strConsulta, " limit 1");
        iError = mysql_real_query(myServerConnect, strConsulta,
            strlen(strConsulta));

        if (!iError)
        {
            resRegistros = mysql_store_result(myServerConnect);

            if (resRegistros)
            {
                iColumnas = mysql_num_fields(resRegistros);
                mysql_free_result(resRegistros);
                strNumero = itoa(iColumnas);
                strcpy(strError, strNumero);
                strcat(strError, " Campos Encontrados");
            }
        }
    }
}

```

```

        return iColumnas;
    }
    else
    {
        strcpy(strError, mysql_error(myServerConnect));

        return -1;
    }
}
else
{
    strcpy(strError, mysql_error(myServerConnect));

    return -1;
}
}
}

long Insertar(char* Tabla, char* Campos, char* Valores)
{
    int iError;
    char* strNumero;
    MYSQL_RES *resRegistros;

    if (!Tabla && !Valores)
        return -1;
    else
    {
        strcpy(strConsulta, "Insert into ");
        strcat(strConsulta, Tabla);
        if (strlen(Campos) > 0)
        {
            strcat(strConsulta, " (");
            strcat(strConsulta, Campos);
            strcat(strConsulta, ")");
        }
        strcat(strConsulta, " values(");
        strcat(strConsulta, Valores);
        strcat(strConsulta, ")");
        iError = mysql_real_query(myServerConnect, strConsulta,
            strlen(strConsulta));

        if (!iError)
        {
            resRegistros = mysql_store_result(myServerConnect);

            if (resRegistros)
            {
                mysql_free_result(resRegistros);
                strcpy(strError, mysql_error(myServerConnect));
            }
            else
            {
                if (!mysql_field_count(myServerConnect))
                {
                    lRegistrosAfectados =
(long)mysql_affected_rows(myServerConnect);

```

```

        strNumero = ltoa(lRegistrosAfectados);
        strcpy(strError, strNumero);
        strcat(strError, " Registros Insertados Correctamente");

        return lRegistrosAfectados;
    }
    else
    {
        strcpy(strError, mysql_error(myServerConnect));

        return mysql_errno(myServerConnect);
    }
}
}
else
{
    strcpy(strError, mysql_error(myServerConnect));

    return iError;
}
}
}

long Eliminar(char* Tabla, char* Filtro)
{
    int iError;
    char* strNumero;
    MYSQL_RES *resRegistros;

    if (!Tabla)
        return -1;
    else
    {
        strcpy(strConsulta, "Delete from ");
        strcat(strConsulta, Tabla);

        if (strlen(Filtro) > 0)
        {
            strcat(strConsulta, " where ");
            strcat(strConsulta, Filtro);
        }
        iError = mysql_real_query(myServerConnect, strConsulta,
            strlen(strConsulta));

        if (!iError)
        {
            resRegistros = mysql_store_result(myServerConnect);

            if (resRegistros)
            {
                mysql_free_result(resRegistros);
                strcpy(strError, mysql_error(myServerConnect));
            }
            else
            {
                if (!mysql_field_count(myServerConnect))
                {

```

```

        lRegistrosAfectados =
(long)mysql_affected_rows(myServerConnect);
        strNumero = ltoa(lRegistrosAfectados);
        strcpy(strError, strNumero);
        strcat(strError, " Registros Eliminados Correctamente");

        return lRegistrosAfectados;
    }
    else
    {
        strcpy(strError, mysql_error(myServerConnect));

        return mysql_errno(myServerConnect);
    }
}
}
else
{
    strcpy(strError, mysql_error(myServerConnect));

    return iError;
}
}
}

```

```

long Actualizar(char* Tabla, char* Valores, char* Filtro)
{
    int iError;
    char* strNumero;
    MYSQL_RES *resRegistros;

    if (!Tabla && !Valores)
        return -1;
    else
    {
        strcpy(strConsulta, "Update ");
        strcat(strConsulta, Tabla);
        strcat(strConsulta, " set ");
        strcat(strConsulta, Valores);

        if (strlen(Filtro) > 0)
        {
            strcat(strConsulta, " where ");
            strcat(strConsulta, Filtro);
        }
        iError = mysql_real_query(myServerConnect, strConsulta,
strlen(strConsulta));

        if (!iError)
        {
            resRegistros = mysql_store_result(myServerConnect);

            if (resRegistros)
            {
                mysql_free_result(resRegistros);
                strcpy(strError, mysql_error(myServerConnect));
            }
        }
    }
}

```

```

else
{
    if (!mysql_field_count(myServerConnect))
    {
        lRegistrosAfectados =
(long)mysql_affected_rows(myServerConnect);
        strNumero = ltoa(lRegistrosAfectados);
        strcpy(strError, strNumero);
        strcat(strError, " Registros Actualizados Correctamente");

        return lRegistrosAfectados;
    }
    else
    {
        strcpy(strError, mysql_error(myServerConnect));

        return mysql_errno(myServerConnect);
    }
}
}
else
{
    strcpy(strError, mysql_error(myServerConnect));

    return iError;
}
}
}

```

```

char* Seleccionar(char* Tabla, char* Campos, char* Filtro, char*
Agrupacion, char* Orden, int Registros)

```

```

{
    int iError, iColumnas, iContador, iFilas;
    long lTotalEspacio;
    char* strNumero;
    char* strRegistros = NULL;
    MYSQL_RES *resRegistros;
    MYSQL_ROW rowRegistro;

    if (!Tabla)
        return NULL;
    else
    {
        strcpy(strConsulta, "Select ");
        if (strlen(Campos) > 0)
            strcat(strConsulta, Campos);
        else
            strcat(strConsulta, "*");
        strcat(strConsulta, " from ");
        strcat(strConsulta, Tabla);
        if (strlen(Filtro) > 0)
        {
            strcat(strConsulta, " where ");
            strcat(strConsulta, Filtro);
        }
        if (strlen(Agrupacion) > 0)
        {

```

```

    strcat(strConsulta, " group by ");
    strcat(strConsulta, Agrupacion);
}
if (strlen(Orden) > 0)
{
    strcat(strConsulta, " order by ");
    strcat(strConsulta, Orden);
}
if (Registros > 0)
{
    strNumero = itoa(Registros);
    strcat(strConsulta, " limit ");
    strcat(strConsulta, strNumero);
}
iError = mysql_real_query(myServerConnect, strConsulta,
strlen(strConsulta));

if (iError == 0)
{
    resRegistros = mysql_store_result(myServerConnect);

    if (resRegistros)
    {
        iColumnas = mysql_num_fields(resRegistros);
        lRegistrosAfectados = mysql_num_rows(resRegistros);

        if (lRegistrosAfectados > 0)
        {
            lTotalEspacio = 1;

            for (iFilas = 0; iFilas < lRegistrosAfectados; iFilas++)
            {
                rowRegistro = mysql_fetch_row(resRegistros);
                for(iContador = 0; iContador < iColumnas; iContador++)
                {
                    if (rowRegistro[iContador] != NULL)
                        lTotalEspacio += strlen(rowRegistro[iContador]) + 1;
                    else
                        lTotalEspacio += strlen("null") + 1;
                }
            }
            strRegistros = (char*)malloc(lTotalEspacio * sizeof(char));
            strcpy(strRegistros, "");

            mysql_data_seek(resRegistros, 0);
            for (iFilas = 0; iFilas < lRegistrosAfectados; iFilas++)
            {
                rowRegistro = mysql_fetch_row(resRegistros);
                for(iContador = 0; iContador < iColumnas; iContador++)
                {
                    if (rowRegistro[iContador] != NULL)
                        strcat(strRegistros, rowRegistro[iContador]);
                    else
                        strcat(strRegistros, "null");

                    if (iContador < iColumnas - 1)
                        strcat(strRegistros, "\t");
                }
            }
        }
    }
}

```

