# INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA

UNIDAD PROFESIONAL CULHUACAN

SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

"**MONITOREO DE LA DISPONIBILIDAD DEL SISTEMA DE CONTROL Y ADQUISICIÓN DE DATOS SCADA**"

Tesina que para obtener el grado de Especialista en Seguridad Informática y Tecnologías de la Información presenta:

**ÁNGEL TEJADA COTERO**

ASESORES:
M. en C. ELEAZAR AGUIRRE ANAYA
Dr. GABRIEL SANCHEZ PEREZ

FEBRERO DE 2009

SIP-14

# INSTITUTO POLITÉCNICO NACIONAL

## SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

*ACTA DE REVISIÓN DE TESINA*

En la Ciudad de México, D. F. siendo las 18:00 horas del día 27 del mes de abril del 2009 se reunieron los miembros de la Comisión Revisora de Tesina designada por el Colegio de Profesores de Estudios de Posgrado e Investigación de SEPI-ESIME-CULHUACAN para examinar la tesina titulada:

**"Monitoreo de la Disponibilidad del Sistema de Control y Adquisición de Datos *SCADA*"**

Presentada por el alumno:

| Tejada | Cotero | Angel |
|---|---|---|
| Apellido paterno | Apellido materno | Nombre(s) |

Con registro: B 0 7 1 9 1 2

aspirante de:

**ESPECIALIDAD EN SEGURIDAD INFORMÁTICA Y TECNOLOGÍAS DE LA INFORMACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron ***SU APROBACIÓN DE LA TESINA***, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Director de tesina

M. en C. Eleazar Aguirre Anaya

Co-Director de Tesina

Dr. Gabriel Sánchez Pérez

M. en C. Lorena Azuara Pérez

INSTITUTO POLITÉCNICO NACIONAL
ESTADOS UNIDOS MEXICANOS
S.E.P.

EL PRESIDENTE DEL COLEGIO

SECCION DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
ESIME CULHUACAN

Dr. Héctor Manuel Pérez Meana

**INSTITUTO POLITÉCNICO NACIONAL**

**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

## CARTA CESIÓN DE DERECHOS

En la Ciudad de México el día 08 del mes Mayo del año 2009, el (la) que suscribe Ángel Tejada Cotero alumno (a) del Programa de Especialización en Seguridad Informática y Tecnologías de la Información con número de registro B071912, adscrito a SEPI ESIME Culhuacan, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de, M en C. Eleazar Aguirre Anaya, Dr. Gabriel Sánchez Pérez y cede los derechos del trabajo intitulado "MONITOREO DE LA DISPONIBILIDAD DEL SISTEMA DE CONTROL Y ADQUISICIÓN DE DATOS, SCADA", al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección angel.tejada@cfe.gob.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Ing. Ángel Tejada Cotero

# RESUMEN

A continuación se dará una breve descripción del sistema de datos SCADA y de su importancia de permanecer disponible el mayor tiempo posible:

El sistema de datos, [Sistem Control And Data Adquisition], Sistema de Control y Adquisición de Datos, por sus siglas en inglés “SCADA”, provee una serie de parámetros y variables pertenecientes a las subestaciones de potencia el cual pertenecen al sistema eléctrico de potencia occidental, en el caso del área de control occidente, que es donde se desarrollara el proyecto antes mencionado, dichas variables son: voltajes, amperes y watts, así como el estado que guardan las subestaciones, sus interruptores, sus cuchillas y alarmas que se generan dentro de ellas, estos parámetros y valores, son enviados en tiempo real al área de control, siendo estas muy importantes para la toma de decisión del operador en turno, y generar más electricidad o apagar generación, abrir o cerrar interruptores en caso de un disturbio del sistema eléctrico, y poder suministrar energía eléctrica con calidad y rapidez al momento de una falla.

Es importante mencionar, que las variables antes mencionadas son recibidas en el área de control, por el servidor de datos SCADA, y que éste algunas ocasiones llega a fallar y quedar fuera de línea, presentando subestaciones fuera de barrido.- esto es que el operador del área no recibe datos de dicha subestación, quedando imposibilitado de realizar maniobras a telecontrol, por

ello la importancia de que dicho servidor de datos, no quede fuera de línea y de ser así, que sea el menor tiempo posible.

En este proyecto, se tiene contemplado la investigación y documentación de un sistema de aviso falla fuera de línea servidor de datos SCADA, a través de envío de mensaje vía celular o a varios celulares a personal del área de control, esto con la finalidad de reducir el tiempo que permanece el servidor de datos fuera de línea, ya que en algunas ocasiones durante el turno, queda fuera de 20 a 30 minutos sin que el operador realice el reporte, ya que pueden tener algunas otras actividades relacionadas con el sistema, y al transcurrir ese tiempo, reportan a personal de comunicaciones, también dicha falla se muestra en los monitores del sistema, **pero en ocasiones nadie esta verificando dichos monitores** y se pierde información importante, por lo tanto se considera de suma importancia dicho proyecto.

En general, el proyecto esta desarrollado en el lenguaje de programación PERL, ya que es un lenguaje robusto y seguro, comparado con JAVA y PHP, más adelante en la figura 1.1 se realiza una comparación versus estos 2 lenguajes de programación, y ventajas de utilizar PERL.

# ABSTRACT

Then there will be a brief description of the data system SCADA and its importance to stay online as long as possible:

The SCADA system data, , (Control and Data Acquisition System, system control and data acquisition in real-time), provides a set of parameters and variables belonging to the power substation, in the case of control area , which is where you develop the project mentioned above, these variables are: voltage amps and watts, as well as the status of the substations, as their switches, alarms and their blades that are generated within them, these parameters and values are sent in real time to control area, this is very important for decision-making operator in turn, as it is, generate more electricity generation or off, open or close switches in the event of a disturbance in the power system and power supply electrical power quality and speed at the time of failure.

It is important to mention that the variables above are received in the control area, the SCADA data server, and it sometimes reaches fail and be off-line, introducing substations outside sweep.- this is that the operator in the area do not have control of that substation, and is unable to carry out maneuvers to remote, why the importance of such data server, not outside line, and if so, which is the shortest possible time.

In this project, there are plans to research and documentation of a warning system fails offline data server SCADA through sending messages via cell phone or several cells staff control area,  with the purpose of reducing time server data offline, because sometimes during the day, offline 20 to 30 minutes without the operator to make the report because it may have some other activities related to the system, and to pass this time, reported to communications personnel also such failure is shown in the monitors of the system, but sometimes no one is checking these monitors and important information is lost, therefore we consider extremely important this project.

Overall, the project is developed in the PERL programming language, and that language is a robust and secure, compared with Java and PHP, later way of comparison versus these 2 programming languages, and advantages of using PERL.

# DEFINICIÓN DEL PROBLEMA

Actualmente no se tiene un sistema rápido y eficiente, solo se tiene un sistema normal de aviso en los monitores, de falla de los servidores de datos SCADA

Pero si ninguna persona esta observando dicha alarma, podrá pasar hasta 30 min. Fuera de línea, poniendo en riesgo el suministro de energía eléctrica

Ya que estos servidores están proporcionando información en forma segura para la correcta operación del sistema eléctrico occidental.

# JUSTIFICACIÓN

Si el servidor de datos permaneciera fuera de línea durante un periodo de tiempo muy prolongado, afectaría en no tener la información procedente de campo, en tiempo real, ya que las personas que están operando el sistema eléctrico occidental, se proveen de esta información, para poder controlar las variables concernientes al sistema, tales como, voltajes, amperes, watts, etc.

Esto daría un sistema eléctrico seguro y robusto, además de proveer un servicio de energía con mejor calidad, y respuesta ante cualquier eventualidad ocurrida en el mismo, da una aportación benéfica al país, coadyuvando al progreso de México, siendo una empresa de clase mundial.

La solución de dicho problema, beneficiara al personal de operación, ya que se tendrá la pronta recuperación del servidor de datos SCADA.

Reduciendo el tiempo sin información importante para los operadores que se encuentran controlando el voltaje en el área occidental, evitando así operaciones erróneas en este sistema.

# OBJETIVO

Desarrollo de un sistema encargado de enviar alarmas vía celular en tiempo real a personal de guardia del área de control

Con este proyecto se tendrá una respuesta rápida para la activación del servidor en el menor tiempo posible

# HIPÓTESIS

El servidor de datos SCADA tendrá comunicación vía red TCP/IP con el servidor de intranet Web. Del área de control colocando los datos, donde serán enviados a través de un URL// a la página de celulares para su envío correspondiente.

# INDICE

# ÍNDICE DE TABLAS Y FIGURAS

## Capítulo I



## Capítulo II



## Capítulo III

# CAPÍTULO I

## 1.1.-SERVIDOR DE DATOS SCADA

### Sistema de Supervisión

Este sistema generalmente está definido como forma de control remoto, comprendiendo un arreglo para el control selectivo de las instalaciones localizadas remotamente.

Para tele controlar una instalación en forma económica y acertada se requiere de información confiable, oportuna y adecuada, representando los parámetros importantes. Basándose en dicha información la persona encargada (el operador) del sistema podrá tomar decisiones con mayor rapidez y acierto para mantenerlo dentro de su rango óptimo de operación.

Un servidor de datos SCADA, se puede definir como un conjunto de subsistemas ó elementos independientes, integrados a una lógica común, es un esclavo de una inteligencia de alto orden en el sistema que hace las veces de ojos, manos y oídos, con respecto a la unidad maestra en tiempo real.

## 1.2.- Antecedentes

Las primeras patentes de estos equipos fueron emitidas entre 1890 y 1930. Esas patentes fueron otorgadas principalmente a ingenieros trabajadores en telefonía y otras industrias de la comunicación. En realidad, casi todas las patentes involucraban control remoto y cercanamente supervisión siguiendo las técnicas del cambio del primer teléfono automático instalado en 1892 La Porte, por la Compañía “Automatización eléctrica”.

