

PRAT, RODRIGO GASTON
Protocolos de comuni

2012

71200

71200

MFN:
Clasif:
T-762

71200



**PROCOLOS DE COMUNICACIÓN
APLICADOS A SISTEMAS DE GESTIÓN DE
ENERGÍA**

por

Rodrigo Gastón Prat

Tesis presentada para la obtención del grado de Magíster en
Ciencias de la Ingeniería
Mención: Ingeniería Eléctrica

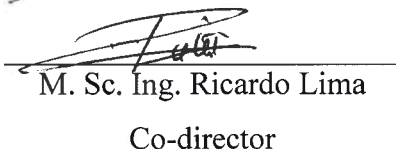
**FACULTAD DE INGENIERÍA
UNIVERSIDAD NACIONAL DE RÍO CUARTO
MAYO 2012**

PROTOCOLOS DE COMUNICACIÓN APLICADOS A SISTEMAS DE GESTIÓN DE ENERGÍA

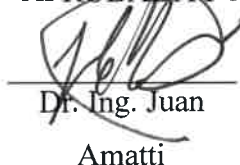
por

Rodrigo Gastón Prat

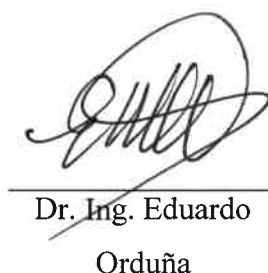

Dr. Ing. Fernando Magnago
Director

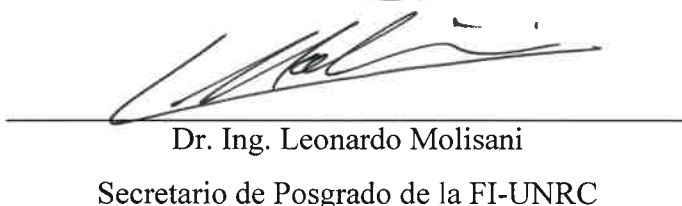

M. Sc. Ing. Ricardo Lima
Co-director

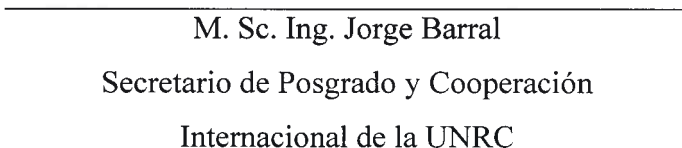
APROBADA POR:


Dr. Ing. Juan
Amatti


M. Sc. Ing. Fernando
Corteggiano


Dr. Ing. Eduardo
Orduña


Dr. Ing. Leonardo Molisani
Secretario de Posgrado de la FI-UNRC


M. Sc. Ing. Jorge Barral
Secretario de Posgrado y Cooperación
Internacional de la UNRC

*Dedico este trabajo a mis padres, y hermana por el apoyo y la comprensión que me han
brindado*

AGRADECIMIENTOS

El autor expresa su agradecimiento al Dr. Ing. Fernando Magnago, Director de la tesis, por haber ayudado de muchas formas a terminar este trabajo.

Al Ing. Gabriel Competelli y al Dr. Ing. Juan C. Amatti por el apoyo.

A la Dra. Lic. en Ciencias de la Educación Rita Lilián Amieva por el apoyo en la corrección gramatical de la tesis.

Al Grupo de Análisis de Sistemas Eléctricos de Potencia (GASEP) del Departamento de Electricidad y Electrónica de la Facultad de Ingeniería, en cuyo ámbito se generó esta línea de trabajo. Al Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), a la Secretaría de Ciencia y Técnica de la UNRC y a la Agencia Nacional de Promoción Científica y Tecnológica por el apoyo económico brindado

ÍNDICE

ÍNDICE	5
ÍNDICE DE TABLAS	8
ÍNDICE DE FIGURAS	9
RESUMEN	12
ABSTRACT	13
I. INTRODUCCIÓN	14
1.1. Descripción del problema	14
1.2. Objetivos	17
1.2.1. Objetivo general	17
1.2.2. Objetivos particulares	17
1.3. Metodología utilizada.....	17
1.4. Estructura de la tesis.....	18
II. SISTEMAS DE SUPERVISIÓN CONTROL Y ADQUISICIÓN DE DATOS	
APLICADOS A SISTEMAS DE GESTIÓN DE ENERGÍA	19
2.1. Introducción	19
2.2. Evolución de los sistemas SCADA.....	19
2.3. Arquitectura física de un SCADA.....	22
2.3.1. La estación maestra	22
2.3.2. RTU	22
2.3.3. IED	23
2.4. Arquitectura de software de los SCADA.....	24
2.5. Redes SCADA y redes LAN.....	24
2.6. Conclusiones	25
III. SISTEMAS DE COMUNICACIÓN UTILIZADOS EN SISTEMAS SCADA	26
3.1. Introducción	26
3.2. Introducción a los sistemas de comunicación	26
3.2.1. Modelo de referencia de Interconexión de Sistemas Abiertos	26
3.2.2. Tecnologías de red.....	34
3.3. Paradigmas de comunicación en sistemas SCADA	42
3.4. Evolución de los sistemas de comunicación en sistemas SCADA	43
3.5. Protocolos de comunicación más utilizados en sistemas SCADA.....	45
3.6. Conclusiones	48

IV. REALIZACIÓN DE UN SISTEMA SCADA PARA LA SUBESTACIÓN PRINCIPAL DE LA UNRC UTILIZANDO MODBUS TCP	49
4.1. Introducción	49
4.2. Marco Teórico	50
4.3. Sistema de distribución de la UNRC	51
4.3.1. Localización de las subestaciones	51
4.3.2. Equipamiento del sistema de distribución de la UNRC	53
4.3.3. Equipamiento de protección y monitoreo de la subestación IPSEP.....	55
4.3.4. Selección de los módulos de comunicación para el equipamiento de protección y monitoreo de la subestación IPSEP	56
4.4. Arquitectura de software	58
4.4.2. Resultados	63
4.5. Conclusiones	65
V. SISTEMAS DE AUTOMATIZACIÓN EN SISTEMAS SCADA.....	67
5.1. Introducción	67
5.2. Introducción a la automatización de subestaciones eléctricas	67
5.3. Definiendo un estándar para la automatización de subestaciones eléctricas	69
5.3.1. Interfaces de comunicación en un sistema de automatización de subestaciones	70
5.3.2. Requerimientos físicos para el sistema de comunicación	71
5.4. Introducción al estándar IEC61850.....	72
5.4.1. Independencia entre el sistema comunicación y las aplicaciones	72
5.4.2. Parámetros, herramientas de ingeniería y nodos lógicos	73
5.5. Conclusiones	76
VI. DESCRIPCIÓN DE LA NORMA IEC61850	77
6.1. Introducción	77
6.2. Interface Abstracta de los Servicios de Comunicación.....	78
6.2.1. Modelos de información básicos.....	79
6.2.2. Modelos de intercambio de información.....	81
6.2.3. Servicios provistos por el ACSI	82
6.2.4. Definiciones de tipo de dato	83
6.2.5. Generalidades en la definición de los modelos de las clases más importantes	86
6.2.6. Tiempo y modelo de sincronización de tiempo.....	88
6.3. Mapeo a Servicio de Comunicación Específico - Mapeo a MMS	89

6.3.1. Mapeo de modelos de datos y servicios ACSI al protocolo MMS	90
6.3.2. Protocolos de comunicación.....	92
6.4. Lenguaje de Configuración de Subestaciones.....	94
6.4.1. Importancia del lenguaje SCL en el proceso de ingeniería	94
6.4.2. El modelo de objetos SCL.....	95
6.4.3. Sintaxis y elementos de los archivos SCL.....	100
6.4.4. Tipos de Archivos SCL	100
6.4.5. Secciones presentes en cada tipo de archivo SCL.....	101
6.5. Conclusiones	102
VII. DISEÑO GENERAL DE UN SISTEMA DE AUTOMATIZACIÓN DE	
 SUBESTACIONES BAJO LA NORMA IEC61850 EN LAS	
 SUBESTACIONES DE LA UNRC	103
7.1. Introducción	103
7.2. Diagrama esquemático del sistema de automatización	103
7.3. Generación de archivos SCL.....	105
7.4. Selección de equipos	107
7.5. Diseño e Implementación piloto del IED.....	107
7.5.1. Parte servidor - Arquitectura de software.....	110
a) Módulo Principal	111
b) Módulo de Inicialización SCL	111
c) Módulo del Modelo de Información	112
d) Módulo del Modelo de Comunicación	114
7.5.2. Arquitectura de software de la parte cliente	115
7.5.3. Prueba de los servicios implementados en el servidor	115
7.6. Conclusiones	117
VIII. CONCLUSIONES.....	119
8.1. Introducción	119
8.2. Trabajos futuros	120
8.3. Aportes realizados	121
REFERENCIAS.....	122
ANEXOS.....	127

ÍNDICE DE TABLAS

Tabla III-1 - Tecnologías Ethernet.....	38
Tabla III-2 - Protocolo Modbus en el modelo OSI	46
Tabla III-3 - Estándar IEC 60870-5-101 en el modelo OSI.....	47
Tabla III-4 - Estándar DNP 3.0 en el modelo OSI.....	47
Tabla IV-1 - Datos Modbus	51
Tabla IV-2 - Códigos de las Funciones Modbus.....	51
Tabla V-5 - Grupos de Nodos Lógicos y sus funciones.....	75
Tabla VI-1 - Tipos de mensajes a mapear	90
Tabla VI-2 - Mapeo de clases de modelos de datos ACSI a clases MMS	90
Tabla VI-3 - Mapeo de los servicios ACSI a servicios MMS.....	91
Tabla VI- 4 - Secciones en los archivos SCL	101

ÍNDICE DE FIGURAS

Figura 2-1 - Sistema SCADA tipo panel-sensores.....	19
Figura 2-2 - Sistema SCADA DCS.....	21
Figura 2-3 - Sistema SCADA con IED	21
Figura 2-4 - Hardware de una RTU	23
Figura 3-5 - Niveles del modelo OSI	28
Figura 3-6 - Topologías de red.....	29
Figura 3-7 - Unidades de datos del modelo OSI	32
Figura 3-8 - Comunicación - Modelo OSI	33
Figura 3-9 - Correspondencia entre el modelo TCP/IP y el modelo OSI	34
Figura 4-1 - Pila de protocolos utilizada en Modbus	50
Figura 4-2 - Ubicación geográfica de las subestaciones de la UNRC	52
Figura 4-3 -Diagrama unifilar de las subestaciones de la UNRC	53
Figura 4-4 - Subestación IPSEP con el equipamiento de protección y monitoreo instalado	56
Figura 4-5 - Conexión de los equipos de la subestación IPSEP	57
Figura 4-6 - Arquitectura de software del sistema SCADA	59
Figura 4-7 - Bloque Modbus TCP de Labiew	59
Figura 4-8 - Diagrama de bloques de una aplicación cliente Modbus TCP en Labview	61
Figura 4-9 - Panel frontal de una aplicación cliente Modbus TCP en Labview	61
Figura 4-10 - Registros del Power Logic CM4250	62
Figura 4-11 - Diagrama de bloques de una aplicación parcial en Labview	62
Figura 4-12 - Panel frontal de la aplicación final.....	63
Figura 4-13 - Página web con los resultados	64
Figura 4-14 - Perfil de tensión semanal	65
Figura 4-15 - Perfil de carga activa semanal.....	65
Figura 5-10 - Arquitectura de comunicación de un sistema de automatización de subestaciones.....	69
Figura 5-11 - Interfaces de comunicación en un sistema de automatización de subestaciones.....	71

Figura 5-12- Interfaces físicas y lógicas de un Sistema de Automatización de Subestaciones	71
Figura 5-13 - Conceptos, ACSI y SCSM	73
Figura 5-14 - Herramientas de Ingeniería	74
Figura 6-1 - Modelos de información y de servicios ACSI	79
Figura 6-2 - Modelos de información ACSI básicos	80
Figura 6-3 - Modelo de un IED.....	80
Figura 6-4 - Modelo de intercambio de información ASCII.....	82
Figura 6-5 - Composición de los datos	83
Figura 6-6 - Ejemplo de la composición de datos.....	84
Figura 6-7 - Ejemplo de una instancia de dato.....	85
Figura 6-8 - Estructura jerárquica de las clases	87
Figura 6-9 - Diagrama de bloques de los modelos de las clases más importantes.....	88
Figura 6-10 - Modelo de sincronización de tiempo IEC61850.....	88
Figura 6-11 - Perfil de protocolos de comunicación IEC61850	89
Figura 6-12 - Perfiles A y T especificados por la norma	92
Figura 6-13 - Transferencia de archivos y parametrización.....	95
Figura 6-14 - Puntos de acceso de los buses de proceso y de estación de un IED	98
Figura 6-15 - Referenciamiento de los objetos de datos a nivel de aplicación	99
Figura 6-16 - Ejemplo de la estructura de un IED	99
Figura 6-17 - Referenciamiento a un dato específico	100
Figura 7-1 - Diagrama esquemático del Sistema de Automatización de Subestaciones reducido.....	104
Figura 7-2 - Diagrama generado con KALKI SCL MANAGER	106
Figura 7-3 - Esquema general para el Sistema de Automatización de Subestaciones	108
Figura 7-4 - Dispositivos reales y dispositivos virtuales.	109
Figura 7-5 - Diagrama UML que indica las partes cliente y servidor en el sistema. .	110
Figura 7-6 - Diagrama UML del modelo de información ACSI básico.	113
Figura 7-7 - Diagrama UML del intercambio de información para el método GetServerDirectory de la clase Server.	113
Figura 7-8 - Código en c++ del método GetServerDirectory de la clase Server.	114
Figura 7-9 - Diagrama UML de la clase Communication.....	115
Figura 7-10 - Cliente desarrollado en Matlab.	116
Figura 7-11 - Cliente desarrollado en c++.	116

Figura 7-12 - Consola de la parte servidor del IED. 117

RESUMEN

En los últimos años, los Sistemas de Gestión de Energía aplicados a Sistemas Eléctricos de Potencia (SEP) han presentado una importante evolución. Esta evolución implica nuevos desafíos y oportunidades tanto para las redes eléctricas, como así también, para los sistemas informáticos encargados de su gestión y los sistemas de comunicación asociados.

Actualmente, dentro de la estructura de los Sistemas de Gestión de Energía aplicados a SEP, el manejo de información representa uno de los factores más importantes al momento de proporcionar energía de manera confiable a los consumidores finales. Esto requiere de un sistema de comunicación cada vez más avanzado que se adapte a las necesidades actuales y requerimientos futuros. Por lo tanto, el diseño de redes de comunicación confiables, rápidas y de bajo costo constituyen el centro de atención en las aplicaciones relacionadas con estos sistemas.

Atendiendo a esta situación, el objetivo de este trabajo de tesis de maestría, es analizar la evolución de los sistemas de comunicación utilizados en los Sistemas de Gestión de Energía aplicados a SEP, y basado en el estado del arte en los mismos, desarrollar herramientas de programación relacionadas con estos sistemas, utilizando diferentes plataformas y estándares de comunicación. En este trabajo, el sistema de distribución eléctrica de la Universidad Nacional de Río Cuarto es utilizado como sistema de prueba.

ABSTRACT

In the past years, the Energy Management Systems applied to Electrical Power Systems (EPS), face important transformations. These transformations imply new challenges and opportunities for different fields such as electrical power networks, information technology and communication systems.

Nowadays, within the structure of an Energy Management system applied to EPS, the information management represents one of the key factors in order to provide energy to customers in a reliable manner. Therefore, this task requires having an advance communication system adapted to actual and future requirements. Hence, the design of low cost, reliable and fast communications networks becomes one of the most important part of the application design related with this type of systems.

Taking into account this scenario, the aim of this work is to analyze the evolution of the communications systems used by the Energy Management Systems applied to SEP. Based on this analysis, different software tools are designed and developed using different communications platforms and standards. In this work, the Electrical Distribution System of Universidad Nacional de Río Cuarto is used as the test system.

I. INTRODUCCIÓN

1.1. Descripción del problema

En la actualidad, una de las transformaciones más importantes que está experimentando el sistema eléctrico es pasar de una estructura centralizada, donde la red eléctrica es controlada por los productores de energía, hacia una red mucho más descentralizada e interactiva desde el punto de vista de los consumidores.

Estos cambios están asociados a la actual tendencia de utilizar Recursos de Energía Distribuidos (DER, por sus siglas en inglés, *Distributed Energy Resources*), sistemas de comunicación avanzados y tecnologías de control para generar electricidad de una manera más eficiente, de forma más económica, con menos intensidad de emisiones de gases de efecto invernadero y en respuesta a las necesidades de los consumidores. La convergencia de estos factores jugará un papel primordial para ofrecer una solución real a la demanda creciente de energía, mejorar la eficiencia en su uso e incorporar generación distribuida.

Todo parece indicar que la tendencia actual del mercado eléctrico converge a lo que se denomina Red Inteligente (SG, por sus siglas en inglés, *Smart Grid*) [1]. Este nuevo concepto cambia el paradigma de generación a un esquema bidireccional, donde los consumidores pueden ser a la vez productores de electricidad e introducir sus excedentes en la red [2] [3]. Esto es factible gracias a la integración de sistemas informáticos, tecnologías de comunicación, sensores y dispositivos de medición, que posibilitan a los servicios públicos de electricidad y a los consumidores controlar con precisión la entrega y el uso de la energía eléctrica.

De esta forma, surge el problema de que aun contando con adecuadas planificaciones e inversiones, la dinámica del sistema obliga a una utilización más eficiente y racional de la energía. Esto implica potenciar tanto desde el punto de vista tecnológico como de investigación a todos aquellos medios tendientes a una eficiente integración de los recursos disponibles, lo cual solo es posible si se optimiza cada uno de los componentes del sistema. Para ello es necesario que se disponga —además de los constituyentes principales para la generación, transmisión y distribución de energía eléctrica— de una eficaz gestión del sistema en su conjunto.

Así, las nuevas Tecnologías de Comunicación e Informática (ICT, por sus siglas en inglés, *Information and Communication Technology*) juegan un rol fundamental en la gestión y el control de la red.

De esta manera, los Sistemas de Gestión de Energía eléctrica (EMS, por sus siglas en inglés, *Energy Management System*), o más específicamente los Sistemas de Gestión de Distribución eléctrica (DMS, por sus siglas en inglés, *Distribution Management System*), no solo se deben ocupar de lo que sucede en las subestaciones principales, como ocurre actualmente, sino también, administrar los lugares de generación distribuida y de consumo domiciliario.

En este escenario, resulta imprescindible disponer de un sistema de telecontrol que utilice sistemas de comunicación eficientes. Este sistema de telecontrol utiliza una red de transmisión de datos entre los distintos despachos eléctricos que permite efectuar acciones relacionadas con operación a distancia, detección de alarmas, vigilancia e intercambio de información corporativa. El sistema encargado de realizar estas tareas dentro de un DMS es el Sistema de Supervisión, Control y Adquisición de Datos (SCADA, por sus siglas en inglés, *Supervisory Control And Data Acquisition*), el mismo consiste en la combinación de telemetría y adquisición de datos, y es un componente que cumple un rol fundamental en un EMS. El SCADA, para poder realizar las acciones mencionadas anteriormente, adquiere los datos a través de Unidades Terminales Remotas (RTU, por sus siglas en inglés, *Remote Terminal Unit*).

Tradicionalmente, los sistemas de comunicación utilizados en sistemas SCADA, eran propietarios y cerrados; es decir, no existían estándares abiertos, de forma que el desarrollo entre diferentes fabricantes era incompatible, no existía interoperabilidad [4].

Con la evolución de las diferentes tecnologías y la necesidad de estandarización por parte de los usuarios, este escenario fue cambiando [5]. Ya los fabricantes no sólo se interesaban en desarrollar sus sistemas sino también en que los mismos fueran interoperables.

Al principio, estandarizaron las interfaces físicas, y si bien los dispositivos se podían conectar, los protocolos de comunicación que utilizaban eran diferentes [6].

Hasta 1978 no había sistemas estandarizados, todos eran propietarios, cerrados y muy básicos, y existían pocos fabricantes de equipos; en aquel año la empresa Modicon lanzaba el protocolo Modbus, que tuvo una gran aceptación en el área.

En 1994, el Instituto de Investigación de Potencia Eléctrica (EPRI, por sus siglas en inglés, *Electric Power Research Institute*), empezó a trabajar en un proyecto denominado Arquitectura de Comunicaciones de Utilidad (UCA, por sus siglas en inglés, *Utility Communications Architecture*) que intentaba solucionar los problemas de estandarización y agregaba automatización al sistema de gestión. Para este proyecto, el EPRI se basó en un conjunto de estándares internacionales, que venían siendo utilizados en los sistemas de comunicación del área de computación y redes de datos [7].

En 1996, el Comité Técnico 57 (TC por sus siglas en inglés, *Technical Committee 57*) de la Comisión Electrotécnica Internacional (IEC, por sus siglas en inglés, *International Electrotechnical Commission*) empezó a trabajar en un proyecto similar, bajo la denominación “IEC61850” [8].

En 1997, ambos grupos se unieron para definir un estándar internacional que sigue el nombre del segundo proyecto [9], “IEC61850”, el cual fue lanzado finalmente en 2004. Este estándar representa el estado del arte en automatización de subestaciones.

En la actualidad, la norma IEC61850 está siendo utilizada por muchos fabricantes de dispositivos, y se espera que gracias a la aceptación que está teniendo en el mercado, el desarrollo de la misma sea algo dinámico que impulse a la mejora del funcionamiento del sistema eléctrico gracias al objetivo principal de la norma que persigue la interoperabilidad entre dispositivos de automatización de diferentes fabricantes [10].

Las tecnologías de red más utilizadas para los sistemas que trabajan bajo esta norma están basadas en redes Ethernet de alta velocidad que funcionan sobre fibra óptica [11].

Se puede deducir de la evolución de los sistemas de gestión de energía, que actualmente, los sistemas de comunicación en el área son fundamentales, es por ello que este trabajo de tesis se centra en investigar el estado del arte en estos sistemas de comunicación; más específicamente, lo referido a las tecnologías y los estándares de comunicación utilizados en DMS y SCADA, incluyendo áreas relacionadas tales como Automatización de Subestaciones y Redes Inteligentes. Con relación a estos estándares de comunicación, en este trabajo de tesis se describe la implementación de una aplicación piloto que trabaja con la norma IEC61850. Se trata, además, de forma teórica y práctica la manera en que se aplican los estándares de comunicación a los sistemas de protección y monitoreo de las redes eléctricas. Se describe también, la implementación de un sistema SCADA real para los equipos de protección que están actualmente siendo

utilizados en la subestación principal de la UNRC. Los mismos trabajan con el protocolo de comunicaciones Modbus TCP, el cual, si bien no es un protocolo recientemente desarrollado, el mismo se ha adoptado como protocolo de comunicación en muchas implementaciones de sistemas SCADA actuales.

En la siguiente sección se detallan los objetivos principales de este trabajo de tesis.

1.2. Objetivos

1.2.1. Objetivo general

Este trabajo tiene por objetivo general desarrollar herramientas relacionadas con los sistemas de comunicación en sistemas de distribución de energía eléctrica, utilizando diferentes plataformas y estándares de comunicación.

1.2.2. Objetivos particulares

- Desarrollar de un sistema SCADA real para los equipos de la subestación principal de la UNRC.
- Desarrollar una aplicación piloto bajo la norma IEC61850 que representa el estado del arte en sistemas de automatización de subestaciones.

Estos dos objetivos servirán además, como referencia para diseñar una materia optativa de comunicaciones en la carrera Ingeniería Eléctrica de la UNRC, como así también, para trabajos futuros en el área.

1.3. Metodología utilizada

En primer lugar, se realiza una revisión bibliográfica a efectos de tener un conocimiento más profundo en el tema y poder contrastar las diferentes alternativas que se están utilizando en la actualidad. Esto ha permitido conocer con exactitud cuáles son los requerimientos en los sistemas de comunicación utilizados en sistemas SCADA.

La metodología de validación de protocolos implementada, es del tipo experimental, evaluándose la robustez de los mismos en una red compuesta por dispositivos afines.

Se realiza la simulación de las plataformas estudiadas para analizar la performance del sistema piloto en el lenguaje de programación C++.

Además se describe el desarrollo de un sistema SCADA en Labview totalmente funcional para los equipos de protección y monitoreo instalados en las subestaciones de la UNRC, y se comparan los resultados obtenidos con soluciones comerciales para corroborar el correcto funcionamiento del sistema.

1.4. Estructura de la tesis

El presente trabajo está organizado de la siguiente manera: en el Capítulo 2 se presentan los conceptos básicos de sistemas SCADA, analizando la evolución que tuvieron los mismos a lo largo del tiempo. En el Capítulo 3, se analizan los diferentes sistemas de comunicación utilizados en sistemas SCADA, y se tratan también aspectos generales de los sistemas de comunicación, necesarios para comprender el tema. En el Capítulo 4 se describe un sistema SCADA desarrollado para la subestación principal de la UNRC. En el Capítulo 5 se presenta el concepto de automatización de subestaciones, detallando las estructuras de los sistemas y definiendo aspectos básicos relacionados con la temática. En el Capítulo 6 se presenta la norma que representa el estado del arte en sistemas de automatización de subestaciones. En el Capítulo 7, se realiza el diseño y la implementación de un sistema piloto que trabaja bajo los conceptos de dicha norma. Finalmente, en el Capítulo 8, se presentan las conclusiones.

II. SISTEMAS DE SUPERVISIÓN CONTROL Y ADQUISICIÓN DE DATOS APLICADOS A SISTEMAS DE GESTIÓN DE ENERGÍA

2.1. Introducción

Se define como Sistema de Supervisión, Control y Adquisición de Datos (SCADA, por sus siglas en inglés, *Supervisory Control And Data Acquisition*) a una aplicación de software que permite el acceso a datos de dispositivos remotos y el control de los mismos utilizando algún sistema de comunicación. Si bien en sus comienzos eran simples, a medida que crecen las necesidades de monitorear y controlar en la planta, los mismos se hacen cada día más complejos, tanto el SCADA en sí como así también los dispositivos utilizados en los lugares remotos, lo que se puede ver como una evolución de software, hardware y de comunicaciones.

En este capítulo se tratarán los SCADA. En la Sección 2.2 se expondrá la evolución de tales sistemas. A continuación, en la Sección 2.3 se presentará la arquitectura física de un SCADA, detallando los elementos utilizados por el mismo; en la Sección 2.4 se describirán las diferentes arquitecturas de software; en la Sección 2.5 se presentarán las redes de comunicación que se utilizan en estos sistemas. Finalmente, en la Sección 2.6 se efectuarán las conclusiones del capítulo.

2.2. Evolución de los sistemas SCADA

Los primeros SCADA mostraban y operaban los datos por medio de medidores de paneles, luces e interruptores [12]. Tanto el control como la supervisión eran realizados por el operador, que manualmente operaba varios controles con botones y llaves. La Figura 2-1 esquematiza un sistema de este tipo.

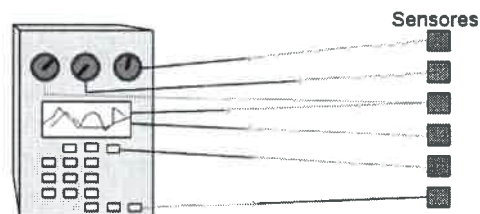


Figura 2-1 – Sistema SCADA tipo panel-sensores.

Estos sistemas presentaban las ventajas de ser analógicos y simples, sin necesidad de contar con un microprocesador, memoria o algún software.

Las desventajas asociadas a los primeros SCADA eran varias; las más relevantes: que una persona debía monitorearlos las veinticuatro horas; los tipos de datos eran muy básicos; cuando la cantidad de sensores era grande, la cantidad de cable crecía aún más haciendo muy complejo su manejo; el grabado de datos era mínimo; no había alarmas o monitoreo de forma remota; y era difícil reconfigurar el sistema ante una ampliación o modificación del mismo [13]. En algunos casos, se siguen utilizando en paralelo con sistemas más modernos ante una eventual falla de estos últimos.

Desde un punto de vista más actual, SCADA se refiere a la combinación de telemetría y adquisición de datos.

La Telemetría es necesaria para conectar equipos y sistemas separados por largas distancias, las cuales podrían ir hasta miles de kilómetros. Se utiliza para enviar comandos desde un sitio central y recibir información de monitoreo desde ubicaciones remotas.

Así, el SCADA se encarga de adquirir los datos a través de Unidades Terminales Remotas (RTU, por sus siglas en inglés, *Remote Terminal Unit*) [14], que envían los datos a un sitio central sin ningún procesamiento o control previo y opcionalmente los muestra en alguna pantalla o display local. Las acciones de control requeridas se realizan luego en un centro destinado a tal efecto, en donde a su vez existen sistemas gráficos de mayor complejidad.

Con la aparición del microprocesador y otros dispositivos electrónicos, los fabricantes incorporaron la electrónica digital en sus equipos creando el Controlador Lógico Programable (PLC, por sus siglas en inglés, *Programmable Logic Controller*), el cual es aún uno de los más utilizados en sistemas de control en la industria. Aunque inicialmente las RTU eran utilizadas en un sistema SCADA, los PLC a menudo son empleados en su lugar [15].

Como las necesidades de monitorear y controlar en la planta crecieron, estos dispositivos fueron utilizados de forma distribuida. Además, el sistema central se hizo más inteligente y pequeño. Nació el concepto de Sistema de Control Distribuido (DCS, por sus siglas en inglés, *Distributed Control System*) [16], que se ilustra en la figura 2-2.

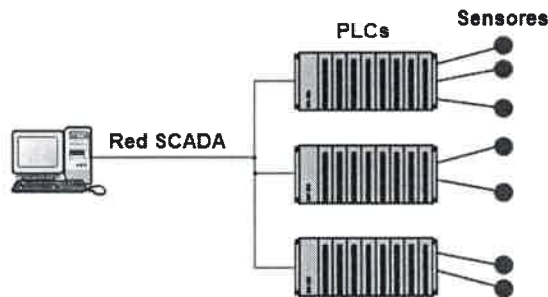


Figura 2-2 - Sistema SCADA DCS

Las ventajas más importantes de un sistema SCADA DCS son: que se pueden grabar los datos; una gran cantidad de sensores distribuidos en una área amplia pueden ser conectados al sistema minimizando el cableado; se pueden incorporar simulaciones del sistema en tiempo real, muchos tipos de datos pueden ser adquiridos de dispositivos remotos; y se puede acceder a la información desde muchos lugares.

Como desventajas podríamos decir que es un sistema más complicado, que requiere personal capacitado y que el operador no tiene acceso directo a los sensores sino que se limita hasta los PLC.

Mientras los requerimientos por sistemas más inteligentes y más pequeños crecían, los sensores fueron diseñados con la inteligencia de los PLC. Estos dispositivos son conocidos como Dispositivos Electrónicos Inteligentes (IED, por sus siglas en inglés, *Intelligent Electronic Device*) [17]. Los mismos tienen la capacidad de adquirir datos, comunicarse con otros dispositivos y correr un programa. Pueden tener mas de un sensor, típicamente un IED puede combinar sensores de entrada/salida analógica/digital y sistema/s de comunicación. Éste tipo de sistema SCADA se esquematiza en la Figura 2-3.

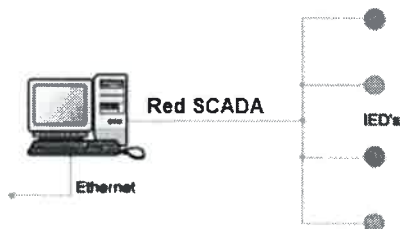


Figura 2-3 - Sistema SCADA con IED

Las ventajas de este tipo de sistema son que el operador puede llegar hasta los sensores, los dispositivos son *plug and play* facilitando la instalación y el reemplazo, y

los equipos son más pequeños. Mientras que las desventajas consisten en un mayor costo del equipamiento y la necesidad de disponer de personal más entrenado y capacitado.

2.3. Arquitectura física de un SCADA

Si bien, como se presentó en la sección anterior, los dispositivos de adquisición de datos utilizados por un SCADA son muy variados dependiendo de su tecnología, para explicar su arquitectura física hablaremos de las RTU.

Así, un sistema SCADA puede constar de un número de RTU que recogen los datos y los envían a una estación maestra local a través de un sistema de comunicaciones. La estación maestra muestra los datos adquiridos; también permite al operador realizar tareas de control y supervisión en forma remota; y opcionalmente permite el envío de los datos adquiridos a un centro de control, en donde se realizan tareas de planificación, entre otras.

Las RTU proveen a la estación maestra una interface a los sensores digitales y analógicos de campo situados en cada sitio remoto.

El sistema de comunicación provee el vínculo entre la estación maestra y los sitios remotos. Este vínculo puede ser cable, fibra óptica, radio, línea telefónica, etc. Para una transferencia de datos óptima y eficiente, se utilizan protocolos específicos.

La estación maestra recoge los datos de varias RTU y generalmente provee una interfaz al operador para mostrar la información y controlar los sitios remotos [6].

2.3.1. La estación maestra

Una estación maestra tiene como funciones principales obtener datos de campo de las RTU y controlarlas remotamente. Su interface gráfica está compuesta de alarmas, elementos que representan a los interruptores, registro de eventos, entre otros.

2.3.2. RTU

Una RTU es una unidad de control y adquisición de datos que generalmente cuenta con un microprocesador. Es la encargada de monitorear y controlar equipos de campo en una ubicación remota, hace de interface entre éstos y la estación maestra. Generalmente, también tiene la capacidad de guardar su configuración o poder descargarla desde la estación maestra.

Aunque tradicionalmente solo se comunica con la estación maestra, también es posible la comunicación entre ellas.

El hardware típico de una RTU incluye un microprocesador, memoria, entradas y salidas analógicas y digitales, e interfaces de comunicación. Esto se muestra en la Figura 2-4 [6].

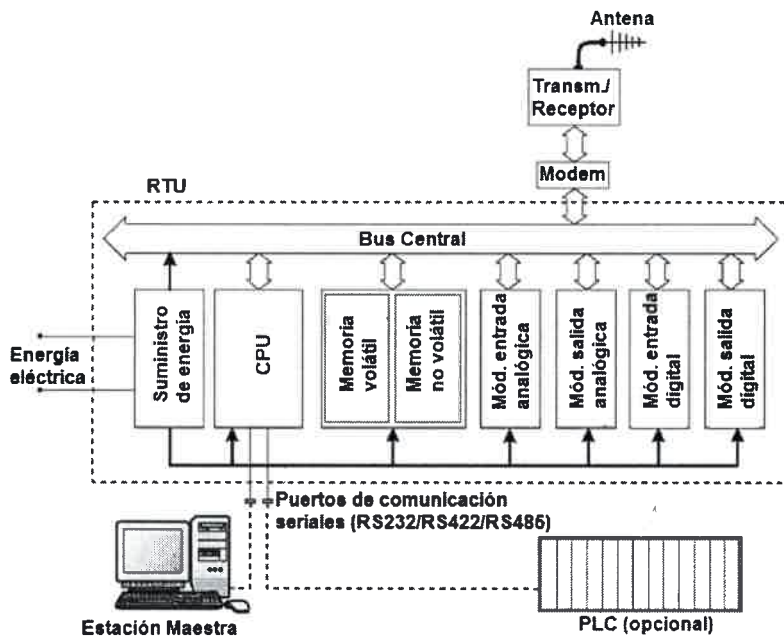


Figura 2-4 - Hardware de una RTU

2.3.3. IED

De forma general, cualquier dispositivo electrónico que posea algún tipo de inteligencia podría ser llamado IED. Muchos relés pueden realizar las funciones de protección, control y monitoreo, pero necesitan la asistencia de una RTU o procesador de comunicaciones para conectarse con un SCADA maestro. En éste caso se puede llamar relé inteligente pero no IED. De igual manera, algunos relés modernos pueden comunicarse directamente con un SCADA pero carecen de funcionalidades de control. Estos relés son comúnmente utilizados en conjunto con otros dispositivos que proveen las funcionalidades de control necesarias para formar un sistema de automatización. Nuevamente, estos relés no podrían ser llamados IED.

En la industria de automatización de sistemas eléctricos de potencia y protección, el término realmente intenta describir a un dispositivo que tiene funciones de protección, inteligencia de control local avanzada, habilidades de monitoreo y la

capacidad de comunicarse directamente con un SCADA y otros IED. La capacidad de un IED de realizar todas las funciones de protección, control, monitoreo, medición y comunicaciones de alto nivel de forma independiente y sin la ayuda de otros dispositivos como una RTU es lo que se define como IED en sistemas de automatización de subestaciones [6] [18].

2.4. Arquitectura de software de los SCADA

Los componentes típicos de un software SCADA incluyen: interfaz de usuario con alarmas y gráficos, interfaces con varias RTU e IED, acceso a los datos, bases de datos, escalabilidad, trabajo en red, tolerancia a fallas, redundancia y procesamiento distribuido cliente/servidor [6].

Una clasificación que se puede realizar con respecto a arquitectura de software es: propietaria y abierta.

