

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA ELÉCTRICA



TEMA

**DISEÑO Y EXPERIMENTACIÓN DE UN SISTEMA DE MEDICIÓN REMOTA DE
VARIABLES ELÉCTRICAS**

**TRABAJO DE GRADUACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO ELECTRICISTA**

PRESENTAN

**GERARDO RENE VIZCARRA MORENO
SAMUEL ALEXANDER LUCERO CULI**



ASESOR

ERICK ALEXANDER BLANCO

MARZO 2009

SOYAPANGO - CIUDADELA DON BOSCO - EL SALVADOR

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA

AUTORIDADES

ING. FEDERICO MIGUEL HUGUET RIVERA
RECTOR

PBRO. VÍCTOR BERMUDEZ YÁNEZ
VICE-RECTOR ACADÉMICO

INGA. XIOMARA MARTÍNEZ
SECRETARIO GENERAL

ING. ERNESTO GODOFREDO GIRÓN
DECANO FACULTAD DE INGENIERÍA

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERÍA

TRIBUNAL EXAMINADOR



ING. HÉCTOR CARIÁS
LECTOR



ING. ERICK BLANCO
ASESOR

INDICE

Páginas

INTRODUCCIÓN

1) Capítulo I Generalidades	
1.1 Antecedentes	1
1.2 Proyección de Sistemas de Medición Remota.....	2
1.3 Objetivos.....	7
1.3.1 Objetivo General	7
1.3.2 Objetivos Específicos	7
1.4 Justificación del Tema	8
1.5 Marco Teórico y Práctico	9
1.5.1 Alcances	10
1.5.2 Delimitaciones	10
2) Capítulo II Variables Eléctricas a Monitorear	
2.1 Generalidades.....	11
2.2 Variables Eléctricas.....	12
2.2.1 Tensión o Diferencia de Potencial.....	12
2.2.2 Corriente Eléctrica.....	13
2.2.3 Potencia Eléctrica.....	14
2.2.3.1 Potencia Aparente (S).....	15
2.2.3.2 Potencia Activa (P).....	16
2.2.3.3 Potencia Reactiva (Q).....	17
2.2.3.4 Factor de Potencia.....	18
2.3 Punto de Medición.....	24
2.4 Medición de Variables Eléctricas.....	24
3) Capítulo III Transmisión y Recepción de Datos del Medidor al Ordenador	
3.1 Generalidades.....	26
3.2 Medio de Transmisión y Recepción de Datos.....	27
3.3 Transmisión de Datos.....	30
3.4 Recepción de Datos.....	37

3.4.1 Recepción de Datos en el Medidor de Energía Circutor CVMk.....	38
3.4.2 Recepción de Datos en el Medidor de Energía VERIS H8437.....	39

4) Capítulo IV Descripción de Equipo Utilizado

4.0 Características Técnicas.....	42
4.1 Analizador de Redes Circutor CVMk.....	42
4.1.1 Descripción de Analizador de Redes CVMk.....	42
4.1.2 Modo de Operación.....	45
4.1.3 Módulos de Expansión CVMk.....	50
4.1.4 El Estándar RS 232.....	52
4.1.5 El Conector RS 485.....	54
4.1.6 Módulo Convertidor Inteligente (Accesorio de Comunicación).....	55
4.1.7 Esquema de Conexión del CVMk.....	59
4.1.8 Software PowerStudio.....	60
4.2 Punto de Acceso Utilizado para la Conexión a Internet.....	62
4.3 Tarjeta de Red Inalámbrica PCI D-Link.....	62
4.4 Hawkeye 8400V Series del grupo VERIS INDUSTRIES.....	63
4.4.1 Especificaciones Generales.....	63
4.5 Presupuesto.....	66

5) Capítulo V Características del Sistema Implementado

5.1 Sistema Eléctrico a Monitorear.....	68
5.2 INTERFAZ MEDIDOR DE ENERGÍA–ORDENADOR SERVIDOR.....	70
5.3 Software Utilizado Para El Procesamiento de Datos.....	71
5.3.1 Software PowerStudio para el Medidor CVMk.....	71
5.3.2 Software desarrollado en Matlab para el Medidor Veris H8437.....	76
5.4 Comunicación de Datos.....	84
5.4.1 Generalidades.....	84

5.4.2 Forma de Acceso Remoto.....	92
5.5 Diagrama de bloques del sistema.....	94
5.6 Estudio de Factibilidad de un sistema de medición en varios puntos.....	96
6) Conclusiones y Recomendaciones	
6.1 Conclusiones.....	100
6.2 Recomendaciones.....	102

Anexos

Anexo A Fotografías en Cuarto de Subestación #2 del CITT

A1. Medidor	
Circutor CVMk empotrado en gabinete.....	106
A2. Medidor	
Circutor CVMk en gabinete visto desde atrás.....	106
A3. Vista de Barras dentro del panel.....	107
A4. Vista de los Transformadores	
De corriente (Donas) dentro del panel.....	107
A5. Vista del Convertidor RS232/RS485 i.....	108
A6. Vista del CPU Utilizado Como	
Servidor en el cual se puede Observar la	
Antena de la tarjeta de Red inalámbrica.....	108
A7. Subestación 501 KVA a la	
Cual el Medidor CVMk censa sus datos.....	109
A8. Conexión Y-Y de la	
Subestación #2 del CITT.....	109
A9. Vista de la distribución de	
Espacios en el cuarto de Subestación #2.....	110
A10. Cuarto de Subestación #2	
Frente al Edificio 5 del CITT.....	110

Anexos B Dispositivos Utilizados

B1. Tarjeta Inalámbrica D–Link instalada En ordenador ubicado en subestación #2.....	111
B2. Dispositivo DWL 3200 AP utilizado como Punto de acceso inalámbrico.....	111
B3. Características del convertidor RS485/RS232 CRSI.....	112
B4. Registro de las variables eléctricas en el CVMk.....	113
B5. Características del Medidor de Energía Veris H8437.....	114
B6. Registros de variables eléctricas en Medidor Veris H8437.....	115
B7. DDE - Intercambio Dinámico de Datos.....	115

Anexos C Ejecución del Programa

C1. Software Power Studio Versión 1.0.....	118
C1.1 Pantalla de error en Power Studio para el Medidor de energía CVMk.....	118
C1.2 Pantalla de Selección de Variables para ser Graficadas.....	119
C1.3 Pantalla que muestra el comportamiento de la Energía activa en kW-H para el día 10 de marzo de 2009.....	120
C1.4 Segunda opción de visualización de Pantalla principal.....	121
C1.5 Comportamiento de las tensiones de fase En la subestación #2 para el día 10 de Marzo de 2009.....	122
C1.6 Comportamiento de las tensiones de línea	

De la subestación #2 para el día 10 de Marzo de 2009.....	123
C1.7 Comportamiento de las corrientes de fase de la Subestación #2 para el día 10 de marzo de 2009.....	124
C1.8 Comportamiento del factor de potencia Promedio para el 10 de marzo de 2009.....	125
C1.9 Comportamiento de las potencias aparente y Activa trifásicas para el 10 de marzo de 2009.....	126
C1.10 Variación de la frecuencia el día 10 de marzo De 2009.....	127
C2. Programa en MATLAB	
C2.1. Base de datos obtenida con el software LABVIEW, A partir de la cual se importan los datos desde MATLAB.....	128
C2.2. Código Pantalla Inicial.....	129
C2.3. Código Pantalla principal.....	131
C2.4. Código Pantalla de tensiones.....	138
C2.5. Código Pantalla de Potencia.....	144
Anexos D Artículo de Tesis.....	153
Glosario.....	159

INDICE DE FIGURAS

Páginas

1) Capítulo Generalidades	
Figura 1.1 Esquema general de un sistema AMR.....	4
Figura 1.2 Bloques funcionales de un medidor de energía con interface serial.....	5
Figura 1.3 Esquema típico de un sistema de medición remoto de energía.....	9
2) Capítulo II Variables Eléctricas a Monitorear	
Figura 2.1 Diagrama unifilar de Subestación # 2.....	12
Figura 2.2 Relación entre potencia activa, aparente y reactiva. Triángulo de potencias.....	15
3) Capítulo III Transmisión y Recepción de Datos del Medidor al Ordenador.	
Figura 3.1 Medidor de Energía Circutor CVMk.....	26
Figura 3.2 Medidor de Energía Veris H8400 Series.....	27
Figura 3.3. Esquema de los dispositivos involucrados en la transmisión y recepción de datos en el sistema Medidor-Ordenador.....	27
Figura 3.4 Pines de las interfaces RS-485 y RS-232 con conector DB9.....	28
Figura 3.5 Conexión interna de la Interfaz RS-232 y de sus conectores DB9.....	29
Figura 3.6. Conexión de terminales del Medidor Veris H8437 al conector DB9 de la interfaz RS-485.....	30
Figura 3.7 Código gráfico de captura de datos del programa en Labview.....	40
4) Capítulo IV Descripción de Equipo Utilizado	
Figura 4.1 Modo de conexión del dispositivo CVMk.....	44
Figura 4.2 Módulo de Expansión.....	51
Figura 4.3 Módulo de Expansión conectado a CVMk.....	51
Figura 4.4 Conector DB9 macho y hembra.....	53
Figura 4.5 Carátula del Convertidor Inteligente RS-232/RS-485.....	56
Figura 4.6 Microswitch del Convertidor Inteligente RS-232/RS-485.....	56
Figura 4.7 Convertidor RS-232/RS-485 y Convertidor Inteligente RS-232/RS-485i.....	57
Figura 4.8 Bornera de conexiones del analizador de redes Circutor.....	58

Figura 4.9 Esquema de conexión a TC's del analizador de redes Circutor.....	59
Figura 4.10 Hawkeye Series del Grupo Veris Industries.....	63

5) Capítulo V Características del Sistema Implementado

Figura 5.1 Lista de dispositivos compatibles con el software PowerStudio versión 1.0.....	72
Figura 5.2 Asignación de nombre y descripción del medidor seleccionado.....	72
Figura 5.3 Selección de la conexión que se usa para la comunicación de datos.....	73
Figura 5.4 Asignación del número de periférico.....	73
Figura 5.5 Pantalla principal en PowerStudio para el medidor de energía CVMk	74
Figura 5.6 Pantalla de Inicio de Aplicación en Matlab.....	77
Figura 5.7 Pantalla Principal.....	79
Figura 5.8 Pantalla de Tensiones.....	81
Figura 5.9 Pantalla de Potencias y Factor de Potencias.....	83
Figura 5.10 Ubicación de los edificios involucrados y el medio de comunicación utilizado.....	92
Figura 5.11 Acceso al escritorio remoto de la computadora con dirección IP 10.0.19.10.....	93
Figura 5.12 Diagrama de bloques del sistema.....	94
Figura 5.13 Dispositivos agregados a PowerStudio como software Centralizado.....	98

INDICE DE TABLAS

Páginas

Capítulo IV Descripción de Equipo Utilizado

Tabla 4.1 Características medidor CVMk.....	43
Tabla 4.2 Características Técnicas medidores serie CVM.....	44
Tabla 4.3 Parámetros que indican los LED	46
Tabla 4.4 Compatibilidad Módulo de Expansión CVMk.....	52
Tabla 4.5 Pines y su función de un DB9 o puerto serie.....	53
Tabla 4.6 Pines y su descripción de un DB9 o puerto serie.....	54
Tabla 4.7 Comparación características PowerStudio y CIRNET.....	61
Tabla 4.8. Costos de la Aplicación.....	66

Capítulo V. Características del Sistema Implementado

Tabla 5.1 Costos de Inversión.....	97
------------------------------------	----

Anexos

Anexos B Dispositivos Utilizados

B3. Características del convertidor RS485/RS232 CRSI.....	110
B4. Registro de las variables eléctricas en el CVMk.....	111
B5. Características del Medidor de Energía Veris H8437.....	112
B6. Registros de variables eléctricas en medidor Veris H8437.....	113

Anexos C Ejecución del Programa

C2.1. Base de datos obtenida con el software LABVIEW, a partir de la cual se importan los datos desde MATLAB.....	126
--	-----

INTRODUCCIÓN.

El consumo de energía eléctrica se ha incrementado cada año, debido al crecimiento poblacional y de empresas. Las empresas de distribución de energía eléctrica y las empresas consumidoras de este servicio, necesitan determinar el consumo eléctrico con el fin de realizar una facturación, por parte de la distribuidora, y monitorear el consumo eléctrico, por parte de la empresa consumidora; y de esta manera tener controlado el consumo de energía eléctrica y poder planificar en base a estudios de una determinada red de energía.

Una manera de hacer práctica la lectura de los medidores de energía es, mediante la medición remota de variables eléctricas, tales como potencia, energía, tensión, corriente, factor de potencia y distorsión armónica.

Este trabajo está enfocado a la medición remota de variables eléctricas en una subestación de la universidad Don Bosco, para lo cual se hace uso de equipo de medición que la universidad posee actualmente pero que no se está utilizando como debe ser, para lograr un monitoreo remoto del consumo de energía y de otros parámetros eléctricos importantes mencionados anteriormente.

CAPÍTULO I. GENERALIDADES

1.1 ANTECEDENTES

El medidor de energía ha pasado de ser un dispositivo de uso limitado a uno de importancia crítica dentro de la cadena de servicios. Este cambio se dio rápida y dramáticamente debido a varios factores. Entre estos, se encuentran la desregulación y competencia, la privatización, los avances tecnológicos y un enfoque incrementado en la necesidad de datos precisos y confiables a través del proceso de la prestación de los servicios en el sector eléctrico.

Es conocido que los países industrializados están migrando a sistemas de medición remota para tener mejor control sobre lo que se debe cobrar por el servicio prestado, las compañías de suministro de electricidad instalan en cada una de las residencias o puntos de facturación un medidor para este fin. Con una periodicidad no menor a un mes, personal autorizado realiza de forma manual y localmente la lectura en el medidor del servicio y anota el consumo correspondiente. A medida que fue creciendo la cantidad de clientes, se incrementó la demanda y por ende se invierte más en mano de obra y la frecuencia de medición es cada vez más irregular; como consecuencia propia de la operación manual se producen errores en perjuicio de los clientes que tienen que pagar mayor tarifa, y de las compañías que tienen que invertir recursos en el procesamiento de los reclamos. Este sistema tradicional de medición tiene el inconveniente de ser no muy eficiente porque es hecho por personal diferente para cada servicio y en el caso de hogares donde el medidor está dentro de la propiedad demanda la presencia de personas permanentemente en el hogar.

Actualmente en la universidad Don Bosco se utiliza este mismo método de revisión del medidor para posteriormente enviarles su factura con lo correspondiente al mes que se está cobrando. La Universidad en este momento cuenta con medidores de la marca Circutor modelo CVMk instalados en las

subestaciones, de los cuales, solo uno cuenta con un puerto serie de recolección de datos el cual utilizaremos para nuestro trabajo de graduación. Y se desarrolla una aplicación en Matlab para utilizar con la base de datos generada por el programa Labview a partir de los datos obtenidos de un Medidor Veris 8437 con el cual ya se realizaron pruebas en la tesis “Análisis técnico-económico del comportamiento de variables eléctricas que inciden en la facturación y consumo de energía eléctrica en el centro de investigación y transferencia de tecnología (CITT)”.

1.2 PROYECCIÓN DE SISTEMAS DE MEDICIÓN REMOTA

En épocas recientes se han desarrollado tres áreas que han permitido implementar nuevas técnicas de medición y facturación más eficientes, confiables y económicas: la primera relacionada con el desarrollo de la electrónica que conlleva a la aparición de equipos de medición más precisos y económicos denominados comúnmente "Instrumentación inteligente", la segunda involucra los avances en los sistemas de comunicaciones y las redes públicas que brindan diversas formas de conectarse remotamente y en forma bidireccional, los dispositivos inteligentes instalados en cada residencia y los centros de control o facturación, y, finalmente, la última que tiene que ver con el desarrollo de sistemas basados en hardware normalizado de fácil adquisición en el mercado y de software especializado en este tipo de aplicaciones . La integración de éstos permite la implementación de soluciones que, con base a la información obtenida de consumo generan una factura "libre de errores". Estos tres elementos componen los sistemas de lectura remota conocidos por sus siglas en inglés AMR (*Automatic Meter Reading*).

Los sistemas de medición remota están compuestos por los siguientes elementos: Instrumentación de campo o medidores, sistema de comunicación y sistema de recolección y procesamiento de la información.

La instrumentación o dispositivos de medición, son dispositivos electrónicos inteligentes que constan principalmente de un sensor de medición y un transductor. Estos elementos también poseen integrado un elemento de

comunicación que les permite integrarse en una red de medidores a un sistema de adquisición centralizado, el cual les permite recibir comandos y enviar datos por solicitud. Una de las últimas innovaciones en este campo es el desarrollo de "chips" que permiten la comunicación de esta instrumentación vía Internet con el centro de control sin la necesidad de utilizar una red LAN o una computadora con módem conectado a una línea telefónica. La gran capacidad de estos equipos de medición permite la adquisición y envío de otras variables que pueden ayudar a mejorar la calidad del servicio prestado. El desarrollo tecnológico alcanzado por las compañías dedicadas al suministro de esta instrumentación ha crecido notablemente en los últimos tiempos, permitiendo una reducción en los costos en la fabricación de los mismos y haciendo atractiva la implementación de esta tipo de medición.

El desarrollo de la instrumentación inteligente permite tener disponibles en el mercado dispositivos capaces de recolectar y enviar varios datos relacionados con variables relativas al consumo eléctrico y sobre la calidad del servicio. También las capacidades de la instrumentación inteligente pueden aprovecharse para ejecutar programas de mantenimiento.

La comunicación o integración de la instrumentación inteligente con el centro de control o facturación se hace de diversos modos y utilizando diversos medios. El desarrollo de las redes de comunicación ha hecho posible que se implementen diversas formas de comunicación aprovechando las diversas plataformas públicas existentes. La comunicación entre la medición y su centro de control, puede hacerse a través de medios diversos tales como cable coaxial, cable de par trenzado UTP o STP, fibra óptica, utilizando telefonía tradicional, o de manera inalámbrica por medio de la telefonía celular o Internet.

Los sistemas de recolección y de procesamiento de datos están ubicados en las unidades de cobro o facturación de la compañía suministradora del servicio. Estos sistemas de procesamiento se componen de equipos basados en computadoras personales o estaciones de trabajo, módem u otro hardware de comunicación

como equipo de enlace y software desarrollado para este tipo de aplicaciones. Esta plataforma permite recolectar y procesar la información suministrada por cada medidor y procesarla. Con base a la información recolectada y procesada, el sistema es capaz de generar de forma automática la facturación del servicio. Estos sistemas permiten además, presentar una sola factura. Una compañía será capaz, mediante el uso de estos sistemas, emitir una factura única por servicio, de esta forma haciendo más eficiente el cobro y procesamiento de reclamos por parte del cliente. Por lo general las empresas de servicios eligen en primera instancia puntos de medición de tipo industrial y de tipo comercial para iniciar con sistemas AMR. A los medidores de consumo actual se añade un MODEM que se encarga de “entender” el protocolo del medidor y transcribir esta información a un formato que permita su transmisión por varios medios de comunicación como son: sistemas RF, redes de telefonía pública conmutadas PSTN ó redes de datos celular GPRS, aunque también depende de la tecnología del medidor de energía. Estos MODEMS se conectan por los medios de comunicación mencionados anteriormente a una base de datos y sistema de control centrales, donde se almacena la información de cada medidor.

Control de medidores por AMR

La técnica de lectura remota de medidores de energía AMR, se está aplicando con éxito en varios países donde es necesario contar con un control dinámico y exacto del consumo eléctrico, obteniendo perfiles y registros del consumo actualizados.

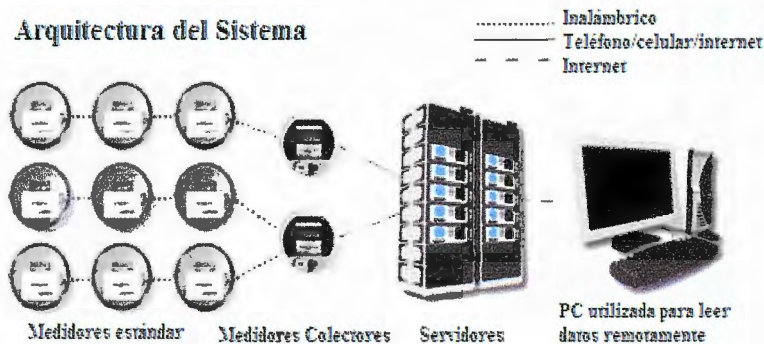


Figura 1.1 Esquema general de un sistema AMR¹.

¹ Telelectura de medidores de energía con tecnologías AMR y AMI, www.afinidadelectrica.com

Los primeros beneficiarios de esta técnica fueron los grandes consumidores de energía. La solución basada originalmente en una línea fija y un módem telefónico, fue reemplazada por un módem inalámbrico o también conocido como *Wireless Module*.



Figura 1.2. Bloques funcionales de un medidor de energía con interface serial².

Las técnicas de lectura remota se dividen en varios tipos, con características y funcionalidades diferentes. Con base a su complejidad tenemos las siguientes familias de medidores:

AMR (Automatic Meter Reading): Contempla comunicación bidireccional entre el medidor y el servidor de datos. Registra la medición de energía mensual o por intervalos de tiempo predefinidos (por lo general 15 o 30 minutos.).

AMI (Advanced Meter Infrastructure): Adiciona la lectura del consumo “on-demand”, opciones de precios diferenciados por punto medido y registros de la demanda, como también la programación de intervalos de “carga” previamente acordados con cada cliente.

SMART METERS: Incorpora la opción de conexión y/o desconexión remota del suministro al cliente, además de informar los parámetros respecto de la calidad de la energía como tensión, corriente, factor de potencia y frecuencia. Contempla la

² Telectura de medidores de energía con tecnologías AMR y AMI, www.afinidadelectrica.com

programación remota desde el centro de gestión y la actualización remota del software de medición.

En el caso particular de este trabajo de investigación se utiliza la técnica de lectura remota *AMR*. Aunque los datos pueden leerse en tiempo real en el software del medidor, el registro o almacenamiento de estos datos se efectúa en periodos de 5 minutos.

1.3 OBJETIVOS.

1.3.1 Objetivo General.

“Medir remotamente las principales variables eléctricas proporcionadas por un medidor de energía con el fin de monitorear el comportamiento de éstos parámetros”.

1.3.2 Objetivos Específicos

- ✓ Mediante cableado UTP, tarjetas de red inalámbricas y puntos de acceso inalámbrico, obtener remotamente datos proporcionados por un medidor de energía, utilizando la red de datos de la universidad.
- ✓ Almacenar en un ordenador, mediante un determinado software, los datos proporcionados por el medidor de energía.
- ✓ Diseñar un sistema de envío y recepción de datos desde el lugar donde se instale el medidor de energía hasta el lugar donde estará el ordenador para leer los datos de manera remota.
- ✓ Elaborar un artículo que refleje lo más relevante del tema de investigación.
- ✓ Estimar presupuesto para la implementación del sistema de medición remota que será diseñado.
- ✓ Establecer factibilidad de implementar un sistema de medición remota en un solo punto de medición y realizar una serie de pruebas de funcionamiento al mismo.

1.4 JUSTIFICACIÓN DEL TEMA.

Un sistema como el que se pretende implementar, permitirá utilizar en su máxima capacidad la lectura de las variables eléctricas hechas por los medidores de energía, lo que conlleva a agilizar el proceso de determinación de consumo por un usuario en particular y a su pronta facturación haciendo de esta manera un sistema más eficaz.

Lo descrito en el párrafo anterior, desde el punto de vista de distribuidora de energía eléctrica, es de suma importancia, ya que a pesar que se requiere un costo inicial elevado para la instalación de miles de sistemas de medición remota, a largo plazo se tendría un gran ahorro, ya que evitaría la necesidad de utilizar recurso humano para leer mes a mes los miles de medidores que tenga instalados para cada usuario.

A partir de lo expuesto anteriormente se pretende implementar un sistema el cual mida en forma remota, es una respuesta a esta situación y ayuda a las empresas distribuidoras principalmente a aumentar sus ganancias económicas y poder estar a la vanguardia con la tecnología aplicada a este servicio.

Sin embargo, el prototipo que se quiere crear, no tendrá dispositivos de alta tecnología, pero se quiere mostrar el principio que se debe tomar en nuestro país en este rubro, tomando en cuenta que las empresas distribuidoras de energía eléctrica en El Salvador han venido presentando incrementos en el número de usuarios a los cuales proveen este servicio; incremento que se ve reflejado en la demanda de energía de nuestro país, razón por la cual, el sector eléctrico se ha visto en la necesidad de aumentar su capacidad instalada en las centrales generadoras de energía.

1.5 MARCO TEÓRICO Y PRÁCTICO.

El esquema general de medición que se está implementando alrededor del mundo, por las diferentes empresas distribuidoras de energía eléctrica, se ilustra en la figura 1.3 En este se distinguen tres bloques principales que, con variantes en sus denominaciones son:

- ✓ Instalaciones de medición
- ✓ Medios de Comunicación
- ✓ Centros de Control

En nuestro caso nos enfocaremos en los primeros dos. El punto de la medición está formado básicamente por medidores de energía eléctrica, concentradores de datos e interfases a los medios de comunicación. Los canales y protocolos de comunicación constituyen los medios de comunicación, sirven de enlace entre el centro de procesamiento de información y los dispositivos de medición de energía eléctrica.

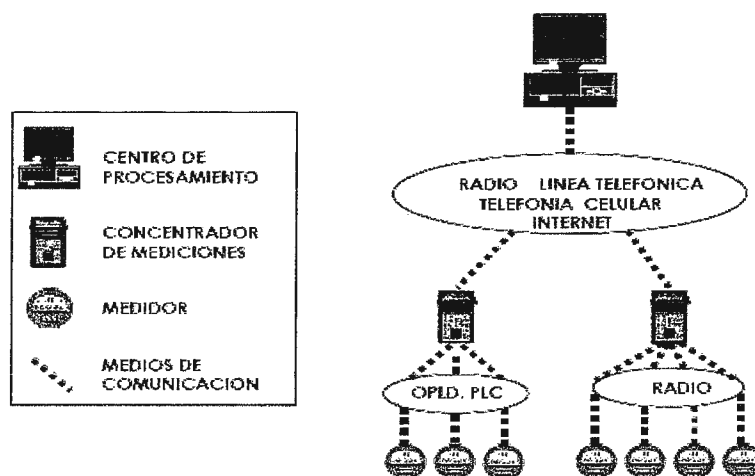


Figura 1.3 Esquema típico de un sistema de medición remoto de energía³.

³ Medición remota de energía, Universidad Tecnológica de la Mixteca.

1.5.1 ALCANCES

Medir remotamente variables eléctricas actualizadas proporcionadas por un medidor de energía eléctrica, estas variables serán: corrientes y tensiones de línea y de fase (dependiendo del sistema eléctrico), potencia activa, potencia reactiva, y factor de potencia promedio.

El medidor se encuentra instalado en la subestación eléctrica #2 del CITT de la Universidad Don Bosco, y para la transmisión de datos se usará como medio la red de datos de dicha universidad.

Los datos que proporcione el medidor en la subestación, podrán observarse desde la computadora de la persona encargada del mantenimiento eléctrico de dicha subestación, para poder analizar de una mejor manera las variables eléctricas.

1.5.2 DELIMITACIÓN

Por ser un prototipo, se utilizará solamente un medidor y el acceso al mismo se hará desde una determinada computadora.

El aparato de medición será propiedad de la Universidad Don Bosco, por lo que se tienen que adaptar ciertas características a la capacidad del mismo.

La instalación del sistema de medición remota se demostrará, solo mientras dure el periodo de prueba del sistema, es decisión de la universidad si desea colocarlo permanentemente por cuenta propia después de finalizado el trabajo de graduación, para lo cual sería instalado por personal de la misma universidad, tomado como base la experiencia adquirida en este trabajo.

CAPÍTULO II. VARIABLES ELÉCTRICAS A MONITOREAR.

2.1 GENERALIDADES.

Un punto de gran importancia que se debe considerar, es el tipo de conexión del sistema eléctrico existente en la universidad Don Bosco. De esto depende cada una de las variables eléctricas involucradas para la medición remota.

Puesto que el punto de medición será la subestación #2 (Costado Sur), de la universidad Don Bosco, el estudio se enfocará en dicha subestación, tanto en su tipo de conexión como el tipo de carga que maneja.

La subestación #2 alimenta los siguientes edificios del Centro de Investigación y Transferencia de Tecnología (CITT)⁴:

- Edificio #5. Comprende principalmente los departamentos de Metrología, Proyectos Educativos, academia CISCO y la parte de capacitaciones.
- Edificio #6 comprende el taller de mecánica y los centros de cómputo.
- Edificio #7 este con excepción de los demás es alimentado por dicha subestación y por una subestación monofásica aérea de 100 KVA.

Esta subestación cuenta con una planta de emergencia de 205 kW la cual sirve como respaldo para algunas cargas de estos edificios, la planta se encuentra conectada en paralelo con la línea mediante un interruptor de transferencia el cuál mediante un controlador lógico programable entra en operación cuando hay cortes en el servicio de energía eléctrica proporcionado por CAESS.

⁴ Trabajo de graduación: "Análisis técnico-económico del comportamiento de variables eléctricas que inciden en la facturación y consumo de energía eléctrica en el centro de investigaciones y transferencia de tecnología (CITT)

En la figura 2.1 se muestra el diagrama unifilar correspondiente a la subestación #2, en el cual se muestra dicha subestación y la barra donde sale la alimentación para los tableros principales de cada uno de los edificios de la universidad a los cuales suministra energía eléctrica esta subestación. Además, se muestra el grupo electrógeno de 256 kVA y 205kW conectado a esa barra que sirve para alimentar principalmente a los edificios 5 y 6 en los departamentos de metrología y computación respectivamente.

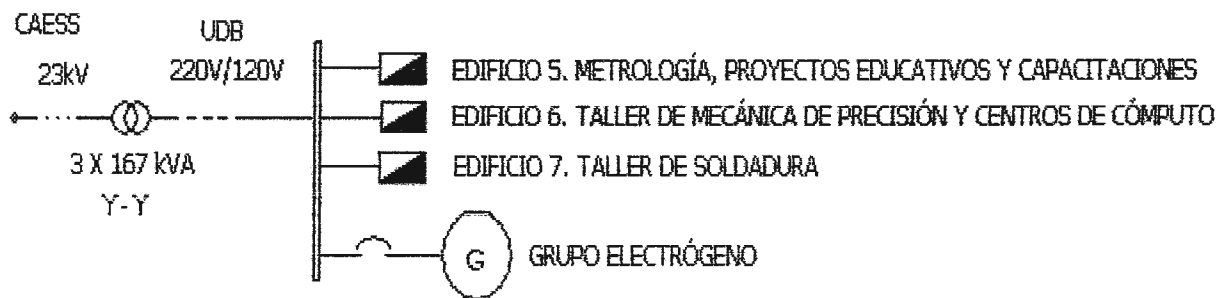


Figura 2.1. Diagrama unifilar de Subestación # 2.

Es importante conocer el tipo de conexión y también el tipo de cargas conectadas a la red para saber si lo que se está midiendo en términos generales está acorde a lo que se espera.

2.2 VARIABLES ELÉCTRICAS

En este apartado se hace una breve descripción de cada una de las variables eléctricas que se toman en cuenta para mediciones de sistemas de potencia. Además, se justifica la importancia de conocer los valores de cada una de ellas en un determinado momento.

2.2.1 TENSIÓN O DIFERENCIA DE POTENCIAL

La tensión o diferencia de potencial es la presión que ejerce una fuente de suministro de energía eléctrica o fuerza electromotriz (FEM) sobre las cargas

eléctricas o electrones en un circuito eléctrico cerrado, para que se establezca el flujo de una corriente eléctrica.

A mayor diferencia de potencial o presión que ejerza una fuente de FEM sobre las cargas eléctricas o electrones contenidos en un conductor, mayor será la tensión existente en el circuito al que corresponda ese conductor.

Esta magnitud eléctrica es de gran importancia monitorearla, porque si no se está recibiendo la tensión adecuada por parte de la compañía distribuidora de energía eléctrica, pueden darse dos situaciones:

- a. Que la tensión sea mayor a la nominal, en un porcentaje establecido por la ley. Lo que producirá daños a equipos sensibles a estas variaciones, pudiéndose dañar permanentemente dichos dispositivos conectados a la red de la universidad.
- b. Que la tensión sea menor a la nominal. Lo cual baja la eficiencia de los equipos y dispositivos conectados a la red eléctrica de la institución disminuyendo de una manera considerable la vida útil de éstos.

De esta manera se justifica claramente la importancia de mantener monitoreados los valores de tensiones en la subestación eléctrica.

2.2.2 CORRIENTE ELÉCTRICA

Es el flujo de carga por unidad de tiempo que recorre un material. Se debe a un movimiento de los electrones por el interior del material. Se mide en amperios y se indica con el símbolo A. Una corriente eléctrica, puesto que se trata de un movimiento de cargas, produce un campo magnético.

En este caso particular la corriente eléctrica se transporta por medio de conductores de cobre hasta donde se encuentran ubicadas las cargas.

Al decir que la corriente eléctrica será transportada por conductores de cobre, es de mencionar que se debe monitorear la corriente eléctrica porque cada uno de esos conductores de cobre que transportan la corriente eléctrica hacia los centros de carga, poseen una resistividad eléctrica propia de cada material ($\rho_{Cu} = 1.67 \times 10^{-8} \Omega \cdot m$ a $20^\circ C$) del que esté hecho el conductor eléctrico.

Para cada centro de carga se ha diseñado un conductor eléctrico bajo ciertas condiciones de operación. Si a un conductor eléctrico, con una sección transversal o calibre definido, se le hace circular a través de él una corriente por encima de la capacidad de corriente para el cual ha sido diseñado; se producirá un calentamiento en este conductor que provocará que se den pérdidas por efecto Joule ($P = I R^2$ [Watts]) incurriendo en un gasto mayor en el consumo de energía eléctrica traducido a un mayor costo económico.

En el peor de los casos pueden generarse fallas de cortocircuito en el sistema eléctrico producto de la fundición de un determinado conductor que trabajó a valores de corriente mucho más altos en comparación con el valor de corriente para el cual fue diseñado ese conductor.

Justificando de esta manera la importancia de monitorear esta magnitud eléctrica, recordando que en las redes eléctricas siempre se presentan aumentos en la carga, y si no se sabe hasta que valores de corriente soporta el conductor, se dan fallas en el sistema, afectando en este caso a dispositivos o equipos conectados a la red donde ha ocurrido la falla y la red de conductores misma, si las protecciones no se han dimensionado y coordinado correctamente.

2.2.3 Potencia eléctrica.

Haciendo un enfoque de lo que es la potencia eléctrica y como afecta en una red eléctrica desde el punto de vista de la corriente alterna, se conoce que la potencia eléctrica alterna posee ciertos componentes que se relacionan entre sí mediante el triángulo de potencias (figura 2.2) y cada una de ellas repercute en el buen rendimiento de una red eléctrica de potencia.

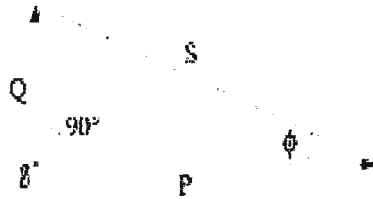


Figura 2.2.- Relación entre potencia activa, aparente y reactiva. Triángulo de potencias.

Las variables eléctricas que se relacionan directamente en el triángulo de potencia son las siguientes:

- Potencia aparente (S)
- Potencia Activa o Real (P)
- Potencia Reactiva (Q)
- Factor de potencia (Cos ϕ o f.p.)

Si bien, como se dijo anteriormente, cada una de ellas están estrechamente relacionadas, afectan de manera diferente a una determinada red eléctrica. A continuación se describen cada una de estas variables eléctricas y se expone de una forma breve la importancia del monitoreo para su correcto control.

2.2.3.1 Potencia aparente (S)

La potencia aparente (magnitud de la potencia compleja) de un circuito eléctrico de corriente alterna es la suma (vectorial) de la energía que disipa dicho circuito en cierto tiempo en forma de calor o trabajo y la energía utilizada para la formación de los campos eléctricos y magnéticos de sus componentes que fluctuará entre estos componentes y la fuente de energía.

Esta potencia no es la realmente consumida o "útil", salvo cuando el factor de potencia es la unidad ($\cos \phi = \text{f.p.} = 1$), pero en términos generales, como su nombre lo indica es la potencia que "aparentemente" se posee, se designa con la letra S y se mide en volt-amperios (VA).