```

        else
            strcat(strRegistros, "\n");
    }
}
mysql_free_result(resRegistros);
strcpy(strError, mysql_error(myServerConnect));

return strRegistros;
}
else
{
    if (!mysql_field_count(myServerConnect))
    {
        lRegistrosAfectados =
(long)mysql_affected_rows(myServerConnect);
        strNumero = ltoa(lRegistrosAfectados);
        strcpy(strError, strNumero);
        strcat(strError, "Registros Seleccionados Correctamente");

        return strError;
    }
    else
    {
        strcpy(strError, mysql_error(myServerConnect));

        return NULL;
    }
}
}
else
{
    strcpy(strError, mysql_error(myServerConnect));

    return NULL;
}
}
}

long RegistrosAfectados()
{
    return lRegistrosAfectados;
}

char* NuevoID(char* Tabla, char* Campo, char* Filtro)
{
    int iError, iColumnas, iContador, iFilas;
    char* strNuevoID;
    long lActualID;
    MYSQL_RES *resRegistros;
    MYSQL_ROW rowRegistro;

    if (!Tabla && !Campo && !Filtro)
        return NULL;
    else
    {
        strcpy(strConsulta, "Select ");
    }
}

```

```

strcat(strConsulta, Campo);
strcat(strConsulta, " from ");
strcat(strConsulta, Tabla);

if (strlen(Filtro) > 0)
{
    strcat(strConsulta, " where ");
    strcat(strConsulta, Filtro);
}
strcat(strConsulta, " order by ");
strcat(strConsulta, Campo);
strcat(strConsulta, " desc limit 1");
iError = mysql_real_query(myServerConnect, strConsulta,
strlen(strConsulta));

if (!iError)
{
    resRegistros = mysql_store_result(myServerConnect);

    if (resRegistros)
    {
        iColumnas = mysql_num_fields(resRegistros);
        iFilas = mysql_num_rows(resRegistros);

        if ( iColumnas >= 1)
        {
            if (iFilas == 1)
            {
                rowRegistro = mysql_fetch_row(resRegistros);
                lActualID = atol(rowRegistro[0]);
            }
            else
                lActualID = 0;
        }
        else
            lActualID = -1;

        mysql_free_result(resRegistros);
        strcpy(strError, mysql_error(myServerConnect));

        lActualID++;
        if (lActualID > 0)
            return ltoa(lActualID);
        else
            return NULL;
    }
    else
    {
        strcpy(strError, mysql_error(myServerConnect));

        return NULL;
    }
}
else
{
    strcpy(strError, mysql_error(myServerConnect));
}

```



```

        return NULL;
    }
}

char* EjecutarComando(char* Comando)
{
    int iError, iColumnas, iFilas, iContador, lTotalEspacio;
    char* strNumero;
    char* strRespuesta = NULL;
    MYSQL_RES *resRegistros;
    MYSQL_ROW rowRegistro;

    if (!Comando)
        return NULL;
    else
    {
        iError = mysql_real_query(myServerConnect, Comando, strlen(Comando));

        if (iError == 0)
        {
            resRegistros = mysql_store_result(myServerConnect);

            if (resRegistros)
            {
                iColumnas = mysql_num_fields(resRegistros);
                lRegistrosAfectados = mysql_num_rows(resRegistros);

                if (lRegistrosAfectados > 0)
                {
                    lTotalEspacio = 1;

                    for (iFilas = 0; iFilas < lRegistrosAfectados; iFilas++)
                    {
                        rowRegistro = mysql_fetch_row(resRegistros);
                        for (iContador = 0; iContador < iColumnas; iContador++)
                        {
                            if (rowRegistro[iContador] != NULL)
                                lTotalEspacio += strlen(rowRegistro[iContador]) + 1;
                            else
                                lTotalEspacio += strlen("null") + 1;
                        }
                    }
                    strRespuesta = (char*)malloc(lTotalEspacio * sizeof(char));
                    strcpy(strRespuesta, "");

                    mysql_data_seek(resRegistros, 0);
                    for (iFilas = 0; iFilas < lRegistrosAfectados; iFilas++)
                    {
                        rowRegistro = mysql_fetch_row(resRegistros);
                        for (iContador = 0; iContador < iColumnas; iContador++)
                        {
                            if (rowRegistro[iContador] != NULL)
                                strcat(strRespuesta, rowRegistro[iContador]);
                            else
                                strcat(strRespuesta, "null");
                        }
                    }
                }
            }
        }
    }
}

```

```

        if (iContador < iColumnas - 1)
            strcat(strRespuesta, "\t");
        else
            strcat(strRespuesta, "\n");
    }
}
}
mysql_free_result(resRegistros);
strcpy(strError, mysql_error(myServerConnect));

    return strRespuesta;
}
else
{
    if (!mysql_field_count(myServerConnect))
    {
        lRegistrosAfectados =
(long)mysql_affected_rows(myServerConnect);
        strNumero = ltoa(lRegistrosAfectados);

        return strNumero;
    }
    else
        return NULL;
}
}
else
{
    strcpy(strError, mysql_error(myServerConnect));

    return strError;
}
}
}

char* VersionCliente()
{
    return mysql_get_client_info();
}

char* VersionServidor()
{
    if (myServerConnect)
        return mysql_get_server_info(myServerConnect);
    else
        return NULL;
}
}

```

Archivo: conversiones.h

```

#ifndef _CONVERSIONES_H
#define _CONVERSIONES_H

char* itoa(int);
char* uitoa(unsigned int);
char* ltoa(long);

```

```
char* ultoa(unsigned long);  
  
#endif
```

Archivo: conversiones.c

```
static char cBuffer[12];  
  
char* itoa(int value)  
{  
    char *strNumero;  
    int iFlag = 0;  
  
    if( value < 0 )  
    {  
        iFlag++;  
        value= -value;  
    }  
  
    strNumero = ultoa(value);  
    if(iFlag)  
        *--strNumero = '-';  
  
    return strNumero;  
}  
  
char* ultoa(unsigned int value)  
{  
    char *strNumero;  
  
    strNumero = cBuffer + sizeof(cBuffer);  
    *--strNumero = '\\0';  
  
    do  
    {  
        *--strNumero = '0' + value % 10;  
        value /= 10;  
    }  
    while(value);  
  
    return strNumero;  
}  
  
char * ltoa(long value)  
{  
    char *strNumero;  
    int iFlag = 0;  
  
    if( value < 0 )  
    {  
        iFlag++;  
        value= -value;  
    }  
  
    strNumero = ultoa(value);  
    if(iFlag)
```

```

        *--strNumero = '-';

    return strNumero;
}

char * ultoa(unsigned long value)
{
    char *strNumero;

    strNumero = pBuffer + sizeof(cBuffer);
    *--strNumero = '\\0';

    do
    {
        *--strNumero = '0' + value % 10;
        value /= 10;
    }
    while(value);