Desde 1900 hasta los inicios de 1920, muchas variedades de control remoto y sistemas supervisados fueron desarrollados. Muchos de estos, sin embargo, fueron de una y otra clase, es decir, control remoto o supervisión remota

Quizá uno de los precursores del sistema automatizado moderno fue un equipo diseñado en 1921 por John B. Harlow. Este sistema detectaba automáticamente un cambio de estación remota y reportaba ese cambio a un centro de control. En 1923, John J. Ballamy y Rodney G. Richardson, desarrollaron un sistema de control remoto empleando un equivalente de nuestra técnica moderna “confirmar antes de operar (“check before-operate “ ) **[1]**

---

1.- Apuntes del curso, “Sistemas SCADA”: 2005.- central escuela Celaya. CFE

Quizá el primer sistema de registro fue diseñado por Harry E. Hershey en 1927. Este sistema monitoreaba información desde una posición remota e imprimía cualquier cambio en el estado de este sistema, junto con el tiempo y la fecha de cuando el cambio tenía lugar **[1]**

En ese tiempo, por supuesto, existía pequeña diferencia en el tipo de componentes disponibles, todos los sistemas fueron electromecánicos.

Justo como los requerimientos de los sistemas de control hace años fueron más bien simples, así fueron empleadas muchas técnicas. Naturalmente, como el alcance de las aplicaciones de estos equipos cambiaron, así hubo en los fundamentos de la tecnología de los mismos. Los patrones de código fueron mejorados para dar más seguridad y eficacia.

Las técnicas de comunicación fueron cambiadas para permitir elevar la velocidad de transmisión de datos. El advenimiento de los circuitos de estado sólido abrió nuevas posibilidades en la operación y capacidad. En otras palabras, como en cualquier tecnología dinámica, estos equipos de Hoy es el mismo de Ayer.

---

1.- Apuntes del curso, "Sistemas SCADA": 2005.- central escuela Celaya. CFE

El sistema de inspección es un equipo que ha sido diseñado con la finalidad de obtener la información y control de las instalaciones, de un sistema eléctrico a control remoto desde una central como estación maestra, mediante la cual se hace posible la ejecución de controles para apertura o cierre de interruptores, inicio o paro de secuencias automáticas en centrales generadoras, adquisición de información analógica como Voltajes, Amperes, Kilo watts, Kilo watts/Hora y Adquisición Digital, como señalización del estado que guardan los interruptores en una subestación, al igual que también se obtiene la información de alarmas y protecciones de los diferentes dispositivos de que se compone la subestación, todo esto con el fin de proporcionar un mejor servicio y a la vez prever fallas en las subestaciones o centrales generadoras**[1]**.

Para que un sistema de supervisión pueda realizar las tareas asignadas requiere de varios elementos, teniendo cada uno de ellos sus funciones específicas.

En el caso de las redes eléctricas, se requiere de dicho sistema para adquirir información sobre la condición de la red y así mismo poder dirigir señales de mando a los dispositivos a controlar por medio de estaciones remotas ubicadas en las subestaciones y centrales generadoras.

---

1.- Apuntes del curso, "Sistemas SCADA": 2005.- central escuela Celaya. CFE

Ya que estos sitios están geográficamente distribuidos, se requiere de sistemas de comunicaciones para concentrar toda esta información en un centro de control situado en un lugar estratégico. En este centro de control, un sistema de computación se encarga del procedimiento, almacenaje y presentación de la información al operador.

El operador toma las decisiones de acuerdo a los objetivos y metas preestablecidas, con el objeto de mantener el sistema eléctrico dentro de sus límites predefinidos de frecuencia, tensión y economía.

Estaciones de Trabajo o Consolas:

El Sistema incluye 4 consolas con 3 monitores, 5 consolas con 2 monitores y 2 consolas con 1 monitor.

Todas las consolas son estaciones de trabajo graficas tipo Compaq Workstation 500au/GL.

El ambiente grafico de las consolas de operación del sistema está desarrollado bajo X-Windows de UNIX, por medio de estos entornos se genera la Interface Grafica, también se pueden acceder las Terminales X del UNIX en un mismo monitor de consola o a través de la Red.

En la **Figura 1.1** se muestran las características físicas de las Estaciones de Trabajo (Workstations)

**Figura 1.1.-** Características Físicas de las Estaciones de Trabajo (Workstation).

***Servidores de Adquisición de Datos RDAS 1 y 2:***

Estos servidores funcionan como la interfase final del sistema para recibir en sus Modems la conexión física de los canales configurados y procesa la información enviada por los Servidores de Aplicaciones (RAS 1 y 2) y/o la

información recibida desde las UTRs (Unidades Terminales Remotas) configuradas.

Además comunica comandos de control a las UTRs (Unidades Terminales Remotas), de los cuales recibe y verifica señal de retorno y confirmación de la orden y ejecución de la misma. Monitorea y controla los enlaces de comunicación entre el Sistema y las UTRs (Unidades Terminales Remotas).

Los Servidores de Adquisición de Datos (RDAS 1 y 2) son capaces de ejecutar la adquisición de datos desde las UTRs (Unidades Terminales Remotas) y de las UTLs (Unidades Terminales Locales) debido a que están equipados con sistemas ICP (Puerto Inteligente de Comunicaciones) con 8 canales de comunicación serial con las UTRs (Unidades Terminales Remotas). El programa de los ICP procesa información en modo síncrono o asíncrono con las UTRs (Unidades Terminales Remotas).

Este programa esta diseñado para manejar el número de Servidores de Adquisición de Datos (RDAS) y de ICPs (Puerto Inteligente de Comunicaciones) dentro del Sistema, con esto da la facilidad de expansión futura de Servidores de Adquisición de Datos (RDAS) o ICPs (Puerto Inteligente de Comunicaciones), ya que dicho número de equipos es configurable.

La comunicación con las UTRs (Unidades Terminales Remotas) es manejada por el programa de los ICPs (Puerto Inteligente de Comunicaciones) el cuál traduce el protocolo de las UTRs (Unidades Terminales Remotas) a protocolo de Administración de Redes estándar.

Al igual que los Servidores de Aplicaciones (RAS 1 y 2), los Servidores de Adquisición de Datos (RDAS 1 y 2) trabajan en un esquema de Primario y Alterno, cada uno esta equipado con 32 canales seriales de comunicación. Estos canales de comunicación son soportados por el CPU de los Servidores de Adquisición de Datos (RDAS), ubicados en un gabinete modelo [Compaq Modular Computer] (DMCC) basados en tecnología Alpha, el CPU se comunica con los ICPs (Puerto Inteligente de Comunicaciones) por medio del BUS PCI, además tiene 2 controladores Ethernet PCI.

El gabinete [Compaq Modular Computer] (DMCC) además equipado con un gabinete adicional que contiene 14 ranuras PCIMG con 4 ICPs (Puerto Inteligente de Comunicaciones) de 8 puertos que pueden comunicarse a una velocidad de 64 kbps por puerto. Los puertos de los ICPs (Puerto Inteligente de Comunicaciones) tienen como interfase con las UTRs (Unidades Terminales Remotas) un MODEM por puerto. También incluye una tarjeta serial de 4 puertos como interfase para el Tablero Mímico, el Reloj Satelital GPS y los Displays Digitales. Cada Servidor de Adquisición de Datos (RDAS 1 y 2) se comunican directamente con los Servidores de Aplicaciones (RAS 1 y 2) y los

Servidores de Administración de Bases de Datos (RAS 3 Y 4) vía TCP/IP por medio de conmutadores (switches) redundantes Ethernet de 8 puertos a una velocidad de 100 Mbps por medio de las redes LAN1 y LAN2. También se comunican directamente con las consolas del sistema mediante conmutadores (switches) redundantes de 24 puertos a una velocidad de 100 Mbps por medio de las redes LAN3 y LAN4.

**Figura 1.2.-** Características Físicas de los Servidores de Adquisición de Datos (1 y 2).

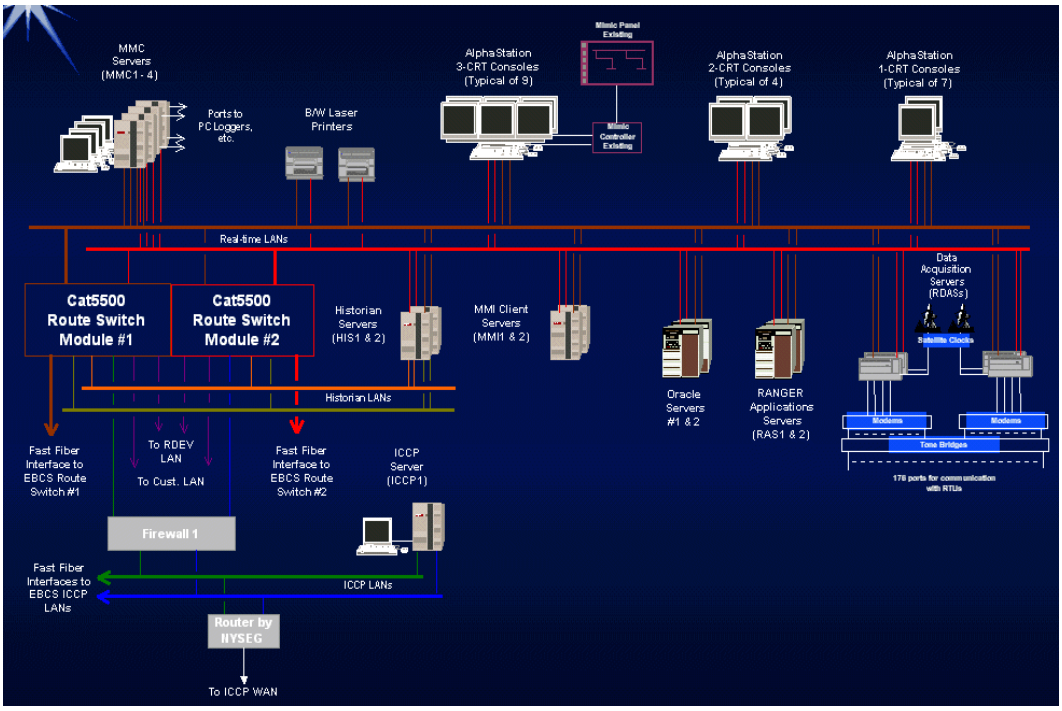
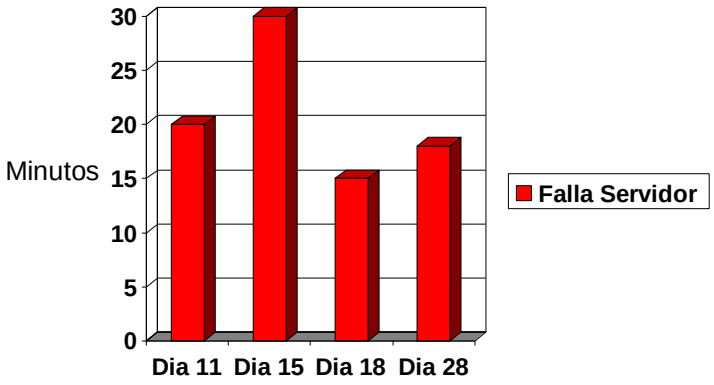



