



## Comunicación de sistemas eléctricos basados en la Norma IEC 61850: Desarrollo de caso usando sampled values en servicios Cliente-Servidor

### Communication of electrical systems in the IEC 61850 Standard: Development of cases using sampled values in Client-Server services

Fabián Andrés Acevedo-Cardozo <sup>1a</sup>, Laura Camila Calderón-Soto <sup>1b</sup>, Juan Camilo González-Castillo <sup>1c</sup>,  
Hermann Raúl Vargas-Torres <sup>1d</sup>

<sup>1</sup>Grupo de Investigación en Sistemas de Energía Eléctrica (Gisel), Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones, Universidad Industrial de Santander, Colombia.

Correo electrónico: <sup>a</sup> [fabian.acevedo@correo.uis.edu.co](mailto:fabian.acevedo@correo.uis.edu.co),

<sup>b</sup> [laura.calderon@correo.uis.edu.co](mailto:laura.calderon@correo.uis.edu.co), <sup>c</sup> [juan.gonzalez5@correo.uis.edu.co](mailto:juan.gonzalez5@correo.uis.edu.co),

<sup>d</sup> [hrvargas@uis.edu.co](mailto:hrvargas@uis.edu.co)

Recibido: 16 abril, 2018. Aceptado: 11 enero, 2019. Versión final: 13 febrero, 2019.

#### Resumen

Este artículo presenta una implementación del estándar internacional IEC 61850, diseñado para automatización, control y protección de subestaciones, usando servicios Cliente / Servidor, permitiendo consultar remotamente datos de variables. Con este propósito, se muestra el procedimiento realizado y se analiza la funcionalidad del protocolo de comunicaciones, desde el intercambio de datos y la información entre los dispositivos que componen la red propuesta. La realización de este trabajo se debe a la mayor penetración e influencia de la norma y cómo se debe implementar, entre otros aspectos.

**Palabras clave:** automatización de subestaciones; comunicación de subestaciones; GOOSE; IEC 61850; IED; Sampled Values; SCADA.

#### Abstract

This article presents an implementation of IEC 61850 international standard, designed for automation, control and protection of substations, using Client / Server services, allowing to remotely query variable data. For this purpose, the procedure performed is shown and the functionality of the communications protocol is analyzed, from the exchange of data and information between the devices that make up the proposed network. The realization of this work is due to the greater penetration and influence of the norm, how it should be implemented, among other aspects.

**Keywords:** IEC 61850; GOOSE; Sampled Values; automation of substations; communication of substations; IED; SCADA.



## 1. Introducción

Implementar sistemas de automatización de subestaciones (SAS) es una inversión de alto costo que tienen que asumir las compañías eléctricas. Uno de los principales objetivos de la automatización de subestaciones es la gestión eficiente del sistema de potencia mediante el control de la relación producción-consumo. Inicialmente, los protocolos de comunicación entre los dispositivos electrónicos que conforman el SAS son propios de cada fabricante, lo cual acarrea varios inconvenientes para dicho sistema. Uno de estos es la interoperabilidad de los dispositivos electrónicos inteligentes (IEDs), implicando a dispositivos de distintos fabricantes que cumplen las mismas funciones dentro del SAS no puedan operar dentro del mismo. Esto obliga a las compañías a tener que usar dispositivos de un único fabricante para cualquier reemplazo de los mismos. En algunos casos, se usan convertidores de protocolo costosos para garantizar la interoperabilidad de los IEDs [1]. Otro inconveniente, quizá el más importante, es la implementación de nuevas funciones de automatización debido a ampliaciones de la subestación, implicando un alto costo en el SAS.

Para hacer frente a estos problemas el EPRI investigaba la posibilidad de crear un estándar para sistemas de comunicación en subestaciones. Por otro lado, y de manera independiente, el comité técnico 57 de la IEC trabajaba en pro del mismo objetivo. Finalmente, en 1997 decidieron unir esfuerzos y el resultado de su trabajo fue el estándar IEC 61850 para las redes y sistemas de comunicación en subestaciones.

El estándar IEC 61850 permite mediante un modelo orientado a objetos, describir los dispositivos presentes en el SAS, sus funciones, interfaces de comunicación entre ellos para llevar a cabo las diversas tareas de automatización. En este documento se muestra cómo se modeló una subestación eléctrica siguiendo el estándar IEC 61850 para acceder a las variables eléctricas en los barrajes de un sistema de potencia (subestaciones) desde un PC (gestor del sistema) durante una simulación en estado transitorio del sistema de potencia. La simulación se realizó en ATP (Alternative Transient Program) y el estado de las variables se podrá consultar a través de la interfaz gráfica *clientGUI* de la librería *OpenIEC61850*.

## 2. Antecedentes y descripción general del protocolo

Los mecanismos de intercambio de información se basan primordialmente en modelos de información bien estructurados. Los modelos de información y los métodos de modelado son la principal característica del estándar IEC 61850. Este estándar emplea el enfoque representado en la figura 1 para modelar la información común encontrada en los dispositivos de las subestaciones. Toda la información disponible para intercambio entre dispositivos de automatización se define en el estándar. El modelo proporciona para el SAS una imagen del “mundo análogo” [2].

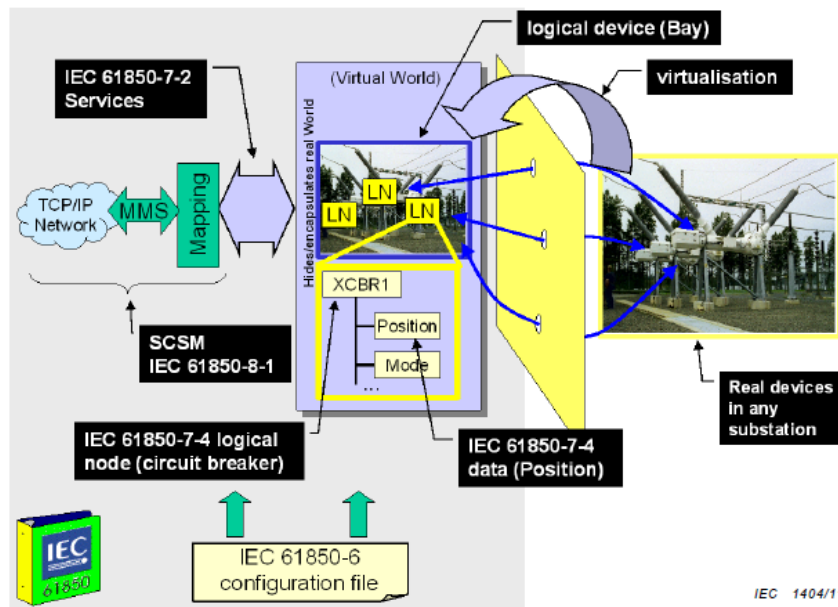


Figura 1. Enfoque conceptual de los modelos de información [3].

El estándar emplea el concepto de virtualización, el cual proporciona una visión de los aspectos relevantes de un dispositivo real (ej. interruptor) que son de interés para el inter-cambio de información entre los IEDs.

Como se describe en la norma IEC 61850, el enfoque del estándar es descomponer las funciones de la aplicación en entidades más pequeñas, las cuales se emplean para el intercambio de la información, esas entidades se denominan nodos lógicos (LN). Por ejemplo, una representación virtual de un interruptor se describe por el

nodo lógico XCBR (ver figura 1). Una agrupación de varios nodos lógicos constituye un dispositivo lógico (LD) el cual podría representar una unidad de bahía.

De esta manera se pueden crear dispositivos lógicos (ver figura 2). Dicha figura muestra el modelo de un dispositivo físico, representado mediante un dispositivo lógico compuesto de varios nodos lógicos. El dispositivo lógico estará alojado en un dispositivo físico el cual tiene una dirección dentro de la red de comunicaciones de la subestación.