    return strNumero;
}

```

4.2.5.4. Procedimientos en PHP, JavaScript y HTML – Aplicación cliente en web

En la aplicación de cliente se utiliza una combinación de tecnologías en cuanto a lenguaje que se ejecuta en el cliente y que se ejecuta en el servidor. En la capa media del servidor de aplicaciones se tiene PHP 4.4 con los módulos de acceso para MySQL. En el cliente se utiliza JavaScript para ciertas validaciones y el despliegue de calendario para el ingreso sin errores de fechas.

La aplicación en el servidor de aplicaciones consta de las páginas:

- Página principal
- Página de introducción al sistema
- Página de tecnología utilizada
- Página de monitoreo en tiempo real con verificación de usuario
- Página de estadísticas de monitoreo con verificación de usuario

La página principal tiene una pequeña presentación con los enlaces hacia las otras páginas restantes, está realizada con HTML simple.

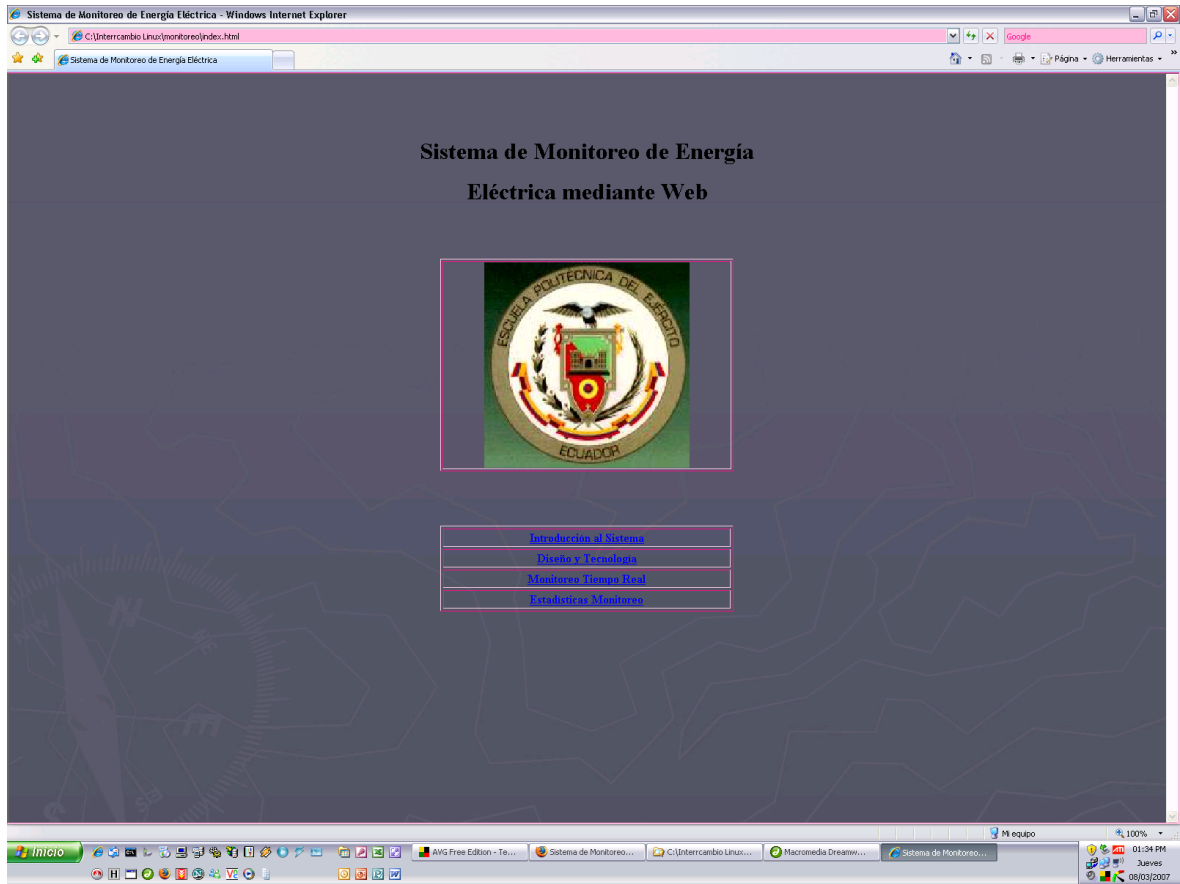


Figura 4.57 – Página Principal con enlaces

La página de introducción al sistema contiene una explicación general de cómo funcionan los sistemas de adquisición de datos y su interacción con el software.

Sistema de Monitoreo de Energía Eléctrica mediante Web

Este proyecto está concebido para industrias pequeñas y medianas que deseen implementar un sistema de monitoreo de energía eléctrica para tener un control del uso de sus recursos, para las grandes industrias se enfoca en el ahorro de recursos económicos que se utilizarán cuando se implementen este tipo de sistemas que en la actualidad usan dispositivos de propósito específico y sin posibilidades de crecimiento o cambio de requerimientos.

El diseño del proyecto tiene dos partes bien definidas en hardware y software. Se utiliza como base la arquitectura de un Sistema de Adquisición de Datos Estándar y tiene la combinación de otras tecnologías como los Sistemas de Administración de Bases de Datos y los clientes basados en Web.

Un Sistema de Adquisición de Datos utiliza una combinación sincronizada de hardware y software en una secuencia determinada mostrada a continuación:

Sistema de Adquisición de Datos basado en PC

- Hardware
 - Proceso o fenómeno físico
 - Transductores (transformación de señales)
 - Acondicionadores de señales
 - Tarjetas de Adquisición de Datos (Internas o externas)
 - Computador Personal (IBM Compatible, Macintosh)
- Software
 - Software de instrumentación virtual
 - Servidor de base de datos
 - Servidor de transporte de datos para tiempo real
 - Servidor de aplicaciones (Web)
 - Cliente (específico o basado en Web)

Siempre debe existir un proceso físico real que es el punto de partida, el cual monitoreamos para tener datos estadísticos y en base a estos poder controlar en forma automática o tomar decisiones con respecto a su funcionamiento. Para lograr esta funcionalidad se utilizan los transductores que son dispositivos electrónicos que transforman de un tipo de energía a otra; para el caso de nuestro proyecto se hace una transformación de un tipo de energía eléctrica a otro tipo de energía eléctrica, es decir, voltaje a corriente o viceversa.

Luego se utiliza los acondicionadores de señales, que sirven para colocar las señales en los niveles adecuados para que las tarjetas de adquisición puedan tomar los valores de forma correcta y sin errores. En el proyecto se realiza un conjunto de actividades que son tomadas como acondicionamiento de señales y son las siguientes: amplificación, aislamiento y filtrado. Una vez que se tienen las señales en los niveles adecuados, estas ingresan a las tarjetas de adquisición de datos y aquí se realiza la conversión analógica a digital de los datos, que puede ser mediante ADC's dedicados o multiplexados. Las tarjetas de adquisición de datos pueden ser internas colocadas dentro del PC en interfaces ISA, EISA o PCI y; externas como

Figura 4.58 – Página Descriptiva de los Sistemas DAQ y el Software

La página de diseño y tecnología contiene una explicación más técnica sobre como está desarrollado el sistema de monitoreo en lo que se refiere al software de la aplicación servidora como de la aplicación de cliente. Aquí se describe los lenguajes, motores de bases de datos, servidores web y otras piezas de software utilizadas para la construcción del sistema.

La página de monitoreo en tiempo real está realizada con PHP y JavaScript. Está primera verifica si se ha establecido una sesión de usuario del sistema para poder visualizar los circuitos a los que se tiene acceso, de otro modo despliega la pantalla de acceso al sistema y luego se redirecciona a la página para visualizar el monitoreo en tiempo real.

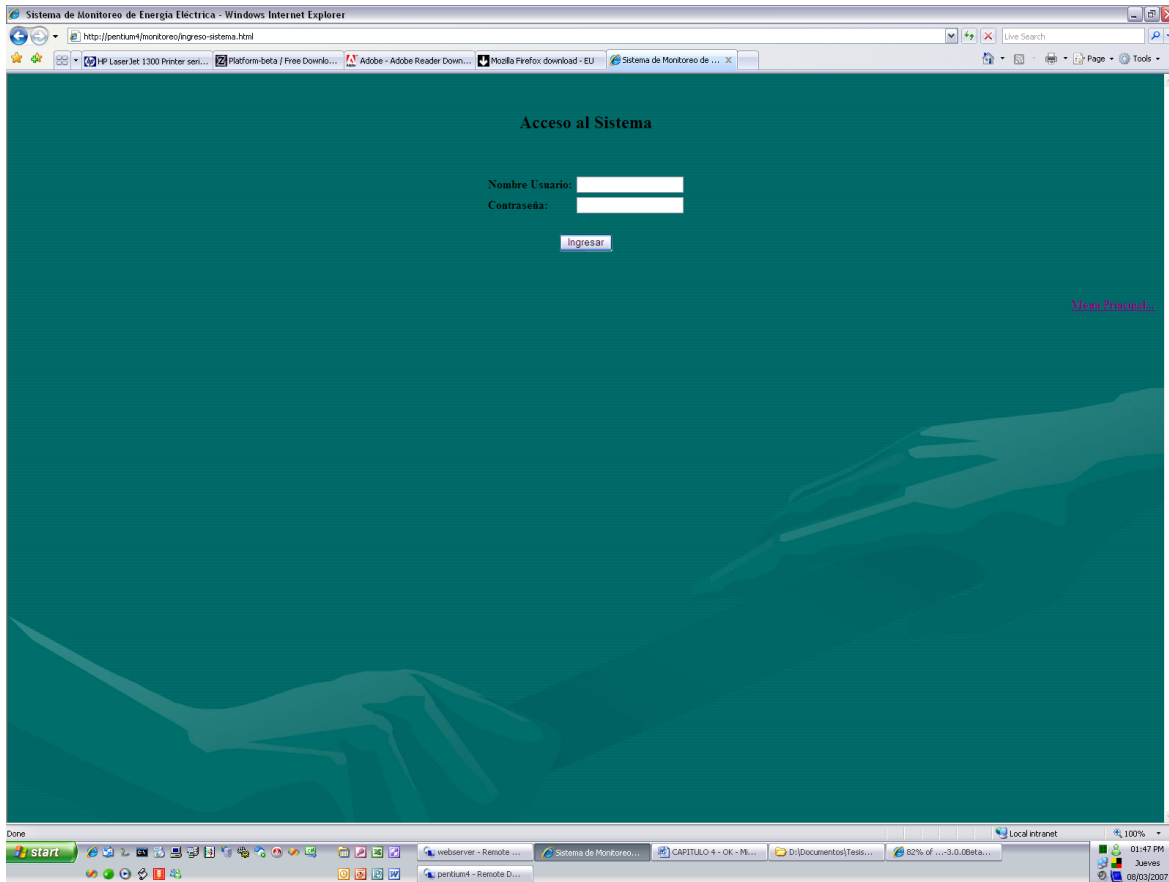


Figura 4.59 – Página para Verificación de Usuario del Sistema

A continuación se mostrará el código PHP que contiene la página de verificación de usuarios y su redirección.