**Figura 1.3.-** Configuración del Sistema SCADA

ESTADÍSTICAS DE DISPONIBILIDAD DEL SERVIDOR




**Total del Mes.- 83min**
**1hr23min.**

**FUENTE.- SERVIDOR HISTORICO**

**Tabla 1.4.-** Estadísticas de Disponibilidad del Servidor.

Los datos obtenidos de las estadísticas anteriores, son datos prevenientes de los servidores de datos SCADA

Existe una partición en el servidor, donde los datos son almacenados e historiados, cada que se genera cualquier alarma, por mínima que sea, así como de gran severidad. Solo con ingresar a esta partición, colocándole una información mínima, que es la fecha, podrá mostrar las alarmas generadas dentro de ese periodo, estamos hablando que el histórico tiene una capacidad de hasta 1 año donde almacena las alarmas, una ves que se encuentra lleno, borra el evento mas viejo manteniendo así un ciclo.

También dichas alarmas son enviadas a los diferentes monitores que comprenden el sistema de datos, pero ha sucedido que en ocasiones, **NO SE ESTÁN OBSERVADO LAS ALARMAS**, o solo aquellas que tienen prioridad, esto quiere decir que, si el servidor de datos SCADA esta fuera de línea, pero de momento no representa mayor peligro para el sistema eléctrico, puede pasar varios minutos, antes de que esta persona realice el reporte, y si una línea eléctrica de dispara o se abre, en ese momento toman acciones correctivas, por citar un ejemplo.

# CAPÍTULO II

## 2.1.- EL LENGUAJE DE PROGRAMACION

El lenguaje de programación es de vital importancia en el desarrollo de este proyecto, ya que nos lleva a la vanguardia en sistemas informáticos, que día a día la programación se ha vuelto necesaria en todo sistema de cómputo. Es por ello que a continuación hablaremos del lenguaje de programación de PERL, el cual se utilizara en este proyecto. y que es un lenguaje de programación de alto nivel que surge a raíz del lenguaje de programación C, con la finalidad de desarrollar uno de menor extensión que cubriera las deficiencias del sed (1),[ awk (2), el Unix Shell] y muchos otros lenguajes y herramientas que tienen para la manipulación de textos y procesos. Una de las razones por la que es ampliamente aceptado en el mundo es porque la programación en alto nivel permite escribir de manera más familiar las instrucciones que desarrolla un equipo, las cuales son traducidas al código máquina mediante un intérprete. Programar en PERL no es vistoso, pero sí funcional. La facilidad de manipulación de archivos y textos que otorga, permite que sea ampliamente empleado para actividades que incluyen el desarrollo rápido de prototipos, sistemas de utilerías, herramientas de software, actividades de administración de sistemas, acceso a bases de datos, programación gráfica, redes y programación de WWW **[1]**

---

1.- extracto del artículo “entérate” de la UNAM publicado en abril 2006

## 2.2.- Antecedentes

La versión número 1.0 de PERL fue liberada el 18 de diciembre de 1987; el core del programa escrito por Larry Wall, comprende una serie de funciones y módulos pequeños que le dan su esencia. Miles de personas han participado en el desarrollo de PERL, al hacer desarrollos, generar módulos, redactar documentación o participar en modificaciones al core: todos cooperan para que el programa funcione mejor y sea empleado ampliamente, lo cual ha provocado su impresionante utilización alrededor del mundo.

## 2.3.- Como funciona PERL

Originalmente ideado para la manipulación de textos, PERL busca ser un lenguaje fácil de usar, capaz de soportar la programación de procesos y la que va orientada a objetos, que es una técnica de programación basada en el uso de bloques donde se agrupan objetos, los cuales a su vez, son unidades de información que contienen datos, procesos u operaciones; su arreglo es más sencillo y con ello se puede resolver un problema específico de manera sencilla.

Basado en el programa de traducción del tipo intérprete para entregar el lenguaje de máquina, tiene como características principales la interfaz DBI, que es el módulo de PERL, en el cual se define una serie de métodos, variables y convenciones que permiten una interfaz independiente porque las aplicaciones de PERL se pueden conectar con diferentes tipos de bases de datos de manera

múltiple al mismo tiempo y permiten mover los datos entre ellas. Además, tiene la capacidad de trabajar con lenguajes del tipo [markup], que son códigos que dan formato a los textos que lee un explorador de Internet (como HTML, XML, entre otros). Asimismo, cuenta con un gran número de ampliaciones, por la capacidad que tiene de ser empleado en diversos sistemas operativos, por lo que está de moda entre desarrolladores, administradores de sistemas, autores de scripts CGI (3), matemáticos, reporteros e inclusive, en gerentes de empresas.

Larry Wall, autor de PERL, creía en la cultura original de compartir recursos por Internet y sus convicciones lo llevaron a apoyar la idea de que fuera de acceso libre para todos los interesados, quienes a su vez han aportado diversas mejoras al programa con la creación de catálogos y módulos compartidos de manera libre.

En el mercado existe una cantidad considerable de lenguajes de programación pero, probablemente, dos de los principales competidores de PERL son JAVA y PHP.

Un factor que ha influido para que PHP se convierta en una herramienta común entre programadores inexpertos es la facilidad que otorga, desde un punto de vista, en el diseño de página Web. PHP es un lenguaje de propósito general especialmente empleado para el desarrollo de páginas Web y puede ser embebido o incrustado (embedded), lo cual significa que es un lenguaje que se interpreta en un modelo de cliente-servidor, donde el servidor se encarga de

ejecutar la programación que puede encontrarse incrustada dentro del lenguaje HTML, pero el control lo tiene el programa. Las páginas PHP están construidas de manera similar a las de HTML. Un código PHP se inserta en la página y se ejecuta cuando la página es solicitada desde Internet.

En el lado opuesto se encuentran los scripts desarrollados en PERL, que corren como programas aislados y crean páginas HTML cuando éste corre. Otro factor es la velocidad y la eficiencia que tiene la programación en PHP sobre PERL. Existen formas para hacer que PERL tenga un desempeño tan veloz como el de PHP, así que esto no es crucial para optar por PHP **[1].**

---

1.- extracto del artículo "entérate" de la UNAM publicado en abril 2006.

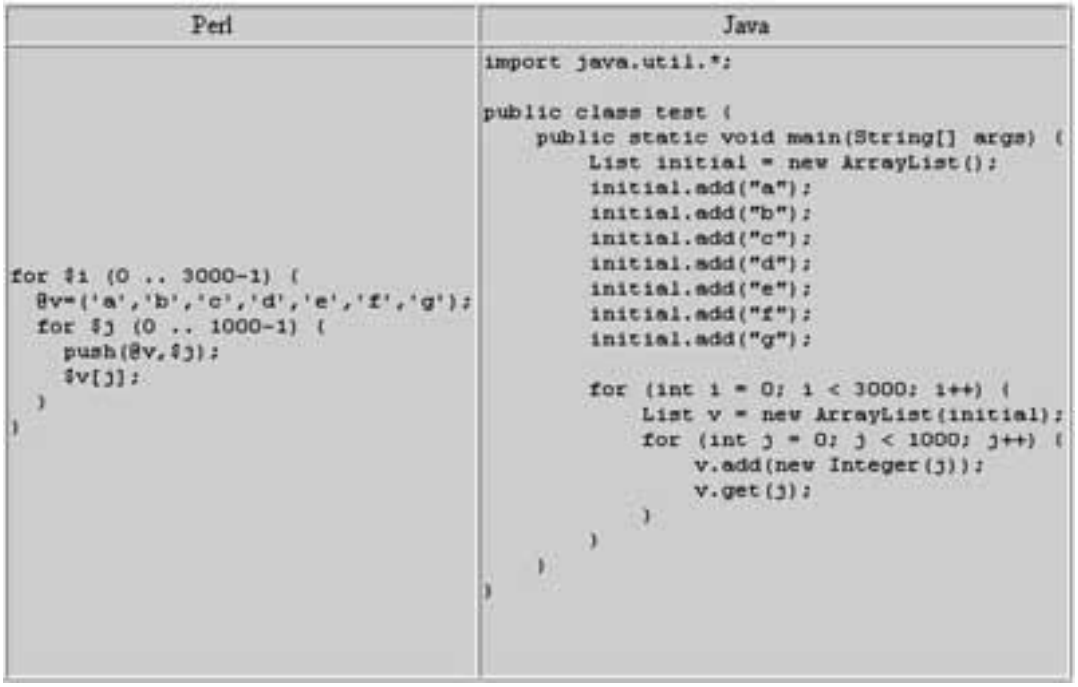
Los programadores con experiencia deben poder manejar ambos programas, sin embargo, el vasto arreglo de opciones con que cuenta PERL lo hace más poderoso y robusto, inclusive, con un poco más de historia dentro del campo de los lenguajes de programación, lo cual es otro punto a su favor debido a que su eficiencia ha sido probada y mejorada, comparado con el reciente PHP disponible desde el año 1995, cuando fue creado a raíz de un conjunto de scripts de PERL empleados para controlar accesos a su trabajo online.

Si se piensa simplemente en el desarrollo de sitios Web, PERL está sobrado en capacidad y esto puede ser un plus ya que permitirá el desarrollo de nuevas herramientas para los portales o servidores, mientras que PHP (Personal Home Page Tools) se encuentra limitado debido a que originalmente se pensó para aplicaciones pequeñas.