Su fórmula es: $S = I \cdot V$

Es de gran importancia porque la capacidad de las subestaciones viene dada en voltio-amperios (VA), En el caso de la subestación #2 de la universidad Don Bosco es de 501 kVA, la cual esta compuesta por tres transformadores monofásicos de 167kVA cada uno.

La potencia aparente se relaciona con el factor de potencia, con la potencia activa y con la potencia reactiva, por lo tanto también está relacionada con el tipo y número de cargas eléctricas conectadas a la red eléctrica vista en el secundario de la subestación.

2.2.3.2 Potencia activa (P)

Es la potencia que representa la capacidad de un circuito para realizar un proceso de transformación de la energía eléctrica en trabajo. Los diferentes dispositivos eléctricos existentes convierten la energía eléctrica en otras formas de energía tales como: mecánica, lumínica, térmica, química, etc. Esta potencia es, por lo tanto, la realmente consumida por los circuitos. Cuando se habla de demanda eléctrica, es esta potencia la que se utiliza para determinar dicha demanda.

Se designa con la letra P y se mide en vatios (W). De acuerdo con su expresión, la ley de Ohm y el triángulo de impedancias:

$$P = I \cdot V \cdot \cos \phi = I \cdot Z \cdot I \cos \phi = I^2 \cdot Z \cdot \cos \phi = I^2 \cdot R$$

Resultado que indica que la potencia activa se debe a los elementos resistivos.

Esta potencia debe controlarse más que todo para incrementos que se vayan haciendo a la red eléctrica, porque con dicho incremento, aumenta la corriente eléctrica y por ende el calibre de conductor que debe utilizarse para un determinado valor de potencia activa, aspecto que si no se toma en cuenta, puede representar serios problemas al manejar altas corrientes con un calibre de conductor no diseñado para dicha capacidad de corriente elevada a partir del incremento en la carga.

Otro aspecto por el cual debe controlarse es para mantener lo más balanceado posible el sistema trifásico. Recordemos que cuando a un sistema trifásico balanceado se le agregan cargas monofásicas, éstas a veces no se reparten uniformemente para cada fase a menos que la cantidad de cargas sea múltiplo de tres y que éstas a la vez sean de las mismas características de operación, ya que si no se balancean las fases se pueden presentar problemas asociados con corriente elevadas en el conductor neutro.

Como se dijo con anterioridad, la potencia activa es la que consumen los elementos resistivos en un sistema eléctrico como por ejemplo cocinas, luminarias de filamentos (incandescentes), hornos tostadores, etc.

2.2.3.3 Potencia reactiva (Q)

Esta potencia se debe principalmente a efectos relacionados directamente con la existencia de bobinas o condensadores en los circuitos. La potencia reactiva tiene un valor medio nulo, por lo que no produce trabajo útil. Por ello que se dice que es una potencia que no produce vatios, se mide en volt-amperios reactivos (VAR) y se designa con la letra Q.

A partir de su expresión,

$$Q = I \cdot V \cdot \sin \phi = I \cdot Z \cdot I \sin \phi = I^2 \cdot Z \cdot \sin \phi$$
$$Q = I^2 \cdot X = I^2 \cdot (X_L - X_C) = S \cdot \sin \phi$$

Lo que reafirma que esta potencia es debida únicamente a los elementos reactivos.

La medición de la potencia reactiva es necesaria para tener un control de las cargas reactivas que se encuentren en un determinado sistema. Recordemos que si en algún momento se necesita hacer una compensación para mejorar el factor de potencia, se debe tomar la potencia reactiva actual en el sistema y compensarla con dispositivos reactivos comúnmente capacitores.

El tipo de carga reactiva que mayormente se tiene, tanto en la industria como en el caso de la universidad Don Bosco, es la potencia reactiva inductiva. Esto se ve reflejado en el tipo de cargas que en su mayoría son motores o transformadores.

Ejemplos:

- Motores de fresadoras
- Motores de tornos.
- Balastos de luminarias fluorescentes
- Aires acondicionados.
- Compresores.
- Transformadores de lámparas de haluro metálico
- Otras aplicaciones de motores.

Potencia trifásica

La representación matemática de la potencia activa en un sistema trifásico equilibrado está dada por la ecuación:

$$P_{3\phi} = \sqrt{3} \cdot I \cdot V \cdot \cos\Phi$$

Si es un sistema trifásico se tendrá potencia por fase del sistema y potencia trifásica. Lo anterior se cumple para cada uno de los tres tipos de potencia eléctrica: activa, reactiva y aparente.

2.2.3.4 Factor de potencia

Se define factor de potencia, FP, de un circuito de corriente alterna, como la relación entre la potencia activa, P, y la potencia aparente, S, o bien como el coseno del ángulo que forman los fasores de la intensidad de corriente y la tensión, designándose en este caso como $\cos\phi$, siendo ϕ el valor de dicho ángulo. De acuerdo con el triángulo de potencias de la figura 2.2:

$$FP = \cos \phi = \frac{P}{S}$$

El dispositivo utilizado para medir el FP se denomina cosímetro o medidor de ángulo de fase.

Importancia del factor de potencia

Para comprender la importancia del factor de potencia se van a considerar dos receptores con la misma potencia, 1000 W, conectados a la misma tensión de 230 V, pero el primero con un FP alto $\cos \varphi_1 = 0,96$ y el segundo con uno bajo $\cos \varphi_2 = 0,25$.

- Primer receptor

$$I_1 = \frac{P_1}{V \cos \varphi_1} = \frac{1000}{230 \cdot 0,96} \simeq 4,53 A$$

$$S_1 = VI_1 = 230 \cdot 4,53 \simeq 1.042 VA$$

- Segundo receptor

$$I_2 = \frac{P_2}{V \cos \varphi_2} = \frac{1000}{230 \cdot 0,25} \simeq 17,39 A$$

$$S_2 = VI_2 = 230 \cdot 17,39 \simeq 4.000 VA$$

Comparando ambos resultados, se obtienen las siguientes conclusiones:

- Un factor de potencia bajo comparado con otro alto, origina, para una misma potencia, una mayor demanda de corriente eléctrica, lo que implica la necesidad de utilizar cables de mayor sección.
- La potencia aparente es tanto mayor cuanto más bajo sea el factor de potencia, lo que origina una mayor dimensión de los generadores o transformadores relacionados a ese equipo.

Ambas conclusiones nos llevan a un mayor coste de la instalación alimentadora. Esto no resulta práctico para las compañías eléctricas, puesto que el gasto es mayor para un factor de potencia bajo. Es por ello que las compañías suministradoras penalizan la existencia de un factor de potencia bajo, obligando a su mejora o imponiendo costes adicionales.

Influencia del tipo de cargas

El valor del factor de potencia viene determinado por el tipo de cargas conectadas en una instalación. De acuerdo con su definición, el factor de potencia es adimensional y solamente puede tomar valores entre 0 y 1. En un circuito resistivo puro recorrido por una corriente alterna, la intensidad y la tensión están en fase ($\varphi=0$), esto es, cambian de polaridad en el mismo instante en cada ciclo, siendo por lo tanto el factor de potencia la unidad. Por otro lado, en un circuito reactivo puro, la intensidad y la tensión están en cuadratura ($\varphi=90^\circ$) siendo nulo el valor del FP.

En la práctica, los circuitos no pueden ser puramente resistivos ni reactivos, observándose desfases, más o menos significativos, entre las formas de onda de la corriente y la tensión. Así, si el FP está cercano a la unidad, se dirá que es un circuito altamente resistivo por lo que su FP es alto, mientras que si está cercano a cero que es fuertemente reactivo y su FP es bajo. Cuando el circuito sea de carácter inductivo, caso más común, se tendrá un FP en atraso, mientras que se dice en adelanto cuando lo es de carácter capacitivo.

Las cargas inductivas, tales como transformadores, motores de inducción y, en general, cualquier tipo de inductancia (tal como las que acompañan a las lámparas fluorescentes) generan potencia inductiva con la corriente atrasada respecto a la tensión.

Las cargas capacitivas, tales como bancos de condensadores o cables enterrados, generan potencia reactiva con la corriente adelantada con respecto a la tensión.

Mejora del factor de potencia

A menudo es posible ajustar el factor de potencia de un sistema a un valor muy próximo a la unidad. Esta práctica se conoce como mejora o corrección del factor de potencia y se realiza mediante la conexión de bancos de capacitores o condensadores a través de conmutadores generalmente automáticos. Por ejemplo, el efecto inductivo de las cargas de motores puede corregirse localmente mediante la conexión de condensadores. En determinadas ocasiones pueden instalarse motores síncronos con los que se puede inyectar potencia capacitiva o reactiva con tan solo variar la corriente de excitación del motor.

Las pérdidas de energía en las líneas de transporte de energía eléctrica aumentan con el incremento de la corriente. Como se ha comprobado, cuanto más bajo sea el FP de una carga, se requiere más corriente para conseguir la misma cantidad de energía útil. Por tanto, como ya se ha comentado, las compañías suministradoras de electricidad, para conseguir una mayor eficiencia de su red, requieren que los usuarios, especialmente aquellos que utilizan grandes potencias, mantengan los factores de potencia de sus respectivas cargas dentro de límites especificados, estando sujetos, de lo contrario, a pagos adicionales por energía reactiva.

La mejora del factor de potencia debe ser realizada de una forma cuidadosa con objeto de mantenerlo lo más alto posible. Es por ello que en los casos de grandes variaciones en la composición de la carga es preferible que la corrección se realice por medios automáticos.

En El Salvador existen penalizaciones o cargos a la facturación mensual de las empresas si éstas poseen un factor de potencia debajo de 0.9. Condición que es importante cumplir para no incurrir en mayores gastos en el consumo de energía eléctrica.

Las penalizaciones se desglosan a continuación⁵:

- 1) Si el FP es igual o mayor que 0.75 y menor que 0.90, el cargo por energía será aumentado en 1% por cada centésima que el FP sea inferior a 0.90;
- 2) Si el FP es igual o mayor que 0.60 y menor que 0.75, el cargo por energía será aumentado en 15% más el 2% por cada centésima que el FP sea inferior a 0.75; y,
- 3) Si el FP fuese inferior a 0.60, el Distribuidor, previa notificación, podrá suspender el suministro hasta tanto el usuario final adecue sus instalaciones a fin de superar dicho valor límite.

Con lo que se respalda la importancia de tener controlados bajo supervisión periódica las variables eléctricas expuestas anteriormente.

Componentes no senoidales

En los circuitos que tienen solamente corrientes y tensiones senoidales, el efecto del factor de potencia se presenta solamente como la diferencia en fase entre la corriente y el voltaje. Esto es menos conocido como "factor de potencia de desplazamiento". El concepto se puede generalizar a una distorsión total, o a un verdadero factor de potencia donde la potencia aparente incluye todos los componentes armónicos. Esto es de importancia en los sistemas de energía prácticos que contienen cargas no lineales tales como rectificadores, algunas formas de iluminación eléctrica, hornos de arco voltaico, equipos de soldadura y otros dispositivos como las computadoras.

Un ejemplo particularmente importante son los millones de computadores personales que típicamente incorporan fuentes de alimentación conmutadas con salidas cuyo rango de potencia va desde 150W hasta 500W. Históricamente, estas fuentes de alimentación de muy bajo costo incorporan un simple rectificador de onda completa que conduce sólo cuando la tensión instantánea excede la tensión de los capacitores de entrada. Esto conduce a relaciones entre las corrientes pico y promedio, lo que también lleva a una distorsión en el FP lo que origina un mayor consumo de energía eléctrica.

⁵ Términos y Condiciones Generales Agosto de 2008. Artículo 56. (SIGET)

Un multímetro típico dará resultados incorrectos cuando trata de medir la corriente AC que pasa por una carga que requiera corriente no-sinusoidal y luego calcule el FP. Debe usarse un multímetro con RMS verdadero para medir las corrientes y tensiones RMS reales (y por tanto la potencia aparente). Para medir la potencia real o la reactiva, debe usarse un vatímetro diseñado para trabajar adecuadamente con corrientes no sinusoidales.

En conclusión es importante conocer cada una de las variables consideradas importantes en el caso de una facturación, para nuestro caso la facturación que se realiza a la Universidad Don Bosco.

Por ejemplo es importante conocer la medida de tensión para garantizar que se está proporcionando el nivel de tensión que se ha contratado a la compañía distribuidora o que ésta oscila en un rango aceptable, de igual manera la corriente eléctrica que ayuda a determinar el consumo de potencia de toda la instalación, el factor de potencia el cual, garantiza que la energía se esté aprovechando de la mejor manera y no se sancione con una multa por tener un factor de potencia muy bajo; es por ello que para que la universidad pueda tener un historial de datos que garanticen que se le está cobrando lo justo, mes a mes, y para facilitar la verificación de dichos parámetros en tiempo real es el por qué de la importancia de nuestro trabajo de graduación.

Tradicionalmente algunos de los medidores comúnmente utilizados para medir todas las variables anteriormente explicadas son los siguientes:

Corriente → Amperímetro

Tensión → Voltímetro

Potencia → Vatímetro

Factor de Potencia → Cosímetro

En un medidor de energía sofisticado, como los que se utilizan en este estudio para llevar a cabo la medición remota, cada uno de los dispositivos antes mencionados se encuentran internamente incorporados. Esto quiere decir que este tipo de

medidor de energía proporciona: Potencia activa, reactiva, aparente y factor de potencia; corriente, tensión y frecuencia. Tanto para cada fase del sistema trifásico como las variables trifásicas en sí.

2.3 PUNTO DE MEDICIÓN

El punto de medición que se escoge para realizar las pruebas correspondientes a este estudio de medición remota, es la subestación atrás del edificio cinco del CITT (Subestación #2), en dicha subestación se encuentra la planta de emergencia y actualmente está conectado un medidor de la marca Circutor CVMk con puerto serie de comunicación el cual se utiliza únicamente para que personal se traslade a la subestación si desea obtener una medición de los parámetros eléctricos, es por ello, que este ha sido escogido como el punto donde se realizarán las conexiones apropiadas para poder realizar un enlace con la red de datos del Edificio 5 para transmitir los datos en tiempo real a través de la red de datos de la universidad, evitando así el desplazamiento de personal a la subestación, pudiendo realizar esta actividad de visualización del comportamiento de las variables eléctricas, desde su oficina. Dejando en todo momento el medidor y el servidor encargado de recibir y enviar los datos en un lugar seguro y considerando que se puede obtener una lectura aceptable de parámetros eléctricos.

El lugar se eligió porque el medidor de energía instalado allí, posee como se mencionó anteriormente, la tarjeta de expansión que posee un puerto serial con un Terminal DB9 macho que se utiliza para transmitir datos al servidor.

2.4 MEDICIÓN DE VARIABLES ELÉCTRICAS

En el caso de este trabajo se miden los siguientes parámetros eléctricos, tensión, corriente, potencias, factor de potencia y la frecuencia ya que son las magnitudes más comúnmente utilizadas para efectos de cobro por parte de las compañías distribuidoras.

Para llevar a cabo la medición se hace uso de transformadores de corriente para obtener la señal de corriente en el medidor de energía. Esto se hace porque los valores de corriente por fase que actualmente están en un rango entre 200 y 300 amperios, corriente que no puede manejar directamente el medidor de energía, por lo que se usan transformadores de corriente con una relación 1200:5, relación que se configura en el medidor y se detalla en el capítulo IV.

Para el caso de la señal de tensión, no es necesario utilizar transformador de potencial porque la señal se toma del secundario del transformador de la subestación trifásica. Valores de tensión que el medidor soporta.

El medidor Circutor CVMk proporciona más datos pero para efectos de nuestro trabajo utilizaremos las variables antes mencionadas. Este medidor se encargará de enviar los datos a través de un convertidor RS232 / RS485⁶ al CPU que se utilizará como receptor de los datos y en el cual todos los datos se almacenarán para posteriormente visualizarlos a través de la red desde cualquier punto dentro de la red con la que cuenta la Universidad.

Para este caso en particular se eligió como remoto de visualización de los datos, el ordenador del encargado del mantenimiento eléctrico y de subestaciones de dicha universidad (Edificio 4).

Si bien no se aborda en su totalidad en este estudio, porque se ha hecho antes en otros estudios⁷, pero existe otro modelo de medidor de energía que se puede utilizar y del cual la universidad también es propietaria. Este medidor es el Veris H8437 que tiene un funcionamiento bastante similar al medidor de energía Circutor modelo CVMk.

⁶ Ver capítulo 4

⁷ Trabajo de graduación: "Análisis técnico-económico del comportamiento de variables eléctricas que inciden en la facturación y consumo de energía eléctrica en el centro de investigaciones y transferencia de tecnología (citt)".

CAPÍTULO III. TRANSMISIÓN Y RECEPCIÓN DE DATOS DEL MEDIDOR AL ORDENADOR.

3.1 Generalidades

Este estudio sobre medición remota de variables eléctricas se ha enfocado en medidores de energía que son propiedad de la Universidad Don Bosco, en este caso particular se tendrán dos medidores los cuáles son el Circutor CVMk y el medidor de energía Veris H8437. Por ser este último único en su marca y modelo, y por existir ya estudios respecto a este medidor⁸, la aplicación más importante para el caso específico de la Universidad Don Bosco, recae en el medidor de energía Circutor CVMk.



Figura 3.1 Medidor de Energía Circutor CVMk

La universidad Don Bosco posee actualmente tres medidores CVMk y dos Veris H8437, estos últimos principalmente para usos didácticos. De los tres medidores marca Circutor, dos se encuentran actualmente instalados en cada una de las dos subestaciones del centro de investigación y transferencia de tecnología (CITT), mientras que el tercer medidor se instalará próximamente en la subestación del aula magna C de la universidad.

⁸ Consultar el trabajo de graduación: ANÁLISIS TÉCNICO-ECONÓMICO DEL COMPORTAMIENTO DE VARIABLES ELÉCTRICAS QUE INCIDEN EN LA FACTURACIÓN Y CONSUMO DE ENERGÍA ELÉCTRICA EN EL CENTRO DE INVESTIGACIONES Y TRANSFERENCIA DE TECNOLOGÍA (CITT), Capítulo III



Figura 3.2 Medidor de Energía Veris H8400 Series

3.2 Medio de transmisión y recepción de datos.

Tanto para el medidor Veris H8437 como para el Circutor CVMk el protocolo de comunicación es el protocolo Modbus, que utiliza una interfaz RS-485 a la salida del puerto de comunicación del medidor de energía. Esta interfaz RS-485 debe convertirse a una RS-232 para obtener datos desde el medidor vía el puerto serial hacia el ordenador que se utilizará para la obtención de datos del medidor.

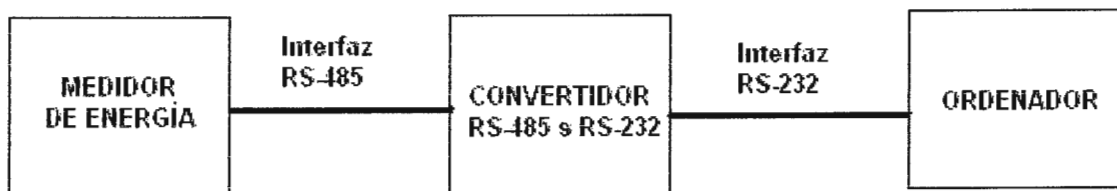


Figura 3.3. Esquema de los dispositivos involucrados en la transmisión y recepción de datos en el sistema Medidor-Ordenador.

Interfaces RS-485 y RS-232

La conexión RS-485 correspondiente al medidor de energía, y la interfaz RS-232 de la computadora, La conexión de estas interfaces se muestra en la figura 3.4, especificando además los pines involucrados en dicha conexión.

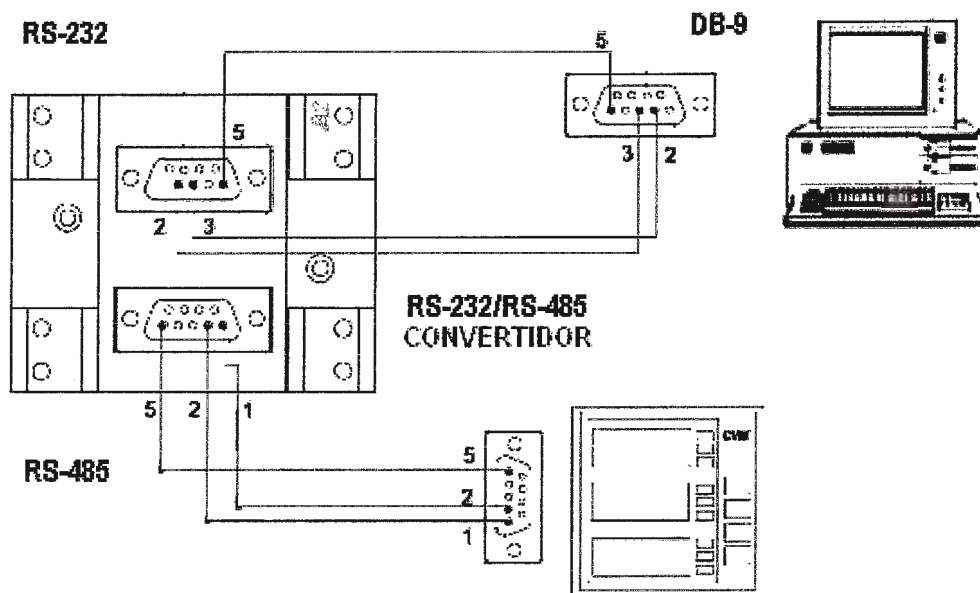


Figura 3.4 Pines de las interfaces RS-485 y RS-232 con conector DB9⁹.

Como se aprecia en la figura anterior, no se usan todos los pines del conector DB-9, y además se muestra la manera de conectar cada interfaz en el convertidor RS-485/RS-232¹⁰.

Cada una de las interfaces necesita de tres pines, los cuales se utilizan cada uno de ellos en transmisión de datos, recepción de datos, y el tercer pin es el de tierra como referencia (pin 5).

Ambos extremos de la interfaz RS-485 utilizan conectores DB-9 hembra, mientras que la interfaz RS-232 utiliza un conector DB-9 hembra en el extremo que se conecta con el ordenador, y un conector DB-9 macho en el extremo que se conecta con el convertidor.

Es fácil observar en la figura 3.4 que se utilizan solamente 3 pines en la interfaz RS-232, pero en realidad los demás pines llevan una conexión interna entre ellos

⁹ Manual del analizador de redes Circutor CVMk

¹⁰ Ver capítulo IV

para lograr una configuración que permita la comunicación del convertidor al ordenador.

La interconexión interna de la interfaz RS-232 se muestra en la figura 3.5

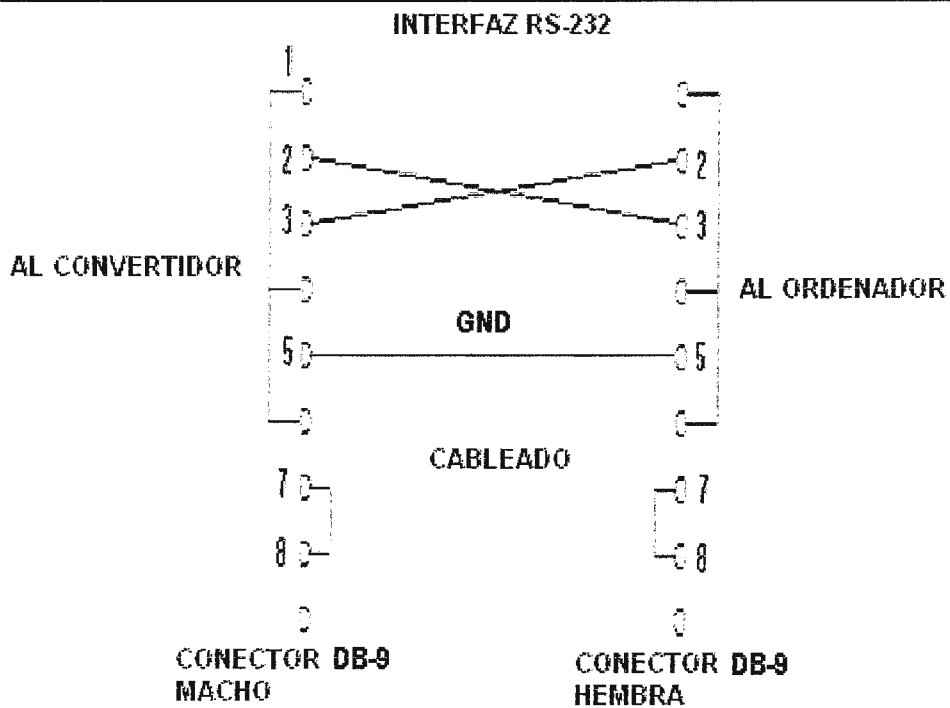


Figura 3.5 Conexión interna de la Interfaz RS-232 y de sus conectores DB-9.

En el caso de la interfaz RS-485, no existe ninguna conexión interna de sus pines, por lo que se utilizan solamente los tres pines que se aprecian en la figura 3.4.

3.3 Transmisión de Datos

Para la transmisión de datos, como se mencionó anteriormente, se utiliza el protocolo de comunicación Modbus, al cual corresponde una interfaz RS-485.

Para la comunicación Modbus se utilizan tres terminales:

+ Se utiliza para transmisión de datos.

- Se utiliza para recepción de datos y también para transmisión.

GND Es la referencia.

Para el medidor Circutor CVMk que se ha utilizado para llevar a cabo la medición remota, no es necesario hacer conexión alguna más que la que aparece en la figura 3.4. Esto gracias a que el medidor en cuestión posee en su tarjeta de comunicación un conector DB9 hembra, el cual posee conectados internamente en sus pines (1,2 y 5) los terminales mencionados anteriormente.

Por otro lado, para el medidor Veris H8437 se necesita hacer una conexión extra si se quiere utilizar el conector DB9 del convertidor, caso contrario se utilizan las entradas de conductores que posee el convertidor. Si se utiliza el conector DB9 del convertidor, la conexión necesaria se muestra en la figura 3.6.

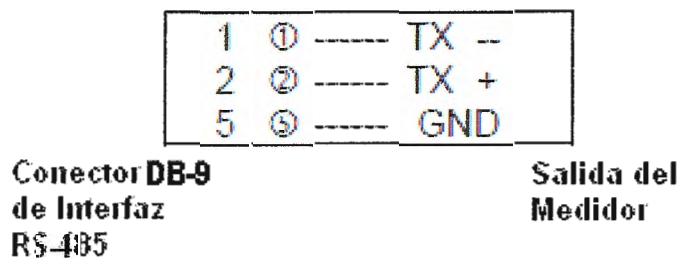


Figura 3.6. Conexión de terminales del Medidor Veris H8437 al conector DB9 de la interfaz RS-485

Como se ha mencionado anteriormente, en el caso del medidor marca Circutor en cuestión, la conexión de la figura 3.6 es interna en la respectiva tarjeta de comunicación.

Protocolo *Modbus*¹¹

Modbus/TCP se introdujo por *Schneider Automation* como una variante de la familia de protocolos MODBUS, ampliamente utilizada para la supervisión y el control de equipo de automatización. Específicamente el protocolo define el uso de mensajes MODBUS en un entorno intranet o Internet usando los protocolos TCP/IP. Y es el protocolo que utiliza el medidor Circutor CVMk ya que utiliza un convertidor RS232/RS485i para enviar y recibir los datos. Y el Medidor Veris utiliza Modbus/RTU en puerto serie RS485 y Ethernet ya que este se conecta directamente al servidor para enviar y recibir datos.

La especificación Modbus/TCP define un estándar interoperable en el campo de la automatización industrial, el cual es simple de implementar para cualquier dispositivo que soporte sockets TCP/IP. Todas las solicitudes son enviadas vía TCP sobre el puerto registrado 502 y normalmente usando comunicación half-duplex sobre una conexión dada. Es decir, no hay beneficio en enviar solicitudes adicionales sobre una conexión única mientras una respuesta está pendiente. Modbus/TCP básicamente encapsula una trama MODBUS dentro de una trama TCP en una manera simple.

Principales características:

a. Mecanismo de conexión:

MODBUS es un protocolo maestro/esclavo en el que cada solicitud del maestro es tratada de forma independiente por el esclavo, sin relación con las anteriores. Esto facilita proporcionar intercambios de datos resistentes a rupturas, requiriendo mínima información de recuperación para mantener una transacción en cualquiera de los dos terminales.

¹¹ Implementación de una red con Modbus/TCP, Universidad del Valle

De otro lado, las operaciones de programación esperan una comunicación orientada a la conexión, es decir, las máquinas de origen y de destino deben establecer un canal de comunicaciones antes de transferir datos. Este tipo de operaciones son implementadas de diferentes maneras por las diversas variantes de MODBUS (Modbus RTU, Modbus ASCII, Modbus PLUS).

En Modbus/TCP una conexión se establece inicialmente en la capa de aplicación y esta única conexión puede llevar múltiples transacciones independientes. Además, TCP permite establecer un gran número de conexiones concurrentes, de modo que el cliente (maestro) puede tanto re-usar una conexión previamente establecida como crear una nueva en el momento de requerir una transacción de datos.

En Modbus/TCP se usa el protocolo orientado a la conexión TCP en lugar del protocolo orientado a datagramas UDP. Se busca mantener el control de una transacción individual encerrándola en una conexión que pueda ser identificada, supervisada y cancelada sin requerir acción específica por parte de las aplicaciones cliente y servidor. Esto da al mecanismo una amplia tolerancia a cambios del desempeño de la red y permite que se puedan añadir fácilmente herramientas de seguridad tales como firewalls y proxies.

El monitoreo continuo de datos o "*streaming data*" no es muy eficiente con el protocolo Modbus/TCP, debido básicamente a la sobrecarga que impone el protocolo de transporte TCP. Esta sobrecarga se debe al servicio de entrega de datos confiable y el acuse de recibo para cada paquete transmitido, incrementándose considerablemente el tráfico en la red cuando se monitorea en todo momento un esclavo Modbus/TCP particular. Sin embargo, esto tiende a ser un problema menor a medida que Ethernet aumenta de velocidad. La solución para la supervisión de datos continuos sobre una red Ethernet es la utilización del protocolo de transporte UDP, pero se pierde confiabilidad.

b. Modelo de datos:

MODBUS basa su modelo de datos sobre una serie de tablas las cuales tienen características distintivas. Las cuatro principales son:

Entradas discretas: bit único; suministradas por un sistema I/O (entrada/salida); de sólo lectura.

Salidas discretas: bit único; alterable por un programa de aplicación; de lectura-escritura.

Registros de entrada: 16 bits suministrados por un sistema I/O; de sólo lectura.

Registros de salida: 16 bits, alterables por un programa de aplicación; de lectura-escritura.

La distinción entre entradas y salidas, y entre datos direccionables por bit y direccionables por palabra no implica algún comportamiento de la aplicación. Es aceptable y común considerar las cuatro tablas solapándose una con otra, si ésta es la interpretación más natural sobre la máquina (esclavo MODBUS) en cuestión.

c. Importancia de la longitud:

Todas las solicitudes y respuestas MODBUS están diseñadas en tal forma que el receptor pueda verificar que un mensaje está completo. Para códigos de función donde la solicitud y respuesta tienen longitud fija, basta el código de función.

Para códigos de función llevando una cantidad variable de datos en la solicitud ó respuesta, la porción de datos estará precedida por un campo que representa el número de bytes que siguen.

Cuando MODBUS es transportado sobre TCP, se adiciona información de longitud en el prefijo (o encabezado) para permitir al receptor reconocer los límites del mensaje, así el mensaje haya sido dividido en múltiples paquetes para la transmisión. La existencia de reglas de longitud implícitas o explícitas y el uso de un código de chequeo de error CRC-32 (sobre Ethernet), resulta en una

probabilidad muy pequeña de corrupción no detectada sobre un mensaje de solicitud o respuesta.

d. Ventajas del protocolo MODBUS/TCP:

Es escalable en complejidad. Un dispositivo que tenga un propósito simple necesita implementar sólo uno o dos tipos de mensaje. Es simple para administrar y expandir.

El protocolo puede utilizarse para comunicar una gran cantidad instalada de dispositivos MODBUS, identificando cada uno de ellos con diferente número de periférico a la hora de configurar el medidor.

Es de muy alto desempeño, limitado típicamente por las capacidades de comunicación del sistema operativo del computador. Se pueden obtener altas tasas de transmisión sobre una estación única y la red puede configurarse para lograr tiempos de respuesta garantizados en el rango de milisegundos. No existe una especificación precisa acerca del tiempo de respuesta requerida para una transacción sobre MODBUS o Modbus/TCP. Esto debido a que se espera que Modbus/TCP se utilice en la más amplia variedad posible de situaciones de comunicación, desde sistemas I/O esperando temporización en milisegundos, a enlaces de radio de larga distancia con retardos de varios segundos.

En general, los dispositivos tales como PLC's responderán a solicitudes de entrada en un tiempo que típicamente puede variar entre 20 y 200 ms. Desde la perspectiva del cliente, ese tiempo de respuesta debe extenderse por los retardos de transporte a través de la red a un tiempo de respuesta razonable. Tales retardos pueden ser de milisegundos en una red de área local, hasta cientos de milisegundos para una conexión de red de área amplia (WAN).

Toda temporización utilizada en un cliente debe ser mayor que el tiempo máximo de respuesta razonable, para así evitar una excesiva congestión en el dispositivo

servidor o en la red, lo cual puede causar errores. En la práctica las temporizaciones usadas en aplicaciones de alto desempeño serán probablemente algo dependientes de la topología de la red y el desempeño esperado del servidor.

Los clientes pueden cerrar y re-establecer conexiones Modbus/TCP cuando el *timeout* ha expirado. Sin embargo, al retransmitir una solicitud es aconsejable establecer un *timeout* más grande para permitir al servidor restablecerse de una posible condición de falla.

Si bien en la universidad no se utilizará el protocolo de comunicación Modbus para aplicaciones con PLC's ni con otros dispositivos de automatización; es necesario manejar el protocolo Modbus porque los medidores de energía utilizados en este trabajo de comunicación, poseen entre sus características, la de comunicación mediante este protocolo.

Sin embargo se convierte el protocolo Modbus, el cual utiliza una interfaz RS-485, a una interfaz RS-232 que es con la que normalmente trabajan las computadoras de escritorio.

Más adelante en este trabajo de investigación sobre medición remota, se abordará el tipo de convertidor utilizado, y también la configuración de la interfaz que permite la comunicación entre el medidor de energía (Circutor CVMk o Veris H8400 Series) y el ordenador en el cual se descargarán los datos enviados por el medidor de energía.

Para lograr la transmisión de datos, primero se debe hacer una petición de lo que se quiere leer. En este caso cada variable eléctrica está en determinados registros del medidor al cual hace referencia el software con el cual se quieren leer los datos. Estos registros dependen del medidor con que se trabaje.

La forma de hacer la petición para que el medidor Circutor CVMk transmita datos es la siguiente¹²:

Petición → 0A 03 00 26 00 10 A4 B6

0A Número de periférico del medidor del cual se quieren leer datos.

03 Función de Lectura

00 26 Dirección inicial (primer registro)

00 10 Número de registros a leer

A4B6 caracter CRC

Respuesta → 0A 03 20 00 00 00 D4 00 00 23 28 00 00 0F A0 00 00 00 00 00 00 00 00 00 00 60 00 00 01 F4 00 00 0F A0 B7 8B

0A Número de periférico del medidor del cual se quieren leer datos.

03 Función de Lectura

20 Bytes de respuesta de datos

00 00 00 D4 Vprom trifásico (registro 26 Hex), en decimal 212 V

00 00 23 28 mA promedio trifásico, en decimal 9000 mA

00 00 0F A0 W trifásico en decimal 4000 W

00 00 00 00 varL trifásico, en decimal 0 varL

00 00 00 00 varC III en decimal 0 varC

00 00 00 60 PF en decimal 96 PF

00 00 01 F4 Hz en decimal 50 Hz

00 00 0F A0 VA III en decimal 4000 mA

B7 B8 carácter CRC

De manera muy similar se hace para el medidor Veris H8437, pero como se mencionó anteriormente, se le otorgará mayor importancia al medidor CMVk por factores mencionados con anterioridad.

Lo anterior se hace cuando se quiere crear un software para la lectura de datos; en este caso particular, el software de cada uno de los medidores hace ese trabajo.

¹² Manual del Analizador de redes Circutor CVMk

En el caso del Circutor CVMk se utiliza para la descarga de datos el software Power Studio versión 1.0; y para el medidor Veris H8437 se utiliza un software en el ambiente del software Labview¹³, los cuales permiten la medición de datos en tiempo real.