```
<?php require_once('Connections/Monitoreo.php'); ?>
<?php
if (!isset($_SESSION))
    session_start();

if (isset($_POST['NombreUsuario']))
{
    $strNombreUsuario = $_POST['NombreUsuario'];
    $strContraseña = $_POST['Contraseña'];
    mysql_select_db($database_Monitoreo, $Monitoreo);

    $strConsulta = sprintf("SELECT usrNombreUsuario, usrContraseña, usrNivelSeguridad
FROM usuario WHERE usrNombreUsuario='%s' AND usrContraseña='%s'",
    get_magic_quotes_gpc() ? $strNombreUsuario : addslashes($strNombreUsuario),
    get_magic_quotes_gpc() ? $strContraseña : addslashes($strContraseña));

    $resRegistros = mysql_query($strConsulta, $Monitoreo) or die(mysql_error());
    $iNumeroUsuarios = mysql_num_rows($resRegistros);
```

```

if ($iNumeroUsuarios)
{
    $arrRegistros = mysql_fetch_assoc($resRegistros);
    $_SESSION['NivelSeguridad'] = $arrRegistros['usrNivelSeguridad'];
    $_SESSION['IngresoSistema'] = 'OK';

    header("Location: " . $_SESSION['Origen']);
}
else
{
    unset($_SESSION['IngresoSistema']);
    unset($_SESSION['NivelSeguridad']);
    //header("Location: " . $MM_redirectLoginFailed );
}
mysql_free_result($resRegistros);
}
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Sistema de Monitoreo de Energía Eléctrica - Verificación Usuarios</title>
</head>

<body background="./imagenes/acceso.JPG">
</body>
</html>

```

Código de la página PHP de verificación de usuario

Una vez que se ha realizado la verificación del usuario del sistema se redirecciona hacia la página que haya sido solicitada, monitoreo en tiempo real o estadísticas de monitoreo. A continuación se visualizará la página de monitoreo en tiempo real y su código en PHP con JavaScript. Esta página verifica que se haya seleccionado un circuito antes poder presentarlos datos de monitoreo en tiempo real. Como se necesita que los datos sigan actualizándose constantemente para reflejar el cambio en los valores que están siendo monitoreados se refresca la página automáticamente cada 3 segundos y de manera tal que no causa carga excesiva sobre la red.

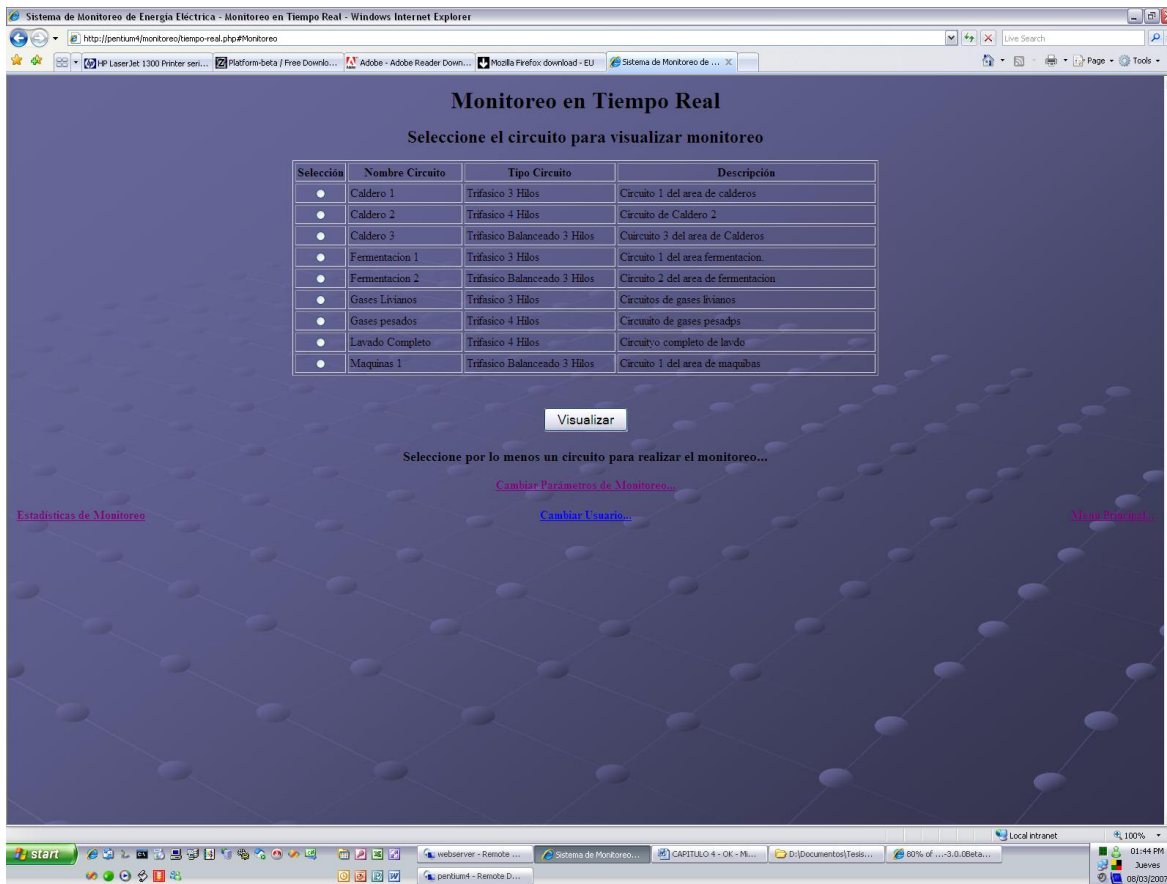


Figura 4.60 – Página de Mediciones en Tiempo Real