Otro de los grandes competidores de PERL es Java, un lenguaje de programación orientado a objetos, desarrollado por James Gosling (junto con colegas de Sun Microsystems), a principios de 1990. Es una derivación de C++ con una sintaxis más sencilla que la de éste, en un ambiente de aplicación más robusto, lo que simplifica la administración de la memoria requerida para su desempeño. Fue concebido para funcionar como una plataforma independiente, es decir, puede operar sin importar el sistema operativo del que se trate; además de que busca ejecutar el código desde fuentes remotas de manera que permita hacerlo en modo seguro.

## 2.4.- Comparativo PERL vs. JAVA

Probablemente, existan muchas personas que prefieran Java, sin embargo en la figura 2.1 se observa la longitud necesaria para desarrollar un pequeño programa que realiza un listado en los lenguajes de PERL y de Java. A pesar de su poder de aplicación en múltiples plataformas de Java, el primero sigue siendo más sencillo.



Perl

```
for $i (0 .. 3000-1) {
  @v=('a','b','c','d','e','f','g');
  for $j (0 .. 1000-1) {
    push(@v,$j);
    $v[j];
  }
}
```

Java

```
import java.util.*;

public class test {
    public static void main(String[] args) {
        List initial = new ArrayList();
        initial.add("a");
        initial.add("b");
        initial.add("c");
        initial.add("d");
        initial.add("e");
        initial.add("f");
        initial.add("g");

        for (int i = 0; i < 3000; i++) {
            List v = new ArrayList(initial);
            for (int j = 0; j < 1000; j++) {
                v.add(new Integer(j));
                v.get(j);
            }
        }
    }
}
```

**Figura 2.1.-.** Listado de funciones PERL vs JAVA

---

Apéndice A. Historia de PHP y proyectos relacionados. (9 de agosto, 2003). PHP: Historia de PHP y proyectos relacionados. Consultado en
http://mx.php.net/history

En esta prueba se observa el tamaño del código comparado entre PERL y Java. Al ejecutar el programa ambos programas corrieron a velocidad lenta.

Perl suele ser mucho más ineficiente que Java, sobre todo debido al tamaño del intérprete que el sistema operativo debe cargar en memoria antes de poder siquiera comenzar el programa. Sin embargo la flexibilidad del lenguaje, su multitud de bibliotecas existentes, el soporte a orientación a objetos y el manejo automático de memoria (emplazamiento y desplazamiento) lo hacen mucho más seguro y robusto.

PERL es un software del tipo código abierto, es decir, los programadores pueden leer, redistribuir y modificar el código fuente de una pieza del software para que la misma evolucione, de esta manera, las personas lo emplean, lo adaptan y reparan los errores que pudiera tener la programación. Esto ha permitido que en comparación con el software cuyo código se encuentre oculto a los usuarios, se desarrolle a una gran velocidad y sea muy confiable en su desempeño. PERL se encuentra bajo una Licencia publica general, lo cual asegura que se tiene la libertad para distribuir copias de software libre y realizar los cambios que fueran necesarios; se cuenta asimismo, con el código fuente o puede solicitarse, de forma tal, que se pueden realizar las modificaciones en el software o usar piezas de éste en programas nuevos que también sean gratuitos.

Afortunadamente, la utilización de PERL no está restringida para una plataforma en particular, gracias a la flexibilidad intrínseca del programa y a la importante participación de los usuarios; este lenguaje puede emplearse en sistemas Unix, Windows, Macintosh, VMS, Sun, Solaris, etcétera.

Muchos de los beneficios que tienen los desarrolladores que emplean este lenguaje se deben a la utilización del intérprete, programa que traduce una línea de código fuente a la vez y una vez que la traduce, la ejecuta.

Tal vez ésta sea la razón por la que se ha confiado este lenguaje al desarrollo de proyectos de misión crítica en sectores públicos y privados, entre los que se pueden mencionar la generación de reportes en tiempo real para tableros de mando de múltiples industrias, sistemas para transacciones bancarias y sistemas de autenticación y registro único de usuarios.

"PERL interpreta una instrucción dentro de un contexto, por ello el resultado puede o no ser lo que se buscaba, si no se realiza dentro del contexto adecuado", comenta el Ing. Enzo Molino, director de Informática del Instituto Nacional para la Evaluación de la Educación (INEE), "no es tan fácil entrarle" puntualiza después de indicar que el desarrollador de este lenguaje tenía como profesión la de lingüista. Continua diciendo: "pero tiene grandes ventajas: es elegante, es eficiente y tiene una gran fuente de bibliotecas desarrolladas por otras personas con PERL es una herramienta muy poderosa impulsada por el interés de los usuarios para proponer mejoras y tener de esta forma software gratuito de calidad y ajustable a las necesidades particulares de cada usuario

pero, por otro lado, dada esa complejidad que maneja, se requiere bastante tiempo para hacer desarrollos en este lenguaje cuando uno es principiante, en ese sentido, se traduce muchas veces en un rechazo por parte de los programadores con experiencia en otros lenguajes.

El lenguaje de programación PERL, es un lenguaje de programación robusto y completo, como se observa en este capitulo, es por eso que se toma la decisión de usarlo en este proyecto, esto garantizara un buen desempeño del programa y por ende el envío oportuno de mensajes a través de Internet al celular de guardia del ACOC, y con esto una mejor respuesta oportuna para tener disponible el servidor de datos SCADA. **[1].**

---

1.-Instituto Nacional para la Evaluación de la Educación.-2007

# CAPÍTULO III

## 3.1.-SISTEMA PROPUESTO

### 3.1.- Desarrollo

El archivo que se encarga de realizar el monitoreo del sistema y a la vez de enviar el mensaje o mensajes a los diferentes celulares de guardia del personal, es el, NOTIFICA.PL, dicho archivo es el encargado de realizar el envío a través de una dirección la cual se encarga de enviar los mensajes cuando el servidor de datos SCADA se encuentra fuera de línea.

El proceso inicia cuando se ejecuta la tarea [script], en el servidor web, dicha tarea se estará ejecutando cada 2 minutos, esto con el fin de no saturar dicho servidor y poder tener un margen máximo de 2 minutos en caso de que existiera una falla del servidor SCADA.

Cuando ocurre la falla en el servidor, la tarea será capaz de darse cuenta, ya que se tiene una tabla de eventos, al aparecer alarma de FALLA, inmediatamente abre un browser donde tiene declarada la pagina de envió de mensajes, y en base a el orden que se encuentran los campos, estos son llenados de manera correcta, como son, numero de celular, remitente y finalmente el envió al celular correspondiente.

La siguiente figura muestra la interacción de los diferentes procesos.

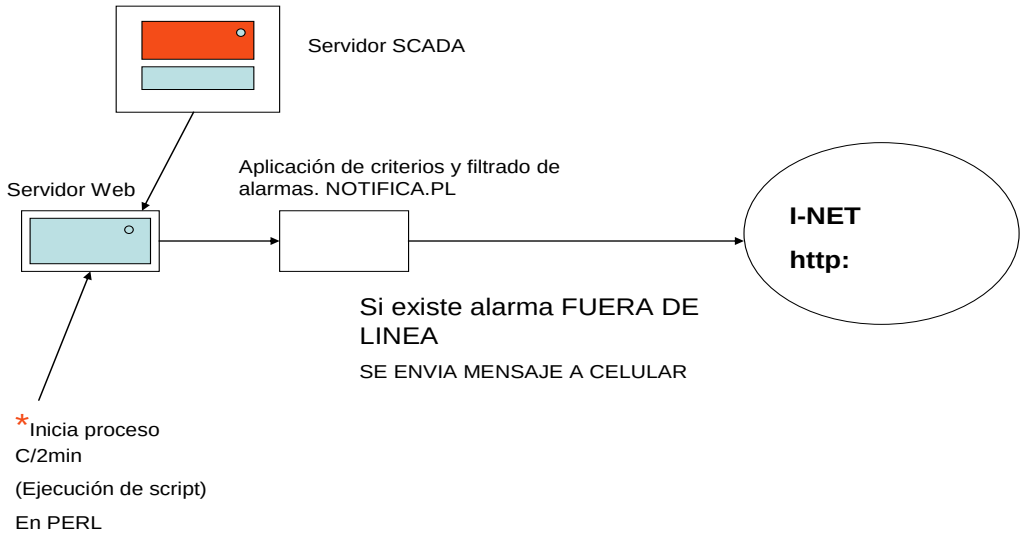



**Figura 3.1** diagrama a bloques.

## El programa funciona de la siguiente manera:

Como punto inicial, se debe tener instalado y configurado el programa PERL, en el servidor Web, así mismo una conexión entre éste servidor y el servidor de datos SCADA, para posteriormente instalar el lenguaje de programación PERL, y finalmente cargar el archivo de texto que contiene la programación.

El archivo NOTIFICA.PL, es el encargado de realizar la ejecución para los diferentes criterios como lo son, el de estar recibiendo el estado que guarda el servidor, así como una vez que es producida una alarma, poder abrir un browser

y colocar el texto en el campo correspondiente, para finalmente enviar el mensaje de aviso.