Para el caso del Power Studio, tiene la opción de historial por fechas de almacenamiento de datos. Y en el caso del Labview, exporta los datos para almacenarlos en una hoja de cálculo de Microsoft Excel, desde el cual se manejarán los datos para llevar a cabo la medición remota.

3.4 Recepción de Datos.

La recepción de datos desde el medidor como ya se dijo anteriormente¹⁴ se hará mediante dos interfaces con conector DB9. Estas interfaces corresponden a un protocolo Modbus para el medidor y un protocolo TCP/IP para el ordenador donde se recibirán los datos.

Las interfaces no tienen una configuración común para un conector DB-9, la configuración tanto para la parte RS485 como para la RS232, como se mencionó anteriormente, trabajan con 3 conductores cada una, incluyendo el cable de tierra.

El cambio de RS485 a RS232 se hace mediante el convertidor inteligente marca Circutor RS485/RS232¹⁵, en el cual se establece la velocidad de comunicación; en este caso de 9600 baudios dada la cercanía del medidor al ordenador (dos metros aproximadamente). Se debe considerar que cuanto mayor sea la distancia entre el medidor y el ordenador mayor debe ser la velocidad en baudios a la cual debe establecerse la comunicación par evitar pérdida de datos en la transmisión de los mismos a grandes distancias(Ver <http://html.rincondelvago.com/medios-de-transmision.html>.)

¹³ Consultar el trabajo de graduación: ANÁLISIS TÉCNICO-ECONÓMICO DEL COMPORTAMIENTO DE VARIABLES ELÉCTRICAS QUE INCIDEN EN LA FACTURACIÓN Y CONSUMO DE ENERGÍA ELÉCTRICA EN EL CENTRO DE INVESTIGACIONES Y TRANSFERENCIA DE TECNOLOGÍA (CITT), Capítulo III.

¹⁴ Ver apartado 3.3

¹⁵ Ver capítulo 4

En el caso particular del medidor de energía y el ordenador ubicados en la subestación #2 del CITT¹⁶ de la universidad Don Bosco, no es necesaria una tasa de comunicación de datos muy alta debido a la cercanía entre los dos dispositivos.

Al hablar de recepción de datos en el ordenador desde el medidor de energía hay que separar la recepción si se usan medidores de marca diferente. Para este caso en particular se han tomado dos medidores de energía, el Circutor CVMk y el Veris H8437, siendo este último de menor relevancia en relación al uso que se le da en la universidad. Recordemos que el medidor Circutor CVMk es el que actualmente se encuentra instalado en cada una de las tres subestaciones del CITT.

3.4.1 RECEPCIÓN DE DATOS EN EL MEDIDOR DE ENERGÍA CIRCUTOR CVMk.

Para el medidor CVMk se cuenta con el software PowerStudio versión 1.0, este se encarga de enlazar los datos que recibe el puerto COM del ordenador y los procesa para mostrarlos en pantalla.

El puerto serial de un ordenador posee un conector tipo DB9 de los cuales para la recepción de datos y para la petición o solicitud de lectura al medidor de energía se utilizan solamente los pines 2,3 y 5; para recepción de datos, transmisión de datos y el tierra respectivamente.

El ordenador mediante un software transmite por el pin 3 una petición de lectura al medidor de energía y este le devuelve el número o los números de registros solicitados por el pin 2, por el cual recibe datos el ordenador.

Lo anterior se logra mediante el software PowerStudio el cual hace la petición de un determinado número de registros al medidor de energía Circutor CVMk y luego procesa esta información. Es de mencionar que los datos que envía el medidor, al

¹⁶ Centro de Investigación y Transferencia de Tecnología

ordenador, no todos vienen representando el valor real de la magnitud medida. Por tal razón el fabricante proporciona una tabla de los registros¹⁷ para hacer la correcta petición o solicitud de lectura y además dependiendo de la variable eléctrica solicitada, esta puede o no, tener un factor incluido o por incluir tal como se muestra en dicha tabla de registros, para adecuar la lectura con la señal medida.

El software PowerStudio posee diferentes características en los cuales se puede monitorear varios modelos de analizadores de redes marca Circutor, con un historial de datos almacenados que se pueden presentar por días, por semana o por mes con el objetivo de obtener un mejor análisis sobre el sistema para el cual se encuentre instalado.

Es de suma importancia utilizar el convertidor inteligente RS485/RS232, porque sin él, la computadora no reconoce los datos que le envía el medidor y de hecho ni siquiera pudiera el ordenador solicitar datos al medidor por encontrarse con protocolos de comunicación diferentes (RS232 y RS485).

3.4.2 RECEPCIÓN DE DATOS EN EL MEDIDOR DE ENERGÍA VERIS H8437

El equipo analizador de redes VERIS H8437 no tiene la capacidad de almacenamiento de mediciones incorporada dentro de si mismo, en vista de esto se ha tenido la necesidad de utilizar un programa desarrollado en el Software LabView¹⁸ el cual permite hacer programas compatibles con el medidor, ya que este lenguaje de programación es orientado a procesos industriales. Por medio del programa creado capturan los parámetros eléctricos necesarios para el análisis y los almacena en una hoja de cálculo en Excel.

¹⁷ Ver anexo B4

¹⁸ Consultar el trabajo de graduación: ANÁLISIS TÉCNICO-ECONÓMICO DEL COMPORTAMIENTO DE VARIABLES ELÉCTRICAS QUE INCIDEN EN LA FACTURACIÓN Y CONSUMO DE ENERGÍA ELÉCTRICA EN EL CENTRO DE INVESTIGACIONES Y TRANSFERENCIA DE TECNOLOGÍA (CITT), Capítulo III.

El medidor envía pulsos de tensión por medio de tres cables hacia un convertidor de puerto serie, los cuales corresponden a un protocolo de comunicación Modbus al igual que el analizador de redes CVMk, el cual comunica al medidor con la computadora.

Cada variable del medidor posee un registro específico en su memoria interna¹⁹, por ejemplo: en el listado del manual se puede buscar el registro para el factor de potencia promedio que es 006, y este dato en la programación solo se antepone el número 1.40 para darle el direccionamiento y luego el código.

Los datos provenientes del registro solicitado necesitan ser multiplicados por un factor para obtener el valor real, este factor se obtiene de los rangos de valores máximos y mínimos de los parámetros. En la figura 3.7 se presenta la codificación gráfica para la realización de la captura de datos de FP promedio, voltaje de línea promedio, voltaje de fase promedio y la corriente promedio.

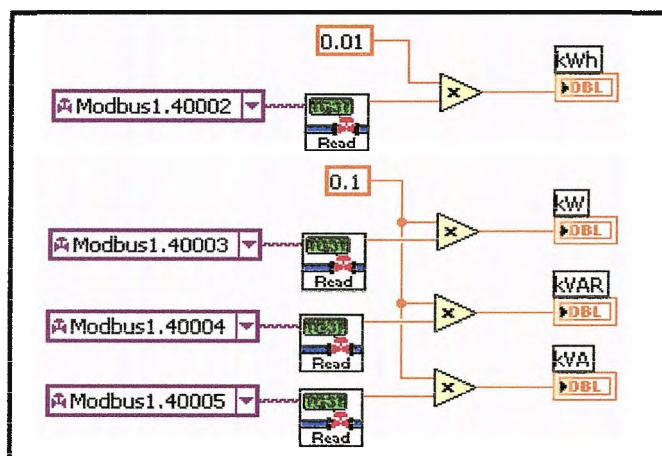


Figura 3.7 Código gráfico de captura de datos del programa en Labview.²⁰

¹⁹ Ver anexo B6

²⁰ Consultar el trabajo de graduación: ANÁLISIS TÉCNICO-ECONÓMICO DEL COMPORTAMIENTO DE VARIABLES ELÉCTRICAS QUE INCIDEN EN LA FACTURACIÓN Y CONSUMO DE ENERGÍA ELÉCTRICA EN EL CENTRO DE INVESTIGACIONES Y TRANSFERENCIA DE TECNOLOGÍA (CITT), UDB.

Como se dijo anteriormente este programa en Labview permite descargar e ir almacenando datos que proporcione este medidor en una hoja de cálculo de Excel, desde donde los datos se pueden procesar para otras aplicaciones.

En este caso particular se hará un programa en Matlab para poder acceder más fácilmente a la lectura de este archivo de Excel desde un lugar remoto e importar los datos desde Matlab para obtener una lectura con mayor facilidad de entendimiento. Para nuestro trabajo de graduación si bien es cierto que Labview es orientado a procesos industriales, específicamente a la instrumentación virtual se ha escogido hacer la aplicación en Matlab primero debido al conocimiento que como estudiantes tenemos de dicho programa y en segundo lugar a la facilidad con que se puede abrir archivos de cualquier otra aplicación de Microsoft.

Otra razón de peso es que en el departamento de eléctrica se cuenta con ordenadores que tienen licencia en Matlab, aprovechando de esta manera el software actualmente instalado en dicho departamento.

CAPÍTULO IV. DESCRIPCIÓN DEL EQUIPO UTILIZADO

4. CARACTERÍSTICAS TÉCNICAS

Es de gran importancia conocer las características técnicas de cada uno de los dispositivos que permitirán la medición remota de las variables eléctricas que proporcionará el analizador de redes marca Circutor y modelo CVMk y también conocer estas características para el otro medidor con que cuenta la universidad Don Bosco, el medidor VERIS H8437.

4.1 ANALIZADOR DE REDES CIRCUTOR CVMk.

El circutor CVMk

²¹, posee ciertas características tanto constructivas como abstractas que permiten optimizar el uso de este analizador.

4.1.1 Descripción de analizador de redes CVMk.

- ✓ Es un instrumento de panel de dimensiones (144 x 144 mm).
- ✓ Medición en verdadero valor eficaz.
- ✓ Memorización de los valores máximos y mínimos.
- ✓ Visualización de los parámetros con escala automática de unidades.
- ✓ Display de LCD ó LED (tipos CVMk ó CVMk- L).
- CVMk y CVMk-ITF: Display cristal liquido, 4 dígitos (LCD)
- CVMk L y CVMk L - ITF: Display de LED (4 ½ dígitos)
- CVMk-H: Mide THD y el resto de parámetros de un CVMk estándar
 - ✓ Teclado con 4 botones, para el control y programación (reset, display, max y min).
 - ✓ Dispone de 3 x 3 LED luminosos (rojo, verde y amarillo), para indicar el parámetro visualizado en display.

²¹ Ver Anexo A.1

- ✓ Posibilidad de comunicación RS-232 ó RS-485 (protocolo CIRCUTOR ó MODBUS).

El CVMk permite la visualización de hasta 30 parámetros eléctricos (43 parámetros mediante módulos expansión), mediante tres pantallas numéricas de grandes dimensiones. En las pantallas se visualizan: (Pantalla 1) La tensión simple o compuesta de las tres fases. (Pantalla 2) 3 parámetros a elegir de los 30 posibles. (Pantalla 3) 3 parámetros a elegir de los 30 posibles.

Para cada variable eléctrica en cada fase corresponde un código individual para programar lo que se quiere mostrar en pantalla.

Las características de la variedad de modelos de medidores Circutor de la serie CVM se resumen en la siguiente tabla:

Modelo	Trifásico 50..60 Hz.	Medida tensión 500Vca	Medida tensión 110Vca	Verdadero valor eficaz (RMS)	Analizador de redes	Medidor armónico hasta 50	Tipo Display	Entradas corrientes aisladas (ITF)	2 cuadrantes	4 cuadrantes	Alimentación 230/400 Vca	Alimentación 24/120 Vcc	Medida C.C.	Medida demanda Max.	Medida THD (V/A)	Comunicaciones PC*	Transformador externo
Mk	✓	✓	op	✓	✓		LCD		✓		✓			✓		✓	
Mk-ITF	✓	✓	op	✓	✓		LCD	✓	✓		✓			✓		✓	op
Mk-ITF/SDC24..120V	✓	✓		✓	✓		LCD	✓	✓			✓		✓		✓	op
Mk-ITF M110/SDC24..120V	✓		✓	✓	✓		LCD	✓	✓	op	op	✓		✓		✓	
Mk L	✓	✓	op	✓	✓		LED		✓		✓	op		✓		✓	
Mk L-ITF	✓	✓	op	✓	✓		LED	✓	✓		✓	op		✓		✓	op
Mk H-ITF	✓	✓	op	✓	✓		LCD	✓	✓	op	✓	op		✓	✓	✓	op
Mk L-H-ITF	✓	✓	op	✓	✓		LED	✓	✓	op	✓	op		✓	✓	✓	op
Mk-HAR-ITF	✓	✓	op	✓		✓	LCD	✓	✓		✓	op			✓	✓	op
Mk-HAR-L-ITF	✓	✓	op	✓		✓	LED	✓	✓		✓	op			✓	✓	op
Mk-4C-ITF	✓	✓	op	✓	✓		LCD	✓		✓	✓	op		✓	✓	✓	op

Tabla 4.1 Características medidor CVMk²²

²² Obtenida de Página Web: <http://www.ctas-sa.com/circutor/> fecha de descarga 15/01/09

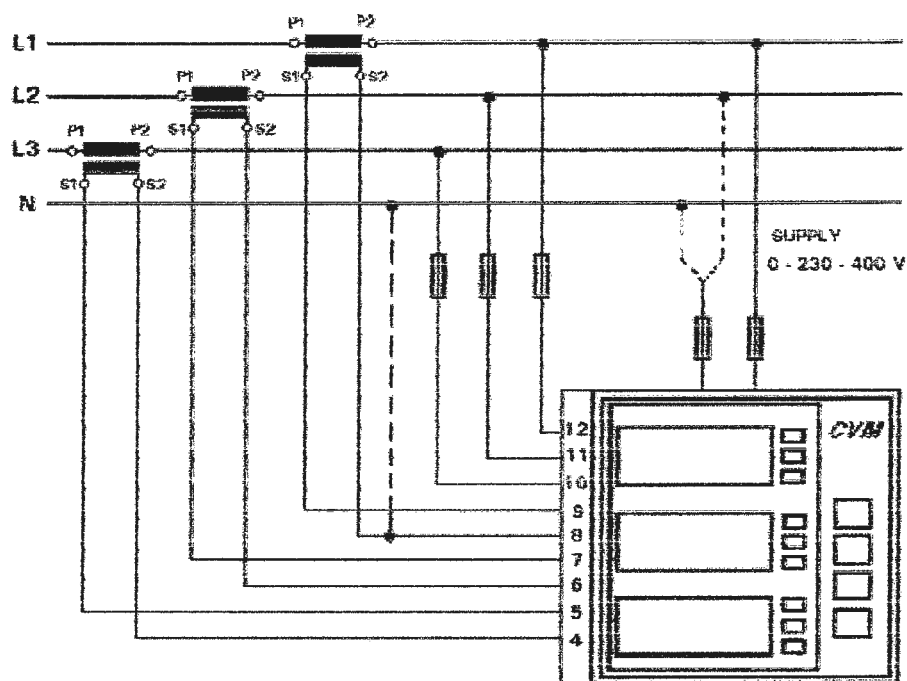


Figura 4.1 Modo de conexión del dispositivo CVMk(Obtendida de Manual Circutor)

En general cada medidor de la Serie CVM tiene las siguientes características que se muestran en la siguiente tabla:

Modelos		CVMk, CVMk-HAR	CVM 96	CVM 144	CVM-BD
características de funcionamiento	Tensión	230/400 Vca	230 Vca		
	Tolerancia tensión	+10% / -15%			
	Frecuencia	50 .60 Hz			
	Consumo	3 a 6 VA según tipo			
características de medición	Tensión Nominal (Vfn/Vff)	500/865 Vca	300/520 Vca	500/865 Vca	
	Otras tensiones	A través de transformadores	A través de transformadores	A través de transformadores	
	Corriente Nominal	In = 5 A. (Entrada aislada tipo ITF)	In = 5 A. (Entrada aislada tipo ITF)		
	Sobrecarga permanente	1.2 In			
	Consumo circuito corriente	0,6 VA	0,75 VA	0,6 VA	
características de precisión	Tensión	< 0,5 % de la lectura +/-2 dígitos			
	Corriente	< 0,5 % de la lectura +/-2 dígitos			
	Potencia	< 1 % de la lectura +/-2 dígitos			
características constructivas	Montaje	Superficie de panel			Riel DIN
	Conexión	Por regleta			Bornes metálicas con tornillos
	Protección (montaje frontal)	IP41	IP55	IP55	IP41
	Dimensiones	144x144mm	96x96mm	144x144mm	140x110x70mm
normas	IEC 801, IEC 571-1, IEC 1010-1, IEC 664, VDE 0110, UL 94				

Tabla 4.2 Características Técnicas medidores serie CVM

Mediciones Eléctricas

La medida se realiza en verdadero valor eficaz, mediante tres entradas de tensión CA y tres entradas de corriente CA. La medida se realiza con transformadores de corriente. / 5 A. (entradas aisladas en el tipo ITF).

4.1.2 MODO DE OPERACIÓN

El instrumento tiene tres pantallas, y cada uno tiene tres indicadores LED (rojo, verde y amarillo). Cada LED indica el parámetro actual que se muestra en pantalla. Cuando encienda el suministro de energía de los CVMk-H, los 9 indicadores LED se encenderán algunos segundos, y verá en pantalla: "Circutor xxxx" (Versión del programa) y después leerás "CARD TYPE xxxx" (identificación del módulo conectado). Después de algunos segundos, el instrumento está listo para trabajar, mostrando una de las tres posibles pantallas. Uno de los LED al lado de cada pantalla indica el parámetro que se mide.

Botón DISPLAY en el Medidor CVMk:

Cuando el primer LED (rojo) de cada pantalla está encendido, significa que estamos leyendo los valores de la tensión. Es decir, la primera pantalla muestra el voltaje de la fase L1 (V1), la segunda, la tensión de la fase L2 (V2) y la tercera de la tensión de la fase L3 (V3). Los LEDs colocados junto a cada pantalla indican los parámetros que se muestran en los DISPLAY. Presionando la tecla "DISPLAY" se accede a las diversas pantallas de visualización, y son identificados por cada LED. La pantalla por defecto es mostrado a continuación, aunque esta configuración puede ser definido por el usuario con el fin de mostrar otros parámetros.

Para una mejor idea de que nos indica cada LED se muestra la siguiente tabla para identificar que parámetro se muestra en el Display:

PARAMETRO QUE SE MUESTRA EN PANTALLA			LED
V1	V2	V3	ROJO
I1	I2	I3	VERDE
KW III	PF III	HZ	AMARILLO
THD ²³ V1	THD V2	THD V3	ROJO Y VERDE
THD I1	THD I2	THD I3	ROJO Y AMARILLO

Tabla 4.3 Parámetros que indican los LED

Botón max en el medidor CVMk

Al presionar el botón "max", los valores máximos para los parámetros que se muestran aparecen en la pantalla. Esta función sólo es válida mientras se mantenga pulsando el botón "max". Si usted deja de pulsarlo el valor instantáneo aparece de nuevo. Cuando se muestran los valores máximos los indicadores LED se mantienen parpadeando.

Botón min en el medidor CVMk

Pulsando el botón "min", los valores mínimos para los parámetros que se muestran aparecen en la pantalla. Esta función sólo es válida mientras se mantenga pulsando el botón "min". Si usted deja de pulsarlo el valor instantáneo aparece de nuevo. Cuando se muestran los valores mínimos los indicadores LED se mantienen intermitentes.

Botón reset en el medidor CVMk

Pulsando el botón "reset" el sistema se reinicia. Esto equivale a apagar la fuente de alimentación del instrumento. Los valores máximos y mínimos almacenados se eliminan automáticamente de la memoria interna.

²³ Ver definición en el Glosario

Si usted se encuentra en el proceso de configuración y presione el botón "reset", saldrá sin guardar ninguna modificación que usted haya hecho y reinicia el sistema.

SETUP

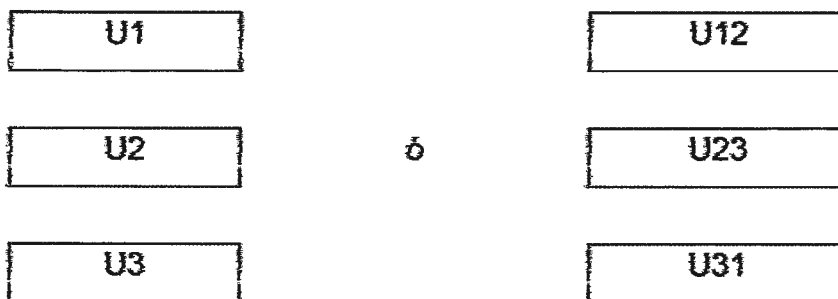
Para acceder al menú de configuración siga estos pasos:

- (a) Conectar (alimentación) el instrumento.
- (b) Presione los dos botones verdes (max, min) simultáneamente.

Verá durante unos pocos segundos la palabra "SET". Esto significa que estamos en el proceso de configuración. A continuación se muestran las diferentes opciones paso a paso:

- Tensiones Fase a Fase o Fase – Neutro.

Después de la palabra "Set" se verá en los tres Display los voltajes de las fases L1, L2, L3.



Tensión Fase a Neutro: U1, U2, U3

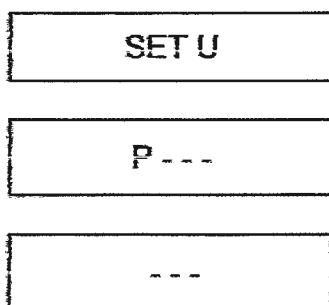
Tensión Fase a Fase: U12, U23, U31

- a) Para seleccionar una de las opciones de voltaje simplemente pulse la tecla verde "max" y ambas opciones aparecen alternativamente.
- b) Al llegar en la pantalla a la opción que desea pulse el botón "Display" para validarla y acceder a la siguiente opción de configuración.

➤ **Transformadores de tensión primaria**

Para los transformadores de tensión primaria podrá tenerse a disposición 6 dígitos para introducir el valor de la tensión que se desea y para el transformador de tensión secundario únicamente tres dígitos por ejemplo: 000001/001

Para el caso de la configuración del transformador de tensión primaria en la pantalla se lee la palabra "SET U", seguido de 6 dígitos. Que nos permiten establecer el voltaje primario del transformador.



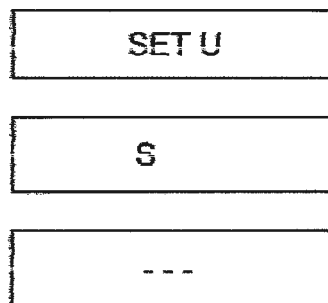
El último dígito del primer Display indica "U" (Tensión) y el primer dígito del segundo Display indica "P" (Primaria). Esto significa que podemos configurar el voltaje primario del transformador. Para estar seguros que se configura el valor de tensión el LED rojo permanecerá iluminado.

- Para escribir o modificar el valor es necesario presionar en repetidas ocasiones el botón "max" el dígito se volverá intermitente y se incrementará.
- Cuando el valor en la pantalla es el adecuado, podemos pasar al siguiente dígito pulsando el botón "Min" con el fin de modificar los otros valores.
- Cuando el dígito intermitente es el último, pulsando el botón "min" se retorna al valor inicial: los valores pueden ser modificados de nuevo.

- Presiona "Display" para pasar a la siguiente opción de configuración.

➤ Tensión secundaria del transformador.

En este punto podemos establecer el valor de tensión del secundario del transformador. Para este sólo tres dígitos están disponibles:



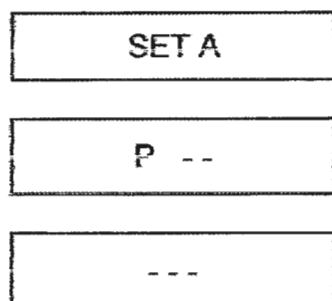
Se utiliza el mismo procedimiento para la tensión primaria del transformador.

- El Botón "max": Permite modificar el valor del dígito que parpadea. Cada vez que se pulsa el valor se incrementa.
- El Botón "min": Permite la validación del dígito que parpadea y pasa al siguiente.
- Pulse "display" para pasar a la siguiente opción de configuración. Si el CVMk-H está conectado directamente a la red (sin transformador de tensión) los valores del primario y el secundario deben ser los mismos.

➤ Transformador de Corriente Primario.

En este punto aparecerá en pantalla "SET A" y en la segunda pantalla "P", cinco dígitos aparecen en la pantalla los cuales nos permite

configurar el transformador de corriente primario. Los LED verdes encienden y nos indican que se está configurando los valores de corriente.



Se utiliza el mismo procedimiento descrito anteriormente para los botones max, min y Display.

Nota:

- El Valor Máximo programable es de 10.000.
- El transformador de corriente secundario no es programable. Automáticamente toma valores normalizados de 5 A (... 5 A AC).

4.1.3 Módulos de Expansión CVMk

- Tarjetas que se conectan en la parte posterior del medidor CVMk (Máximo 1 por CVMk).
- Opciones: Contadores de energía (KWH, KVARLh y KVARCh).
- Comunicaciones serial RS232/485.
- Salidas 4-20 mA, con escala programable (Offset y plena escala).
- Salidas de Relés, programables como pulsos de energía o alarmas de cualquier parámetro.
- Triple tarifa con nueve contadores de energía.

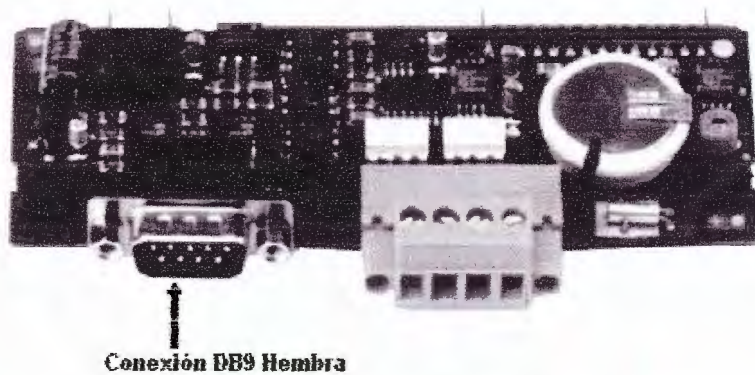


Figura 4.2 Módulo de Expansión (Obtenida de Manual Circutor CVMk)²⁴

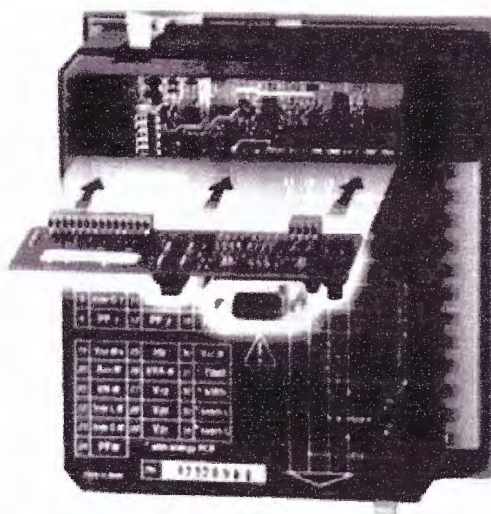


Figura 4.3 Módulo de Expansión conectado a CVMk(Obtendida Manual Circutor)

²⁴ Ver Página Web: <http://www.clas-sa.com/circutor/>

En la siguiente tabla se muestran los diferentes modelos de medidores Circutor de la serie CVM para los cuales es compatible el módulo de expansión CVMk:

Modelo	Energía + reloj	Comunicación PC (RS-232)	Comunicación PC (RS-485)	Com. Perif. (RS-485)	Salidas Relés	Salidas 4..20mA	Alimentación externa	Ent. sincron. máx. tipo tarif.	Máx. demanda	Compatibilidad		
										CVMk	CVM-DC	CVMk-HAR
CVM / ER	✓								✓	✓		✓
CVM / 485			✓						✓	✓	✓	✓
CVM / 232		✓							✓	✓	✓	✓
CVM / ER-485	✓		✓						✓	✓	✓	✓
CVM / ER-232	✓	✓							✓	✓	✓	✓
CVM / RED	✓		✓	✓					✓	✓	✓	✓
CVM / ER 420-1	✓					1			✓	✓		
CVM / ER 420-2	✓					2			✓	✓		
CVM / ER C-2	✓				2				✓	✓	✓	✓
CVM / ER C 420-1	✓				1	1			✓	✓	✓	
CVM / RED-MAX	✓		✓	✓				✓	✓	✓		
CVM / RED 420-1	✓		✓	✓		1	✓		✓	✓		
CVM / RED C-2	✓		✓	✓	2				✓	✓	✓	✓

Tabla 4.4 Compatibilidad Módulo de Expansión CVMk (Obtenida de Manual Circutor)

Conectores utilizados con el módulo de expansión CVMk:

4.1.4 El Estándar RS 232

El RS-232C consiste en un conector tipo DB-25 (25 pines), aunque es normal encontrar la versión de 9 pines DB-9, mas barato e incluso más utilizado para cierto tipo de periféricos (como el ratón serie de la computadora personal). En cualquier caso, estas computadoras no suelen emplear más de 9 pines aún utilizando el conector DB-25. Las señales con las que trabaja este puerto serie son digitales, de +12V (0 lógico) y -12V (1 lógico), para la entrada y salida de datos, y a la inversa en las señales de control. El estado de reposo en la entrada y salida de datos es -12V. Dependiendo de la velocidad de transmisión que se emplea, es posible tener cables de hasta 15 metros.

Cada pin puede ser de entrada o de salida, teniendo una función específica para cada uno de ellos. Las más importantes son:

Pin	Función
TXD	Transmitir datos
RXD	Recibir Datos
DTR	Terminal de Datos listo
DSR	Equipo de Datos listo
RTS	Solicitud de Envío
CTS	Libre para Envío
DCD	Detección de portadora

Tabla 4.5 Pines y su función de un DB9 o puerto serie

Las señales TXD, DTR y RTS son de salida, mientras que RXD, DSR, CTS y DCD son de entrada. La masa de referencia para todas las señales es SG (Señal de Tierra o por sus siglas en inglés *Signal Ground*). Finalmente, existen otras señales como RI (Indicador de Llamada), y otras poco comunes que no se explican en esta oportunidad ya que no serán necesarias para nuestro proyecto.

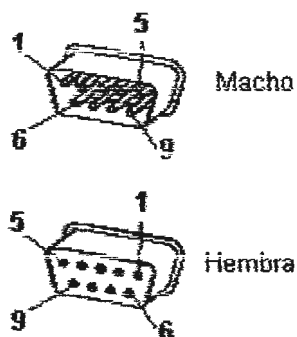


Figura 4.4 Conector DB9 macho y hembra.

A continuación se muestran las señales que le corresponden a cada uno de los pines del conector DB9:

Numero de Pin	Señal	Descripción	E/S
En DB-9			
1	-	Masa chasis	-
3	TxD	Transmit Data	S
2	RxD	Receive Data	E
7	RTS	Request To Send	S
8	CTS	Clear To Send	E
6	DSR	Data Set Ready	E
5	SG	Signal Ground	-
1	CD/DCD	(Data) Carrier Detect	E
-	TxC(*)	Transmit Clock	S
-	RxC(*)	Receive Clock	E
4	DTR	Data Terminal Ready	S
9	RI	Ring Indicator	E
-	RTxC(*)	Transmit/Receive Clock	S

Tabla 4.6 Pines y su descripción de un DB9 o puerto serie

4.1.5 Conector RS 485

El principio de operación eléctrico se basa en el uso de transmisores diferenciales con tensiones oscilantes entre 0 y 5 V.

El estándar RS-485 se creó para permitir la conexión de varios dispositivos a través de una línea común. Cada dispositivo representa una Unidad de Carga (*UL: Unit Load*) y se denomina nodo. Una red RS-485 puede contener hasta 32 *UL*, donde cada *UL* representa una carga de 12 K Ω .

En resumen, las características sobresalientes de la RS-485 son:

- Alta inmunidad al ruido.
- Largo máximo de los conductores: 1300 metros (depende de la velocidad).
- Alta velocidad (hasta 10 Mbps).
- Capacidad de conectar hasta 32 nodos.
- Capacidad de configuración en modo multi-Master.
- Tensión de emisión: nivel «0»: +1.5 V a +5 V, nivel «1»: -1.5 V a -5 V.
- Nivel de recepción: nivel «0»: > +0.2 V, nivel «1»: < -0.2 V.
- Puerto serie DB9 para consola de configuración y monitorización.

4.1.6 Módulo Convertidor Inteligente (Accesorio de comunicación).

Convertidor RS-232/RS-485²⁵

- Permite comunicación bidireccional de las distintas configuraciones.
- Convertidor bajo entorno Windows.
- Convertidor inteligente, donde no es necesario controlar la señal RTS (Convertidor RS-232/RS-485).

El módulo aislador convertidor inteligente cumple una doble función:

1. Aislar eléctricamente las tierras de la computadora y el cable.
2. Convertir los niveles de tensión del RS-232 a RS-485 y viceversa.

Debe conectarse al PC por un cable RS232 de 9 pines para ambos terminales. En el caso del convertidor inteligente, no será necesario controlar la señal RTS, lo cual es muy útil para usuarios que comuniquen con autómatas o que desarrollen el software de comunicación.

El convertidor inteligente es necesario para el software de entorno Windows.

²⁵ Ver hoja de datos del fabricante en Anexo B.3

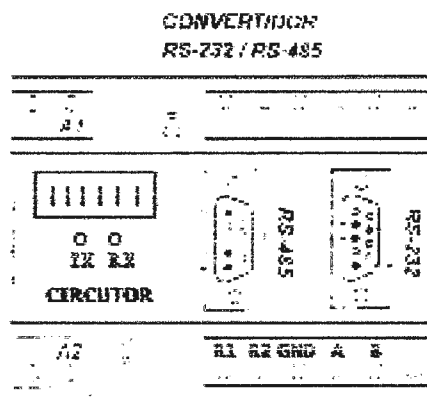


Figura 4.5 Carátula del Convertidor Inteligente RS-232/RS-485.

El convertidor dispone de los siguientes elementos:

- Dos bornes para alimentar el equipo (A1, A2): 220 V \pm 10 %, alimentados con CA.
- Entrada – Salida Serie RS-232 (conector DB-9 hembra).
- Entrada – Salida Serie RS-485 (conector DB-9 macho).

Siendo las características de este elemento:

- El consumo es de 3 VA.
- El rango de velocidad puede oscilar entre: 4800 a 38400 baudios.
- Su estructura de longitud de datos: 10 – 11 bits.
- Se puede colocar un puente entre los terminales R1 y R2 para conectar una resistencia final de línea para poder acoplar las impedancias y evitar así rebotes de la señal.
- Posee también dos bornes para poder conectar directamente el cable de la RS-485 que proviene del equipo CVMK, siendo el Terminal A que corresponde a TX+ y el Terminal B que corresponde a TX-.

Este convertidor posee un MICROSWITCH:

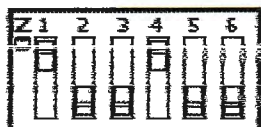


Figura 4.6 Microswitch del Convertidor Inteligente RS-232/RS-485.

Estos interruptores tienen las siguientes funciones:

A. Longitud de Datos (Se debe seleccionar uno solo)

- Interruptor 1 (Posición de ON): Selecciona 10 bits.
- Interruptor 2 (Posición de ON) : Selecciona 11 bits.

B. Velocidad en Baudios (Se debe seleccionar uno solo):

- Interruptor 3 (Posición de ON): Selecciona la velocidad de 4800.
- Interruptor 4 (Posición de ON): Selecciona la velocidad de 9600.
- Interruptor 5 (Posición de ON): Selecciona la velocidad de 19200.
- Interruptor 6 (Posición de ON): Selecciona la velocidad de 38400.

En dicho ejemplo se seleccionó la longitud de datos de 10 bits, a una velocidad de 9600 baudios.

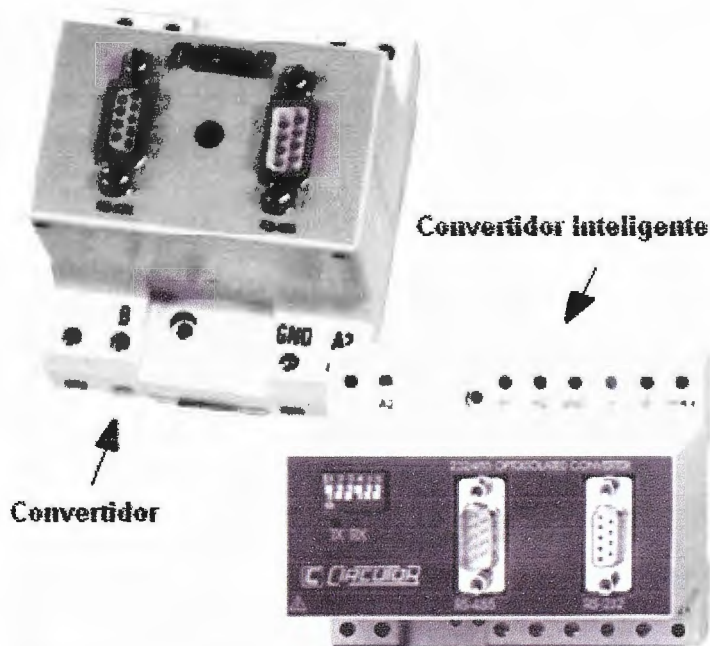


Figura 4.7 Convertidor RS-232/RS-485 y Convertidor Inteligente RS-232/RS-485i.