Las empresas de SCADA más conocidas en el área desarrollan software propietario para comunicar su hardware. Estos sistemas son vendidos como soluciones llave en mano. El problema principal de los mismos es que el único proveedor es la empresa fabricante, con todos los contras que eso conlleva, como pueden ser altos costos de mantenimiento y actualización, indisponibilidad de productos, falta de soporte, etc.

Como se mencionó anteriormente, también existen sistemas de software abierto, en [19] podemos encontrar alguna de las soluciones de este tipo para sistemas SCADA.

2.5. Redes SCADA y redes LAN

Las redes sirven para compartir información y recursos. Para permitir que todos los nodos de una red compartan información, los mismos deben estar conectados por algún medio de transmisión. La forma de conexión es conocida como topología de red.

Los nodos necesitan compartir este medio de transmisión de forma que permita a todos ellos acceder al medio sin interferir con alguna conexión ya establecida.

Una Red de Área Local (LAN, por sus siglas en inglés, *Local Area Network*) es un medio de comunicación entre dispositivos, los cuales pueden ser computadoras, servidores de archivos, terminales, y otros equipos periféricos inteligentes. Una LAN permite una completa conectividad entre todos los dispositivos de la misma.

La tecnología LAN más ampliamente utilizada en la actualidad es Ethernet [11], dado que es muy simple de utilizar. Soporta altas tasas de transferencia y es de un costo relativamente bajo.

Históricamente, las redes SCADA han sido del tipo propietarias, y su escalabilidad baja.

Las redes SCADA que utilizan RTU pueden convivir con redes LAN por medio de algún conversor de protocolos, aprovechando que esta última permite la conectividad de cualquier integrante de la compañía con el software y los permisos adecuados. Dado que los datos son mantenidos en bases de datos, el usuario puede limitarse a sólo leer la información.

En redes SCADA que tienen IED –o RTU muy actuales– se puede acceder directamente a los IED ya que los mismos son dispositivos que componen la red LAN, siempre que las políticas de red lo permitan.

2.6. Conclusiones

Desde sus comienzos, los sistemas SCADA han presentado una importante evolución, que ha involucrado aspectos relacionados con el software, el hardware y los sistemas de comunicación; temas que se trataron en este capítulo, describiendo los componentes del sistema, el tipo de software y las redes de comunicación utilizadas.

Los temas abordados en este capítulo son de gran importancia para la tesis, ya que comprenden las bases para analizar diferentes tecnologías y estándares de comunicación utilizados en los sistemas de gestión de Sistemas Eléctricos de Potencia, temas centrales del trabajo.

III. SISTEMAS DE COMUNICACIÓN UTILIZADOS EN SISTEMAS SCADA

3.1. Introducción

Los Sistemas de Comunicación (CS, por sus siglas en inglés, *Communication System*) utilizados en Sistemas de Supervisión, Control y Adquisición de Datos (SCADA, por sus siglas en inglés, *Supervisory Control And Data Acquisition*) han evolucionado dinámicamente.

Para poder entender cualquier CS, es esencial comprender aspectos básicos, los cuales se explicarán en la Sección 3.2. Inicialmente, en 3.2.1 se presentará el modelo de referencia y en la Sección 0 se detallarán las diferentes tecnologías de red. En la Sección 3.3 se mostrarán los paradigmas de comunicación que se han utilizado en los SCADA. Posteriormente, en la Sección 3.4 se expondrá la forma en que han evolucionado los CS en el área, siendo en un principio propietarios y cerrados, llegando a la actualidad con sistemas que persiguen la interoperabilidad utilizando estándares internacionales. En la Sección 3.5, se presentarán los protocolos de comunicación que fueron más utilizados en los SCADA. Finalmente, en la sección 3.6 se presentan las conclusiones del capítulo.

3.2. Introducción a los sistemas de comunicación

En las siguientes subsecciones se presentarán aspectos fundamentales que caracterizan a un CS. Para empezar, en 3.2.1 se explicará el modelo de referencia que se utiliza a la hora de definir la pila de protocolos a utilizar, entendiendo el concepto de capas de protocolos y otras características del modelo. Finalmente en 0 se verán las topologías de red más utilizadas en un CS.

3.2.1. Modelo de referencia de Interconexión de Sistemas Abiertos

En ésta sección se presentará el modelo de Interconexión de Sistemas Abiertos (OSI, por sus siglas en inglés, *Open Systems Interconnection*) que utiliza como referencia para cualquier CS, el mismo es esencial para entender el tema. La norma que define al modelo [20] será el tema central de ésta sección y las subsecciones contenidas.

A principios de la década del '80 el desarrollo de las redes de datos tuvo un crecimiento explosivo. Las mismas se agregaban o expandían casi a la misma velocidad a la que se introducían las nuevas Tecnologías de Red (NT, por sus siglas en inglés, *Network Technologies*).

Las empresas que utilizaban estas NT comenzaron a sufrir las consecuencias de la rápida expansión. De la misma forma en que las personas que no hablan un mismo idioma tienen dificultades para comunicarse, las redes que utilizaban diferentes especificaciones o implementaciones no podían intercambiar información.

Para afrontar el problema de incompatibilidad de redes, la Organización Internacional para la Estandarización (ISO, por sus siglas en inglés, *International Standardization Organization*) investigó modelos como el de la red DECnet (*Digital Equipment Corporation*, Corporación de Equipos Digitales), la SNA (*Systems Network Architecture*, Arquitectura de Sistemas de Red) y TCP/IP (*Transport Control Protocol/Internet Protocol*, Protocolo de Control de Transporte/ Protocolo de Internet), entre otros, a fin de encontrar un conjunto de reglas estándares.

En base a esta investigación, la ISO desarrolló un modelo de red que apunta a la interoperabilidad.

Las especificaciones del modelo final adoptado se encuentran en un estándar que tiene dos denominaciones, ISO/IEC 7498-1 y ITU-T X.200, ya a que el desarrollo del mismo fue realizado por ambos organismos de estandarización, y se conoce como modelo OSI de referencia.

El mismo provee una base común para la coordinación y desarrollo de estándares con el propósito de interconexión de sistemas, mientras que permite a antiguos estándares tener un lugar dentro del modelo.

Este modelo ha sido adoptado como modelo de referencia por casi todas las estandarizaciones internacionales, tal es así que hasta el modelo que se utiliza en internet (modelo TCP/IP) está basado en él.

El objetivo perseguido por el modelo OSI presenta las siguientes particularidades:

- Estructura multinivel
- El nivel superior utiliza los servicios de los niveles inferiores
- Dependencias de Niveles
- Encabezados
- Unidades de datos

En las siguientes subsecciones se explicará cada una de estas características.

a) *Estructura multinivel*

Se diseñó una estructura multinivel con la idea de que cada nivel se dedique a resolver una parte específica de la comunicación. Existen 7 niveles o capas (ver Figura 3-5).

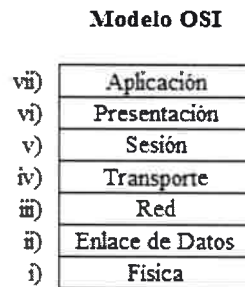


Figura 3-5 - Niveles del modelo OSI

A continuación se explicará brevemente cada capa de la modelo OSI por separada:

i) **Capa Física**

Está relacionada con las conexiones físicas de los dispositivos que componen una red, involucra conectores, tipo de cable, técnicas de modulación, codificación e interpretación de las señales. Se encarga de transformar una trama de datos proveniente de la capa de enlace de datos en forma adecuada al medio de transmisión (fibra óptica, par trenzado de cobre, entre otros).

ii) **Capa de enlace de datos**

Debe ser capaz de proporcionar una transmisión sin errores, es decir, un tráfico de datos fiable a través de un enlace físico. Debe crear y reconocer los límites de las tramas, como así también resolver los problemas derivados del deterioro, pérdida o duplicidad de tramas. También puede incluir algún mecanismo de regulación de tráfico que evite la saturación de un receptor que sea más lento que el emisor. Además se ocupa del direccionamiento físico, del acceso al medio, de la notificación de errores, de la distribución ordenada de tramas, del control del flujo y de la topología de la red. En la Figura 3-6 se muestran las topologías de red más comunes.

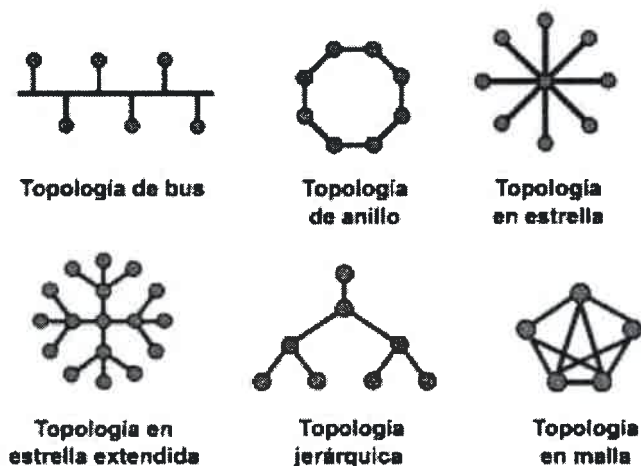


Figura 3-6 - Topologías de red

La tarjeta NIC (*Network Interface Card*, Placa de Red) que se encarga que tengamos conexión, posee una dirección MAC (*Media Access Control*, Control de Acceso al Medio). Dentro de una Red de Área Local (LAN, por sus siglas en inglés, *Local Area Network*), los dispositivos (comúnmente conocidos por su nombre en inglés, *host*) se comunican con direcciones MAC. Un conmutador (comúnmente conocidos por su nombre en inglés, *switch*), que sirve para conectar diferentes hosts en una LAN, posee una tabla en donde se mapea en cada uno de sus puertos la dirección MAC del host que está conectado, de ésta manera puede direccionar tramas transmitidas entre los hosts de una misma LAN. Aquí existe un direccionamiento físico, ya que como dijimos anteriormente la dirección MAC pertenece a la placa de red, o sea es a nivel hardware.

iii) Capa de red

El objetivo de la capa de red es hacer que los datos lleguen desde el origen hasta el destino cuando los hosts no están en una misma LAN. Las redes LAN están unidas por medio de ruteadores (comúnmente conocidos por su nombre en inglés, *routers*) que direccionan los paquetes con direcciones de red, estos dispositivos trabajan en esta capa.

Adicionalmente la capa de red lleva un control de la congestión, que es el fenómeno que se produce cuando hay una saturación producida por los hosts.

En este nivel se realiza el direccionamiento lógico (o direccionamiento de red) y la determinación la ruta de los datos hasta el receptor final.

iv) Capa de transporte

Este nivel actúa como un puente entre los tres niveles inferiores, totalmente orientados a las comunicaciones, y los tres niveles superiores, totalmente orientados al procesamiento. Este nivel define como direccionar diferentes servicios dentro de un mismo host, asignando una dirección única de transporte, conocida como puerto, a cada servicio de un host, permitiendo que un host pueda soportar más de un servicio.

v) Capa de sesión

Esta capa establece, gestiona y finaliza las conexiones entre procesos o aplicaciones finales. Ofrece varios servicios que son cruciales para la comunicación:

- Control de la sesión a establecer entre el emisor y el receptor: se encarga de transmitir, escuchar y realizar un seguimiento de las sesiones.
- Mantener puntos de verificación: esto sirve para que ante una interrupción de transmisión, la misma se pueda reanudar desde el último punto de verificación en lugar de repetirla desde el principio.

Por lo tanto, el servicio provisto por esta capa provee la capacidad de asegurar que, dada una sesión establecida entre dos hosts, la misma se pueda efectuar para las operaciones definidas de principio a fin, reanudándolas en caso de interrupción. En muchos casos, los servicios de la capa de sesión son parcial o totalmente prescindibles.

vi) Capa de presentación

El objetivo de la capa de presentación es encargarse de la representación de la información, de manera que aunque distintos equipos puedan tener diferentes representaciones internas de caracteres (ASCII, Unicode, EBCDIC), números (little-endian tipo Intel, big-endian tipo Motorola), sonido o imágenes, los datos lleguen de manera reconocible.

Esta capa es la primera en trabajar más con el contenido transmitido en la comunicación que con el establecimiento de la misma. En ella se tratan aspectos tales como la semántica y la sintaxis de los datos transmitidos, ya que distintas computadoras pueden tener diferentes formas de manejarlas.

Por lo tanto, podemos resumir definiendo a esta capa como la encargada de manejar las estructuras de datos abstractas y realizar las conversiones de representación de datos necesarias para la correcta interpretación de los mismos.

Esta capa también permite cifrar los datos y comprimirlos. Podríamos decir que es un traductor de la información contenida.

vii) Capa de Aplicación

Ofrece a las aplicaciones la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como por ejemplo POP (*Post Office Protocol*, Protocolo de la Oficina de Correo) y SMTP (*Simple Mail Transfer Protocol*, Protocolo de Transferencia de Correo Simple) para correo electrónico, FTP (*File Transfer Protocol*, Protocolo de Transferencia de Archivos) para transferencia de archivos, entre otros. Hay tantos protocolos como aplicaciones distintas y puesto que continuamente se desarrollan nuevas aplicaciones el número de protocolos crece dinámicamente.

Cabe aclarar que el usuario no interactúa directamente con la capa de aplicación, esta da soporte a los programas para que puedan transmitir sus datos, ocultando la complejidad subyacente.

b) *El nivel superior utiliza los servicios de los niveles inferiores*

Cada nivel se comunica con su par en el dispositivo remoto, pero debe hacerlo enviando el mensaje a través de los niveles inferiores. El nivel N utiliza los servicios del nivel N-1 y proporciona servicios al nivel N+1. Existe el concepto de encapsular datos, donde el nivel N encapsula los datos del N+1 y luego de agregar un encabezado (ver Sección d)) y un código de corrección de errores envía los datos al nivel N-1, donde esto ocurre nuevamente, siguiendo así hasta que los datos salen de manera apropiada por el canal. En el receptor ocurre lo contrario, los datos se van desencapsulando en las diferentes capas hasta que llega a la aplicación de destino.

c) *Dependencias de Niveles*

Cada nivel es independiente del nivel inferior y también del superior. Con lo cuál se podría reemplazar el protocolo utilizado en un nivel sin afectar al resto.

d) *Encabezados*

En cada nivel se incorpora al mensaje un campo de control que permite a cada nivel en el receptor enterarse que su par en el emisor esta enviándole información. Todos los niveles incorporan un encabezado al mensaje, por esta razón, se considera

que un mensaje esta constituido por dos partes: encabezado e información; entonces, la incorporación de encabezados es necesaria a pesar de ser información extra a transmitir. El destino retira los encabezados en orden inverso a como fueron incorporados en el origen, por lo tanto el proceso o aplicación en el dispositivo receptor sólo recibe el mensaje original.

e) *Unidades de datos*

En cada nivel, la unidad de información tiene diferente nombre y estructura, lo cual se muestra en la Figura 3-7.

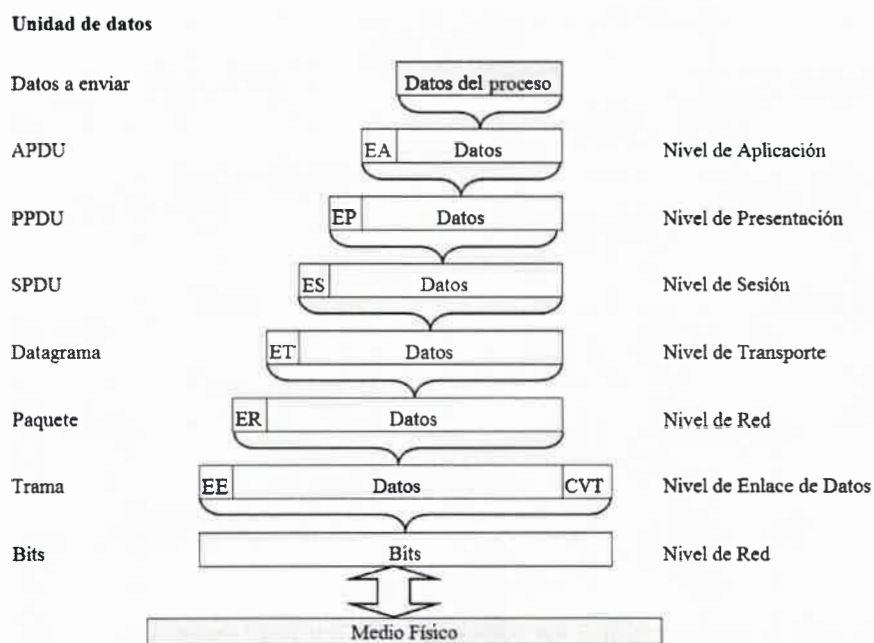


Figura 3-7 - Unidades de datos del modelo OSI

En la Figura 3-7 vemos como los datos del proceso del dispositivo transmisor son encapsulados en la capa de aplicación y a esto se le agrega un Encabezado de Aplicación (EA). La unidad de información para la capa de aplicación se denomina APDU (*Application Protocol Data Unit*, unidad de datos de protocolo de aplicación). Para la capa inferior (en éste caso capa de presentación), esta APDU representa los datos de capa de presentación, a los cuáles se le agrega un Encabezado de Presentación (EP), conformando un PPDU (*Presentation Protocol Data Unit*, unidad de datos de protocolo de presentación) que es la unidad de datos para esta capa. Lo mismo sucede con todas las capas hasta que se llega a la capa física, donde todos los encabezados y

datos superiores son transmitidos en forma de bits al medio físico de forma adecuada (niveles de tensión, ondas de radio, luz, entre otros).

f) *Como interviene el modelo OSI en una comunicación*

Para comprender los aspectos más importantes del modelo nos basaremos en la Figura 3-8, donde las columnas a la izquierda y a la derecha representan a la implementación del modelo OSI en dos dispositivos cualquiera (PC, terminales, IED, etc.) y la columna del medio representa al modelo OSI implementado en un dispositivo de ruteo (router o ruteador).

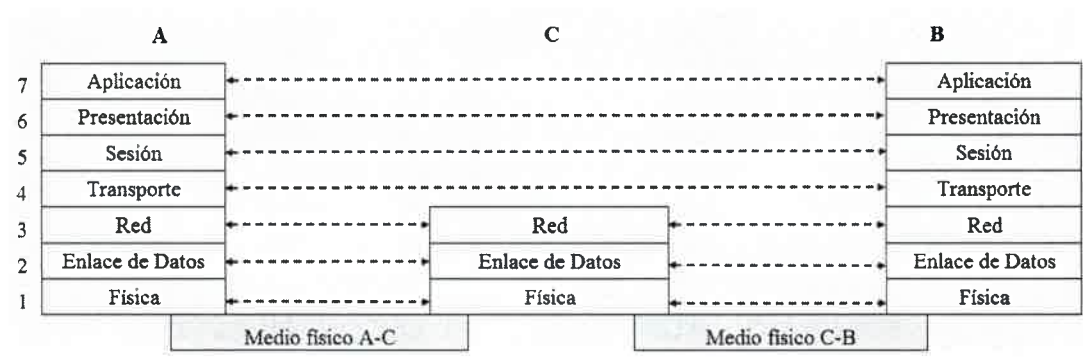


Figura 3-8 - Comunicación - Modelo OSI

Si una aplicación del dispositivo A quisiera enviarle información a otra en el dispositivo B, esta información se toma como datos para la capa de aplicación, a los cuales se le agrega un encabezado y se encapsula todo lo anterior como datos en la capa inferior, así sucesivamente para todas las capas (ver Figura 3-7), entre ambos dispositivos esta el ruteador C (podría haber más de uno), el cuál solo implementa las capas referidas al direccionamiento de los paquetes (capa 1 a 3 del modelo); dependiendo a quien vaya dirigido el mismo elige la mejor ruta ,ya que en general hay más de una, y lo envía. En el host B pasa lo inverso a lo que explicamos al principio. Los paquetes se van desencapsulando desde la capa inferior a la superior hasta que los datos transmitidos desde la aplicación del dispositivo A llegan finalmente la aplicación del dispositivo B.

Las líneas a trazos en la Figura 3-8 indican conexiones lógicas, es decir, para capas de un mismo nivel entre dos dispositivos no hay una conexión física, sino que existen protocolos que se comunican con sus pares en el dispositivo remoto.

La conexión física solo existe por debajo de la capa física, el medio puede ir cambiando, o sea que el medio físico A-C podría ser diferente al C-B (ver Figura 3-8),

en éste caso el ruteador (dispositivo C de la Figura 3-8) tendrá las interfaces correspondientes a cada medio.

La norma especifica que no todas las capas son necesarias en una estandarización, por ejemplo en el modelo TCP/IP (*Transport Control Protocol/ Internet Protocol*, protocolo de control de transporte/ protocolo de internet) que nombramos anteriormente solo cuatro de las siete se utilizan, en la Figura 3-9 se muestra la correspondencia entre el modelo TCP/IP y el modelo de referencia OSI.

Modelo OSI	Modelo TCP/IP
Aplicación	Aplicación
Presentación	
Sesión	
Transporte	Transporte
Red	Internet
Enlace de Datos	Acceso a la Red
Física	

Figura 3-9 - Correspondencia entre el modelo TCP/IP y el modelo OSI

Algunos ejemplos de los protocolos más utilizados en el modelo TCP/IP son HTTP (*HyperText Transfer Protocol*, protocolo de transferencia de hipertexto) y FTP (*File Transfer Protocol*, protocolo de transferencia de archivos) para la capa de aplicación, TCP y UDP (*User Datagram Protocol*, protocolo de datagramas del usuario) para la capa de transporte, IP para la capa de internet y Ethernet para la capa de acceso a la red.

3.2.2. Tecnologías de red

Basados en el modelo OSI, las tecnologías de red están referidas en general a la capa física y de enlace de datos. O sea que una tecnología de red define entre otras cosas, los conectores, cables, topologías, formas de onda, método de direccionamiento físico y método de acceso al medio.

Los medios de transmisión más utilizados son: fibra óptica, par trenzado de cobre, aire (enlaces inalámbricos y satelitales), cable coaxil y portadora en línea de potencia.

Aquí presentaremos las tecnologías de red que han sido más empleadas como también las que se utilizan en la actualidad. Cualquiera de éstas puede ser aplicada a sistemas SCADA:

- RS-232
- RS-485
- Dial-up
- xDSL (*Digital Subscriber Line*, línea de suscripción digital)
- Ethernet
- Token Ring
- FDDI (*Fiber Distributed Data Interface*, interface de datos distribuidos de fibra)
- ATM (*Asynchronous Transfer Mode*, modo de transferencia asíncrona)
- X.25
- Wifi (*Wireless Fidelity*, fidelidad inalámbrica)
- WiMax (*Worldwide Interoperability for Microwave Access*, acceso por microondas para la interoperabilidad mundial)
- Satelital

En las siguientes subsecciones describiremos estas tecnologías.

a) RS-232

En telecomunicaciones, éste es un estándar para la transmisión de datos binarios en forma serial entre un Equipo Terminal de Datos (DTE, por sus siglas en inglés, *Data Terminal Equipment*) y un Equipo de terminación de Circuito de Datos (DCE por sus siglas en inglés, *Data Circuit-terminating Equipment*). Fue desarrollado por la EIA (*Electronics Industries Association*) en 1969, puede operar de manera bidireccional (fullduplex) y utiliza señales de $\pm 3-15$ voltios. La norma recomienda utilizar el conector DB25 pero en general se utiliza el DB9 con cable de cobre serial, para respetar la capacitancia máxima especificada por la norma la distancia límite es de 15 metros, con velocidades especificadas de hasta 20 kbps.

b) RS-485

Está definido como un sistema en bus de transmisión multipunto diferencial, es ideal para transmitir a altas velocidades sobre largas distancias (35 Mbps hasta 10 metros y 100 Kbps a 1.200 metros) y canales ruidosos, ya que reduce las interferencias producidas en la línea de transmisión. El medio físico de transmisión es un par entrelazado que admite hasta 32 estaciones en 2 hilos. La comunicación es halfduplex (se puede transmitir en ambos sentidos pero no a la vez).

c) Dial-up

En ésta tecnología se transmiten señales analógicas en líneas telefónicas tradicionales, para esto se utiliza un modem dial-up. El ancho de banda utilizado en estos sistemas es de 4 KHz. EL CCITT (*Consultative Committee for International Telegraphy and Telephony*) ha definido diferentes normas Dial-up:

- V.21: Velocidad de transmisión de 300 baudios.
- V.22: Velocidad de transmisión 1200 bps a 600 baudios.
- V.22bis: Velocidad de transmisión 2400 bps a 600 baudios.
- V.32: Un estándar fullduplex que puede transmitir a 9600 bps a 2400 baudios. La velocidad es ajustada automáticamente en función de la calidad de las líneas. Las últimas versiones del estándar llegaron a 19200 bps.
- V.34: Compatible con V.32 y es capaz de transmitir a 28 kbps
- V.34bis: Velocidad de transmisión de 33.6 kbps.
- V.42: Mejora de V.34 con corrección de error.
- V.90: Éste es el estándar más rápido disponible para la transmisión analógica sobre éste medio, es capaz de transmitir a 56kbps en modo fullduplex.
- V.92: Presenta algunas mejoras con el anterior y es al que comúnmente nos referimos cuando hablamos de Dial-up.

d) xDSL

Utiliza el par trenzado de hilos de cobre convencionales de las líneas telefónicas para la transmisión de datos a gran velocidad.

A diferencia de Dial-up utiliza un ancho de banda mayor y es una tecnología de transmisión digital de datos. Se han definido diferentes tecnologías:

- IDSL: Utiliza tecnología basada en ISDN (*Integrated Services Digital Network*, red digital de servicios integrados). Alcanza una velocidad de 128 kbps.
- HDSL (*High data rate DSL*, DSL de alta velocidad): fue la primera tecnología DSL en utilizar un espectro de frecuencias mayor al utilizado para llamadas telefónicas. Alcanza velocidades de 1.544 Mbps.
- SDSL (*Symmetric DSL*, DSL simétrico): Flujo de datos simétrico de 2 Mbps.
- ADSL (*Asymmetric DSL*, DSL asimétrico): la velocidad es asimétrica en la subida y la bajada.
- ADSL2 y ADSL2+: versiones mejoradas de ADSL.
- ADSL++: Extiende la velocidad de bajada a 50 Mbps utilizando un espectro de 3.75 MHz.
- RADSL (*Rate-Adaptive DSL*, DSL de tasa adaptiva): mejora la tolerancia al ruido.
- VDSL y VDSL2 (*Very High Speed DSL*, DSL de muy alta velocidad).
- GDSL (*Gigabit DSL*), basado en tecnologías MIMO (*Multiple-input Multiple-output*, entrada múltiple salida múltiple).
- UHDSL (*Universal DSL*, DSL Universal: HDSL sobre fibra óptica).

e) Ethernet

Ethernet se planteó en un principio como un protocolo destinado a cubrir las necesidades de las redes LAN. A partir de 2001 Ethernet alcanzó los 10 Gbps, hecho que dio mucha popularidad a la tecnología. En la

Tabla III-1 se muestran en forma cronológica las diferentes tecnologías Ethernet.

Tabla III-1 - Tecnologías Ethernet

Tecnología	Velocidad de transmisión	Tipo de cable	Distancia máxima	Topología
10Base2	10 Mbps	Coaxial	185 m	Bus (Conector T)
10BaseT	10 Mbps	Par Trenzado	100 m	Estrella (Hub o Switch)
10BaseF	10 Mbps	Fibra óptica	2000 m	Estrella (Hub o Switch)
100BaseT4	100Mbps	Par Trenzado (categoría 3UTP)	100 m	Estrella. Half Duplex (hub) y Full Duplex (switch)
100BaseTX	100Mbps	Par Trenzado (categoría 5UTP)	100 m	Estrella. Half Duplex (hub) y Full Duplex (switch)
100BaseFX	100Mbps	Fibra óptica	2000 m	No permite el uso de hubs
1000BaseT	1000Mbps	4 pares trenzados (categoría 5e ó 6UTP)	100 m	Estrella. Full Duplex (switch)
1000BaseSX	1000Mbps	Fibra óptica (multimodo)	550 m	Estrella. Full Duplex (switch)
1000BaseLX	1000Mbps	Fibra óptica (monomodo)	5000 m	Estrella. Full Duplex (switch)

Las nuevas tecnologías 10 Gigabit Ethernet pueden utilizar par trenzado, fibra óptica monomodo o multimodo, y alcanzan distancias que van desde 26 metros hasta 40 km (dependiendo del medio).

f) Token Ring

Token Ring es una arquitectura de red desarrollada por IBM en los años 1970 con topología lógica en anillo y técnica de acceso de paso de testigo. Token Ring se sitúa en el estándar IEEE 802.5. Actualmente no se emplea en el diseño de redes, ha quedado en desuso por la popularidad de Ethernet.

g) FDDI

FDDI fue desarrollado por el comité de estándares ANSI X3T9.5 en 1983, culminando con SMT (*Synchronous Transport Module*, módulo de transporte síncrono) en 1994. Surgió como una tecnología LAN alternativa a Ethernet y Token Ring que además ofreciera una mayor confiabilidad. En la actualidad, debido a sus superiores velocidades y menor costo, se prefiere utilizar Fast Ethernet y Gigabit Ethernet en lugar de FDDI.

FDDI es un conjunto de estándares ISO y ANSI para la transmisión de datos en redes LAN mediante cable de fibra óptica. Se basa en la arquitectura Token Ring y permite una comunicación tipo full duplex (bidireccional). Dado que puede abastecer a miles de usuarios, una LAN FDDI suele ser empleada en redes WAN.

También existe una implementación de FDDI en cables de hilo de cobre conocida como CDDI.

h) ATM

Es una tecnología desarrollada para hacer frente a la creciente demanda en capacidad de transmisión para servicios y aplicaciones.

Con esta tecnología, a fin de aprovechar al máximo la capacidad de los sistemas de transmisión, sean estos de cable o radioeléctricos, la información no es transmitida y conmutada a través de canales dedicados, sino en forma de cortos paquetes de longitud constante (celdas ATM) y que pueden ser enrutadas individualmente mediante el uso de los denominados canales virtuales y trayectos virtuales.

ATM ofrece un servicio orientado a la conexión, en el cual no hay un desorden en la llegada de las celdas al destino. Esto lo hace gracias a Caminos Virtuales (VP, por sus siglas en inglés, *Virtual Path*) y Circuitos Virtuales (VC, por sus siglas en inglés, *Virtual Circuits*). En el caso de ATM, los VP, son los caminos que siguen las celdas entre dos enrutadores ATM pero este camino puede tener múltiples VC.

En el momento de establecer la comunicación con una Calidad de Servicio (QoS, por sus siglas en inglés, *Quality of Service*) deseada y un destino, se busca el camino virtual que van a seguir todas las celdas. Este camino no cambia durante toda la comunicación, de forma que si se cae un nodo, la comunicación se pierde. Durante la

conexión se reservan los recursos necesarios para garantizar durante toda la sesión, la QoS al usuario.

i) Frame Relay

Frame Relay es una técnica de comunicación mediante retransmisión de tramas introducida por la ITU-T a partir de la recomendación I.122 en 1988.

Proporciona conexiones entre usuarios a través de una red pública, del mismo modo que lo haría una red privada punto a punto, esto quiere decir que es orientado a la conexión.

Las conexiones pueden ser del tipo permanente, (PVC, *Permanent Virtual Circuit*, circuito virtual permanente) o conmutadas (SVC, *Switched Virtual Circuit*, circuito virtual conmutado). Los nodos de comunicación intermedios son conmutadores, y las tramas deben llegar ordenadas al destinatario, ya que todas siguen el mismo camino a través de la red. Puede manejar tráfico ofreciendo QoS.

j) X.25

X.25 es un estándar ITU-T para redes de WAN y al igual que ATM y FrameRelay utiliza circuitos virtuales.

k) Wifi

Esto hace referencia a una de las tecnologías de comunicación inalámbrica más utilizada en la actualidad. Las redes Wifi son también llamadas WLANs (*Wireless LANs*, LANs inalámbricas) y están especificadas en el estándar IEEE 802.11. Veremos los más utilizados:

Los estándares IEEE 802.11b y IEEE 802.11g gozan de una aceptación internacional debido a que utilizan la banda de 2.4 GHz que está disponible casi universalmente para ser utilizada sin licencia, con una velocidad de hasta 11 Mbps y 54 Mbps, respectivamente.

Por otra parte existe el estándar IEEE 802.11a, que opera en la banda de 5 GHz, con canales relativamente limpios en cuanto a interferencias y ruido de transmisión. La banda de 5 GHz ha sido recientemente habilitada y no existen otras tecnologías que la estén utilizando, a diferencia de la banda de 2.4 GHz donde también tenemos Bluetooth, microondas, ZigBee, WUSB, por lo tanto existen muy pocas interferencias. Su alcance es algo menor que el de los estándares que trabajan a 2.4 GHz

(aproximadamente un 10%), debido a que la frecuencia es mayor y por lo tanto hay mayor atenuación.

El estándar IEEE 802.11n y 802.11g trabajan a 2.4 GHz y a una velocidad de 108 Mbps. Con su velocidad y alcance (100-150 metros) lo convierten en una opción ideal para el acceso a internet sin cables.

Al tratarse de conexiones inalámbricas, no es difícil que la comunicación fuera interceptada y se acceda al flujo de información. Por esto, es recomendable la encriptación de la transmisión para aumentar la seguridad. En Wifi esto es posible gracias al WPA (*Wifi Protected Access*, acceso protegido para Wifi), mucho más seguro que su predecesor WEP (*Wired Equivalent Privacy*, privacidad equivalente a cableado) y con nuevas características de seguridad, como la generación dinámica de la clave de acceso. También se puede limitar el acceso a través de las direcciones MAC de los dispositivos de red inalámbricos.

l) WiMax

Es una tecnología de transmisión de datos inalámbrica, sus especificaciones están en el estándar IEEE 802.16. Se presenta como muy adecuada para dar servicios de banda ancha en zonas donde el despliegue de cobre, cable o fibra por la baja densidad de población presenta costos por usuario muy elevados.

Los perfiles del equipamiento que existen en la actualidad o que están en desarrollo trabajan en las frecuencias de 2.5 y 3.5 GHz (con licencia).

En el futuro se prevé desarrollar también perfiles de certificación y de interoperabilidad para equipos que cumplan el estándar IEEE 802.16e, lo que posibilitará movilidad, así como una solución completa para la estructura de red que integre tanto el acceso fijo como el móvil.

Actualmente se contemplan dentro del estándar dos escenarios:

- Uno de acceso fijo, en el que se establece un enlace radio entre la estación base y un equipo de usuario. Las velocidades teóricas máximas que se pueden obtener son de 70 Mbps en un ancho de banda de 20 MHz. Sin embargo, en entornos reales se han conseguido velocidades de 20 Mbps con radios de hasta 6 Km, ancho de banda que es compartido por todos los usuarios de la celda.
- Otro de movilidad completa que permite el desplazamiento del usuario, el WiMax móvil aun no se encuentra desarrollado.

m) Satelital

En las comunicaciones satelitales, las ondas electromagnéticas se transmiten gracias a la presencia en el espacio de satélites artificiales situados en órbita alrededor de la Tierra; que actúan básicamente como un repetidores situados en el espacio, reciben las señales enviadas desde la estación terrestre y las reenvían a otro satélite o de vuelta a los receptores terrestres. En realidad hay dos tipos de satélites de comunicaciones:

- Pasivos: se limitan a reflejar la señal recibida sin llevar a cabo ninguna otra tarea.
- Activos: amplifican las señales que reciben antes de retransmitirlas hacia la Tierra.

3.3. Paradigmas de comunicación en sistemas SCADA

Se pueden definir dos paradigmas bien marcados, dados por la tecnología de comunicación, los mismos son: antiguo paradigma y nuevo paradigma.

a) Antiguo paradigma

Tradicionalmente se utilizaba el paradigma maestro/esclavo. El centro de control era la estación maestra y las RTU eran los esclavos, y se realizaba un muestreo desde el maestro a los esclavos en forma de consultas sucesivas. La topología podía ser punto a punto o multipunto, dependiendo de si había uno o más RTU respectivamente. La infraestructura de comunicación podía ser variada como así también los protocolos de comunicación. Las velocidades de transmisión eran bajas. El maestro tenía el control y utilizaba el sistema de comunicación para enviar y recibir datos de los esclavos, en este esquema podía no ser necesaria la transmisión simultánea en ambos sentidos.

b) Nuevo paradigma

Aquí el centro de control es un cliente y los RTU o IED servidores que mantienen su información en bases datos. A su vez, estos servidores también tienen comunicación entre sí, formando una red, lo que permite incorporar automatización en el sistema [21].

3.4. Evolución de los sistemas de comunicación en sistemas SCADA

Si se analiza la forma en que han evolucionado los sistemas de comunicación en sistemas SCADA, se observa que, al principio, los mismos eran propietarios y cerrados; es decir, no existían estándares abiertos, de forma que el desarrollo entre diferentes fabricantes era incompatible, no existía interoperabilidad [4].

Estos primeros sistemas no utilizaban las mismas interfaces físicas de comunicación ni protocolos. Cada fabricante desarrollaba un sistema de comunicación en función de las necesidades de sus equipos; de esta forma, hasta dispositivos del mismo fabricante de diferentes generaciones eran incompatibles [22].

