Implementación y análisis de la librería OpenIEC61850 para la simulación de las comunicaciones de datos seguras en subestaciones eléctricas.

Juan Pablo Velandia Malagon

Trabajo de Grado para optar por el título de Ingeniero de Sistemas

Director

Pedro Javier Trujillo Tarazona

Magister en Informática

Universidad Industrial de Santander

Facultad de Ingenierías Físico-Mecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2018

Tabla de contenido

Tabla de figuras

Glosario

ACSI (Abstract Common Services Interface): Punto de acceso a la serie de servicios definidos por IEC 61850 para la comunicación de datos.

CID (Configured IED Description): Archivo que configura un IED basado en ICD, es usado para comunicarse con las herramientas que pueden administrar IEDs.

Compilación (Build): Se refiere en este caso al acto de compilar y desplegar el software para distribución o pruebas resultando en uno o varios archivos preparados.

DER (Distributed Energy Resources): Los recursos en la red de generación de energía, es decir los dispositivos que se encuentra dentro de la misma.

Dispositivo inteligente (IDE): Un dispositivo físico dentro de la subestación eléctrica con la capacidad de conectarse a una red y comunicar datos, no todos los dispositivos presentes en una subestación son considerados IDEs.

ICD (IED Capability Description): Archivo que define las capacidades de un IDE.

IEC (International Electrotechnical Commission): Organizaion internacional de estándares que abarca todas las tecnologías relacionadas a la electricidad y electrónica.

IEC 61850: Serie de estándares elaborados por la IEC para implementar SAS de manera Inter

Javadoc: Documentación autogenerada que describe la implementación de clases Java.

MICS (Model Implementation Conformance Statement)

operativa entre fabricantes.

MMS (Manufacturing Message Specification): Estándar que define las capas OSI de protocolos implementadas para comunicaciones de datos sobre TCP/IP.

OpenIEC61850: Librería de código abierto que implementa el estándar IEC 61850 para comunicar datos entre IEDs.

OSI (Open Systems Interconnection model): Modelo conceptual para la implementación de protocolo en las telecomunicaciones.

PICS (Protocol Implementation Conformance Statement)

PIXIT (Protocol Implementation eXtra Information for Testing)

SAS (Substation Automation System): Sistema de control y automatización para subestaciones en las cuales existan IDEs.

SCD (Substation Configuration Description): Archivo que describe toda la subestación eléctrica en detalle, la combinación de archivos ICD y SSD constituyen un SCD.

SCL (Substation Configuration description Language): Lenguaje basado en XML para representar y configurar IEDs, SCL se refiere al lenguaje, lo cual es diferente a los tipos de formatos, como por ejemplo un archivo .icd que son la implementación de un aspecto de SCL.

Software libre y/o código abierto: Software de distribución libre que no tiene ningún costo y cuyo código esta disponible al publico para ser usado.

SSD (System Specification Description): Archivo que describe el sistema de la subestación.

Subestación eléctrica: Una instalación dentro de la red eléctrica que modifica los niveles de voltaje y transmite la energía a otras instalaciones.

TC (Technical CommiTtee) 57: Grupo de trabajo de la IEC que produjo el estándar IEC 61850.

XML (Extensible Markup Language): Lenguaje que capas de definir objetos de datos para que sea entendido por humanos y al mismo tiempo procesado por máquinas

RESUMEN

TITULO: IMPLEMENTACIÓN Y ANÁLISIS DE LA LIBRERÍA OPENIEC61850 PARA LA SIMULACIÓN DE LAS COMUNICACIONES DE DATOS SEGURAS EN SUBESTACIONES ELÉCTRICAS[*]

AUTOR: JUAN PABLO VELANDIA MALAGON[**]

PALABRAS CLAVE: IEC 61850, SAS, OPENIEC61850, DISPOSITIVOS ELECTRICOS.

DESCRIPCION:

Dentro de una subestación eléctrica existe una red dispositivos inteligentes que intercambian información vital para su funcionamiento, este intercambio de datos debe seguir la serie de estándares para las comunicaciones IEC 61850 la cual busca garantizar integridad e interoperabilidad de los datos, así como la compatibilidad entre los dispositivos en la red.