Bornera de Conexiones.

El CVMK dispone de una bornera de conexiones en el lateral del aparato para poder conectar la alimentación del equipo y las señales de medición de la red.

Dicha bornera se muestra en la siguiente figura:

Termn. nr.	Parameter
13 *	Neutral
12	VL1
11	VL2
10	VL3
9	S1 IL1
8	S2 IL1
7	S1 IL2
6	S2 IL2
5	S1 IL3
4	S2 IL3
Supply	
CVMk-H....-	
A.C. Supply	
3 V
2 V
1	0 V
CVMk-H- / SDC	
D.C. Supply	
3	
2	--
1	+d.c.

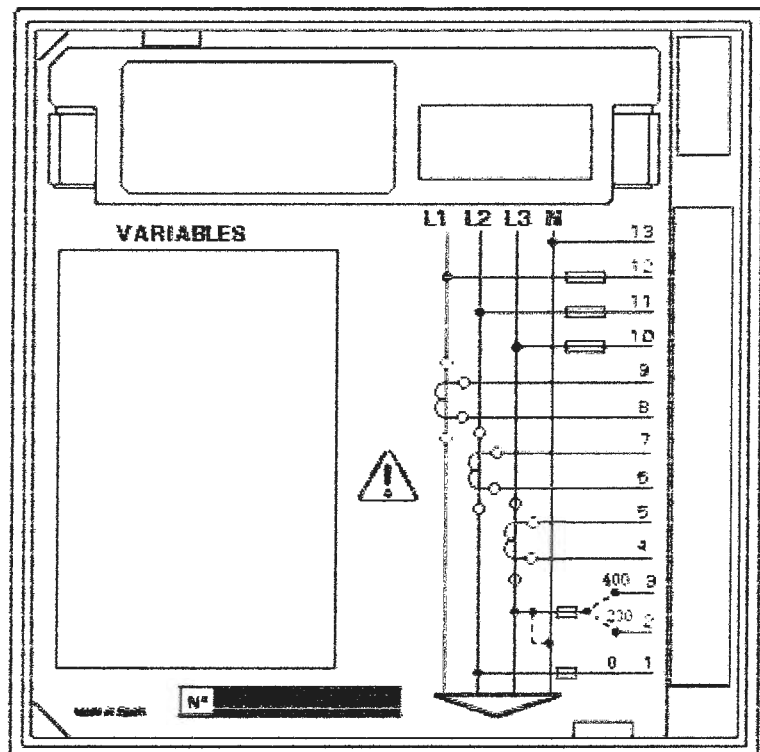


Figura 4.8 Bornera de conexiones del analizador de redes CVMK.

4.1.7 Esquema de Conexión del CVMK.

A continuación se muestra el esquema de conexión del CVMK en una Red Trifásica de baja tensión:

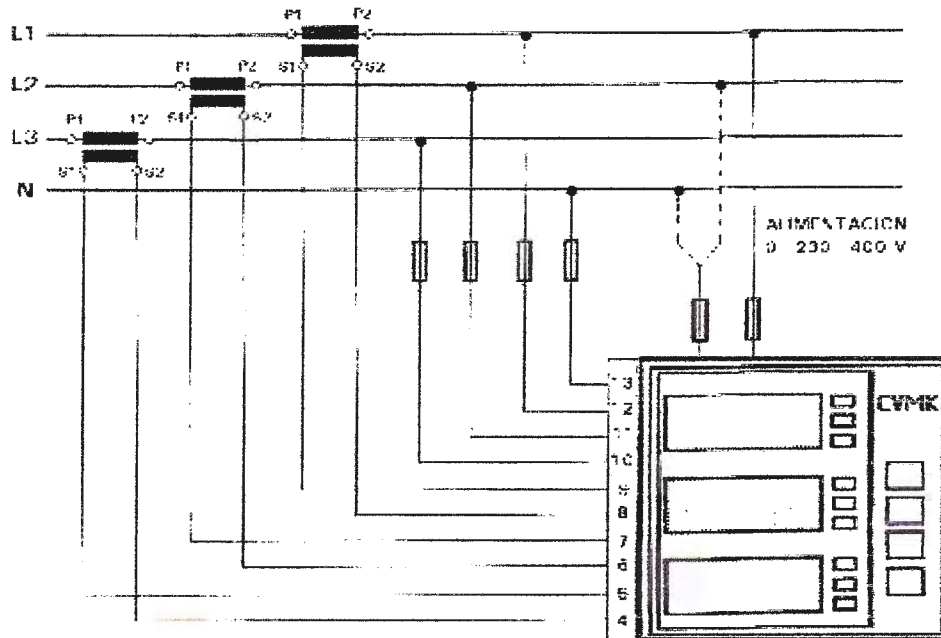


Figura 4.9 Esquema de conexión a TC's del analizador de redes Circutor.

Conexión RED RS-485 a un Ordenador PC (Mediante la interfaz serial RS-232).

La conexión RS-485 se realizó con cable de comunicación de par trenzado con malla de apantallamiento, de tres hilos como mínimo, con una distancia máxima entre el CVMK y el último periférico de 1200 metros. El CVMK con el módulo CVM/RED (Tarjeta de expansión) utiliza una línea de comunicación RS-485 en la que pueden conectarse hasta un máximo de 32 equipos en paralelo (Mediante el Bus Multipunto) por cada COM del ordenador utilizado. Para mayores detalles de la conexión de comunicación referirse al capítulo 3.

4.1.8 Software PowerStudio²⁶

- Destinado al control energético de una instalación eléctrica, para usuarios de equipo Círculo: Analizadores de redes, medidores de energía, indicadores digitales de proceso, centralizadores de pulsos (es un equipo con multitud de entradas, encargado de mandar la información a la PC)
- Características:
 - ✓ Servidor Web Incorporado (Multiusuario).
 - ✓ Visualización en tiempo real de todos los parámetros eléctricos o señales de proceso.
 - ✓ Visualización de gráficos de energías.
 - ✓ Visualización de tablas de datos.
 - ✓ Visualización de Datos Históricos.
 - ✓ Posibilidad de exportación de históricos (XML o Excel) en algunas versiones.
 - ✓ Posibilidad de exportación de parámetros en tiempo real, mediante servidor DDE (*Dinamic Data Exchange*)²⁷ o XML, para entazar con otras aplicaciones externas.
 - ✓ Máxima conectividad a nivel interno (Intranet) y a nivel externo (Internet).
- Principales Aplicaciones:
 - ✓ Controlar niveles de tensión, corriente, potencias, factor de potencia, potencia reactiva, armónicos, etc.
 - ✓ Determinar incrementos de energía, y variaciones del consumo eléctrico.
 - ✓ Mantenimiento preventivo de líneas, tanto a nivel de corrientes residuales, como consumo excesivo.
 - ✓ Subcontaje de energía para líneas de poca potencia vía serie contadores MK, AMK y LM.

²⁶ Consultar en Manual de Power Studio los requerimientos mínimos del ordenador.

²⁷ Ver Definición en Anexo B7

- ✓ Estudios de Calidad de suministro eléctrico.
- ✓ Registro de señales externas de proceso (digitales o analógicas).

Software Cimet:

Hace ya muchos años, CIRCUTOR desarrolló su primer software SCADA (en entorno MS-DOS) que permitía ya, a sus clientes, realizar sus propias aplicaciones. Con la experiencia del primer software, CIRCUTOR ha desarrollado otro, en entorno Windows, aprovechando las ventajas de este sistema operativo y añadiendo un gran número de equipos que permiten realizar al usuario aplicaciones, tanto energéticas como de control CIRNET; una Potente Herramienta de Supervisión y Control, el cual es también un potente SERVER DDE (*Dynamic Data Exchange*), que permite pasar los datos leídos por el sistema SCADA a otras aplicaciones Windows como Excel, Access, etc. A su vez también es posible solicitar datos a CIRNET mediante lenguajes como Visual Basic o programas creados por el usuario, en lenguaje C para Windows.

En la siguiente tabla se muestra la comparación de características técnicas de los Software PowerStudio y Cimet:

Características	Software	
	POWER STUDIO	CIRNET
MS Windows	-	-
Configuración equipo		-
Lectura de datos instantánea	-	-
Visualización de históricos	-	-
Guarda históricos en .dbf		-
Exportación de parámetros en DDE o XML	-	
Scada (monitoreo y control)		-

Tabla 4.7 Comparación características PowerStudio y CIRNET

4.2 Punto de acceso utilizado para la conexión a internet²⁸:

- ✓ Acces point D-Link DWL 3200 AP para redes inalámbricas 54 Mbps
- ✓ Frecuencia 2.4, 5.0 GHZ.
- ✓ Funciona en exteriores.
- ✓ Es compatible con el estándar 802.11a, 802.11b y 802.11g
- ✓ Seguridad WPA Enterprise, WPA Personal, 64-, 128-, 152- bit WEP, MAC Address Acces Control List, 802.11i-Ready.
- ✓ LEDES: Power,LAN, 802.11b/g
- ✓ Dimensiones: Ancho: 15 cm, Profundo: 27 cm, Alto: 4.5 cm.

4.3 Tarjeta de Red Inalámbrica PCI D-Link²⁹:

- ✓ IEEE802.11B/G 54MBPS
- ✓ Compatible con estándares IEEE 802.00b y 802.11g para red (LAN) inalámbrica de 2.4GHZ
- ✓ Compatible con PCI2.2 estándar
- ✓ Capaz de trabajar hasta 128-bit WEP, WPA, WPA2 802.1x
- ✓ Compatible con Windows 98/ME/2000/XP/Vista
- ✓ Bajo consumo de electricidad.

²⁸ Ver Imagen en Anexo B?

²⁹ Ver Imagen en Anexo B1

4.4 Hawkeye 8400V Series del grupo VERIS INDUSTRIES³⁰:



Figura 4.10 Hawkeye Series del Grupo Veris Industries. (Obtenida de Manual de Medidor Veris)

4.4.1 ESPECIFICACIONES GENERALES

Características Eléctricas:

- Tipo de medición Valor verdadero RMS hasta el 15vo armónico. Para Sistemas trifásicos (CA).
- Precisión de medición:
- Corriente y tensión ANSI C12.16, el 0,5%
- Potencia ANSI C12.16, 1%
- Frecuencia $\pm 0.1\text{Hz}$ de 45 a 65Hz
- Velocidad de actualización de datos 1s

Características de entradas de Tensión:

- Medición de tensión de 20 a 600VAC (directo L-L), y de 10 a 347VAC (directo L-N)
- Rango de medición +20%
- Impedancia de 2 M Ω (L-L) / 1M Ω (L-N)
- Rango de frecuencias 45 a 65 Hz

³⁰ Ver Anexo B5

Características de entradas de corriente:

- CT Escala Primaria: Ajustable de 5A a 32.767 A
- Rango de medición de entrada de 0 a 1.0VAC (+20% sobre el rango)
- Impedancia 100KΩ
- Control de Potencia AC: 100 a 415 ± 10% VAC, 5VA; DC: 125 a 250 ± 20 VDC, 3W
- Control de flujo de potencia a través del tiempo de 100 ms a 120VAC

➤ **Salidas:**

Pulso de salida # 1 H8453V/VB, H8463V/VB (KWH) N.O (Normalmente Abierto), salida estática (240VAC o 300VDC, 100mA máx. @ 25 ° C, disminución de 0.56mA (por cada 25 ° C) 2.41kV RMS de aislamiento.

- Pulso de salida # 2 H8453V/VB (Detecta fase ó kVARh) N.O, salida estática (240VAC o 300VDC, 100mA máx. @ 25 ° C. disminución de 0.56mA (por cada 25 °C) 2.41kV RMS de aislamiento.

- Pulso de salida # 2 H8463V/VB (Detecta Fase o kVARh) N.C (normalmente cerrados), salida estática (240VAC o 300VDC, 100mA máx. @ 25 ° C, disminución de 0.56mA (por cada 25 ° C) 2.41kV RMS de aislamiento.

- Puerto RS485; H8436V/VB; H8437V/VB 2-hilos, 2400 a 19200 baudios, Modbus RTU

Características Mecánicas:

- Peso H8400V = 0,81 libras; H8400VB = 6,35 libras; H8400VBS = 9,15 libras
Grado de protección IP (IEC 60529) IP40 pantalla frontal, medición de cuerpo IP30 para VB.
- Dimensiones: H8400V = 3.78 "x3.78" x1.97 " en la parte de atrás; H8400VB = 8.25" Wx9.5 "Hx4.08" D

Condiciones ambientales

- Temperatura de funcionamiento del medidor (0 ° C a 60 ° C), Pantalla (-10 ° C a 50 ° C)
- Medidor de Temperatura de almacenamiento y de pantalla (-40 ° C a 85 ° C)
- Rango de humedad 5 a 95% RH @ 50 ° C (sin condensación)

Categoría de medición:

- Norte América CAT III, para sistemas de distribución de hasta 347 V L-N /600VAC L-L
- CE CAT III, para sistemas de distribución de hasta 277 V L-N/480VAC L-L
Capacidad dieléctrica por UL 508, EN61010
- Conducción y emisiones irradiadas FCC parte 15 clase B, EN55011

Normas de Seguridad:

- Norte América UL508/C22.2
- Europa CE por IEC 61010

Características de pantalla de visualización:

- Dimensiones 73 x 69 mm de fondo con pantalla LCD verde (6 LÍNEAS TOTAL, 4 valores concurrentes)
* Para el control de tensiones > 415VAC se requiere H84xxVBS

4.5 Presupuesto

Los costos del proyecto se estiman en *costos indirectos*, los cuales involucran a los equipos e infraestructura utilizada, pero que pertenece a la entidad que permitió el desarrollo de la aplicación (Universidad Don Bosco); y *costos directos*, los cuales están relacionados con la inversión económica hecha por los interesados en desarrollar la aplicación (incluyendo materiales, equipo y las horas-hombre invertidas en diseño, desarrollo e implementación de la aplicación).

MATERIALES, EQUIPO Y MANO DE OBRA

Cotización: CSH S.A. de C.V.				
No.	Descripción	Unidades	Precio	Precio
			Unitario	Total
Equipo				
1	Analizadores de Redes CVMk	1	\$546.00	\$546.00
2	Convertidor Inteligente RS 232/485	1	\$392.61	\$392.61
3	PC con características mínimas	1	\$150.00	\$150.00
4	Tarjeta de Red Inalámbrica D- link	1	\$ 26.00	\$ 26.00
5	Software Power Studio	1	\$113.40	\$113.4
6	Módulos de Expansión CVM/RED	1	\$221.6	\$221.6
7	Cable para RS-485 p/metro	2	\$ 0.25	\$0.50
8	Cable para RS-232 p/metro	2	\$ 0.55	\$1.10
		Subtotal 1		\$1451.21
Costos Indirectos				
9	Horas-hombre invertidas en el diseño, desarrollo e implementación de la Aplicación	300	\$10.00	\$3000.00
		Subtotal 2		\$3000.00
10	Suministro e Instalación (15% de Subtotal 2)	1		\$ 483.5
11	Impuesto al valor agregado (IVA 13%)	1		\$419.02
<i>Costos Indirectos+Suministro+Instalación+Impuestos</i>			TOTAL	\$3902.52
GRAN TOTAL				\$5353.73

Tabla 4.8. Costos de la Aplicación.

Para una empresa en particular el costo de inversión en equipos y software puede variar, debido a que ésta puede tener parte del equipo instalado, o ningún equipo que

verifique el estado del sistema que quiere supervisarse. Por esta razón el costo de inversión dependerá de la cantidad de equipos que conformen la red, distancias entre los mismos, etc. Por lo tanto estos costos variarán con respecto a los costos presentados con anterioridad. Este presupuesto es sacado en base a un medidor Circutor instalado en la subestación que tiene planta de emergencia frente al Edificio 5 del CITT.

CAPÍTULO V. CARACTERÍSTICAS DEL SISTEMA IMPLEMENTADO

5.1 SISTEMA ELÉCTRICO A MONITOREAR

Para realizar el diseño se debe partir del sistema que será sujeto a monitoreo remoto de sus parámetros eléctricos. En este caso ese sistema es la subestación #2³⁰ del CITT.

La subestación #2 del CITT consta de un banco de transformadores monofásicos 23kV / 220V con una capacidad de 167kVA cada uno lo que quiere decir que la capacidad de la subestación #2 es de 501kVA con un sistema homogéneo en estrella³¹.

Esto quiere decir que la tensión de fase tiene una magnitud de 120V y la tensión de línea 220V, y que las corrientes de línea y de fase son las mismas. De los datos obtenidos se verifica que para el tiempo corto que se ha monitoreado la red de la subestación #2 los voltajes se mantienen dentro de su rango permitido, salvo algunas ocasiones como lo ocurrido el 10 de marzo de 2009³² que se tuvieron voltajes bastantes bajos en relación al voltaje nominal. Mediante un monitoreo constante se podrá detectar posteriormente si fue una eventualidad o si esta irregularidad se da con cierta frecuencia en el sistema eléctrico de esta subestación.

Es de gran importancia, como se mencionó anteriormente, conocer el tipo de cargas que maneja esta subestación, porque de la carga depende grandemente el comportamiento del sistema. La subestación #2 del CITT suministra energía eléctrica para computadoras, aires acondicionados, luminarias, tomas corriente, pulidora, esmeriladoras, soldadores eléctricos, tornos, fresadoras, taladros, sierras eléctricas, hornos.

³⁰ Ver Anexo A7

³¹ Ver Anexo A8

³² Ver Anexos C1.5 C1.6

A pesar de predominar el tipo de carga inductiva el factor de potencia promedio³³ se mantiene la mayor parte del tiempo en condiciones buenas para la subestación #2.

Para el valor de corriente que se maneja por fase, un valor entre 250 y 300A³⁴ se debe hacer uso de transformadores de instrumento para obtener las señales con el medidor de energía. Esto ocurre solo para el caso de la corriente.

Para el medidor de energía Circutor CVMk se cuenta con transformadores de corriente³⁵ con una regulación de 1200:5 correspondiéndole una relación de transformación de corriente (RTC) de 240. Esto quiere decir que si en promedio la subestación proporciona 275 amperios por fase, la señal de corriente que recibe el medidor de energía CVMk es de 1.15 amperios. Valor aceptable tomando en cuenta que los dispositivos de protección y de medición se diseñan para valores de señal de corriente de 5 amperios, esa es la razón por la que los transformadores de corriente en su mayoría, a excepción de los que proporcionan 1 amperio en el secundario, proporcionan un valor de corriente de 5 amperios en el secundario para un valor determinado máximo en el primario.

Para el medidor de industrias Veris modelo H8437, los transformadores de corriente que posee son de una relación de 800:5, para una relación de transformación de corriente de 160.

Los anteriores valores de RTC para los medidores de energía pueden ser cambiadas siempre teniendo en cuenta los valores de corriente que se tienen en el momento de hacer el cambio de transformadores de corriente. Quiere decir que para las condiciones actuales de la subestación se pueden usar transformadores de corriente de 500:5 (RTC = 100), pero se deja muy pequeño el margen tomando en cuenta que

³³ Ver Anexo C1.8

³⁴ Ver Anexo C1.7

³⁵ Ver Anexo A4

actualmente hay días que la demanda de la carga asociada a la subestación #2 alcanza los 350 amperios³⁶

En el caso de la tensión, no es necesario utilizar transformadores de potencial, porque el valor de la tensión que se toma como señal de entrada en los medidores de energía, no excede su valor máximo permitido³⁷ con lo cual pueden trabajar con normalidad sin necesidad de transformación de la señal de tensión.

A ambos medidores de energía debe programárseles las relaciones de transformación para obtener datos reales, de lo contrario mostrarán datos que no corresponden a valores reales. También, aunque no se usen transformadores de potencial, se deben programar los medidores con una relación de tensión de 1:1, si la tiene programada por defecto no es necesario. Pero siempre se debe verificar como se encuentra programado un determinado medidor de energía para obtener datos esperados.

5.2 INTERFAZ MEDIDOR DE ENERGÍA – ORDENADOR SERVIDOR³⁸.

Para esta parte se hicieron dos interfaces, las cuales convergen en el convertidor inteligente RS485/RS232³⁹ el cual permite la comunicación entre el protocolo Modbus del medidor de energía y el protocolo TCP/IP de la computadora donde se descargan datos de los parámetros eléctricos.

Para el medidor de energía Veris H8437 no es necesario utilizar el convertidor inteligente porque el programa de recolección de datos se desarrolló en el software Labview⁴⁰, el cual posee a característica que puede leer datos en protocolo Modbus utilizando ciertas herramientas que posee en su sistema. Situación que no sucede con el software PowerStudio.

³⁶ Ver Anexo C1.7

³⁷ Ver capítulo IV

³⁸ Ver capítulo III

³⁹ Ver Anexo B3

⁴⁰ Consultar el trabajo de graduación: ANÁLISIS TÉCNICO-ECONÓMICO DEL COMPORTAMIENTO DE VARIABLES ELÉCTRICAS QUE INCIDEN EN LA FACTURACIÓN Y CONSUMO DE ENERGÍA ELÉCTRICA EN EL CENTRO DE INVESTIGACIONES Y TRANSFERENCIA DE TECNOLOGÍA (CITT).

Para las interfaces, como se mencionó en el capítulo III, se utilizan únicamente tres conductores entre cada uno de los conectores DB9, que corresponden a la transmisión y recepción de datos, y el tercer conductor corresponde al tierra.

5.3 SOFTWARE UTILIZADO PARA EL PROCESAMIENTO DE DATOS.

Para cada uno de los medidores se utilizó un software distinto, esto debido a que cada medidor requiere un software específico para realizar el envío y recepción de datos, en el caso del medidor de energía Circutor CVMk se utilizó el software de la misma empresa Circutor denominado PowerStudio® versión 1.0. Y para el caso del medidor Veris se utilizó un software desarrollado en el ambiente de Labview® para la recolección de datos y un software desarrollado en Matlab® para la lectura remota de los parámetros eléctricos.

En este caso particular solo se abordarán el software PowerStudio® para el medidor de energía Circutor CVMk y el software Matlab® para el medidor de energía Veris H8437.

5.3.1 Software PowerStudio® para el Medidor CVMk

Al iniciar el software, sino se tienen dispositivos agregados anteriormente, el programa pregunta si se desea agregar nuevos dispositivos, al aceptar esta petición del programa, este despliega una lista de diferentes modelos de analizadores de redes Circutor, entre los cuales se encuentra el de interés particular para este trabajo realizado como se muestra en la figura 5.1.

Al seleccionar el medidor de energía con que se cuenta se da clic en siguiente para pasar a la ventana donde se asigna un nombre y una descripción al dispositivo seleccionado como se muestra en la figura 5.2.



Figura 5.1 Lista de dispositivos compatibles con el software PowerStudio versión 1.0.

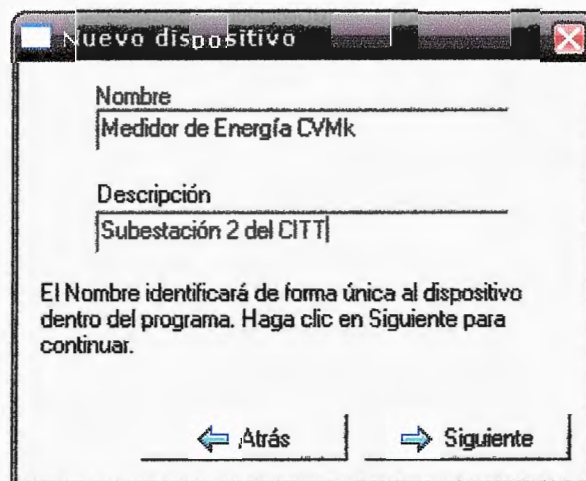


Figura 5.2 Asignación de nombre y descripción del medidor seleccionado

En el siguiente paso se debe seleccionar el tipo de comunicación que se utilizará para enlazar el ordenador donde se encuentra instalado el PowerStudio® y el analizador de redes Circutor. (Ver figura 5.3)

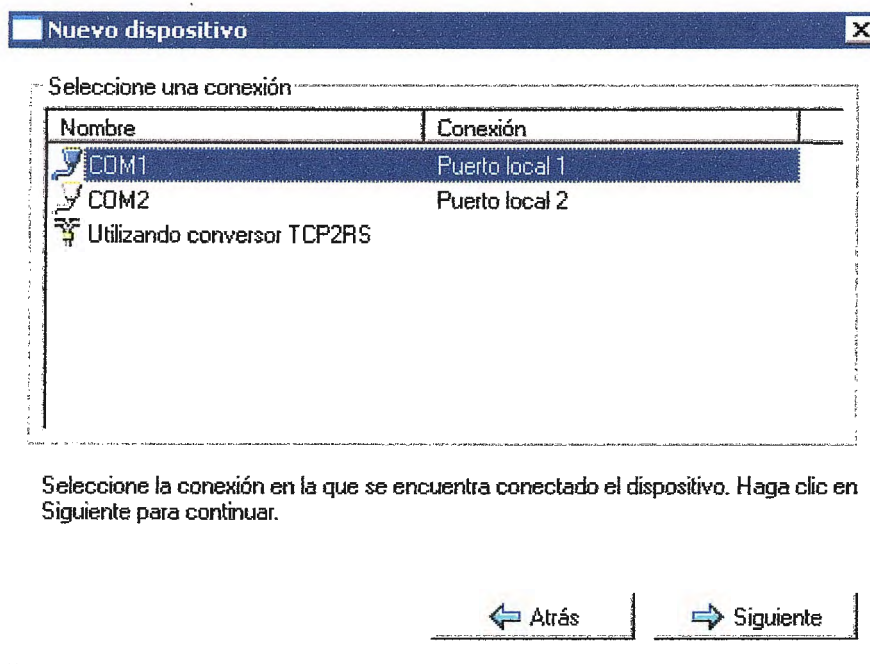


Figura 5.3 Selección de la conexión que se usa para la comunicación de datos.

El siguiente paso es asignar el número de periférico que corresponderá al medidor que se ha seleccionado. Recordemos que el protocolo Modbus me permite tener más de un medidor conectado a un servidor, por lo que es de suma importancia que estén bien definidos los números de periféricos de los dispositivos. El número de periférico a ser asignado debe coincidir con el número de periférico que se encuentra programado en cada uno de los medidores de energía marca Circutor. Para el caso particular de este estudio, se ha asignado al medidor de energía CVMk el número de periférico 001, que coincide con el número que tiene el medidor de energía instalado en la subestación #2 del CITT.

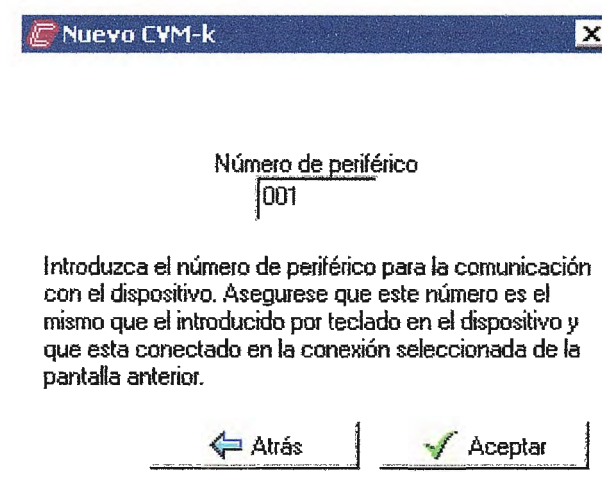


Figura 5.4 Asignación del número de periférico.

Al aceptar comienza la lectura o el enlace entre el ordenador y el medidor de energía hasta que se comprueba el medidor de energía cuando no hay errores en la comunicación y se despliega la pantalla principal del analizador de redes.

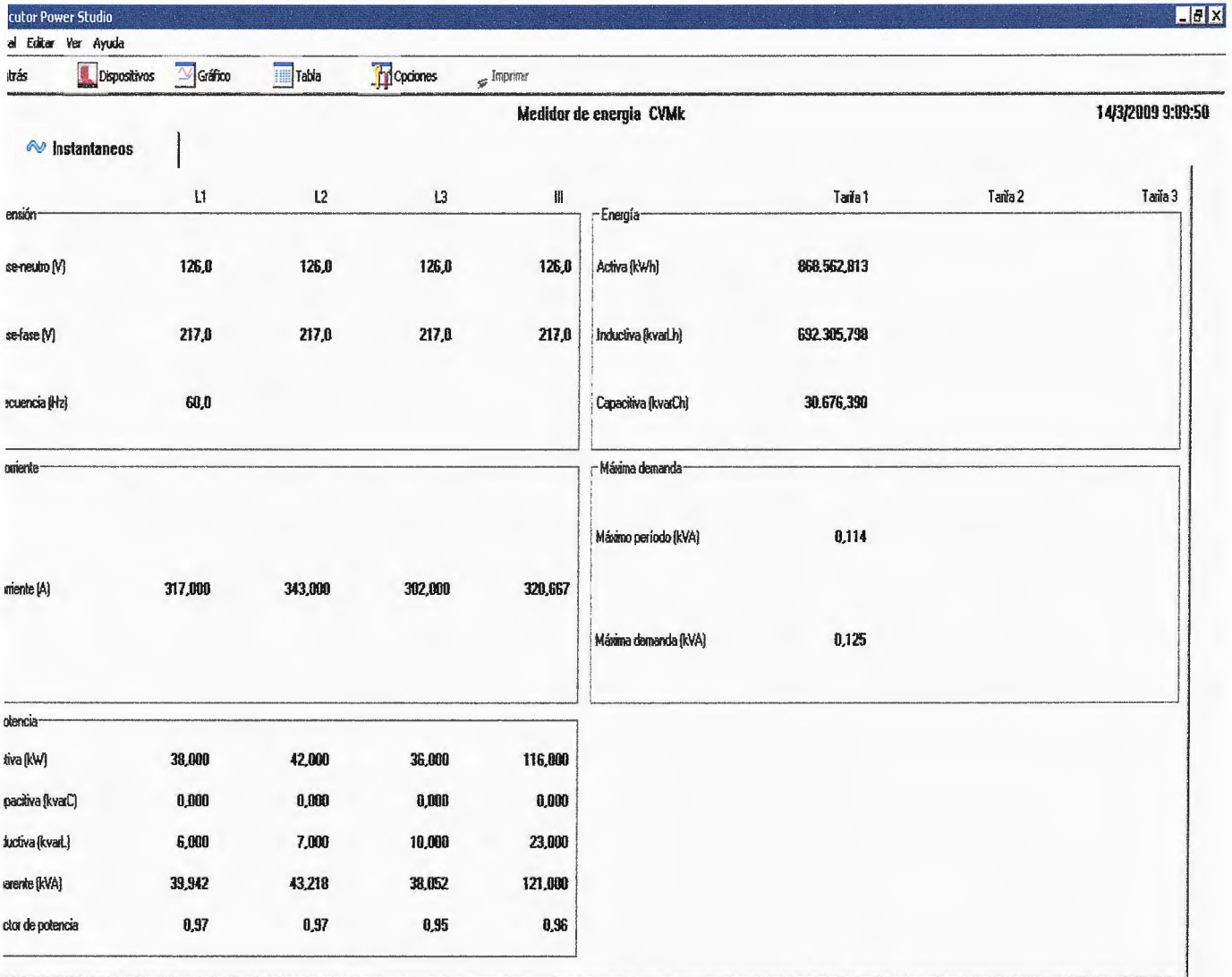


Figura 5.5 Pantalla principal en PowerStudio para el medidor de energía CVMk

Si la comunicación no se establece correctamente no se desplegará la pantalla principal, y en su lugar aparecerá un mensaje sombreado de color rojo diciendo que hubo errores de comunicación entre el dispositivo y el ordenador⁴¹.

Posibles causas de error de comunicación entre analizador de redes CVMk y el ordenador:

- Que no se haya instalado el convertidor RS485/RS232 queriendo asociar dos protocolos distintos.
- Que cada una de las interfaces expuestas en el capítulo III, se encuentren mal configuradas o conectadas.
- Que la tarjeta de expansión del medidor se encuentre defectuosa.
- Que el medidor de energía CVMk no se encuentre en operación al momento de tratar de hacer el enlace.
- Que el puerto COM de la computadora no se encuentre en óptimas condiciones.
- Que algún conductor de las interfaces se encuentre cortado internamente.}
- Que de la lista de analizadores de redes desplegados en la figura 5.1, se haya seleccionado un modelo de medidor de energía incorrecto.
- Que el número de periférico asignado en el PowerStudio sea diferente al que tenga programado el medidor de energía Circutor que se esté utilizando.
- Que la velocidad de comunicación no sea la misma entre el medidor de energía, el convertidor y el ordenador. En este caso se asignó 9600 baudios a cada uno.

La pantalla principal del software PowerStudio posee una segunda opción de visualización⁴² que depende del gusto del usuario.

⁴¹ Ver Anexo C1.1

⁴² Ver Anexo C1.4

Para graficar los parámetros eléctricos, se deben seleccionar las variables eléctricas que se desean graficar⁴³ depende de las variables eléctricas seleccionadas, así será la gráfica o las gráficas mostradas. Por ejemplo si se selecciona únicamente la energía kW-H del sistema, se mostrará únicamente la gráfica de la energía activa del sistema⁴⁴.

Se puede también seleccionar solo una variable eléctrica pero para cada una de las tres fases del sistema y también un valor promedio de las tres⁴⁵. Pero también se pueden mostrar diferentes variables eléctricas en una misma ventana, como por ejemplo potencia aparente y potencia activa, ambas trifásicas⁴⁶.

Se puede ver el comportamiento mediante gráficas para todas las variables eléctricas que puede medir el Circutor CVMk como por ejemplo frecuencia⁴⁷ y factor de potencia⁴⁸.

5.3.2 Software desarrollado en Matlab® para el Medidor Veris H8437.

El programa desarrollado en el ambiente Matlab® se convierte en una ayuda para facilitar la lectura remota de los parámetros eléctricos. Este no funciona directamente con el medidor de energía. La diferencia del software utilizado para la recolección de datos PowerStudio y el desarrollado en Labview para el Veris H8437, radica que este último crea un archivo de Excel⁴⁹ para ir guardando las variables eléctricas.

A partir de lo anterior se procedió a desarrollar el programa en Matlab para ayudar a la lectura remota de una forma sencilla de los parámetros eléctricos. Esto quiere decir que la aplicación desarrollada en Matlab permite el importar datos desde cualquier otro software que cree un archivo de Excel para ir almacenando diferentes parámetros.

⁴³ Ver anexo C1.2

⁴⁴ Ver anexo C1.3

⁴⁵ Ver anexos C1.5-C1.7

⁴⁶ Ver anexo C1.9

⁴⁷ Ver anexo C1.10

⁴⁸ Ver anexo C1.8

⁴⁹ Ver anexo C2.1

Para desarrollar el programa se utilizó la interfaz gráfica de Matlab® GUIDE, en el cual se hizo un programa para que el usuario interactúe gráficamente con el software.

En este se crean dos archivos. Un formulario donde se desarrolla la parte gráfica con extensión .fig y un archivo donde se encuentra el código que se establece al crear ese formulario y es en ese código donde se modifican a conveniencia los diferentes elementos creados en el formulario para llevar a cabo la interacción.

Pantalla inicial⁵⁰

Esta ventana muestra la presentación del programa, donde muestra para que es dicho software y que institución lo ha desarrollado y en qué año.