Con la evolución de las diferentes tecnologías y la necesidad de estandarización por parte de los usuarios, este escenario fue cambiando [5]. Ya los fabricantes no sólo se interesaban en desarrollar sus sistemas sino también en que fueran interoperables. Con respecto a las comunicaciones, esto se logra en diferentes aspectos, desde la parte física (conectores, niveles de tensión, etc.), hasta los protocolos que definen el lenguaje a utilizar por los dispositivos.

Al principio, estandarizaron las interfaces físicas, y si bien los dispositivos se podían conectar, los protocolos de comunicación que utilizaban eran diferentes [6].

En las primeras fases de estandarización de los protocolos de comunicación, las especificaciones carecían de información, dejando demasiada libertad en el diseño y hasta la posibilidad de una mala interpretación de las especificaciones, con lo cual el problema continuaba [23].

Hasta 1985 no había sistemas estandarizados, todos eran propietarios, cerrados y muy básicos y existían pocos fabricantes de equipos. La arquitectura de red era de árbol jerárquico con un solo maestro y las subestaciones estaban aisladas (no había comunicación entre ellas) [24] [25] [26] [27] [28]. Los medios de comunicación eran RS232/RS422/RS449/ RS485 [4] [5] [27] [29], enlaces por la línea telefónica (mas conocido como dial-up) [27] [28], enlaces de radio [4] [26] o a través de las líneas de alta tensión por medio de onda portadora [4] con velocidades de transmisión que no superaban los 1200 bps (bits por segundo). Los protocolos más utilizados en esa época eran: Modbus, SEL, WISP y Conitel 2020 [30].

Entre 1985 y 1990, al surgir mayor cantidad de fabricantes, aparecieron conversores de protocolos [31]. Las redes incluían enlaces redundantes y tenían múltiples maestros. Se utilizaban sistemas de comunicación con tecnologías como X.25

o Frame Relay a modo de leasing [32] [33] o enlaces de radio orientados a paquetes, alcanzando velocidades de transmisión entre 9600 a 19200 bps. Los protocolos que tuvieron mayor utilización fueron DNP3 Serie y las primeras fases de IEC 870 [13].

A partir de 1990 se introdujeron las redes LAN (*Local Area Network*, red de área local) en las subestaciones, interconectándolas con redes WAN (*Wide Area Network*, red de área amplia) [34]. Las tecnologías de red utilizadas eran Frame Relay, ATM (*Asynchronous Transfer Mode*, modo de transferencia asíncrona) [34], enlaces de radio con espectro ensanchado y redes Ethernet [34]. Las velocidades eran mayores a 1 Mbps. En las subestaciones se empezaron a utilizar protocolos de red estándares como TCP/IP (*Transport Control Protocol/ Internet Protocol*, protocolo de control de transporte/ protocolo de internet) [35], HTTP (*HyperText Transfer Protocol*, protocolo de transferencia de hipertexto) y FTP (*File Transfer Protocol*, protocolo de transferencia de archivos) [36]. Muchos protocolos existentes fueron adaptados para trabajar sobre TCP/IP como Modbus, DNP3, IEC 60870, entre otros [37] [38] [39], y se empezó a trabajar en las primeras estandarizaciones [40].

Tras la evolución de las RTU hacia los denominados Dispositivos Electrónicos Inteligentes (IED, por sus siglas en inglés, *Intelligent Electronic Device*), a los antiguos sistemas SCADA se les sumó la automatización de subestaciones [41].

En 1994, el Instituto de Investigación de Potencia Eléctrica (EPRI, por sus siglas en inglés, *Electric Power Research Institute*), empezó a trabajar en un proyecto de automatización de subestaciones denominado Arquitectura de Comunicaciones de Utilidad, (UCA, por sus siglas en inglés, *Utility Communications Architecture*) que intentaba solucionar los problemas de estandarización. Para este proyecto, el EPRI se basó en un conjunto de estándares internacionales, que venían siendo utilizados en los sistemas de comunicación del área de computación y redes de datos [7].

La dificultad que tuvo el proyecto fue que para funcionalidades comunes se podía optar por una serie de alternativas de protocolos de comunicación; así, las implementaciones de diferentes fabricantes no eran compatibles.

En 1996, el Comité Técnico 57 (TC, por sus siglas en inglés, *Technical Commitee 57*) de la Comisión Electrotécnica Internacional, (IEC, por sus siglas en inglés, *International Electrotechnical Commission*) empezó a trabajar en un proyecto similar, bajo la denominación IEC61850 [8].

En 1997, ambos grupos se unieron para definir un estándar internacional que sigue el nombre del segundo proyecto [9], IEC61850, el cual fue lanzado finalmente en 2004. Este estándar representa el estado del arte en automatización de subestaciones.

En la actualidad, la norma IEC61850 está siendo utilizada por muchos fabricantes de dispositivos, y se espera que gracias a la aceptación que está teniendo en el mercado, el desarrollo de la misma sea algo dinámico que impulse a la mejora del funcionamiento del sistema gracias al objetivo principal de la norma que es la interoperabilidad entre dispositivos de automatización de diferentes fabricantes [10].

Las tecnologías de red más utilizadas para los sistemas bajo esta norma están basadas en redes Ethernet de alta velocidad que funcionan sobre fibra óptica [11].

3.5. Protocolos de comunicación más utilizados en sistemas SCADA

Si bien, analizando la evolución de los sistemas de comunicación utilizados en el área, en la sección anterior se hizo mención a los protocolos que fueron utilizados sobre las diferentes tecnologías utilizadas, en las siguientes subsecciones se presentarán estos protocolos de manera detallada, limitándonos a los más importantes, los mismos son:

- Modbus
- Wisp
- IEC 60870 y DNP3

a) Modbus

Modbus es un protocolo desarrollado por Modicon en 1978, años más tarde la empresa liberó el protocolo para ser utilizado por desarrolladores en sus equipos y aplicaciones. De esta manera ha sido ampliamente utilizado para comunicar dispositivos en sistemas de gestión de energía.

El mismo aplica el principio maestro/esclavo, donde un maestro se puede comunicar con uno o más esclavos y un esclavo puede solo responder a peticiones del maestro. Soporta dos modos de transmisión diferentes: Modbus serie, utilizando interfaces tales como RS232 y RS485. Utiliza un Código de Redundancia de Suma de verificación (CRC, por sus siglas en inglés, Cyclic-Redundant Checksum). De ésta forma utiliza registros de almacenamiento de 16 bits.

Modbus soporta dos modos de transmisión diferentes: Modbus RTU, que emplea codificación de datos binarios y Modbus ASCII que se basa en el código ASCII para codificar los datos en cadenas de caracteres [42].

En 1989 se introduce Modbus TCP, que utiliza tecnologías de comunicación basadas en TCP/IP [42], nos referiremos como Modbus serie al estándar original para diferenciarlo de Modbus TCP.

En la Tabla III-2 se muestra la representación de todos los modos de Modbus en el modelo OSI.

Tabla III-2 - Protocolo Modbus en el modelo OSI

Capa de Aplicación	Modbus	
Capa de Presentación	----	
Capa de Sesión	----	
Capa de Transporte	----	TCP (RFC 793)
Capa de Red	----	IP (RFC 791)
Capa de Enlace de datos	RTU (codificación binaria)	Ethernet (IEEE 802.3)
Capa física	ASCII (codificación ASCII) RS232 (V.24), RS485	

b) Wisp

Desarrollado por Westinghouse. Existen dos variantes del protocolo, Wisp+ y Wisp+extendido y la principal diferencia entre ellos es el mayor rango de direcciones que presenta la variante extendida. La trama esta compuesta de un bit de comienzo, los datos y un bit de parada [43].

c) IEC 60870 y DNP3

Mientras emergía el modelo OSI de referencia de siete capas, la IEC propuso una versión simplificada de tres capas como un modelo apropiado para las comunicaciones en sistemas SCADA. Éste es conocido como modelo de arquitectura de funcionamiento mejorado (EPA, por sus siglas en inglés, *Enhanced Performance Architecture*) [43].

En 1990 el modelo EPA fue adoptado como la base para la transmisión de datos de telemetría en el estándar IEC 60870. Las partes principales de éste estándar fueron desarrolladas y publicadas entre 1990 y 1993. Una de las partes más importantes

es la número uno, que define cuatro formatos de tramas que podrían ser adecuadas para aplicaciones de telecontrol [43].

Fue en éste mismo tiempo que el estándar DNP 3.0 fue desarrollado por Harris, el cuál utiliza uno de éstos formatos de trama. El estándar IEC 60870 continuo su desarrollo, tal es así que en 1995 se publica el IEC 60870-5-101 de tareas de telecontrol básicas [43].

Por lo tanto, durante los '90 dos estándares separados fueron desarrollados en paralelo por diferentes organizaciones, trabajando con una base común provista por la norma IEC 60870-5. Estos protocolos, poseen muchas similitudes en sus procedimientos y manejo de comunicaciones hasta la capa de enlace. En su capa más alta las funcionalidades y los objetos de datos utilizados son completamente diferentes [43].

DNP es más utilizado en América, África y Australia, mientras que IEC 60870-5-101 es más utilizado en Europa.

En la Tabla III-3 y Tabla III-4 se muestra la representación en el modelo OSI de los estándares IEC 60870-5-101 y DNP 3.0 respectivamente.

Tabla III-3 - Estándar IEC 60870-5-101 en el modelo OSI

Capa de Aplicación	IEC 60870-5-101, IEC 60870-5-5, IEC 60870-5-4, IEC 60870-5-3	
Capa de Presentación	----	
Capa de Sesión	----	
Capa de Transporte	----	TCP
Capa de Red	IP	
Capa de Enlace de datos	IEC 60870-5-2	
Capa física	IEC 60870-5-1 (FT 1.2) RS232 (V.24), X.24/X.27	IEC 60870-5-104 Ethernet

Tabla III-4 - Estándar DNP 3.0 en el modelo OSI

Capa de Aplicación	DNP 3.0 (Capa de aplicación y librería de objetos de datos)	
Capa de Presentación	----	
Capa de Sesión	----	
Capa de Transporte	DNP 3.0 (Funciones de transporte)	TCP
Capa de Red	----	IP

Capa de Enlace de datos	DNP 3.0 (Capa de enlace de datos)	
Capa física	IEC 60870-5-2 IEC 60870-5-1 RS232 (V.24)	Ethernet

3.6. Conclusiones

Los sistemas de comunicación en sistemas SCADA han sufrido una gran evolución desde sus comienzos, desde cambios en los paradigmas, tecnologías de red y protocolos, hasta la incorporación de dispositivos que permiten trabajar en sistemas de automatización.

En éste capítulo se trataron inicialmente los sistemas de comunicación de manera general, para luego describir concretamente como fue el comportamiento de los mismos en sistemas SCADA. Los conceptos definidos aquí son de gran importancia para el desarrollo de los siguientes capítulos, ya que son necesarios para comprender cualquier sistema de comunicaciones.

El protocolo de comunicación que ha tenido mas aceptación en el área ha sido Modbus, de ésta forma, en el siguiente capítulo se presentará detalladamente al mismo y se describirá el desarrollo de un SCADA para los equipos de la subestación principal de la UNRC con dicho protocolo.

IV. REALIZACIÓN DE UN SISTEMA SCADA PARA LA SUBESTACIÓN PRINCIPAL DE LA UNRC UTILIZANDO MODBUS TCP

4.1. Introducción

En la actualidad, los aspectos de comunicación juegan un papel fundamental en el área de Sistemas Eléctricos de Potencia (SEP), debido a nuevos requerimientos por parte de las redes eléctricas y nuevas necesidades por parte de los usuarios. Es por eso que en esta tesis se analizarán los estándares de comunicación utilizados en los sistemas de protección y monitoreo de las redes eléctricas; se comenzará en este capítulo con la implementación de un Sistema de Supervisión, Control y Adquisición de Datos (SCADA, por sus siglas en inglés, *Supervisory Control And Data Acquisition*) en el sistema eléctrico de la UNRC y en los capítulos sucesivos se continuará con temas referentes a conceptos más novedosos como Sistemas de Automatización de Subestaciones (SAS), SmartGrids y diferentes estándares de comunicación utilizados en los mismos.

En este capítulo se describirá el desarrollo y la implementación de un sistema SCADA para los equipos de protección que se utilizan actualmente en la subestación principal de la UNRC, denominada subestación IPSEP. Los mismos trabajan con el protocolo de comunicaciones Modbus TCP, el cuál si bien no fue recientemente desarrollado, se utiliza en muchas implementaciones de sistemas SCADA actuales.

En primer lugar, en la Sección 4.2 se profundizará el marco teórico expuesto en la Sección 3.5.a) del Capítulo 3 acerca del protocolo Modbus. Esto brindará mayor conocimiento acerca del mismo y sus características de funcionamiento. Se situarán las diferentes variantes del protocolo en el modelo OSI. Se describirán también los tipos de datos y funciones disponibles.

Luego, en la Sección 4.3, se presentará de forma detallada al sistema de distribución eléctrica de la UNRC; incluyendo diagramas unifilares y equipamiento.

Posteriormente, en la Sección 4.3.4 se determinará cuáles son los módulos de comunicación necesarios para que el sistema funcione con el protocolo propuesto.

Una vez que el equipamiento o hardware está definido, en la Sección 4.4 se detallará la arquitectura de software del sistema SCADA; la cual comprende: una aplicación principal desarrollada en Labview, software específico para sistemas

SCADA en tiempo real; como así también, servidores de bases de datos, de páginas WEB y otros lenguajes.

Finalmente, en la Sección 4.5 se presentarán las conclusiones del capítulo.

4.2. Marco Teórico

Modbus serie es un protocolo del tipo maestro/esclavo de capa de aplicación introducido al mercado por Modicon en el año 1978 para la comunicación de sus Controladores Lógicos Programables (PLC, por sus siglas en inglés, *Programmable Logic Controller*). El mismo utiliza RS 232/485 como medio de transmisión, y se convirtió en un estándar de facto en la industria de los Sistemas de Automatización de Subestaciones (SAS). Existen dos modos a utilizar en Modbus serie, RTU o ACSII, cada uno de ellos utiliza diferentes códigos de codificación y de error.

En el año 1989 se adaptó para trabajar sobre TCP/IP utilizando redes Ethernet, éste último fue denominado Modbus TCP. Este protocolo es similar a Modbus serie, solo que agrega a las tramas de datos un encabezado y luego es encapsulado sobre el protocolo de comunicaciones TCP/IP, lo cuál permite la ubicación distribuida de los diferentes dispositivos comunicados a través de una red TCP/IP. De esta forma, la información se puede rutear a través de internet, ya que los datos contienen una dirección IP y un puerto (en Modbus TCP el servidor trabaja en el puerto TCP 502).

Modbus, en todas sus variantes, ha sido ampliamente utilizado en SAS. En la Figura 4-1 se muestra la pila de protocolos utilizada tanto para Modbus serie como para Modbus TCP.

Capa de Aplicación	Modbus	
Capa de Presentación		
Capa de Sesión		
Capa de Transporte		TCP
Capa de Red		IP
Capa de enlace de datos	RTU/ASCII	
Capa física	RS232/RS485	Ethernet

Figura 4-1 - Pila de protocolos utilizada en Modbus

Con Modbus serie, la estación primaria accede a los datos de las diferentes RTU por medio de consultas sucesivas. En la Tabla IV-1 se muestran los tipos de datos que se pueden utilizar, sus modos de acceso y sus direcciones de memoria. Dicho protocolo soporta diferentes métodos o funciones que pueden ser aplicados a sus datos

dependiendo de la naturaleza de los mismos. Estos métodos están definidos a través de códigos, que se describen en la

Tabla IV-2.

Tabla IV-1 - Datos Modbus

Nombre del tipo de datos	Tipo de dato	Modo de acceso	Dirección de memoria
Coil	Bit simple	Lectura y escritura	0x00000
Entrada discreta	Bit simple	Solo lectura	0x10000
Registro de entrada	Palabra de 16 bit	Solo lectura	0x30000
Registros de soporte	Palabra de 16 bit	Lectura y escritura	0x40000

Tabla IV-2 - Códigos de las Funciones Modbus

Método	Código
Leer Entradas Discretas	02
Leer Coils	01
Escribir un Coil simple	05
Escribir múltiples Coils	15
Leer Registro de Entrada	04
Leer Registro de Soporte	03
Leer múltiples Registros	23
Escribir Registro simple	06
Escribir múltiples Registros	16

4.3. Sistema de distribución de la UNRC

En esta sección se detalla la ubicación de las diferentes subestaciones de la UNRC y el equipamiento instalado.

4.3.1. Localización de las subestaciones

El sistema de distribución eléctrica actual de la UNRC esta compuesto por 6 subestaciones, las cuales están distribuidas geográficamente según se muestra en la Figura 4-2.



Figura 4-2 - Ubicación geográfica de las subestaciones de la UNRC

Las distintas subestaciones de la Figura 4-2 son:

1. Subestación del Instituto de Protecciones de Sistemas Eléctricos de Potencia (IPSEP).
2. Subestación de la Plazoleta San Martín.
3. Subestación de la Central Telefónica.
4. Subestación de la Planta Piloto.
5. Subestación del área Deportes.
6. Subestación de la Residencia Estudiantil Universitaria (REU).

En la Figura 4-3 se describe el diagrama unifilar del sistema eléctrico, donde se puede observar que el mismo es del tipo radial.

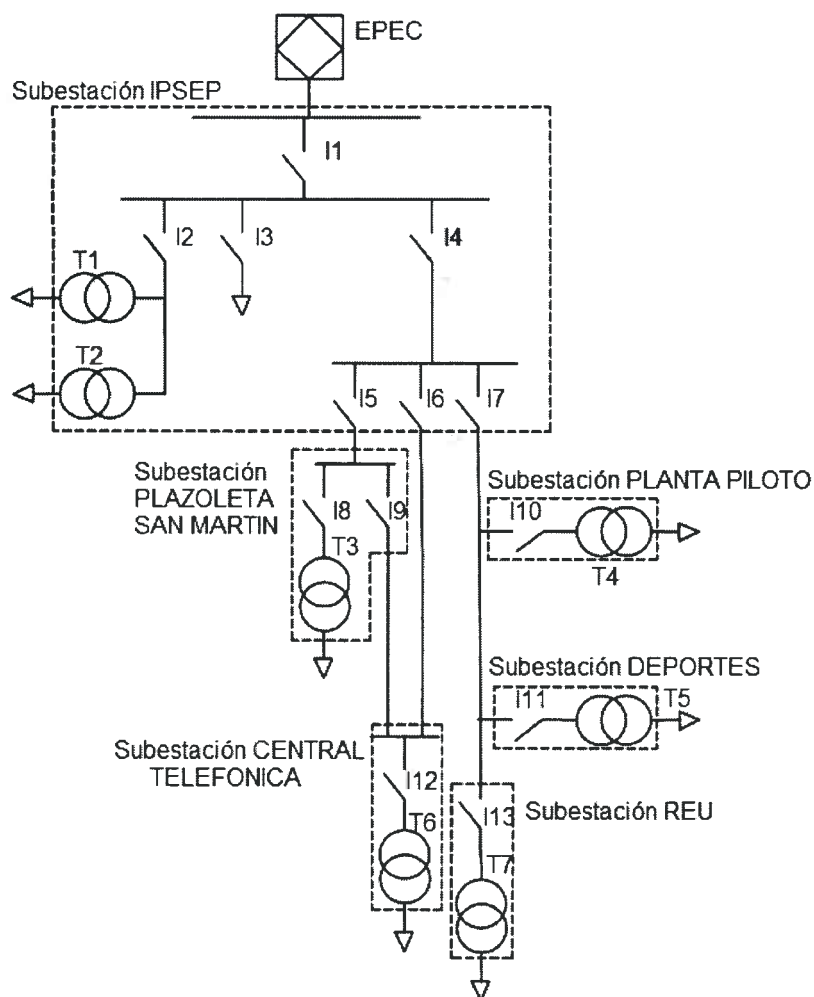


Figura 4-3 -Diagrama unifilar de las subestaciones de la UNRC

4.3.2. Equipamiento del sistema de distribución de la UNRC

Haciendo referencia a la Figura 4-2 y a la Figura 4-3, se describirá el equipamiento instalado en cada una de las subestaciones:

En la subestación IPSEP, esta ubicada la acometida en 13,2 kV que es donde se realiza la interconexión con la Empresa Provincial de Energía de Córdoba (EPEC) a través del interruptor I1. Como vemos tenemos tres ramas en la subestación, cada una de ellas esta compuesta, siguiendo la Figura 4-3 de izquierda a derecha, de la siguiente manera: la primer rama esta provista por el interruptor I2 y dos transformadores de 1 MVA – 13,2/0,4-0,231 kV (T1 y T2) que alimentan el Laboratorio de Ensayos en BT del LEC/IPSEP; la segunda rama tiene un interruptor (I3) para la alimentación del Laboratorio de Ensayos de Media Potencia en 13,2 kV del LEC/IPSEP; y, finalmente, la

tercer rama tiene el Interruptor General de alimentación de la UNRC (I4). A través de seccionadores en SF₆, se generan 3 alimentadores que energizan las distintas subestaciones distribuidas geográficamente según se muestra en la Figura 4-2. Dos de ellos, a través de cables subterráneos de cobre, que pueden trabajar en anillo, alimentan en conjunto o en forma independiente las subestaciones Plazoleta San Martín y Central Telefónica (I5 y I6). Y el tercero (I7), a través de una línea aérea con conductor de aluminio alimenta las subestaciones Planta Piloto, Deportes y Residencia Estudiantil Universitaria (REU).

La subestación Plazoleta San Martín es a nivel y cerrada. En la misma se encuentra un transformador de 1 MVA (T3), con el cuál se provee energía a distintas facultades y edificios colindantes. Su protección primaria es a través de un seccionador trifásico con fusibles HH (I8), el seccionador I9 alimenta a la subestación Central Telefónica.

La subestación Central Telefónica está situada al lado del edificio que aloja a la central telefónica de la UNRC, la misma es a nivel, abierta y tiene un transformador de 630 KVA (T6), la misma es alimentada a través del seccionador I12. Con la misma se brinda energía al sector oeste de la UNRC (Facultad de Veterinaria y Agronomía, Jardín Maternal, Rectorado y otros).

La subestación Planta Piloto es a nivel y cerrada, tiene un transformador de 630 kVA (T4) y un interruptor en SF₆ (I10), la misma alimenta al nuevo edificio de Planta Piloto del Departamento de Ingeniería Química, al edificio de Microbiología y otros, que son los edificios aledaños a la misma.

La subestación Deportes, posee un transformador de 16 kVA del tipo rural (T5), sobre poste y alimenta al campo de deportes, piletas y quincho. Su protección primaria se realiza a través de seccionadores autodesconectores tipo XS (I11) y fusibles de expulsión.

Por último, la subestación REU está ubicada en el sur oeste de la UNRC sobre plataforma aérea en un biposte. La misma alimenta el Complejo de Residencias Estudiantiles Universitarias con un transformador de 315 KVA (T7). Su protección primaria se realiza a través de seccionadores autodesconectores tipo XS (I13) y fusibles de expulsión.

A partir del diagrama unifilar de la Figura 4-3, en las siguientes secciones se realizará un SCADA basado en los equipos de operación, monitoreo y control que actualmente está instalado, dicho equipamiento esta en la subestación IPSEP.

4.3.3. Equipamiento de protección y monitoreo de la subestación IPSEP

El Equipamiento de protección, control y adquisición de la subestación de la universidad es de la Marca Schneider y se encuentra en la subestación IPSEP; en la Figura 4-4 se muestra el diagrama unifilar de dicha subestación con el equipamiento de protección y monitoreo instalado.

El equipo CM4250 es el más completo de la línea Power Logic (SQUARE D - Grupo Schneider); es un equipo trifásico de monitoreo y registro de calidad de energía, el mismo está homologado por el Ente Nacional Regulador de la Energía (ENRE) para Medición de Calidad de Servicio y Calidad de Producto Técnico. Se utiliza para la medición de consumo de energía como control interno y paralelamente, como información para la comparación con las mediciones efectuadas por EPEC.

Los SEPAM 1000+ son una línea de relés de protección por sobrecorriente de la marca (tienen las funciones de protección básica 50/51); el modelo S20 está diseñado para proteger alimentadores o distribuidores, por tanto tienen la posibilidad de habilitarle la función de recierre (79) para eliminar fallas transitorias; el modelo T20, en cambio, es un dispositivo de protección específico para transformadores y tiene la posibilidad de habilitarle la protección de temperatura a través de sensores/termocuplas (38 - con la plaqueta adicional específica de entrada analógica) y/o de imagen térmica (49).

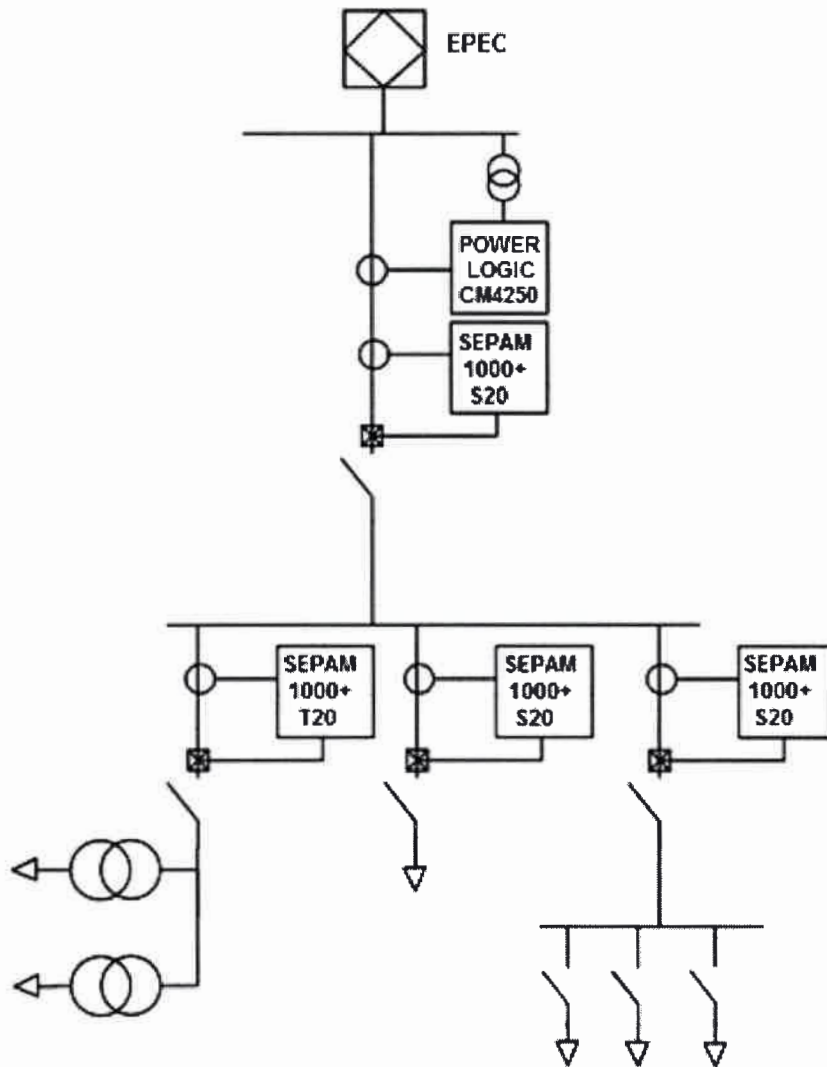


Figura 4-4 - Subestación IPSEP con el equipamiento de protección y monitoreo instalado

4.3.4. Selección de los módulos de comunicación para el equipamiento de protección y monitoreo de la subestación IPSEP

Se hizo un análisis de las interfaces de comunicación necesarias para acceder de forma remota a los datos adquiridos por los equipos y operar ciertas variables. Se tuvo como premisa seguir el estado del arte en el área, para que la subestación no solo quede operativa, sino que además sirva a fines académicos y para futuros trabajos de investigación.

Las tecnologías de comunicación soportadas por los equipos son RS232/422/485, y solo para el caso del Power Logic CM4250, Ethernet; siendo ésta la única tecnología que permite el acceso de forma remota.

Analizando las arquitecturas físicas disponibles para los modelos de SEPAM, no existen interfaces de comunicación Ethernet, pero es posible conectarnos a los mismos a través de adaptadores que si la poseen, denominados gateways, que posibilitan acceder de forma remota a los equipos SEPAM. Los equipos se conectan en guirnalda (en inglés, *daisy chain*) al gateway de acuerdo a la configuración que se muestra en la Figura 4-5.

Los gateways que provee la marca son:

- EGX 100/300/400
- PowerLogic CM4250 con el módulo ECC21
- ECI850

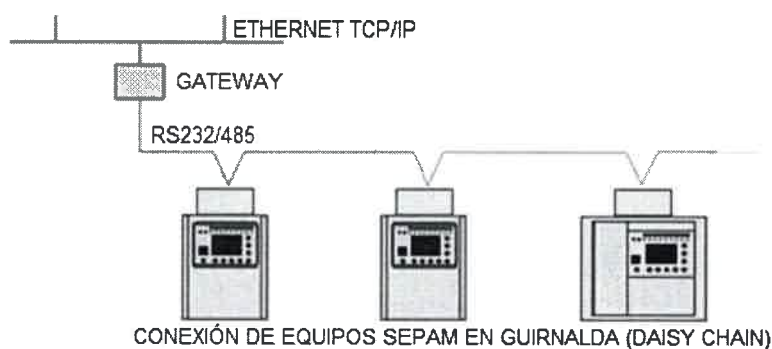


Figura 4-5 - Conexión de los equipos de la subestación IPSEP

Con el gateway de la tercera opción podemos utilizar al protocolo IEC61850 en los SEPAM, mientras que las otras dos opciones permiten utilizar protocolo Modbus Plus. El problema de utilizar el ECI850 es que el PowerLogic CM4250 no soporta IEC61850, y para trabajar con un mismo protocolo se optó por Modbus TCP. Sumado a esto, como el CM4250 ya tiene instalado el módulo ECC21, se optó por esta alternativa, que sería la opción dos, descartando la opción de comprar un Gateway EGX que tendría la misma finalidad.

El módulo ECC21 posee dos interfaces, una RS485 que es utilizada como Modbus serie maestro y otra Ethernet. La primera es utilizada para realizar la conexión en guirnalda con los equipos SEPAM y la otra que sirve para conectarlo a Internet. En esta configuración, los equipos SEPAM utilizan el módulo de comunicación ACE959.

De ésta forma se accede a los registros de los SEPAM a través del CM4250 utilizando Modbus TCP.

En esta sección se seleccionaron los módulos de comunicación a utilizar en base al protocolo propuesto para la implementación, los mismos definen la arquitectura de red del sistema.

4.4. Arquitectura de software

Una vez definida la arquitectura de la red y el protocolo de comunicación se analizaron las diferentes alternativas para el sistema SCADA. Si bien a fines operativos se utilizará una solución brindada por Schneider, se realizó un desarrollo propio a fines académicos y de investigación.

Se hizo un análisis de cuáles serían los aspectos necesarios en la aplicación a implementar. En primer lugar, con el objetivo de comunicarse con los equipos, es necesario un cliente Modbus TCP para acceder a los datos de los mismos a través del Power Logic CM4250 quien funciona como Gateway. Se necesita además, un motor de bases de datos para almacenar a los mismos, y por último, una aplicación web que permita visualizarlos en forma de historial.

En base a estos requerimientos, se utilizó Labview [44] como software principal en el desarrollo del SCADA. El mismo está diseñado específicamente para sistemas de este tipo; donde además se le adicionaron librerías para trabajar con Modbus TCP y con bases de datos. Para lo último se utilizaron Conectores de Bases de Datos Abiertos (ODBC, por sus siglas en inglés, *Open DataBase Connectivity*) [45], ya que para las bases de datos se optó por MySQL [46] y Labview no soporta dicho lenguaje. Para el desarrollo web se utilizó a APACHE [47] como servidor y al lenguaje PHP [48] embebido en HTML para interactuar con el lenguaje de base de datos MySQL. La página es del tipo dinámica, donde en cada campo, pueden utilizarse filtros para buscar por valores mínimos o máximos de las corrientes, tensiones, hora y fecha, etc. En la Figura 4-6 se muestra el esquema completo de la solución que se pretende realizar y los dispositivos con los que interactúa (las interfaces ODBC no figuran en el esquema, ya que no aportan al funcionamiento del sistema). Como ya se dijo en la Sección 4.3.4, el PowerLogic CM4250 no solo cumple su función de medidor, sino que además, gracias a la placa ECC21 que tiene instalada funciona como gateway, de esta forma permite el acceso a la información contenida en los SEPAM a través de Modbus TCP.

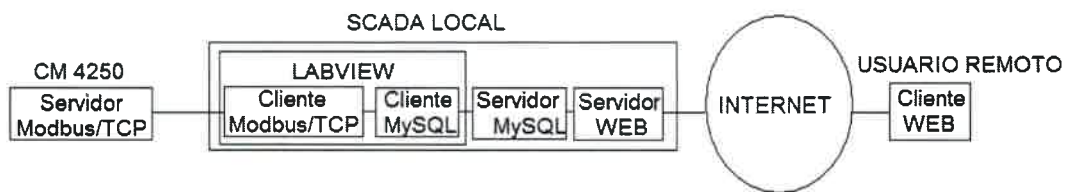


Figura 4-6 - Arquitectura de software del sistema SCADA

4.4.1. Aplicación principal

En ésta sección se describirá la aplicación SCADA en Labview, los otros módulos de software, como bases de datos y páginas web se detallan en el Anexo A, agregando también mayor detalle a la aplicación principal.

En la Figura 4-7 se muestra el bloque principal que permite conectarnos con los equipos utilizando Modbus TCP. Como puede observarse en el menú desplegable, el bloque puede ser utilizado para las diferentes funciones del protocolo que se trataron en la Sección 4.2. En este desarrollo se utilizó la seleccionada en la figura, que sirve para leer los registros del Power Logic CM4250 y los SEPAM

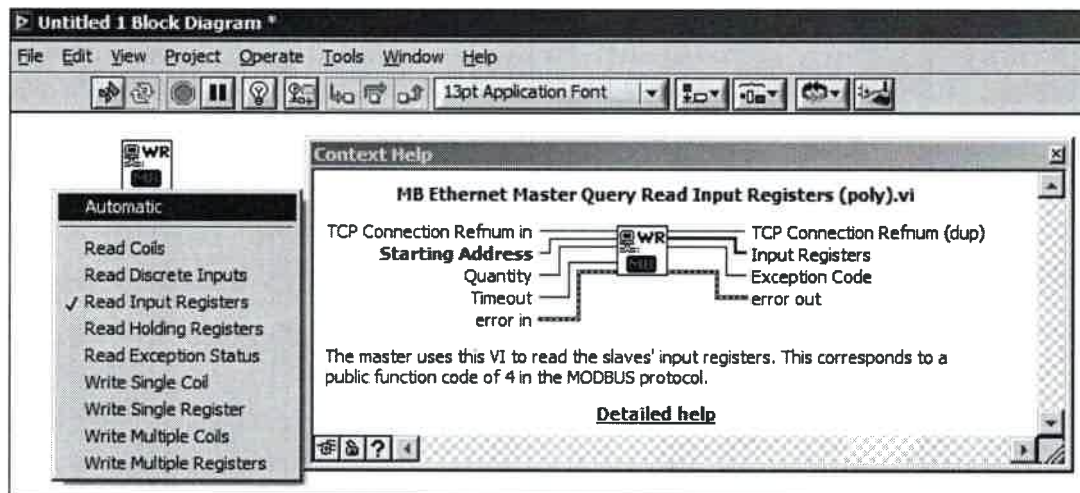


Figura 4-7 - Bloque Modbus TCP de Labview

Conociendo el funcionamiento básico del bloque principal a utilizar, el siguiente paso es interconectar al mismo con los parámetros necesarios a la entrada y a la salida con los indicadores.

En la Figura 4-8 se realiza una aplicación básica que funciona como cliente Modbus TCP en Labview. En primer lugar se detallan algunos bloques y elementos

básicos y luego se explicarán los bloques de mayor importancia y el funcionamiento completo de la aplicación; en la Figura 4-9 se muestra el *panel frontal* de la aplicación.

Recorriendo la Figura 4-8 de izquierda a derecha, en primer lugar tenemos el bloque de inicio de sesión TCP/IP, al cuál se le proporcionará como variables de entrada la dirección IP y el puerto TCP donde se debe conectar. Estos parámetros son provistos desde el *panel frontal* como variables de control. Como variables de salida, el bloque tiene un control de error y otro del estado de la conexión, que viajan de bloque en bloque hasta el final del programa.

El recuadro que contiene a la mayor parte del programa es un bucle *while* que se ejecuta cada 1 segundo, parámetro que fue configurado directamente como una constante desde el *diagrama de bloques* como argumento de entrada al bloque *Time Delay*.

El bloque principal del programa, que fue analizado en detalle haciendo referencia a la Figura 4-7, tiene como argumentos de entrada:

- El estado de la conexión
- Control de error brindado por el bloque de inicio de conexión TCP/IP
- Número de registros a leer
- Dirección de inicio de estos registros

Como salida, este bloque tiene la lectura de los registros, que se muestra en el *panel frontal* y también los parámetros del estado de la conexión y chequeo de error.