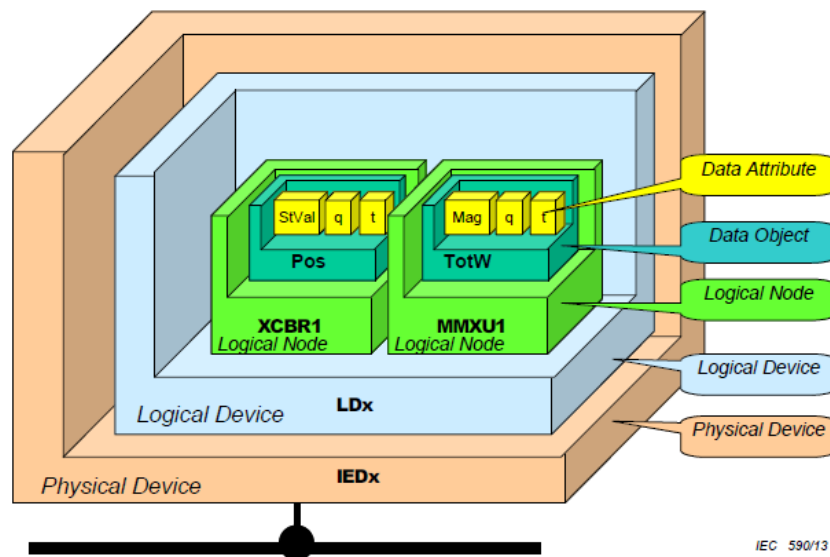


Figura 2. Esquema de un dispositivo lógico [4].

Un dispositivo lógico siempre se implementa en un dispositivo físico (IED), por lo tanto, estos no son distribuidos.

De acuerdo con su funcionalidad, un nodo lógico contiene una lista de datos, y cada dato, una lista de atributos dedicados. Por ejemplo, el nodo lógico XCBR contiene el dato “Pos”, el cual representa la posición del interruptor, y este último contiene el atributo “stVal”, el cual indica el valor actual de la posición del interruptor (abierto, cerrado, estado de transición o, defectuoso).

Los datos tienen una estructura y una semántica bien definida (dentro del contexto de un SAS), la información representada por los datos y sus atributos se intercambia mediante servicios de acuerdo con reglas bien definidas y con criterios de desempeño que se describen en la sección 5 de la norma IEC 61850. Los servicios se implementan por un medio de comunicación específico y

concreto (SCSM, por ejemplo, usando MMS, TCP/IP y Ethernet entre otros).

Los nodos lógicos y los datos contenidos en un dispositivo lógico, son cruciales en la descripción y el intercambio de la información del SAS y en la interoperabilidad entre los diversos IEDs.

Los dispositivos lógicos, los nodos lógicos y los datos que contienen se deben configurar. La razón principal de la configuración es seleccionar los nodos lógicos apropiados y sus datos (los cuales se encuentran normalizados) y asignar los valores específicos en las distintas instancias, por ejemplo, referencias concretas entre instancias de los nodos lógicos (sus datos), los mecanismos de intercambio, y los valores iniciales para los datos del proceso.

El estándar IEC 61850 especifica un formato de archivo para describir las configuraciones de comunicación entre los IEDs y los parámetros de los mismos, las

configuraciones del sistema de comunicación, las funciones desempeñadas por los equipos primarios (de patio), y las relaciones entre ellos.

El lenguaje definido se denomina “*Lenguaje de descripción de la Configuración de Subestaciones*” (SCL, por sus siglas en inglés) y se basa en XML (Extensible Markup Language version 1.0). [5][6] proporcionan una descripción de SCL.

### 3. Caso de estudio

#### 3.1 Modelado de la subestación

El diagrama unifilar del sistema de potencia propuesto se muestra en la figura 3. Está compuesto de cuatro barras, dos de ellas conectadas a generadores. Las otras dos barras, de carga (PQ), tienen conectada a cada una de ellas una carga estática. De las cuatro subestaciones presentes en el sistema, se modeló únicamente la que está conectada a uno de los generadores, y su diagrama unifilar (ver figura 4).

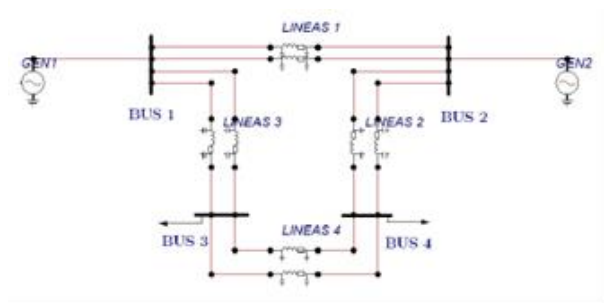


Figura 3. Sistema de potencia propuesto. Fuente: elaboración propia.

Como se mencionó anteriormente los archivos SCL se describen mediante un esquema XML, proporciona un diagrama UML de la estructura del esquema SCL (ver figura 5).

Los elementos `tSubstation`, `tCommunication` y `tIED` describen los modelos de la subestación, IEDs y el protocolo de comunicaciones. El elemento `tHeader` sirve para identificar un archivo de configuración SCL, su versión, y para especificar las opciones para la asignación de nombres a las señales. El elemento `tDataTypeTemplates` describe todos los tipos de nodos lógicos instanciables creados por el usuario en forma de plantillas. Cuando todos los elementos están totalmente descritos, todo el diseño de SAS está completo.

Para el modelado de la subestación propuesta se utilizó la versión de evaluación de la herramienta Visual SCL de la empresa ASE Systems Inc. [7].

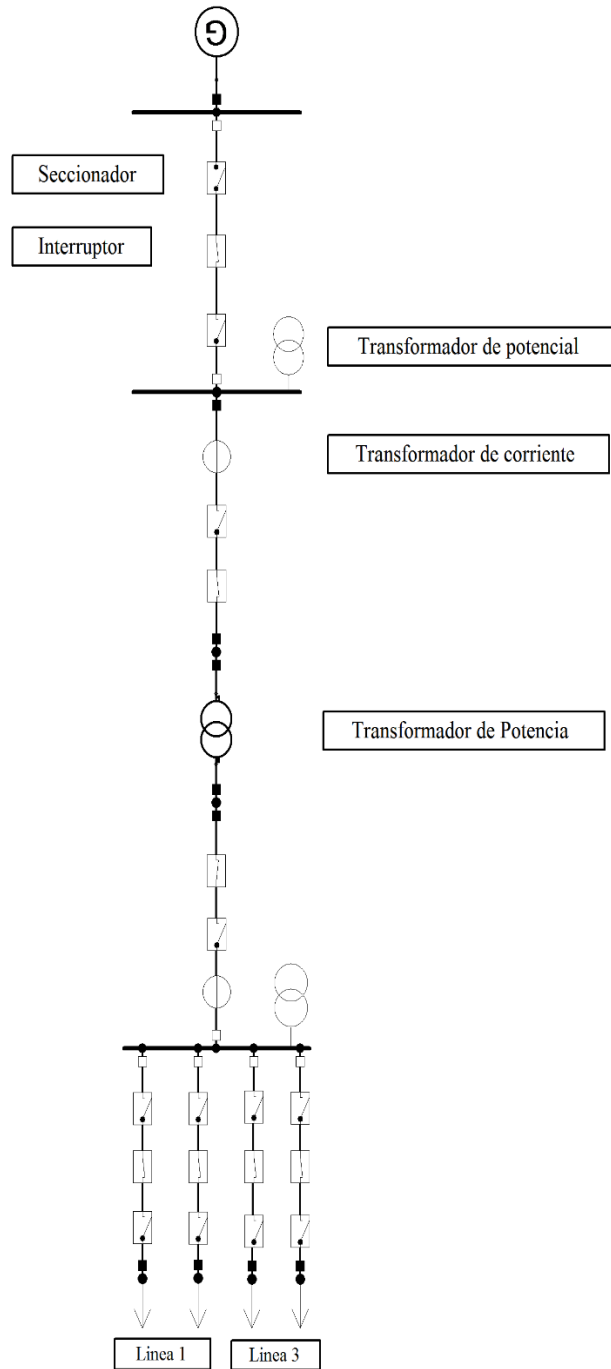


Figura 4. Subestación propuesta. Fuente: elaboración propia.