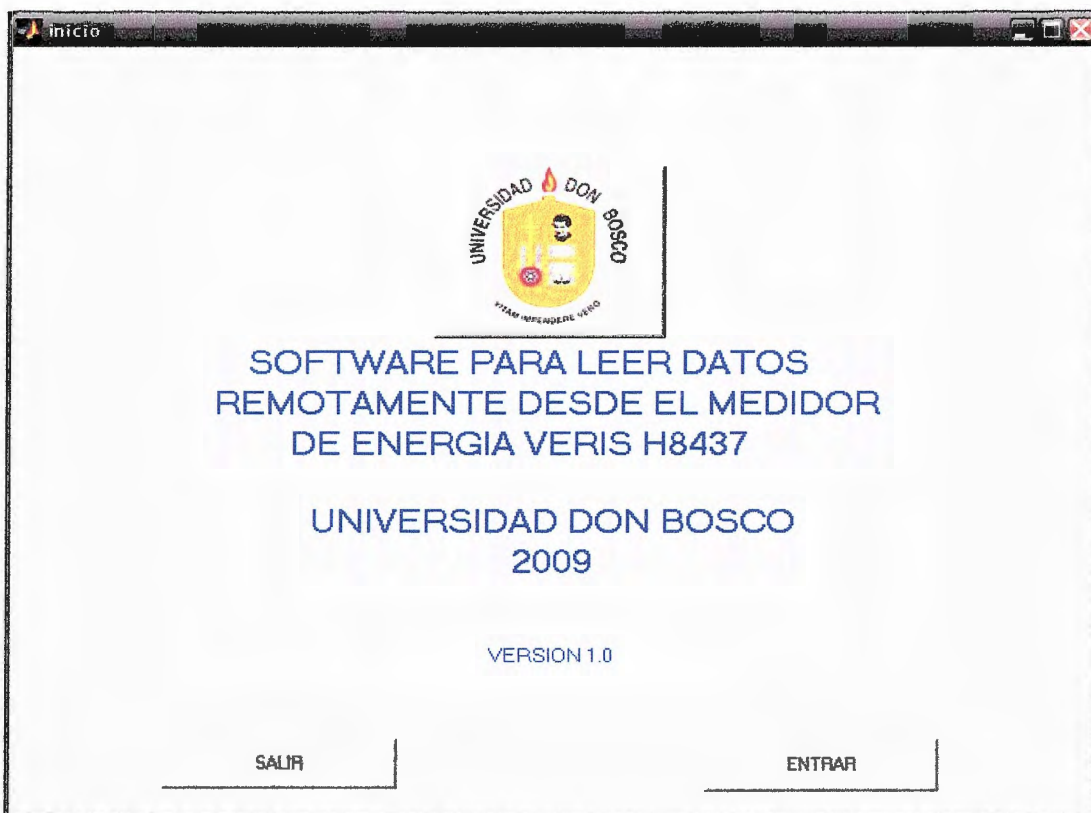


Figura 5.6 Pantalla de Inicio de Aplicación en Matlab

⁵⁰ Ver Código de Programa en Anexo C2.2

La ventana consta de dos botones pulsadores, *SALIR* y *ENTRAR*, el primero se utiliza para salir del programa si el usuario así lo deseara, y el segundo sirve para entrar a la pantalla principal del programa.

Para salir del programa se utiliza el comando *gcf* de Matlab (*get current figure*) que sirve para obtener el valor del formulario en el cuál se está trabajando en un momento dado. Acompañado del comando *gcf* se utiliza el comando *close* que se utiliza para cerrar una determinada aplicación. En el caso de del pulsador *SALIR* se utiliza la línea de código: *close (gcf)*, el cual cierra el formulario de trabajo llamado *inicio* para salir de la ejecución del programa.

Para entrar a la ventana principal se utiliza el pulsador *ENTRAR*, que la igual al caso anterior, utiliza los comandos *gcf* y *close* para cerrar la ventana abierta llamada *inicio* en este caso. La diferencia con respecto al caso anterior es que para este caso no basta con cerrar el formulario *inicio* porque si fuese el caso, el pulsador *ENTRAR* estaría cumpliendo exactamente la misma función que el pulsador *SALIR*; razón por la cual la línea de código para el pulsador *ENTRAR* es:

```
close(gcf)
principal
```

En donde *principal* hace referencia al nombre del archivo que se quiere ejecutar. En este caso se utiliza para abrir el formulario correspondiente a la pantalla principal del programa.

Pantalla Principal⁵¹

En esta ventana se presentan las principales variables eléctricas que son: corrientes por fase, voltaje promedio de línea y de fase, potencia activa trifásica, factor de potencia promedio y la energía activa acumulada.

Para que aparezcan los valores de las variables eléctricas mencionadas en el párrafo anterior, necesita ejecutarse el código en la parte de función de apertura del archivo

⁵¹ Ver código de programa en anexo C2.3

.m correspondiente a este formulario *principal*, de lo contrario los cuadros de texto aparecerían en blanco.

La diferencia con el programa en Labview y el software PowerStudio es que no se muestran los parámetros eléctricos variando constantemente. Para ver estos cambios en cada parámetro eléctrico visualizado en pantalla basta con dar clic sobre el pulsador *Actualizar* que se encuentra en cada una de las ventanas del programa a excepción de la pantalla inicial.

Cuando se ejecuta el código fuente del pulsador *Actualizar* lo que se hace es sencillamente reiniciar el formulario y al volverse a cargar, éste siempre lee la última fila de datos que se guarda en el archivo de Microsoft Excel que para este caso se llama Sub.xls, archivo en el cual se descargan datos periódicamente mediante el programa desarrollado en Labview. Para la base de datos en Excel denominada Sub.xls se han guardado datos en periodos de diez minutos.

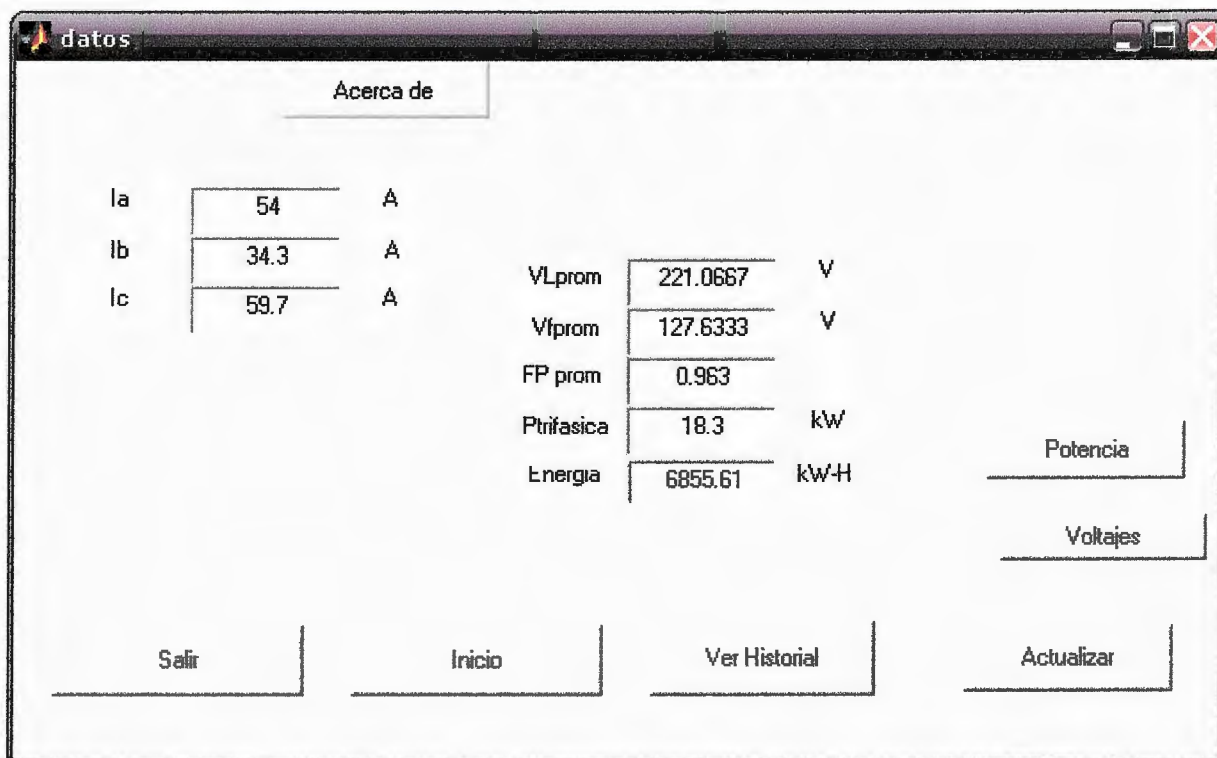


Figura 5.7 Pantalla Principal

Los datos que se obtienen de la base de datos de Excel poseen un formato numérico, por lo que es innecesario convertir de cadena de caracteres a formato numérico para realizar operaciones algebraicas dentro del código fuente en el archivo con extensión .m, pero si es necesario convertir de formato numérico a formato de cadena de caracteres para mostrar datos en los cuadros de texto.

Como cada cuadro de texto en Matlab tiene asignado un determinado valor numérico que se utiliza como manejador de un cuadro de texto específico, es necesario utilizar este valor de manejador dependiendo del cuadro de texto donde se quiere mostrar un determinado dato. Al tener cada cuadro de texto su propio valor de manejador, así se basa el código para hacer referencia a un determinado cuadro de texto. El valor de un determinado manejador de un cuadro de texto se obtiene con el siguiente código: *handles.edit1*. Para transformar de un formato de cadena de caracteres a un formato numérico se utiliza el comando *str2num* (*string to number*) y para convertir de número a cadena de caracteres es *num2str* (*number to string*).

Cabe preguntarse como hace el programa para saber que tiene que leer la última fila guardada en el archivo Sub.xls. Para ello se ejecuta la línea de código siguiente:

```
matriz=xlsread('Sub');
```

```
ax=size(matriz);
```

En donde el comando *xlsread* se utiliza para leer la primera hoja de cálculo del archivo de Microsoft Excel especificado en este caso con el nombre Sub. De esta manera la matriz de datos de la hoja de cálculo se guarda en la variable *matriz*. Luego se procede a asignar en la variable *ax* el tamaño de la matriz de datos mediante el comando *size*, el cual devuelve un vector fila con dos datos; el primero corresponde al número total de filas que posee la hoja de cálculo donde se encuentran guardados los datos descargados del medidor de energía, y el segundo dato, es el número total de columnas de dicha hoja de cálculo.

Con lo anterior se garantiza que los datos que leerá el programa son los últimos datos que se han guardado en la hoja de cálculo.

El historial de datos se puede visualizar mediante el pulsador *Ver Historial*, el cual utiliza la línea de código: *winopen('Sub.xls)*, en donde el comando *winopen* se utiliza en Matlab para abrir aplicaciones de Microsoft Windows, como por ejemplo programas del paquete de Microsoft Office como Excel, Word, etc.

Pantalla de Tensiones⁵²

En esta ventana del programa se muestran cada uno de las tensiones de fase del sistema y su respectivo valor promedio. Al igual que las tensiones de línea acompañadas por el valor promedio de las tensiones de línea.

Cabe resaltar que debido a la conexión estrella – estrella del sistema éstos deben ser diferentes en una relación de raíz de tres.

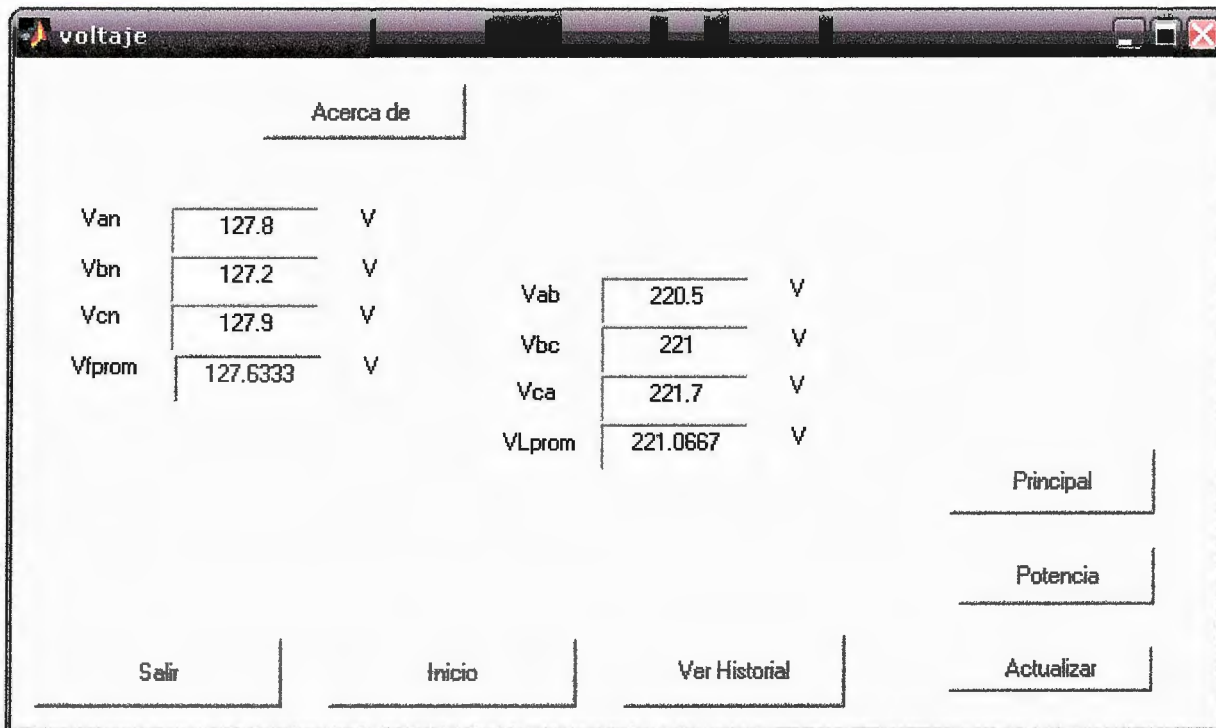


Figura 5.8 Pantalla de Tensiones

⁵² Ver código de programa en anexo C2.4

Cada uno de los pulsadores que se encuentran a la derecha en cada una de las ventanas del programa en Matlab, se utilizan para abrir las ventanas a cuyo nombre hacen referencia. Para ello se utiliza una línea de código similar a la que se ha utilizado en el pulsador *ENTRAR* de la ventana inicial, con la diferencia que después del código *close(gcf)* debe colocarse el nombre del formulario que se desea abrir.

Pantalla de potencias y factor de potencia⁵³.

En esta pantalla se muestra la potencia activa y reactiva que consume cada una de las tres fases del sistema eléctrico que alimenta la subestación eléctrica #2 del CITT. Además, se muestran los valores de potencia trifásica, tanto activa como reactiva.

Otro punto de gran importancia que se muestra en esta ventana es el factor de potencia para cada una de las tres fases del sistema y el valor del factor de potencia promedio del mismo.

El factor de potencia es importante mantenerlo alto para que la universidad no se vea involucrada en penalizaciones por parte del distribuidor de energía eléctrica.

Cada una de las ventanas que forman parte del programa tienen un botón pulsador llamado *Ver historial* que sirve para acceder al historial de la base de datos que en este caso particular es una hoja de cálculo de Microsoft Excel como se mencionó anteriormente.

⁵³ Ver código de programa en anexo C2.5

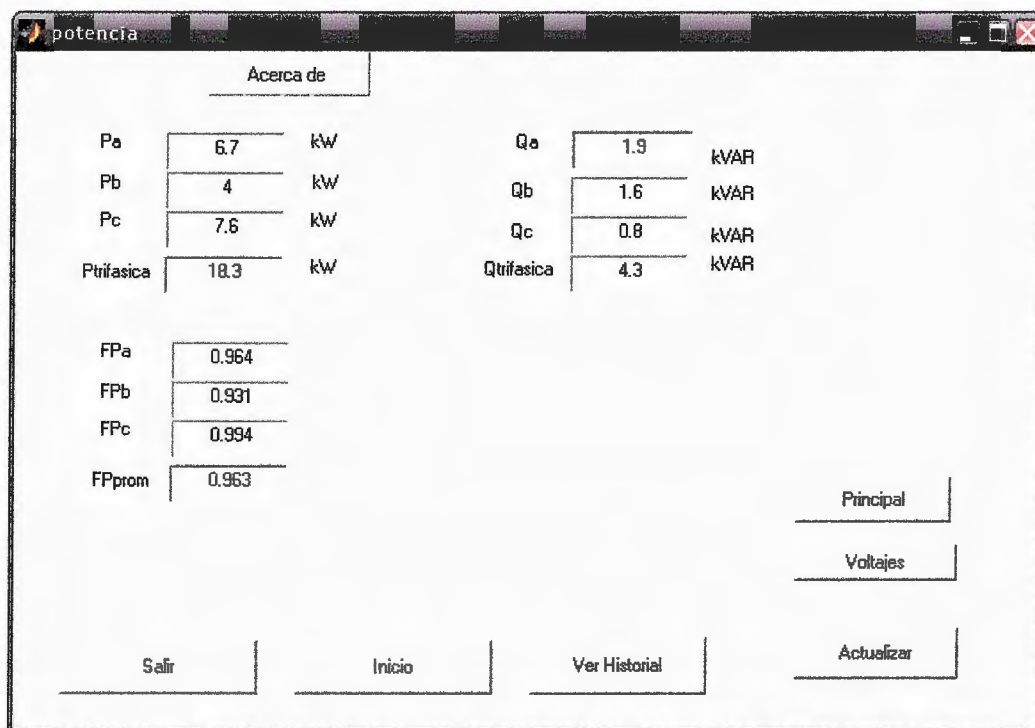


Figura 5.9 Pantalla de Potencias y Factor de Potencias

Como se puede apreciar en cada una de las pantallas del programa, a excepción de la ventana *inicio* se encuentran pulsadores comunes como *Salir*, *Inicio*, *Ver Historial*, *Acerca*, *Actualizar* que cumplen funciones específicas en el sistema. Por ejemplo, siempre que se pulse la opción *Salir* se dejará de ejecutar el programa y se cerrarán las ventanas del mismo. Siempre que se pulse *Ver Historial* se abrirá el archivo de Microsoft Excel donde se han descargado los datos y, siempre que se accione el pulsador *Actualizar*, se actualizarán los datos que se tengan en pantalla dependiendo si se está en la ventana *principal*, *tensiones* o *potencia*.

5.4 COMUNICACIÓN DE DATOS

5.4.1 Generalidades.

Cuando se desarrolla un esquema de direccionamiento de IP para una institución, la consideración principal es si la intranet está conectada a Internet.

Si la institución no está conectada a Internet, es técnicamente posible elegir cualquier ID de red de IP, con clase o sin clase, sin tener en cuenta el solapamiento de direcciones con otras que se usan en Internet. Sin embargo, es recomendable elegir un rango de direcciones privadas.

Para las instituciones conectadas a Internet, deberían elegir entre el uso de direcciones públicas o privadas.

Direcciones públicas

InterNIC⁵⁴ asigna las direcciones públicas dentro del espacio público de direcciones que consiste en todas las posibles direcciones unicast (desde un único transmisor a un único receptor) en la Internet mundial. Históricamente, InterNIC⁵⁵ asignaba ID de red con clase a las organizaciones conectadas a Internet sin tener en cuenta su ubicación geográfica. Hoy en día InterNIC asigna bloques CIDR a los ISP según su ubicación geográfica. Los ISP subdividen los bloques CIDR asignados entre sus clientes. La subdivisión del resto del espacio de direcciones de clase C según su ubicación geográfica se realiza para disponer de un enrutamiento jerárquico. El objetivo era minimizar el número de rutas en los enrutadores troncales de Internet. Las direcciones públicas se garantiza que son únicas globalmente.

Cuando a una organización o a un ISP se le asigna un bloque de direcciones del espacio público de direcciones, existe una ruta en las tablas de enrutamiento de los enrutadores de Internet por la que las direcciones públicas asignadas se pueden

⁵⁴ *Internet's Network Information Center*

⁵⁵ Encaminamiento Inter-Dominios sin Clases, por su siglas en inglés

alcanzar a través del ISP. Históricamente, se añadía a todos los enrutadores de Internet un ID de red con clase. En la actualidad, se añade a las tablas de enrutamiento de los enrutadores de Internet e ISP regionales una ruta que consiste en un rango de direcciones asignadas.

El rango de direcciones de IP públicas asignadas a una organización se resume en uno o más pares (ID de red, máscara). Estos pares se convierten en rutas en los enrutadores de Internet y de los ISP de forma que se pueda alcanzar las direcciones de IP de la organización.

Direcciones solapadas o ilegales

Las organizaciones que no están conectadas a Internet directa ni indirectamente son libres de elegir cualquier esquema de direccionamiento sin tener en cuenta las direcciones que tienen asignadas otras organizaciones a otros proveedores de servicio de Internet. Sin embargo, si la organización decide, más tarde, conectarse a Internet, se requiere implantar un nuevo esquema de direccionamiento.

Las direcciones asignadas cuando la organización no estaba conectada a Internet podrían incluir direcciones públicas que InterNIC hubiese asignado a otras organizaciones o ISP⁵⁶. En este caso, esas direcciones están duplicadas, en conflicto con las direcciones asignadas. Se conoce como direccionamiento solapado, o ilegal.

El tráfico de Internet desde los hosts que usan direcciones ilegales se reenvía a los enrutadores de la organización cuyas direcciones se asignaron originalmente. Por tanto, las organizaciones que usan direcciones ilegales no se encuentran alcanzables en Internet.

Por ejemplo, una organización que no está conectada a Internet decide usar el espacio de direcciones 207.46.130.0/24 para su intranet. Mientras la organización no se conecte a Internet el uso de 207.46.130.0/24 no representa ningún problema. Si la organización se conecta a Internet con una ruta directa, el uso de 207.46.130.0/24 es

⁵⁶ Proveedor de servicio de Internet, por sus siglas en inglés

ilegal y no se reciben respuestas de los hosts en el segmento de red 207.46.130.0/24.

Con esta configuración, cuando un host envía tráfico a un sitio en Internet, envía el tráfico con la dirección de IP de origen dentro del espacio de direcciones de 207.46.130.0/24. Cuando un host de Internet responde, envía la respuesta a la dirección de IP de destino dentro del espacio de direcciones de 207.46.130.0/24. InterNIC asignó este espacio de direcciones a Microsoft Corporation y, por tanto, existe una ruta en los enrutadores de Internet para reenviar el tráfico con direcciones de IP de destino en este rango a los enrutadores de Microsoft Corporation. Por tanto, las respuestas al tráfico enviado por los hosts con el espacio de direcciones ilegal 207.46.130.0/24 se reenvían a los enrutadores de Microsoft Corporation, y no a los enrutadores de la organización con las direcciones ilegales.

Es una práctica común entre los proveedores de servicios de Internet descartar los paquetes de IP que envía un sitio de un cliente cuando el campo de dirección de IP de origen no es una dirección pública válida asignada al cliente. Esto previene el envío de tráfico desde hosts que usen direcciones ilegales y suplantación de direcciones. La suplantación de direcciones es el envío de tráfico de IP desde una dirección de IP de origen que no está asignada a un host.

Direcciones privadas

Con el crecimiento exponencial de Internet, la demanda de direcciones de IP públicas creció exponencialmente. Como cada nodo de la intranet de una organización requiere una dirección de IP pública única y global, las organizaciones solicitaban de InterNIC suficientes direcciones de IP para asignar direcciones únicas a todos los nodos de sus organizaciones.

Sin embargo, cuando se realizaba un análisis de las direcciones de IP dentro de las organizaciones, las autoridades de Internet se dieron cuenta de que la mayoría de las organizaciones actuales necesitan muy pocas direcciones públicas. Los únicos hosts que requieren direcciones de IP públicas son los que se comunican

directamente con los sistemas de Internet. Ejemplos de ellos son los servidores Web, los servidores de FTP⁵⁷, los servidores de correo, los servidores proxy y los servidores de seguridad. La mayoría de los hosts de la intranet de una organización acceden a los recursos de Internet a través de pasarelas del Nivel de Aplicación como servidores proxy y servidores de correo.

Para los hosts de la intranet de la organización que no requieren acceso directo a Internet no es necesario usar un espacio de direcciones de IP legal. Para este propósito las autoridades de Internet crearon el espacio privado de direcciones, un subconjunto del espacio de direcciones de IP de Internet que se puede usar sin conflicto con otra organización, para los hosts que no requieren una conexión directa a Internet.

Los espacios de direcciones público y privado están separados y no se superponen. InterNIC nunca asigna direcciones privadas -las direcciones dentro del espacio privado de direcciones- a ninguna organización ni ISP. Esto significa que las direcciones de IP privadas no son alcanzables en Internet.

Como las direcciones privadas no son alcanzables, los hosts de una intranet con direccionamiento privado no pueden conectarse directamente a Internet. En lugar de ello, los hosts de una intranet con direccionamiento privado deben conectarse indirectamente a Internet usando un traductor de direcciones de red o una pasarela del Nivel de Aplicación como un servidor Proxy, caso de la universidad Don Bosco.

Un traductor de direcciones de red es un enrutador que traduce entre direcciones privadas y direcciones públicas para el tráfico de Internet. El servidor proxy recibe una petición de un host en la intranet de recursos en Internet. El servidor proxy envía entonces una petición del recurso en Internet y el tráfico de respuesta se reenvía al host solicitante. Cuando el servidor proxy envía la petición por el recurso de Internet, usa su dirección pública. Tanto los servidores proxy como los traductores de

⁵⁷ *File Transfer Protocol*

direcciones de red tienen direcciones privadas en su interfaz con la intranet y direcciones públicas en su interfaz con Internet.

El espacio de direcciones privadas se define en los siguientes bloques de direcciones:

10.0.0.0/8. La red privada 10.0.0.0/8 es un espacio de direcciones de 24 bits que se puede usar para cualquier esquema de creación de subredes dentro de una organización privada.

172.16.0.0/12. La red privada 172.16.0.0/12 es un espacio de direcciones con 20 bits para el host que se puede usar para cualquier esquema de creación de subredes dentro de una organización privada. Desde la perspectiva de clases, la red privada 172.16.0.0/12 es un ID de red de un rango de 16 ID de red de clase B desde la 172.16.0.0/16 hasta la 172.31.0.0/16.

192.168.0.0/16. La red privada 192.168.0.0/16 es un espacio de direcciones con 16 bits para el host que se puede usar para cualquier esquema de creación de subredes dentro de una organización privada. Desde la perspectiva de clases, la red privada 192.168.0.0/16 es el rango de 256 ID de red de clase C desde la 192.168.0.0/24 hasta la 192.168.255.0/24.

Para una mejor organización en el reparto de rangos las redes se han agrupado en cuatro clases, de manera que según el tamaño de la red se optará por un tipo u otro.

Las direcciones de clase A

Corresponden a redes que pueden direccionar hasta 16.777.214 máquinas cada una.

Las direcciones de red de clase A tienen siempre el primer bit a 0.

0 + Red (7 bits) + Máquina (24 bits)

Solo existen 124 direcciones de red de clase A.

Ejemplo:

	Red	Máquina		
Binario	0 0001010	00001111	00010000	00001011
Decimal	10	15	16	11

Rangos (notación decimal):

1.xxx.xxx.xxx - 126.xxx.xxx.xxx

Las direcciones de clase B

Las direcciones de red de clase B permiten direccionar 65.534 máquinas cada una.

Los dos primeros bits de una dirección de red de clase B son siempre 01.

01 + Red (14 bits) + Máquina (16 bits)

Existen 16.382 direcciones de red de clase B.

Ejemplo:

	Red	Máquina		
Binario	01 000001	00001010	00000010	00000011
Decimal	129	10	2	3

Rangos (notación decimal) :

128.001.xxx.xxx - 191.254.xxx.xxx

Las direcciones de clase C

Las direcciones de clase C permiten direccionar 254 máquinas.

Las direcciones de clase C empiezan con los bits 110

110 + Red (21 bits) + Máquina (8 bits)

Existen 2.097.152 direcciones de red de clase C.

Ejemplo:

	Red	Máquina		
Binario	110 01010	00001111	00010111	00001011
Decimal	202	15	23	11

Rangos (notación decimal):

192.000.001.xxx - 223.255.254..xxx

Las direcciones de clase D

Las direcciones de clase D son un grupo especial que se utiliza para dirigirse a grupos de máquinas. Estas direcciones son muy poco utilizadas. Los cuatro primeros bits de una dirección de clase D son 1110.

En el caso de la red utilizada para enlazar el ordenador ubicado en la subestación #2 del CITT y las computadoras pertenecientes a la red del edificio #4, se utilizó una red privada clase A cuya dirección IP asignada (provisional) es 10.0.19.10, dirección mediante la cual se puede acceder a los documentos compartidos de dicha máquina.

Para la comunicación entre la computadora ubicada en la subestación #2 del CITT y los dispositivos de comunicación del edificio 5, se requirió de la instalación de un punto de acceso inalámbrico⁵⁸ en el edificio 5 para lograr tener un buen alcance y por ende la adecuada señal para la tarjeta inalámbrica⁵⁹ del ordenador ubicado en el cuarto de subestaciones #2.

⁵⁸ Ver capítulo IV

⁵⁹ Ver capítulo IV

Esa parte se instaló recientemente para poder enlazar y añadir a la red de la universidad el ordenador de la subestación #2. El punto de acceso inalámbrico utiliza una red DDE⁶⁰, al cual también se le da el nombre de esa red “DDE” y a ésta se le ha asignado una clave de acceso por el personal de Soporte Técnico de datos de la universidad.

El punto de acceso instalado provisionalmente tiene una capacidad de 100Mbps, sin embargo, el switch que está conectado a dicho punto de acceso, es el de una subred del switch principal del edificio 5 y, solo posee una capacidad de 10Mbps, limitando de esta manera al punto de acceso inalámbrico instalado actualmente.

La distribución de la red de datos entre los edificios 4, 5 y 6 se muestra en la figura 5.5. El medio de comunicación utilizado es fibra óptica. En el edificio 6 se recibe la señal principal y de ahí se distribuye por medio de la fibra óptica a cada uno de los edificios que conforman el centro de investigación y transferencia de tecnología.

Como se muestra en la figura 5.10 el edificio 5 se eligió para la instalación del punto de acceso inalámbrico dada su cercanía con el cuarto de subestación #2.

Actualmente se espera la compra de un punto de acceso DWL3200ap que será instalado en el edificio 6, lugar donde quedará permanentemente y se eliminará la instalación provisional del edificio 5.

La aprobación de la compra se logró gracias a las gestiones hechas por el jefe del departamento de eléctrica, para el cual este trabajo representa un gran interés.

⁶⁰ Ver anexo B7

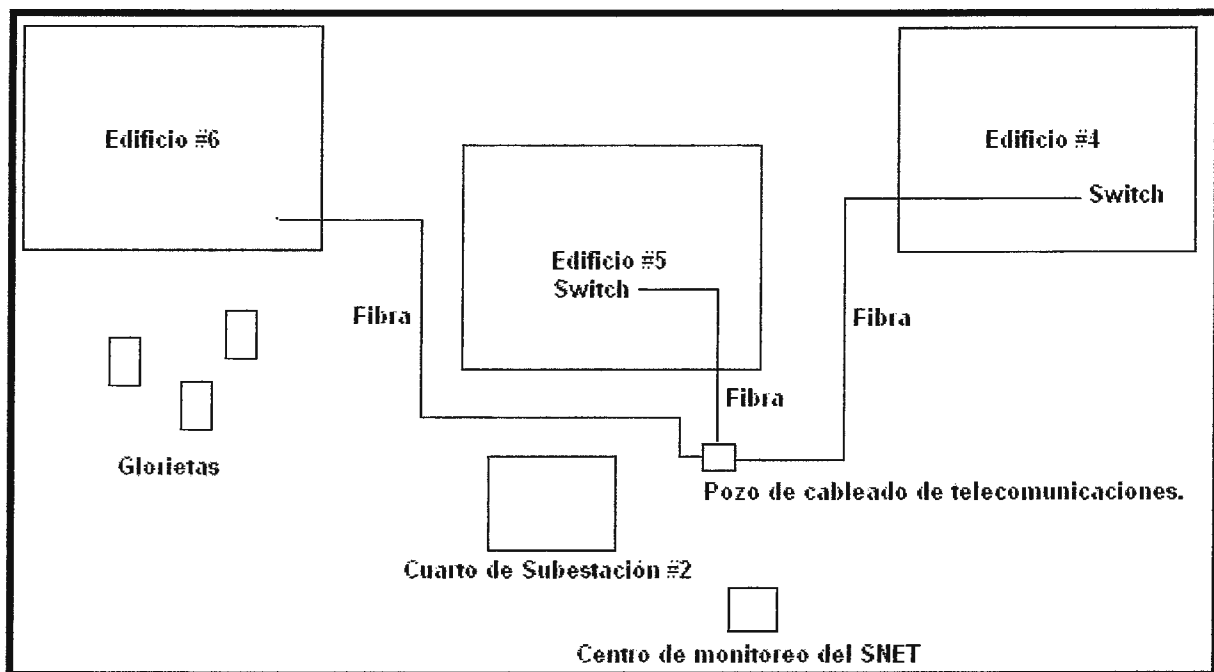


Figura 5.10 Ubicación de los edificios involucrados y el medio de comunicación utilizado.

5.4.2 Forma de Acceso Remoto.

Para la utilización del software PowerStudio, se utilizó la herramienta de Windows *Conexión a Escritorio Remoto (Remote Desktop Connection)* con el cual se puede acceder a monitorear los parámetros eléctricos en cualquier momento, o bien, para analizar el comportamiento de un determinado parámetro eléctrico para un determinado periodo de toma de datos.

Para utilizar esta herramienta de Escritorio Remoto se debe dar permiso desde la computadora que se encuentra en la subestación. Debido a que esta se puso en red con las máquinas del edificio 4, se decidió asignar una clave de acceso para que no cualquiera pueda entrar a la aplicación.

Para lograr acceder a la máquina que se encuentra en la subestación #2 se necesita la dirección IP de la computadora o bien el nombre de la misma, y la clave de acceso asignada a dicha computadora como se muestra en la figura 5.11.



Figura 5.11 Acceso al escritorio remoto de la computadora con dirección IP 10.0.19.10

Al dar clic en conectar, se ingresará al escritorio de la máquina cuya dirección IP se quiere acceder. Para ello como se mencionó anteriormente, las máquinas deben estar en red. Si la computadora a la cual se quiere acceder remotamente no se la ha asignado una clave de usuario, no habrá ningún problema. Pero si se la ha asignado una clave de usuario, para acceder se deberá introducir la respectiva clave de acceso.

Para el caso de la aplicación en Matlab se puede hacer lo mismo que para la aplicación del PowerStudio, utilizando una conexión de escritorio remoto. Pero para evitar que tanto el PowerStudio como el Matlab se ejecuten en la misma máquina, lo que se ha hecho es utilizar la carpeta de Documentos Compartidos de la computadora que se encuentra en la subestación #2 para crear una subcarpeta y guardar ahí tanto el archivo de Excel como los ejecutables de Matlab para que no ocurra un error con el *Path* de Matlab. Por tal razón deben ponerse en la misma carpeta.

5.5 DIAGRAMA DE BLOQUES DEL SISTEMA IMPLEMENTADO.

La figura 5.12 muestra el diagrama de bloques del sistema implementado. El medidor de energía, mediante los transformadores de corriente, obtiene la señal de corriente del sistema, y haciendo uso de pinzas obtiene la señal de tensión de las barras en el gabinete principal perteneciente a la subestación #2 del CITT.

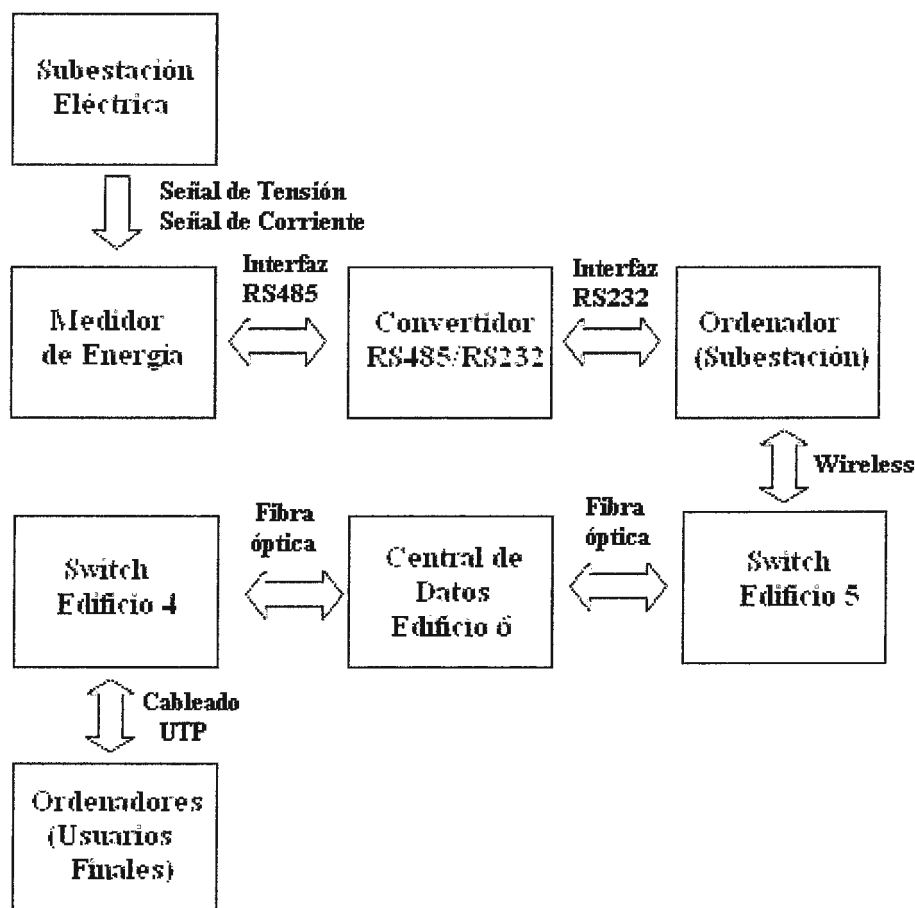


Figura 5.12. Diagramas de bloques del sistema

El protocolo Modbus del analizador de redes utiliza una interfaz RS485, que por medio del convertidor RS485/RS232 se logra convertir la señal y mediante una interfaz RS232 se conecta al ordenador que es donde se encuentra instalado el software para la recepción y almacenamiento de datos que obtiene el medidor de energía del sistema eléctrico de la subestación #2.