Finalmente, los bloques de la derecha del bucle *while* sirven para cerrar la conexión TCP/IP y mostrar algún error, si hubiera ocurrido. Adicionalmente existe una variable de control que sirve para detener el programa y se puede ver tanto en el *Block Diagram* como en el *Front Panel*.

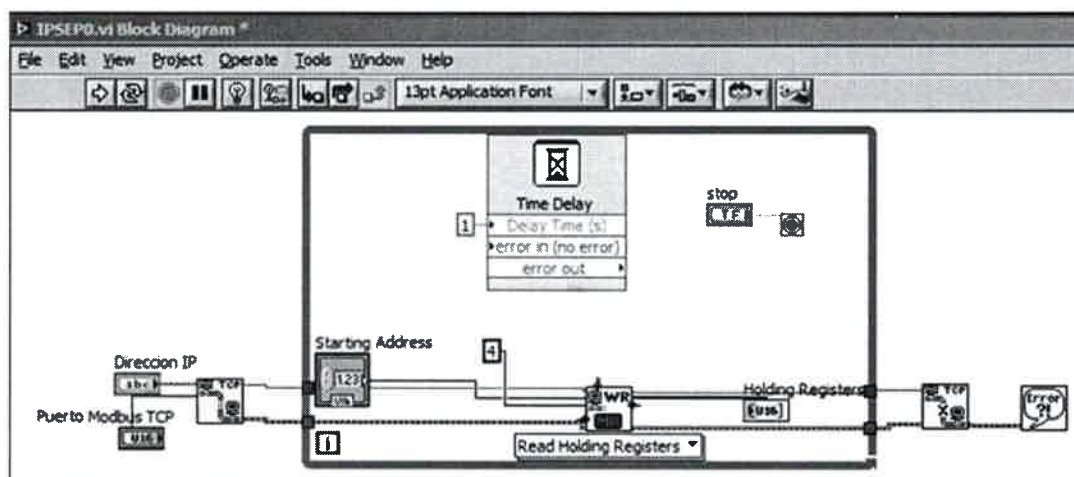


Figura 4-8 - Diagrama de bloques de una aplicación cliente Modbus TCP en Labview



Figura 4-9 - Panel frontal de una aplicación cliente Modbus TCP en Labview

En la *panel frontal* se observan las mediciones que corresponden con los cuatro registros consecutivos al registro número 999, en realidad internamente el programa utiliza en la numeración un valor menos, el registro real es el 1000, de esta forma, a partir del manual del equipo, vemos que estas mediciones corresponden a las corrientes en las líneas y en el neutro. En la Figura 4-10 se muestra la sección del manual que corresponde a los registros que se utilizaron. En el Anexo B se exponen todos los registros disponibles, tanto para el Power Logic CM4250 como para los SEPAM 1000+.

ABBREVIATED REGISTER LISTING

Table C-3 contains an abbreviated register list for the circuit monitor.

Table C-3: Abbreviated Register List

Reg	Name	Size	Type	Access	NV	Scale	Units	Range	Notes
100 ms Metering—Current									
1000	Current, Phase A	1	Integer	RO	N	A	Amperes/Scale	0 – 32,767	RMS
1001	Current, Phase B	1	Integer	RO	N	A	Amperes/Scale	0 – 32,767	RMS
1002	Current, Phase C	1	Integer	RO	N	A	Amperes/Scale	0 – 32,767	RMS
1003	Current, Neutral	1	Integer	RO	N	B	Amperes/Scale	0 – 32,767 (-32,768 if N/A)	RMS 4-wire system only

Figura 4-10 - Registros del Power Logic CM4250

Teniendo así un cliente Modbus TCP configurado que tiene acceso a los datos del servidor, lo que resta es configurar el cliente de base de datos para que se conecte a la base de datos, con lo que la aplicación básica estaría terminada. En la Figura 4-11, se ilustra una aplicación que contempla todos los requisitos planteados inicialmente, teniendo en cuenta los aspectos de comunicación y de bases de datos.

Así, los datos no solo se mostrarán en pantalla, sino que además estarán disponibles en una tabla de una base de datos.

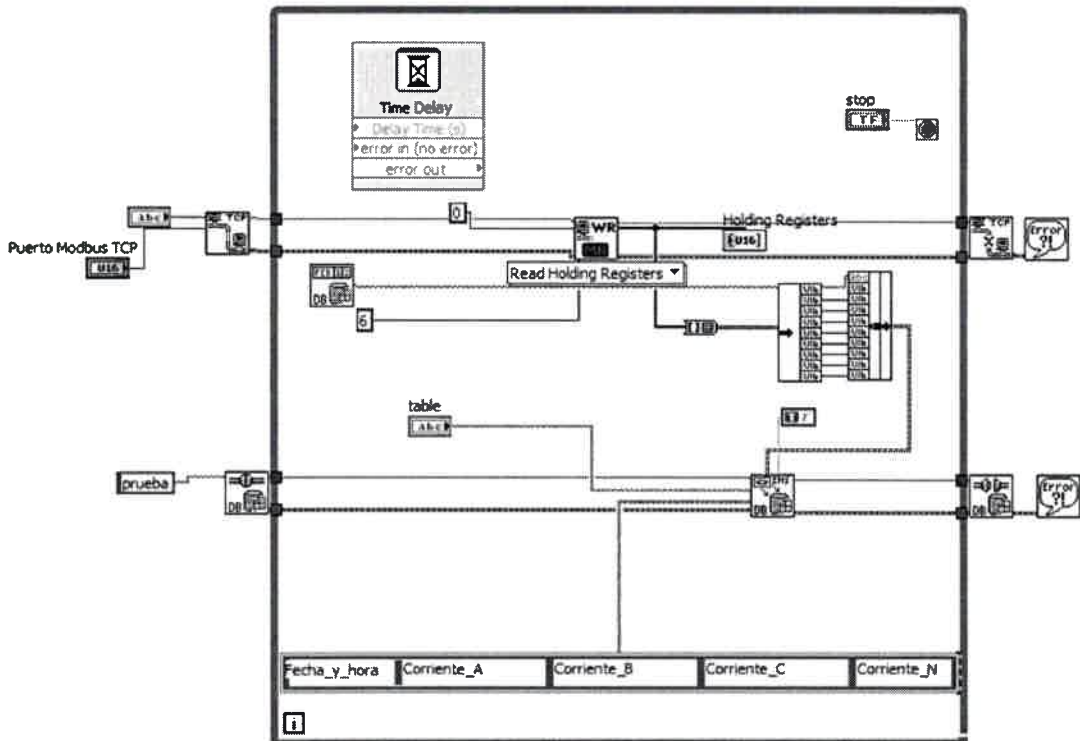


Figura 4-11 - Diagrama de bloques de una aplicación parcial en Labview

La aplicación final esta basada en el esquema de la Figura 4-11, con el agregado en forma secuencial de bloques similares, encargados de buscar todos los datos en los equipos. De esta forma, el mismo tiene asociado un panel frontal, que se muestra en la Figura 4-12, donde se pueden observar indicadores visuales y sonoros para las diferentes alarmas.

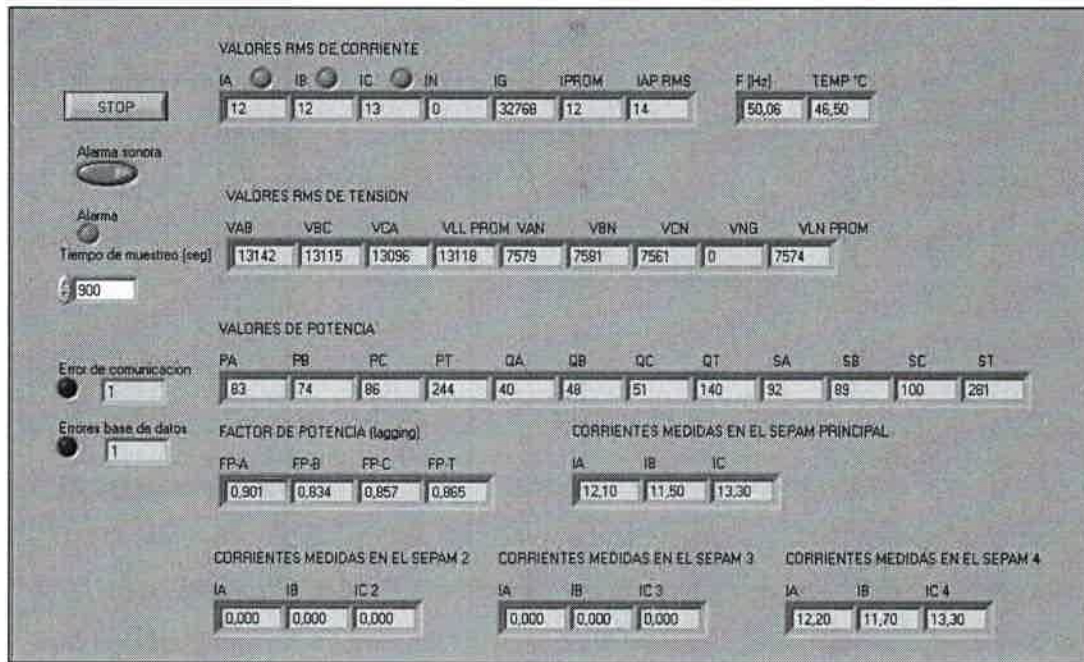


Figura 4-12 - Panel frontal de la aplicación final

4.4.2. Resultados

Los resultados obtenidos en forma histórica por el sistema SCADA realizado en la Sección 4.4.1, son almacenados en una base de datos. En esta sección se muestran los resultados obtenidos a partir de una página web especialmente diseñada para este fin. La información puede ser filtrada por valores límites o por fecha, y luego pueden ser exportados a Microsoft Excel para manipular los datos y realizar gráficas. Desde aquí los datos se pueden exportar a Matlab y ser analizados. En la Figura 4-13 se muestra una captura de pantalla de lo que muestra la página.

192.168.0.195/web_rep/

CM 4250

Resultados

0000-00-00 00 00 00	0	0	0	0
9999-99-99 99 99 99	1000	1000	1000	1000

DATE	IA	IB	IC	IN	IG	IPROM	IARMS	VAB	VBC	VCA	VLLPROM	VAN	VBN	VCN	VNG	VLNPROM	PA	PB	PC	PT	QA	QB	QC	QT	SA	S
2011-04-14	12	12	13	0	32768	12	14	13114	13086	13064	13088	7562	7566	7542	0	7556	83	77	88	248	40	48	49	137	92	9
2011-04-14	12	12	13	0	32768	12	14	13142	13115	13096	13118	7579	7581	7561	0	7574	83	74	86	244	40	48	51	140	92	8

Exportar a Excel

Figura 4-13 - Página web con los resultados

En la Figura 4-14 y la Figura 4-15 se muestran gráficas realizada en Excel a partir de los datos obtenidos desde la página web. Estos datos fueron obtenidos cada 15 minutos durante la semana que va desde las 00:00 horas del día 24 de Marzo de 2011 hasta las 23:45 horas del día 30 de Marzo, teniendo éste período los dos primeros días feriados.

En la Figura 4-14 se muestra la variación de la tensión de línea expresada en por unidad; en la Figura 4-15 se muestra la curva de potencia activa consumida por la Universidad Nacional de Río Cuarto, donde para los primeros cuatro días el consumo es bajo dado que corresponde a días feriados y fin de semana, luego para los últimos tres días el consumo aumenta debido a que comprende días laborales. En esta curva se ve claramente la variación entre el día y la noche.

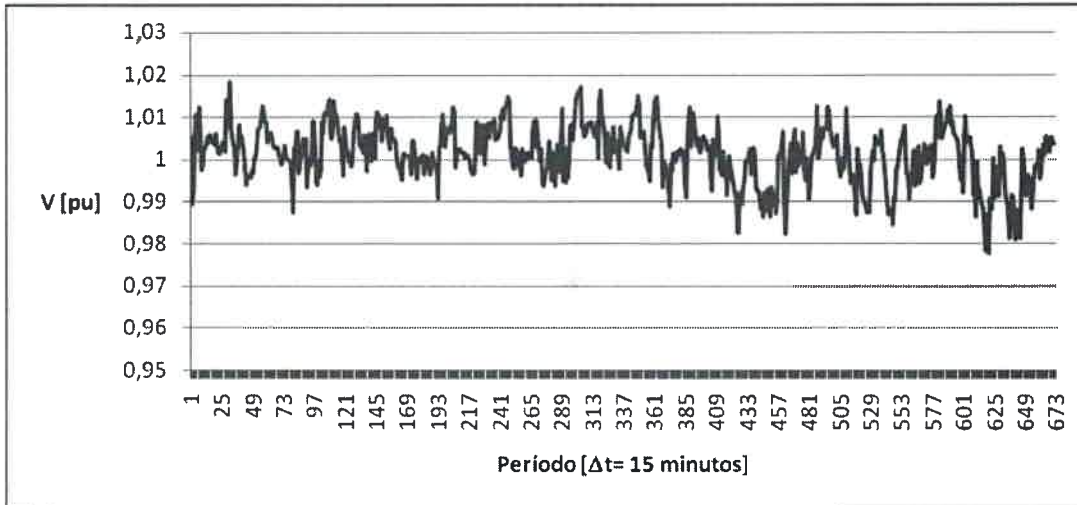


Figura 4-14 - Perfil de tensión semanal

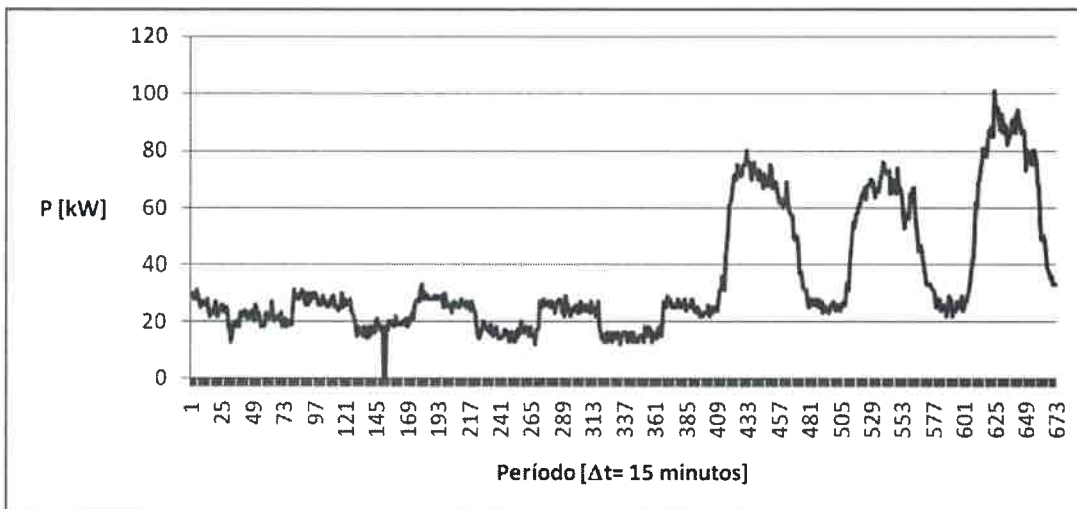


Figura 4-15 - Perfil de carga activa semanal

4.5. Conclusiones

Debido a los avances de las redes eléctricas, la generación distribuida y nuevas demandas de los usuarios, emerge la necesidad de incorporar nuevas tecnologías y sistemas de comunicación, para brindar un servicio acorde a las necesidades de los clientes.

En éste capítulo se trato en detalle al protocolo Modbus TCP. Comenzando con un marco teórico necesario para comprender aspectos de implementación que se trataron luego. De esta forma, se realizó una breve reseña del software a utilizar y se

implementó una aplicación SCADA que sirve para interactuar con el equipamiento de protección y monitoreo instalado en la Universidad Nacional de Río Cuarto.

Este capítulo sirvió no solo para contar con una solución SCADA propia, sino para proponer el mismo esquema en otros lenguajes de programación para trabajos futuros de investigación.

Además, en base a éste capítulo se propondrá para la UNRC, el armado de un curso para la carrera de Ingeniería Electricista que contemple aspectos básicos de comunicación y la forma de diseñar e implementar un sistema SCADA. El curso contará además con laboratorios relacionados con lo que se presentó en este capítulo.

En capítulos posteriores se tratarán temas referentes Automatización de Subestaciones, SmartGrids y los estándares de comunicación utilizados en estos ambientes.

V. SISTEMAS DE AUTOMATIZACIÓN EN SISTEMAS SCADA

5.1. Introducción

En el Capítulo 3 se presentaron a los sistemas de comunicación aplicados a los Sistemas de Supervisión, Control y Adquisición de Datos (SCADA, por sus siglas en inglés, *Supervisory Control And Data Acquisition*). Gracias a la evolución de los elementos de las subestaciones, surge adicionalmente lo que se conoce como automatización de subestaciones, de esta forma, utilizando algún sistema de comunicación adecuado, los dispositivos que componen al sistema SCADA pueden no solo comunicarse sino que también funcionar de forma automatizada. En la Sección 5.2 se realizará una introducción a los conceptos de automatización de subestaciones eléctricas, característica adicional a la simple comunicación entre los dispositivos de la misma. En la Sección 5.3 se presentará la forma en que se debe definir un estándar de automatización de subestaciones. En la Sección 5.4 se introducirá un tema central de la tesis, el estándar IEC61850, que representa al estado del arte en el área de automatización de subestaciones. Finalmente, en la Sección 5.5 se presentarán las conclusiones.

5.2. Introducción a la automatización de subestaciones eléctricas

La arquitectura de protección y control de subestaciones ha estado bajo desarrollo desde la década de los '90. La mejora substancial de la arquitectura de subestación clásica se ha dado gracias al fuerte desarrollo de circuitos integrados, lo que permitió la realización de microprocesadores avanzados, rápidos y muy poderosos. Como resultado, este desarrollo permitió la evolución del equipamiento secundario de la subestación, de dispositivos electromecánicos a dispositivos digitales. De esta manera se llega a la posibilidad de implementar dispositivos de protección y control programables y componentes adicionales del sistema tales como consola local, consola remota, enlaces para control a distancia y un sistema de administración de las protecciones.

Los Sistemas de Automatización de Subestaciones (SAS, por sus siglas en inglés, *Substation Automation System*). Básicamente consisten de uno o más IED que utilizan microprocesadores para controlar, proteger y monitorear al Sistema Eléctrico de Potencia (SEP) y las subestaciones.

La implementación de la automatización de subestaciones está basada en comunicaciones muy confiables que permitan operar el sistema de manera ideal, optimizando tanto los tiempos de respuesta de eventos que ocurren en la red, como así también, el planeamiento y administración de los recursos.

Aplicando el principio general de que las funciones están ubicadas en el nivel en el que se tiene suficiente información para realizar una tarea, desde el punto de vista lógico, la automatización de subestaciones divide al sistema en tres niveles:

- Nivel de Proceso, es el nivel más bajo, incluye sensores, transformadores de corriente y de tensión y dispositivos de actuación (seccionadores e interruptores), lo requerido para operar y monitorear la subestación.
- Nivel de Bahía, es el nivel intermedio, incluye sistemas de control y protección. Estos dispositivos realizan dichas funciones dentro de una bahía (subdivisión de la subestación) en la que están ubicados y en algunos casos pueden también incluir funciones relacionadas con la operación de otras bahías [8].
- Nivel de Subestación, es el nivel más alto dentro de la subestación, incluye consolas locales y unidades centrales conectadas a los centros de control.

Un bus de proceso es el medio que conecta los niveles de proceso y de bahía, mientras que un bus de estación conecta el nivel de bahía con el de subestación.

La conexión entre niveles por medio de buses, se muestra de modo esquemático en la Figura 5-10.

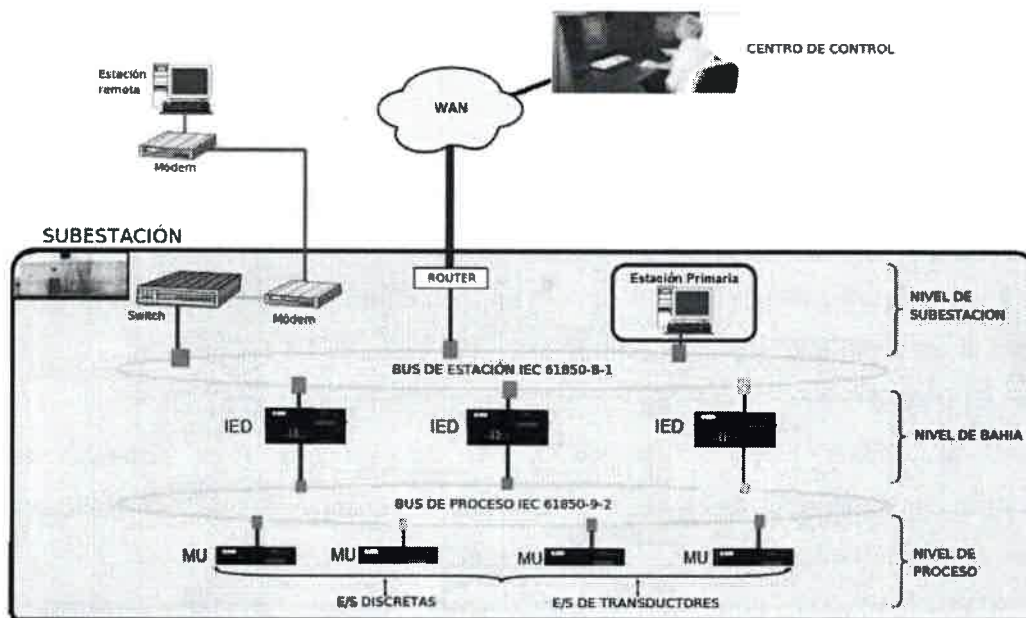


Figura 5-10 - Arquitectura de comunicación de un sistema de automatización de subestaciones

Esta arquitectura permite la introducción de nuevos conceptos asociados a la automatización de subestaciones, como funciones lógicas utilizadas por nuevos dispositivos y redes de comunicación, protocolos de comunicación, etc.

La automatización de subestaciones ha evolucionado, reemplazando los procesos existentes por procesos más sofisticados.

Durante muchos años, los fabricantes de dispositivos de protección y control no coordinaban el desarrollo del sistema, la organización de funciones o el empleo de protocolos de comunicación. Esta falta de coordinación significaba una gran pérdida de tiempo en el diseño y problemas en la interoperabilidad en la integración de dispositivos de diferentes fabricantes.

Finalmente, en los últimos años el objetivo ha sido definir una arquitectura de comunicación que permita la integración de IED de diferentes fabricantes dentro del sistema de automatización y que los mismos puedan interactuar, persiguiendo una arquitectura estándar.

5.3. Definiendo un estándar para la automatización de subestaciones eléctricas

En el capítulo anterior se realizó un sistema SCADA para la subestación principal utilizando el protocolo Modbus TCP. Tanto Modbus serie, como Modbus TCP

aunque muchas veces se utilicen como protocolos de automatización [42] son protocolos de comunicación.

Los primeros estándares de automatización en el área, como pueden ser IEC60870 y DNP 3.0 han sido diseñados específicamente para subestaciones, y en general casos son protocolos propietarios [42].

Tanto UCA como IEC61850 son estándares definidos para SAS. Estos últimos cambian el concepto que existía anteriormente, ya que aquí todos los protocolos que se utilizan en las capas del modelo OSI son estándares internacionales [8]. De esta forma, en vez de utilizar estándares especialmente desarrollados para sistemas de automatización de subestaciones, empieza a surgir el concepto de utilizar estándares internacionales definidos para el modelo de referencia OSI.

IEC61850 es una norma que se utiliza en SAS, tiene como objetivo principal la interoperabilidad de dispositivos de diferentes fabricantes, estandarizando aspectos de comunicación y manejo de la información. La norma no solo se ocupa de la automatización sino también del equipamiento de potencia de la subestación. Utiliza un lenguaje de configuración y un modelo de objetos para describir los componentes del sistema. Además, define la forma en que se deben comunicar los mismos y cuál es la pila de protocolos que interviene en la comunicación, la que está compuesta de protocolos estándares.

5.3.1. Interfaces de comunicación en un sistema de automatización de subestaciones

Las interfaces de comunicación dentro de un SAS pueden ser apreciadas en la Figura 5-11 [49], adviértase que las interfaces entre los bloques funcionales no representan interfaces físicas sino interfaces lógicas genéricas.

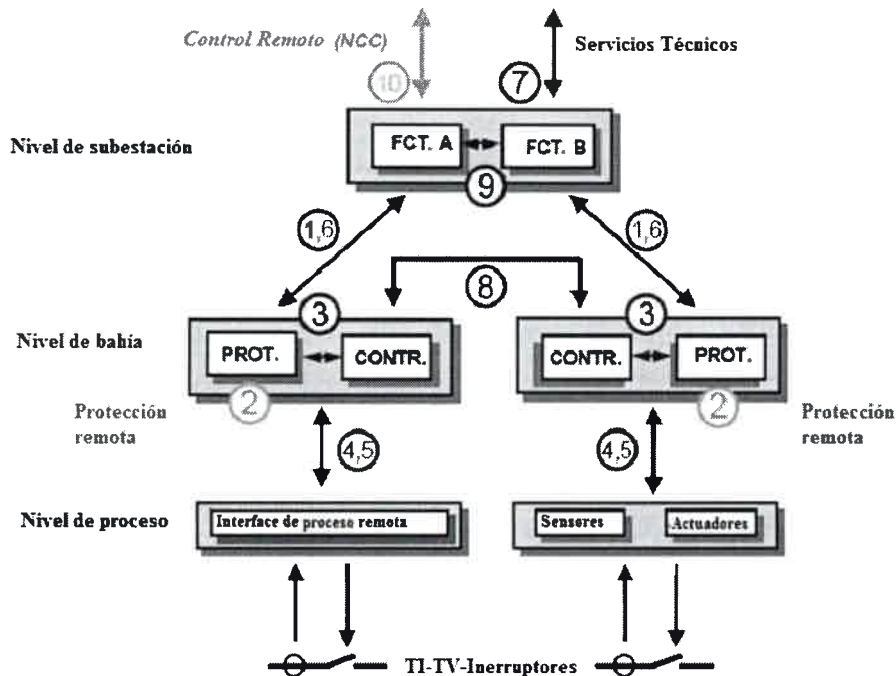


Figura 5-11 - Interfaces de comunicación en un sistema de automatización de subestaciones

5.3.2. Requerimientos físicos para el sistema de comunicación

Las interfaces lógicas deben ser mapeadas a interfaces físicas. Un bus de estación normalmente implementa las interfaces lógicas 1, 3, 6 y 9 (ver Figura 5-11); mientras que un bus de proceso debe cubrir las interfaces lógicas 4 y 5. La interface lógica 8 (comunicación entre bahías) puede ser mapeada a alguno de los buses o a ambos (Figura 5-12).

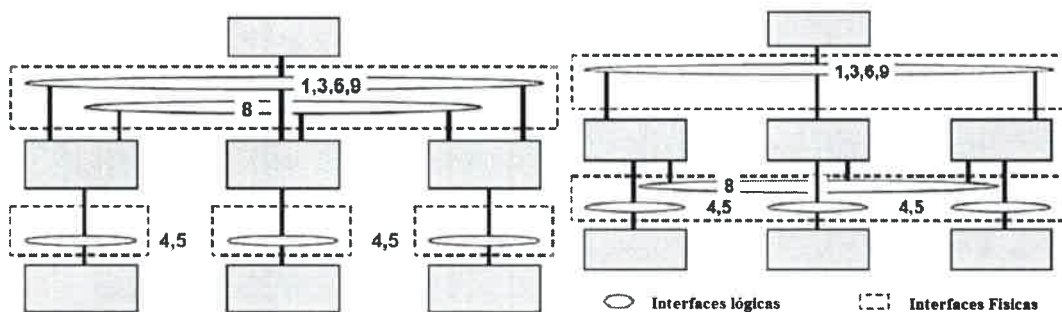


Figura 5-12- Interfaces físicas y lógicas de un Sistema de Automatización de Subestaciones

El mapeo de todas las interfaces lógicas en un solo bus es posible si el mismo satisface los requerimientos de ancho de banda.

5.4. Introducción al estándar IEC61850

IEC61850 sigue los conceptos definidos en este capítulo, como la estructura de tres niveles que se vio en la Sección 5.2, y las interfaces lógicas y físicas que se vieron en las secciones 5.3.1 y 5.3.2, respectivamente. Además, incorpora conceptos adicionales para facilitar el manejo de los datos y optimizar la transmisión de los mismos. En las siguientes subsecciones se describirán, de forma general, aspectos que servirán para entender el capítulo 5 que se centra en la descripción del estándar.

5.4.1. Independencia entre el sistema comunicación y las aplicaciones

Este estándar especifica un set de servicios abstractos y objetos que podrían permitir a las aplicaciones ser sobrescritas de forma independiente a los protocolos de comunicación utilizados.

Así, el fabricante debe contar con una aplicación para invocar o responder a lo que se denomina Interface Abstracta de los Servicios de Comunicación (ACSI, por sus siglas en inglés, *Abstract Communication Service Interface*).

La norma estandariza este set de servicios abstractos a ser usados entre las aplicaciones y los objetos que utilizan las aplicaciones, permitiendo el intercambio de información entre los componentes de un SAS. Sin embargo, estos servicios y objetos abstractos deben ser instanciados a través del uso de protocolos de aplicación y de comunicación concretos.

El ACSI es mapeado en un Mapeo Específico de los Servicios de Comunicación (SCSM, por sus siglas en inglés, *Specific Communication Service Mapping*), que incluye protocolos de aplicación y de comunicación específicos.

En la Figura 5-13 se muestran los conceptos discutidos anteriormente [50].

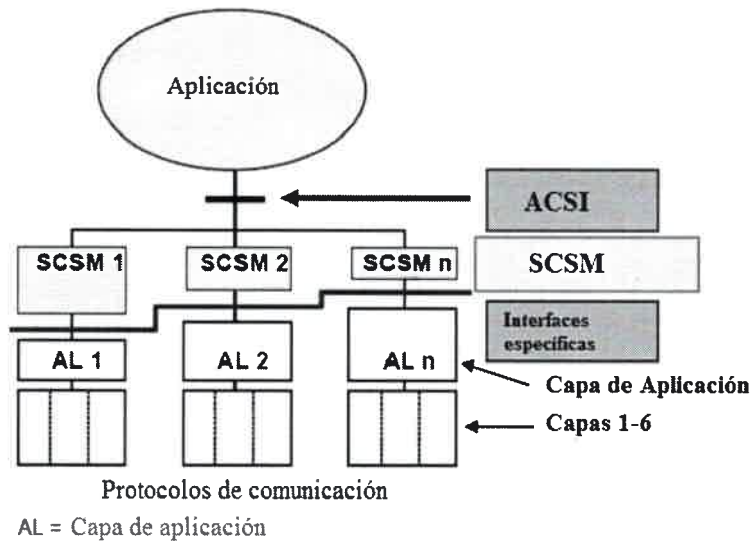


Figura 5-13 - Conceptos, ACSI y SCSM

5.4.2. Parámetros, herramientas de ingeniería y nodos lógicos

Cuando se planifica la construcción de un SAS, y se intenta combinar IED de diferentes vendedores, se espera no solo interoperabilidad entre las funciones y dispositivos, sino también un manejo del sistema uniforme.

Por esta razón la norma IEC61850 cubre también aspectos relacionados con propiedades cualitativas de herramientas de ingeniería, y administración de la configuración.

Dichas herramientas deben ser capaces de intercambiar parámetros de configuración del sistema y de detectar inconsistencias. Gráficamente presentamos esto en la Figura 5-14 [51] [52] [53].

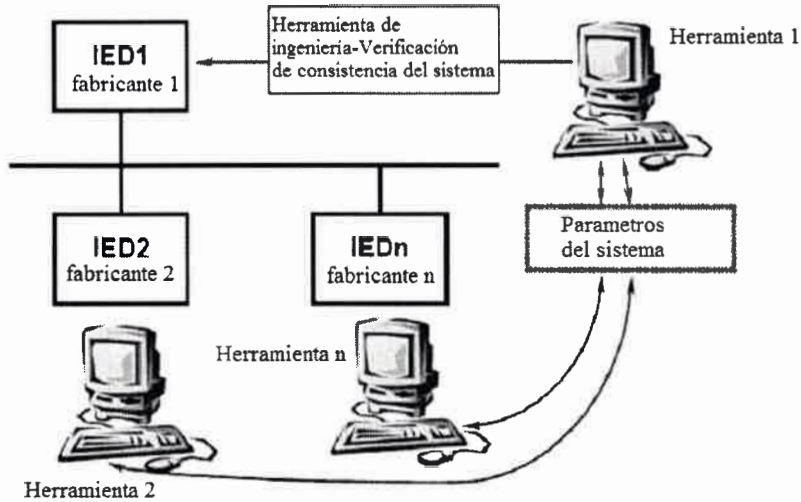


Figura 5-14 - Herramientas de Ingeniería

La norma define Nodos Lógicos (LN, por sus siglas en inglés, *Logical Node*) utilizados en el SAS por medio de los IED, cada uno de estos LN cumple una función específica y existe una relación entre los mismos. Todos éstos LN en conjunto conforman al SAS.

Los LN definidos por la norma están divididos en grupos que corresponden a diferentes funciones y se diferencian por la primera letra de su designación, en la Tabla V-5 [54] se muestran estos grupos con sus funciones y cantidad de nodos lógicos. A modo de ejemplo, en el grupo X, tenemos dos nodos lógicos, XCBR que es un controlador de un interruptor ubicado en la subestación y XSWI que es un controlador de un seccionador ubicado en la misma.

Tabla V-5 - Grupos de Nodos Lógicos y sus funciones

Grupo de LN	Función del Grupo de LN	Número LN
A	Control de Proceso Automático	4
C	Control	5
G	Genéricos	3
I	Interfaces y registros	4
L	Identificación del dispositivo	3
M	Medición	8
P	Protección	28
R	Protección	10
S	Monitoreo	8
T	Controlador de transformadores de medición	2
X	Controlador de Interruptor/Seccionador	2
Y	Controlador de transformadores de potencia	4
Z	Equipamientos adicionales del sistema de potencia	15

En la arquitectura de tres niveles las funciones o LN no están restringidas a un nivel específico en la estructura.

En el nivel de estación tenemos como los grupos de LN más comunes al I, A y C. Por ejemplo, IHMI para una interfaz HMI; CALH, para el manejo de alarmas y ATCC, para el control automático de voltaje.

En el nivel de bahía los nodos lógicos representan funciones de protección, control, y monitoreo a nivel de bahía. Si no hubiera bus de proceso los LN de nivel de bahía y de nivel de proceso estarían juntos en un mismo dispositivo físico; por ejemplo, un XCBR representaría la placa de interface entrada/salida para interruptor y el XSWI

el procesador de control del mismo. Los prefijos más comunes son P, C y A. Por ejemplo, CILO para sincronización, MMXU para propósitos de medición, CSWI para controlar un seccionador y PDIS para protección a distancia.

En el nivel de proceso los LN representan al sistema de potencia (sistema primario). El sistema de potencia es visto desde el sistema secundario, que son IED compuesto por estos LN que pueden ser considerados como controladores de entrada/salida del sistema primario. Pueden tener funcionalidades básicas como supervisión y bloqueo. En general éstos son controlados por LN del nivel de bahía. Los prefijos mas comunes son T, X, Y y Z. Por ejemplo, TVTR como controlador de un transformador de tensión y XCBR como controlador de un interruptor.

5.5. Conclusiones

En este capítulo se presentó una introducción a los sistemas de automatización de subestaciones. Se especificó también como sería la definición de un estándar en el área. El capítulo finalizó con una breve introducción al estándar IEC61850, tema que representa el estado del arte en automatización de subestaciones eléctricas. Esto, sumado a los conceptos que se vieron en el Capítulo 3, permitirá comprender el Capítulo 6.

VI. DESCRIPCIÓN DE LA NORMA IEC61850

6.1. Introducción

IEC61850 es una norma que se utiliza en Sistemas de Automatización de Subestaciones y tiene como objetivo principal la interoperabilidad entre dispositivos de diferentes fabricantes; para esto estandariza aspectos de comunicación y manejo de la información. La norma no solo se ocupa de la automatización de la subestación sino también del equipamiento de potencia de la misma. Utiliza un lenguaje de configuración y un modelo de objetos para describir a los componentes del sistema. Por último, la norma, define de qué forma se deben comunicar dichos componentes y cuál es la pila de protocolos que interviene en la comunicación, la cual está compuesta de estándares.

El estándar se divide en 10 secciones [55], las mismas son:

- IEC61850-1 Introducción y visión global: Se establecen algunos conceptos generales de automatización de subestaciones y se hace una introducción a la norma.
- IEC61850-2 Glosario: Colección de términos y abreviaciones utilizadas en la norma.
- IEC61850-3 Requisitos generales.
- IEC61850-4 Gestión del proyecto y del sistema.
- IEC61850-5 Requisitos de comunicación para funciones y modelos de los dispositivos
- IEC61850-6 Lenguaje de descripción de la configuración para la comunicación en subestaciones eléctricas relacionado con IED
- IEC61850-7-1 Estructura de comunicación básica para la subestación y equipos de alimentación – Principios y modelos
- IEC61850-7-2 Estructura de comunicación básica para la subestación y equipos de alimentación – Interface de servicio de comunicación abstracta (ACSI)
- IEC61850-7-3 Estructura de comunicación básica para la subestación y equipos de alimentación – Clases de datos comunes
- IEC61850-7-4 Estructura de comunicación básica para la subestación y equipos de alimentación – Clases de nodos lógicos compatibles y clases de datos
- IEC61850-8 Mapeo de servicio de comunicación específico (SCSM), referido a los servicios de alto nivel.