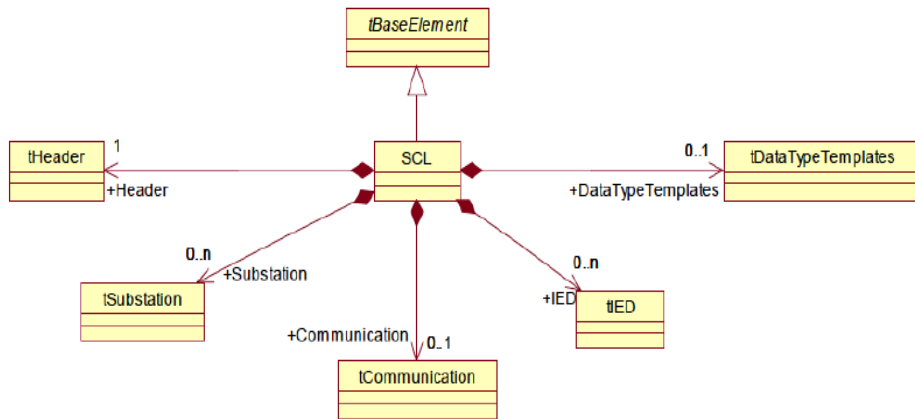


Figura 5. Diagrama UML de la estructura del esquema SCL. Fuente: elaboración propia.

SCL propone los siguientes tipos de archivos (con sus respectivas extensiones) para la descripción del SAS:

- System Specification Description (\*.SSD file extension).
- Substation Configuration Description (\*.SCD file extension).
- IED Capability Description (\*.ICD file extension).
- Configured IED Files (\*.CID file extension).

Cada uno de ellos fue desarrollado por la IEC con base en las distintas maneras y etapas del diseño de un sistema de automatización de subestaciones. Para el desarrollo de este caso se requiere el archivo de extensión \*.ICD, el cual describe las capacidades (funciones) de un IED. Este archivo se carga en los IEDs, los cuales se encargan de interpretarlos y así quedar “programados” para desempeñar las diversas tareas del SAS.

La figura 6 muestra el resultado final del elemento Substation. En ella se pueden apreciar, fuera de los cuadros azules, los nodos lógicos ubicados a nivel de bahía. Los nodos lógicos ubicados en los equipos primarios se observan a la derecha de los mismos.

El tercer elemento del modelado corresponde a los IEDs (elemento IED), en este caso un PC actúa como IED de una subestación de generación, por lo tanto, existe un solo dispositivo físico en la misma. Dentro de cada dispositivo físico se deben establecer los dispositivos lógicos, encargados de desempeñar las tareas de automatización y control.

Para la subestación de generación se crearon ocho dispositivos lógicos (ver figura 7).

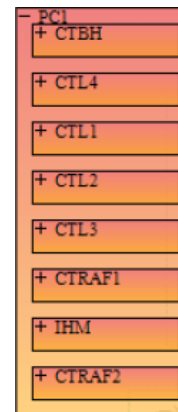


Figura 6. Dispositivos lógicos modelados. Fuente: elaboración propia.

Estos dispositivos cumplen las siguientes funciones:

- CTBH: controlador de bahía de la línea proveniente desde el generador. Se encarga de alojar los distintos relés para la protección de la línea desde ese lado de la subestación, además alberga los nodos lógicos de los equipos de corte y seccionamiento. Tiene los nodos lógicos CILO, CSWI, XSWI y XCBR.
- CTRAF1: controlador del primario del transformador. Se encarga de alojar la protección de sobrecorriente del devanado primario, de controlar la posición de los equipos de corte y seccionamiento del primario del transformador, también alberga los nodos lógicos de los equipos primarios (transformadores de instrumentación, de potencia y equipos de corte y seccionamiento).
- Tiene los nodos lógicos CILO, CSWI, IHMI, MMXU, PIOC, PTRC, TCTR, TVTR, XSWI, XCBR y YPTR.

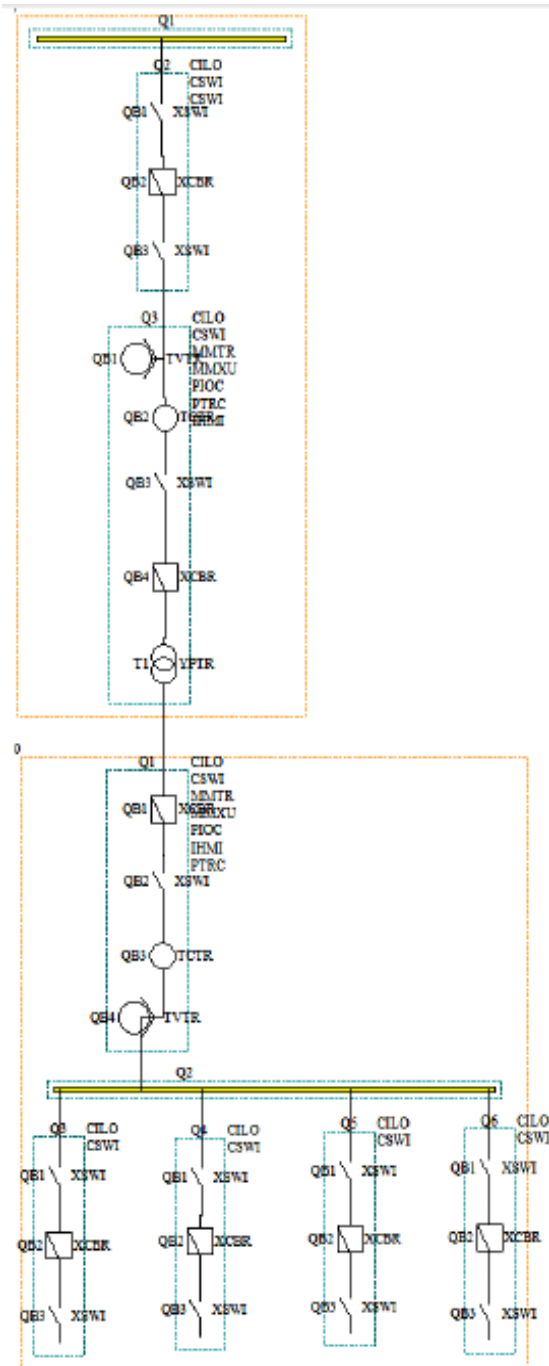


Figura 7. Descripción del elemento Substation. Fuente: elaboración propia.

- d) del secundario del transformador. Se encarga de alojar la protección de sobrecorriente del devanado secundario, de controlar la posición de los equipos de corte y seccionamiento del secundario del transformador, también alberga los nodos lógicos de los equipos primarios (transformadores de

instrumentación y equipos de corte y seccionamiento). Tiene los nodos lógicos CILO, CSWI, IHMI MMXU, PIOC, PTRC, TCTR, TVTR, XSWI y XCBR.

- e) CTL1: controlador de bahía de línea. Se encarga de alojar los distintos relés para la protección de la primera de las cuatro líneas, desde ese lado de la subestación; además alberga los nodos lógicos de los equipos de corte y seccionamiento. Tiene los nodos lógicos CILO, CSWI, XSWI y XCBR. Los dispositivos lógicos CTL2, CTL3 y CTL4 son idénticos al CTL1.
- f) IHM: representa la interfaz hombre máquina, contiene el nodo lógico IHMI y se localiza a nivel de bahía y de estación o donde se requiera una interfaz hombre máquina; para el caso particular de la subestación se ubicó un solo IHMI a nivel de bahía dentro de cada nivel de tensión.

Para el elemento Communication se estableció un único Access Point (S1) para toda la subestación, al igual que una sola subred (W01).

El siguiente elemento corresponde a la plantilla de datos (DataTypeTemplates). Allí se describieron todos los tipos de nodos lógicos instanciables que se usaron para el modelado del sistema. Esta descripción se basa en la elección de los nodos lógicos, los datos y los atributos de datos que los conforman.

### 3.2 Simulación del sistema propuesto

Con el objetivo de obtener valores de proceso de la subestación y transmitirlos a través de la red se plantearon dos simulaciones, una en estado estable y otra en estado transitorio; la primera para obtener los valores de estado estable del sistema de potencia debido al flujo de carga, y la segunda para analizar el transitorio electromagnético debido a un evento ocurrido dentro del sistema de potencia.