Para llevar a cabo el enlace de la computadora que se utiliza como servidor con la red de datos de la universidad Don Bosco, se hace vía *wireless* utilizando una tarjeta de red inalámbrica D-Link la cual se enlaza con un punto de acceso inalámbrico conectado al switch del edificio 5 mediante el cual se puede llegar a la red de datos cableada de la universidad.

Del switch del edificio 5 se conecta al switch principal en la central de datos del edificio 6, utilizando como medio de transporte de datos un cableado de fibra óptica, que a la vez es utilizado para enlazar cada uno de los edificios del CITT individualmente con el edificio 6. Por lo tanto del switch principal se hace el enlace con el switch del edificio 4, también mediante fibra óptica.

La fibra óptica se utiliza para optimizar el envío y recepción de datos, puesto que para distancias mayores a cien metros no es recomendable utilizar cableado UTP porque la caída de tensión conlleva pérdida de información, recordando que entre mayor es la distancia, mayor es la caída de tensión en un conductor dado.

El switch del edificio 4 distribuye la señal de datos internamente a las computadoras de dicho edificio. Debido a que la computadora que se encuentra en la subestación #2 se ha enlazado con toda la subred del switch del edificio 4, quiere decir que cualquier ordenador perteneciente a esa subred tiene acceso al ordenador de la subestación mediante la herramienta *Conexión a Escritorio Remoto* de Windows, razón por la cual ha sido asignada una contraseña para poder ingresar a la computadora que se encuentra en la subestación #2 del CITT. La contraseña establecida se maneja internamente en el Departamento de Eléctrica, usuario principal del sistema implementado.

5.6 ESTUDIO DE FACTIBILIDAD TÉCNICO Y ECONÓMICO PARA UN SISTEMA DE MEDICIÓN REMOTO EN VARIOS PUNTOS.

La universidad Don Bosco cuenta con tres subestaciones eléctricas de 501kVA en el Centro de Investigación y Transferencia de Tecnología. Actualmente se ha instalado el sistema de medición remota solamente en una de esas tres subestaciones.

Es necesario poder monitorear constantemente las tres subestaciones para verificar la calidad de energía servida por la empresa distribuidora y a la vez para identificar oportunidades de ahorro energético. En el poco tiempo que hasta la fecha ha estado en operación la computadora en la subestación #2, ha habido variaciones de tensión tal que, la misma computadora se ha apagado como consecuencia de ello, en especial a causa de los bajones de tensión que se producen en el sistema. Además ha habido cortes de energía que ha sufrido el sistema a lo largo del último mes de marzo.

Teniendo un registro de los parámetros eléctricos y de las variaciones en la calidad de la energía eléctrica se puede exigir a la distribuidora por el mal servicio y así evitar el daño de equipos que en algunas veces resultan ser costosos para la universidad.

Por lo anteriormente expuesto es necesario tener un sistema de medición continuo, no necesariamente remoto, pero como se ha demostrado con la instalación de un sistema de medición remoto solamente par a una subestación, la diferencia en la instalación de un sistema continuo de monitorización de las variables eléctricas de un sistema a un sistema de monitorización remoto en este caso se reduce solamente a una tarjeta de red inalámbrica D-Link tomando en cuenta que se puede enlazar con la red de datos de la universidad en los lugares donde se encuentran la subestaciones 1 y 3 del CITT.

La tabla 4.8 que muestra los costos de una instalación como la implementada para este trabajo de graduación, muestra costos para una empresa cualquiera que quiera llevar a cabo la implementación de un sistema de medición remoto. En el caso de la universidad Don Bosco esos costos se pueden reducir considerablemente tomando en cuenta que posee mano de obra calificada y cuenta con equipos como los que se utilizaron en la implementación del sistema de medición remoto, como por ejemplo computadoras que cumplen con las características mínimas, interfaces para crear las interfaces RS232 y RS485, tarjetas inalámbricas D-Link, analizadores de red CVMk, una tarjeta extra de expansión CVM/RED.

La universidad Don Bosco puede llevar a cabo una inversión tomando en cuenta que únicamente necesitaría comprar una tarjeta de expansión CVM/RED y dos convertidores inteligentes RS485/RS232 teniendo una suma de \$1006.82 en base a la tabla 5.1.

No.	Descripción	Cantidad	Precio unitario	Precio total
1	Convertidor Inteligente RS 232/485	2	\$392.61	\$785.22
2	Módulos de Expansión CVM/RED	1	\$221.6	\$221.6
			Total	\$1006.62

Tabla 5.1 Costos de Inversión

Queda denotada una gran diferencia con respecto a los costos de la tabla 4.8, mostrando la factibilidad que tendría la universidad de instalar sistemas de medición remotos de los parámetros eléctricos para las subestaciones 1 y 3.

Para otra empresa constituida en el rubro industrial, sería más factible afrontar los gastos de instalación y de equipo de un sistema de medición remota porque este tipo de empresas cuentan con un mayor presupuesto económico. En cambio, el presupuesto de la institución universitaria es bastante limitado.

Sin embargo, dependiendo de la industria y del tamaño de una determinada empresa, convendrá instalar ya sea un sistema inalámbrico o un sistema cableado.

Para poder utilizar un software centralizado se necesitan puertos de expansión ETHERNET y convertidores Serial-Ethernet para poder llevar los datos a la red de la universidad, sin embargo la factibilidad de este tipo de sistemas de medición con software centralizado se ve afectado porque debe implementarse mediando cableado UTP. Sin embargo, las distancias entre la subestación #3 del aula magna C y la subestación #2 del CITT, no son extremadamente grandes y en este caso se puede usar comunicación vía ETHERNET con cableado UTP aprovechando las características del software PowerStudio que permite monitorear más de un dispositivo mediante el protocolo Modbus.

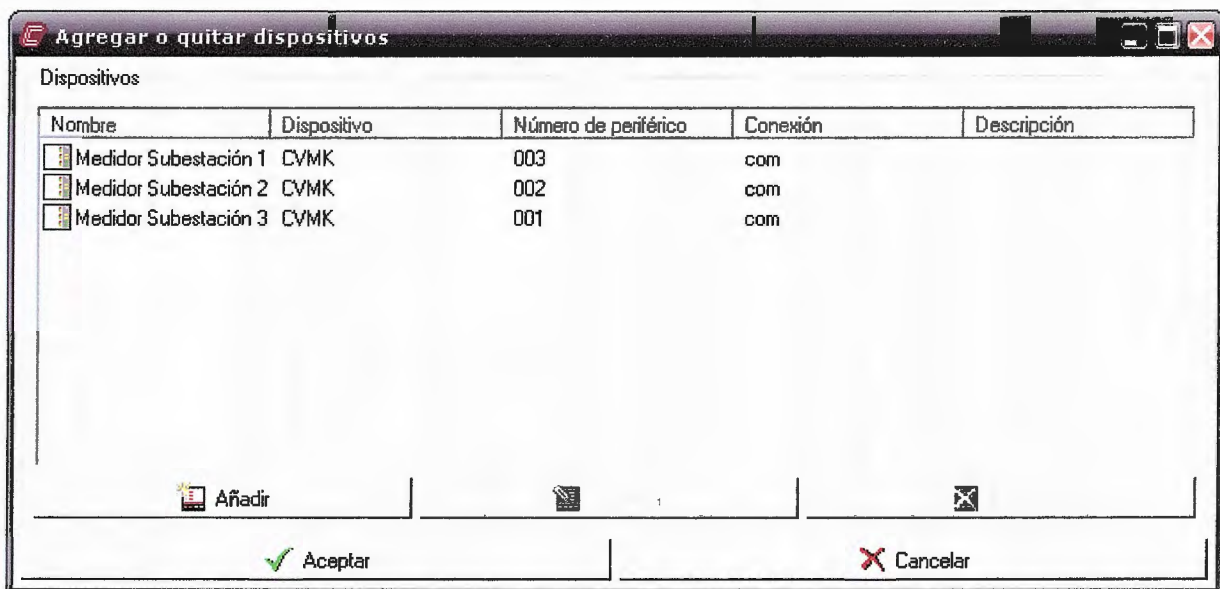


Figura 5.13 Dispositivos agregados a PowerStudio como software centralizado.

En la figura 5.13 se muestran agregados tres medidores de energía modelos CVMk que bien pudieran ser los tres medidores de energía correspondientes a las tres subestaciones del Centro de Investigación y Transferencia de Tecnología de la universidad Don Bosco.

Por lo antes descrito resulta factible a la universidad implementar otros dos sistemas de medición remota para las subestaciones restantes del CITT, para monitorear los parámetros eléctricos del sistema de energía.

Otro punto que se puede monitorear es el grupo electrógeno que se utiliza como planta de emergencia en el cuarto de subestación #2, para verificar la calidad de suministro de energía eléctrica que brinda y siempre mantener el sistema y los equipos pertenecientes al sistema en buen estado.

La factibilidad se ve reflejada en la inversión que habría de hacer la universidad para implementar los sistemas de medición remota, alrededor de los mil dólares se necesitan para llevar a cabo este proyecto y así garantizar un buen servicio de suministro eléctrico exigiéndole éste a la distribuidora de energía eléctrica.

6. CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- Se logró realizar la medición remota desde la subestación #2 del CITT hasta el centro de computo del laboratorio de potencia del edificio #4 por lo que se pudo supervisar el comportamiento de los parámetros eléctricos durante el período que se mantuvo conectado el sistema de medición remota en la subestación con lo que queda demostrado que si es posible implementar este tipo de sistema en las instalaciones eléctricas de la universidad.
- Para la universidad el trabajo realizado representará una gran ayuda al departamento de eléctrica, tanto para el mantenimiento o para realizar proyecciones sobre la red eléctrica interna de la universidad; puesto que se monitorea en tiempo real el comportamiento de los parámetros eléctricos de interés en un punto determinado como lo es la subestación No. 2.
- Se ha demostrado que la aplicación desarrollada en este trabajo de graduación representa una gran ayuda, no solamente para la universidad Don Bosco, sino también en el ámbito empresarial para generar una equidad y una concordancia entre los clientes y las distribuidoras para evitar problemas legales entre ellos mismos por diferentes motivos relacionados a la facturación del consumo de energía eléctrica.
- Se demostró la factibilidad de un sistema como el implementado, tomando en cuenta el mejor y fácil control sobre el comportamiento de los parámetros eléctricos en la universidad Don Bosco o cualquier otra empresa para realizar el análisis de sus respectivos sistemas eléctricos de potencia.

- Tomando en cuenta lo mencionado anteriormente, podría tener un visto bueno por parte de las empresas, porque a partir de éste se pueden detectar oportunidades de ahorro energético y por ende, ahorro económico y este sistema a la vez sirve como respaldo para la verificación de la calidad de energía que suministra la empresa distribuidora de energía eléctrica.
- Se comprobó que el sistema de medición utilizado actualmente en la universidad no es el óptimo, pero también se demostró que se puede mejorar mediante la implementación de aplicaciones como la desarrollada en este trabajo para la subestación No.2, para cargas más puntualizadas en la red eléctrica de la universidad.
- Se detectaron debilidades al sistema implementado, parte de ellas debido a la provisionalidad del sistema de medición remota, como las variaciones de tensión a las que se somete el ordenador ubicado en el cuarto de subestación No.2, por lo que se incluyen las respectivas recomendaciones en el siguiente apartado.

6.2 RECOMENDACIONES.

- Debido a la baja tensión que se produce cuando el grupo electrógeno utilizado como planta de emergencia ubicado en el cuarto de subestación #2, se pone en funcionamiento una vez a la semana como parte del mantenimiento y verificación de buen estado de esta planta, se recomienda la instalación de un UPS para mantener en operación y en buen estado el ordenador ubicado en ese cuarto para que el monitoreo de los parámetros eléctricos no sea interrumpido.
- Se necesita explotar mejor los medidores de energía instalados actualmente en las tres subestaciones del CITT. Una manera de lograrlo es instalar por lo menos un ordenador para tener un historial de datos de los parámetros eléctricos de cada uno de los subsistemas eléctricos de la universidad Don Bosco para tener un monitoreo de los mismo y de esta forma, poder hacer una planificación de la red interna de la universidad y mantener márgenes apropiados para la capacidad nominal de las subestaciones y de los conductores eléctricos y no llegar a una sobrecarga de la red.
- Al hacer futuras compras de medidores de energía como los que se tienen en las tres subestaciones del CITT, comprar un paquete que incluya la tarjeta de expansión y el convertidor inteligente RS485/RS232 para poder llevar a cabo proyectos como el desarrollado en la subestación #2 del CITT. Porque si posteriormente a la compra del medidor de energía se quiere hacer la compra de los otros dispositivos complementarios, éstos tienen un costo mayor que al comprarse en paquete. Actualmente solo se posee un paquete completo.

- Fomentar el desarrollo de la medición remota en las dos subestaciones faltantes para que al igual que en la subestación #2, se tenga un monitoreo constante del consumo de energía eléctrica de la universidad Don Bosco.
- Utilizar este tipo de trabajos para temas como calidad de energía y ahorro energético. Teniendo historiales de los parámetros eléctricos se puede detectar mediante análisis si a la universidad o cualquier otra institución se le cumple con la calidad del servicio de energía eléctrica.
- Por lo general en un sistema eléctrico es conocido que los aires acondicionados implican una gran demanda de energía. En el edificio 6 del CITT los salones casi en su totalidad tienen equipos de aire acondicionado instalados, lo cual representa una cifra significativa de la demanda energética del CITT. Para reducir esta carga de energía una medida con los aires acondicionados sería normar la temperatura mínima a 20°C, recordemos que entre más baja es la temperatura más tiene que exigírsele al compresor para llegar a esa temperatura. Y es que la temperatura a 20°C es bastante razonable adecuada tanto para las personas como para las computadoras que se encuentran en los centros de cómputo del edificio 6.
- Se puede ahorrar el uso de una computadora en cada subestación mediante la compra de un dispositivo denominado TCP2RS, el cual permite el enlace entre un medidor de energía Circutor y un HUB de una red de datos. De esta manera para el monitoreo remoto de los parámetros eléctricos se necesitaría solamente de una computadora la cual estaría en el punto de acceso remoto a la lectura de estas variables eléctricas. Este se puede incluir también en el paquete a la hora de comprar los dispositivos.
- Se hizo a pequeña escala un sistema que puede ser implementado tanto por un consumidor para respaldar el cobro que hace una compañía distribuidora de energía eléctrica, como para la misma empresa

distribuidora para realizar la facturación de energía para un determinado cliente. Para ello, la empresa distribuidora de energía eléctrica tendría que contratar los servicios de una empresa de telecomunicaciones para que administre la red de datos para la compañía de distribución de electricidad. A las personas que al final de mes toman la lectura de los medidores de energía, se les podría capacitar para que sean ellos los que analicen el consumo energético mensual de los clientes de la distribuidora.

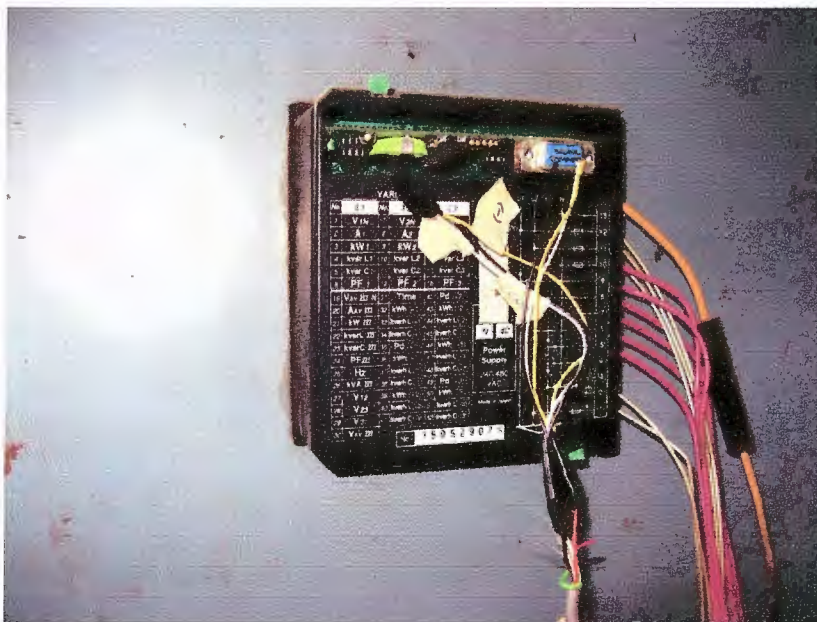
- El trabajo realizado en la subestación #2 abre el camino a nuevas oportunidades de mejora en lo concerniente a la parte eléctrica del CITT; detectando horas picos para los 7 días de la semana para poder establecer nuevos horarios de trabajo para poder generar un ahorro de la energía eléctrica y por ende un ahorro económico para la institución educativa.
- El trabajo hecho a pequeña escala permite la visualización de lo que podría ser un proyecto a gran escala para una empresa distribuidora de energía eléctrica para llevar a cabo la facturación de energía a sus clientes. La información la pueden recaudar desde un centro de operación que recibiría la señal de los medidores de energía que ellos instalen con el propósito de monitorización remota.

ANEXOS

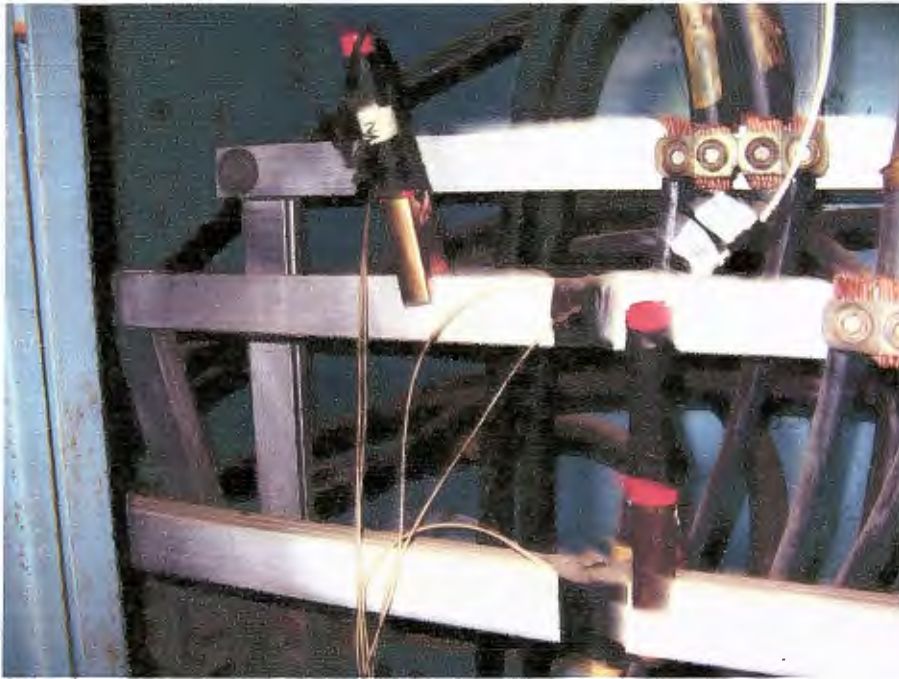
Anexo A. FOTOGRAFIAS EN CUARTO DE SUBESTACIÓN #2 DEL CITT.



A1. Medidor Circutor CVMk empotrado en gabinete.



A2. Medidor Circutor CVMk en gabinete visto desde atrás.



A3. Vista de Barras dentro del panel



A4. Vista de los Transformadores de corriente (Donas) dentro del panel



A5. Vista del Convertidor RS323/RS485 i



A6. Vista del CPU utilizado como servidor en el cual se puede observar la antena de la tarjeta de Red inalámbrica.



A7. Subestación 501 KVA a la cual el medidor CVMk censa sus datos



A8. Conexión Y-Y de la Subestación #2 del CITT.



A9. Vista de la distribución de espacios en el cuarto de Subestación #2



A10. Cuarto de Subestación #2 frente al Edificio 5 del CITT

Anexo B. DISPOSITIVOS UTILIZADOS



B1. Tarjeta Inalámbrica D-Link instalada en ordenador ubicado en subestación #2.



B2. Dispositivo DWL 3200 AP utilizado como punto de acceso inalámbrico.

DESCRIPCIÓN

- Permiten pasar de una comunicación RS-232 a una RS-485 y viceversa.
- Puertos RS-232 y RS-485 optocoplados.
- Dispositivos transparentes al sistema.

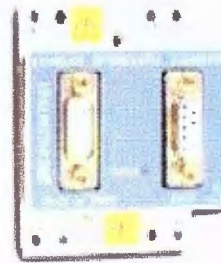
CARACTERÍSTICAS TÉCNICAS

Alimentación:
230 V c.a.

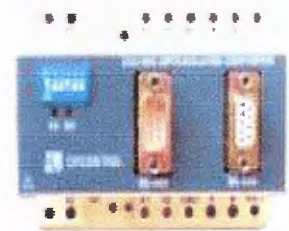
Comunicación RS-232:
Conector DB-9 hembra

Comunicación RS-485:
Conector DB-9 macho

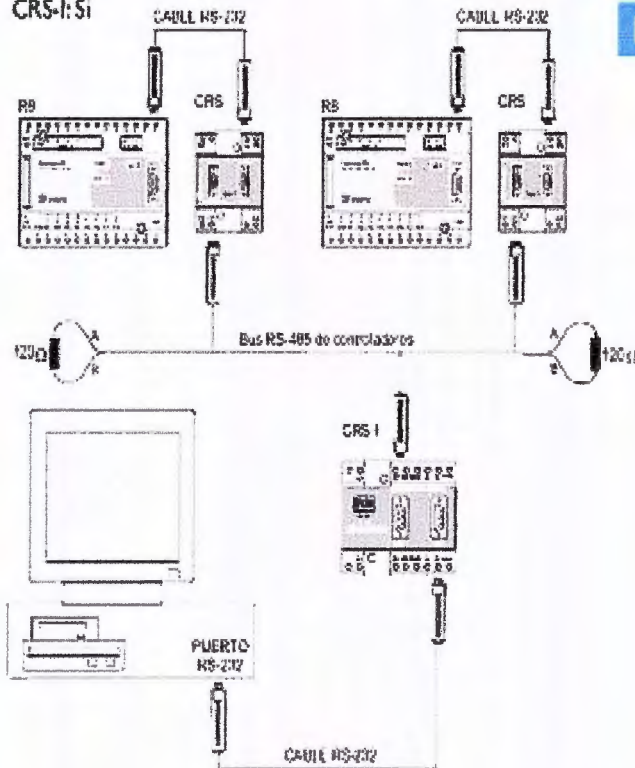
Control RTS:
CRS: No
CRS-I: Si



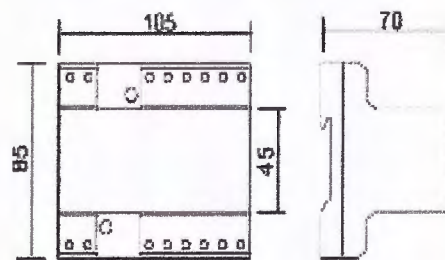
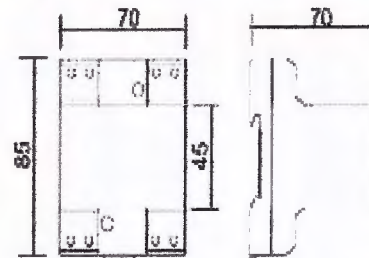
CRS



CRS-I



DIMENSIONES



B3. Características del convertidor RS485/RS232 CRSI

B4. Registro de las variables eléctricas en el CVMk

VARIABLE	REGISTERS		VARIABLE	REGISTERS	
	DECIMAL	HEXA-DECIMAL		DECIMAL	HEXA-DECIMAL
DATE / HOUR *NOTE 1	0 - 1	00 - 01		38 - 39	26 - 27
V 1	2 - 3	02 - 03	Vav III N	40 - 41	28 - 29
mA 1	4 - 5	04 - 05	mAv III	42 - 43	2A - 2B
W 1	6 - 7	06 - 07	W III	44 - 45	2C - 2D
varL 1	8 - 9	08 - 09	varL III	46 - 47	2E - 2F
varC 1	10 - 11	0A - 0B	varC III	48 - 49	30 - 31
PF 1 (x100)	12 - 13	0C - 0D	PF III 1 (x100)	50 - 51	32 - 33
V 2	14 - 15	0E - 0F	Hz (x10)		
mA 2	16 - 17	10 - 11	VA III	52 - 53	34 - 35
W 2	18 - 19	12 - 13	V 12	54 - 55	36 - 37
varL 2	20 - 21	14 - 15	V 23	56 - 57	38 - 39
varC 2	22 - 23	16 - 17	V 31	58 - 59	3A - 3B
PF 2 1 (x100)	24 - 25	18 - 19	Vav III	60 - 61	3C - 3D
V 3	26 - 27	1A - 1B	+ Wh - TARIFF 1	62 - 63	3E - 3F
mA 3	28 - 29	1C - 1D	+ varh L -TAR 1	64 - 65	40 - 41
W 3	30 - 31	1E - 1F	+ varh C -TAR 1	66 - 67	42 - 43
varL 3	32 - 33	20 - 21	Pd (last period)	68 - 69	44 - 45
varC 3	34 - 35	22 - 23	- Wh TARIFF 1	70 - 71	46 - 47
PF 3 1 (x100)	36 - 37	24 - 25	- varh L TAR 1	72 - 73	48 - 49
			- varh C TAR 1	74 - 75	4A - 4B

VARIABLE	REGISTERS		VARIABLE	REGISTERS	
	DECIMAL	HEXA-DECIMAL		DECIMAL	HEXA-DECIMAL
THD V 1	84 - 85	54 - 55	mA 1	114 - 115	72 - 73
THD V 2	86 - 87	56 - 57	mA 2	116 - 117	74 - 75
THD V 3	88 - 89	58 - 59	mA 3	118 - 119	76 - 77
THD I 1	90 - 91	5A - 5B	W 1	120 - 121	78 - 79
THD I 2	92 - 93	5C - 5D	W 2	122 - 123	7A - 7B
THD I 3	94 - 95	5E - 5F	W 3	124 - 125	7C - 7D
DATE / HOUR *NOTE 1	100 - 101	64 - 65	varL 1	126 - 127	7E - 7F
V 12	102 - 103	66 - 67	varL 2	128 - 129	80 - 81
V 23	104 - 105	68 - 69	varL 3	130 - 131	82 - 83
V 31	106 - 107	6A - 6B	+ Wh - TARIFF 1	132 - 133	84 - 85
V 1	108 - 109	6C - 6D	+ varh L -TAR 1	134 - 135	86 - 87
V 2	110 - 111	6E - 6F	- Wh TARIFF 1	136 - 137	88 - 89
V 3	112 - 113	70 - 71	- varh L TAR 1	138 - 139	8A - 8B

B5. Características del Medidor de Energía Veris H8437

Model	H8436X/37x	H8453x	H8463x
CT Primary Ratio	100A	100A	100A
CT Secondary Ratio (A models only)	5A	5A	5A
PT Ratio	1:1 (none)	1:1 (none)	1:1 (none)
Frequency	60Hz	60Hz	60Hz
Setup Password	00000	00000	00000
Reset Password	00000	00000	00000
Display Mode	IEEE	IEEE	IEEE
System Type	40	40	40
Modbus Address	001	N/A	N/A
Modbus Baud Rate	19200 baud	N/A	N/A
Modbus Parity	None	N/A	N/A
1st Pulse Contact	N/A	N.O.	N.O.
KWh Pulse Width	N/A	100ms	100ms
KWh Pulse Energy Index	N/A	1 KWh/Pulse	1KWh/Pulse
2nd Pulse Contact	N/A	N.O.	N.C.
2nd Pulse Contact Output	N/A	KVARh	Phase loss
KVARh Pulse Width	N/A	100ms	100ms
KVARh Pulse Energy Index	N/A	1 KVARh/Pulse	1 KVARh

B6. Registros de variables eléctricas en medidor Veris H8437

H8400 REG.	R/W	INV	FORMAT	UNITS	Scale Factor/ Mult.	RANGE	DESCRIPTION
Energy Statistics							
3-Phase Summary							
• 001	R	INV	ULong	kWh	E	0-0xFFFF	Real Energy Consumption (LSW)
• 002	R	INV	ULong	kWh	E	0-0xFFFF	Real Energy Consumption (MSW)
• 003	R		UInt	kW	W	0-32767	Total Instantaneous Real Power (3 Phase Total)
DS --							
• 004	R		UInt	kVA	W	0-32767	Total Instantaneous Apparent Power (3 Phase Total)
• 005	R		UInt	kVAR	W	0-32767	Total Instantaneous Reactive Power (3 Phase Total)
• 006	R		UInt	Ratio	0.0001	0-10000	Total Power Factor (Total kW / Total kVA)
• 007	R		UInt	Volt	V	0-32767	Voltage, L-L, Average of 3 Phases
• 008	R		UInt	Volt	V	0-32767	Voltage, L-N, Average of 3 Phases
• 009	R		UInt	Amp	I	0-32767	Current, Average of 3 Phases
Per Phase							
• 010	R		UInt	kW	W	0-32767	Real Power, Phase A
• 011	R		UInt	kW	W	0-32767	Real Power, Phase B
• 012	R		UInt	kW	W	0-32767	Real Power, Phase C
• 013	R		UInt	Ratio	.0001	0-10000	Power Factor, Phase A
• 014	R		UInt	Ratio	.0001	0-10000	Power Factor, Phase B
• 015	R		UInt	Ratio	.0001	0-10000	Power Factor, Phase C
• 016	R		UInt	Volt	V	0-32767	Voltage, Phase A-B
• 017	R		UInt	Volt	V	0-32767	Voltage, Phase B-C
• 018	R		UInt	Volt	V	0-32767	Voltage, Phase A-C
• 019	R		UInt	Volt	V	0-32767	Voltage, Phase A-N
• 020	R		UInt	Volt	V	0-32767	Voltage, Phase B-N
• 021	R		UInt	Volt	V	0-32767	Voltage, Phase C-N
• 022	R		UInt	Amp	I	0-32767	Current, Instantaneous, Phase A
• 023	R		UInt	Amp	I	0-32767	Current, Instantaneous, Phase B
• 024	R		UInt	Amp	I	0-32767	Current, Instantaneous, Phase C
DS --							
• 025	R		UInt	Amp	I	0-32767	Current, Instantaneous, Neutral
• 026	R		UInt	Hz	0.01	4500-6500	Frequency (derived from Phase A)
• 027	R	INV	UInt	kW	W	0-32767	Total Real, Power Minimum
• 028	R	INV	UInt	kW	W	0-32767	Total Real, Power Maximum
Accumulated Energy							
• 029	R	INV	ULong	KVAh	E	0-0xFFFF	Apparent Energy Consumption (LSW)
• 030	R	INV	ULong	KVAh	E	0-0xFFFF	Apparent Energy Consumption (MSW)
• 031	R	INV	ULong	KVARh	E	0-0xFFFF	Reactive Energy Consumption (LSW)
• 032	R	INV	ULong	KVARh	E	0-0xFFFF	Reactive Energy Consumption (MSW)

B7. DDE - INTERCAMBIO DINÁMICO DE DATOS.

Cualquier aplicación basada en Windows que posea funciones de DDE puede establecer vínculos de Intercambio Dinámico de Datos con PowerStudio.

El DDE (Dynamic Data Exchange) es un protocolo de Windows entre aplicaciones. Se puede actuar en modo cliente (Solicitar datos a otras aplicaciones) y/o modo servidor (Suministrar datos a otras aplicaciones). El programa PowerStudio actúa en modo servidor, es decir, una vez establecida una comunicación con otra aplicación

Windows (Excel, Paradox, Word, Programas en C, en Visual Basic, etc.) PowerStudio le suministra a esta, los valores leídos en los drivers. Así, por ejemplo, se puede estar viendo en tiempo real una tensión, corriente, etc. en Excel. Aquellas aplicaciones que permiten forzar un valor a través del DDE podrán modificar un parámetro del driver, como por ejemplo salidas digitales, registros internos, etc.

La comunicación DDE que se establece tiene las siguientes características:

- El vínculo que se establece es del tipo "*enlace en caliente*". Es decir, PowerStudio avisa a las aplicaciones clientes cuando los datos han cambiado y éstas los solicitan si es necesario.
- El programa puede enviar datos vía DDE a varias aplicaciones, la limitación está en la memoria del PC.
- Para cada cliente se establece un vínculo por cada una de las variables solicitadas.
- El protocolo DDE consta básicamente de tres elementos: la ***Conversation***, el ***topic***, y el ***item***.

Conversation

El *Conversation* es un texto que distinguen entre si las aplicaciones, tiene que ser único por aplicación. Esto permite que no se crucen preguntas y respuestas de aplicaciones distintas. En nuestro caso es:

"PWSTDValues". Por ejemplo "=PWSTDValues|Dispositivo1|V11".

Topic

El *Topic*, al igual que el *Item*, no tiene un significado concreto, depende del uso que le dé cada aplicación. En el caso del programa PowerStudio, el *Topic* identifica el dispositivo del que se desea obtener la información.