“SCRIPT NOTIFICA.PL”

```
use DBI;
use Net::SMTP;
use Date::Manip;
use POSIX qw(strftime);
use WWW::Mechanize;
use Storable;



## comparación de strings:
                        # $x = $a cmp $b
                        # $x queda con  -1  si $a lt $b
                        # $x queda con   0  si $a eq $b
                # $x queda con   1  si $a gt $b

## match del patron en un string...
                        # aqui la expresion regular es una expresión lógica...
                        # devuelve verdad si el string contiene un patron
                #     $a = "abcdef";
                        #     $a =~ /bc/;    #  es verdadero   # if ($a =~ /bc/)
                        #     $a =~ /ba/;    #  es falso

# puntos de referencia
                        #^  es el comienzo del string
                #\w es un caracter de palabra: digito o letra
                #[abcef]  una de esas 5 letras
```

ESTE ES EL FILTRO DONDE SOLO SE ACTIVA CON  ALARMA INITIAL

```
%ClavesAlarmas= (
        "Servidor A enlace OCOC1                STNDBY"=>PRG,
        "Servidor A enlace OCOC1                INITAL"=>PRG,= FALLA
        "Servidor A enlace OCOC1                ESTABL (ococ1)"=>PRG,
```

```
$tiempo_ini = "07:00";   # $tiempo_fin = "18:00";   # $error=0;  ##

        ## 0 Implica que no existe error en el sistema




        print "Tiempo Actual: $tiempoActual\n";

        print "Tiempo hace 2 min: $hace5min\n";

        &ConsultaOracle($tiempoActual,$hace5min);

        #print "TIEMPO\t\t\t\tSUBNAM\t\tPNTNAM\t\t\tMENSAJE\n";




        #}

        $i=0;
        foreach $alarma ( keys %alarmasValidas )  ##ciclo foreach para cada elemento
llave del arreglo asociativo %alarmasValidas

        {
           $b = exists $ClavesAlarmas{$alarmasValidas{$alarma}{MESSAGE}};   ##
Esta función asigna 1 a $b si encuentra
#             if ( $b == 1)                                  ## el mensaje de alarma leído en el
arreglo
#             { print "$b\t";  }                              ## ClavesAlarmas, Si no lo
encuentra $b tendrá " "


#             undef ($txtAlarma);
            $txtAlarma=$alarmasValidas{$alarma}{MESSAGE};
            $tiempoAlarma=substr($alarmasValidas{$alarma}{TIEMPO},11,5);
            #print "$tiempoAlarma\n";

            ### Este if realiza la busqueda de la cadena /******/ en el mensaje de alarma
```

“parte de La programación  se encuentra dentro de los anexos al final de este proyecto, para mayor referencia”.

Dicho script el cual corre cada 2 minutos, está cargado en el servidor web I-net, esto con el fin de no saturar al servidor Web, y tener como máximo un tiempo fuera del servidor de 2 minutos, y así poder responder en forma mas inmediata a la atención del mismo.

Dicho script se da de alta dentro del servidor web y se da parámetro de ejecución cada 2 minutos, el cual esta supervisando las alarmas generadas del sistema, de acuerdo a los criterios de evaluación que se muestran en el archivo NOTIFICA.PL, que se encuentra en los anexos en la parte final.

El servidor Web de intranet, tiene una conexión hacia el servidor de datos, de donde extrae de la tabla evtmsg, este archivo esta almacenando los eventos ocurridos en el servidor de datos, de donde se genera la alarma de FALLA FUERA DE LÍNEA y es enviada a los monitores, así mismo será ya enviada a los celulares del personal de guardia del Área de control

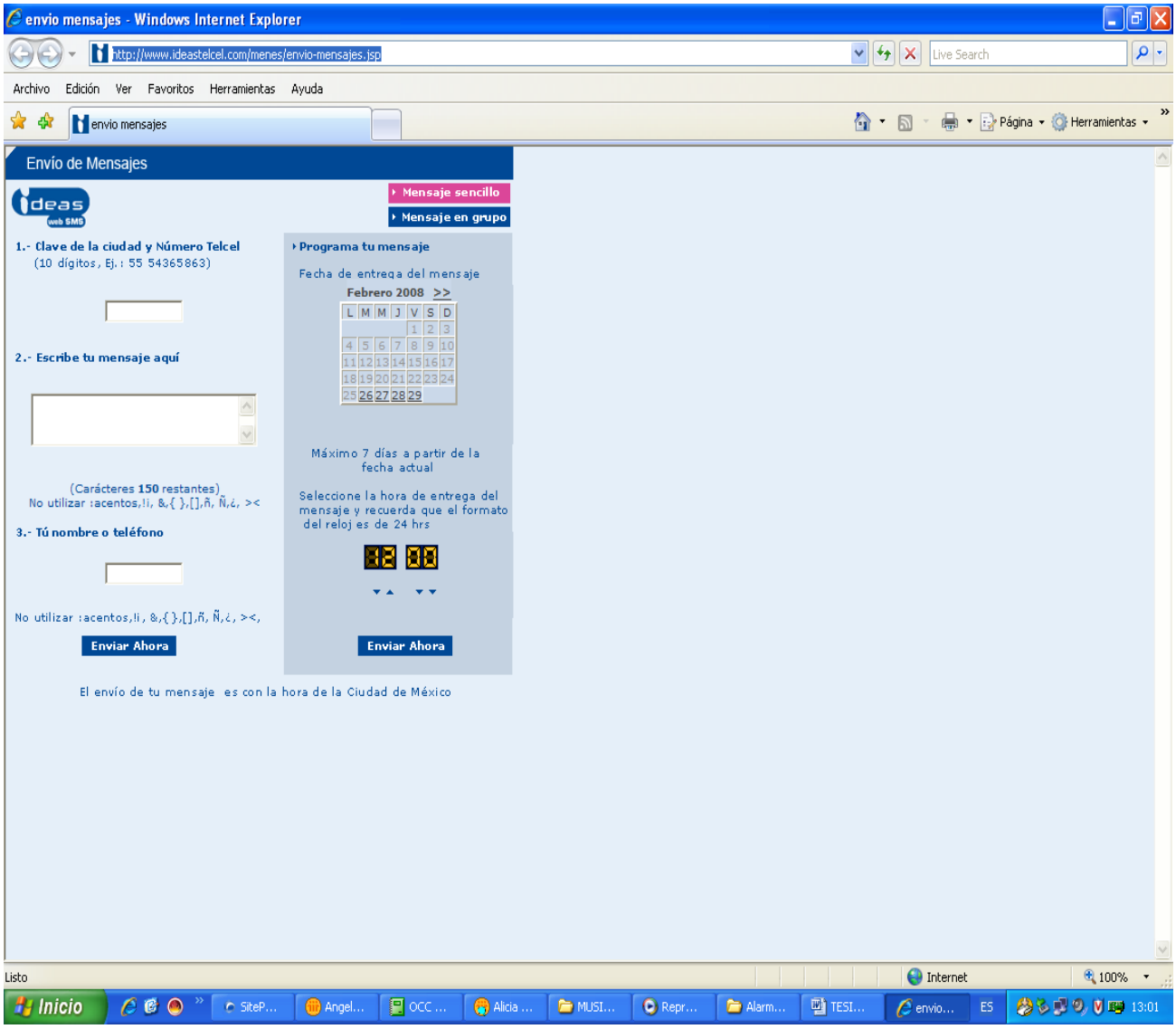



**Figura 3.2.-** pagina de envío mensajes a celular.

En esta página el programa tiene la capacidad de accesar al navegador, colocar los campos necesarios, así como el envió al número celular previamente configurado con anterioridad, alertando a la persona que se encuentre de guardia, teniendo con un máximo de 2 minutos fuera, comparado con alrededor de 30 hasta 45 minutos fuera el servidor

A continuación veremos el código y es aquí donde ocurre el momento en que el scritp ejecuta abrir el browser, es parte de la programación del NOTIFICA.PL

AQUÍ SE ABRE UN BROWSER WWW.IDEASTELCEL, SE ENVIA EL MENSAJE

```
#--- Funcion EnviarSMS
sub EnviarSMS
{
   $browser = WWW::Mechanize->new() or return ($error=8);
   $browser->get("http://www.ideastelcel.com/menes/envio-mensajes.jsp") or return
($error=8);
   $remitente=" ,Remitente: COMNES OCC “
   $fields = {
      'tel1' => $telefono,
      'mensaje' => $Mensaje,
      'nomtel' => $remitente
     };

   $browser->submit_form(form_name => 'msgpop',fields => $fields) or return
($error=8);
   return($error=0);
}
#--- FIN Funcion EnviarSMS
```

## 3.2.- Requisitos previos, antes de que el programa pueda funcionar:

| |
|---|
| - Tener establecida y con salida a Internet el servidor WEB |
| - Tener instalado y configurado el PERL en el servidor WEB |
| - Tener abierto un puerto para que éste servidor se pueda conectar a la instancia dbe del Network manager y obtener información de la tabla evtmsg. |
| - El teléfono del usuario final debe contar con el servicio de envío de mensajes por Internet, este lo debe contratar el Área de Control |
| |

**Fig. 3.3.-** Requisitos previos

Los teléfonos celulares pertenecientes al personal de guardia, ya se tienen contratados dichos servicios y que van incluidos en el pago mensual que realiza la administración

De igual manera, las políticas que se establecieron en los patrones de búsqueda de texto, también pueden ser modificadas y ajustadas a sus conveniencias, ya que se aprovechan las propiedades de búsqueda de patrones regulares que ofrece el Perl

## 3.3.- Instalación y Configuración

A continuación se instalara el programa PERL, para tener el compilador y poder ejecutar el script.