La simulación en estado estacionario se llevó a cabo utilizando la herramienta computacional PowerWorld Simulator.

El sistema eléctrico de potencia (ver figura 3), consta de cuatro barras, dos de ellas destinadas a generación y las dos restantes destinadas a carga. El sistema agrupa ocho líneas de transmisión a nivel de 230 kV, el grupo de generación está conformado por máquinas síncronas a 13,8 kV y un transformador encargado de elevar dicha tensión al nivel de las líneas de transmisión.

Los parámetros de las líneas de transmisión se tomaron de Power System Stability And Control, McGraw-Hill [8] y corresponden al nivel de tensión de 230 kV (ver figura 8).

Los parámetros utilizados para el resto de elementos son

los que se muestran a continuación, (ver figuras 9, 10, 11 y 12).

De la simulación en estado estacionario del sistema propuesto se obtuvieron los datos relacionados en la figura 12.

[MVA] Base	150	R1 [Ω/km]	0,05	X1 [Ω/km]	0,488	
Z Base [Ω]	352,6667	Tensión	Longitud	Impedancia por circuito		
Líneas	[kV]	[km]	R1 [Ω]	X1 [Ω]	R1 [pu]	X1 [pu]
BUS1 - BUS2	230	15	0,75	7,32	0,002127	0,020756
BUS1 - BUS3	230	50	2,50	24,40	0,007089	0,069187
BUS2 - BUS4	230	40	2,00	19,52	0,005671	0,055350
BUS3 - BUS4	230	35	1,75	17,08	0,004962	0,048431

Figura 8. Parámetros para líneas de transmisión utilizados en Power World. Fuente: elaboración propia.

[MVA] Base	150						
Transformador	Capacidad [MVA]	Tensión H [kV]	Tensión L [kV]	Conexión H	Conexión L	Z1 [%]	Z1 [pu]
TRAF01	150	230	13,8	Y - gnd	Y - gnd	6	0,06
TRAF02	150	230	13,8	Y - gnd	Y - gnd	6	0,06

Figura 9. Parámetros para transformadores utilizados en Power World. Fuente: elaboración propia.

[MVA] Base	150						
Generador	Capacidad [MVA]	Tensión [kV]	max [MW]	max [MVA <sub>r</sub> ]	min [MVA <sub>r</sub> ]	X1 [%]	X1 [pu]
GEN1	140	13,8	133	73,75	43,71	9,9148	0,106250
GEN2	140	13,8	133	73,75	43,71	9,9148	0,106250

Figura 10. Parámetros para generadores utilizados en Power World. Fuente: elaboración propia.

Cargas	Carga [MVA]	Factor de Potencia	Activa [MW]	Reactiva [MVA <sub>r</sub> ]	Tensión [kV]
BUS3	120	0,85	102	63,214	230
BUS4	120	0,85	102	63,214	230

Figura 11. Parámetros para cargas utilizados en Power World. Fuente: elaboración propia.

Barra	Tensión [pu]	Tensión [kV]	ang [deg]	Carga [MW]	Carga [Mvar]	Gen [MW]	Gen [Mvar]
BUS1	0,972	223,545	-2,400				
BUS2	0,972	223,463	-2,420				
BUS3	0,955	219,699	-3,660	102,000	63,210		
BUS4	0,956	219,908	-3,590	102,000	63,210		
GEN1	1,000	13,800	0,000			101,670	72,290
GEN2	1,000	13,800	0,010			103,000	73,240

Figura 12. Resultados del flujo de carga con Power World. Fuente: elaboración propia.

Falla en la barra 'BUS3'	
Corriente [pu]	10,0880
Corriente [KA]	3,7980
Ang	-85,2300

Figura 13. Resultados de la falla simulada para la barra 3. Fuente: elaboración propia.

Para la simulación en estado transitorio [9] se utilizó la herramienta computacional ATPDraw (Alternative Transient Program)[10] con el objetivo de analizar los fenómenos del sistema de potencia bajo distintos eventos, además de proporcionar los datos de proceso que fueron transmitidos entre las computadoras implementando el protocolo IEC 61850. Los datos de proceso escogidos

para la transmisión fueron las tensiones y corrientes en las barras del sistema y las posiciones de los equipos de conmutación, ya que son servicios soportados por la librería.

El esquema de la simulación en estado transitorio, empleando ATPDraw (ver figura 14).

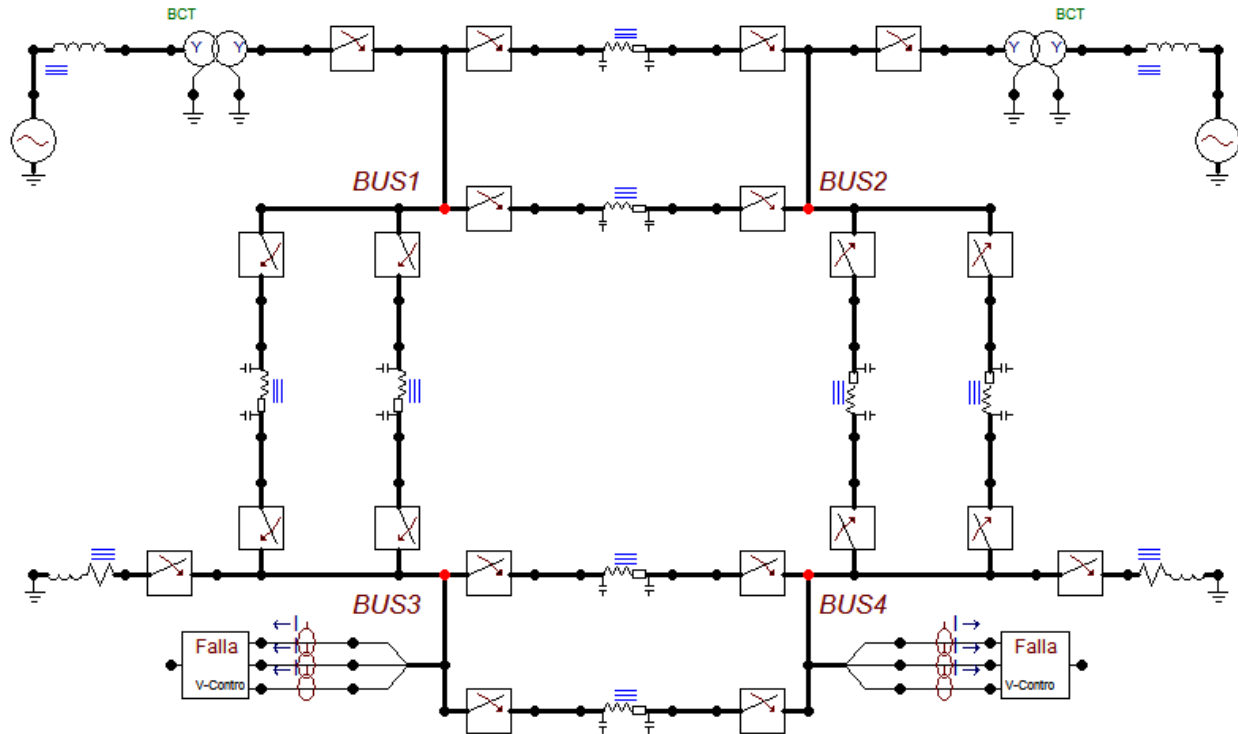


Figura 14. Esquema de la simulación en estado transitorio. Fuente: elaboración propia.

El modelo utilizado para el transformador de potencia fue el BCTTRAN, conectado como una bancada de transformadores monofásicos. Este se escogió debido a la similitud con los parámetros del modelo del transformador de PowerWorld.

Los parámetros requeridos por los modelos de líneas de transmisión son los mismos en ambos programas. En ATP se utilizó el modelo LINEPI\_3.

Los valores de estado estable obtenidos a partir de la simulación del flujo de carga en PowerWorld, fueron utilizados como valores iniciales para la simulación en estado transitorio y se emplearon para calcular los valores de las impedancias de las cargas justo en el instante anterior a la falla.

Las fallas trifásicas sólidas fueron realizadas en las barras de carga. Todas las fallas fueron realizadas a los 0,1 segundos con respecto al inicio de la simulación, algunos de los resultados para la falla en la barra 3 (ver figuras 15 y 16).