Los estándares definen las pautas a seguir para implementar el sistema de automatización de la subestación conocido como SAS, y crean la necesidad de tanto herramientas software como de personal capacitado para su implementación.

Este proyecto analiza las características específicas de comunicaciones definidas en la norma IEC 61850, procede entonces a implementar en una solución software la librería de código abierto OpenIEC61850, la cual se basa en estas normas, con el fin de analizar su implementación.

Se desarrolla una interfaz de usuario que permite acceder a las clases lógicas de la librería, y esta se ejecuta en un ambiente de red para trasmitir datos entre dispositivos que siguen la norma IEC 61850, se plantean entonces modelos de dispositivos y subestaciones que son montados sobre la herramienta software, y finalmente se analiza el proceso de comunicación de datos, estudiando los protocolos de comunicación usados durante las pruebas y las características que se resaltan entre desarrollo y ejecución.

---

[*] Trabajo de grado

[**] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Pedro Javier Trujillo Tarazona, Magister en Informática.

ABSTRACT

TITLE: IMPLEMENTATION AND ANALYSIS OF THE OPENIEC61850 LIBRARY FOR THE SIMULATION OF SECURE DATA COMUNICATION IN ELECTRIC SUBSTATIONS[*]

AUTHOR: JUAN PABLO VELANDIA MALAGON[**]

KEYWORDS: IEC 61850, SAS, OPENIEC61850, ELECTRONIC DEVICES.

DESCRIPTION:

Inside an electric substation there is a network of intelligent devices that exchange vital information for its functioning, this data exchange must follow the series of standards for the communications IEC 61850 that aims to ensure the integrity and interoperability of the data as well as the compatibility between the devices in the network.

The standards define the guidelines to implement the automation system of a substation known as SAS and create the need of software tools as well as capable personal for its implementation.

This project analyses the specific communication characteristics defined in the standard IEC 61850, then it proceeds to implement a software solution of the open source library OpenIEC61850, which is built based on the standards, with the goal of analyse its implementation.

An user interface is developed so it allows access to the logical classes of the library, and then this is executed in a network environment to transport data between devices that follow the standard IEC 61850, models for devices and substations are proposed so they are implemented on the software tool, and finally the data communication process is analysed, studying the communication protocols used in testing and the highlighting characteristics from development to execution.

---

[*] Bachelor Thesis

[**] Faculty of Physical and Mechanical Engineering. School of Systems and Informatics Engineering. Director: Pedro Javier Trujillo Tarazona, Master's in Informatics.

Introducción

El sistema eléctrico es una enorme red de dispositivos que se comunican entre sí, intercambiando datos sensibles al mismo tiempo que son controlados y monitoreados, los sistemas que automatizan las subestaciones en las que se encuentran estos dispositivos se vuelven cada día más una necesidad en la industria eléctrica y un reto de ingeniería.

Funcionando como parte fundamental en el proceso de planeación, diseño y construcción de una subestación eléctrica, un Sistema de Automatización de Subestaciones (SAS, Substation Automation System) busca asegurar la transmisión integral de datos y la compatibilidad de los dispositivos dentro de la misma.

Frente a estos retos, principalmente la compatibilidad, se desarrolló el Comité Técnico 57 (Technical Committee 57) por parte de la Comisión Internacional Electrotécnica (IEC, International Electrotechnical Commission) el cual crea el estándar IEC 61850 para definir los dispositivos, sistemas y procesos de comunicación dentro de una subestación eléctrica.

Estandarizar estos sistemas garantiza compatibilidad entre fabricantes de dispositivos y alta confianza en su funcionamiento, del mismo modo crea la necesidad de profesionales capacitados en los estándares para poderlos implementar, así como soluciones software que faciliten el desarrollo del ciclo de vida de un SAS.

En vista a estas necesidades el Instituto Fraunhofer de Alemania en conjunto con empresas del sector de los sistemas de la información y eléctrico, desarrollo OpenIEC61850 como parte un proyecto de investigación en el tema de las redes eléctricas, una librería de código abierto que implementa es estándar IEC 61850 para comunicar dispositivos inteligentes.