**#! / Usr / bin / perl**

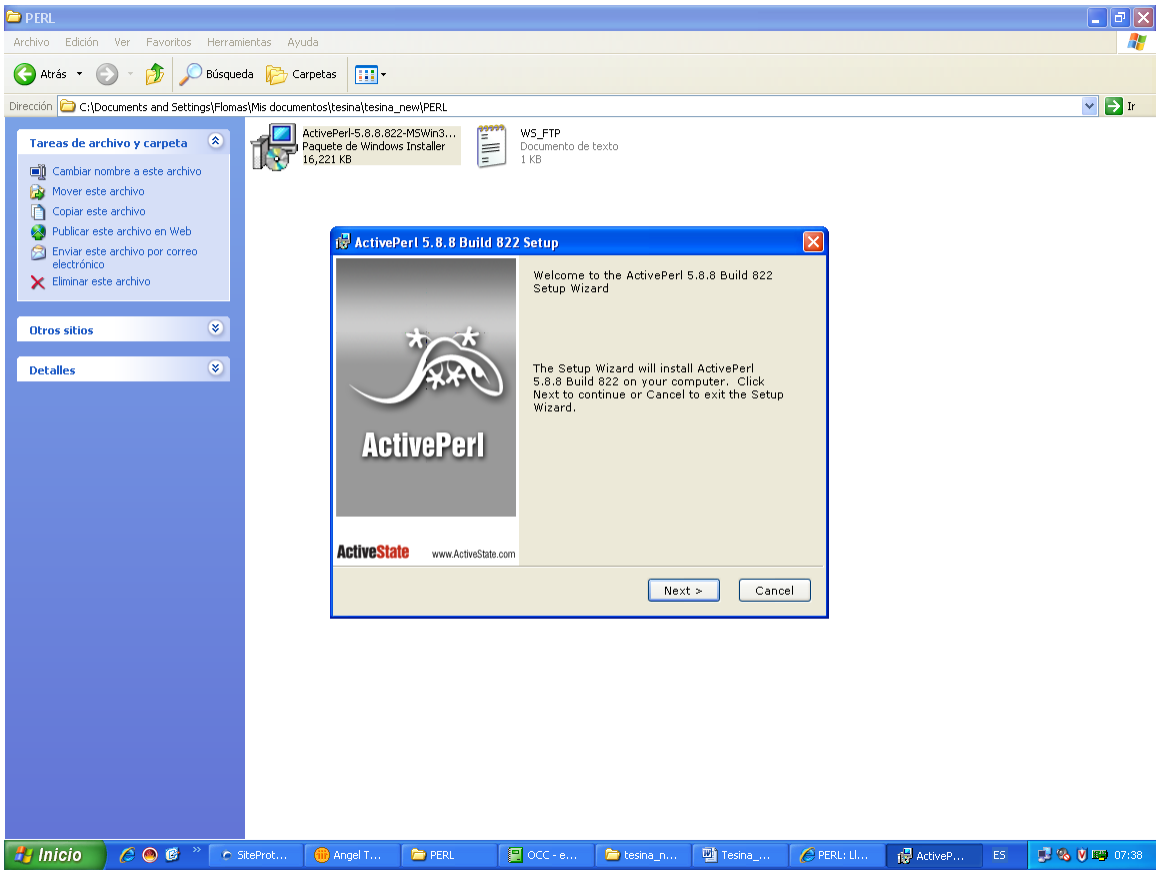



**Figura 3.3.-** instalación Perl en Windows.

Se instala el programa, activeperl versión 5.8.8 en el servidor Web de intranet, en este servidor reside toda la programación y la tarea que ejecutara la aplicación para el correcto envío del mensaje.

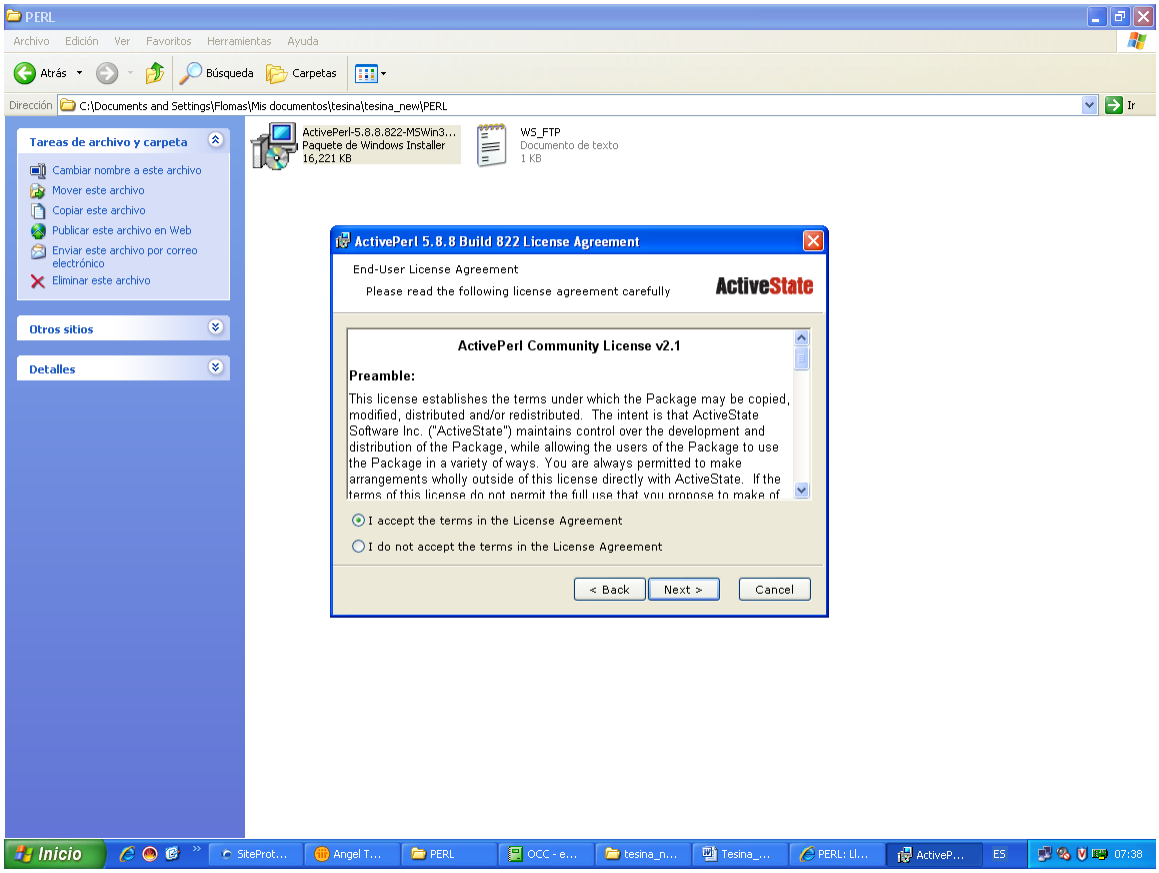



**Figura 3.4.-** Cont. Instalación.

## 3.4.- Configuración

Dentro de la configuración del sistema, se tiene que, una vez instalado perl, dentro del servidor Web, deberá existir conexión física del servidor web hacia el servidor que se quiere proteger, en este caso el servidor SCADA, asi mismo se tendrá vía TCP/IP la comunicación entre los servidores

Dentro del servidor SCADA, se tiene una tabla que se llama event.msg, que es donde llegan todas y cada una de las alarmas, provenientes de dichos servidores, como en esta tabla se tienen en forma cruda las alarmas, fue necesario determinar el criterio de evaluación y de programación, para poder dejar pasar por alto las alarmas que no tienen relación directa con la falla del servidor SCADA, ya que si falla dicho servidor, las alarmas se almacenan de forma histórica, esto significa que no perdemos ninguna alarma, pero de qué sirve, si en el momento en que se generan **NO PODEMOS SUPERVISARLAS.**

Es por eso que dicho sistema es viable y que tendría en un alto grado de disponibilidad el servidor, teniendo datos en tiempo real y seguros.

## Resumen

En esta etapa se inicia con la programación que se observa en el **anexo 1** así como la creación en el servidor Web de intranet el script a ejecutarse cada 2 minutos, aplicando los criterios de filtrado y envío mensaje vía celular.

# CAPÍTULO IV

## 4.1.- CONCLUSIONES

Con los sistemas implementados para el telecontrol en todas las áreas de Control como en Subáreas y Subestaciones, se ha logrado aumentar su seguridad y disminuir su tiempo de restablecimiento en alguna falla dando así un mejor servicio a los usuarios de este producto, de igual manera ha facilitado a los operadores el control de la Red Eléctrica Nacional, todo esto gracias a los avances electrónicos, de comunicaciones, aplicaciones y equipos de computo.

De igual manera se han disminuido los costos, sin duda son avances que se podría aplicar en cualquier tipo de empresa publica o privada en la que tengan el control de dispositivos que forman parte de procesos de las mismas.

Es así como en conclusión se pude decir que los sistemas SCADA son una de las áreas donde personal calificado puede encontrar una fuente de trabajo ya que se combinan distintas áreas pudiendo desempeñarse los Ingenieros en Comunicaciones y Electrónica, Ingenieros en Computación, Licenciados en Informática, Físicos Matemáticos, Ingenieros Mecánico

Electricista, ingenieros industriales, entre otras afines, contribuyendo en conjunto con los conocimientos adquiridos en la Universidad a los problemas presentados en la vida cotidiana mejorando tiempo, dinero, esfuerzo y seguridad a los trabajadores.

Así mismo podemos concluir que con el desarrollo de un sistema encargado de enviar alarmas vía celular en tiempo real a personal de guardia del área de control, y que Con este proyecto se tendrá una respuesta rápida para la activación del servidor en el menor tiempo posible, teniendo así un control total del sistema eléctrico occidental, ofreciendo calidad y servicio a nuestros clientes, para poder continuar emprendiendo logros día a día, y siendo mejores, manteniéndonos como una EMPRESA DE CLASE MUNDIAL.

Durante el desarrollo de la metodología, se observo la necesidad de poner en practica los conocimientos adquiridos, no solo de la facultad de ingeniería, sino de la especialidad en seguridad informática y manejo de la información, ya que se observo la necesidad de preservar la seguridad ante todo, y no comprometer los equipos para un posible ataque, ya que el servidor de datos SCADA tendrá comunicación vía red TCP/IP con el servidor de intranet Web. Del área de control

colocando los datos, donde serán enviados a través de un URL// a la página de celulares para su envío correspondiente.

Tenemos hasta el momento todas las herramientas y las bases para poner en marcha el proyecto antes mencionado, mejorando así considerablemente, tanto los recursos del Área, aprovechando el uso de la tecnología actual y sobre todo mantener en un alto grado de disponibilidad y confiabilidad los diferentes sistemas que interactúan para un buen y correcto funcionamiento del sistema eléctrico occidental.

No obstante cabe mencionar los trabajos a futuro, ya que el alcance de esta tesina es solo investigación y desarrollo, y que se espera que continuando con los esfuerzos de **COMISIÓN FEDERAL DE ELECTRICIDAD Y DEL INSTITUTO POLITÉCNICO NACIONAL**, continúen con la capacitación, formación y preparación del personal, y enfrentar los retos venideros de seguridad informática, con la continuación de la Maestría en dicha materia, continuando con la mejora continua y poder realizar las pruebas piloto, la puesta en servicio y operación de este interesante proyecto, contribuyendo y mejorando los sistemas de electricidad, dando un servicio de calidad.