Los valores de posfalla obtenidos a partir de PowerWorld fueron contrastados con los obtenidos a través de ATP, y difieren en un error menor al 5% para las tensiones, como para las corrientes.

Las figuras 17 y 18 muestran la corriente y la tensión en la carga 2 debido a una falla en la barra 3 despejada a los 0,1 segundos, (200 milisegundos después de iniciar la simulación).



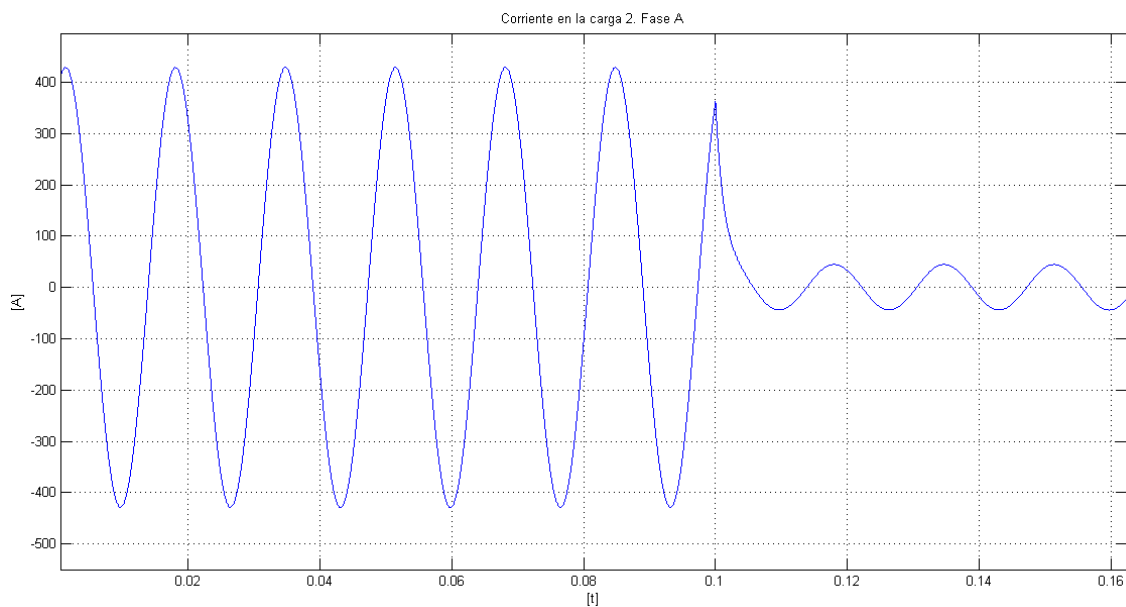


Figura 15. Corriente en la carga 2 debido a la falla en la barra 3. Fuente: elaboración propia.

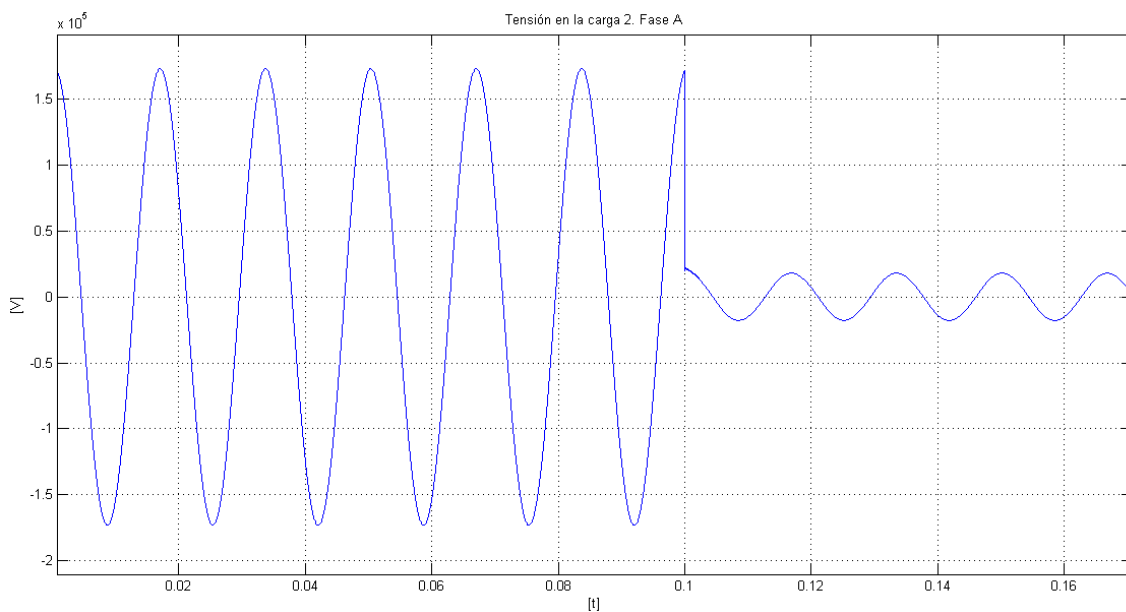


Figura 16. Tensión en la carga 2 debido a la falla en la barra 3. Fuente: elaboración propia.

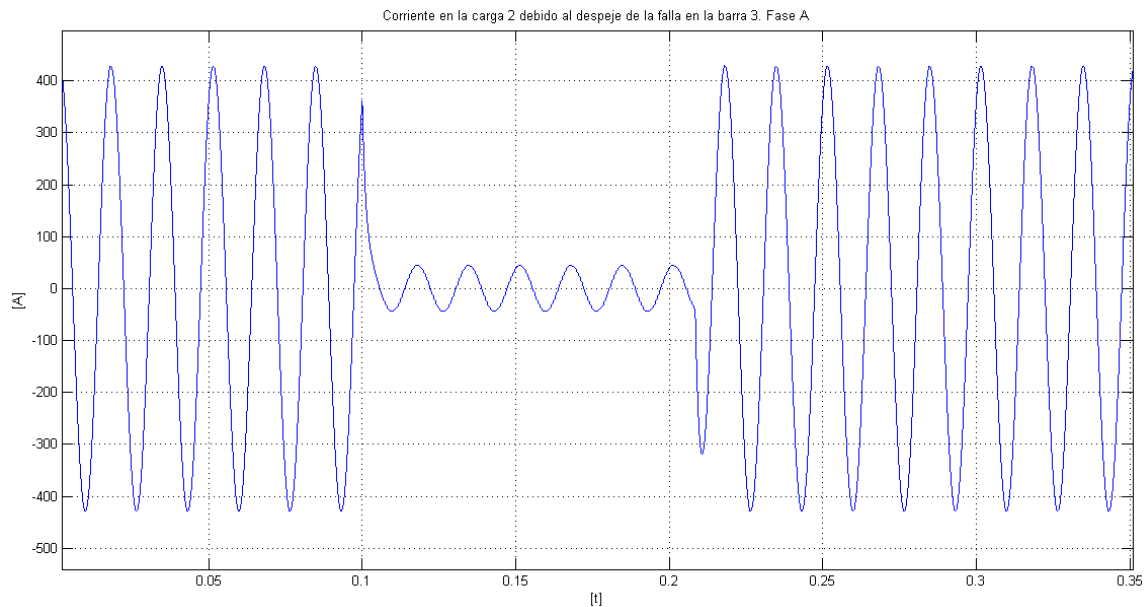


Figura 17. Corriente en la carga 2 debido al despeje de la falla en la barra 3. Fase A. Fuente: elaboración propia.