Este trabajo de grado

La extensión y complejidad de los estándares son un obstáculo para escribir software en base a él, OpenIEC61850 presenta muchas ventajas para abordar el tema, es de código abierto y está escrita en Java, lo cual, de la flexibilidad y portabilidad, del mismo modo abstrae mucha de la complejidad de los estándares a posibles desarrolladores y deja abiertas posibilidades para múltiples ampliaciones.

El primer capítulo define los objetivos que se buscan lograr con este trabajo de grado, subsecuentemente para conseguirlo el capítulo 2 muestra los fundamentos de marco teórico más importantes, las áreas de conocimiento pueden ser muy amplias pero se busca introducir al lector a los 3 temas más importantes abarcados en este trabajo, las subestaciones eléctricas, la norma IEC 61850 que indica la implementación de SAS dentro de estas subestaciones, y finalmente la librería OpenIEC61850 que lleva al campo de implementación software los temas tratados.

Posteriormente en el capítulo 3 se muestra la implementación de la librería, el estudio de esta, las herramientas de programación, entorno de desarrollo y lógica general, para ser llevadas a una interfaz que puede utilizar sus funciones.

En el capítulo 4 se plantearán modelos propuestos para ejecutar el software obtenido en el capítulo 3 dentro de un ambiente de red con datos basado en IEC 61850, y en el capítulo 5 se resumen las observaciones de los procedimientos observados en la sección de procedimientos 4.3.

Finalmente, en los capítulos 6 y 7 se muestran los aspectos resaltados de la experiencia y conocimiento adquirido al realizar el trabajo junto con recomendaciones para abortar otros proyectos con temas similares o relacionados.

1. Objetivos

1.1 Objetivo general

Implementación de las funcionalidades y características de la librería OpenIEC61850 para la simulación de las comunicaciones de datos en subestaciones eléctricas de acuerdo a la norma IEC 61850, modelando dispositivos de subestaciones que interactúen en red de área local.

1.2 Objetivos específicos

- Implementar la librería OpenIEC61850 en un ambiente de red de área local para simular las comunicaciones en una subestación eléctrica.
- Determinar el rendimiento y seguridad de la librería OpenIEC61850 conforme se estipula en los estándares IEC 61850.
- Modelar procesos de comunicaciones dentro de una subestación eléctrica usando la librería OpenIEC61850 definiendo los dispositivos de la subestación según los requisitos y características de una subestación eléctrica actual.
- Desarrollar una interfaz de usuario para el aprovechamiento de la librería OpenIEC61850 en la simulación de subestaciones eléctricas.
- Verificar el desempeño de la implementación desarrollado mediante pruebas de funcionalidad de comunicaciones.

2. Fundamentos

2.1 Subestaciones Eléctricas

Una subestación eléctrica es una instalación física dentro de la red eléctrica compuesta de dispositivos eléctricos con el fin de modular o establecer niveles de tensión para poder ser distribuida o transmitida a diferentes partes de la red según es requerido.

El tipo más común de subestación es la cual baja la tensión eléctrica hacia niveles en los que pueda ser usado por el consumidor, y es luego transferida, a, por ejemplo, residencias.

La figura 1 muestra dos tipos de subestación, transmisión y distribución y su ubicación dentro de la red eléctrica general.