```

<?php require_once('conexion_mysql.php'); ?>
<?php
if (!isset($_SESSION))
{
    session_start();
    $_SESSION['Origen'] = $_SERVER['PHP_SELF'];
    if (!isset($_SESSION['IngresoSistema']))
        header("Location: ingreso-sistema.html");
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Sistema de Monitoreo de Energía Eléctrica - Monitoreo en Tiempo Real</title>
</head>

<script language="javascript">
    function SetearIntervalo()
    {

```



```

        <td>' . $rowCircuito['tctNombreTipoCircuito']
    . '</td>
        <td>' . $rowCircuito['crcDescripcion'] . '</td>
    </tr>;
    }
    ?>
</table>
<input type="hidden" name="Accion" value="Visualizar" />
<br /><br />
<table width="100%" border="0" cellspacing="3" cellpadding="2">
    <tr>
        <td align="center">
            <input type="submit" value="Visualizar"
style="font-size:larger; font:bold" />
        </td>
    </tr>
</table>
</form>

<?php
    if (isset($_POST) && $_POST['Accion'] == 'Visualizar')
    {
        if (count($_POST) == 1)
            echo '<h3 align="center">Seleccione por lo menos un circuito para
realizar el monitoreo...</h3>';
        else
        {
            echo '<br /><h2 align="center">Monitoreo en Tiempo Real
por Circuito</h2>';

            $resConfiguracion = fopen('Config.xml', 'r');
            if ($resConfiguracion)
            {
                while (!feof($resConfiguracion))
                {
                    $strConfiguracion .= fgets($resConfiguracion);
                }
                fclose($resConfiguracion);

                $iPosicion = strpos($strConfiguracion, '<chart');
                if (!$iPosicion === false)
                    $strNuevaConfiguracion = str_replace('<chart',
'<chart caption="Monitoreo Tiempo Real - Circuito ' . $strNombreCircuito . '",
$strConfiguracion);

                $iPosicion = strpos($strNuevaConfiguracion, 'datos-
tiempo-real.php');

                if (!$iPosicion === false)

```

```

                                $strNuevaConfiguracion = str_replace('datos-
tiempo-real.php', 'datos-tiempo-real.php?IDCircuito=' . $strIDCircuito,
$strNuevaConfiguracion);

                                $resConfiguracionNueva = fopen('./Data.xml', 'w');
                                if ($resConfiguracionNueva)
                                {
                                    $iTotal = fputs($resConfiguracionNueva,
$strNuevaConfiguracion);
                                }
                                else
                                    echo '<h3 align="center">No se puede escribir
el archivo de configuración...</h3>';
                                fclose($resConfiguracionNueva);
                                }
                                else
                                    echo '<h1>No se puede leer el
archivo</h1><br><br>';

                                    echo '<a name="Monitoreo"></a>';
                                    echo '<table width="100%" ><tr><td align="center">';
                                    echo '<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-
444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#versio
n=7,0,19,0" width="800" height="600">
                                <param name="movie" value="./Charts/FI2_RT_Line.swf" />
                                <param name="FlashVars"
value="&chartWidth=800&chartHeight=600&dataURL=Data.xml">
                                <param name="quality" value="high" />
                                <embed src="./Charts/FI2_RT_Line.swf"
FlashVars="&chartWidth=800&chartHeight=600&dataURL=Data.xml" quality="high"
pluginpage="http://www.macromedia.com/go/getflashplayer" type="application/x-
shockwave-flash" width="800" height="600"></embed>
                                </object>;
                                    echo '</td></tr><tr><td>&nbsp;</td></tr>
<tr>
                                <td align="center">
                                    <input type="text" name="Intervalo" size="4"
/>
                                    <input type="button" value="Setear"
onclick="SetearIntervalo();" />
                                </td>
                                </tr>
                                </table>;
                                }
                                }
?>

```

```

        <h4 align="center">
            <a href="#Inicio">Cambiar Parámetros de Monitoreo...</a>
        </h4>

<table width="100%" >
    <tr>
        <td width="33%"><h4><a href="/estadisticas-monitoreo.php">Estadísticas de
Monitoreo</a></h4></td>
            <td width="34%"><h4 align="center"><a href="/cambiar-
usuario.php">Cambiar Usuario...</a></h4></td>
        <td width="33%"><h4 align="right"><a href=".">Menú Principal...</a></h4></td>
    </tr>
</table>
</body>
</html>

```

Código de la página PHP de monitoreo en tiempo real

A continuación se visualizará la página de estadísticas de monitoreo y su código en PHP con JavaScript. Esta página verifica que se haya seleccionado uno o varios circuitos y de que se hayan seleccionado fechas de inicio y fin en un rango válido antes poder presentar los datos de estadísticas de monitoreo.



Figura 4.61 – Página de Estadísticas de Monitoreo

```

<?php require_once('conexion_mysql.php'); ?>
<?php
if (!isset($_SESSION))
{
    session_start();
    $_SESSION['Origen'] = $_SERVER['PHP_SELF'];
    if (!isset($_SESSION['IngresoSistema']))
        header("Location: ingreso-sistema.html");
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Sistema de Monitoreo de Energía Eléctrica - Estadísticas Monitoreo</title>
<script language="javascript"
src="./Scripts/codethatcalendarstd.js"></script>
<script language="javascript" src="./Scripts/calendario.js"></script>
<script>
var dpFechaInicial = new CodeThatCalendar(caldef1);

```

```

        var dpFechaFinal = new CodeThatCalendar(caldef1);

        function AsignarFechas()
        {
            document.Parametros.FechaInicial.value =
document.Parametros.txtFechaInicial.value;
            document.Parametros.FechaFinal.value =
document.Parametros.txtFechaFinal.value;
        }
    </script>
</head>