Por ejemplo "=PWSTDValues|Dispositivo1|V11". Cuando el nombre del dispositivo contenga espacios (Dispositivo 1), se deberá indicar entre comillas simple (') "=PWSTDValues|'Dispositivo 1'|V11"

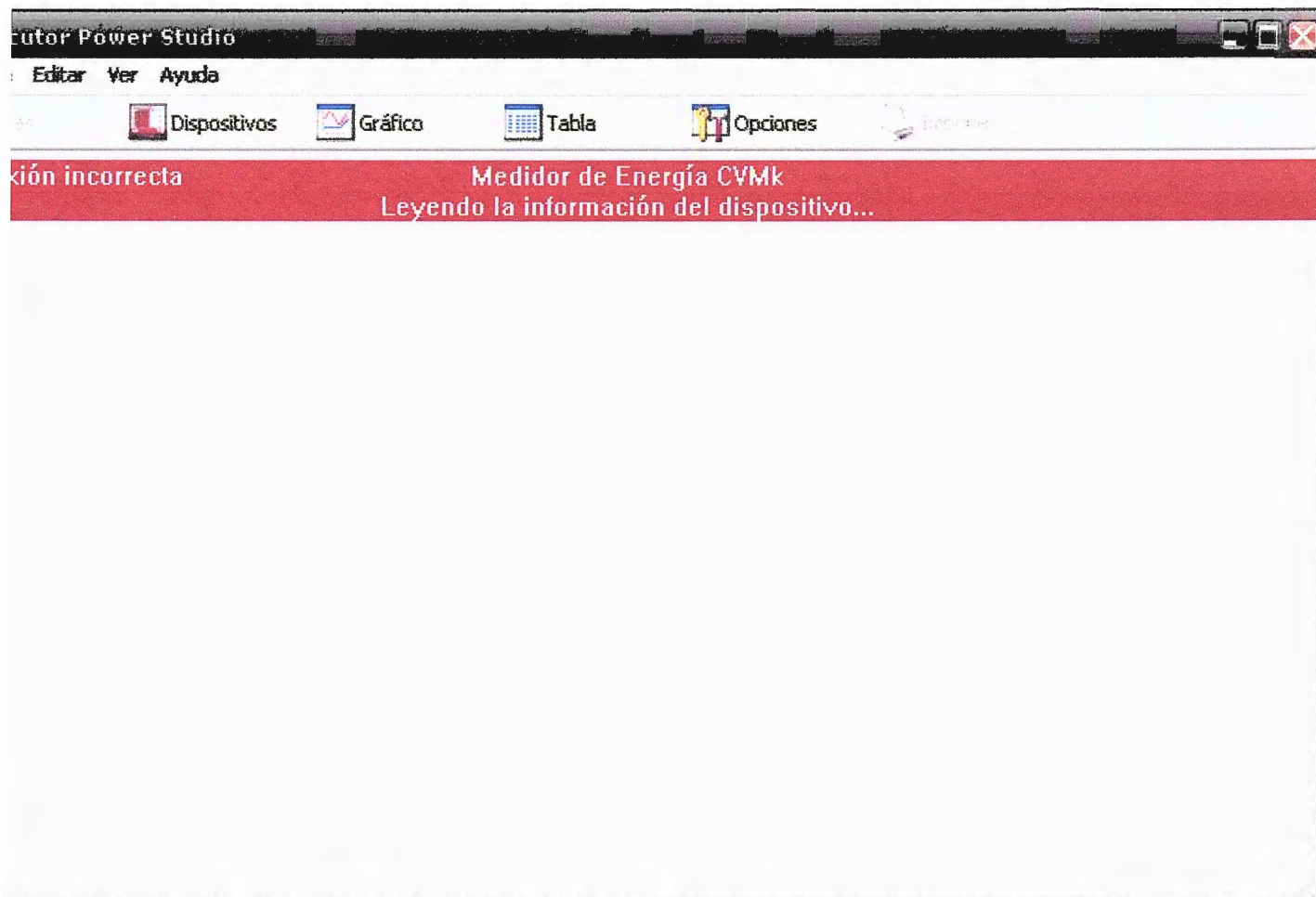
Item

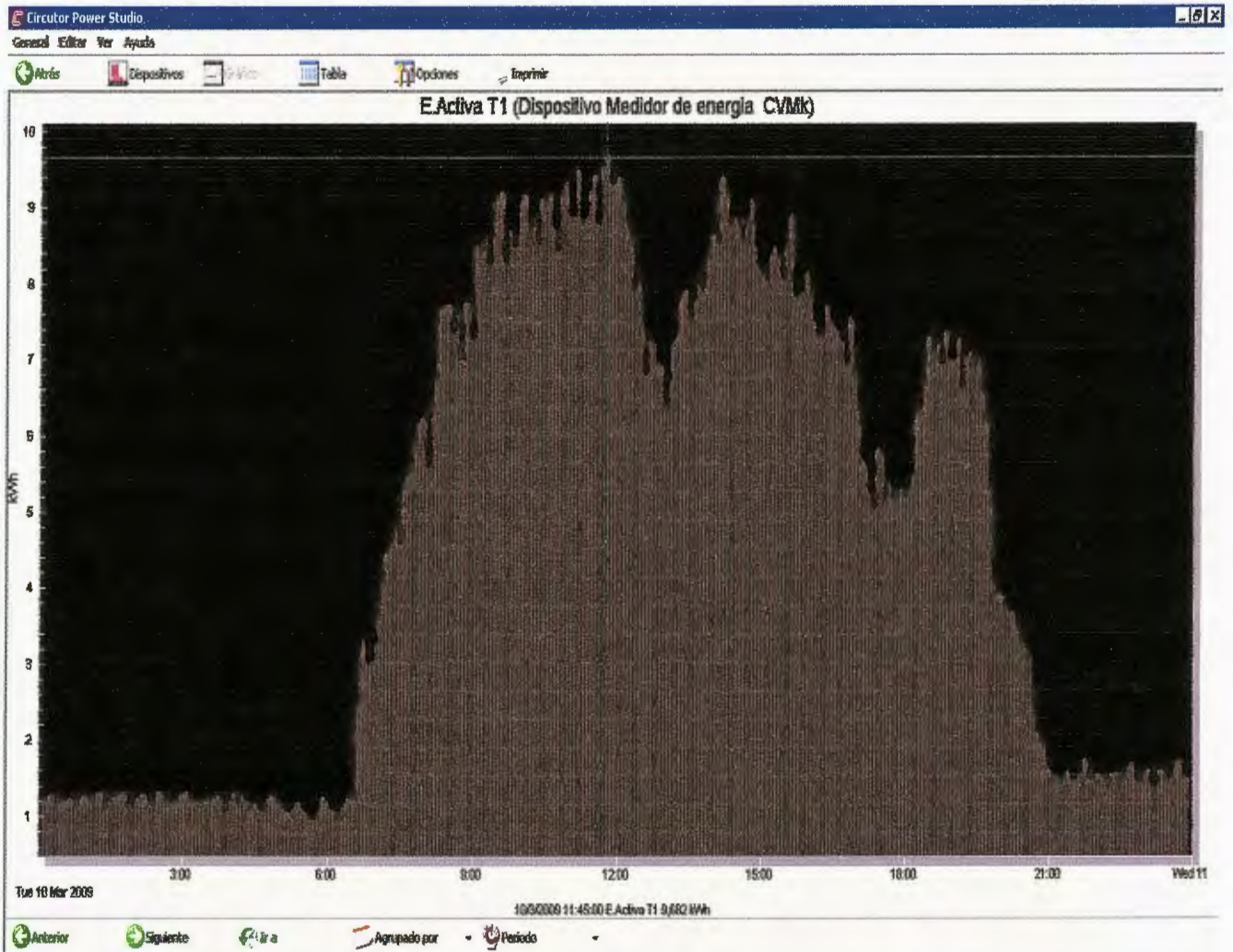
El *Item* es el elemento que las aplicaciones solicitan, tensión fase 1, frecuencia, estado de un relé, etc. Por ejemplo “=PWSTDValues|Dispositivo1|V11”.

Anexo C. EJECUCIÓN DEL PROGRAMA

C1. SOFTWARE POWER STUDIO VERSIÓN 1.0

C1.1 Pantalla de Error en PowerStudio 1.0





C1.3 Pantalla que muestra el comportamiento de la energía activa en kW-H para el día 10 de marzo de 2009.

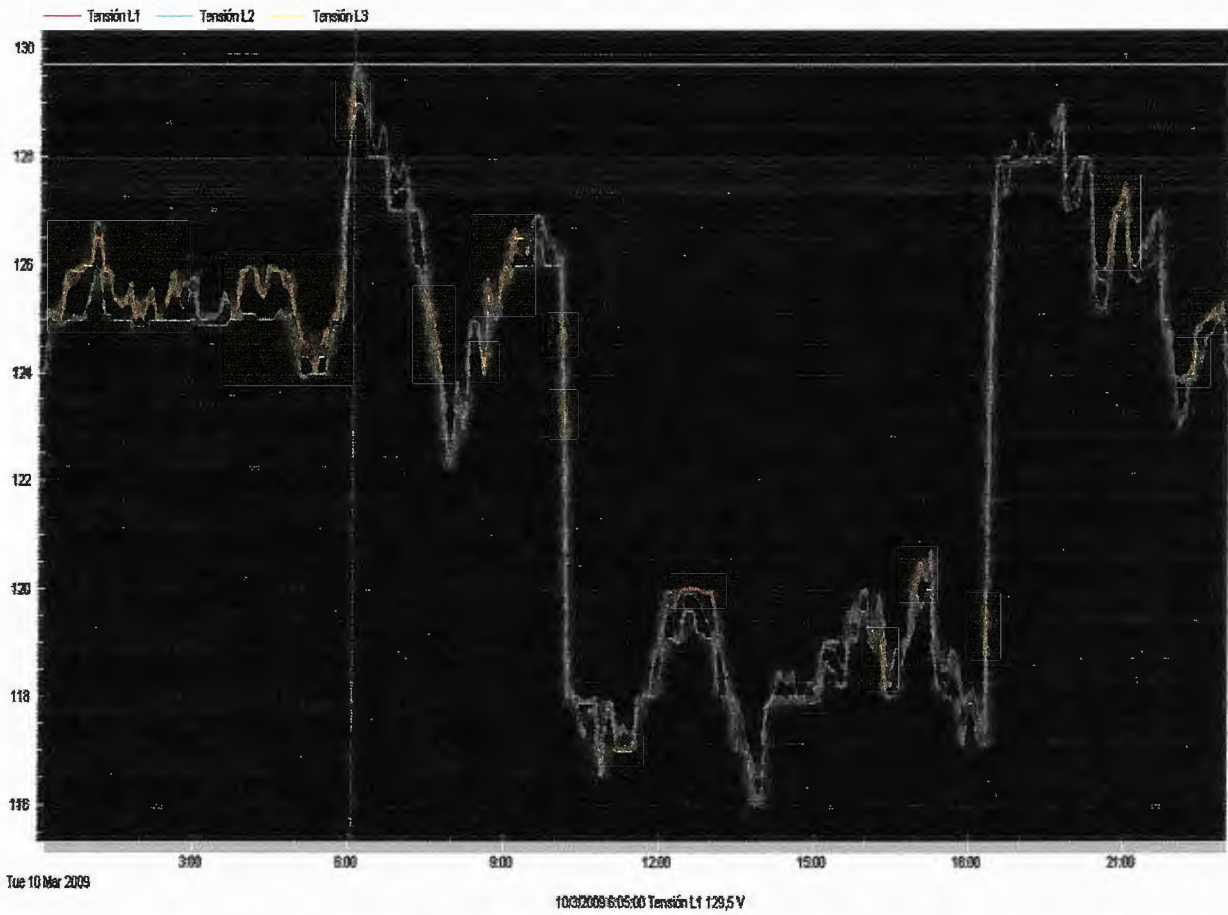
	Instantaneos						Instantaneos		
	Todo						Todo		
	L1	L2	L3	III	T1		T2	T3	
Tensión fase - neutro	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Energía activa	<input checked="" type="checkbox"/>		
Tensión fase - fase	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Energía capacitiva	<input type="checkbox"/>		
Distorsión en tensión						Energía inductiva	<input type="checkbox"/>		
Frecuencia	<input type="radio"/>	<input type="checkbox"/>				Energía activa -			
Corriente	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Energía capacitiva -			
Corriente de neutro						Energía inductiva -			
Distorsión en corriente						Máxima demanda	<input type="checkbox"/>		
Potencia activa	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Potencia capacitiva	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Potencia inductiva	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Potencia aparente	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				
Factor de potencia	<input type="radio"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>				

Aceptar
 Cancelar

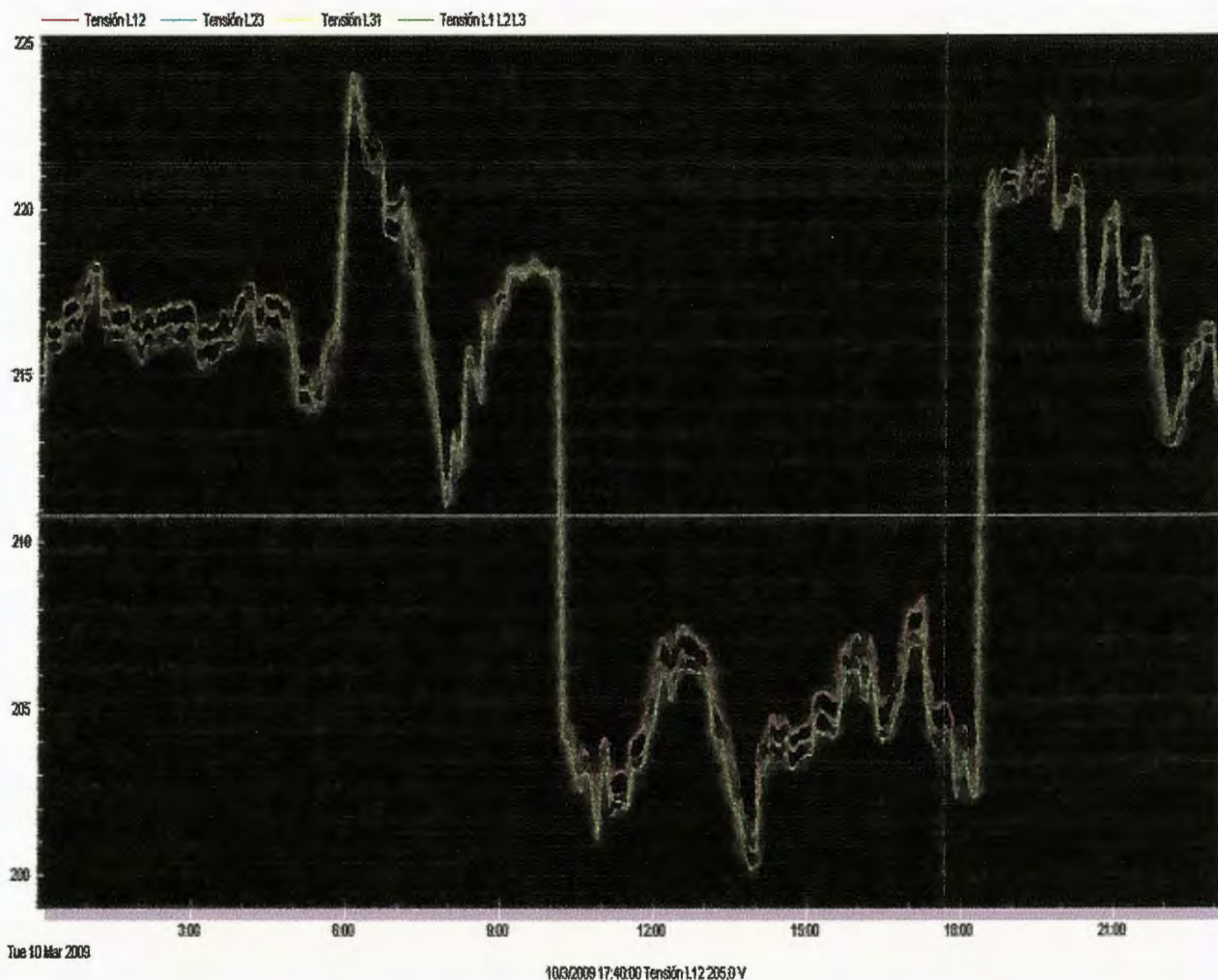
C1.2 Pantalla de Selección de Variables para su graficación.



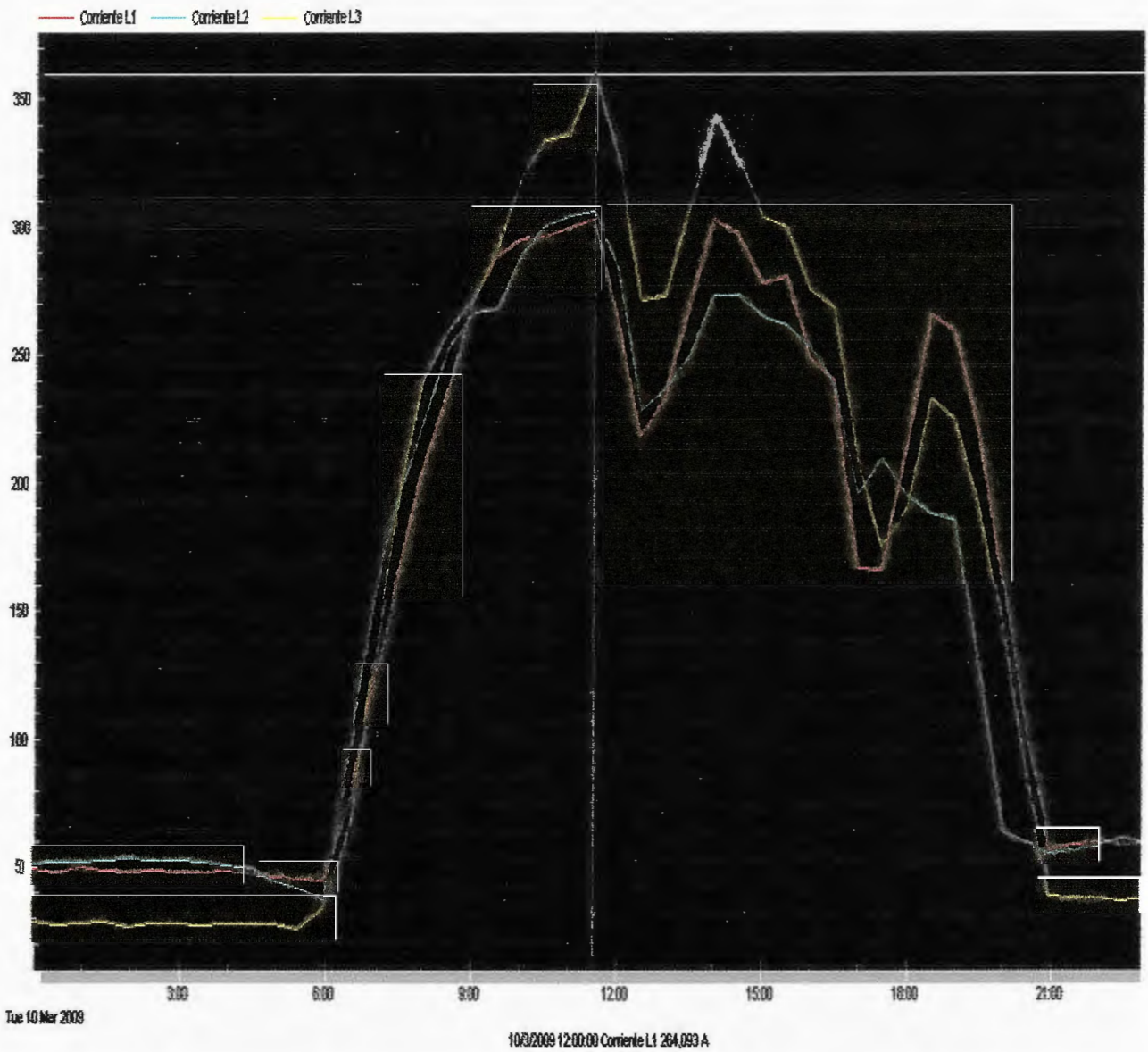
C1.4 Segunda opción de visualización de pantalla principal.



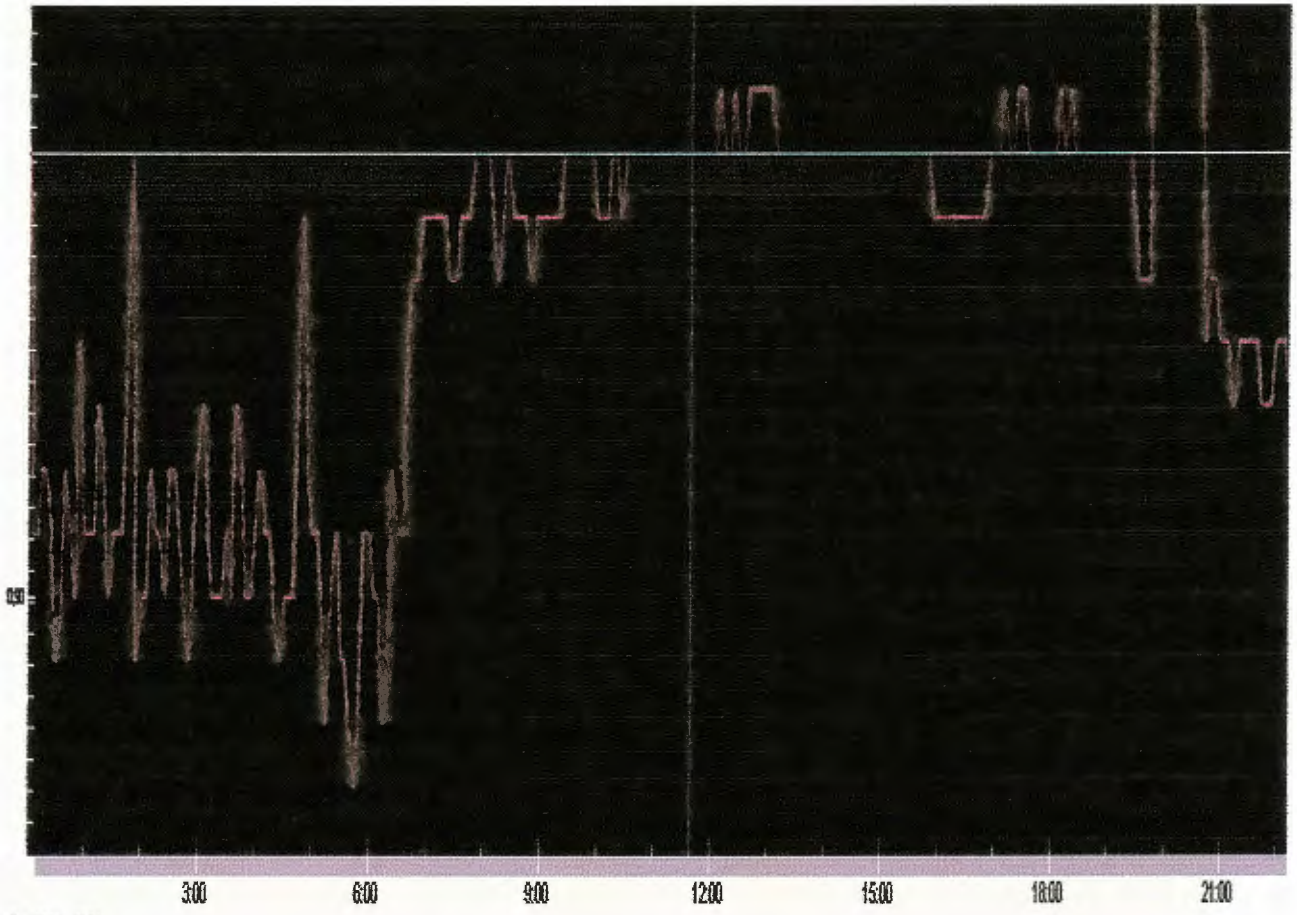
C1.5 Comportamiento de las tensiones de fase en la subestación #2 para el día 10 de marzo de 2009.



C1.6 Comportamiento de las tensiones de línea de la subestación #2 para el día 10 de marzo de 2009.



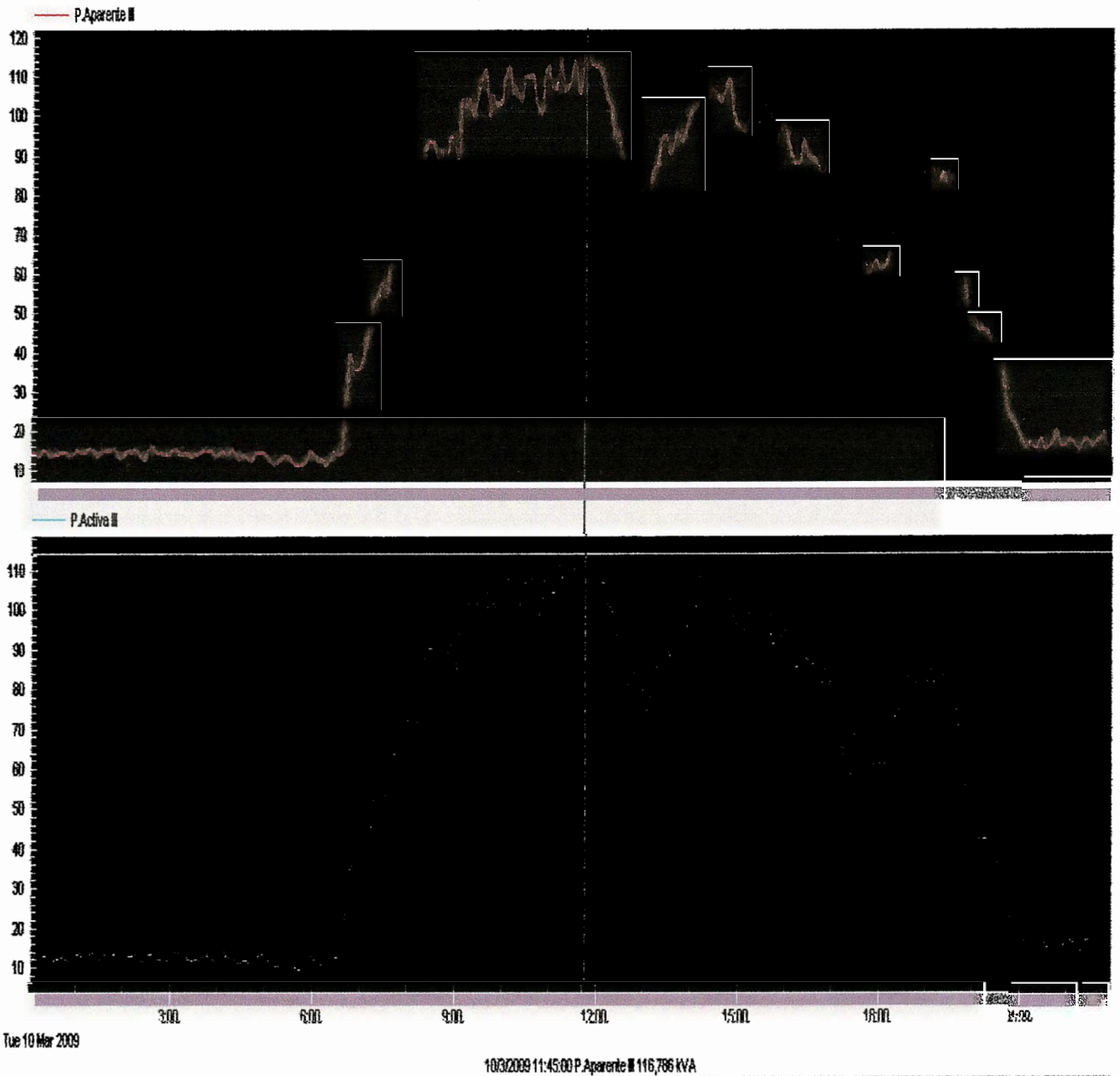
C1.7 Comportamiento de las corrientes de fase de la subestación #2 para el día 10 de marzo de 2009.



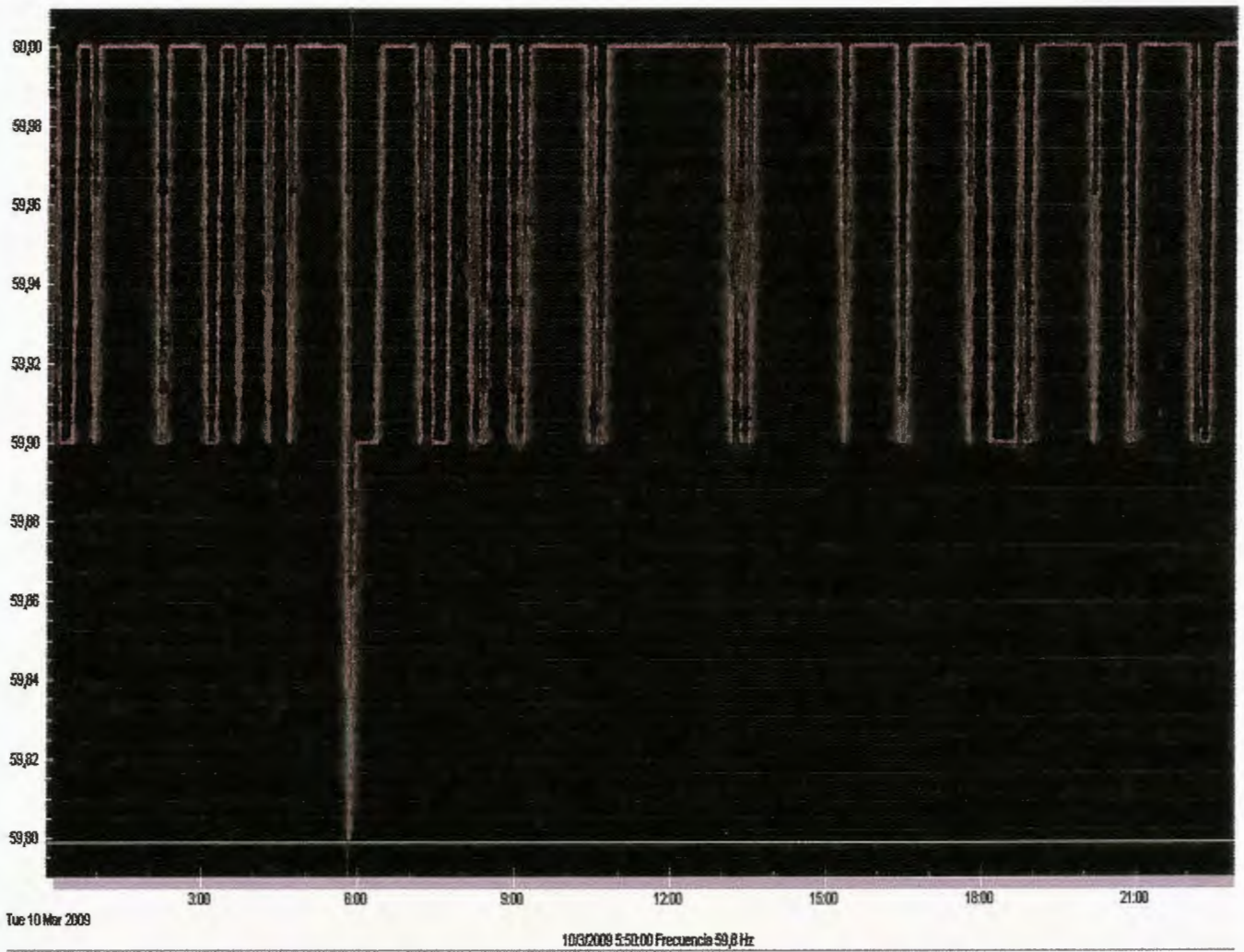
Tue 10 Mar 2009

10/3/2009 11:40:00 F.P. # 0,97

C1.8 Comportamiento del factor de potencia promedio para el 10 de marzo de 2009.



C1.9 Comportamiento de las potencias aparente y activa trifásicas para el 10 de marzo de 2009.



C1.10 Variación de la frecuencia el día 10 de marzo de 2009.

C2. PROGRAMA EN MATLAB

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Fecha y hora	Ia	Ib	Ic	Vab	Vbc	Vca	VLNa	VLNb	VLNc	FPa	FPb	FPc	kWa	kWb	kWc	kWh	kVARh	kVAh	kVARa	kVARb	kVARc
2	09/03/2007 15:58	390,7	345,2	409,1	218	218,7	217,7	125,3	126,4	126,2	0,936	0,96	0,974	46	42,7	50,9	94,3	4,91	0,04	17,2	12,5	11,4
3	09/03/2007 16:08	402,6	333	375,5	217,8	218,6	218,1	125,3	126,3	126,3	0,93	0,952	0,978	46,9	40	48,3	115,17	5,04	0,04	18,2	12,9	9,7
4	09/03/2007 16:18	354,9	271,3	346,1	218,4	219,4	218,6	125,7	126,7	126,6	0,932	0,963	0,965	41,5	33,2	43,5	136,91	4,45	0,04	15,8	9	4,5
5	09/03/2007 16:28	380,3	318,7	385,6	218,4	219,3	218,4	125,7	126,7	126,5	0,922	0,97	0,965	44,1	39,2	48,2	157,86	4,78	0,04	18,6	9,8	8,2
6	09/03/2007 16:38	351,2	261,9	342,8	219,2	220,2	219,4	126,1	127,1	127,2	0,91	0,962	0,99	40,5	32,1	43,3	178,8	4,44	0,04	18,5	9	6,1
7	09/03/2007 16:48	347,5	285,5	352,4	218,9	219,7	218,9	126,1	126,9	126,7	0,92	0,973	0,969	40,5	35,3	44,3	190,23	4,4	0,04	17,5	8,7	6,5
8	09/03/2007 16:58	342	259,5	343,4	220,2	221	220,2	127	127,6	127,4	0,918	0,967	0,981	40,5	32	43,5	210,9	4,37	0,04	17,5	8,4	8,7
9	09/03/2007 17:08	354,1	258,3	347,2	217,5	218,5	217,6	125,3	126,1	126	0,912	0,979	0,993	40,6	31,4	43,4	228,92	4,45	0,04	18,1	6,7	5,3
10	09/03/2007 17:19	276,1	182,1	251,6	218,9	219,8	219,1	126,2	126,8	126,9	0,919	0,957	0,984	32,1	22,4	31,6	244,79	3,49	0,04	13,8	6,6	5,6
11	09/03/2007 17:29	275,4	196,2	259,4	219,5	220,2	219,6	126,4	127,1	127,2	0,92	0,975	0,995	32	24,4	32,7	258,8	3,48	0,04	13,4	5,5	3,3
12	09/03/2007 17:39	266,6	169,1	238,9	220,2	221,1	220,6	126,9	127,5	127,8	0,915	0,963	0,982	31,1	20,8	30,1	272,99	3,39	0,04	13,7	5,8	5,7
13	09/03/2007 17:49	230,4	149,5	199	221,2	221,8	221,6	127,4	128	128,4	0,911	0,954	0,983	26,8	19,5	25,1	286,86	2,94	0,04	12,2	5	4,8
14	09/03/2007 17:59	249,2	179,5	235,6	220,8	221,2	220,9	127,2	127,7	128	0,932	0,976	0,987	29,3	22,4	29,9	300,06	3,15	0,04	11,2	4,9	2,2
15	09/03/2007 18:09	237,7	143,8	206,9	221,1	221,8	221,5	127,5	127,9	128,3	0,907	0,959	0,981	27,5	17,7	26,1	312,24	3,04	0,04	12,9	5,2	5,1
16	09/03/2007 18:19	247,4	150,3	228,6	220,7	221,2	220,9	127,2	127,6	128	0,91	0,969	0,983	28,7	18,6	29	324,54	3,16	0,04	13,2	4,8	5,4
17	09/03/2007 18:29	307,8	131,5	280,1	218,5	219,9	219,1	126	126,6	127,2	0,918	0,956	0,953	35,6	15,9	34	338,99	3,88	0,04	15,6	4,9	10,7
18	09/03/2007 18:39	314,1	122,3	278,9	217,8	219,3	218,8	125,6	126,2	127	0,923	0,937	0,926	36,5	14,5	32,6	353,97	3,95	0,04	15,2	5,4	13,2
19	09/03/2007 18:49	280,1	116,7	248,4	218,8	220	219,5	126,2	126,6	127,3	0,833	0,866	0,839	33	14,3	29,7	366,96	3,54	0,04	12,9	3,8	10,9

643	17/03/2007 5:02	32,9	30,2	31,1	221,6	222	222,6	128,5	127,5	128,6	0,885	0,899	0,969	3,7	3,5	3,9	6831,4	0,42	0,19	2	1,7	1
644	17/03/2007 5:12	29,7	27,2	31,4	221,2	221,6	222,2	128,2	127,4	128,4	0,872	0,912	0,978	3,3	3,1	3,9	6833,4	0,38	0,18	1,9	1,4	0,9
645	17/03/2007 5:22	32,3	28,3	36,4	220,6	220,9	221,6	127,8	127	128	0,886	0,917	0,978	3,7	3,3	4,6	6835,3	0,41	0,19	2	1,4	1
646	17/03/2007 5:32	33,3	24,9	35,4	220,1	220,5	221,2	127,5	126,8	127,8	0,872	0,83	0,976	3,7	2,9	4,4	6837,2	0,43	0,19	2,1	1,2	1
647	17/03/2007 5:42	35,4	30,3	31,6	219,9	220,3	220,8	127,4	126,6	127,5	0,891	0,903	0,973	4	3,5	3,9	6839	0,45	0,19	2	1,6	0,9
648	17/03/2007 5:53	33,3	27	32,4	220,6	220,9	221,6	127,8	127	127,8	0,889	0,911	0,976	3,8	3,1	4,1	6841	0,43	0,19	1,9	1,4	0,9
649	17/03/2007 6:03	29,7	47,9	27,6	222,2	222,2	223,2	128,9	127,7	128,9	0,88	0,961	0,97	3,4	5,9	3,5	6843	0,38	0,19	1,8	1,7	0,9
650	17/03/2007 6:13	33,4	45,5	33,4	222,9	223,1	223,9	129,2	128,3	129,2	0,901	0,97	0,979	3,9	5,7	4,2	6845,3	0,43	0,19	1,9	1,4	0,9
651	17/03/2007 6:23	36,3	35,9	43,2	221,6	222	222,8	128,5	127,7	128,5	0,912	0,947	0,966	4,3	4,3	5,5	6847,4	0,47	0,19	1,9	1,5	1
652	17/03/2007 6:33	32,3	40,3	56,9	221,2	221,5	222,1	128,2	127,4	128,1	0,903	0,946	0,983	3,7	4,9	7,2	6849,9	0,41	0,19	1,8	1,6	0,9
653	17/03/2007 6:43	42,3	42	53,5	220,2	220,5	221,3	127,7	128,9	127,7	0,944	0,955	0,983	5,1	5,1	6,8	6852,7	0,54	0,19	1,8	1,6	0,8
654	17/03/2007 6:53	54	34,3	59,7	220,5	221	221,7	127,8	127,2	127,9	0,964	0,931	0,994	6,7	4	7,6	6855,6	0,69	0,19	1,9	1,6	0,8

C2.1. Base de datos obtenida con el software Labview, a partir de la cual se importan los datos desde Matlab.