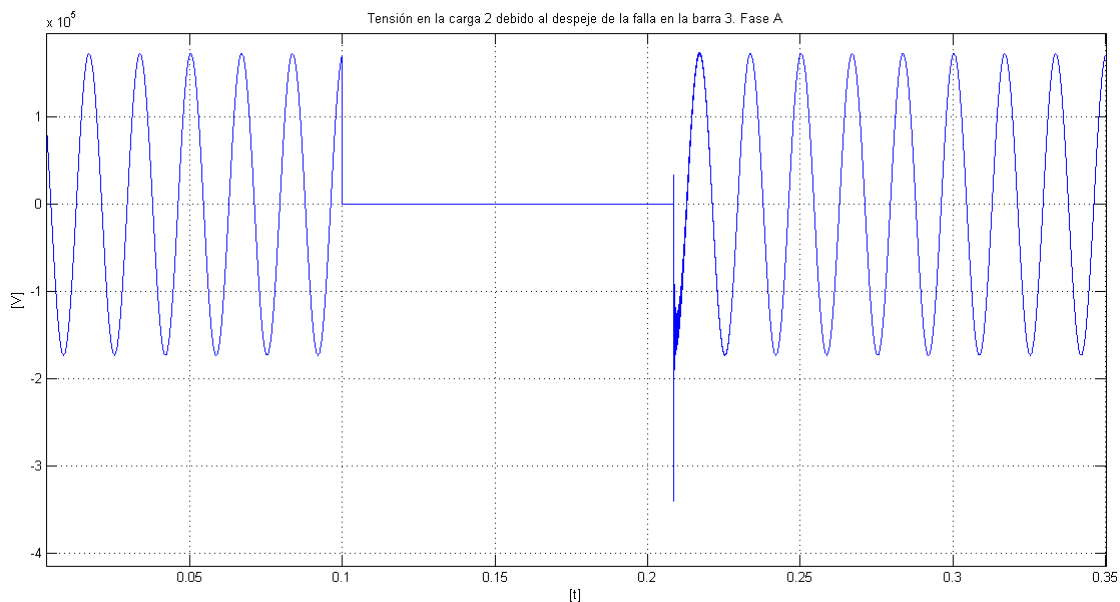


Figura 18. Tensión en la carga 2 debido al despeje de la falla en la barra 3. Fase A. Fuente: elaboración propia.

### 3.3 Implementación

Para llevar a cabo la transmisión de los datos entre computadoras con el protocolo IEC 61850 se utilizó la librería OpenIEC61850 [11] [12], la cual es una implementación de código abierto del estándar IEC 61850 bajo la licencia LGPL. La librería está escrita en Java y se compone de un cliente y el servidor MMS.

Los servicios ACSI que soporta son los siguientes:

- Servicios de modelo de asociación
- Todos los servicios GetDirectory y GetDataDefinition
- GetDataValues y SetDataValues
- Los modelos servicios para DATA-SETS
- Servicios Report-Control-Block (sólo cliente)

f) Control

El desarrollo de OpenIEC61850 fue iniciado por los institutos Fraunhofer ISE y OFFIS, y por la empresa Energy & Meteo Systems GmbH, como parte del proyecto de investigación eTelligence financiado por Ministerio Federal Alemán de Economía y Tecnología. Actualmente, el proyecto de investigación OpenIEC61850 lo patrocina el instituto Fraunhofer ISE.

La librería OpenIEC61850 establece la comunicación entre cliente y servidor implementando el protocolo de comunicación IEC 61850 e interpreta el archivo de extensión \*.icd (creado por el usuario) para escanear los nodos lógicos con todos sus datos y atributos. La consulta de los datos se hace a través de la interfaz gráfica ClientGUI.

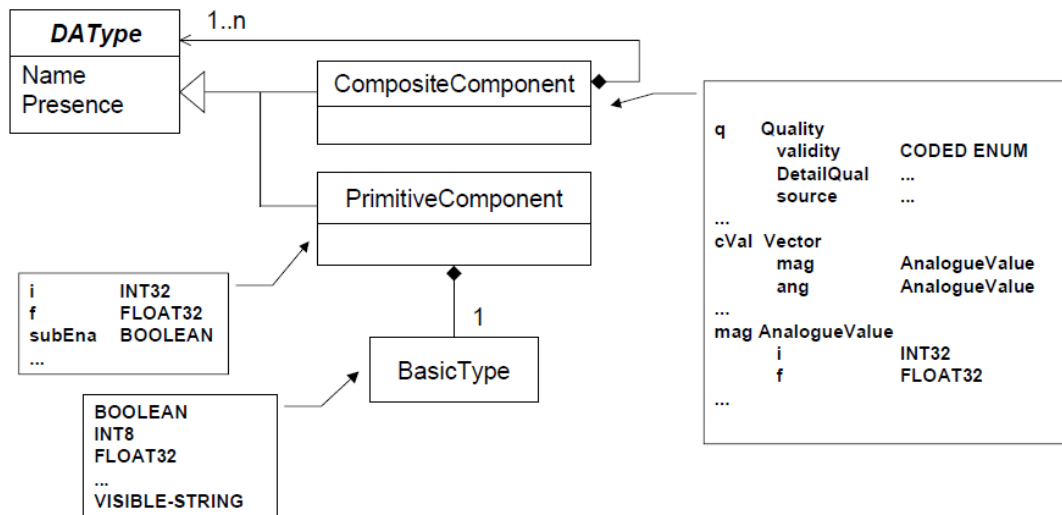
3.3.1 Implementación del servidor de la subestación propuesta

Tal cual como está desarrollada, la librería no permite directamente la escritura de datos (de nivel de proceso de subestación) por parte del usuario, lo cual condujo a modificar parte del código fuente para la inserción de los

datos de proceso provenientes de la simulación desarrollada en ATP. Las modificaciones se hicieron sobre el archivo "SampleServer.java".

Las modificaciones principales al servidor se hicieron sobre la base de los datos y sus correspondientes atributos definidos en el estándar IEC 61850. Un atributo de un dato (DAType) es una clase que tiene un nombre, una indicación (presencia) si el atributo es obligatorio u opcional y, BasicTypes.

Los BasicTypes (por ejemplo, BOOLEAN y INT8) son usados para construir PrimitiveComponents y CompositeComponents. Los PrimitiveComponents deben tener un nombre, una presencia y un BasicType (por ejemplo, Nombre = i, Presencia = Mandatory, BasicType = INT32). Los CompositeComponents se construyen por uno o más PrimitiveComponents de cada BasicType (por ejemplo, Nombre = mag de tipo AnalogueValue que comprende dos PrimitiveComponents i (de INT32) y f (de FLOAT32)). Los CompositeComponents y PrimitiveComponents se emplean para definir las Common Data Class. El concepto de DAType se muestra en la figura 19.



NOTE The attribute Presence is not shown in the examples.

Figura 19. Concepto de DAType. Fuente: (docs.cntd.ru, 2009).

Los BasicTypes se muestran en la figura 20. La figura 21 ilustra cómo se compone un dato (contenido en un LN) por sus atributos, y como estos últimos se componen por CompositeComponents y PrimitiveComponents de cada BasicType.

Con base en la estructura de los DATypes fueron agregadas las líneas encerradas en el rectángulo amarillo que se muestran en la figura 22. Estas líneas sirven para crear variables computacionales (atributos o DATypes) a partir del modelado de la subestación (archivo \*.icd), ejecutando el método findModelNode de la clase ServerModel e indicando su BasicType.

IEC61850_DATATYPE_BOOLEAN	1	Data is of type Boolean, if value is equal to 0 then false, otherwise it true
IEC61850_DATATYPE_INT8	2	An integer of 8 bits
IEC61850_DATATYPE_INT16	3	An integer of 16 bits
IEC61850_DATATYPE_INT32	4	An integer of 32 bits
IEC61850_DATATYPE_INT8U	5	An unsigned integer of 8 bits
IEC61850_DATATYPE_INT16U	6	An unsigned integer of 16 bits
IEC61850_DATATYPE_INT32U	7	An unsigned integer of 32 bits
IEC61850_DATATYPE_FLOAT32	8	A IEEE 754 single precision floating point
IEC61850_DATATYPE_FLOAT64	9	A IEEE 754 double precision floating point
IEC61850_DATATYPE_ENUMERATED	10	Ordered set of values, extended allowed
IEC61850_DATATYPE_CODED_ENUM	11	Ordered set of values. Not allowed to be extended
IEC61850_DATATYPE_OCTET_STRING	12	A String of Octets characters
IEC61850_DATATYPE_VISIBLE_STRING	13	A String of Visible characters
IEC61850_DATATYPE_UNICODE_STRING	14	A String of Unicode characters
IEC61850_DATATYPE_TIMESTAMP	15	TimeStamp type (S.5.3.7.1 of IEC 61850-7-2:2003)
IEC61850_DATATYPE_QUALITY	16	Quality Data type