- IEC61850-9 Mapeo de servicio de comunicación específico (SCSM), referido a los servicios de bajo nivel.
- IEC61850-10 Prueba de aceptación

Las primeras secciones tratan los conceptos generales de la norma cuyos aspectos más relevantes se han discutido en los capítulos anteriores de este trabajo. En el presente capítulo se tratarán primero los aspectos más específicos que hacen referencia a las partes IEC61850-7-X (cuyas referencias se encuentran en [56] [57] [58] [59]); estos conceptos serán necesarios para comprender la filosofía de funcionamiento de un sistema de automatización de subestaciones que trabaje bajo la norma IEC61850. Luego se presentarán aspectos más globales que están referidos a la parte IEC61850-6; estos conceptos permitirán realizar la planificación general del sistema.

En la Sección 6.2 se presentará la estructura de datos que especifica la norma. En la Sección 6.3 se expondrá como esa estructura es mapeada a protocolos concretos. Finalmente, en la Sección 6.4 se describirá el lenguaje de configuración del sistema que especifica la norma.

6.2. Interface Abstracta de los Servicios de Comunicación

Los modelos que están comprendidos dentro de la Interface Abstracta de los Servicios de Comunicación (ACSI, por sus siglas en inglés, *Abstract Communication Services Interface*) se utilizan para definir datos y servicios [60] [61]. Los modelos de información básicos y los servicios de intercambio de información están fuertemente ligados, aunque desde un punto de vista descriptivo los dos aspectos están en cierto grado, separados, tal como se muestra en la Figura 6-1.

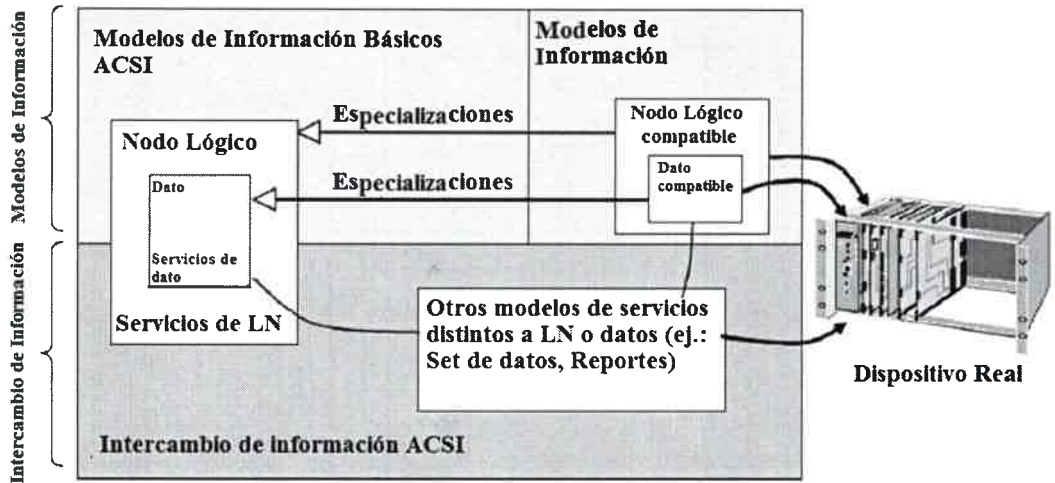


Figura 6-1 - Modelos de información y de servicios ACSI

Los modelos de información tienen asociados sus modelos de servicios. Además existen otros modelos de servicios, como sets de datos y reportes, que están también definidos por ACSI. Los mismos están enlazados a nodos lógicos y datos.

6.2.1. Modelos de información básicos

Los modelos conceptuales para construir modelos de información son [62] [63]:

- Servidor: representa lo que se ve externamente de un IED. Todos los otros modelos ACSI forman parte del servidor.
- Dispositivo Lógico (LD): contiene la información producida por un grupo de funciones de aplicación específicas que son definidas como nodos lógicos.
- Nodo Lógico (LN): contiene la información producida por una función de específica. Por ejemplo una protección de sobrevoltaje o un interruptor.
- Dato: provee medios para definir el tipo de información contenida en los LN.

La información contenida en cada uno de ellos se provee por medio de un modelo de intercambio de información asociado al modelo de información, el cual se describirá con mayor detalle en la Sección 6.2.2.

Los modelos de información son definidos como clases y las mismas comprenden atributos y servicios. El diagrama de clase conceptual de los modelos de información ACSI básicos se muestra en la Figura 6-2.

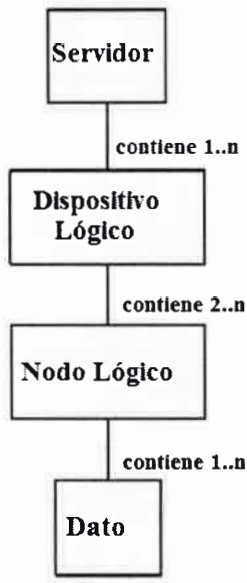


Figura 6-2 - Modelos de información ACSI básicos

En la Figura 6-3 se muestra el rol de este modelo en una subestación real y las partes de la norma que definen cada uno de los componentes del modelo que se discutieron anteriormente.

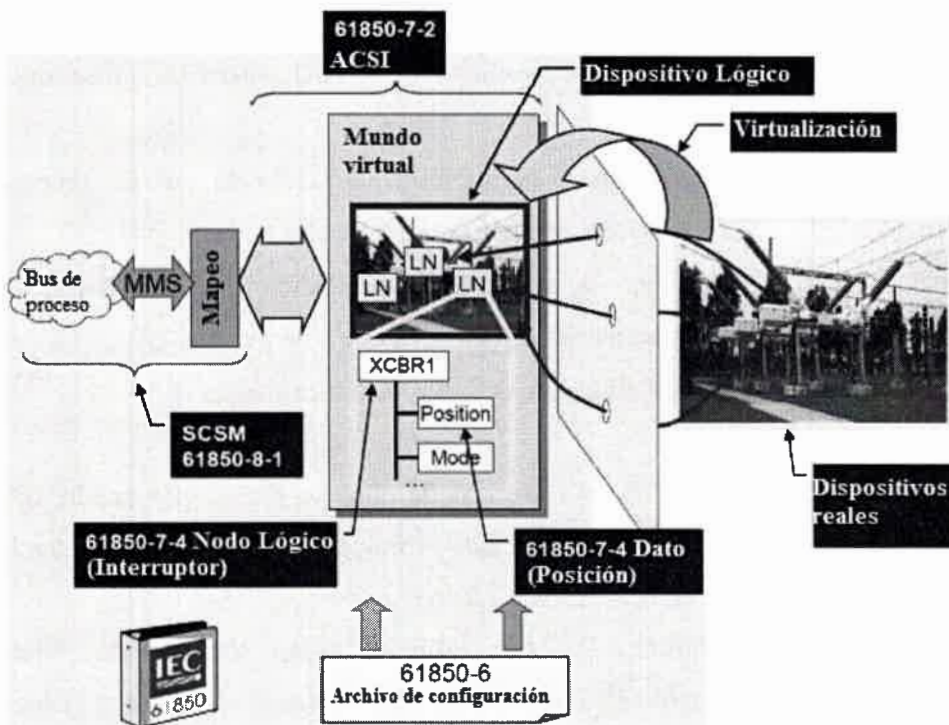


Figura 6-3 - Modelo de un IED

6.2.2. Modelos de intercambio de información

Además de los modelos de servicios de los modelos de información básicos, el ACSI comprende otros modelos denominados modelos de intercambio de información [62] que proveen los servicios que operan en el dato, atributos de dato y sets de datos, éstos están divididos de acuerdo a su función de la siguiente manera:

- Set de datos: permite la agrupación de datos y atributos de dato. Es utilizado para el acceso directo a un grupo de información, para reportes y registros (*logs*).
- Sustitución: soporta el reemplazo de un valor de proceso por otro.
- Control de configuración de grupo: define como cambiar de un set de valores de configuración por otro.
- Control de reportes y registros (*logging*): describe las condiciones para generar reportes y registros basados en parámetros configurados por el cliente. Los reportes deben ser enviados inmediatamente ante cambios en los valores de datos de los procesos, ya que proveen intercambio de información de cambios de estado y secuencia de eventos. Existen reportes con y sin buffer. Los registros pueden ser consultados posteriormente.
- Bloques de control para GSE (*Generic Substation Event*, evento de subestación genérica): soporta un rápido y confiable sistema de distribución de valores de datos de entrada y salida. Es utilizado para mensajes de eventos que requieren retardos muy bajos.
- Bloques de control para la transmisión de SV (*Sampled Values*, valores muestreados): transferencia de muestras rápida y cíclica de las muestras adquiridas por dispositivos ubicados en el nivel de proceso.
- Control: describe servicios a controlar.
- Tiempo y sincronización de tiempo: provee la base de tiempo para los dispositivos del sistema.
- Transferencia de archivos: define el intercambio de grandes bloques de datos tales como programas o archivos de configuración.

Una visión general del modelo de intercambio de información ACSI se muestra en la Figura 6-4.

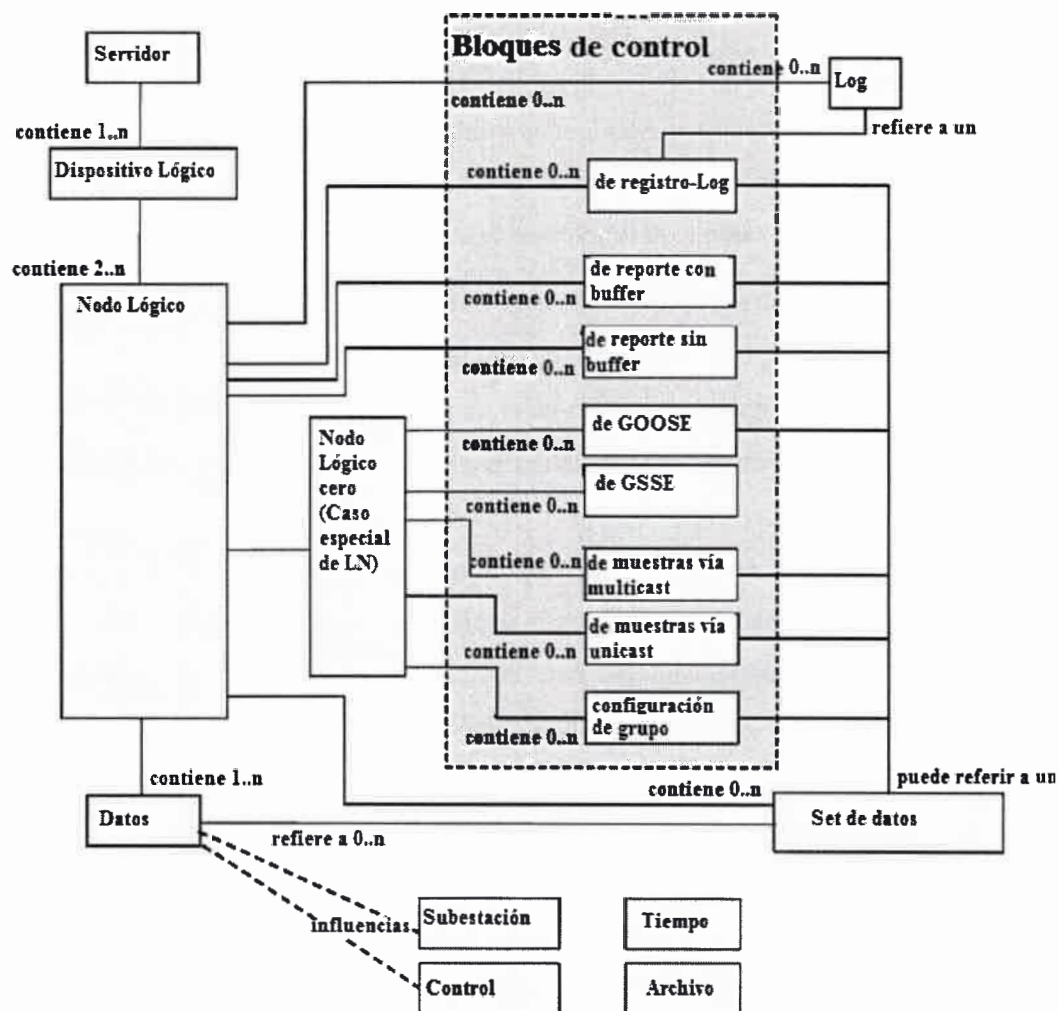


Figura 6-4 - Modelo de intercambio de información ASCII

El bloque de nodo lógico es el que tiene más asociaciones a otros modelos de intercambio de información, por ejemplo control de reportes, control de registros, control de configuración, etc.

Los modelos de intercambio de información son definidos como clases, al igual que los modelos de información.

6.2.3. Servicios provistos por el ACSI

Los servicios provistos por el ACSI permiten acceder a los modelos de información y a los de intercambio de información. Como ya se mencionó anteriormente, todos estos modelos son definidos como clases, en este caso los servicios provistos por el ACSI serían los métodos asociados a esas clases.

A modo de ejemplo, la clase *server* tiene solo un servicio ACSI que es *GetServerDirectory*, el cual devuelve los LDs contenidos en el mismo.

En [57] se definen los servicios de todos los modelos.

6.2.4. Definiciones de tipo de dato

La norma define diferentes tipos de datos de acuerdo a su utilización. En las siguientes subsecciones se detallará cada tipo y aspectos relacionados con los mismos.

a) General

Un aspecto crucial del modelado es la estructura de los objetos de datos, las cuales usualmente comprenden varios atributos de datos.

El concepto básico del dato y sus atributos se muestran en la Figura 6-5 (lo cual también es válido para los bloques de control y sus atributos).

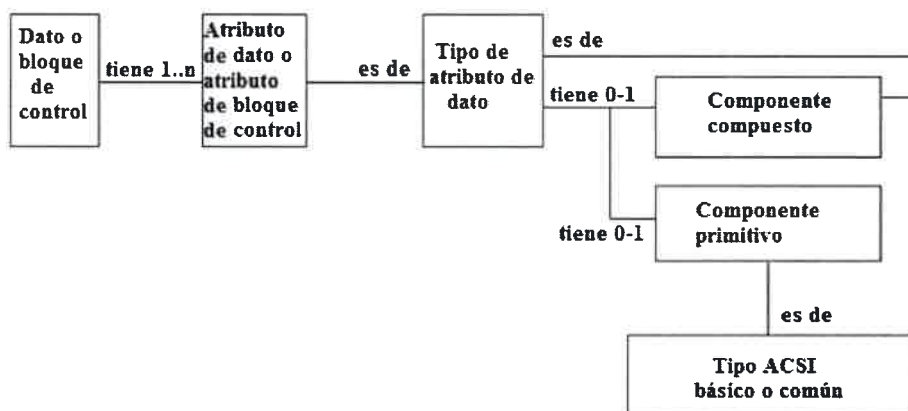


Figura 6-5 - Composición de los datos

El dato (por ejemplo un valor medido) tiene uno o mas atributos de datos (valor de corriente, estampa de tiempo, calidad de información). Cada atributo de dato es de un tipo específico que puede ser primitivo o comprender componentes compuestos (por ejemplo, un vector con un valor de módulo y ángulo) [64].

En la Figura 6-6 el dato está definido como una lista de atributos de datos (db, insMag, etc.) y cada uno de ellos tiene una estructura bien definida. Los de tipo primitivo están representados con línea a rayas (por ejemplo db) y los de tipo compuesto con línea a rayas y puntos (por ejemplo instCVal).

Los tipos compuestos pueden estar formados nuevamente por tipos compuestos, hasta que finalmente los mismos contienen a atributos de tipo primitivos como se ejemplifica a continuación.

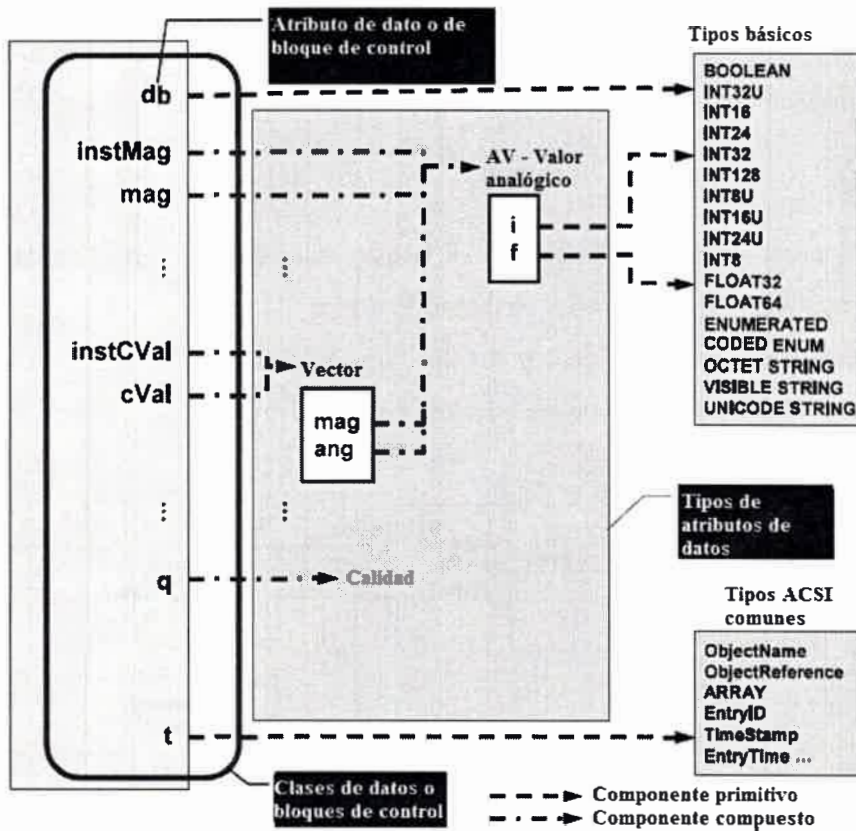


Figura 6-6 - Ejemplo de la composición de datos

En la Figura 6-7 se muestra un ejemplo de una instancia de dato contenido en MMXU1, que es una instancia de la clase de LN MMXU, la cual está compuesta de una instancia de datos de tensión de fase llamada PhV (instancia de la clase WYE). PhV está compuesto de una tensión de fase A phsA (instancia de la clase CMV), la cual está compuesta de un valor complejo Cval (de tipo vector). Este Cval está compuesto de una magnitud de tensión mag (de tipo AnalogValue), compuesta de un valor en punto flotante f (de tipo FLOAT32). El atributo de dato tiene adicionalmente la FC (ver Sección 6.2.3 b), que en este caso es igual a MX (que significa medida).

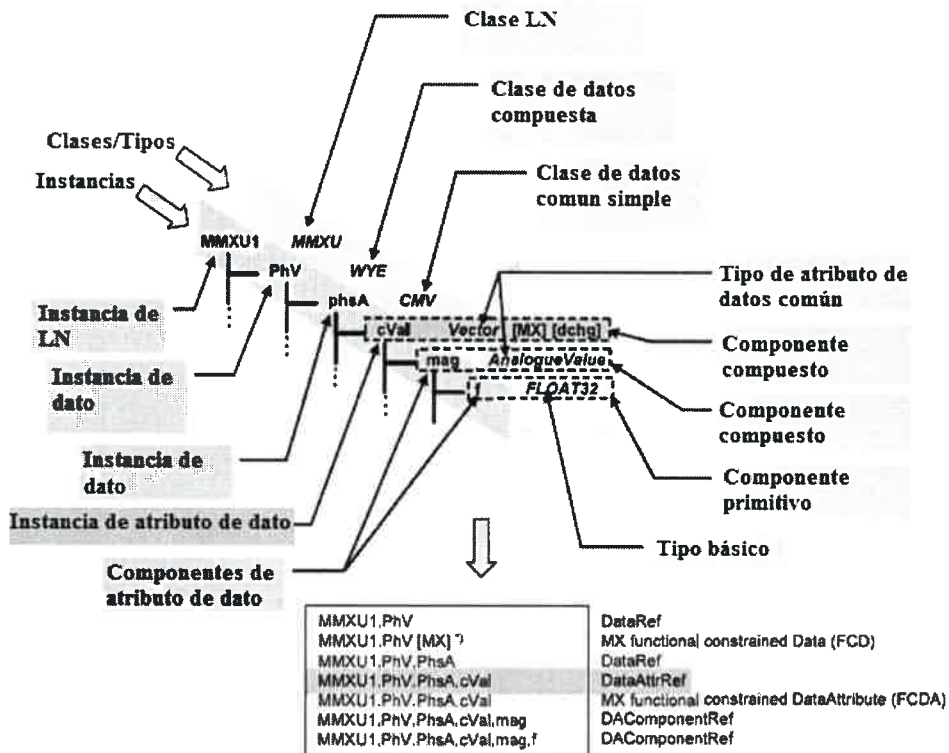


Figura 6-7 - Ejemplo de una instancia de dato

b) Restricción Funcional

Desde el punto de vista de la aplicación, los atributos de datos están clasificados de acuerdo a su uso específico.

La Restricción Funcional (FC, por sus siglas en inglés, *Functional Constraint*) debe ser una propiedad del atributo de dato que caracterice el uso específico del mismo [49]. Por ejemplo, algunos atributos son utilizados para propósitos de control (FC es igual a CO), otros para reportes y registros (FC vale BR para reportes con buffer, RP para reportes sin buffer y LG para registros), otros para configuración (FC vale CF) y otros para indicar medidas (FC vale MX).

El FC es usado en la definición de los datos contenidos en los LN y en los bloques de control. Además, la FC debe indicar los servicios que son permitidos para ser operados en un atributo de dato específico.

La referencia a una colección de atributos de dato de un dato (o de un bloque de control) que tienen la misma FC es llamada FCD (Functional Constraint Data, Dato de Restricción Funcional).

Una referencia a un atributo de dato simple de un dato (o atributo de un bloque de control) que tiene un FC específico es llamada FCDA (Functional Constraint Data Attributo, Atributo de Dato de Restricción Funcional).

En la Figura 6-7 se muestra el FCD y FCDA.

6.2.5. Generalidades en la definición de los modelos de las clases más importantes

En esta sección se detallará cómo están definidas las clases de los modelos más importantes utilizados por la norma.

a) Definición de la clase server

La clase server debe representar al comportamiento del dispositivo visto desde el cliente. Los atributos más importantes del mismo son:

- Punto de acceso del servicio (AP): identifica al servidor dentro del sistema (dirección del mismo). Puede haber más de uno por servidor.
- Dispositivo lógico (LD): identifica a los dispositivos lógicos contenidos en un servidor.
- Archivo: identifica un archivo en el servidor. Puede haber más de uno, o ninguno.
- Asociaciones: identifica asociaciones preconfiguradas, pueden ser multicast o unicast. Puede haber más de una, o ninguna.

Los servicios del servidor son accedidos en forma de consultas recursivas en la estructura jerárquica del mismo. El servicio utilizado para este fin es Get*Directory, el cual va devolviendo la información de los dispositivos lógicos contenidos en el server, nodos lógicos contenidos en un dispositivo lógico, datos contenidos en un nodo lógico y finalmente los atributos de esos datos. El símbolo * tiene diferentes significados dependiendo del nivel de la jerarquía; por ejemplo, en el nivel nodo lógico significa LN, en la Figura 6-8 se muestra el significado para los otros niveles.

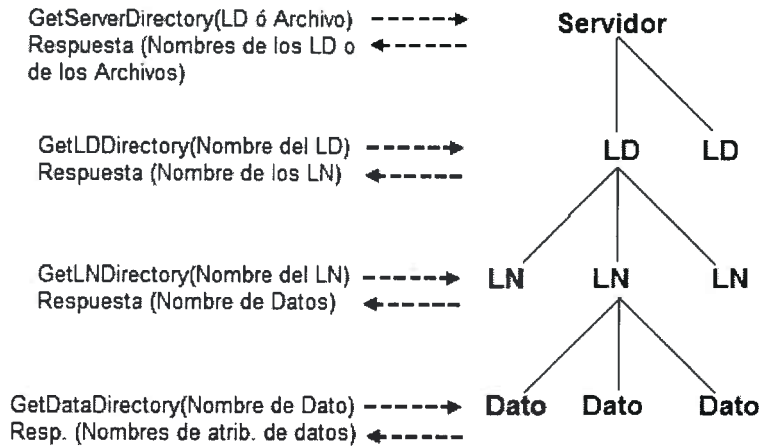


Figura 6-8 - Estructura jerárquica de las clases

b) Definición de clases de los modelos de bloques

- Modelo SCGB (*Setting Group Control Block*, bloque de control de configuración de grupo): este modelo permite tener varios valores para uno o más datos que pueden ser utilizados de a uno a la vez para realizar configuraciones en grupo de dichos datos.
- Modelos RCB (*Report Control Block*, bloque de control de reportes) y LCB (*Log Control Block*, bloque de control de registros): provee mecanismos para la transferencia de valores de datos causados por condiciones y eventos preestablecidos, ya sea desde un LN a un cliente (RCB) o guardando el dato en un registro del servidor (LCB) para futuras consultas. El RCB puede ser con buffer (BRCB) o sin buffer (URCB).
- Modelo GSE (*Generic Substation Event*, evento de subestación genérica): provee la posibilidad de un sistema de transmisión de datos rápido y confiable, puede ser visto como un bloque de reportes.
- Modelo de transmisión de SMV (*Sampled Measured Values*, valores medidos muestreados): se utiliza para el intercambio de valores de datos o set de datos. Utiliza una estructura de buffer y esta sincronizado en el tiempo.
- Modelo de Control: provee la forma de cambiar el estado de procesos internos y externos por un cliente.

En la Figura 6-9 se muestran los conceptos definidos en la Sección 6.2.5 en forma de diagrama de bloques [42].

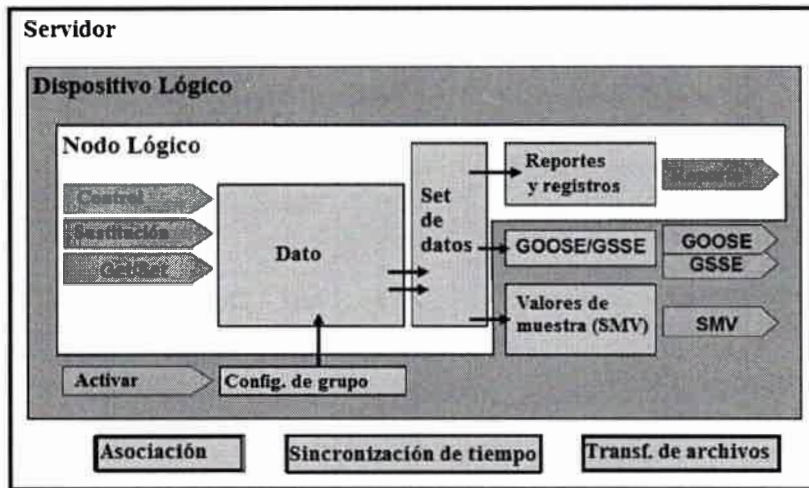


Figura 6-9 - Diagrama de bloques de los modelos de las clases más importantes

6.2.6. Tiempo y modelo de sincronización de tiempo

El tiempo y el modelo de sincronización de tiempo deben proveer el UTC (*Universal Time Coordinated*, tiempo universal coordinado) sincronizado a las aplicaciones ubicadas en un IED. Los componentes del tiempo y del modelo de sincronización de tiempo se muestran en la Figura 6-10 [65].

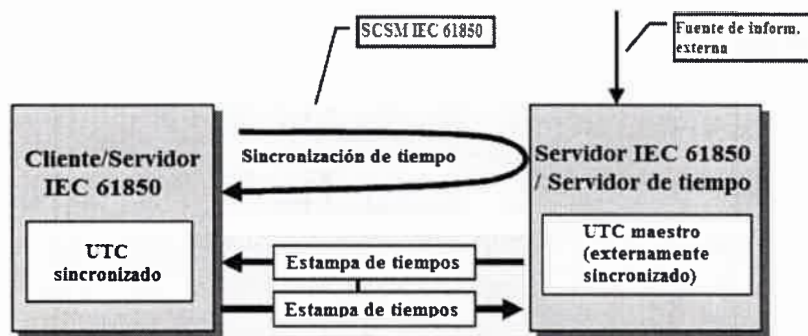


Figura 6-10 - Modelo de sincronización de tiempo IEC61850

El modelo debe tener:

- Información externa requerida por el UTC maestro del servidor de tiempo, obtenida de una fuente de sincronización de otro IED server o cliente.

- El servidor de tiempo provee la fuente para la sincronización de tiempo interna de la subestación y la fuente para la estampa de tiempo (utilizada en los paquetes transmitidos por los IED). El servidor de tiempo puede estar en cualquier IED.
- Un protocolo de sincronización que provea la sincronización de tiempo con otros IED.
- La semántica utilizada por las estampas de tiempo.

6.3. Mapeo a Servicio de Comunicación Específico - Mapeo a MMS

La norma provee especificaciones detalladas de cómo implementar los servicios y objetos ACSI, mapeando a los mismos en diferentes protocolos de comunicación [66] [50]. Este proceso de mapeado se denomina Mapeo a Servicio de Comunicación Específico (SCSM, por sus siglas en inglés, *Specific Communication Service Mapping*).

El perfil de protocolos propuesto por la norma se muestra en la Figura 6-11 [67] [68].

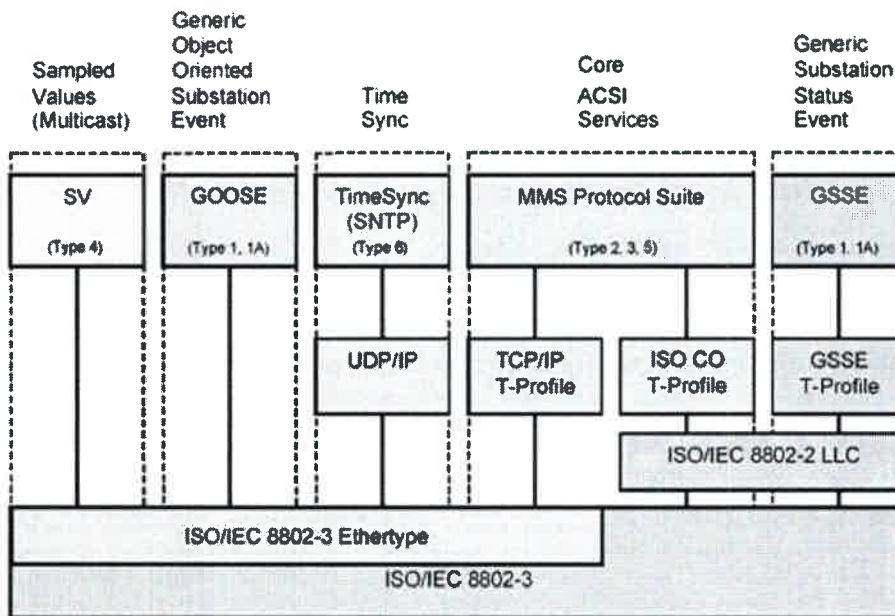


Figura 6-11 - Perfil de protocolos de comunicación IEC61850

El tipo de mensaje que aparece en los bloques de la parte superior de la Figura 6-11 indica el tipo de mensaje a ser mapeado a cada protocolo, el significado de estos tipos se describe en la Tabla VI-1 [69].

Tabla VI-1 - Tipos de mensajes a mapear

Tipo	Corresponde a:
1	Mensaje rápido
2	Mensaje de velocidad media
3	Mensaje de velocidad baja
4	Mensajes de datos raw
5	Funciones de transferencia de archivos
6	Mensajes de sincronización de tiempo

Como ilustra la Figura 6-11, la norma define el uso de diferentes protocolos en función del tipo de mensaje que se transmite. Los valores muestreados (SV) y los mensajes GSE (GOOSE y GSSE) se mapean directamente en capa de enlace de datos, por lo que estos mensajes no tienen la capacidad de ser direccionados a otra red. Para mensajes de sincronismo de tiempo se utiliza el Protocolo de Tiempo de Red Simple (SNTP, por sus siglas en inglés, *Simple Network Time Protocol*). Finalmente, los servicios ACSI se mapean sobre el protocolo de capa de aplicación conocido como Especificación de Mensajes de Fabricación (MMS, por sus siglas en inglés, *Manufacturing Message Specification*).

6.3.1. Mapeo de modelos de datos y servicios ACSI al protocolo MMS

Las clases de los modelos de información y servicios ACSI se mapean al protocolo de capa de aplicación MMS, en la Tabla VI-2 y

Tabla VI-3 se muestra el mapeo correspondiente [9].

Tabla VI-2 - Mapeo de clases de modelos de datos ACSI a clases MMS

Clases ACSI IEC61850 [57]	Clases MMS [70]
SERVER class	Virtual Manufacturing Device (VMD)
LOGICAL DEVICE class	Domain
LOGICAL NODE class	Named Variable
DATA class	Named Variable
DATA-SET class	Named Variable List
SETTING-GROUP-	Named Variable

CONTROLBLOCKclass	
REPOR-CONTROL-BLOCK class	Named Variable
LOG class	Journal
LOG-CONTROL-BLOCK class	Named Variable
GOOSE-CONTROL-BLOCK class	Named Variable
GSSE-CONTROL-BLOCK class	Named Variable
CONTROL class	Named Variable
Files	Files

Tabla VI-3 - Mapeo de los servicios ACSI a servicios MMS

Servicios ACSI IEC61850 [57]	Servicios MMS [70]
GetLogicalDeviceDirectory	GetNameList
GetAllDataValues	Read
GetDataValues	Read
SetDataValues	Write
GetDataDirectory	GetNameList
GetDataDefinition	GetVariableAccessAttributes
GetDataSetValues	Read
DataSetValues	Write
CreateDataSet	CreateNamedVariableList
DeleteDataSet	DeleteNamedVariableList
GetDataSetDirectory	GetNameList
Report (Buffered and Unbuffered)	InformationReport
GetBRCBValues/GetURCBValues	Read
SetBRCBValues/SetURCBValues	Write
GetLCBValues	Read
SetLCBValues	Write
QueryLogByTime	ReadJournal
QueryLogAfter	ReadJournal
GetLogStatusValues	GetJournalStatus
Select	Read/Write

SelectWithValue	Read/Write
Cancel	Write
Operate	Write
Command-Termination	Write
TimeActivated-Operate	Write
GetFile	FileOpen/FileRead/FileClose
SetFile	ObtainFile
DeleteFile	FileDelete
GetFileAttributeValues	FileDirectory

6.3.2. Protocolos de comunicación

El modelo OSI de referencia que se detalló en el Capítulo 3 da las bases para cualquier sistema de comunicaciones, como se verá en la Sección a) el mismo se puede dividir en dos, dependiendo la finalidad de la capa y en b) se detallarán, en base a esas divisiones, las opciones de protocolos que tenemos definidas en la norma.

a) Perfiles A y T

ISO ha reconocido la existencia de perfiles de aplicación (perfiles A) y perfiles de transporte (perfiles T), en la Figura 6-12 se detallan que capas del modelo OSI están involucradas en cada perfil.



Figura 6-12 - Perfiles A y T especificados por la norma

Diferentes combinaciones de perfiles A y perfiles T pueden ser utilizadas para permitir el intercambio de cierto tipo de información/servicio.

b) Modelos utilizados en IEC61850

La norma define cuatro diferentes combinaciones de perfiles, las que son utilizadas para:

- Servicios Cliente/Servidor
- Servicios GSE
- Sincronización de tiempo
- Servicios SV

i) Servicios Cliente/Servidor y sus perfiles de comunicación

Estos servicios son los que utilizan los servicios ACSI, por lo tanto el protocolo de aplicación que interviene es MMS.

La norma define un perfil A y 2 perfiles T para utilizar, uno basado en TCP/IP y el otro en el modelo OSI. En [70] se pueden encontrar los protocolos que intervienen en cada modelo.

ii) Servicios GOOSE/GSE y su perfil de comunicación

Para estos servicios la norma define solo un perfil-A y un perfil-T, los protocolos que intervienen en los mismos se muestran en [70].

iii) Servicios GSSE y su perfil de comunicación

Para estos servicios la norma define solo un perfil-A y un perfil-T, los protocolos que intervienen en los mismos se muestran en [70].

iv) Sincronización de tiempo y su perfil de comunicación

La norma define solo un perfil-A y un perfil-T, los protocolos que intervienen en los mismos se muestran en [70].

v) Servicios SV

Los SV pueden ser transmitidos por puerto Ethernet o serie. Este último ha quedado obsoleto por las velocidades de transmisión que se requieren en la actualidad, por lo que nos basaremos en los perfiles que define la norma para el primer caso. La norma define solo un perfil-A y un perfil-T, los protocolos que intervienen en los mismos se describen en [70].

6.4. Lenguaje de Configuración de Subestaciones

En las siguientes subsecciones se expondrá al Lenguaje de Configuración de Subestaciones (SCL, por sus siglas en inglés, *Substation Configuration Language*). El mismo cumple un rol muy importante en las definiciones de la norma, ya que representa el método utilizado para la planificación y configuración de los componentes del SAS.