<body background="/.imagenes/estadisticas.JPG">
    <a name="Inicio"></a>
    <h1 align="center">
        Seleccione el circuito para visualizar estadísticas
    </h1>
    <?php
        $arrCircuitos = CargarDatos('monitoreo', 'circuito, tipocircuito',
'crcIDCircuito, tctNombreTipoCircuito, crcNombreCircuito, crcDescripcion,
crcNivelSeguridad',
        'circuito.tctIDTipoCircuito = tipocircuito.tctIDTipoCircuito and
crcNivelSeguridad <= ' . $_SESSION['NivelSeguridad'], ', ' . 'crcNombreCircuito');
    ?>
    <form action="/.estadisticas-monitoreo.php" method="post"
name="Parametros">
        <table width="800" border="1" cellspacing="3" cellpadding="2"
align="center">
            <tr>
                <th scope="col" width="20">Selección</th>
                <th scope="col" width="150">Nombre Circuito</th>
                <th scope="col" width="200">Tipo Circuito</th>
                <th scope="col">Descripción</th>
            </tr>
            <?php
                $arrCircuitosMediciones = array();
                foreach ($_POST as $rowPOST => $strValor)
                {
                    $iPosicion = strpos($rowPOST, 'IDCircuito_');
                    if (!$iPosicion === false)
                        $arrCircuitosMediciones[] =
substr($rowPOST, 11);
                }

                foreach ($arrCircuitos as $rowCircuito)
                {
                    echo '<tr>

```

```

                                <td align="center"><input type="checkbox"
name="IDCircuito_' . $rowCircuito['crcIDCircuito'] . '"';
                                if      (in_array($rowCircuito['crcIDCircuito'],
                                $arrCircuitosMediciones))
                                    echo 'checked="checked"';
                                echo ' /> </td>
                                <td>' . $rowCircuito['crcNombreCircuito'] .
'</td>
                                <td>' . $rowCircuito['tctNombreTipoCircuito']
. '</td>
                                <td>' . $rowCircuito['crcDescripcion'] . '</td>
                                </tr>';
                                }
                                ?>
                                </table>
                                <input type="hidden" name="Accion" value="Consultar" />
                                <br /><br />
                                <table width="100%" border="0" cellspacing="3" cellpadding="2">
                                <tr>
                                <td width="50%" align="right">
                                <h4>
                                Fecha Inicial :
                                <input
                                name="txtFechaInicial"
                                type="text" disabled="true" size="12" value="<?php
                                echo $_POST['FechaInicial'];
                                ?>"/>
                                <input name="FechaInicial" type="hidden"
                                id="FechaInicial" />
                                <input type="button" value=" ... "
                                onclick="dpFechaInicial.popup('txtFechaInicial');" />
                                </h4>
                                </td>
                                <td >
                                <h4>Fecha Final :
                                <input
                                name="txtFechaFinal"
                                type="text" disabled="true" size="12" value="<?php
                                echo $_POST['FechaFinal'];
                                ?>"/>
                                <input name="FechaFinal" type="hidden"
                                id="FechaFinal" />
                                <input type="button" value=" ... "
                                onclick="dpFechaFinal.popup('txtFechaFinal');" />
                                </h4>
                                </td>
                                </tr>
                                <tr>
                                <td align="center" colspan="2">

```



```

                                <input type="submit" value="Consultar"
style="font-size:larger; font:bold" onclick="AsignarFechas();" />
                                </td>
                            </tr>
                        </table>
                    </form>

    <?php
        if (isset($_POST) && $_POST['Accion'] == 'Consultar')
        {
            if (count($_POST) == 3)
                echo '<h3 align="center">Seleccione por lo menos un circuito para
realizar la consulta...</h3>';
            else
            {
                $strFechaInicial = $_POST['FechaInicial'];
                $strFechaFinal = $_POST['FechaFinal'];
                if (strlen($strFechaInicial) == 0 || strlen($strFechaFinal) == 0)
                    echo '<br /><h3 align="center">Debe ingresar las fechas para
realizar la consulta...</h3>';
                else
                {
                    if ($strFechaInicial > $strFechaFinal)
                        echo '<br /><h3 align="center">La fecha inicial no
puede ser mayor que la fecha final...</h3>';
                    else
                    {
                        echo '<h2 align="center">Datos Estadísticos
por Circuito</h2>';

                        foreach ($arrCircuitosMediciones as
$IIDCircuito)
                        {
                            foreach ($arrCircuitos as $rowCircuito)
                            {
                                if ($rowCircuito['crcIDCircuito']
== $IIDCircuito)
                                    break;
                            }
                            echo '<br />
<table width="100%" border="1"
cellspacing="3" cellpadding="2">
                                <tr>
                                    <th colspan="3"><h3
style="color:#FF0022">Nombre Circuito</h3></th>

```

```

        <th colspan="15"><h3>'
        . $rowCircuito['crcNombreCircuito'] . ' - ' . $rowCircuito['tctNombreTipoCircuito']
        .</h3></th>

```

```
</tr>
```

```
<tr>
```

```
scope="col">Fecha</th>
```

```
<th
```

```
scope="col">Hora</th>
```

```
<th
```

```
scope="col">Frec.</th>
```

```
<th
```

```
Línea 1</th>
```

```
<th scope="col">Voltaje
```

```
Línea 2</th>
```

```
<th scope="col">Voltaje
```

```
Línea 3</th>
```

```
<th scope="col">Voltaje
```

```
1</th>
```

```
<th scope="col">Voltaje
```

```
2</th>
```

```
<th scope="col">Voltaje
```

```
3</th>
```

```
<th scope="col">Voltaje
```

```
scope="col">Corriente Línea 1</th>
```

```
<th
```

```
scope="col">Corriente Línea 2</th>
```

```
<th
```

```
scope="col">Corriente Línea 3</th>
```

```
<th
```

```
Potencia</th>
```

```
<th scope="col">Factor
```

```
scope="col">Potencia Total</th>
```

```
<th
```

```
scope="col">Potencia Reactiva</th>
```

```
<th
```

```
scope="col">Potencia Aparente</th>
```

```
<th
```

```
scope="col">Energía</th>
```

```
<th
```

```
Reactiva</th>
```

```
<th scope="col">Energía
```

```
</tr>';
```

```

        $arrMediciones = CargarDatos('monitoreo',
        'circuitomedicion', "", 'crcIDCircuito = ' . $iIDCircuito . " and cmdFecha >= " .
        $strFechaInicial . " and cmdFecha <= " . $strFechaFinal . ""', "", 'crcIDCircuito, cmdFecha,
        cmdHora');

```

```

$rowMedicion)
foreach ($arrMediciones as
{
    echo '<tr>'
    <td align="center"> .
    $rowMedicion['cmdFecha'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdHora'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdFrecuencia'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdVoltaje1'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdVoltaje2'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdVoltaje3'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdVoltaje10'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdVoltaje20'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdVoltaje30'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdCorriente1'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdCorriente2'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdCorriente3'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdFactorPotencia'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdPotencia'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdPotenciaReactiva'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdPotenciaAparente'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdEnergia'] . '</td>'
    <td align="center"> .
    $rowMedicion['cmdEnergiaReactiva'] . '</td>'
    </tr>;
}
echo '</table>';
}
}
}

```