# CAPÍTULO V

## 5.1.- BIBLIOGRAFÍA

**BIBLIOGRAFÍA-Referencias**

1. Extreme Networks and Sygate Technologies. Guía de Implementación. <http://www.sygate.com>. [Consulta: Septiembre 2006]

2. Apéndice A. Historia de PHP y proyectos relacionados. (9 de agosto, 2003). PHP: Historia de PHP y proyectos relacionados. Consultado en http://mx.php.net/history

3. Ashton, Elaine; "The timeline of PERL and its culture" (1999-2001). The Timeline of PERL and its Culture v3.0_0505, de http://history. PERL.org/PERLTimeline.html

4. Castagneto, Jesús; Ragwat, Harish; Schumann, Sascha; Scollo, Christopher; Veliath, Deepak (1999). Professional PHP Programming (edición electrónica). pág. 68.

5. Connell, Mike. "Pitón VS Perl VS Java VS C++ Runtimes." (septiembre 9, 2002). Consultado en: http://furryland.org/~mikec/bench/

6. Christiansen, Tom. "Downloading the latest version of PERL" (N. d.) Stable Production Release (Current Version), de: http://www.PERL.com/download.csp

7. Download PERL Dev Kit (2005). Active State – Product to Download, de:

http://www.activestate.com/Products/Download/Download.plex?id=PERLDevKit

8. Frequently Asked Questions (2005). FAQ – search.cpan.org, de: http://search.cpan.org/

9. Hietaniem, Jarkko. PERL Ports (Binary Distributions) (1995-2005). CPAN/ports.

(Encabezado, para. 1), de la página http://www.cpan.org/ports/index.html

10. Java programming language (n. d.). Java programming language – Wikipedia, the free encyclopedia. Consultado el 13 de marzo de 2006.

11. URL: http://en.wikipedia.org/wiki/Java_programming_language

12. Kirrily,Robert (N. d.) PERL 5.8.7 documentation. PERLintroPERLdoc.PERL.org,de: http://PERLdoc.PERL.org/PERLintro.html#What-is-PERL%3f

13. Molino, Enzo (plática entablada el día 27 de enero de 2006). Director de Informática del Instituto Nacional para la Evaluación de la Educación.

14. Núñez Zuleta, José Vicente. "Introducción a la programación en PERL, CGI y Javascript" (n. d.). Consultado el 15 de Diciembre de 2006.

15. Prentice Hall, Inc. (2006) Material para el alumno. Presentación del capítulo 11: daley8_ppt_11.ppt. de: http://wps.prenhall.com/bp_daley_ciyf_8

16. PERL vs PHP. (n. d.) PERL vs PHP. Consultado el 14 de marzo de 2006. URL: http://www.mediacollege.com/internet/perl/perl-vs-php.html

17. The PERL directory: about PERL. (N. d.). About PERL – PERL.org, de: http://www.PERL.org/about.html

18. Apuntes de curso sistemas SCADA: 2005.- central escuela celaya. CFE

# ANEXO 1

En el presente anexo se tiene el Archivo NOTIFICA.PL, el cual se encarga de aplicar los criterios de filtrado de las alarmas, así como la ejecución del envío de mensajes vía celular

Se presenta en este anexo para hacer referencia, pero este archivo esta con extensión .txt, para que el script lo ejecute y se pueda realizar el envío de mensajes.

También se tienen los criterios de enviar el mensaje a diferentes números de celulares, siempre y cuando tengan contratado el servicio de mensajería a través de Internet, esta contratación ya cuenta con ella el área de control, ya que a nivel nacional hay un convenio con la compañía de teléfonos celulares mediante una licitación publica.

```
use DBI;
use Net::SMTP;
use Date::Manip;
use POSIX qw(strftime);
use WWW::Mechanize;
use Storable;


## comparación de strings:
                        # $x = $a cmp $b
                        # $x queda con  -1  si $a lt $b
                        # $x queda con   0  si $a eq $b
                # $x queda con   1  si $a gt $b

## match del patron en un string...
                        # aqui la expresion regular es una expresión lógica...
                        # devuelve verdad si el string contiene un patron
                #     $a = "abcdef";
                        #     $a =~ /bc/;    #  es verdadero   # if ($a =~ /bc/)
                        #     $a =~ /ba/;    #  es falso

# puntos de referencia
                      #^  es el comienzo del string
                #\w es un caracter de palabra: digito o letra
                #[abcef]  una de esas 5 letras
```

ESTE ES EL FILTRO DONDE SOLO SE ACTIVA CON  ALARMA FALLA

```
%ClavesAlarmas= (
        "Servidor A enlace OCOC1                STNDBY"=>PRG,
        "Servidor A enlace OCOC1                INITAL"=>PRG,= FALLA
        "Servidor A enlace OCOC1                ESTABL (ococ1)"=>PRG,



$tiempoActual = strftime "%d-%m-%Y %H:%M", localtime;       ##Variable que
determina la fecha y hora actual en formato DD-MM-AAAA HH AM/PM
$hace5min = strftime "%d-%m-%Y %H:%M", localtime(time-120); ##Variable que
```

```
        print "Tiempo Actual: $tiempoActual\n";
        print "Tiempo hace 2 min: $hace5min\n";
        &ConsultaOracle($tiempoActual,$hace5min);
        #print "TIEMPO\t\t\t\tSUBNAM\t\tPNTNAM\t\t\tMENSAJE\n";
        #$mayusculas=uc($variable);    #### esto convierte la variable string en
mayusculas




        $i=0;
        foreach $alarma ( keys %alarmasValidas )  ##ciclo foreach para cada elemento
llave del arreglo asociativo %alarmasValidas
        {
           $b = exists $ClavesAlarmas{$alarmasValidas{$alarma}{MESSAGE}};   ##
Esta función asigna 1 a $b si encuentra
#            if ( $b == 1)                                  ## el mensaje de alarma leído en el
arreglo
#            { print "$b\t";  }                              ## ClavesAlarmas, Si no lo
encuentra $b tendrá " "

#            undef ($txtAlarma);
            $txtAlarma=$alarmasValidas{$alarma}{MESSAGE};
            $tiempoAlarma=substr($alarmasValidas{$alarma}{TIEMPO},11,5);
            #print "$tiempoAlarma\n";

           #if ($txtAlarma =~ /Cambio No Comandado/)
           #print "$i\t";   #contador del indice del hash
           #print
"$alarmasValidas{$i}{TIEMPO}\t$alarmasValidas{$i}{SUBNAM}\t$alarmasValidas{$

"$alarmasValidas{$alarma}{TIEMPO}\t$alarmasValidas{$alarma}{SUBNAM}\t$alarm
asValidas{$alarma}{PNTNAM}\t$alarmasValidas{$alarma}{MESSAGE}\n";

           if ($alarmasValidas{$alarma}{SUBNAM} =~ /^05   / and
$alarmasValidas{$alarma}{PNTNAM} =~ /^IN/ and
$alarmasValidas{$alarma}{MESSAGE}=~ /Cambio No Comandado/ )
              {
               $Mensaje=$alarmasValidas{$alarma}{TIEMPO}."
".$alarmasValidas{$alarma}{SUBNAM}." ".$alarmasValidas{$alarma}{PNTNAM}."
".$alarmasValidas{$alarma}{MESSAGE};
               $DescripcionError=$Mensaje;
```

AQUÍ SE CONFIGURAN LOS DIFERENTES NUMEROS DE CELULAR

```
            #$telefono="xxxxxxxxxx"; ## Telefono Celular de
            #$telefono="xxxxxxxxxx"; ## Telefono Celular de
                                    #@tels_SIAE=("xxxxxxxxxx","xxxxxxxxxx");
                                    #foreach $telefono (@tel_SIAE)
                                    #{&EnviarSMS($telefono,$mensaje);}
            #&EnviarSMS($telefono,$Mensaje);
            #undef ($telefono);

            #undef (@recipients);
            open(OUTFILE, ">>al_ope.txt") or die "No se puede abrir archivo al_ope.txt:
$!";     ## Esto guarda en un archivo
            print OUTFILE $Mensaje;                                          ## de texto
la alarma enviada
            print OUTFILE "\n";                                               ## con fines
de revision de lo
            close OUTFILE;                                                   ## que se
manda
          }
        else
          { #print "$El punto no es IN\n";
           ########## Verificar alarmas de computo

            if ($alarmasValidas{$alarma}{MESSAGE}=~ /falla del dispositivo de
control/ )
            {

#&ProcesaAlarmaSistema($alarmasValidas{$alarma}{TIEMPO},$alarmasValidas{$alar
ma}{MESSAGE});
              #$Mensaje=$alarmasValidas{$alarma}{TIEMPO}." "."ALARMA DE
SISTEMA"." ".$alarmasValidas{$alarma}{MESSAGE};
              #$DescripcionError=$Mensaje;
              print "$Mensaje\n";
              #$recipients[0]=
              #$recipients[1]=
              #$telefono="xxxxxxxxxx"; ## Telefono Celular de
              #&EnviarSMS($telefono,$Mensaje);
              #undef ($telefono);
              #&NotificaCorreo(@recipients,$DescripcionError); ##Notificacion via
correo electronico
              #$error=0; ##En caso de encontrar un error en el proceso notifica de este
pero continua con el proceso
```

```
                #undef (@recipients);
               }

               if ($alarmasValidas{$alarma}{MESSAGE}=~ /^<MRST>/ )
               {

&ProcesaAlarmaSistema($alarmasValidas{$alarma}{TIEMPO},$alarmasValidas{$alar
ma}{MESSAGE});

&ProcesaAlarmaSistemaSUMAR($alarmasValidas{$alarma}{TIEMPO},$alarmasValida
s{$alarma}{MESSAGE});
               }


               if (substr($alarmasValidas{$alarma}{MESSAGE},0,6) eq "<MFAL>")
               {

&ProcesaAlarmaSistema($alarmasValidas{$alarma}{TIEMPO},$alarmasValidas{$alar
ma}{MESSAGE});

&ProcesaAlarmaSistemaSUMAR($alarmasValidas{$alarma}{TIEMPO},$alarmasValida
s{$alarma}{MESSAGE});
                #$Mensaje=$alarmasValidas{$alarma}{TIEMPO}." "."ALARMA DE
SISTEMA"." ".$alarmasValidas{$alarma}{MESSAGE};
                #$DescripcionError=$Mensaje;
                #print "$Mensaje\n";
                #$recipients[0]='
                #$recipients[1]='
                #$telefono="xxxxxxxxxx"; ## Telefono Celular de
                #&EnviarSMS($telefono,$Mensaje);
                #undef ($telefono);
                #&NotificaCorreo(@recipients,$DescripcionError); ##Notificacion via
correo electronico
                #undef (@recipients);
               }


               if (substr($alarmasValidas{$alarma}{MESSAGE},0,6) eq "<NDBN>")
               {
```

AQUÍ SE ABRE UN BROWSER WWW.IDEASTELCEL, SE ENVIA EL MENSAJE

```
#--- Funcion EnviarSMS
sub EnviarSMS
{
   $browser = WWW::Mechanize->new() or return ($error=8);
   $browser->get("http://www.ideastelcel.com/menes/envio-mensajes.jsp") or return
($error=8);
   $remitente=" ,Remitente: COMNES OCC “
   $fields = {
       'tel1' => $telefono,
       'mensaje' => $Mensaje,
       'nomtel' => $remitente
     };

   $browser->submit_form(form_name => 'msgpop',fields => $fields) or return
($error=8);
   return($error=0);
}
#--- FIN Funcion EnviarSMS
```

# GLOSARIO

**CFE - Comisión Federal de Electricidad**
Compañía encargada de la generación, transformación, transmisión, distribución y control de la energía eléctrica en México.