6.4.1. Importancia del lenguaje SCL en el proceso de ingeniería

El lenguaje SCL facilita la ingeniería de un SAS. El mismo sirve para describir diferentes aspectos de la subestación:

- Diagrama unifilar de la subestación y la ubicación de LN necesarios para la automatización.
- IED preconfigurados (en general por el fabricante) con los LN que soporta, sin estar los mismos asignados a ningún sistema específico.
- IED configurados para alguna aplicación específica.
- Sistema completo, asignando LN de IED a los LN requeridos por el sistema que se mencionaron anteriormente.

Todo esto, teniendo como objetivo principal el diseño del sistema, la comunicación en el mismo y la forma en que se deben intercambiar y combinar dichos aspectos entre diferentes herramientas de ingeniería de forma estandarizada [52] [71] [72] [73].

En la Figura 6-13 se muestra gráficamente y a rasgos generales cómo funciona el proceso de intercambio de información entre diferentes herramientas y cómo finalmente se configura el sistema [74] [75].

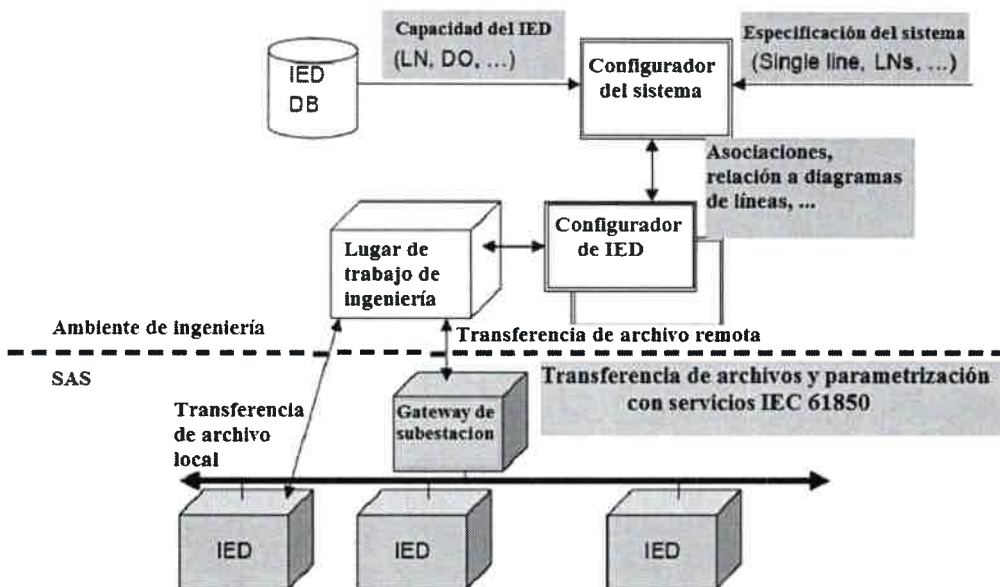


Figura 6-13 - Transferencia de archivos y parametrización

6.4.2. El modelo de objetos SCL

El lenguaje SCL describe los siguientes modelos:

- Modelo de la estructura del sistema primario (de potencia): dispositivos que lo componen y cómo están conectados.
- Modelo del sistema de comunicación: cómo están conectados los IED a subredes y redes, y a cuál de sus puntos de acceso. También comunicación a nivel de aplicación, o sea cómo se agrupan los datos (set de datos) para ser enviados.
- Modelo del IED: nodos lógicos, dispositivos lógicos, reportes, registros y asociaciones de cada IED.

El modelo de objetos de SCL tiene tres partes básicas [76] que se detallan en las siguientes subsecciones.

a) Subestación

En esta parte se describe el equipamiento primario o de potencia de la subestación, sus conexiones y requerimientos en cuanto a qué elementos necesitaría el sistema para la automatización, como así también, el mapeo de éstos con los elementos descritos en la parte de IED (ver Sección 6.4.1b).

Es un modelo jerárquico basado en la estructura funcional de la subestación. Los LN que existen en este contexto son objetos que pueden estar en cualquier nivel de la subestación (refiriéndonos a la estructura de tres niveles).

El propósito más importante de este modelo es relacionar cada LN con una función requerida en la subestación.

Los siguientes objetos de subestación son usados en el modelo SCL. En orden jerárquico son:

- Subestación: objeto que identifica una subestación completa.
- Nivel de tensión: parte identificable de la subestación que tiene el mismo nivel de tensión.
- Bahía: parte identificable o subfunción de la subestación dentro de un nivel de tensión.
- Equipamiento: equipos primarios (de potencia), se conectan mediante nodos de conexión (se define en dos viñetas más adelante).
- Subequipamiento: subparte del anterior, podría ser una fase de un equipo trifásico.
- Nodo de conexión: objeto de conectividad eléctrica que conecta equipos primarios.
- Terminal: punto de conexión terminal de un equipo primario.

b) IED

En esta parte se describen los IED (contiene información de sus nodos lógicos, dispositivos lógicos, reportes, registros) que participan en el sistema, de forma de proporcionar las funciones necesarias para los requerimientos impuestos en la parte de subestación (ver Sección 6.4.2a).

Además, en esta sección se definen las funciones que se utilizan en la subestación para realizar la tarea de automatización. Los siguientes objetos son usados en el modelo SCL; en orden jerárquico son:

- IED: dispositivos que realizan funciones de automatización de subestaciones por medio de nodos lógicos. Normalmente se comunican con otros IED en el SAS.
- Servidor: una entidad de comunicación dentro de un IED, permite acceder por medio del sistema de comunicación (con su único AP) a los datos contenidos en él.
- Dispositivo Lógico: de acuerdo con la norma es el objeto que está contenido directamente en un servidor de un IED. Los mismos contienen a los nodos lógicos.

- **Nodo lógico:** es el objeto que de acuerdo con la norma está contenido en un dispositivo lógico de un IED. Los datos contenidos en el nodo lógico pueden ser consultados por otros nodos lógicos del mismo u otro IED, y de esta forma realizar las funciones de automatización requeridas por el mismo.
- **Objeto de datos:** dato contenido en el nodo lógico.

La norma contempla funciones de IED adicionales a la automatización, éstas son router (relacionado con las comunicaciones) y reloj (para sincronismo).

c) Comunicación

Esta sección está relacionada con subredes y AP de comunicación, y describe las conexiones de comunicación entre IED. También se describe la ubicación de relojes para sincronización.

Este modelo es, en contraste con los otros, un modelo no jerárquico. Modela las conexiones lógicas posibles entre IED en una subred y a través de las mismas. Una subred describe solo un nodo de conexión entre AP. Un LD o un cliente de un IED se conecta a una subred por medio de un AP, utilizando al mismo para realizar asociaciones con otros LDs de otros IED.

Se utilizan los siguientes objetos:

- **Subred:** nodo de conexión para la comunicación directa entre AP (capa 2). Todos los AP conectados a una misma subred se pueden comunicar entre si con el mismo protocolo. Las subredes son un concepto lógico, así muchas subredes pueden convivir en la misma red física.
- **Punto de acceso (AP):** utilizado por uno o más LDs de un IED para comunicarse con otros LDs de otros IED. Puede haber como máximo una conexión entre un LD y una subred, sin embargo, un AP puede servir a más de un LD, así los LN contenidos en un LD pueden utilizar diferentes AP para conectarse a diferentes subredes. Un AP puede ser utilizado por un cliente o por un servidor.
- **Ruteador:** los clientes conectados a una subred solo tienen acceso a los servidores conectados a esa subred. La función del ruteador extiende el acceso a servidores de IED de otras subredes. Sin embargo, un ruteador restringe el acceso a solo servicios que usan la capa de red, algunos servicios como GSE (*Generic Substation Event*, evento de subestación genérico) y SV (*Sampled Values*, valores muestreados) no pueden ser ruteados ya que solo utilizan la capa de enlace, en éste caso son filtrados.

- Reloj: un reloj maestro en la subred, que es utilizado para sincronizar los relojes internos de todos los otros IED conectados a la subred.

Los ruteadores y relojes son conectados a la subred a través de sus AP. En la Figura 6-14 se muestra un ejemplo típico con un IED conectado al bus de estación y al bus de proceso.

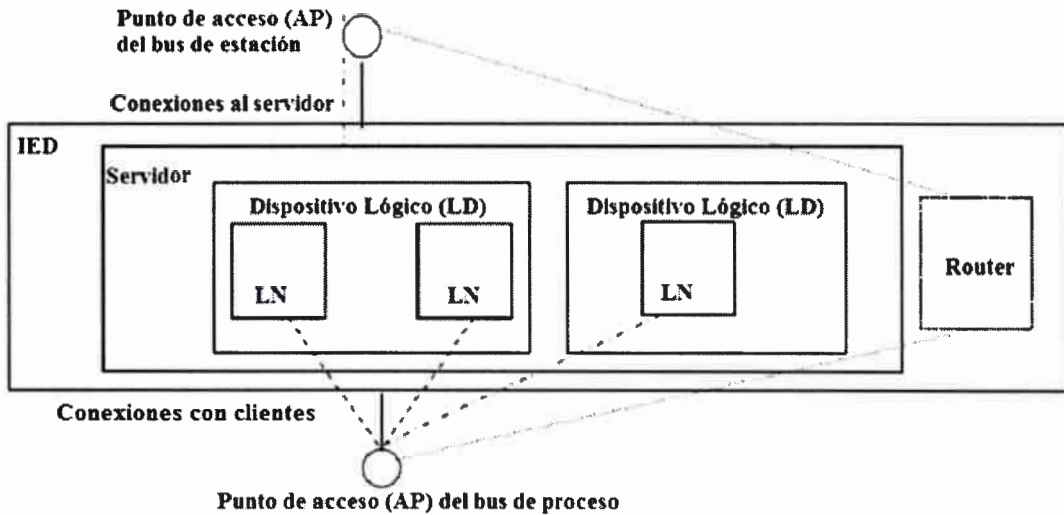


Figura 6-14 - Puntos de acceso de los buses de proceso y de estación de un IED

d) Referenciamiento de los objetos a nivel de aplicación

Permite identificar dentro del modelo jerárquico del IED a todos los objetos que vimos anteriormente en la Sección 6.2.1. La referencia completa del objeto consiste de la concatenación de todos los nombres de los objetos que se encuentran mas arriba en la jerarquía. Las referencias a los diferentes objetos son únicas y los mensajes transmitidos entre los dispositivos deben utilizar este referenciamiento, el cual se muestra en la Figura 6-15.

Las partes 2 y 3 de dicha figura forman el nombre del LN y distinguen diferentes instancias de LN dentro del mismo LD de un IED. Un prefijo de LN es utilizado para enlazar un LN instanciado en un IED a algún proceso. El número de instancia de la parte 3 debe ser usado para distinguir LN instanciados, los cuales no están todavía asignados a ningún proceso, o aquellos que tengan el mismo prefijo. La definición detallada del mapeo de estas señales a los nombres reales, se puede estudiar de las referencias [70] [77] [78].

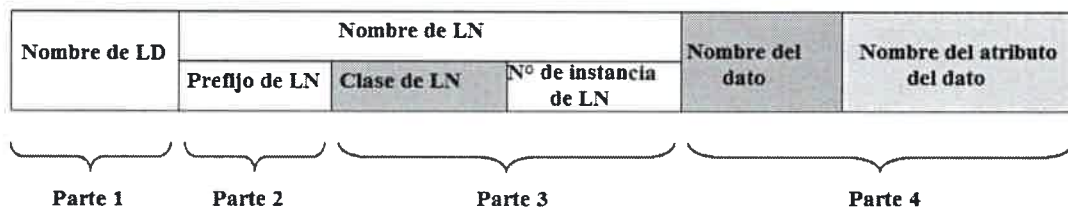


Figura 6-15 - Referenciamiento de los objetos de datos a nivel de aplicación

La sección de definición de las plantillas de datos SCL y los nombres estandarizados definidos en las referencias [58] y [59] determinan los posibles valores para las partes 3 y 4 de la Figura 6-15.

A modo de ejemplo, utilizaremos la estructura ilustrada en la Figura 6-16, donde se muestra la estructura completa de un IED. El ejemplo contiene un LD con el nombre Relay y dos instancias de LN, las mismas son XCBR1 y XCBR2. A su vez, éstas contienen instancias de datos, ctrVal y stVal, que representan valores de control y valores estáticos respectivamente.

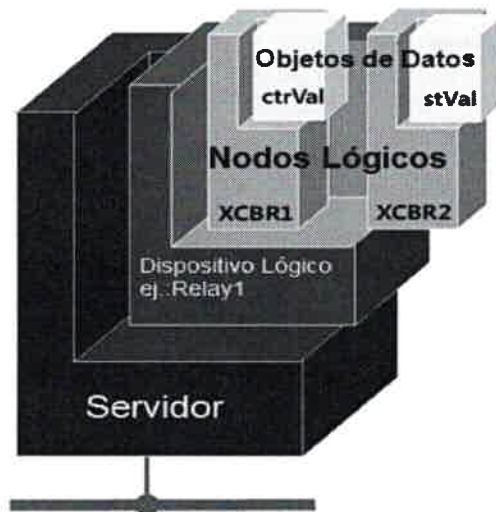


Figura 6-16 - Ejemplo de la estructura de un IED

De esta forma, un dato específico de esa estructura puede referirse según la norma IEC61850 de acuerdo a la sintaxis de la Figura 6-17.

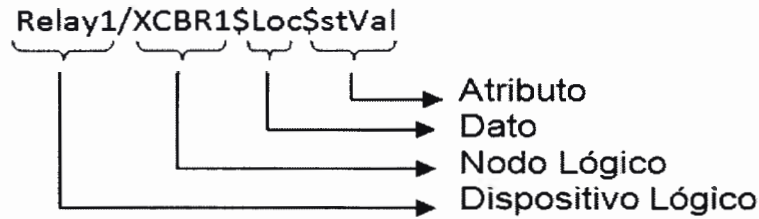


Figura 6-17 - Referenciamiento a un dato específico

6.4.3. Sintaxis y elementos de los archivos SCL

Este lenguaje está basado en el Lenguaje de Marcas Extensible (XML, por sus siglas en inglés, *eXtensible Markup Language*) [51], la norma establece que el archivo SCL debe contener las siguientes secciones [76]:

- *Header* , Encabezado
- Descripción de la subestación
- Descripción del IED
- Descripción del sistema de comunicaciones
- Plantilla de tipo de datos (DataType template)

6.4.4. Tipos de Archivos SCL

En la Sección 6.4.3 se presentó la estructura de los archivos SCL, los mismos son utilizados para intercambiar datos de configuración entre diferentes herramientas de ingeniería. Existen al menos cuatro propósitos diferentes para el intercambio de datos SCL, y por lo tanto, cuatro tipos de archivos distintos para el intercambio de datos entre herramientas. Los mismos se diferencian por la extensión del archivo [79] [52]:

- Archivo con extensión *.ICD* (*IED Capability Description*, descripción de las capacidades del IED): describe las capacidades de un IED. Debe contener exactamente una sección IED en donde se describen sus capacidades y debe contener también definiciones de tipos de LNs. Puede contener una sección de subestación, en dicho caso la vinculación de las instancias de LN con equipamiento primario indica una funcionalidad predefinida. Podría estar también opcionalmente una sección de comunicaciones definiendo posibles direcciones por defecto del IED. Se utiliza para el intercambio de datos desde la herramienta de configuración del IED a la herramienta de configuración del sistema (ver Figura 6-13).

- Archivo con extensión *.SSD (System Specification Description*, descripción de las especificaciones del sistema): describe en forma de diagrama unifilar a la subestación y los LN requeridos. Debe contener una sección de descripción de la subestación y la definición tipos de LN necesarios. Si parte del SAS es ya conocido, puede opcionalmente estar contenido en las secciones de IED y comunicaciones. Se utiliza para el intercambio de datos desde la herramienta de especificación del sistema a la herramienta de configuración del sistema (ver Figura 6-13).
- Archivo con extensión *.SCD (Substation Configuration Description*, descripción de la configuración de la subestación): contiene a todos los IED, sección de configuración de comunicaciones y una sección de descripción de la subestación. Se utiliza para el intercambio de datos desde la herramienta de configuración del sistema a herramientas de configuración de los IED (ver Figura 6-13).
- Archivo con extensión *.CID (Configured IED Description*, descripción del IED configurado): describe un IED instanciado dentro de un proyecto específico. La sección de comunicación contiene la dirección actual del IED. La sección de subestación relacionada con éste IED puede estar presente y en éste caso debe tener el nombre asignado de acuerdo al proyecto. Se utiliza para el intercambio de datos desde la herramienta de configuración del IED al IED (ver Figura 6-13).

Un IED que trabaje bajo la norma IEC61850 debe estar acompañado por un archivo ICD o por una herramienta capaz de generarlo, y debe ser capaz de tomar la parte de comunicación a partir de un archivo SCD (respectivamente acompañado por una herramienta de configuración) [76].

6.4.5. Secciones presentes en cada tipo de archivo SCL

Dependiendo el tipo de archivo, algunas secciones son obligatorias, otras no deben ir y algunas son opcionales, esto se muestra en la Tabla VI- 4.

Tabla VI- 4 - Secciones en los archivos SCL

	Tipo de archivo SCL			
	SSD	ICD	SCD	CID
Encabezado	Si	Si	Si	Si
Subestación	Si	Opcional	Si	Opcional

Comunicación	Opcional	Una instancia	Si	Una instancia
IED	Opcional	Si, valores opcionales	Múltiples	Si, incluyendo valores
Plantilla de tipo de datos	Si fuera necesario	Si fuera necesario	Múltiples	Si fuera necesario

6.5. Conclusiones

En este capítulo se presentó el estándar IEC61850 de manera formal. Se discutieron aspectos del estándar, precisamente relacionados con comunicaciones. También se describió la estructura de datos que especifica la norma y se expuso cómo esa estructura es mapeada a protocolos concretos. Además, se presentó el lenguaje de configuración del sistema especificado por la norma.

En los siguientes capítulos se realizará una aplicación utilizando esta norma.

VII. DISEÑO GENERAL DE UN SISTEMA DE AUTOMATIZACIÓN DE SUBESTACIONES BAJO LA NORMA IEC61850 EN LAS SUBESTACIONES DE LA UNRC

7.1. Introducción

En el capítulo anterior, se discutieron distintos aspectos definidos en la norma IEC61850 tales como modelos de datos y protocolos, y aspectos globales relacionados con la planificación de un Sistema de Automatización de Subestaciones (SAS) y con el lenguaje de configuración de subestaciones que define la norma.

En este capítulo se presentará una aplicación real destinada al sistema de distribución eléctrica de la Universidad Nacional de Río Cuarto (UNRC). Específicamente, se describirá el diseño y la implementación de un Dispositivo Electrónico Inteligente (IED, por sus siglas en inglés, *Intelligent Electronic Device*) que trabaja bajo los conceptos de la norma IEC61850.

El capítulo está organizado siguiendo las pautas de diseño especificadas en la mencionada norma. De esta manera, en primer lugar, en la Sección 7.2, se presentará la planificación del SAS y se situarán los Nodos Lógicos (LN, por sus siglas en inglés, *Logical Node*) en el diagrama unifilar del sistema. A continuación, en la Sección 7.3, mediante el uso de software específico, se describe la generación de los archivos de Lenguaje de Configuración de Subestaciones (SCL, por sus siglas en inglés, *Substation Configuration Language*) del sistema. Luego, en la Sección 7.4 se justificará la selección de los equipos comerciales necesarios. Finalmente, en la Sección 4 se describirá el diseño y la implementación de una aplicación cliente servidor que simula el funcionamiento de un IED genérico bajo los conceptos de la norma IEC61850.

7.2. Diagrama esquemático del sistema de automatización

Como primer paso en la planificación del SAS, en esta sección se realiza un diagrama esquemático que incluye a los LN en el sistema. En el caso particular del sistema elegido, las protecciones seleccionadas en cada subestación son de sobrecorriente. El análisis se realizó para la subestación principal denominada subestación IPSEP, descrita en la Figura 4-4; no obstante, el análisis también es válido para todas las otras subestaciones.

En la Figura 7-1 se ilustra el esquema del SAS propuesto para esta subestación, donde, de acuerdo a la nomenclatura de la norma, se utilizaron los siguientes LN:

- IHMI: panel del operador.
- TVTR: contiene los datos adquiridos de un transformador de tensión, es su controlador.
- TCTR: contiene los datos adquiridos de un transformador de corriente, es su controlador.
- XCBR: es utilizado para accionar en un interruptor, es su controlador.
- XSWI: es utilizado para accionar en un interruptor, es su controlador.
- PTOC: protección por sobrecorriente.
- MMXU: adquiere los valores de los transformadores de tensión y de corriente a través de sus controladores TVTR y TCTR respectivamente.

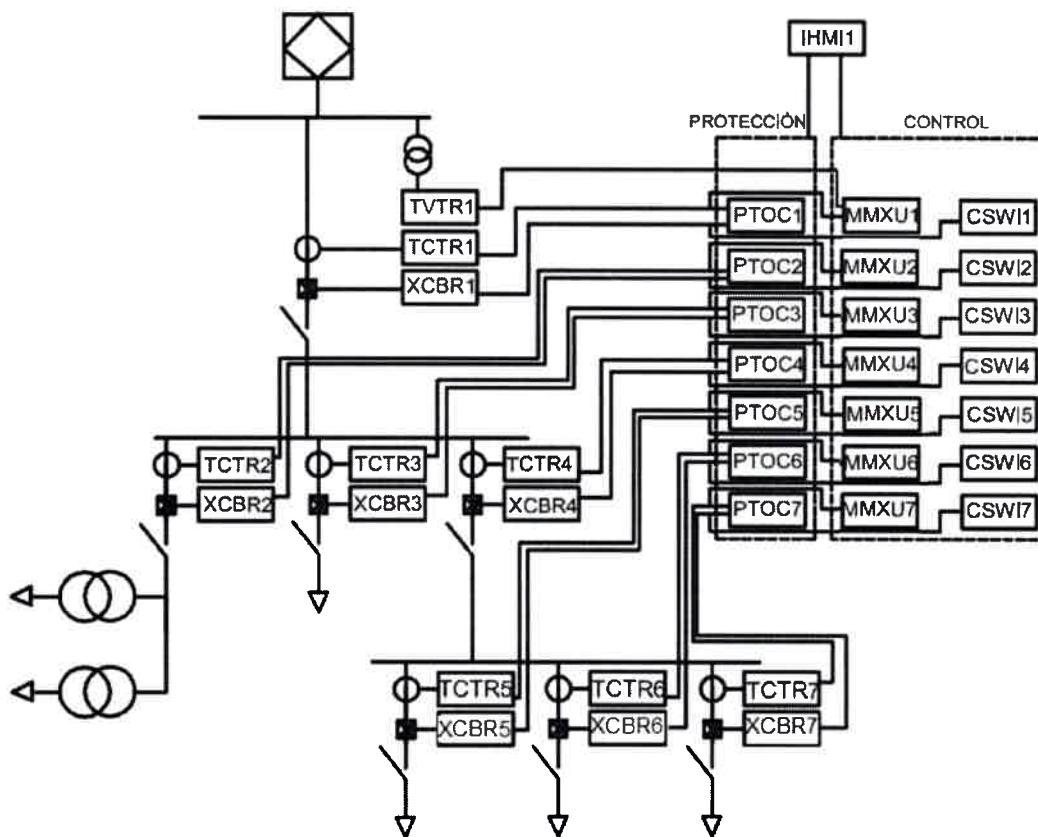


Figura 7-1 - Diagrama esquemático del Sistema de Automatización de Subestaciones

reducido

En la Figura 7-1 se observan LN de nivel de proceso, TCTR, TVTR y XCBR; LN de nivel de bahía, PTOC, MMXU y CSWI; y por último un LN a nivel de subestación, IHMI, que representa a la consola local. En la siguiente sección, se explicará el próximo paso de planificación y diseño del SAS de acuerdo a la norma.

7.3. Generación de archivos SCL

En esta etapa, se describe el diseño para todo el sistema de distribución incluyendo no solo la subestación principal, sino también el resto de subestaciones existentes.

En base al diagrama unifilar de la Figura 4-4 y al diagrama esquemático que se ilustra en la Figura 7-1 se generaron todos los archivos SCL necesarios para el sistema. El diseño se basó en los conceptos de la norma ya descritos en la Sección 6.4.4 del Capítulo 6.

Los archivos SCL se generaron utilizando el programa KALKI SCL Manager versión demo [80].

En primer lugar, se realizó el archivo SSD (*System Specification Description*, Descripción de las Especificaciones del Sistema) que representa al diagrama unifilar de las subestaciones con los requerimientos de LN que representan funciones específicas del sistema de automatización. En la Figura 7-2 se muestra el diagrama generado de acuerdo al formato SSD. La parte izquierda de la figura muestra que ya se ha incluido la sección de comunicación y de IED. Cabe aclarar que tal como se explicó en la Sección 6.4.5 del Capítulo 6, estas secciones son opcionales.

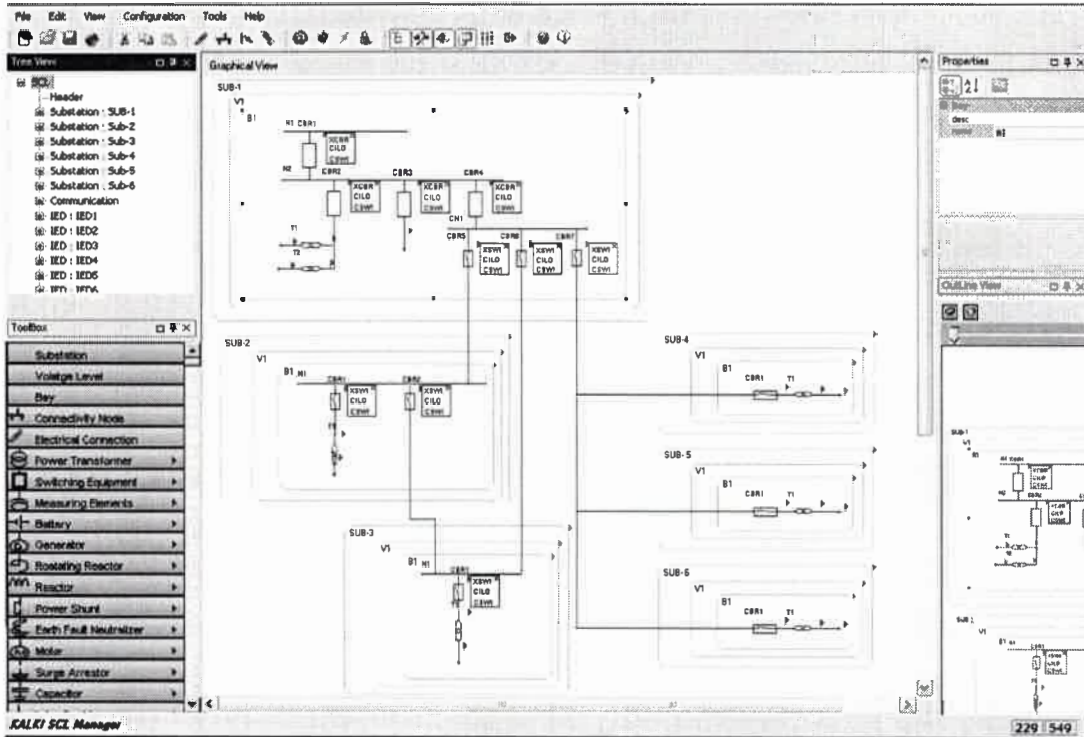


Figura 7-2 - Diagrama generado con KALKI SCL MANAGER

Los archivos ICD (*IED Capability Description*, Descripción de las Capacidades del IED) de los IED que componen el sistema son genéricos, y se obtuvieron a partir del software mencionado; en general éstos son provistos por el fabricante. De esta manera ubicamos en el sistema a los IED necesarios para cumplir con los requerimientos impuestos en el archivo SSD. En esta etapa también se configuraron los parámetros de red de los IED. Luego de realizar los enlaces correspondientes entre los LN de los IED y los LN impuestos como requisitos en el archivo SSD, se obtuvo el archivo SCD (*Substation Configuration Description*, Descripción de la Configuración de la Subestación).

En este punto, al haber realizado las configuraciones específicas de los IED que constituyen el sistema de automatización, se generaron los archivos CID (*Configured IED Description*, Descripción del IED Configurado) que deberán luego ser cargados a los IED correspondientes.

En la siguiente sección se describe la próxima fase del proceso diseño que consiste en la selección de los equipos necesarios.

7.4. Selección de equipos

En la Figura 7-1 se detallan las funcionalidades del SAS propuesto para la subestación IPSEP. Cabe aclarar que los diferentes números de los LN de la Figura 7-1 corresponden a distintos relés; a modo de ejemplo, los nodos lógicos TVTR1, TCTR1, PTOC1, MMXU1 y CSWI1 corresponden al relé número 1 instalado en la subestación IPSEP.

Existen diversos relés comerciales que cumplen con los requisitos necesarios, entre los que se pueden mencionar al relé trifásico SIEMENS SIPROTEC Compact 7SJ80 [81], al relé modelo RED 670 de ABB [82] o REF15 [83] y al modelo 350 de General Electric [84] que trabajan con la norma IEC61850 y poseen todos los LN que se necesitan para esta implementación particular. No obstante esta variedad, se seleccionó el relé de la empresa SIEMENS debido a que se prevé que el mismo sea donado a la UNRC. En el Anexo C se puede ver la hoja de datos del relé seleccionado.

En base al equipo seleccionado, se realizó la implementación de una aplicación cliente servidor que emula a un IED genérico.

7.5. Diseño e Implementación piloto del IED

En esta sección se ha acotado el problema de diseño para realizar una aplicación que cumpla con los aspectos más relevantes de la norma, de acuerdo a lo discutido previamente en el Capítulo 6. El motivo de la simplificación radica en la posibilidad de describir en forma didáctica dichos aspectos, dejando de lado soluciones comerciales para realizar una implementación piloto; esto permitirá continuar con posteriores tareas de desarrollo e investigación.

Para ello, se realizó la implementación parcial de un IED que trabaja bajo la norma IEC61850, cuyo diseño es genérico, de manera que posibilita la expansión del modelo en el futuro.

Primero, se realiza una explicación general del problema para luego ilustrar la aplicación desarrollada.

En la Figura 7-3 [85] se muestra un ejemplo compuesto por dos IED, uno a nivel de proceso que está compuesto por los LN XCBR y TCTR, y otro a nivel de bahía que contiene a los LN CSWI y PIOC, a su vez el SCADA local involucra un LN a nivel de subestación que es IHMI.

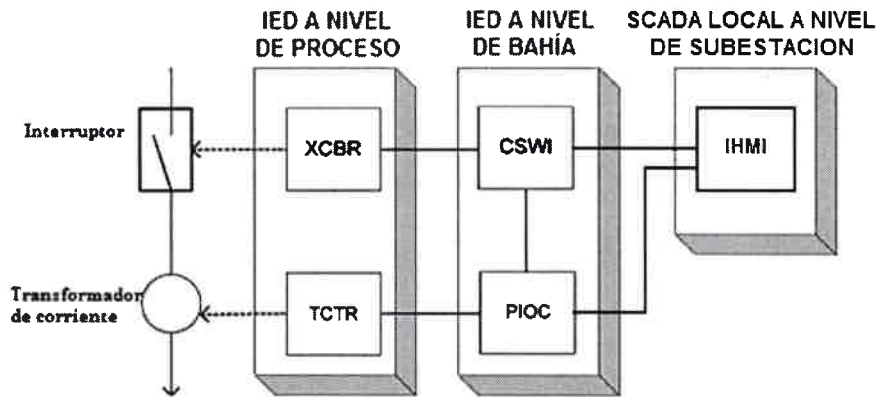


Figura 7-3 - Esquema general para el Sistema de Automatización de Subestaciones

Los LN a nivel de proceso son operados o utilizados por otros LN a nivel de bahía, como por ejemplo un CSWI que está definido para controlar a un XCBR y PIOC que utiliza a TCTR en la obtención de valores RMS para poder operar en el caso que se excedan ciertos límites de corriente sobre el XCBR por medio del CSWI. A su vez el LN IHMI puede acceder a los valores de los LN CSWI y PIOC, y en el caso del primero también puede operar sobre él. En la Sección 5.4.2 del Capítulo 5 se explicó con más detalle este tema.

Analizando la Figura 7-3 de izquierda a derecha, en la figura Figura 7-4 [75] se muestra la relación entre los LN de nivel de proceso y los elementos de la subestación. Tomando como ejemplo al LN XCBR1 se muestran los parámetros más importantes del mismo.

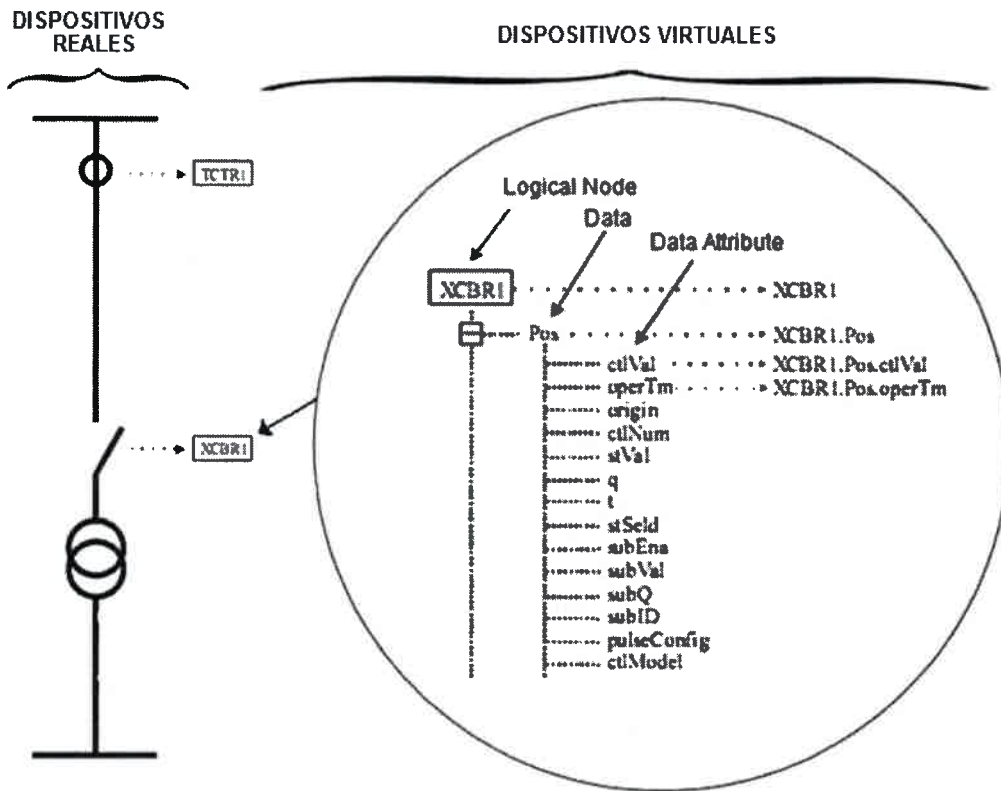


Figura 7-4 - Dispositivos reales y dispositivos virtuales.

De esta forma, desde un LN a nivel de un nivel superior se podría controlar al interruptor por medio de XCBR1.

En base a lo detallado en la Figura 7-3 y la Figura 7-4 se realizó el diseño y la implementación del IED, tanto la parte servidor ACSI como cliente ACSI; desarrollando el modelo de información ACSI y sus servicios. Si bien no se realizó la implementación completa de la norma; la aplicación sirve para comprenderla conceptualmente, ver cómo es la estructura del sistema y el intercambio de información entre las partes.

En referencia a la Figura 7-3, en la Figura 7-5 se muestra un diagrama que indica cuáles son las partes que actúan de servidor y de cliente en el sistema, tanto en un IED como así también en el SCADA.

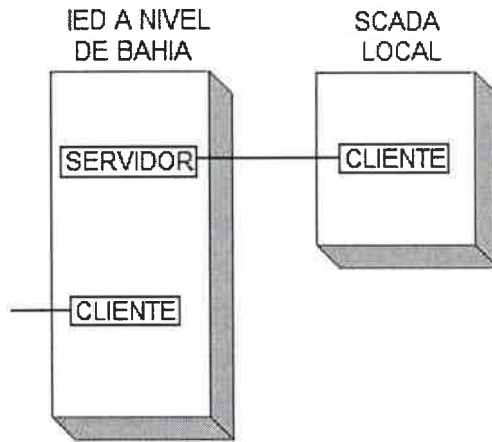


Figura 7-5 - Diagrama UML que indica las partes cliente y servidor en el sistema.

Como se observa, el SCADA local accede como cliente a la parte servidor del IED a nivel de bahía, éste a su vez accedería a la parte servidor del IED a nivel de proceso, el cual está conectado directamente a los equipos de medición y protección.

Esta aplicación se centra en la comunicación entre el cliente del SCADA local y el servidor del IED de nivel de bahía para el diseño.

En la siguiente subsección se verá, de manera modular, cómo inicializar al sistema para formar el modelo de objetos del modelo de información con sus servicios de manera adecuada, y luego se tratarán aspectos de comunicación.