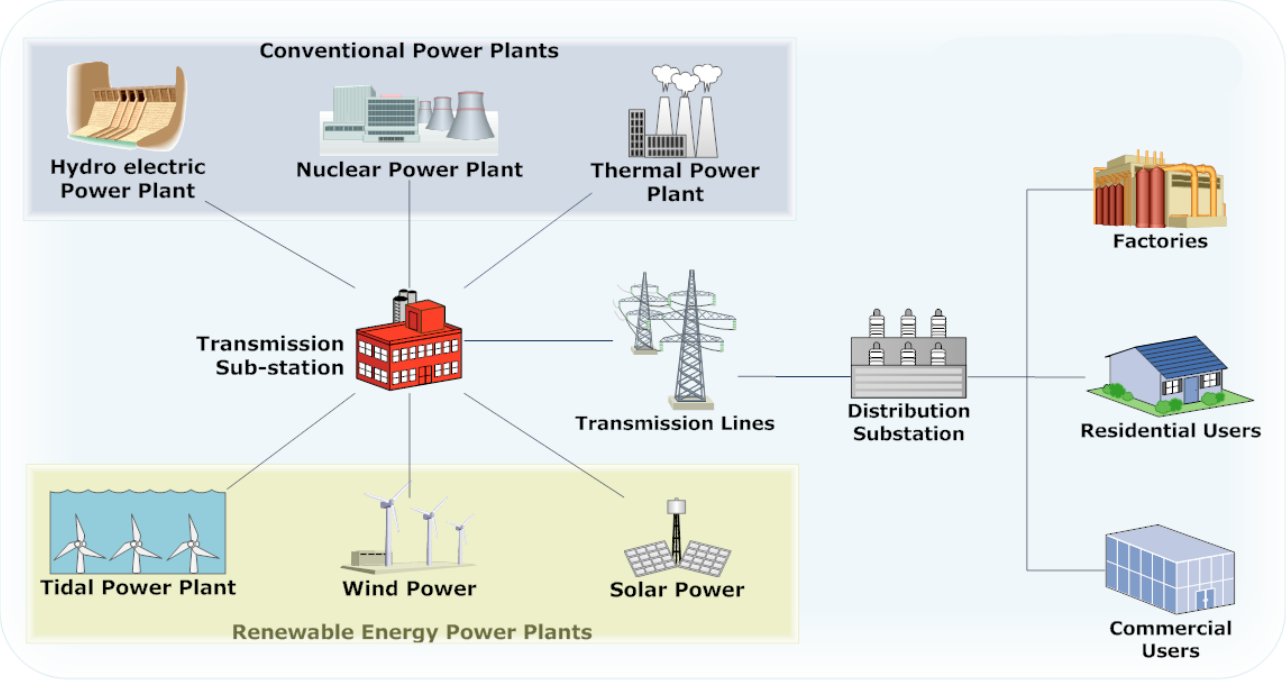


Figura 1: Tipos de subestaciones dentro de la red eléctrica, Fuente: LX Innovations, Intelligence Embedded Power Grids.

Existen diferentes tipos de subestación dependiendo de su tamaño, función o valores de tensión manejados, una subestación eléctrica o en este caso cualquier nodo que se encuentre dentro

de una red eléctrica moderna es sido el producto de múltiples procesos dentro de numerosos campos de la ingeniería, para fines de este trabajo de grado se trabajara dentro del dominio de los sistema software que actúan en estas subestaciones y se introducirán los conceptos básicos necesarios para poder comprender las implementaciones lógicas de software usadas en la industria actualmente, no se busca llegar a niveles avanzados con respecto a aspectos técnicos de electricidad o administrativos del sistema.

Como parte de los esfuerzos de ingeniería dentro de la industria eléctrica existen literatura y procesos relacionados a la construcción de estas subestaciones, esto junto con las necesidades de los clientes creadas por la cantidad proveedores que ofrecen diferentes soluciones para los problemas que se generan al embarcarse en el diseño e implementación de una subestación, produjeron normas como la IEC 61850, la cual busca abordar problemas específicos.

El estándar IEC 61850 abarca su implementación dentro de estas subestaciones eléctricas, el mismo define dos tipos importantes de subestaciones, distribución y transmisión, y a su vez existen diferentes tipos de subclasificaciones, pero en general se abarcan el tamaño de la subestación y las funciones que realiza.

El tamaño de una subestación es dado por la cantidad de dispositivos que posee y los niveles de tensión que maneje, mientras tanto, de acuerdo a su funcionalidad, las subestaciones se pueden dividir en tres grupos, elevadoras, de distribución o reductoras.

Una subestación elevadora toma el voltaje de por ejemplo una planta generadora y lo eleva con el fin de ser transportado más eficientemente, debido a que voltajes más alto reducen la pérdida de poder en las líneas eléctricas, la figura 2 muestra un ejemplo del esquema de un transformador elevador usado en la subestación.