```

        ?>
        <h4 align="center">
            <a href="#Inicio">Cambiar Parámetros Consulta...</a>
        </h4>
    <?php
        }
    }
    ?>
<table width="100%" >
    <tr>
        <td width="33%"><h4><a href="/tiempo-real.php">Monitoreo en Tiempo
Real</a></h4></td>
        <td width="34%"><h4 align="center"><a href="/cambiar-
usuario.php">Cambiar Usuario...</a></h4></td>
        <td width="33%"><h4 align="right"><a href=".">Menú Principal...</a></h4></td>
    </tr>
</table>
</body>
</html>

```

Código de la página PHP de estadísticas de monitoreo

4.3. Implementación de la Aplicación

4.3.2. Instalación de Computadores (Sistema Operativo, Comunicaciones)

Para el equipo que vaya a realizar las funciones de servidor de adquisición de datos se debe tener instalado Mandrake Linux 10.0 para estación de trabajo. A continuación se debe instalar MySQL 4.21 y crear la instancia de base de datos de forma predeterminada.

Para el equipo de cliente se necesita que sea cualquier versión de Microsoft Windows, MacOS o cualquier versión de Linux que posea un browser de internet que soporte CSS's y JavaScript. Para las comunicaciones entre los equipos se necesita que posean una conexión de red de tipo cableado, la cual debe funcionar con el protocolo TCP/IP.

4.3.3. Configuración del Servidor de Base de Datos

Una vez levantado el motor de la base de datos se creará una nueva base de datos con el nombre de 'monitoreo' y se ejecutará el script de creación de tablas y carga de los datos básicos, dejando la base de datos lista para poder operar.

4.3.4. Configuración del Servidor Web

Primero se debe instalar un servidor web que soporte páginas dinámicas de servidor (.php), Apache 2.05 que viene integrado con Mandrake Linux 10.0; luego crear un sitio virtual para poder acceder desde cualquier computador y fijar las seguridades adecuadas de directorios para evitar problemas con la aplicación y la información. En ese sitio virtual se debe colocar las páginas de la aplicación (.html, .js y .php). A continuación se procede a cargar el módulo de MySQL sobre PHP para que pueda conectarse a la base de datos y traer el contenido dinámico.

4.3.5. Instalación y Configuración de la Aplicación de Servidor

Se debe instalar a continuación la aplicación compilada generada en LabVIEW 6i. A continuación se procederá con la configuración al iniciar la aplicación por primera vez.

Inicialmente no se encuentra registrado ningún usuario, por lo tanto se pedirá una contraseña para el usuario Administrador, luego permitirá definir las características de la planta mediante el ingreso de áreas, circuitos por área y puntos de medición por circuito; finalmente se definirán los usuarios con su nivel de seguridad para acceso al sistema. La aplicación de servidor se encuentra lista para entrar en funcionamiento y comenzar la adquisición de datos. La aplicación siempre registrará el ingreso y salida del sistema para saber que usuario está utilizando el sistema.

4.4. Pruebas de Rendimiento sobre Entornos Lan e Internet

Primero se realizará una descripción de las condiciones y equipos que se utilizaron en las pruebas del sistema. Como equipo servidor de web y base de datos se ha utilizado un computador con las siguientes características:

- Procesador Pentium 4 de 2.4 GHz (533MHz FSB)
- 512 MB memoria RAM DDR 333 MHz
- 80 GB disco duro (5400 RPM – modo ATA 100)
- 128 MB memoria RAM DDR de video a 1600*1200
- Sistema Operativo Windows 2000 Professional
- Tarjeta de red 3Com PCI a 100 Mbps

Para la realización de las pruebas se ha utilizado la tarjeta de red 3Com PCI a velocidades de 100 Mbps y 10 Mbps para un entorno de red de área local y; una conexión serial de 56 Kbps y 128 Kbps para simular un entorno de red de área extendida remota.

4.4.1. Pruebas de Velocidad y Desempeño sobre Lan e Internet

En estas pruebas se están considerando los tiempos de conexión a la DBMS, tiempos de recuperación de los datos de configuración de puntos de medición y tiempos de almacenamiento de los datos procesados a la DBMS; todo esto sobre un entorno de tipo local para obtener velocidades de respuesta de la DBMS. A continuación se realizará una

explicación detallada de cada prueba que se ejecutará con la bases de datos para obtener los tiempos de respuesta:

- Conexión al Sistema en LabVIEW para registrarse y comenzar la adquisición (local)
- Almacenamiento de configuración de áreas, circuitos y puntos de adquisición (local)
- Recuperación de configuración de áreas, circuitos y puntos de adquisición con preparación del sistema (local)
- Recuperación de datos estadísticos en cliente web (local y remoto)

Esta prueba mide el tiempo en milisegundos de la carga inicial de la configuración de áreas, circuitos y usuarios hasta que entra en funcionamiento.

- Tiempo medio: 1050 ms

Esta prueba mide el tiempo en milisegundos para las operaciones de inserción, eliminación y actualización para cada uno de los casos de áreas, circuitos y puntos de medición.

- Tiempo medio: 300 ms

Esta prueba mide el tiempo en milisegundos para las operaciones de recuperación de registros de configuración de cada uno de los puntos de medición hasta cuando la aplicación se encuentra lista para empezar las operaciones de adquisición de datos.

- Tiempo medio: 220 ms

Esta prueba mide el tiempo en milisegundos para las operaciones de recuperación de registros de datos estadísticos de cada uno de los circuitos trifásicos especificados en la base de datos de la aplicación. La prueba se realizará con diferentes operaciones de recuperación de registros con parámetros de filtrado. Esta prueba se realizará en forma local y también en forma remota mediante redes WAN de 56 Kbps y 128 Kbps y LAN de 10 Mbps y 100 Mbps respectivamente.

- Tiempo medio: 750 ms

4.4.2. Pruebas de Velocidad y Desempeño sobre Internet (Dial Up, Enlaces Dedicados)

En estas pruebas se están considerando los tiempos de conexión al Servidor de base de datos. La DBMS es la que almacena los datos históricos que se podrán utilizar para la toma de decisiones. Se medirán los tiempos de recuperación de los datos de configuración de puntos de medición así como los tiempos de recuperación de los datos procesados y almacenados en la DBMS sobre un entorno de tipo local y remoto, el cliente para realizar las pruebas mencionadas será un explorador web. El monitoreo de las ondas de corriente y

voltaje en tiempo real de un circuito específico y los datos en valores RMS de todos los circuitos también se visualizarán mediante un explorador de web sobre un entorno local y remoto, aquí se realizarán mediciones de los tiempos de respuesta en los diferentes tipos de conexión y velocidades de transmisión de red para la obtención de datos estadísticos de tiempo, y así poder crear parámetros de desempeño y calidad para posteriores recomendaciones de implementaciones a los futuros usuarios del sistema de monitoreo.

➤ Visualización de valores RMS de voltaje, corriente, potencias y factor de potencia de todos los circuitos (local y remoto)

Las pruebas se realizarán en forma local y remota con las diferentes velocidades mediante redes WAN de 56 Kbps y 128 Kbps y LAN de 10 Mbps y 100 Mbps, respectivamente.

- Local - Tiempo medio: 300 ms
- 56 Kbps (44 Kbps reales) - Tiempo medio: 9500 ms
- 128 Kbps - Tiempo medio: 4000 ms
- 10 Mbps - Tiempo medio: 500 ms