C2.2. Pantalla Inicial

A continuación se muestra su código fuente:

```
function varargout = inicio(varargin)
% INICIO M-file for inicio.fig
%   INICIO, by itself, creates a new INICIO or raises the existing
%   singleton*.
%
%   H = INICIO returns the handle to a new INICIO or the handle to
%   the existing singleton*.
%
%   INICIO('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in INICIO.M with the given input arguments.
%
%   INICIO('Property','Value',...) creates a new INICIO or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before inicio_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to inicio_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help inicio

% Last Modified by GUIDE v2.5 08-Mar-2009 10:09:04

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @inicio_OpeningFcn, ...
    'gui_OutputFcn',  @inicio_OutputFcn, ...
    'gui_LayoutFcn',  [], ...
    'gui_Callback',   []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% — Executes just before inicio is made visible.
function inicio_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to inicio (see VARARGIN)
ax=handles.pushbutton2
imagen=imread('escudo.jpg');
```

```

;set(ax,'CData',imagen)
set(ax,'string','')
% Choose default command line output for inicio
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes inicio wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = inicio_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
datos
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

close(gcf)

```

C2.3. PANTALLA PRINCIPAL.

Código fuente.

```
function varargout = datos(varargin)
% DATOS M-file for datos.fig
%   DATOS, by itself, creates a new DATOS or raises the existing
%   singleton*.
%
%   H = DATOS returns the handle to a new DATOS or the handle to
%   the existing singleton*.
%
%   DATOS('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in DATOS.M with the given input arguments.
%
%   DATOS('Property','Value',...) creates a new DATOS or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before datos_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to datos_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help datos

% Last Modified by GUIDE v2.5 08-Mar-2009 20:08:04

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @datos_OpeningFcn, ...
                  'gui_OutputFcn', @datos_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before datos is made visible.
function datos_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to datos (see VARARGIN)
ed1=handles.edit1;
```

```

ed2=handles.edit2;
ed3=handles.edit3;
ed4=handles.edit4;
ed5=handles.edit5;
ed6=handles.edit6;
ed7=handles.edit7;
ed8=handles.edit8;
matriz=xlsread('Sub');
ax=size(matriz);
la=matriz(ax(1,1),2);
set(ed1,'string',la);

lb=matriz(ax(1,1),3);
set(ed2,'string',lb);

lc=matriz(ax(1,1),4);
set(ed3,'string',lc);

Van=matriz(ax(1,1),8);
Vbn=matriz(ax(1,1),9);
Vcn=matriz(ax(1,1),10);

Vfpromedio=(Van+Vbn+Vcn)/3;
Vfprom=num2str(Vfpromedio);
set(ed6,'string',Vfprom);

Vab=matriz(ax(1,1),5);
Vbc=matriz(ax(1,1),6);
Vca=matriz(ax(1,1),7);

VLpromedio=(Vab+Vbc+Vca)/3;
VLprom=num2str(VLpromedio);
set(ed5,'string',VLprom);

fpa=matriz(ax(1,1),11);
fpb=matriz(ax(1,1),12);
fpc=matriz(ax(1,1),13);

fppromedio=(fpa+fpb+fpc)/3;
fpprom=num2str(fppromedio);
set(ed7,'string',fpprom);

energy=matriz(ax(1,1),17);
set(ed4,'string',energy);

Pa=matriz(ax(1,1),14);
Pb=matriz(ax(1,1),15);
Pc=matriz(ax(1,1),16);

Ptrifasica=(Pa+Pb+Pc);
P3f=num2str(Ptrifasica);
set(ed8,'string',P3f);

```

```

% Choose default command line output for datos
handles.output = hObject;

```

```
% Update handles structure
guidata(hObject, handles);
```

```
% UIWAIT makes datos wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = datos_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% Get default command line output from handles structure
varargout{1} = handles.output;
```

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
acerca
```

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
inicio
```

```
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
```

```
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
potencia
```

```
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton6 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% --- Executes on button press in pushbutton7.
```

```
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
datos
```

```
% --- Executes on button press in pushbutton8.
```

```
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton8 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
voltaje
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function edit1_Callback(hObject, eventdata, handles)
```

```
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit1 as text
```

```
% str2double(get(hObject,'String')) returns contents of edit1 as a double
```

```
% --- Executes during object creation, after setting all properties.
```

```
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
```

```
% See ISPC and COMPUTER.
```

```
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```

function edit2_Callback(hObject, eventdata, handles)
% hObject   handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%       str2double(get(hObject,'String')) returns contents of edit2 as a double

```

```

% — Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit3_Callback(hObject, eventdata, handles)
% hObject   handle to edit3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%       str2double(get(hObject,'String')) returns contents of edit3 as a double

```

```

% — Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit4_Callback(hObject, eventdata, handles)
% hObject   handle to edit4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%       str2double(get(hObject,'String')) returns contents of edit4 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%       str2double(get(hObject,'String')) returns contents of edit5 as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%       str2double(get(hObject,'String')) returns contents of edit6 as a double

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```

```

if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit7_Callback(hObject, eventdata, handles)
% hObject   handle to edit7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%        str2double(get(hObject,'String')) returns contents of edit7 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject   handle to edit8 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit8_Callback(hObject, eventdata, handles)
% hObject   handle to edit8 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%        str2double(get(hObject,'String')) returns contents of edit8 as a double

```

C2.4. PANTALLA DE TENSIONES.

Código Fuente.

```
function varargout = voltaje(varargin)
% VOLTAJE M-file for voltaje.fig
%   VOLTAJE, by itself, creates a new VOLTAJE or raises the existing
%   singleton*.
%
%   H = VOLTAJE returns the handle to a new VOLTAJE or the handle to
%   the existing singleton*.
%
%   VOLTAJE('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in VOLTAJE.M with the given input arguments.
%
%   VOLTAJE('Property','Value',...) creates a new VOLTAJE or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before voltaje_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to voltaje_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help voltaje

% Last Modified by GUIDE v2.5 11-Mar-2009 09:35:25

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @voltaje_OpeningFcn, ...
                  'gui_OutputFcn', @voltaje_OutputFcn, ...
                  'gui_LayoutFcn', [], ...
                  'gui_Callback', []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% — Executes just before voltaje is made visible.
function voltaje_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to voltaje (see VARARGIN)
```

```

ed1=handles.edit1;
ed2=handles.edit2;
ed3=handles.edit3;
ed9=handles.edit9;
ed5=handles.edit5;
ed6=handles.edit6;
ed7=handles.edit7;
ed8=handles.edit8;
matriz=xlsread('Sub');
ax=size(matriz);

Van=matriz(ax(1,1),8);
set(ed1,'string',Van)
Vbn=matriz(ax(1,1),9);
set(ed2,'string',Vbn);
Vcn=matriz(ax(1,1),10);
set(ed3,'string',Vcn);
Vfpromedio=(Van+Vbn+Vcn)/3;
Vfprom=num2str(Vfpromedio);
set(ed9,'string',Vfprom);

Vab=matriz(ax(1,1),5);
set(ed5,'string',Vab);
Vbc=matriz(ax(1,1),6);
set(ed6,'string',Vbc);
Vca=matriz(ax(1,1),7);
set(ed7,'string',Vca);

VLpromedio=(Vab+Vbc+Vca)/3;
VLprom=num2str(VLpromedio);
set(ed8,'string',VLprom);
% Choose default command line output for voltaje
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes voltaje wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = voltaje_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

acerca

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
inicio
```

```
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
```

```
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
potencia
```

```
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
datos
```

```
% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
voltaje
```

```
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
```

```

else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

```

```

% — Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double

```

```

% — Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
% str2double(get(hObject,'String')) returns contents of edit3 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit5_Callback(hObject, eventdata, handles)
% hObject handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit5 as text
% str2double(get(hObject,'String')) returns contents of edit5 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit6_Callback(hObject, eventdata, handles)
% hObject handle to edit6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit6 as text
% str2double(get(hObject,'String')) returns contents of edit6 as a double

```

```

% --- Executes during object creation, after setting all properties.

```

```

function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%       str2double(get(hObject,'String')) returns contents of edit7 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit8_Callback(hObject, eventdata, handles)
% hObject    handle to edit8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%       str2double(get(hObject,'String')) returns contents of edit8 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else

```

```

    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%        str2double(get(hObject,'String')) returns contents of edit9 as a double

```

C2.5 PANTALLA DE POTENCIA.

Código Fuente.

```

function varargout = potencia(varargin)
% POTENCIA M-file for potencia.fig
%   POTENCIA, by itself, creates a new POTENCIA or raises the existing
%   singleton*.
%
%   H = POTENCIA returns the handle to a new POTENCIA or the handle to
%   the existing singleton*.
%
%   POTENCIA('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in POTENCIA.M with the given input arguments.
%
%   POTENCIA('Property','Value',...) creates a new POTENCIA or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before potencia_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to potencia_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help potencia

% Last Modified by GUIDE v2.5 09-Mar-2009 00:10:55

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @potencia_OpeningFcn, ...
                  'gui_OutputFcn',  @potencia_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin & isstr(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});

```

```

end
% End initialization code - DO NOT EDIT

% — Executes just before potencia is made visible.
function potencia_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject   handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
% varargin  command line arguments to potencia (see VARARGIN)
ed1=handles.edit1;
ed2=handles.edit2;
ed3=handles.edit3;
ed4=handles.edit4;
ed5=handles.edit5;
ed6=handles.edit6;
ed7=handles.edit7;
ed8=handles.edit8;
ed9=handles.edit9;
ed10=handles.edit10;
ed11=handles.edit11;
ed12=handles.edit12
matriz=xlsread('Sub');
ax=size(matriz);

Pa=matriz(ax(1,1),14);
Pb=matriz(ax(1,1),15);
Pc=matriz(ax(1,1),16);
set(ed1,'string',Pa);
set(ed2,'string',Pb);
set(ed3,'string',Pc);

Ptrifasica=(Pa+Pb+Pc);
P3f=num2str(Ptrifasica);
set(ed8,'string',P3f);

Qa=matriz(ax(1,1),20);
Qb=matriz(ax(1,1),21);
Qc=matriz(ax(1,1),22);
set(ed4,'string',Qa);
set(ed5,'string',Qb);
set(ed6,'string',Qc);

Qtrifasica=(Qa+Qb+Qc);
Q3f=num2str(Qtrifasica);
set(ed7,'string',Q3f);

fpa=matriz(ax(1,1),11);
set(ed9,'string',fpa);
fpb=matriz(ax(1,1),12);
set(ed10,'string',fpb);
fpc=matriz(ax(1,1),13);
set(ed11,'string',fpc);

fppromedio=(fpa+fpb+fpc)/3;
fpprom=num2str(fppromedio);
set(ed12,'string',fpprom);

```

```

% Choose default command line output for potencia
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes potencia wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = potencia_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
acerca

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
inicio

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
close(gcf)
datos

```

```

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(gcf)
potencia

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close(gcf)
voltaje

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%       str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');

```

```

else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)

```

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
% str2double(get(hObject,'String')) returns contents of edit4 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit5_Callback(hObject, eventdata, handles)
% hObject handle to edit5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit5 as text
% str2double(get(hObject,'String')) returns contents of edit5 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

```

```

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit6_Callback(hObject, eventdata, handles)
% hObject handle to edit6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit6 as text
% str2double(get(hObject,'String')) returns contents of edit6 as a double

```

```

% --- Executes during object creation, after setting all properties.

```

```

function edit7_CreateFcn(hObject, eventdata, handles)
% hObject  handle to edit7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit7_Callback(hObject, eventdata, handles)
% hObject  handle to edit7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%   str2double(get(hObject,'String')) returns contents of edit7 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject  handle to edit8 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit8_Callback(hObject, eventdata, handles)
% hObject  handle to edit8 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%   str2double(get(hObject,'String')) returns contents of edit8 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject  handle to edit9 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles  empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc

```

```

    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit9_Callback(hObject, eventdata, handles)
% hObject    handle to edit9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%       str2double(get(hObject,'String')) returns contents of edit9 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit10_Callback(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit10 as text
%       str2double(get(hObject,'String')) returns contents of edit10 as a double

```

```

% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

```

function edit11_Callback(hObject, eventdata, handles)

```

```

% hObject handle to edit11 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit11 as text
% str2double(get(hObject,'String')) returns contents of edit11 as a double

% --- Executes during object creation, after setting all properties.
function edit12_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit12 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit12_Callback(hObject, eventdata, handles)
% hObject handle to edit12 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit12 as text
% str2double(get(hObject,'String')) returns contents of edit12 as a double

```

DISEÑO Y EXPERIMENTACIÓN DE UN SISTEMA DE MEDICIÓN REMOTA DE VARIABLES ELÉCTRICAS

Presentan: Gerardo René Vizcarra Moreno y Samuel Alexander Lucero Culi Estudiantes de Ingeniería Eléctrica

Resumen. – El siguiente artículo fue elaborado con el fin de facilitar la supervisión y medición de parámetros eléctricos de manera remota (a distancia).

Específicamente a la medición remota de variables eléctricas en la subestación #2 del Centro de Investigación y Transferencia de Tecnología (CITT) de la Universidad Don Bosco, el cual se realizó instalando un CPU con tarjeta de Red Inalámbrica para tener acceso al Internet y valiéndonos del uso del medidor de la marca Circutor Modelo CVMk que ya estaba instalado en la subestación pero que solo se utilizaba para medir los parámetros eléctricos en dicho lugar; la Universidad Don Bosco también cuenta con un medidor de la Marca Veris modelo 8437 (Utilizado para usos didácticos) y que no está conectado actualmente. Para poder llevar a cabo nuestro cometido se utilizó el software PowerStudio® proporcionado por el fabricante del medidor Circutor y una aplicación en Matlab® que elaboramos para el medidor Veris 8437 con el fin de hacer una interfaz de usuario un poco más amigable que utiliza archivos de Microsoft Excel.

INTRODUCCIÓN

El consumo de energía eléctrica se incrementa cada año, debido al crecimiento poblacional y de empresas. Las empresas de distribución de energía eléctrica y las empresas consumidoras de este servicio, necesitan determinar el consumo eléctrico con el fin de realizar una facturación, en el caso de la distribuidora, y monitorear el consumo eléctrico, por parte de la empresa consumidora; y de esta manera tener controlado el consumo de energía eléctrica y poder planificar con base a estudios de una determinada red de energía.

Una manera de hacer práctica la lectura de los medidores de energía es, mediante la medición remota de variables eléctricas, tales como potencia, energía, tensión, corriente, factor de potencia y distorsión armónica.

Para lo cual se hace uso de equipo de medición que la universidad posee actualmente pero que no se está utilizando como debe ser, para lograr un monitoreo remoto del consumo de energía y de otros parámetros eléctricos importantes mencionados anteriormente.

ESQUEMA GENERAL DE LA MEDICIÓN

El siguiente esquema se utiliza alrededor del mundo, por las diferentes empresas distribuidoras de energía eléctrica. En este se distinguen tres bloques principales que, con variantes en sus denominaciones son:

- / Instalaciones de medición
- / Medios de Comunicación
- / Centros de Control

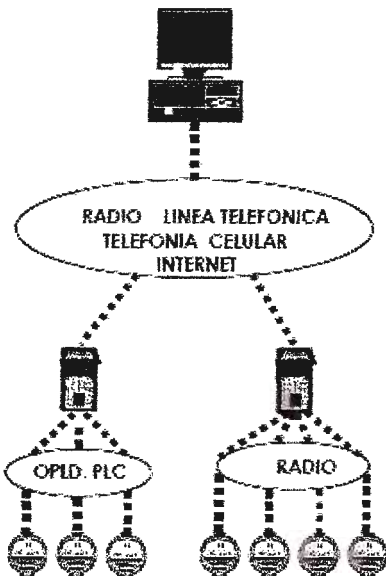


Figura 1 Esquema de Medición¹

En nuestro caso nos enfocaremos en los primeros dos. El punto de la medición está formado básicamente por medidores de energía eléctrica, concentradores de datos e interfaces a los medios de comunicación. Los canales y protocolos de comunicación constituyen los medios de comunicación, sirven de enlace entre el centro de procesamiento de información y los dispositivos de medición de energía eléctrica.

¹ Figura Obtenida de Medición remota de energía, Universidad Tecnológica de la Mixteca

CONSIDERACIONES GENERALES

Del tipo de conexión del sistema eléctrico existente en la universidad Don Bosco depende cada una de las variables eléctricas involucradas para la medición remota. La subestación #2 del CITT consta de un banco de transformadores monofásicos 23kV / 220V con una capacidad de 167kVA cada uno; lo que quiere decir que la capacidad de la subestación #2 es de 501kVA con un sistema balanceado en estrella.

La subestación #2 alimenta los siguientes edificios del Centro de Investigación y Transferencia de Tecnología (CITT)²:

- o Edificio #5 Comprende principalmente los departamentos de Metrología, Proyectos Educativos, academia CISCO y la parte de capacitaciones.
- o Edificio #6 comprende el taller de mecánica y los centros de cómputo.
- o Edificio #7 este con excepción de los demás se alimenta por dicha subestación y por un transformador monofásico de 100 KVA.

Esta subestación cuenta con una planta de emergencia de 205 kW la cual sirve como respaldo para algunas cargas de estos edificios, la planta se encuentra conectada en paralelo con la línea mediante un interruptor de transferencia el cuál mediante un controlador lógico programable entra en operación cuando hay cortes en el servicio de energía eléctrica proporcionado por CAESS.

² Trabajo de graduación: "Análisis técnico-económico del comportamiento de variables eléctricas que inciden en la facturación y consumo de energía eléctrica en el centro de investigaciones y transferencia de tecnología (CITT)

MEDICIÓN DE VARIABLES ELÉCTRICAS

En el caso de este trabajo se miden los siguientes parámetros eléctricos, tensión, corriente, potencias y factor de potencia ya que son las magnitudes más comúnmente utilizadas para efectos de cobro por parte de las compañías distribuidoras.

Para llevar a cabo la medición se hace uso de transformadores de corriente para obtener la señal de corriente en el medidor de energía. Esto se hace debido a que los valores de corriente por fase que actualmente oscilan en un rango entre 200 y 300 amperios, corriente que no puede manejar directamente el medidor de energía, por lo que se usan transformadores de corriente con una relación 1200:5, relación que se configura en el medidor.

Para el caso de la señal de tensión, no es necesario utilizar transformador de potencial porque la señal se toma del secundario del transformador de la subestación trifásica. Valores de tensión que el medidor soporta.

El medidor Circutor CVMk proporciona más datos pero para efectos de nuestro trabajo utilizaremos las variables antes mencionadas. Este medidor se encargará de enviar los datos a través de un convertidor RS232 / RS485 al CPU que se utiliza como receptor de los datos y en el cual estos últimos se almacenan para posteriormente visualizarlos a través de la red desde cualquier punto dentro de la red con la que cuenta la Universidad.

MEDIO DE TRANSMISIÓN Y RECEPCIÓN DE DATOS.

Para el medidor Circutor CVMk el protocolo de comunicación es el protocolo Modbus, que utiliza una interfaz RS-485 a la salida del puerto de comunicación del medidor de energía. Esta interfaz RS-485 debe convertirse a una RS-232 para obtener datos desde el medidor vía el puerto serial hacia el ordenador que se utilizará para la obtención de datos del medidor.

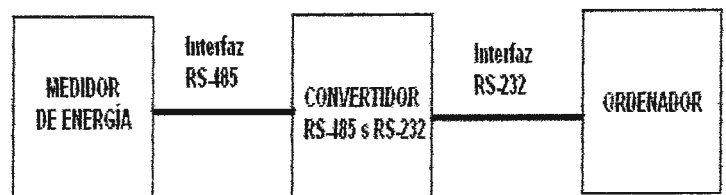


Figura 2. Esquema de los dispositivos involucrados en la transmisión y recepción de datos en el sistema Medidor-Ordenador.

RECEPCIÓN DE DATOS EN EL MEDIDOR DE ENERGÍA CIRCUTOR CVMk.

Para el medidor CVMk se cuenta con el software PowerStudio® versión 1.0, este se encarga de enlazar los datos que recibe el puerto COM del ordenador y los procesa para mostrarlos en pantalla.

El puerto serial de un ordenador posee un conector tipo DB9 de los cuales para la recepción de datos y para la petición o solicitud de lectura al medidor de energía se utilizan solamente los pines 2,3 y 5; para recepción de datos, transmisión de datos y el tierra respectivamente.

El ordenador mediante el software transmite por el pin 3 una petición de lectura al medidor de energía y este le devuelve el número o los números de registros solicitados en el pin 2, por el cual recibe datos el ordenador.

Lo anterior se logra mediante el software PowerStudio el cual hace la petición de un *determinado número de registros al medidor de energía* Circutor CVMk y luego procesa esta información. Cabe mencionar que los datos que envía el medidor, al ordenador, no todos vienen representando el valor real de la magnitud medida. Por tal razón el fabricante proporciona una tabla de los registros para hacer la correcta petición o solicitud de lectura y además dependiendo de la variable eléctrica solicitada, esta puede o no, tener un factor incluido o por incluir tal como se muestra en dicha tabla de registros, para adecuar la lectura con la señal medida.

Termn. nr.	Parameter
13	Neutral
12	VL1
11	VL2
10	VL3
9	S1 IL1
8	S2 IL1
7	S1 IL2
6	S2 IL2
5	S1 IL3
4	S2 IL3
Supply	
CVMk-H....-	
A.C. Supply	
3 V
2 V
1	0 V
CVMk-H....- / SDC	
D.C. Supply	
3	
2	--
1	+d.c.

alimentación del equipo y las señales de medición de la red.

Dicha bornera se muestra en la siguiente figura:

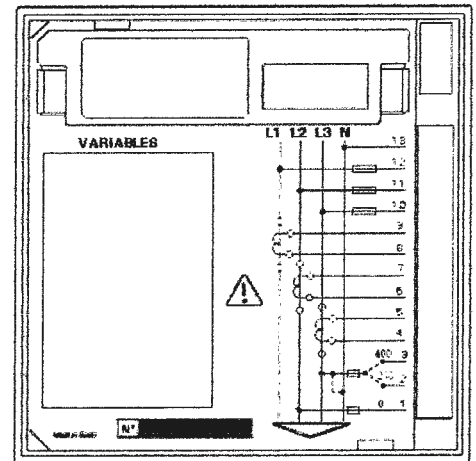


Figura 3 Bornera de conexiones del analizador de redes Circutor.

ESQUEMA DE CONEXIÓN DEL CVMK.

A continuación se muestra el esquema de conexión del CVMK en una Red Trifásica de baja tensión:

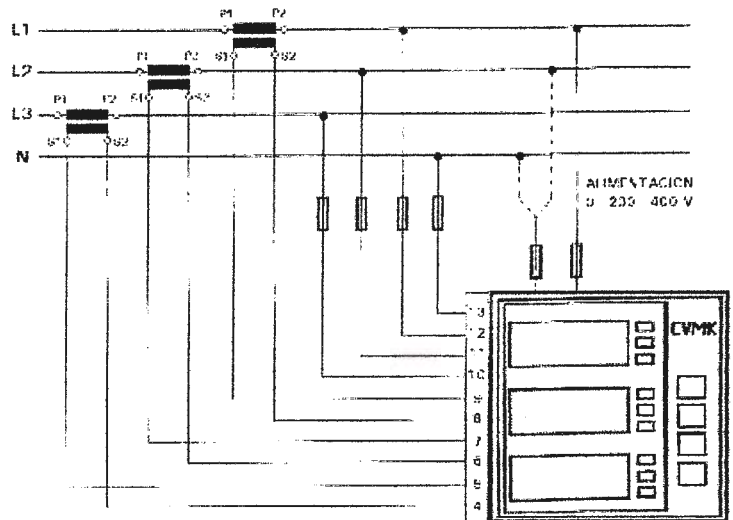


Figura 4 Esquema de conexión a TC's del analizador de redes Circutor.

El software PowerStudio® posee diferentes características en los cuales se pueden monitorear varios modelos de analizadores de redes marca Circutor, con un historial de datos almacenados que se pueden presentar por días, por semana o por mes con el objetivo de obtener un mejor análisis sobre el sistema para el cual se encuentre instalado.

Es de suma importancia utilizar el convertidor inteligente RS485/RS232, porque sin él, la computadora no reconoce los datos que le envía el medidor y de hecho ni siquiera pudiera el ordenador solicitar datos al medidor por encontrarse con protocolos de comunicación diferentes (RS232 y RS485).

BORNERA DE CONEXIONES.

El CVMK dispone de una bornera de conexiones en la parte lateral del aparato para conectar la

CONEXIÓN RED RS-485 A UN ORDENADOR PC (MEDIANTE LA INTERFAZ SERIAL RS-232).

La conexión RS-485 se realiza con cable de comunicación de par trenzado con malla de apantallamiento, de tres hilos como mínimo, con una distancia máxima entre el CVMK y el último periférico de 1200 metros. El CVMK con el módulo CVM/RED (Tarjeta de expansión) utiliza una línea de comunicación RS-485 en la que pueden conectarse hasta un máximo de 32 equipos en paralelo (Mediante el Bus Multipunto) por cada COM del ordenador utilizado.

SOFTWARE POWERSTUDIO®

Destinado al control energético de una instalación eléctrica, para usuarios de equipo Círculo: Analizadores de redes, medidores de energía, indicadores digitales de proceso, centralizadores de pulsos (es un equipo con multitud de entradas, encargado de mandar la información a la PC)

Características:

- ✓ Servidor Web Incorporado (Multiusuario).
- ✓ Visualización en tiempo real de todos los parámetros eléctricos o señales de proceso.
- ✓ Visualización de gráficos de energías.
- ✓ Visualización de tablas de datos.
- ✓ Visualización de Datos Históricos.
- ✓ Posibilidad de exportación de históricos (XML o Excel) en algunas versiones.
- ✓ Posibilidad de exportación de parámetros en tiempo real, mediante servidor DDE o XML, para enlazar con otras aplicaciones externas.
- ✓ Máxima conectividad a nivel interno (Intranet) y a nivel externo (Internet).

Principales Aplicaciones:

- ✓ Controlar niveles de tensión, corriente, potencias, factor de potencia, potencia reactiva, armónicos, etc.
- ✓ Determinar incrementos de energía, y variaciones del consumo eléctrico.
- ✓ Mantenimiento preventivo de líneas, tanto a nivel de corrientes residuales, como consumo excesivo.
- ✓ Medición de energía para líneas de poca potencia vía serie contadores MK, AMK y LM.
- ✓ Estudios de Calidad de suministro eléctrico.
- ✓ Registro de señales externas de proceso (digitales o analógicas).

SOFTWARE DESARROLLADO EN MATLAB PARA EL MEDIDOR VERIS H8437.

El programa desarrollado en el ambiente Matlab® se convierte en una ayuda para facilitar la lectura remota de los parámetros eléctricos. Este no funciona directamente con el medidor de energía.

El programa desarrollado en Labview para el Veris H8437 crea un archivo de Excel para ir guardando las variables eléctricas.

La aplicación desarrollada en Matlab® permite importar datos desde cualquier otro software que cree un archivo de Excel para ir almacenando diferentes parámetros.

Para desarrollar el programa se utilizó la interfaz gráfica de Matlab® GUIDE, en el cual se hizo un programa para que el usuario interactúe gráficamente con el software.

En este se crean dos archivos. Un formulario donde se desarrolla la parte gráfica con extensión .fig y un archivo donde se encuentra el código que se establece al crear ese formulario y es en ese código donde se modifican a conveniencia los diferentes elementos creados en el formulario para llevar a cabo la interacción.

CARACTERÍSTICAS DEL SISTEMA IMPLEMENTADO

Para el medidor de energía Circutor CVMk se cuenta con transformadores de corriente con una relación de 1200:5 correspondiéndole una relación de transformación de corriente (RTC) de 240. Esto quiere decir que si en promedio la subestación proporciona 275 amperios por fase, la señal de corriente que recibe el medidor de energía CVMk es de 1.15 amperios. Valor aceptable tomando en cuenta que los dispositivos de protección y de medición se diseñan para valores de señal de corriente de 5 amperios, esa es la razón por la que los transformadores de corriente en su mayoría, a excepción de los que proporcionan 1 amperio en el secundario, proporcionan un valor de corriente de 5 amperios en el secundario para un valor determinado máximo en el primario.

Al medidor de energía debe programarse las relaciones de transformación para obtener datos reales, de lo contrario mostrarán datos que no corresponden a valores reales. También, aunque no se usen transformadores de potencial, se deben programar los medidores con una relación de tensión de 1:1, si la tiene programada por defecto no es necesario. Pero siempre se debe verificar como se encuentra programado un

determinado medidor de energía para obtener datos esperados.

CONCLUSIONES

- Para la universidad el trabajo realizado representa una gran ayuda al departamento de eléctrica, tanto para el mantenimiento o para realizar proyecciones sobre la red eléctrica interna de la universidad; puesto que se monitorea en tiempo real el comportamiento de los parámetros eléctricos de interés en un punto determinado como lo es la subestación No. 2.
- La aplicación desarrollada en este trabajo de graduación representa una gran ayuda, no solamente para la Universidad Don Bosco, sino también en el ámbito empresarial para generar una equidad y una concordancia entre los clientes y las distribuidoras para evitar problemas legales entre ellos mismos por diferentes motivos relacionados a la facturación del consumo de energía eléctrica.
- Se demostró la factibilidad de un sistema como el implementado, tomando en cuenta la ganancia que tendría la Universidad Don Bosco o cualquier otra empresa para el análisis de sus respectivos sistemas eléctricos de potencia.

GLOSARIO.

Alarma: Mensaje que notifica a un operador o administrador que existe un problema en los elementos de red administrados.

Alta tensión: Tensión igual o superior a 115 kV

Baja tensión: Tensión igual o inferior a 600 Voltios.

BUS: infraestructura cableada en el cual están conectados diferentes dispositivos que pueden conformar una red.

Carga: El término carga, en el lenguaje habitual de la electrotecnia, puede tener varias acepciones:

- Conjunto de valores eléctricos que caracterizan la sollicitación a que está sometido un equipamiento eléctrico (transformador, máquina, etc.).
- Equipamiento eléctrico que absorbe potencia.
- Potencia (o corriente) transferida por un equipamiento eléctrico.
- Potencia instalada.

Por otra parte, para un circuito o equipamiento eléctrico hablamos de funcionamiento en carga cuando está consumiendo potencia y funcionamiento en vacío en caso contrario.

Central generadora: Es el conjunto de equipos utilizados directa e indirectamente para la generación de energía eléctrica, incluidos los edificios y obras civiles necesarias.

COM: puerto de comunicación serial que poseen los computadores, por medio de los cuales se puede establecer comunicación con diversos equipos, por medio de interfaces y periféricos adecuados.

Comercializador: Entidad que compra la energía eléctrica a otros operadores con el objeto de revenderla.

Conexión: Enlace que permite a un usuario final recibir energía eléctrica de una red de transmisión o distribución.

DCS (Distributed Control Systems): Son sistemas de control que geográficamente se encuentran alejados pero que realizan una función común.

DDE (Dynamic Data Exchange): protocolo que permite interactuar entre aplicaciones dentro del sistema operativo Windows; actúa concretamente en modo cliente (solicita datos a otras aplicaciones), o en modo servidor (suministrar datos a otras aplicaciones).

Demanda: Es la carga en las terminales receptoras, tomada como valor medio en un intervalo de tiempo determinado; por lo tanto, hablar de demanda carece de sentido si no se explicita dicho intervalo. Se expresa en kW, kVA o Amperes.

Demanda máxima: Valor máximo constatado de la suma de las demandas simultáneas ocurridas en las centrales de la empresa y en los puntos de compras, si los hubiere, en un período determinado.

Distribuidora: Es la entidad poseedora y operadora de instalaciones, cuya finalidad es la entrega de energía eléctrica en redes de media y baja tensión.

Factor de potencia: Es la relación entre la potencia activa o efectiva y la potencia aparente de una instalación, es decir, que:

$$\text{Factor de potencia} = P/S = \cos \Phi$$

Formulario: ventana de visualización que se genera en aquellos programas que permiten la manipulación de bases de datos.

Interfaz: cable de un número determinado de hilos que permite la comunicación entre los diferentes equipos que conforman una red, ya sea conectándolos directamente (RS 232) o por medio de periféricos llamados convertidores de señal (RS 485 a RS 232), y limitadas por ciertas distancias normadas internacionalmente.

LAN: red de área local (Local Área Network), son redes creadas para trabajar en cierta región o espacio limitado.

Master: del inglés maestro, es el dispositivo que funciona como servidor en una red. Interroga y estimula la comunicación, es el receptor.

Medir: Se refiere a verificar el estado presente o continuo de las variables en el funcionamiento de una planta.

MDI (Multiple Documents Interface): Interfase de múltiples documentos, es la que permite una mejor visualización de los diferentes procesos supervisados por el SCADA, la cual es característica de los programas basados en plataforma Windows.

Periféricos: permiten el enlace entre las diferentes interfaces para poder establecer la comunicación entre los distintos equipos. Funcionan como amplificadores de señales digitales.

Potencia Aparente (S): Se expresa en voltamperios (VA) y se determina a partir de las indicaciones de un amperímetro y un voltímetro. Está expresada por: $S = V I$. Cuando se trata de una instalación de corriente alterna trifásica, la potencia aparente se calcula por la formula: $S = \sqrt{3} V I$.

Potencia Activa (P): Llamada también potencia efectiva o potencia real, que se expresa en vatios (W) y se mide directamente con vatímetro. Solamente esta potencia se puede transformar en potencia mecánica o en potencia calorífica y, por consiguiente, solamente esta potencia es la que interesa al usuario. Esta expresada

por $P = V I \cos \phi$. Si se trata de una instalación de corriente alterna trifásica, la potencia activa esta expresada por $P = \sqrt{3} V I \cos \phi$.

Potencia Reactiva (Q): Llamada también potencia magnetizante, que se expresa en voltamperios reactivos (VAR) y que se calcula a partir de las otras potencias. Resulta necesaria para el funcionamiento de ciertas máquinas y dispositivos eléctricos (motores, transformadores, bobinas, relés, etc.) pero no puede transformarse en potencia mecánica o calorífica útil, esta expresada por $Q = V I \sin \phi$. En el caso de una instalación de corriente alterna trifásica, la potencia reactiva esta expresada por $P = \sqrt{3} V I \sin \phi$.

SCADA (Supervisory Control And Acquisition Data): sistema de control de lazo cerrado, el cual posee las características de controlar de forma supervisoria y por medio de la adquisición de datos que luego pueden ser almacenados como registros del estado de un sistema determinado. En estos sistemas el operario desempeña una función fundamental: cierra el lazo de retroalimentación del sistema de control.

Script: párrafos de programación que se adicionan a los comandos de las distintas funciones SCADA en el sistema, para que ejecuten ciertas acciones, por ejemplo las alarmas se programan dentro de un rango de valores por medio de un script.

Servidor: Computadora que ofrece sus prestaciones a varios ordenadores clientes conectados a una red.

Sistema de distribución: Conjunto integrado de equipos de transporte de energía eléctrica en media y baja tensión. Está formado por los circuitos que se inician en la subestación de distribución y suministran energía a los transformadores de distribución.

Slave: del ingles esclavo, dispositivo que es interrogado y no actúa sino existe un estímulo por parte del maestro.

Subestación: Conjunto de equipos instalados en un lugar, y las obras civiles en el mismo, para la conversión, transformación o control de la energía eléctrica, y para la conexión entre dos o más circuitos.

SSL (Security Socket Layer): protocolo de seguridad que permite tener un acceso controlado y supervisado de un sitio de Internet en particular, el cual posee información que no puede ser divulgada o verificada por cualquier usuario de Internet, sino solo por aquellos que posean una llave de acceso.

Telemetría: Es una manera avanzada y precisa de verificar el estado de una planta a distancia.

THD: La **distorsión armónica** es un parámetro técnico utilizado para definir la señal de audio que sale de un sistema. La distorsión armónica se produce cuando la señal de salida de un sistema no equivale a la señal que entró en él. Esta falta de linealidad afecta a la forma de la onda, porque el equipo ha introducido armónicos que no estaban en la señal de entrada. Puesto que son armónicos, es decir múltiplos de la señal de entrada, esta distorsión no es tan disonante y es menos fácil de detectar.

Usuario final o consumidor: Es quien compra la energía eléctrica para uso propio.

WAN: red de área amplia (Wide Área Network), redes que poseen una mayor cobertura que las de área local, dentro de las cuales se encuentra la red Internet.