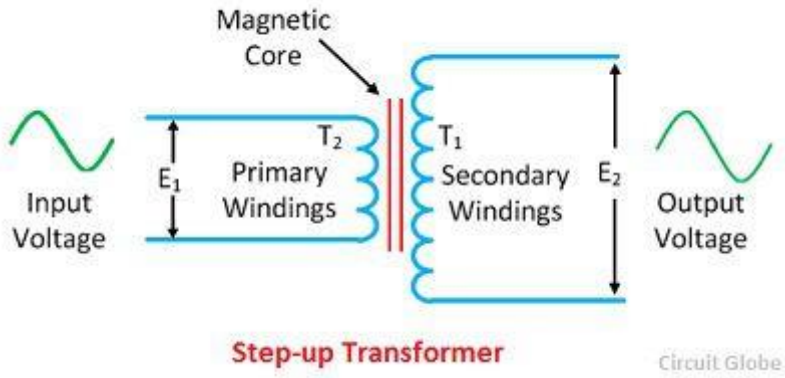


Figura 2: Esquema de un transformador elevador, Fuente: circuitglobe.com

Una subestación reductora baja el voltaje de líneas de transmisión hasta un voltaje de subtransmisión que es generalmente usado en industrias que requieren de voltajes altos o de lo contrario se dirigen a una subestación de distribución, la figura 3 muestra el dispositivo usado en estas subestaciones.

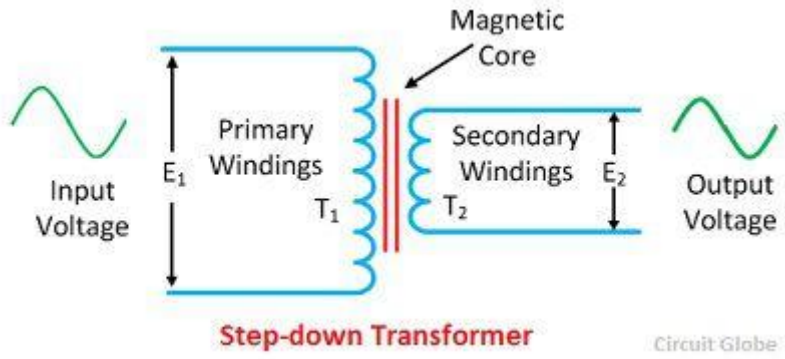


Figura 3: Esquema de un transformador reductor, Fuente: circuitglobe.com

Una subestación de distribución reduce el voltaje incluso más de tal manera que pueda ser usado en redes de distribución, dirigidas a residencias, comercios o industria, para esto se usa un transformador de distribución el cual es el más común en las zonas residenciales.

Para los diagramas de subestaciones usados no habrá diferencia en los tipos de transformadores, a no ser que se indique lo contrario, la figura 4 muestra el símbolo usado para

representar un transformador dentro de la topología de una subestación eléctrica junto con otros dispositivos, sus funciones detalladas se especificaran con forme sea necesario.

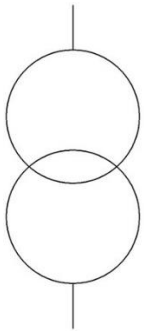


Figura 4: Símbolo usado para representar un transformador dentro del diagrama de una subestación eléctrica.

Las funciones específicas que realiza una subestación están definidas por los dispositivos que posee, y por lo tanto las funciones de estos mismos dispositivos, la topología física de la subestación también influye, sin embargo, para fines de este trabajo de grado nos concentraremos en las funciones físicas y lógicas que realizan los dispositivos.

2.2 Norma IEC 61850

La serie de estándares IEC 61850 es desarrollada por el Comité Técnico 57 de la IEC que se reunió buscando abordar algunos problemas presentes en los sistemas de control y monitoreo dentro del sistema eléctrico, el objetivo principal siempre fue garantizar interoperabilidad entre dispositivos de diferentes fabricantes, con la diferente cantidad de hardware, software y proveedores dentro de la subestación, eran muy comunes los problemas de incompatibilidad, por lo tanto los administradores de subestaciones eléctricas se veían obligados a comprometerse con un único proveedor, lo cual limitaba no solo capacidad y funcionalidad dentro de un sistema en específico si no también reducía la competitividad, capacitación y desarrollo de la industria en general.