**ABB – Asea Brown Bovery**
Compañía que actualmente vende, da soporte y mantenimiento al Sistema SCADA.

**CISCO**
Compañía encargada a vender equipos de comunicaciones en redes de alta velocidad para grandes y pequeñas corporaciones.

**DIGITAL**
Compañía dedicada a ensamblar equipos de computo actualmente absorbida por HP.

**COMPAQ**
Compañía dedicada a ensamblar equipos de cómputo actualmente asociada con HP.

**HP – Hewlett Packard**
Compañía dedicada a ensamblar equipos de cómputo.

**BNM - Bailey Network Manger**
Compañía dedicada a integrar Sistemas SCADA actualmente absorbida por ABBNM.

**ORACLE**
Compañía dedicada a desarrollar sistemas Administradores y Bases de Datos.

**MICROSOFT**
Compañía dedicada a desarrollar Sistemas Operativos y paquetes de Bases de Datos y Oficina tales como procesadores de texto, hojas de cálculo y editores de presentaciones entre otros.

**TELMEX – Teléfonos de México**
Compañía encargada de proporcionar servicios de comunicación por voz y datos a con mayor cobertura en México.

**SCADA – Sistema de Control y Adquisición de Datos**
Conjunto de computadoras que integran un sistema para supervisar, administrar y operar procesos. Tema principal del que se trata esta tesis enfocado a la administración y control de la energía eléctrica.

Es el equipo que se encuentra generalmente en los Centros de Control de la Comisión Federal de Electricidad con el propósito de obtener información de campo y procesarla para la adecuada operación, control y supervisión del Sistema Eléctrico de Potencia.

**UTM - Unidad Terminal Maestra**
El sistema maestro que interroga y concentra los datos obtenidos de las UTRs.

**UTR – Unidad Terminal Remota**
Computadoras que obtienen datos de los puntos que tienen conectados y que regularmente se encuentran instalados en las plantas generadoras o subestaciones en regiones alejadas de la UTM.

Equipos localizados generalmente en las subestaciones o plantas generadoras que se encuentran en puntos remotos con el propósito de reportar la información de tiempo real al Sistema SCADA.

**SEN**
Sistema Eléctrico Nacional.

**SEP**
Sistema Eléctrico de Potencia.

**Centro Nacional de Control de Energía (CENACE)**
Es la entidad de la Comisión Federal de Electricidad encargada de planear, operar, supervisar y controlar el Sistema Eléctrico de Potencia a nivel Nacional. Por su importancia y nivel de tensión supervisado esta subdividido en Centros de Control de área y Centros de Control Subarea.

**Nivel Inferior**
Sistema de supervisión, control o informático considerado en un nivel jerárquico inferior para el intercambio de datos o recepción de comandos.

**Nivel Superior**
Sistema de supervisión, control o informático considerado en un nivel jerárquico superior para el intercambio de datos o recepción de comandos.

**Protocolo de Comunicación**
Formato y/o tecnología que en forma lógica y/o física realiza la función de intercambio de información y comandos entre diferentes sistemas de cómputo independientes o en redes ya sean normalizados o de procesos específicos del tipo SCADA.

**Base de Datos**
Conjunto de datos ordenados u organizados relacionados entre si.

**SICLE - Sistema de Información y Control Local de Estación**
Computadoras con mayor funcionalidad que las UTRs las cuales tienen la capacidad de administrar localmente una subestación o planta generadora en forma local, así como la función principal de reportar datos a una o varias UTMs.

**RUTEADOR**
Equipo encargado de conectar redes LAN o WAN para optimizar la velocidad de transferencia de información y reducir las posibles colisiones de mensajes por protocolo TPC/IP. Tiene la capacidad de manejar puertos de alta velocidad.

**FIREWALL**
Equipo ruteador encargado de manejar permisos de acceso a las redes que se encuentran dentro de su esquema de protección por medio de listas de acceso.

**SWITCH**
Equipo encargado de conectar equipos de cómputo con administración de mensajes que evita colisiones.

**HUB**
Equipo encargado de conectar equipos de cómputo. Actualmente han sido reemplazados por los SWITCH debido a que los HUB generan colisiones en la red.

**LAN – Local área Network**
Red de computadoras de área Local. Termino normalmente utilizado en redes corporativas o de oficina que están en una misma región.

**WAN – Wide área Network**
Red de computadoras de área Amplia. Termino normalmente utilizado en redes corporativas que están compuestas de redes conectadas en diferentes regiones.

**CRT – Cathode Ray Tube**
Termino utilizado para nombrar a los monitores de computadora por su principio de funcionamiento.

**PCP – Programable Comunication Port**
Puerto programable de comunicaciones del Sistema SCADA

**RCI – Rack Communication Interfase**
Gabinete de Interfases de Comunicaciones del Sistema SCADA

**ICP – Intelligent Communication Port**
Puerto de comunicaciones Inteligente del Sistema SCADA

**MODEM**
Tarjeta moduladora-demoduladora para establecer la comunicación por diferentes medios en el Sistema SCADA.

**CANAL DE COMUNICACIÓN**
Medio físico por el cual se envían y reciben datos de las UTRs desde los puntos remotos donde se encuentran.

**RAS**
Servidor de Aplicaciones.

**RDAS**
Servidor de Adquisición de Datos.

**MMI**
Servidor de Interfase Hombre – Maquina.

**DTS**
Servidor de Entrenamiento de Operadores.

**HIS**
Servidor Histórico.

**WS**
Estación de Trabajo.

**PC**
Computadora Personal.

**OPLAT**
Medio de comunicaciones por líneas de transmisión de energía eléctrica Onda Portadora por Líneas Aéreas de Transmisión.

**MICROONDAS**
Medio de comunicación aéreo basado en señales que viajan en forma ondas.

**FIBRA OPTICA**
Medio de comunicación por cable de fibra que permite grandes anchos de banda y altas velocidades.

**UNIX**
Sistema Operativo Multi-tareas que soporta la ejecución de varios procesos en un mismo tiempo.

**WINDOWS**
Sistema Operativo que permite trabajar de forma sencilla con procesos de UNIX debido a sus manejadores de ventanas.

**BELL 202**
Estándar de comunicación por MODEM.

**V.35**
Estándar de comunicación por MODEM.

**ASCII – American Standard Code for Information Interchange**
Estándar utilizado para transferencia e intercambio de información, sus siglas significan Código Americano Estándar para el Intercambio de información

**SQL – Structured Query Language**
Lenguaje estándar de consulta de tablas de bases de datos de diferentes tipos como Oracle, Access, Informix, etc.

**SCSI – Small Computers Systems Interface**
Es un conjunto de estándares ANSI para la conexión de dispositivos a sistemas computacionales. La mayoría de estos dispositivos son de almacenamiento masivo.

**EISA – Extended Industry Standard Architecture**
Estándar de interconexión de computadoras que aumenta la capacidad de la interfase ISA de 16 bit a 32 bit.

**PCIMG - PCI Industrial Computer Manufacturers Group**
Estándar para la conexión de tarjetas ISA o PCI en ranuras de CPU.

**DNP3.0**
Protocolo estándar de comunicación normalmente utilizado para enlazar UTRs y UTMs.

**HARRIS 5000**
Protocolo estándar de comunicación normalmente utilizado para enlazar UTRs y UTMs.

**X.25**
Protocolo de comunicaciones basado en la tecnología de conmutación de paquetes, comúnmente utilizado para transferencia de información de alta velocidad.

**ICCP – Inter-Control Center Protocol**
Protocolo de Comunicaciones entre Centros de Control.
**CCR – Computer to Computer Remote**
Protocolo de comunicaciones de computadora a computadora que permite el enlace de centros de control por ICCP.

**TCP/IP – Transmission Control Protocol / Internet Protocol**
Protocolo de comunicaciones con control de transferencia, tradicionalmente utilizado para conectar grandes redes de computadoras a alta velocidad.

**RTDB – Real Time Data Base**
Base de Datos de Tiempo Real.

**ODBC – Oracle Data Base Conection**
Conector de Bases de Datos Oracle.

**DBMS – Data Base Management Server**
Servidor de Administración de Base de Datos.

**LORUP - Limited On-Line Real Time Data Base Update**
Programa Limitado de Actualización de Base de Datos de Tiempo Real en Línea.

**IDB – Integrated Data Base**
Base de Datos Integrada, termino utilizado para llamar a la base de datos compuesta por puntos calculados o del sistema.

**DBEDIT – Data Base Editor**
Interfase de Edición de Base de Datos del Sistema basada en Microsoft Access.

**DBG - Data Base Generación**
Subsistema que maneja los procesos de generación de Base de Datos.

**DBL – Data Base Language**
Lenguaje de Edición de Base de Datos.

**DBGSYS**
Programa que genera la Interfase de generación de Base de Datos.

**DISGEN – Display Generator**
Programa para Generar Desplegados del Sistema.

**GUI – Graphical User Interfase**
Interfase grafica de Usuario.

**MMC – Man-Machine Console**
Consola Hombre-Maquina.