Figura 20. BasicTypes. Fuente: API User Manual – SystemCORP

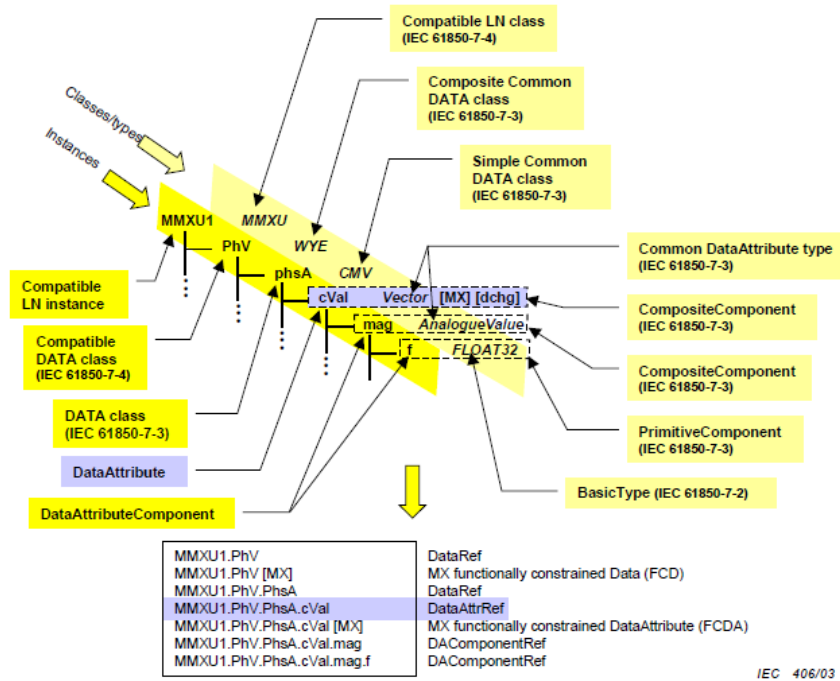


Figura 21. Ejemplo de un dato. Fuente: API User Manual – SystemCORP

```

ServerNode serverNode1 = serverSap.getNodeCopy();

// create a SampleServer instance that can be passed as a callback object to startListening() and
// setDefaultListener()
SampleServer sampleServer = new SampleServer();

serverSap.startListening(sampleServer, sampleServer);
//Agrega comentarios

$defloat32 TRAF19902ApsHA = ($defloat32) serverNode1.findNodeNode("PCICTRAF1/9902.A.phsA.instCVal.mag.f", Fc.W);
$defloat32 TRAF29902ApsHA = ($defloat32) serverNode1.findNodeNode("PCICTRAF2/9902.A.phsA.instCVal.mag.f", Fc.W);
$defloat32 TRAF19902PhVpsHA = ($defloat32) serverNode1.findNodeNode("PCICTRAF1/9902.PHV.phsA.instCVal.mag.f", Fc.W);
$defloat32 TRAF29902PhVpsHA = ($defloat32) serverNode1.findNodeNode("PCICTRAF2/9902.PHV.phsA.instCVal.mag.f", Fc.W);

$defQuality TRAF19902Aq = ($defQuality) serverNode1.findNodeNode("PCICTRAF1/9902.A.phsA.q", Fc.W);
$defTimestamp TRAF19902Lat = ($defTimestamp) serverNode1.findNodeNode("PCICTRAF1/9902.A.phsA.t", Fc.W);
    
```

Figura 22. Instrucciones agregadas para crear DATypes. Fuente: elaboración propia.

Los atributos creados fueron los necesarios para construir los datos que representen las tensiones y las corrientes en el primario y el secundario del transformador, en la barra 1 (subestación) del sistema de potencia propuesto. Adicionalmente, se crearon atributos como la calidad  $q$ , y la estampa de tiempo  $t$ , necesarias en un SAS.

El servidor de OpenIEC61850 organiza los atributos en una lista antes de transmitirlos al cliente, dicha lista fue organizada con los atributos previamente creados, y se muestra en las líneas encerradas dentro del cuadro en la figura 23.

```
BdaQuality TRAF1MMXU1Aq = (BdaQuality) serverModel.findModelNode("PC1CTRAF1/MMXU1.A.phsA.q", Fc.MX);
BdaTimestamp TRAF1MMXU1At = (BdaTimestamp) serverModel.findModelNode("PC1CTRAF1/MMXU1.A.phsA.t", Fc.MX);

List<BasicDataAttribute> datosBdas = new ArrayList<BasicDataAttribute>(4);
datosBdas.add(TRAF1MMXU1ApshA);
datosBdas.add(TRAF2MMXU1ApshA);
datosBdas.add(TRAF1MMXU1PhVpshA);
datosBdas.add(TRAF2MMXU1PhVpshA);

datosBdas.add(TRAF1MMXU1Aq);
datosBdas.add(TRAF1MMXU1At);
```

Figura 23. Instrucciones Agregadas para crear una lista de DATypes. Fuente: elaboración propia.

Los datos provenientes de la simulación fueron guardados en un archivo de extensión \*.csv (datos separados por comas), con el objetivo de utilizar la librería Javacsv para su manipulación dentro de OpenIEC61850.

Las líneas dentro del cuadro rojo de la figura 24 capturan el archivo \*.csv y leen el encabezado del mismo, mientras que las del cuadro violeta, asignan a una variable creada (de tipo flotante. Ej.: TRAF1MMXU1ApshAMag) el valor contenido en la columna del archivo \*.csv (datos de simulación)

correspondiente al nombre del encabezado de dicha columna.

Las líneas agregadas dentro del cuadro verde en la figura 24, asignan a los DATypes creados, los valores de la simulación a través del método setFloat. Las últimas dos líneas dentro del cuadro verde asignan los valores de los atributos  $q$  y  $t$ .

Finalmente, el servidor envía los datos de la simulación al cliente con una frecuencia de 1 ms como se observa en la figura 25.

```
datosBdas.add(TRAF1MMXU1Aq);
datosBdas.add(TRAF1MMXU1At);

try {
    CsvReader datos = new CsvReader("C:/Users/fandres/openiec61850/prueba2.csv");
    datos.readHeaders();

    while (datos.readRecord())
    {
        Float TRAF1MMXU1ApshAMag = Float.parseFloat(datos.get("PC1CTRAF1/MMXU1.A.phsA.instCVal.mag.f"));
        Float TRAF2MMXU1ApshAMag = Float.parseFloat(datos.get("PC1CTRAF2/MMXU1.A.phsA.instCVal.mag.f"));
        Float TRAF1MMXU1PhVpshAMag = Float.parseFloat(datos.get("PC1CTRAF1/MMXU1.PhV.phsA.instCVal.mag.f"));
        Float TRAF2MMXU1PhVpshAMag = Float.parseFloat(datos.get("PC1CTRAF2/MMXU1.PhV.phsA.instCVal.mag.f"));
        //Integer TRAF1MMXU1ApshAq = Integer.parseInt(datos.get("PC1CTRAF1/MMXU1.A.phsA.q"));

        TRAF1MMXU1ApshA.setFloat(TRAF1MMXU1ApshAMag);
        TRAF2MMXU1ApshA.setFloat(TRAF2MMXU1ApshAMag);
        TRAF1MMXU1PhVpshA.setFloat(TRAF1MMXU1PhVpshAMag);
        TRAF2MMXU1PhVpshA.setFloat(TRAF2MMXU1PhVpshAMag);

        TRAF1MMXU1Aq.setValidity(BdaQuality.Validity.GOOD);

        TRAF1MMXU1At.setCurrentTime();
    }
}
```

Figura 24. Asignación de valores a los DATypes creados. Fuente: elaboración propia.

```

serverSap.setValues(datosBdas);//enviar datos desde el servidor al cliente

try {
    Thread.sleep(1);// frecuencia de emisión de los datos [ms]
} catch (InterruptedException e) {

}

}
datos.close();

```

Figura 25. Envío de datos desde el servidor. Fuente: elaboración propia.