Lograr estandarizar la comunicación dentro de un sistema sin tener en cuenta el dispositivo físico o fabricante de este implicaría un beneficio económico y logístico para los clientes, aumentaría la competitividad en la industria facilitando el desarrollo de nuevas tecnologías, y haría converger los conocimientos técnicos necesarios para su implementación. En contraste también implicaría una necesidad de personal capacitado para la correcta implementación de estos estándares.

El estándar surge entonces como una serie de guías y procesos en forma de capítulos que es adaptado rápidamente por la industria.

La serie de estándares se divide en varios capítulos organizados de tal manera que cada uno pueda ser usado para un propósito o aspecto especifico de la subestación, así como para mejorar su entendimiento y facilitar su posterior actualización, la figura 5 muestra la organización básica y dependencias de la serie de estándares cuyo fin es ser implementado en los Dispositivos Electrónicos Inteligentes (IEDs, Intelligent Electronic Devices) dentro de una subestación.

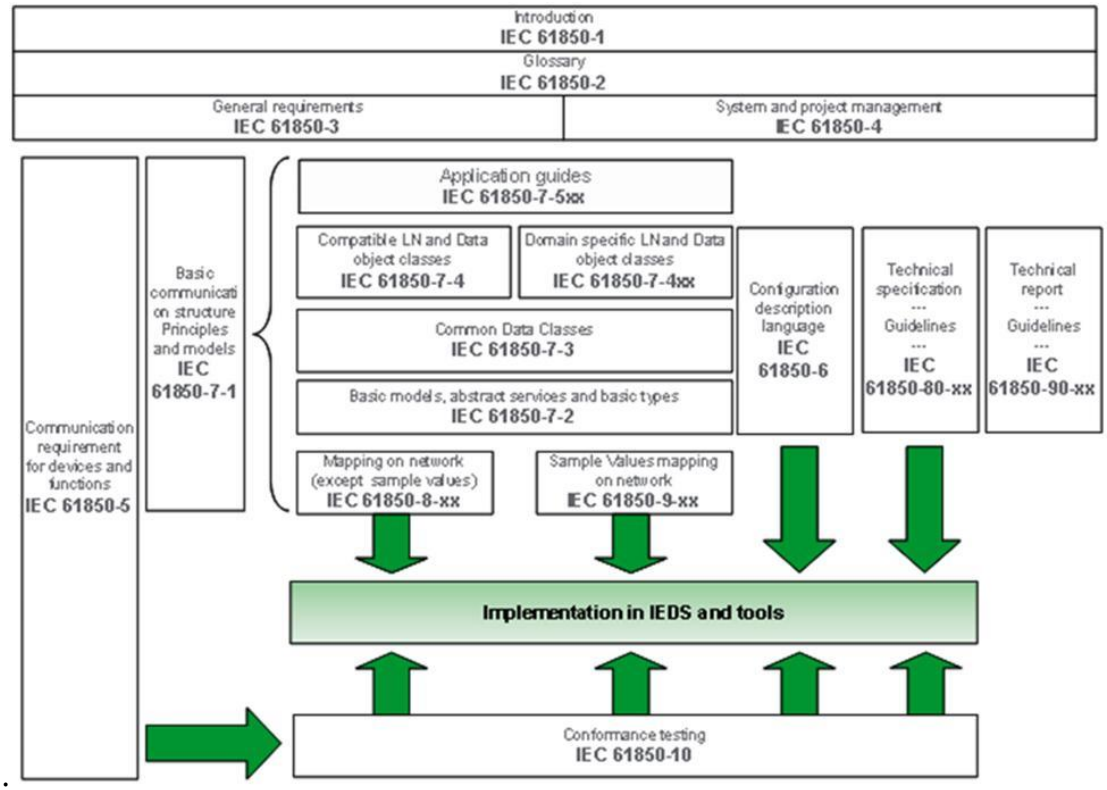


Figura 5: Organización de los capítulos del estándar, Fuente: International Electrotechnical Commission, IEC 61850-1

Durante la existencia del estándar desde su primera versión en 2003 han habido modificaciones, partes agregadas y eliminadas con el fin de mantenerla vigente a las necesidades actuales, se espera del mismo modo que esta siga evolucionando para ajustarse a los avances en la industria y tecnología.