La implementación se realizó en el sistema operativo Microsoft Windows XP, y el lenguaje de programación utilizado tanto para el cliente como para el servidor es c++, utilizando el entorno de desarrollo Microsoft Visual Studio. Además se desarrolló un cliente en Matlab para probar la interoperabilidad del sistema en otros entornos.

7.5.1. Parte servidor - Arquitectura de software

En esta sección se presenta el proceso de diseño y la implementación de la parte servidor ACSI del IED, que se dividirán en diferentes módulos encargados de tareas específicas. Los mismos son:

- Módulo principal: utiliza los otros módulos del sistema de manera coordinada para realizar la tarea completa del servidor.
- Módulo de Inicialización SCL: se encarga de extraer los parámetros de configuración a partir de un archivo .ICD que da la descripción del IED

configurado; luego éste modulo pasa dichos parámetros a los restantes para continuar con la correcta inicialización del sistema.

- Módulo del Modelo de Información: es el que instancia a las clases correspondientes para formar el modelo completo especificado por el archivo .ICD a través de los parámetros proporcionados por el módulo principal.
- Módulo del Modelo de Comunicación: se encarga de la comunicación y transmisión de los datos a través de la red, utiliza parámetros de red proporcionados por el modulo principal.

En las siguientes subsecciones se detallan estos módulos.

a) Módulo Principal

Como se explicó anteriormente, este módulo se encarga de utilizar a los otros módulos del sistema para realizar la tarea completa del servidor. Es el proceso que coordina las diferentes acciones.

Este módulo involucra el código que junta a todos los módulos. El tipo de programación utilizada en este módulo es secuencial.

b) Módulo de Inicialización SCL

Como se detalló en el Capítulo 6, SCL es un lenguaje de configuración de subestaciones especificado por la norma IEC61850. Existen cuatro tipos de archivos SCL, en este caso se utilizará el .ICD que da la descripción del IED configurado.

Como el SCL está basado en el formato XML, los archivos se podrían generar con algún editor de textos, pero existen programas que permiten realizarlos de forma más adecuada. Se utilizó el programa *Kalki SCL Manager* [80], que es un editor gráfico que permite realizar los cuatro tipos de archivos SCL, y posee una sección de validación de los archivos, lo cual evita errores de sintaxis.

El Módulo de Inicialización SCL extrae la información de red (dirección IP, Mascara, Gateway) y del contenido del servidor (dispositivos lógicos, nodos lógicos, etc.), la misma es luego utilizada en los otros módulos.

c) Módulo del Modelo de Información

La norma sugiere en sus definiciones una implementación orientada a objetos casi directa de su modelo de información, tal como se explicó en la Sección 6.2.

El modelo de información ACSI se muestra en la Figura 6-2. El objeto de más alto nivel contenido en un IED es el servidor. Cada servidor contiene uno o más Puntos de Acceso (AP, por sus siglas en inglés, *Access Point*) y un modelo de datos compuesto de dispositivos lógicos, nodos lógicos, datos y atributos de datos.

El modelo es jerárquico de forma anidada y el servidor es el objeto de más alto nivel en el modelo. Un objeto de la clase servidor contiene objetos de la clase dispositivo lógico, a su vez, cada uno de éstos contiene objetos de la clase nodo lógico, los mismos contienen objetos de la clase dato y finalmente éstos últimos contienen objetos de la clase atributos de dato. Cada clase, con sus atributos y métodos, está declarada en un archivo *.h* y la implementación de la misma está en un archivo *.cpp* del mismo nombre. Los atributos son nombre, referencia en el modelo, cantidad de objetos del nivel inferior que contiene, entre los más importantes.

Los nombres que son utilizados en esta implementación para las clases que componen al modelo de información ACSI son: *server* para el servidor, *LD* para los dispositivos lógicos, *LN* para los nodos lógicos, *Data* para los datos y *DataAttribute* para los atributos de datos.

Éste modelo de información ACSI tiene asociados servicios ACSI (ver Sección 6.2.3). Los mismos son: *GetServerDirectory*, *GetLogicalDeviceDirectory*, *GetLogicalNodeDirectory*, *GetAllDataValues*, *GetDataValue* y *SetDataValue* (los objetos de las otras clases tienen solo un método para su configuración inicial, pero el mismo es llamado desde el objeto que está en el nivel superior de la jerarquía).

La clase *server* es la más importante del modelo, ya que contiene todos los métodos disponibles para los clientes, desde la configuración de algún parámetro hasta la obtención de los datos adquiridos. De esta forma, la clase *server* es la interfaz del modelo completo

En el diagrama de la Figura 7-6 se muestran los métodos y atributos de la clase *server*, como así también los de las otras clases que forman el modelo de información ACSI básico.

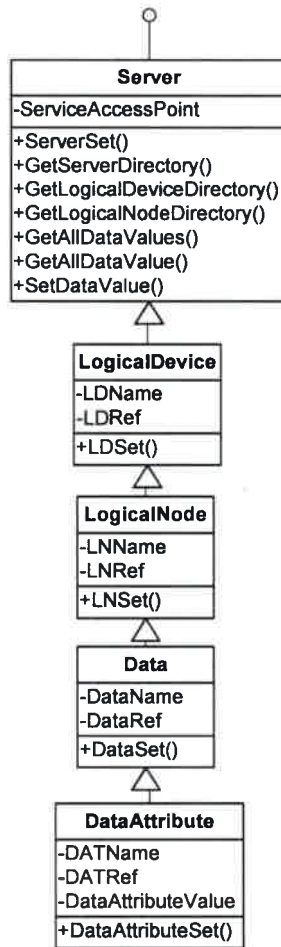


Figura 7-6 - Diagrama UML del modelo de información ACSI básico.

En la Figura 7-7 se muestra el más básico que es *GetServerDirectory*, y en la Figura 7-8, el código de dicho método implementado en c++.

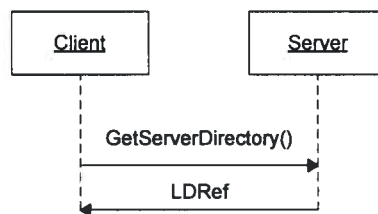


Figura 7-7 - Diagrama UML del intercambio de información para el método *GetServerDirectory* de la clase *Server*.

```

1 void Server::GetLogicalDeviceDirectory(char LDRef[20])
2 {
3     for (i=0;i<LDNum-1;i++)
4     {
5         if ( !strcmp(LD[i].LDRef,LDRef))
6         {
7             strcat(response_aux,"");
8             for (j=0;j<LD[i].LNNum;j++)
9             {
10                strcat(response_aux,LDRef);
11                strcat(response_aux,"/");
12                strcat(response_aux,LD[i].LN[j].LNName);
13                strcat(response_aux,"");
14            }
15        }
16    }
17    strcpy(response,response_aux);
18 }

```

Figura 7-8 - Código en c++ del método GetServerDirectory de la clase Server.

d) Módulo del Modelo de Comunicación

De acuerdo al estándar, el modelo de información ACSI de un IED debe estar acompañado de un Mapeo Específico de los Servicios de Comunicación (SCSM por sus siglas en inglés, *Specific Communication Service Mapping*), que suministre una plataforma de comunicación entre el servidor y los clientes. El estándar propone, en su SCSM, el mapeo de los servicios ACSI al protocolo MMS (*Manufacturing Message Specification*, especificación de mensajes de fabricación), tal como se vio en la Sección 6.3.

En este caso se estudió el uso de algunos SCSM que podrían ser mas adecuados, como Servicios Web (*Web Services*) o el protocolo de administración de red simple (SNMP, por sus siglas en inglés, *Simple Network Management Protocol*), ya que se están utilizando en diversas áreas y presentan muchas características novedosas y muy poderosas frente a MMS.

En este trabajo, la prioridad no es la implementación de un SCSM, por lo que se adoptó un esquema de comunicación robusto más simple de implementar que se adapte a las necesidades del sistema. Por lo tanto, se adoptó el esquema sockets TCP, que permite la comunicación de diferentes dispositivos y servicios a través de una red TCP/IP. Con éste método de comunicación, los clientes pueden comunicarse con cualquier IED a través de internet.

De esta manera, el servidor está a la espera de la petición de información de los clientes en un puerto TCP, utilizando el modelo de red TCP/IP.

La sintaxis utilizada en los mensajes cliente/servidor transmitidos está bajo la norma IEC61850.

La clase *Communication* se encarga del módulo de comunicación. En la Figura 7-9 se muestra el diagrama de la clase con sus atributos y métodos.

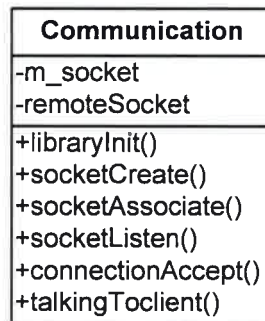


Figura 7-9 - Diagrama UML de la clase Communication.

7.5.2. Arquitectura de software de la parte cliente

En esta sección se define el cliente ACSI, que puede ser aplicable a un IED en su parte cliente o al SCADA local. El mismo se limita a una aplicación que sirve para probar la performance de la parte servidor, de manera que sea capaz de acceder a los servicios provistos por el mismo.

En primer lugar, el cliente debe utilizar el mismo protocolo de comunicación que el servidor, por lo que se utiliza también socket TCP para la comunicación, siguiendo también la misma sintaxis que utiliza el servidor.

De esta forma el cliente envía la consulta al servidor y, luego de procesar dicha petición, el servidor le responderá al cliente la información requerida.

Se implementó el módulo de comunicación para realizar consultas y recibir las respuestas del server, pidiendo en pantalla el ingreso de la consulta. Esto se realizó tanto en Matlab como en c++ para demostrar la flexibilidad de interactuar con distintos tipos de programas.

7.5.3. Prueba de los servicios implementados en el servidor

En ésta sección se ilustra el funcionamiento del servidor bajo la norma IEC61850 desarrollado utilizando los clientes de prueba implementados en Matlab y c++.

En primer lugar, en la Figura 7-10 y la Figura 7-11 se muestran las capturas de pantalla de los clientes de prueba de Matlab y c++ respectivamente.

```

MATLAB 7.6.0 (R2006a)
File Edit Debug Parallel Desktop Window Help
>> clientML
Request: 'Server1.GetServerDirectory("");'
Response =
LD1
LD2
Request: 'Server1.GetLogicalDeviceDirectory("", LD1, "");'
Response =
LD1.XCBR1
LD1.XCBR2
Request: 'Server1.GetLogicalDeviceDirectory("", LD2, "");'
Response =
LD2.TVTR1
LD2.TVTR2
Request: 'Server1.GetLogicalNodeDirectory("", LD1.XCBR1, "", "Data,");'
Response =
Pos.stVal
Pos.ctVal
Pos.q
Pos.t
Request: 'Server1.SetDataValue("", LD1.XCBR1.Pos.ctVal, "", "1.00");'
Response =
ok
Request: 'Server1.GetDataValue("", LD1.XCBR1.Pos.ctVal, "");'
Response =
1
Request: 'Server1.GetAllDataValues("", LD1.XCBR1, "");'
Response =
0
1
0
0
Request:
+ Start | Waiting for input
  
```

Figura 7-10 - Cliente desarrollado en Matlab.

```

E:\mi_tesis\IEC_61850_v_1.03\IEC_61850_v_1.04\IEC_61850_v_1.01\B...
Consulta:
Server1.GetServerDirectory("");
se le consulto al server: " Server1.GetServerDirectory(""); "
Se recibio el siguiente mensaje del server: " LD1.LD2.

Consulta:
Server1.GetLogicalDeviceDirectory("", LD1 "");
se le consulto al server: " Server1.GetLogicalDeviceDirectory("", LD1 ""); "
Se recibio el siguiente mensaje del server: " LD1.XCBR1 LD1.XCBR2.

Consulta:
Server1.GetLogicalNodeDirectory("", LD1.XCBR1, "", "Data.");
se le consulto al server: " Server1.GetLogicalNodeDirectory("", LD1.XCBR1, "", "Data."); "
Se recibio el siguiente mensaje del server: " .Pos.stVal Pos.ctVal Pos.q.Pos.t.

Consulta:
Server1.SetDataValue("", LD1.XCBR1.Pos.ctVal, "", "1.00");
se le consulto al server: " Server1.SetDataValue("", LD1.XCBR1.Pos.ctVal, "", "1.00") "
Se recibio el siguiente mensaje del server: " ok.

Consulta:
Server1.GetDataValue("", LD1.XCBR1.Pos.ctVal ");
se le consulto al server: " Server1.GetDataValue("", LD1.XCBR1.Pos.ctVal "); "
Se recibio el siguiente mensaje del server: " .1.000000.

Consulta:
Server1.GetAllDataValues("", LD1.XCBR1, "");
se le consulto al server: " Server1.GetAllDataValues("", LD1.XCBR1 "); "
Se recibio el siguiente mensaje del server: " 0.000000 1.000000 0.000000 0.000000
00.

Consulta:
  
```

Figura 7-11 - Cliente desarrollado en c++.

Finalmente, en la Figura 7-12 se muestra el resultado que se visualiza en la consola del IED, desde su inicialización hasta el registro (*log*) de las consultas y respuestas que se hacen desde el cliente de prueba.

```

Server created
Server1
->LD1
  ->XCBBR1
    ->Pos
      ->stVal
      ->ctVal
      ->q
      ->t
    ->XCBBR2
      ->Pos
        ->stVal
        ->ctVal
        ->q
        ->t
  ->LD2
    ->TUTR1
      ->instMag
        ->i
        ->f
        ->range_l
        ->range_u
      ->q
      ->t
    ->TUTR2
      ->instMag
        ->i
        ->f
        ->range_l
        ->range_u
      ->q
      ->t
->

Server1.GetServerDirectory(""):
.LD1.LD2
Server1.GetLogicalDeviceDirectory("LD1"):
.LD1.XCBBR1.LD1.XCBBR2
Server1.GetLogicalDeviceDirectory("LD2"):
.LD2.TUTR1.LD2.TUTR2
Server1.GetLogicalNodeDirectory("LD1.XCBBR1","Data"):
.Pos.stVal.Pos.ctVal.Pos.q.Pos.t
Server1.SetDataValue("LD1.XCBBR1.Pos.ctVal","1.00"):
ok
Server1.GetDataValue("LD1.XCBBR1.Pos.ctVal"):
1.000000
Server1.GetAllDataValues("LD1.XCBBR1"):
0.000000 1.000000 0.000000 0.000000
  
```

Figura 7-12 - Consola de la parte servidor del IED.

Las figuras anteriores ilustran el comportamiento del servidor y la interoperabilidad con los clientes, esta plataforma sirve para observar el intercambio de información entre los componentes del sistema.

7.6. Conclusiones

En este capítulo se aplicaron los conceptos estudiados en el Capítulo 6 mediante la implementación de los mismos en el sistema de distribución de energía eléctrica de la UNRC. Primeramente se describió la planificación del sistema de automatización especificando los LN requeridos. Luego se detalló cómo utilizar un software específico para generar los archivos SCL del sistema. A continuación, se

justificó la selección de los equipos necesarios, y finalmente, se describieron los pasos para el diseño y la implementación de un IED que trabaja bajo los conceptos de la norma IEC61850, realizando un sistema piloto tanto en Matlab como en c++, de forma tal de visualizar la transmisión de los datos y comprender conceptualmente el funcionamiento de la norma.

Debido a que esta aplicación servirá de base para la capacitación de estudiantes de ingeniería, la implementación se realizó en forma simple y didáctica, de manera de que sirva para implementar laboratorios en una materia opcional referida a la automatización de subestaciones.

Este capítulo ejemplificó mediante la descripción de un desarrollo específico, los conceptos teóricos tratados en el Capítulo 6, en varios aspectos, desde planificación global de un sistema de automatización hasta el diseño e implementación de un IED.

VIII. CONCLUSIONES

8.1. Introducción

Desde sus comienzos, los Sistemas de Supervisión, Control y Adquisición de Datos (SCADA, por sus siglas en inglés, *Supervisory Control And Data Acquisition*) han presentado una importante evolución en aspectos relacionados con el software, el hardware y los sistemas de comunicación.

Más específicamente, los sistemas de comunicación en sistemas SCADA han experimentado un gran avance, desde cambios en los paradigmas de funcionamiento, las tecnologías de red y los protocolos de comunicación, hasta la incorporación de dispositivos que permiten trabajar en sistemas de automatización. Por otra parte, las redes eléctricas también han experimentado cambios; pues, con la generación distribuida y las nuevas demandas de los usuarios, emerge la necesidad de incorporar nuevas tecnologías y sistemas de comunicación, para brindar un servicio acorde a las exigencias de los clientes.

En este trabajo se trataron temas referentes a los sistemas de comunicación utilizados en sistemas SCADA, Automatización de Subestaciones, SmartGrids y los estándares de comunicación utilizados en estos ambientes.

En el Capítulo 1, se realizó la introducción general al trabajo de tesis.

En el Capítulo 2, se presentó la evolución de los sistemas SCADA, su arquitectura física, las diferentes arquitecturas de software y las redes de comunicación utilizadas en estos sistemas.

En el Capítulo 3, se efectuó una introducción a los sistemas de comunicación y se expuso la forma en que éstos evolucionaron en el área de los sistemas eléctricos de potencia, desde sistemas propietarios hasta los actuales sistemas que persiguen la interoperabilidad utilizando estándares internacionales.

En el Capítulo 4, se describió el sistema SCADA diseñado para los equipos de protección actualmente utilizados en la subestación principal de la UNRC, denominada subestación IPSEP. Para el sistema diseñado se utilizó el protocolo de comunicaciones Modbus TCP. En este capítulo, se presentó de forma detallada al sistema de distribución eléctrica de la UNRC. Además, se definieron los módulos de comunicación necesarios para que el sistema funcione con el protocolo de comunicación propuesto, y se expuso también el desarrollo de la arquitectura de software del sistema SCADA, la cual

comprende una aplicación principal desarrollada en Labview, servidores de bases de datos, servidor de páginas WEB y otros lenguajes.

En el Capítulo 5 se realizó una introducción a los sistemas de automatización de subestaciones eléctricas, presentando al estándar que representa al estado del arte en el tema, IEC61850.

En el Capítulo 6 se presentó y se analizó en detalle al estándar IEC61850; se presentó la estructura de datos definida por la norma, cómo esa estructura es mapeada a protocolos concretos y el lenguaje de configuración que especifica la norma.

En el Capítulo 7, se describió el diseño y la implementación de un Dispositivo Electrónico Inteligente (IED, por sus siglas en inglés, *Intelligent Electronic Device*) que trabaja bajo los conceptos de la norma IEC61850, siguiendo las pautas de diseño especificadas en la misma. De esta manera, primero se realizó la planificación del Sistema de Automatización de Subestaciones (SAS), luego, utilizando un software específico, se generaron los archivos de configuración del sistema y, finalmente, se presentó el diseño y la implementación de una aplicación cliente servidor que simula el funcionamiento de un IED genérico bajo los conceptos de la norma IEC61850.

Se prevé que este trabajo de tesis sirva de base para la formación de estudiantes de ingeniería eléctrica por medio de prácticas de laboratorio en una materia opcional sobre sistemas de comunicación y automatización de subestaciones.

En este trabajo se ha demostrado que con la incorporación de herramientas informáticas y lenguajes de programación adecuados se puede llegar a prescindir de la necesidad de adquirir aplicaciones SCADA comerciales, que en la mayoría de los casos son cerradas y no son interoperables con equipos de diferentes fabricantes; en el caso de la aplicación desarrollada en este trabajo, el código se puede modificar para incorporar dispositivos de diferentes fabricantes.

8.2. Trabajos futuros

Sobre la base de las actividades realizadas en su maestría, el tesista desarrollará su doctorado en el tema “Optimización de Redes Inalámbricas Malladas (WMN, por sus siglas en inglés, *Wireless Mesh Network*) Aplicada a Redes Inteligentes de Energía Eléctrica”. En el mencionado trabajo se propone optimizar el funcionamiento de las redes inteligentes de energía eléctrica por medio del uso de un sistema de comunicación que contempla el acceso a la red de forma cableada e inalámbrica de diferentes tecnologías, facilitando el acceso de los dispositivos en redes eléctricas con recursos

distribuidos. Este trabajo de doctorado será financiado a través de una beca doctoral de CONICET.

El trabajo de maestría deja abiertas diferentes líneas de investigación, específicamente, entre las más importantes pueden citarse:

- Migración al lenguaje de programación c++ de la aplicación SCADA desarrollada en Labview.
- Desarrollo de dispositivos de protección y monitoreo para subestaciones eléctricas o cualquier otra área que lo requiera.
- Abordar nuevos sistemas de comunicación adecuados para los sistemas de gestión de energía y redes inteligentes de energía.

8.3. Aportes realizados

Los aportes realizados a que contribuyeron a la realización de la tesis, y que se originaron durante el transcurso de la confección de la misma, se resumen a continuación:

• Conferencias Internacionales:

- [1] R. Prat, F. Magnago y R. Lima, “Aspectos Noveles del Estándar IEC61850,” ERIAC XIII, 2009.
- [2] R. Prat, F. Magnago y G. Rodriguez, “Implementación de los Modelos de Información Básicos definidos por el estándar IEC61850,” AADECA, 2010.
- [3] R. Prat, F. Magnago y J. Amatti, “Sistemas de Comunicación Adecuados para Redes Inteligentes bajo la Norma IEC61850,” ERIAC XIV, 2011.
- [4] R. Prat, F. Magnago y G. Campetelli, “Planificación de un Sistema de Automatización de Subestaciones para la Universidad Nacional De Río Cuarto basado en la Norma IEC61850,” ERIAC XIV, 2011.

• Revistas Internacionales

- [1] R. Prat, G. Rodriguez y F. Magnago, “Monitoring and Controlling Services for Electrical Distribution Systems based on the IEC61850 standard,” Energy and Power Engineering Journal, vol. 3, pp. 137-147, 2011.

REFERENCIAS

- [1] E. Santacana, G. Rackliffe, T. Le y F. Xiaoming, "Getting Smart," *Power and Energy Magazine*, IEEE, vol. 8, no. 2, pp. 41-48, 2010.
- [2] A. P. Johnson, "The history of the Smart Grid evolution at Southern California Edison," *Innovative Smart Grid Technologies (ISGT)*, 2010, pp. 1-3.
- [3] L. Jingjing, X. Da y A. Qian, "Research on smart grid in China," *Transmission & Distribution Conference & Exposition: Asia and Pacific*, 2009, 2009, pp. 1-4.
- [4] E. H. P. Chan, M. G. Delson, J. L. Green y C. L. Nash, "A Coordinated Statewide Load Management and Scada System," *Power Apparatus and Systems*, IEEE Transactions on, vol. PAS-102, no. 11, pp. 3496-3501, 1983.
- [5] R. E. Witte y P. P. Schumacher, "A Serial Communications Interface for Microprocessor-Based Power Distribution Apparatus," *Power Apparatus and Systems*, IEEE Transactions on, vol. PAS-104, no. 1, pp. 45-51, 1985.
- [6] "IEEE standard definition, specification, and analysis of systems used for supervisory control, data acquisition, and automatic control," IEEE Std C37.1-1994, 1994.
- [7] M. Chaturvedi, "Substation IED communications," *Power Engineering Society Winter Meeting*, 2002. IEEE, 2002, pp. 596 vol.1.
- [8] F. Crispino, C. A. Villacorta, P. R. P. Oliveira, J. A. Jardini y L. C. Magrini, "An experiment using an object-oriented standard-IEC 61850 to integrate IEDs systems in substations," *Transmission and Distribution Conference and Exposition: Latin America*, 2004 IEEE/PES, 2004, pp. 22-27.
- [9] R. E. Mackiewicz, "Overview of IEC 61850 and Benefits," *Transmission and Distribution Conference and Exhibition*, 2005/2006 IEEE PES, 2006, pp. 376-383.
- [10] V. M. Flores, D. Espinosa, J. Alzate y D. Dolezilek, "Case Study: Design and Implementation of IEC 61850 From Multiple Vendors at CFE La Venta II," *Protective Relay Engineers*, 2007. 60th Annual Conference for, 2007, pp. 307-320.
- [11] C. Hoga, "New Ethernet Technologies for Substation Automation," *Power Tech*, 2007 IEEE Lausanne, 2007, pp. 707-712.
- [12] A. Rodriguez, *Sistemas SCADA*, Vol. 2, Editorial Marcombo, 2007.
- [13] C. E. Gordon Clarke, BEng, MBA, Deon Reynders, Pr.Eng, BSc(ElecEng)(Hons), MBA, and Edwin Wright, BSc, BE(Hons)(Elec), MIPENZ, "Practical Modern SCADA Protocols," *DNP3, 60870.5 and Related Systems* 2003.
- [14] H. L. Smith y W. R. Block, "RTUs slave for supervisory systems [power systems]," *Computer Applications in Power*, IEEE, vol. 6, no. 1, pp. 27-32, 1993.
- [15] B. Quartey, D. Shaw y P. Waked, "An application of PLC's as an RTU in SCADA systems," *Petroleum and Chemical Industry Conference*, 1992, Record of Conference Papers., Industry Applications Society 39th Annual, 1992, pp. 271-274.
- [16] D. Ecob, J. Williamson, G. Hughes y J. Davis, "PLC's and SCADA - a water industry experience," *Application of Advanced PLC (Programmable Logic Controller) Systems with Specific Experiences from Water Treatment*, IEE Colloquium on (Digest No.1995/112), 1995, pp. 6/1-6/10.

- [17] M. Kezunovi, T. Djoki y T. Kosti, "Automated Monitoring and Control Using New Data Integration Paradigm," System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on, 2005, pp. 66a-66a.
- [18] EPRI, *Guidelines for Implementing Substation Automation Using IEC61850, the International Power System Information Modeling Standard*, 2004.
- [19] FreeSCADA. "Software SCADA gratuito," <http://free-scada.org/>.
- [20] ISO/IEC, "Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model," vol. 2, 1994.
- [21] D. Renhui, D. Jie y Z. Yefei, "The Data-View Model of IEC 61850 Server," Power System Technology, 2006. PowerCon 2006. International Conference on, 2006, pp. 1-4.
- [22] "IEEE recommended practice for master/remote supervisory control and data acquisition (SCADA) communications," IEEE Std 999-1992, 1993.
- [23] R. F. Gilbert, J. L. Goodin y K. Nelson, "The benefits of a distribution automation system using the utility communications architecture (UCA)," Rural Electric Power Conference, pp. c3-1-5, 1998.
- [24] D. Damsker, "Multiplexing and Networking Through Fiber Optic Links for Scada Systems," Power Apparatus and Systems, IEEE Transactions on, vol. PAS-101, no. 7, pp. 2383-2389, 1982.
- [25] L. A. Thornton, W. L. Snyder, R. D. Christie y D. H. Bowman, "Energy Control System an Advanced Approach for Transalta Utilities," Power Apparatus and Systems, IEEE Transactions on, vol. PAS-102, no. 10, pp. 3446-3452, 1983.
- [26] D. J. Gaushell y H. T. Darlington, "Supervisory control and data acquisition," Proceedings of the IEEE, vol. 75, no. 12, pp. 1645-1658, 1987.
- [27] D. L. Campbell, "How customer need focused the development of a new remote terminal unit line," Computer Applications in Power, IEEE, vol. 1, no. 3, pp. 9-11, 1988.
- [28] C. Hadjimanolis, "Power remote control-East Coast Main Line," Main Line Railway Electrification, pp. 311-316, 1989.
- [29] U. Sandberg y H. Faxer, "Personal Computers for Real-Time Control of Power System Distribution Networks," Power Apparatus and Systems, IEEE Transactions on, vol. PAS-103, no. 7, pp. 1720-1724, 1984.
- [30] V. C. Gungor y F. C. Lambert, "A survey on communication networks for electric system automation," Computer Networks, vol. 50, no. 7, pp. 877-897, 2006.
- [31] D. J. Hoge y J. R. Jensen, "A comparison of protocol conversion methods for the retrofit of SCADA systems," Petroleum and Chemical Industry Conference, 1988, Record of Conference Papers., Industrial Applications Society 35th Annual, 1988, pp. 245-248.
- [32] P. M. Willson, "Application of systems engineering to large scale water industry telemetry schemes," Systems Engineering Contribution to Increased Profitability, IEE Colloquium on, 1989, pp. 8/1-8/6.
- [33] M. Torre Stella, A. Ramirez Molina y C. Zambrano Duque, "Data interchange among utilities dispatch control centers," Devices, Circuits and Systems, 1995. Proceedings of the 1995 First IEEE International Caracas Conference on, 1995, pp. 154-158.
- [34] T. Duong, "The design of next generation SCADA systems," Power Industry Computer Application Conference, 1995. Conference Proceedings., 1995 IEEE, 1995, pp. 431-436.

- [35] J. Liu y X. Yu, "Open communication and control architecture for EMS in electrical integrated information system," Power System Technology, 1998. Proceedings. POWERCON '98. 1998 International Conference on, 1998, pp. 1285-1289 vol.2.
- [36] D. Proudfoot, "Information technologies for substation automation applications," Transmission and Distribution Conference and Exposition, 2001 IEEE/PES, 2001, pp. 349-361 vol.1.
- [37] V. Skendzic y A. Guzman, "Enhancing Power System Automation Through the Use of Real-Time Ethernet," Power Systems Conference: Advanced Metering, Protection, Control, Communication, and Distributed Resources, 2006. PS '06, 2006, pp. 480-495.
- [38] J. A. Beaupre, M. Lehoux y P. A. Berger, "Advanced monitoring technologies for substations," Transmission and Distribution Construction, Operation and Live-Line Maintenance Proceedings. 2000 IEEE ESMO - 2000 IEEE 9th International Conference on, 2000, pp. 287-292.
- [39] H. Alarian, "EMSNova," Power Engineering Society Summer Meeting, 1999. IEEE, 1999, pp. 872-874 vol.2.
- [40] M. Kezunovic, "Future trends in protective relaying, substation automation, testing and related standardization," Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES, 2002, pp. 598-602 vol.1.
- [41] M. Kwok-Hong y B. L. Holland, "Migrating electrical power network SCADA systems to TCP/IP and Ethernet networking," Power Engineering Journal, vol. 16, no. 6, pp. 305-311, 2002.
- [42] S. Mohagheghi, J. Stoupis y Z. Wang, "Communication protocols and networks for power systems-current status and future trends," Power Systems Conference and Exposition, 2009. PES '09. IEEE/PES, 2009, pp. 1-9.
- [43] IPCOMM. "Protocolos de comunicación para sistemas SCADA," <http://www.ipcomm.de>.
- [44] N. Instruments. "SCADA software," <http://www.ni.com/labview/>.
- [45] Microsoft. "Conector de Base de Datos," <http://support.microsoft.com/kb/110093>.
- [46] MySQL. "Motor de Base de Datos," <http://www.mysql.com/>.
- [47] APACHE. "Servidor WEB," www.apache.org.
- [48] PHP. "Lenguaje multipropósito desarrollado para entornos WEB," <http://www.php.net/>.
- [49] A. Apostolov, "Multi-agent systems and IEC 61850," Power Engineering Society General Meeting, 2006. IEEE, 2006, pp. 6 pp.
- [50] T. J. Hammons, "Integrating renewable energy sources into European grids," International Journal of Electrical Power & Energy Systems, vol. 30, no. 8, pp. 462-475, 2008.
- [51] C. Brunner, "The Impact of IEC 61850 on Protection," Developments in Power System Protection, 2008. DPSP 2008. IET 9th International Conference on, 2008, pp. 14-19.
- [52] M. Paulino, "Testes de IEDs Operando Com Redes de Comunicacao Baseados Na IEC 61850," XII ERIAC, 2007.
- [53] A. Apostolov y B. Vandiver, "Functional Testing of IEC 61850 Based Protection Relays," Protective Relay Engineers, 2007. 60th Annual Conference for, 2007, pp. 333-340.

- [54] M. Paulino, "Liberdade de configuração com funções distribuídas através da decomposição funcional estabelecida pela norma IEC61850" XIII ERIAC, 2009.
- [55] IEC-61850-1, "Communication networks and systems in substations," *Part 1: Introduction and overview*, 2003.
- [56] IEC-61850-7-1, "Communication networks and systems in substations," *Part 7-1: Basic communication structure for substation and feeder equipment – Principles and models*, 2003.
- [57] IEC-61850-7-2, "Communication networks and systems in substations," *Part 7-2: Basic communication structure for substation and feeder equipment – Abstract communication service interface (ACSI)*, 2003.
- [58] IEC-61850-7-3, "Communication networks and systems in substations," *Part 7-3: Basic communication structure for substation and feeder equipment – Common data classes*, 2003.
- [59] IEC-61850-7-4, "Communication networks and systems in substations," *Part 7-4: Basic communication structure for substation and feeder equipment – Compatible logical node classes and data classes*, 2003.
- [60] T. Kostic y C. Frei, "Modelling and using IEC 61850-7-2 (ACSI) as an API," *Power Tech*, 2007 IEEE Lausanne, 2007, pp. 713-719.
- [61] C. Brunner, "IEC 61850 for power system communication," *Transmission and Distribution Conference and Exposition, 2008;D. IEEE/PES*, 2008, pp. 1-6.
- [62] B. Shephard, M. C. Janssen y M. Schubert, "Standardised communications in substations," *Developments in Power System Protection, 2001, Seventh International Conference on (IEE)*, 2001, pp. 270-274.
- [63] X. Tianqi, Y. Xianggen, Y. Dahai, Y. Hongwei, W. Yangguang y H. Hui, "Bay level IED modeling and realizing using IEC 61850," *Transmission and Distribution Conference and Exposition, 2008;D. IEEE/PES*, 2008, pp. 1-7.
- [64] A. Apostolov, "Object Models of Protection Devices in IEC 61850," *Transmission and Distribution Conference and Exhibition, 2005/2006 IEEE PES*, 2006, pp. 915-920.
- [65] C. R. Ozansoy, A. Zayegh y A. Kalam, "The Real-Time Publisher/Subscriber Communication Model for Distributed Substation Systems," *Power Delivery, IEEE Transactions on*, vol. 22, no. 3, pp. 1411-1423, 2007.
- [66] L. Andersson, C. Brunner y F. Engler, "Substation automation based on IEC 61850 with new process-close technologies," *Power Tech Conference Proceedings, 2003 IEEE Bologna*, 2003, pp. 6.
- [67] X. Tianqi, H. Hui, Y. Hongwei, Y. Dahai, Y. Xianggen y W. Yangguang, "Analysis on IEC 61850 Interoperability Support," *Power Engineering Society General Meeting, 2007. IEEE*, 2007, pp. 1-6.
- [68] T. S. Sidhu y Y. Yujie, "Modelling and Simulation for Performance Evaluation of IEC61850-Based Substation Communication Systems," *Power Delivery, IEEE Transactions on*, vol. 22, no. 3, pp. 1482-1489, 2007.
- [69] C. Samitier y R. Pellizzoni, "Grupo de trabajo conjunto "IEC61850" en el marco de la Región Iberoamericana de Cigré (RIAC)," XIII ERIAC, 2009.
- [70] IEC-61850-8-1, "Communication networks and systems in substations," *Part 8-1: Specific Communication Service Mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3*, 2004.
- [71] L. Hossenlop, "Engineering perspectives on IEC 61850," *Power and Energy Magazine, IEEE*, vol. 5, no. 3, pp. 45-50, 2007.

- [72] A. Rivas, "IEC 61850 - Retos y Oportunidades en su Implantación," XII ERIAC, 2007.
- [73] L. Hossenlop y H. R., "IEC 61850 - Perspectivas de Operación," XII ERIAC, 2007.
- [74] R. Paulo y A. Carvalho, "Design oriented architecture for integration tools case study of substation automation systems," Industrial Technology, 2004. IEEE ICIT '04. 2004 IEEE International Conference on, 2004, pp. 874-879 Vol. 2.
- [75] G. Igarashi y E. Senger, "Estudo da Norma IEC 61850 e o seu Impacto no Sistema de Automacao de Subestacoes," XII ERIAC, 2007.
- [76] H. Zhihong, Z. Limin y Z. Zhixue, "Research on Graphics Model Design for IEC61850 SCL Visual Configuration," Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on, 2008, pp. 700-704.
- [77] IEC-61850-9-1, "Communication networks and systems in substations," *Part 9-1: Specific Communication Service Mapping (SCSM) – Sampled values over serial unidirectional multidrop point to point link*, 2003.
- [78] IEC-61850-9-2, "Communication networks and systems in substations," *Part 9-2: Specific Communication Service Mapping (SCSM) – Sampled values over ISO/IEC 8802-3*, 2004.
- [79] F. Calero, D. Dolezilek y E. Palma, "Conceptos IEC 61850 para el Ingeniero de Protección," XII ERIAC, 2007.
- [80] KalkiManager. "Software SCL," www.kalkitech.com.
- [81] SIEMENS-SIPROTEC-Compact7SJ80.
<http://www.energy.siemens.com/fi/en/automation/power-transmission-distribution/protection/overcurrent-relays/siprotec-compact-7sj80.htm>.
- [82] ABB-RED670.
<http://www.abb.com/cawp/seitp202/2cd32f1c6063de47c12570ce0040e1af.aspx>.
- [83] ABB-REF615.
<http://www.abb.com/product/db0003db004281/c12573e700330419c12573cc00431e30.aspx>.
- [84] GE-350. <http://www.gedigitalenergy.com/multilin/catalog/350.htm#com>.
- [85] C. R. Ozansoy, A. Zayegh y A. Kalam, "The Application-View Model of the International Standard IEC 61850," Power Delivery, IEEE Transactions on, vol. 24, no. 3, pp. 1132-1139, 2009.