- 100 Mbps - Tiempo medio: 310 ms

Esta prueba trata de medir el tiempo en milisegundos de conexión para permitir la visualización de los datos de valores RMS de voltaje, corriente, potencia aparente, potencia real, potencia reactiva y factor de potencia para cada uno de los circuitos trifásicos ingresados en el sistema. Las pruebas se realizarán en forma local y remota con las diferentes velocidades mediante redes WAN de 56 Kbps y 128 Kbps y LAN de 10 Mbps y 100 Mbps, respectivamente.

- Local - Tiempo medio: 110 ms
- 56 Kbps (44 Kbps reales) - Tiempo medio: 3200 ms
- 128 Kbps - Tiempo medio: 1450 ms
- 10 Mbps - Tiempo medio: 200 ms
- 100 Mbps - Tiempo medio: 110 ms

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- La implementación sobre Linux tiene algunos problemas debido a que es una plataforma sobre la que no existe mucho soporte porque no es tan popular como Microsoft Windows, además sigue muchos rumbos diferentes en su desarrollo.

- Una ventaja que tiene Linux a su favor es que persiguen un ideal que sea el software libre y de código abierto, lo cual permite una rápida y gran expansión con el aporte de todos los programadores a nivel mundial.
- La base de Linux es el lenguaje C que hasta el momento es el más poderoso por ser de bajo y alto nivel al mismo tiempo y que brinda una gran estabilidad para procesos como el que se está llevando a cabo que es el monitoreo de ambientes industriales.
- Muchas piezas de software que no existen se las puede realizar con lenguaje C y acoplarlas para lograr los objetivos; tal es el caso de la librería compartida que se desarrolló para permitir la comunicación entre LabVIEW 6i para Linux y el motor de base de datos MySQL 4.21.
- National Instruments se ha dado cuenta de la importancia que está ganando Linux en el mercado industrial; por esta razón con su versión 8.20 de LabVIEW ha elaborado un conjunto de controladores bastante amplio para sus dispositivos y que permite el desarrollo de controladores personalizados. LabVIEW consume muchos recursos de procesador cuando realiza la adquisición de datos, debido a que es un programa que funciona secuencialmente en un bucle indefinido. También ha definido un protocolo propietario como DataSocket para comunicación entre procesos que está basado sobre otra tecnología propietaria como es ActiveX en lugar de utilizar sockets que si es un estándar entre plataformas.
- Se puede encontrar piezas de software libre muy útiles como el proyecto Comedi, que permitió una comunicación con los dispositivos de adquisición de datos que LabVIEW 6i no los posee.
- Es extraño que se haya propuesto a OPC (OLE para Control de Procesos) como un estándar industrial, pues está desarrollado sobre las bases de DCOM que es una tecnología propietaria de Microsoft. Muy recientemente OPC Foundation se ha dado cuenta de este error y están creando una especificación técnica de OPC sobre XML que si es un estándar internacional independiente de la plataforma.
- En las pruebas se observó que la aplicación de cliente tiene un rendimiento muy bueno sobre entornos de redes locales donde la velocidad mínima en la actualidad es de 10 Mbps y puede llegar hasta 1 Gbps en entornos tecnológicos de punta. Sin embargo en entornos remotos las velocidades de los enlaces pueden ser tan bajas como 9600 bps en comunicaciones celulares hasta un máximo de 128 Kbps de un enlace de alta de banda ancha.

5.2. Recomendaciones

- Se recomienda utilizar una unidad de disco duro o de almacenamiento dedicado y con gran capacidad ya que los datos provenientes de la adquisición empezarán a crecer con gran rapidez y en forma exponencial de acuerdo al número de puntos

monitoreados si no se hace un mantenimiento de la información o se desea mantener los registros históricos por largo tiempo.

- Cuando la aplicación de monitoreo empiece a crecer en cuanto a cantidad de puntos monitoreados, se recomienda utilizar 3 computadores distintos, repartidos de la forma siguiente con enlaces de red de alta velocidad.
 - a. Servidor de Base de Datos
 - b. Servidor de Aplicaciones (Web) y DataSocket
 - c. Aplicación de Servidor de Monitoreo
- Se recomienda realizar la aplicación en un lenguaje orientado a objetos y a eventos, para de esta forma aprovechar las características del mismo y por lo tanto implementar un funcionamiento en eventos para evitar saturar el procesador por los lazos indefinidos que utiliza LabVIEW.
- Hay que tomar en cuenta que a mayor cantidad de muestras para una onda, mejor es la calidad de la reconstrucción de la misma pero también toma más tiempo de proceso de la matriz de datos adquiridos. Se recomienda disponer de un computador con una cantidad adecuada de memoria RAM.
- Es conveniente aportar con ideas y con código a la comunidad de software libre ya que de esta manera se está garantizando que el trabajo no quede almacenado como sucede en la mayoría de casos, además se debe aportar como un agradecimiento hacia las cosas que se han obtenido por la misma vía del software libre y el código abierto. De esta manera se está aportando para obtener un software estable y de buena calidad a precios muy accesibles o en forma gratuita.
- Se recomienda buscar otro software para instrumentación virtual que pertenezca a la comunidad de software libre porque LabVIEW sigue siendo un lenguaje de tipo gráfico con limitaciones como la carencia de ejecución por eventos.

Anexo A

**DPTO DE
CIENCIAS DE LA
COMPUTACION**

ANEXO B
SANGOLQUI, 20 DE ABRIL DEL 2007

**UN SISTEMA DE
MONITOREO DE
ENERGIA
ELECTRICA EN
AMBIENTES
INDUSTRIALES
MEDIANTE WEB**

AÑO
2007

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. RONNY DENYSS ESTÉVEZ MONTALVO CANDIDATO A INGENIERO como requerimiento parcial a la obtención del título de INGENIEROS EN SISTEMAS

24 de Mayo de 2007

ING. RAUL CORDOVA

DEDICATORIA

Este trabajo se lo dedico a mis padres, quienes a pesar de todos los contratiempos surgidos en el desarrollo del mismo han estado presentes para brindar su apoyo; pero principalmente a mi esposa quien ha estado presente en los momentos más duros, difíciles y extraviados de

mi vida siempre tendiéndome una mano, dándome su apoyo y sus palabras de aliento para llegar a la culminación de mi carrera profesional y permitiéndome escalar un peldaño más en la formación diaria de la compleja existencia humana.

Denyss Estévez Montalvo

HOJA DE LEGALIZACION DE FIRMAS

ELABORADO POR

Sr. Ronny Denyss Estévez Montalvo

DECANO DE LA FACULTAD DE INGENIERIA

Ing. Ramiro Degado

Lugar y fecha : Sangolquí, 24 de Mayo de 2007

REFERENCIAS BIBLIOGRÁFICAS

- JOHNSON; Johnny, **Análisis de Circuitos**, 5ta. Edición, Ed. Prentice Hall, México – 1996, 835 pág.
- NATIONAL INSTRUMENTS CORPORATION, **LabVIEW 6i – User Manual**, 3ra. Edición, Ed. National Instruments, U.S.A. – 2000, 272 pág.
- www.mandriva.com/es/linux, Mandriva Linux
- dev.mysql.com/downloads/mysql/4.1.html, MySQL Server 4.1
- www.php.net, PHP: Hypertext Preprocessor
- www.comedi.org, Comedi – Control and Measurement Interface
- forums.openg.org, OpenG Discussion Forums
- www.ni.com/labview, National Instruments – LabVIEW
- zone.ni.com/devzone/cda/main, Nacional Instruments - NI Developer Zone
- www.zend.com/products/zend_studio, Zend- Zend Studio
- www.zend.com/products/zend_platform, Zend – Zend Platform

➤ www.desarrolloweb.com/php, Desarrollo Web – Manual de PHP