Las figuras 26 y 27 muestran la implementación del protocolo IEC 61850 en un modelo de comunicación cliente-servidor, que transmite los datos de proceso (simulación) de la subestación modelada, en una red LAN cableada por dos laptops usando un “switch” industrial. Se eligió una red LAN cableada para no depender del medio de transmisión aéreo y disminuir la

latencia. En las subestaciones se usa fibra óptica como medio de transmisión.

Se puede observar en la figura 26 el servidor iniciado, enviando la estructura del modelo jerárquico de la subestación solicitada por el cliente, y en la figura 27 resolviendo la solicitud del cliente, que pide los valores de los atributos.

```

SampleGen [Java Application] C:\Program Files\Java\jdk1.7.0_71\bin\java.exe [13/10/2013 9:23:58]
09:23:58.491 [pool-2-thread-1] INFO o.o.openeic61850.server.Server - Server iniciado
09:24:00.758 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Set a SetNameListDirectory (Set a SetNameList(NAME_VARIABLE)) request
09:24:00.767 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Set a SetNameListDirectory (Set a SetNameList(NAME_VARIABLE)) request
09:24:00.787 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Set a SetNameListDirectory (Set a SetNameList(NAME_VARIABLE)) request
09:24:00.801 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Set a SetNameListDirectory (Set a SetNameList(NAME_VARIABLE)) request
09:24:00.813 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Set a SetNameListDirectory (Set a SetNameList(NAME_VARIABLE)) request
09:24:00.824 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Set a SetNameListDirectory (Set a SetNameList(NAME_VARIABLE)) request
09:24:00.838 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Set a SetNameListDirectory (Set a SetNameList(NAME_VARIABLE)) request

```

Figura 26. Servidor enviando modelo de la subestación. Fuente: elaboración propia.

```

SampleGen [Java Application] C:\Program Files\Java\jdk1.7.0_71\bin\java.exe [13/10/2013 9:23:58]
09:24:01.914 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Get a GetDataValues request.
09:24:01.935 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Get a GetDataValues request.
09:24:01.957 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Get a GetDataValues request.
09:24:01.978 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Get a GetDataValues request.
09:24:01.989 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Get a GetDataValues request.
09:24:01.991 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Get a GetDataValues request.
09:24:01.992 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Get a GetDataValues request.
09:24:01.994 [pool-2-thread-1] INFO o.o.openeic61850.ServerAssociation - Get a GetDataValues request.

```

Figura 27. Servidor enviando valores de los atributos. Fuente: elaboración propia.

La figura 28 muestra al cliente leyendo los valores del atributo instCVal.mag.f, t y q, enviados desde el servidor en instantes distintos.

#### 4. Conclusiones

Se modeló una subestación eléctrica junto con los equipos de corte y seccionamiento, protección de sobrecorriente y medida siguiendo el estándar IEC 61850 usando la herramienta VisualSCL.

Se utilizó la librería OpenIEC61850 para establecer la comunicación entre dos computadoras dentro de un modelo de comunicación cliente-servidor. Adicionalmente se empleó la interfaz gráfica ClientGUI

para monitorizar las variables eléctricas durante ciertos escenarios de falla del sistema.

La velocidad de entrega de los datos dependerá de que tan filtrada esté la información enviada, cuántas variables desea enviar y cuántas realmente de estas utilizará su sistema, los canales de comunicación y la redundancia de la red en caso de fallas, entre más claro se tenga estos parámetros se aprovechará mucho mejor este protocolo de comunicación.

La ventaja del protocolo de comunicaciones es tener variables del sistema de una manera más rápida y eficaz,

pudiendo mejorar valores de mi servicio como prestador de energía, despejar fallas de manera rápida pudiendo evitar catastrofes en mis equipos, el cambio de cableado de cobre por fibra optica para mejor aprovechamiento de

espacios en tableros y bandejas de cableado de sus subestaciones, ademas de la intercomunicacion de equipos de proteccion y control de diferentes marcas.

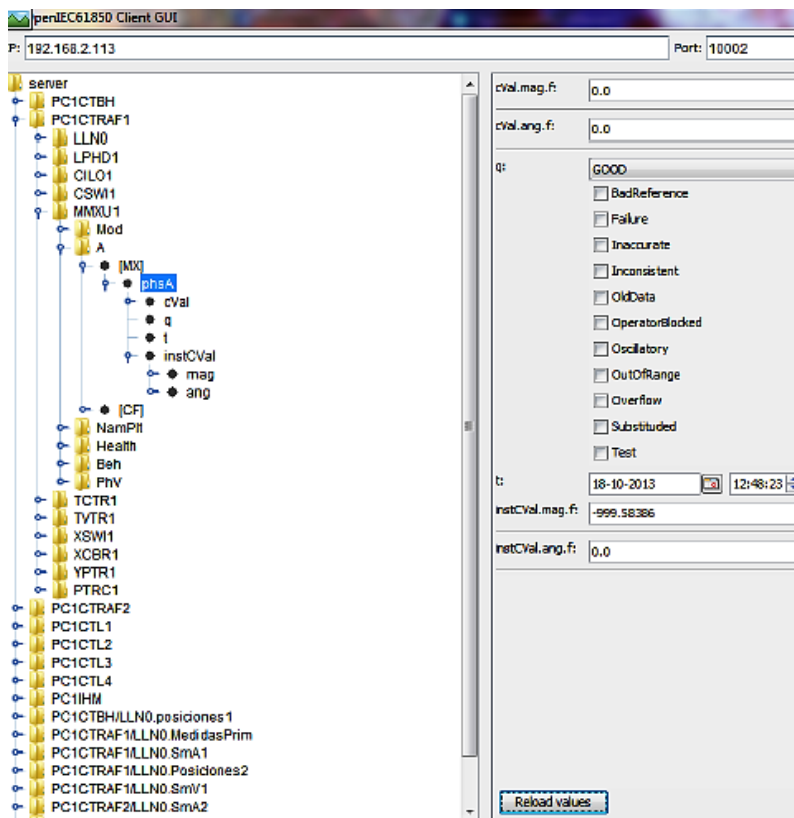


Figura 28. Cliente leyendo los atributos enviados por el servidor. Fuente: elaboración propia.

## 5. Recomendaciones

Dado que el proyecto OpenMuc Framework del Fraunhofer Institute, desarrolla software relacionado con la comunicación dentro de redes inteligentes (Smart Grids) sin ningún tipo de restricción de uso, ya que todo el software es liberado bajo licencia LGPL, se recomienda explorar los proyectos de medición inteligente jDLSM, jMbus y jSML, con el objetivo de crear aplicaciones relacionadas con la medición eléctrica inteligente.

## Agradecimientos

Agradecemos a los ingenieros desarrolladores de la librería OpenIEC61850, especialmente a Michael Zillgith del Fraunhofer ISE, por atender desinteresadamente nuestras inquietudes. Igualmente, a ASE Systems Inc., por otorgarnos la licencia del software Visual SCL para la etapa de modelado.

## Referencias

- [1] R. E. Mackiewicz, "Overview of IEC 61850 and Benefits", IEEE Power Engineering Society General Meeting, 2006.
- [2] IEC 61850, IEC, Edition 2.0 2013-03
- [3] IEC 61850-7, IEC 2011
- [4] IEC 61850-1, IEC 2011
- [5] K-P. Brand, et al. "Substation Automation Handbook". Utility Automation Consulting Lohmann, 2003
- [6] Germán Pugliese-ABB, "IEC 61850: El estándar de integración eléctrica del futuro", 2005.

[7] Visual SCL User Manual. Applied Systems Engineering Inc., 2010.

[8] Kundur Prabha, Power System Stability And Control, McGraw-Hill Education (India) Pvt Limited, 1994.

[9] Arrillaga, C. P. et al, Computer modelling of electrical power systems, John Wiley, 1983

[10] László Prikler, Hans Kristian Høidalen, ATP Draw, SINTEF Energy Research.

[11] IEC 61850 Protocol API User Manual - Protocol Integration Stack, SystemCORP Pty Ltda, 2009

[12] OpenIEC61850 - Overview, Disponible en <http://www.openmuc.org/index.php?id=35>, [Última consulta: 31 de Enero de 2017]