ANEXOS

En esta sección del trabajo de tesis se adjuntan los anexos.

ANEXO A: SERVIDORES DE BASES DE DATOS, PÁGINAS WEB Y LENGUAJES ASOCIADOS

A.1. Introducción

En este anexo se realizará una descripción detallada del proceso realizado en el Capítulo 4 de este trabajo de tesis, el cual consiste de un un SCADA que trabaja con el protocolo Modbus TCP, para los equipos instalados en la subestación IPSEP de la UNRC.

Se utilizará un paquete denominado XAMPP [6], el cuál tiene un servidor APACHE para paginas web que soporta PHP, lenguaje que se utilizará para mostrar los datos en la página WEB a partir de una base de datos MySQL contenida en un servidor que también está incluido en el paquete. En el servidor MySQL, el usuario por defecto es *root* y la contraseña <vacía>; el servicio trabaja en el puerto 3306. La página diseñada debe ser alojada en la carpeta *c:/XAMPP/htdocs*.

A.2. Programa en Labview

En primero lugar, como se dispone de una herramienta que se conecte directamente desde Labview a la base de datos MySQL, se utilizarán conectores ODBC.

Se instala el paquete “*Database Connectivity Toolkit*” de Labview, el cual proporciona herramientas para trabajar con bases de datos.

Luego, se instalan los conectores MySQL ODBC, que se descargan desde la página web de MySQL [7].

En este punto se crea un conector ODBC en Microsoft Windows XP y se asociará el mismo a una base de datos MySQL, la instalación del motor de base de datos no se explica en este trabajo.

Se crea el conector ODBC desde *Inicio>Panel de control>Herramientas administrativas>Origenes de datos (ODBC)*, en donde al hacer doble click se selecciona la pestaña *DSN del usuario* y luego *agregar*, seguido a esto se elige la opción *MySQL ODBC 5.1 Driver*. En la Figura A-1 se muestra ventana que emerge si se siguen esos pasos, se configurarán los siguientes parámetros:

Data Source Name: Representa un identificador de la base de datos que se utilizará en Labview para almacenar los datos.

Description: es para colocar una descripción, no es relevante.

Server: aquí se coloca la dirección IP del servidor MySQL, como esta en esta misma computadora se puso localhost, se podría utilizar 127.0.0.1, 3306 es el puerto TCP de comunicaciones al cuál nos debemos conectar a dicho servidor.

User/Password: aquí se coloca el nombre de usuario y la contraseña para acceder al servidor de base de datos MySQL.

Database: se coloca el nombre de la base de datos a utilizar, ya que dentro del servidor podemos tener más de una.

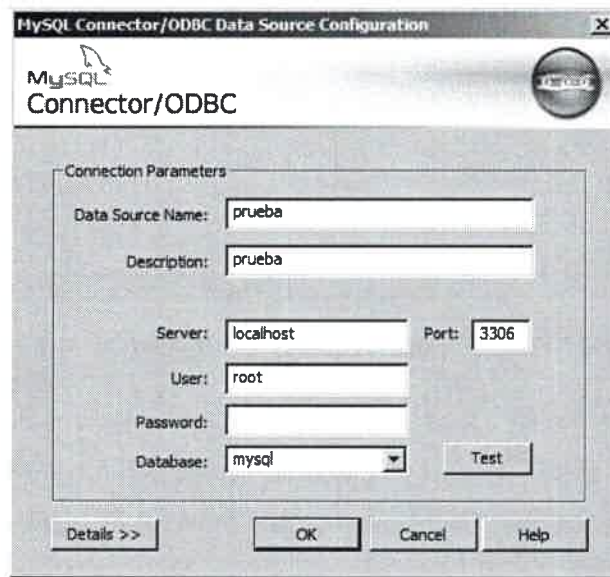


Figura A-1 - Conector ODBC de MySQL

Una vez que esta todo configurado para utilizar la base de datos, se descarga de la página de National Instruments la librería de Modbus TCP para labview desde este enlace [8].

Al iniciar Labview 8.6 con el paquete de Modbus TCP, y el de bases de datos instalado, se procede a iniciar el proceso de creación del programa.

En primer lugar, se selecciona la opción "Blank VI", como se muestra en la Figura A-2.

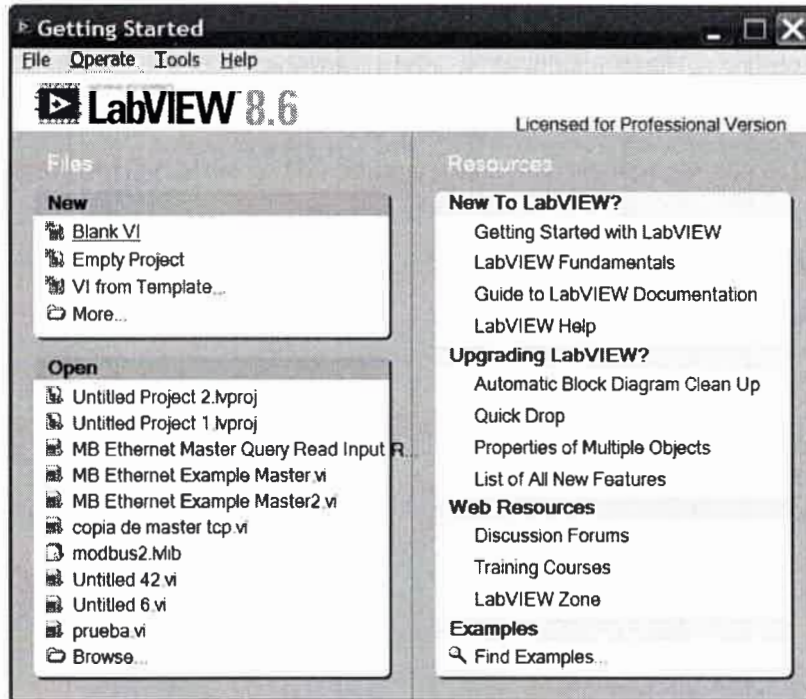


Figura A-2 - Pantalla para comenzar el proyecto en Labview

En el “Block Diagram” se selecciona *view>Functions Palette*. El bloque que se utiliza se encuentra en *User Libraries>NI Modbus>mastr MB* (el que esta más abajo, al posicionar el mouse por encima del mismo dice *MB Ethernet Example Master.vi*, ya que el otro es para Modbus serie), esto es un cliente Modbus TCP, que permite la comunicación nuestro servidor Modbus TCP en el PowerLogic 4250.

Se arrastra el bloque “mastr MB” al “Block Diagram” y se hace doble click, de nuevo, el mismo tendrá “Block Diagram” y “Front End” (si solo aparece uno con ctrl+E aparece el otro). Desde el “Block Diagram” se tiene un “switch case”, tal como se muestra en la Figura A-3, donde el cliente utiliza todos los métodos de Modbus, los 3 casos que contempla el switch son:

- 0: Se ejecuta cada 100 ms, chequea que no haya un cambio en alguna variable.
- 1 y 2: cuando desde el FrontEnd se setea un valor.

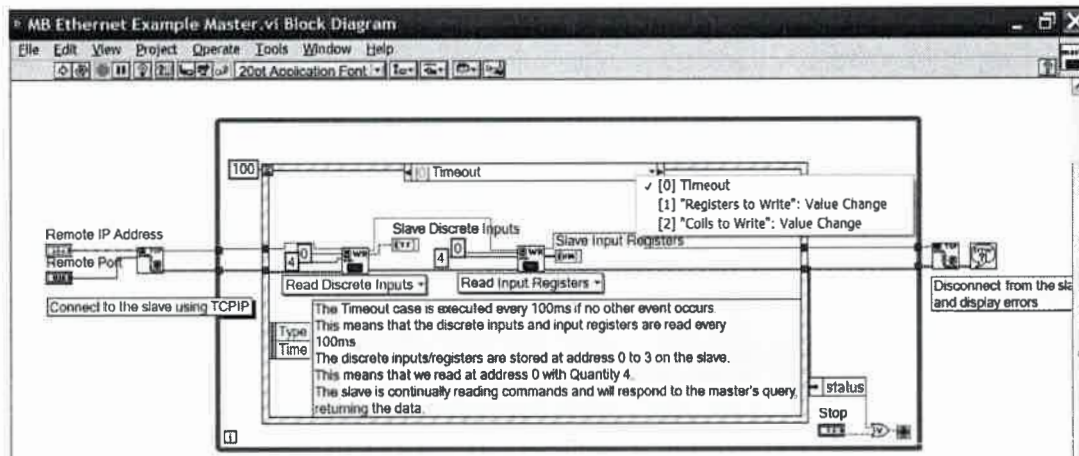


Figura A-3 - Vista del Block Diagram del bloque Modbus TCP

Simplificando el bloque, de manera de utilizar solo lo que sea necesario. Los casos 1 y 2 se dejan ya que si no se toca nada desde el FrontEnd el programa no hace nada, se quita del caso 0 la parte de Read Discrete Inputs, con lo cual nos queda lo que se muestra en la Figura A-4.

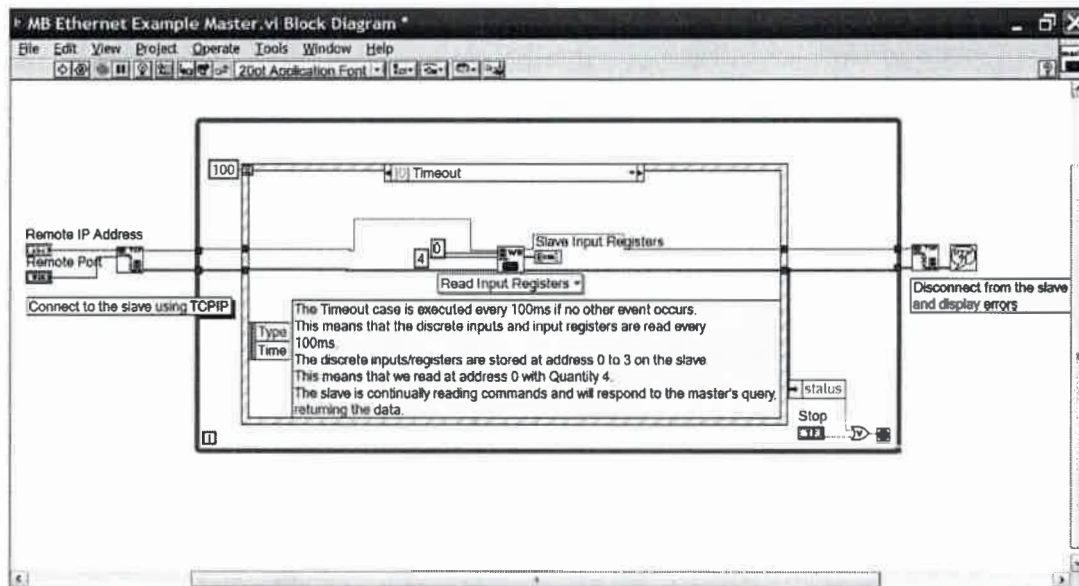


Figura A-4 - Vista del Block Diagram del bloque Modbus/TCP modificado

Si se intenta hacer funcionar el programa por defecto el mismo arroja un error en los datos de Modbus, ya que hay un parámetro que está por defecto en la librería de LabView que no es compatible con el CM4250.

Esto se puede detectar a través de captura de paquetes en la red por medio del software "Wireshark" [9], comparando las respuestas del servidor al cliente Labview y a

otro denominado “SimpleModBus” [10], en la Figura A-5 y la Figura A-6 se muestran las respuestas al cliente Labview y al “SimpleModbus” respectivamente:

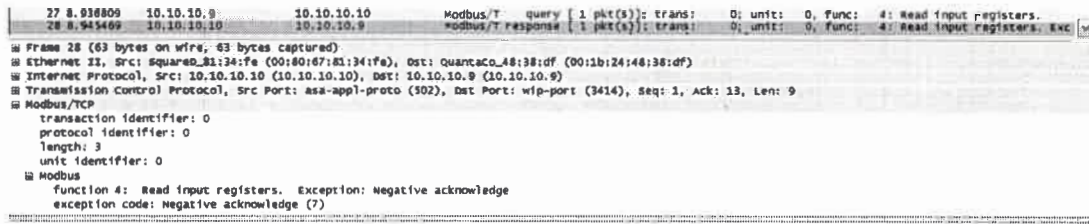


Figura A-5 - Respuesta del servidor al cliente Modbus/TCP de Labview

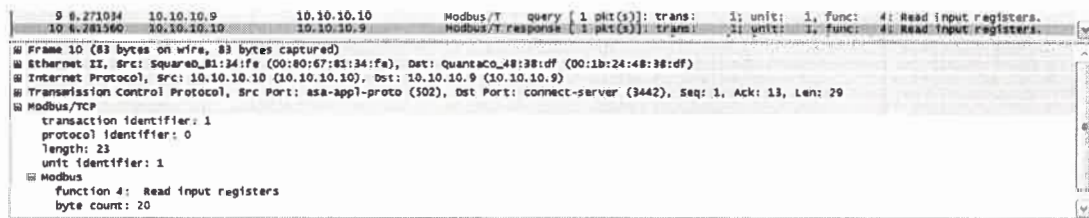


Figura A-6 - Respuesta del servidor al cliente Simply Modbus

De aquí se trabaja en la librería de LabView para cambiar el parámetro “unit identifier” que representa a un dispositivo en el servidor.

Para cambiar este parámetro se hace doble click en el bloque WR y lo se configura para que quede en 1 en “Unit ID” de la Figura A-7.

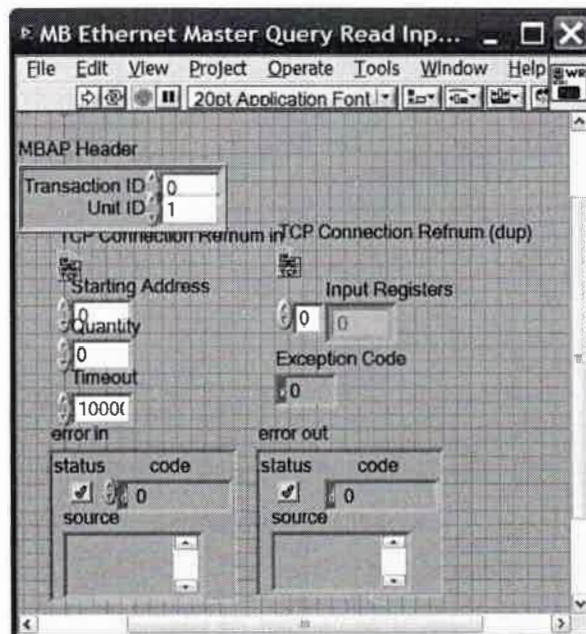


Figura A-7 - Cambio para configurar la ubicación servidor

Luego, desde el “BlockDiagram” se configura en lo mismo en “MBAP Header” para que el valor anterior quede como predeterminado, como se muestra en la Figura A-8.

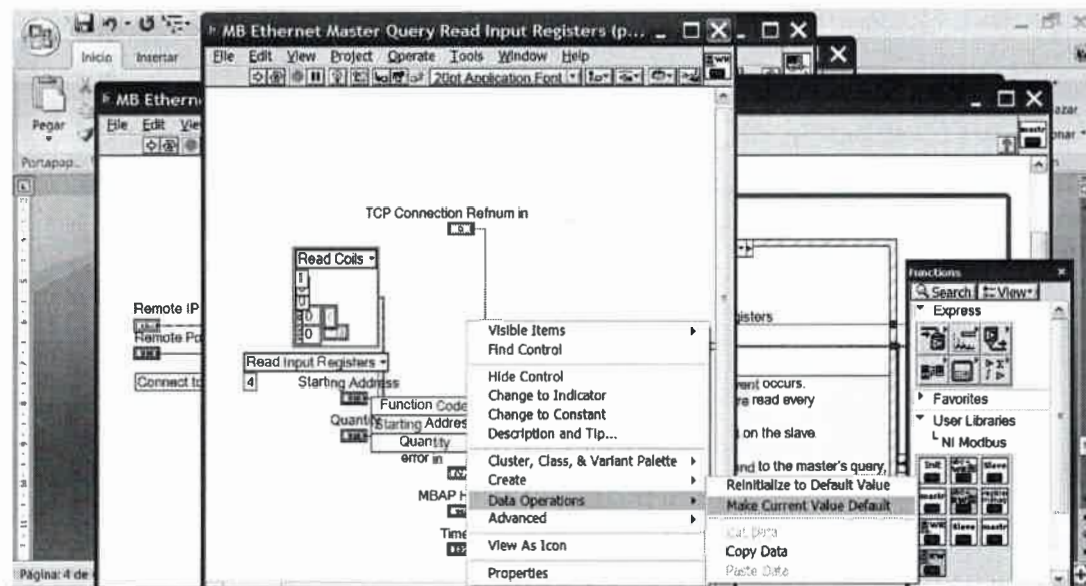


Figura A-8 - Captura de pantalla

En este punto el cliente Modbus TCP ya está configurado y se tiene acceso a los datos del servidor, ahora debemos configurar el cliente de base de datos para que se conecte a la base de datos MySQL a través del conector ODBC creado anteriormente.

Los bloques disponibles para manipular bases de datos se muestran en la Figura A-9, y los necesarios para nuestro programa son solo los tres primeros y dentro del quinto el que sirve para obtener la hora y fecha, se observan en la Figura A-10.

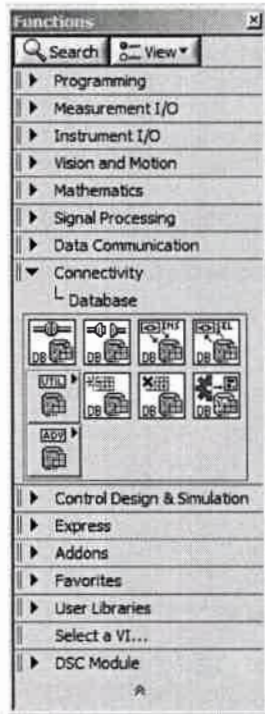


Figura A-9 - Bloques de Labview para utilizar Bases de Datos

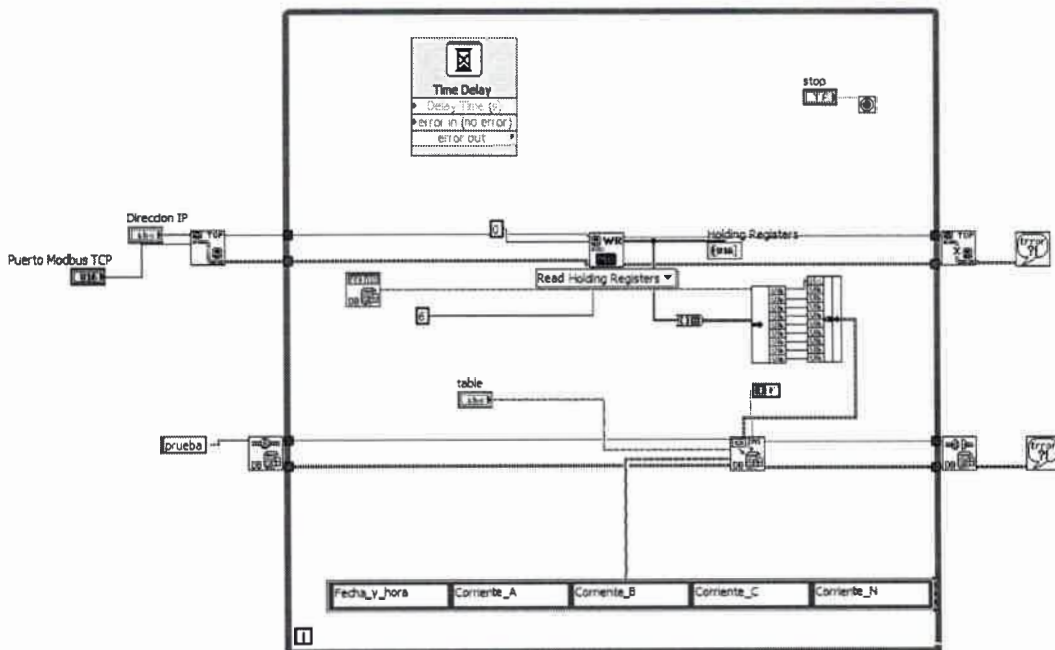


Figura A-10 - Diagrama Final del cliente en Labview

A.3. Resultados

Desde la página web del CM4250 podemos ver las lecturas instantáneas, esto se muestra en la Figura A-11.

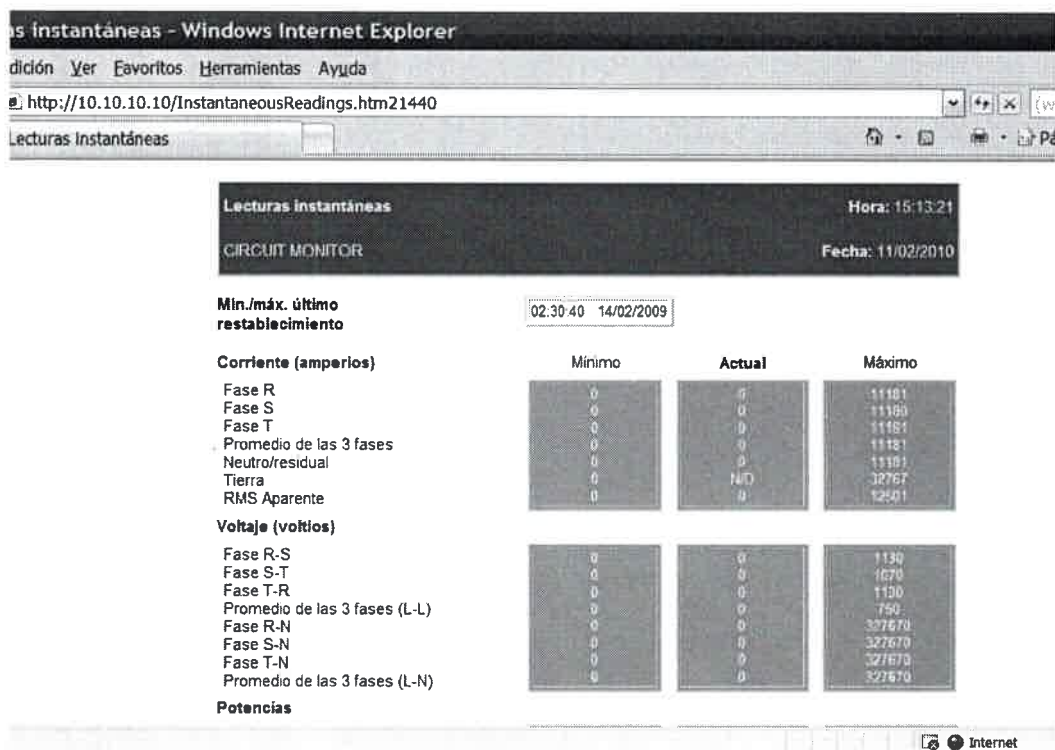


Figura A-11 - Datos de la página web embebida del equipo POWER LOGIC CM4250

Se extraen los valores máximos de las corrientes con nuestro cliente, para lo cual, se debe conocer cuales son los registros que se tienen que consultar. Para esto se utiliza manual del CM4250, en forma resumida, los registros a utilizar detallan en la Figura A-12.

Maximum—Current									
1500	Maximum Current, Phase A	1	Integer	RO	Y	A	Amperes/Scale	0 – 32,767	RMS
1501	Maximum Current, Phase B	1	Integer	RO	Y	A	Amperes/Scale	0 – 32,767	RMS
1502	Maximum Current, Phase C	1	Integer	RO	Y	A	Amperes/Scale	0 – 32,767	RMS
1503	Maximum Current, Neutral	1	Integer	RO	Y	B	Amperes/Scale	0 – 32,767 (-32,768 if N/A)	RMS 4-wire system only
1504	Maximum Current, Ground	1	Integer	RO	Y	C	Amperes/Scale	0 – 32,767 (-32,768 if N/A)	Maximum calculated RMS current
1505	Maximum Current, 3 Phase Average	1	Integer	RO	Y	A	Amperes/Scale	0 – 32,767	Maximum calculated mean A, B & C

Figura A-12 - Tabla de los registros utilizados en el POWER LOGIC CM4250

Seteando estos valores en nuestro cliente se obtienen los resultados que se exponen en la Figura A-13.

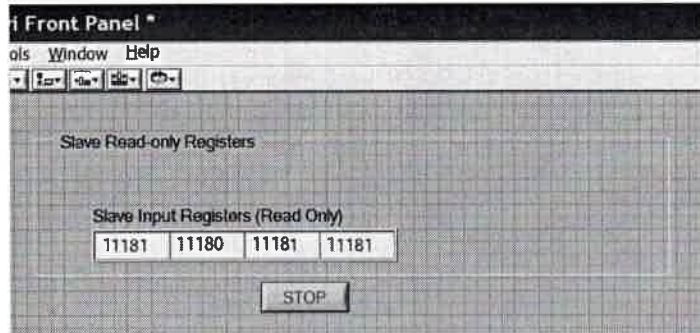


Figura A-13 - Datos obtenidos de la aplicación en Labview

Que coinciden con los valores arrojados en la página del equipo, esto es lo que se guarda en la base de datos y luego se puede visualizar por medio de la página Web que se diseñó.

ANEXO B: REGISTROS MODBUS DISPONIBLES EN EL POWER LOGIC CM4250 Y EN LOS SEPAM 1000+

B.1. Introducción

Se presenta un listado parcial de los registros disponibles en los equipos utilizados en el Capítulo 4 de esta tesis de maestría, los cuáles fueron extraídos de las hojas de datos de los mismos. El listado completo se puede encontrar en el manual de los equipos.

ABBREVIATED REGISTER LISTING

Table C-3 contains an abbreviated register list for the circuit monitor.

Table C-3: Abbreviated Register List

Reg	Name	Size	Type	Access	NV	Scale	Units	Range	Notes
100 ms Metering—Current									
1000	Current, Phase A	1	Integer	RO	N	A	Amperes/Scale	0 – 32,767	RMS
1001	Current, Phase B	1	Integer	RO	N	A	Amperes/Scale	0 – 32,767	RMS
1002	Current, Phase C	1	Integer	RO	N	A	Amperes/Scale	0 – 32,767	RMS
1003	Current, Neutral	1	Integer	RO	N	B	Amperes/Scale	0 – 32,767 (-32,768 if N/A)	RMS 4-wire system only
1004	Current, Ground	1	Integer	RO	N	C	Amperes/Scale	0 – 32,767 (-32,768 if N/A)	RMS 4-wire system only
1005	Current, 3-Phase Average	1	Integer	RO	N	A	Amperes/Scale	0 – 32,767	Calculated mean of Phases A, B & C
1006	Current, Apparent RMS	1	Integer	RO	N	A	Amperes/Scale	0 – 32,767	Peak instantaneous current of Phase A, B or C divided by $\sqrt{2}$
100 ms Metering—Voltage									
1020	Voltage, A-B	1	Integer	RO	N	D	Volts/Scale	0 – 32,767	Fundamental RMS Voltage measured between A & B
1021	Voltage, B-C	1	Integer	RO	N	D	Volts/Scale	0 – 32,767	Fundamental RMS Voltage measured between B & C
1022	Voltage, C-A	1	Integer	RO	N	D	Volts/Scale	0 – 32,767	Fundamental RMS Voltage measured between C & A
1023	Voltage, L-L Average	1	Integer	RO	N	D	Volts/Scale	0 – 32,767	Fundamental RMS 3 Phase Average L-L Voltage
1024	Voltage, A-N	1	Integer	RO	N	D	Volts/Scale	0 – 32,767 (-32,768 if N/A)	Fundamental RMS Voltage measured between A & N 4-wire system only
1025	Voltage, B-N	1	Integer	RO	N	D	Volts/Scale	0 – 32,767 (-32,768 if N/A)	Fundamental RMS Voltage measured between B & N 4-wire system only
1026	Voltage, C-N	1	Integer	RO	N	D	Volts/Scale	0 – 32,767 (-32,768 if N/A)	Fundamental RMS Voltage measured between C & N 4-wire system only
1027	Voltage, N-G	1	Integer	RO	N	E	Volts/Scale	0 – 32,767 (-32,768 if N/A)	Fundamental RMS Voltage measured between N & G 4-wire system with 4 element metering only
1028	Voltage, L-N Average	1	Integer	RO	N	D	Volts/Scale	0 – 32,767 (-32,768 if N/A)	Fundamental RMS 3-Phase Average L-N Voltage 4-wire system only
100 ms Metering—Power									
1040	Real Power, Phase A	1	Integer	RO	N	F	kW/Scale	-32,767 – 32,767 (-32,768 if N/A)	Real Power (PA) 4-wire system only
1041	Real Power, Phase B	1	Integer	RO	N	F	kW/Scale	-32,767 – 32,767 (-32,768 if N/A)	Real Power (PB) 4-wire system only
1042	Real Power, Phase C	1	Integer	RO	N	F	kW/Scale	-32,767 – 32,767 (-32,768 if N/A)	Real Power (PC) 4-wire system only
1043	Real Power, Total	1	Integer	RO	N	F	kW/Scale	-32,767 – 32,767	4-wire system = PA+PB+PC 3 wire system = 3-Phase real power
1044	Reactive Power, Phase A	1	Integer	RO	N	F	kVAr/Scale	-32,767 – 32,767 (-32,768 if N/A)	Reactive Power (QA) 4-wire system only
1045	Reactive Power, Phase B	1	Integer	RO	N	F	kVAr/Scale	-32,767 – 32,767 (-32,768 if N/A)	Reactive Power (QB) 4-wire system only
RO = Read only. R/CW = Read configure writeable if in a setup session. NV = Nonvolatile. ① See "How Power Factor is Stored in the Register" on page 178. ② See "How Date and Time Are Stored in Registers" on page 178.									

Status zone

The **status zone** is a table which contains the Sepam check-word, pre-assigned remote annunciation bits (TS), and logic inputs.

Status	Word address	Bit address	Access	Modbus function enabled	Format
Sepam check-word	100	1000	R	3/4 or 1, 2, 7	X
TS1-TS16	101	1010	R	3/4 or 1, 2	B
TS17-TS32	102	1020	R	3/4 or 1, 2	B
TS33-TS48	103	1030	R	3/4 or 1, 2	B
TS49-TS64	104	1040	R	3/4 or 1, 2	B
Logic inputs	105	1050	R	3/4 or 1, 2	B

Measurement zone
(S20, S23, T20, T23 and M20 types)

Measurements	Word address	Access	Modbus function enabled	Format	Unit
I1 phase current (gain x 1)	106	R	3/4	16NS	0.1 A
I2 phase current (gain x 1)	107	R	3/4	16NS	0.1 A
I3 phase current (gain x 1)	108	R	3/4	16NS	0.1 A
I0 residual current (gain x 1)	109	R	3/4	16NS	0.1 A
Im1 average phase current (x 1)	10A	R	3/4	16NS	0.1 A
Im2 average phase current (x 1)	10B	R	3/4	16NS	0.1 A
Im3 average phase current (x 1)	10C	R	3/4	16NS	0.1 A
I1 phase current (gain x 10)	10D	R	3/4	16NS	1 A
I2 phase current (gain x 10)	10E	R	3/4	16NS	1 A
I3 phase current (gain x 10)	10F	R	3/4	16NS	1 A
I0 residual current (gain x 10)	110F	R	3/4	16NS	1 A
IM1 average phase current (x10)	111	R	3/4	16NS	1 A
IM2 average phase current (x10)	112	R	3/4	16NS	1 A
IM3 average phase current (x10)	113	R	3/4	16NS	1 A
IM1 peak demand phase current	114	R	3/4	16NS	1 A
IM2 peak demand phase current	115	R	3/4	16NS	1 A
IM3 peak demand phase current	116	R	3/4	16NS	1 A
Reserved	117	R	3/4	-	-
Itrip1 tripping current	118	R	3/4	16NS	10 A
Itrip2 tripping current	119	R	3/4	16NS	10 A
Itrip3 tripping current	11A	R	3/4	16NS	10 A
Itrip0 tripping current	11B	R	3/4	16NS	1 A
Cumulative breaking current	11C	R	3/4	16NS	1 (kA) ²
Number of operations	11D	R	3/4	16NS	1
Operating time	11E	R	3/4	16NS	1 ms
Charging time	11F	R	3/4	16NS	1 sec
Reserved	120	R	3/4	-	-
Running hours counter	121	R	3/4	16NS	1 hr
Thermal capacity used	122	R	3/4	16NS	%
Operating time before overload tripping	123	R	3/4	16NS	1 min
Waiting time after overload tripping	124	R	3/4	16NS	1 min
Unbalance ratio	125	R	3/4	16NS	% lb
Starting time / overload	126	R	3/4	16NS	0.1 sec
Starting current overload	127	R	3/4	16NS	1 A
Start inhibit time delay	128	R	3/4	16NS	1 min
Number of starts allowed	129	R	3/4	16NS	1
Temperatures 1 to 8	12A/131	R	3/4	16S	1 °C
Reserved	132/1EF	Prohibited			

Note: Only the measurements related to the Sepam function are significant. The values of the others are zero.

**ANEXO C: HOJA DE DATOS DEL RELÉ SIEMENS SIPROTEC
COMPACT 7SJ80**

C.1. Introducción

Se presenta información detallada del relé seleccionado en el Capítulo 7.



SIPROTEC Compact 7SJ80 Multifunction Protection Relay



Fig. 5/56
SIPROTEC Compact 7SJ80
multifunction protection relay

Description

The SIPROTEC Compact 7SJ80 relays can be used for line/feeder protection of high and medium-voltage networks with grounded, low-resistance grounded, isolated or a compensated neutral point. The relays have all the required functions to be applied as a backup relay to a transformer differential relay.

The 7SJ80 features "flexible protection functions". 20 additional protection functions can be created by the user. For example, a rate of change of frequency function or a reverse power function can be created.

The relay provides circuit-breaker control, additional primary switching devices (grounding switches, transfer switches and isolating switches) can also be controlled from the relay. Automation or PLC logic functionality is also implemented in the relay. The integrated programmable logic (CFC) allows the user to add own functions, e.g. for the automation of switch-gear (including: interlocking, transfer and load shedding schemes). The user is also allowed to generate user-defined messages.

The communication module is independent from the protection. It can easily be exchanged or upgraded to future communication protocols.

Highlights

Removable current and voltage terminals provide the ideal solution for fast and secure replacement of relays.

Binary input thresholds and current taps are software settings. There is thus no need to ever open the relay to adapt the hardware configuration to a specific application.

The relay provides 9 programmable function keys that can be used to replace push-buttons, select switches and control switches.

The battery for event and fault recording memory can be exchanged from the front of the relay.

The relay is available with IEC 61850 for incredible cost savings in applications (e.g. transfer schemes with synch-check, bus interlocking and load shedding schemes).

This compact relay provides protection, control, metering and PLC logic functionality. Secure and easy to use one page matrix IO programming is now a standard feature.

The housing creates a sealed dust proof environment for the relay internal electronics. Heat build up is dissipated through the surface area of the steel enclosure. No dusty or corrosive air can be circulated over the electronic components. The relay thus will maintain its tested insulation characteristic standards per IEC, IEEE, even if deployed in harsh environment.

Function overview

Protection functions

- Time-overcurrent protection (50, 50N, 51, 51N)
- Directional time-overcurrent protection (67, 67N)
- Sensitive dir./non-dir. ground-fault detection (67Ns, 50Ns)
- Displacement voltage (64)
- High-impedance restricted ground fault (87N)
- Inrush restraint
- Undercurrent monitoring (37)
- Overload protection (49)
- Under-/overvoltage protection (27/59)
- Under-/overfrequency protection (81O/U)
- Breaker failure protection (50BF)
- Phase unbalance or negative-sequence protection (46)
- Phase-sequence monitoring (47)
- Synch-check (25)
- Auto-reclosure (79)
- Fault locator (21FL)
- Lockout (86)

Control functions/programmable logic

- Commands for the ctrl. of CB, disconnect switches (isolators/isolating switches)
- Control through keyboard, binary inputs, DIGSI 4 or SCADA system
- User-defined PLC logic with CFC (e.g. interlocking)

Monitoring functions

- Operational measured values V, I, f
- Energy metering values W_p, W_q
- Circuit-breaker wear monitoring
- Minimum and maximum values
- Trip circuit supervision (74TC)
- Fuse failure monitor
- 8 oscillographic fault records

Communication interfaces

- System/service interface
 - IEC 61850
 - IEC 60870-5-103
 - PROFIBUS-DP
 - DNP 3.0
 - MODBUS RTU
- Ethernet interface for DIGSI 4
- USB front interface for DIGSI 4

Hardware

- 4 current transformers
- 0/3 voltage transformers
- 3/7 binary inputs (thresholds configurable using software)
- 5/8 binary outputs (2 changeover/Form C contacts)
- 1 live-status contact
- Pluggable current and voltage terminals

U.N.R.C.
Biblioteca Central



71200

71200