Para fines de este trabajo solo se cubrirán en detalle los fundamentos necesarios únicamente de ciertos capítulos, debido a su longitud y complejidad, no se espera tener un conocimiento profundo de estas normas debido precisamente a que la intención de una solución software es facilitar al usuario su uso ocultando parte de su complejidad.

Los capítulos cubiertos a continuación especifican las bases en las que funciona y se implementa la librería OpenIEC61850 en cual se apoya este trabajo, y en la sección 1.3 se explora

como se implementó específicamente el estándar en software ejecutable, que funciones ofrece y cuál es el alcance de la implementación.

La sección del estándar resaltada que se va a mostrar a continuación cubre:

- Una introducción básica a los SAS y sus características.
- Requisitos para implementar el estándar.
- Configuración y descripción de los dispositivos sobre los cuales se implementa.
- Modelado de datos y comunicaciones.
- Una sección de pruebas y verificación por parte del estándar.

Los capítulos restantes pueden ser ignorados para fines de este trabajo debido a que no se aplican a este caso particular, abarcan temas de administración del proyecto fuera del enfoque técnico de este trabajo que no son usados, son implementaciones de bajo nivel en la librería que abstrae su complejidad escudando al ingeniero de tener que cubrir ciertos campos o cubre casos de implementaciones más complejos no relacionados con los ejemplos básico-usados.

2.2.1 IEC 61850-1: Introduction and Overview. El primer capítulo de la serie de estándares es una introducción a sí mismo y a los capítulos que le siguen, define el alcance y el campo de acción de los estándares con los cuales fue ideado, es el punto de partida ideal para abarcar la documentación requerida si se desea adentrar en este campo.

Funciona como un soporte a los motivos por la existencia del estándar e indica sobre que bases se construyó, todo su contenido se encuentra detallado durante la realización de los capítulos siguientes y por lo tanto también durante la extensión de este trabajo.

Uno de los aspectos más importantes a resaltar es los principios sobre los cuales se construyó el estándar:

El estándar debe asegurar, junto con otras cosas, las siguientes características:

- Que el perfil completo de comunicaciones está basado en estándares IEC/IEEE/ISO/OSI, existentes.
- Que los protocolos usados permitirán usar y soportar dispositivos auto descriptivos. Debe ser posible agregarles una nueva funcionalidad.
- Que el estándar está basado en objetos de datos relacionados a las necesidades de la industria de energía eléctrica.
- Que la sintaxis y semántica de las comunicaciones están basadas en el uso de objetos de datos comunes relaciones al sistema de energía eléctrico.
- Que el estándar de comunicaciones considera las implicaciones de que la subestación es un nodo en la red eléctrica, por ejemplo, que el SAS es un elemento del sistema de control eléctrico.

Esto indica principalmente que el fin del estándar es reunir todos aspectos técnicos de implementación bajo un mismo nombre y otorgar flexibilidad y funcionalidad a los sistemas de control que lo implementen.

En los siguientes capítulos resaltados se mostrarán los aspectos técnicos pertinentes a este trabajo y la implementación software de manera más detallada.

2.2.2 IEC 61850-5: Communication requirements for functions and device models. Esta parte del estándar define los requerimientos para las funciones y modelos de los dispositivos en las subestaciones, esto significa que la manera en la que el estándar interpreta un dispositivo es basada en sus funciones, es decir un IDE es solo un conjunto de funciones que deben ser efectuadas en la subestación. Existen funciones de control, monitoreo y protección de los dispositivos, así como funciones de mantenimiento para el SAS como configuración y administración de software.

Esta es una de las partes más extensas del estándar y también una de la más técnicas ya que define todos los tipos de funciones y dispositivos que pueden actuar en una subestación, pero lo más importante a resultar es que si bien el estándar comprende las funciones que se efectúan por los dispositivos, no comprende los dispositivos físicos como tal, el dispositivo implementa lo que el estándar define como nodos lógicos (LN, Logical Node) el cual es la parte más pequeña de una función que puede comunicar datos, en la figura 6 de muestra cómo se dividen funciones de nodos lógicos y dispositivos con algunos ejemplos.

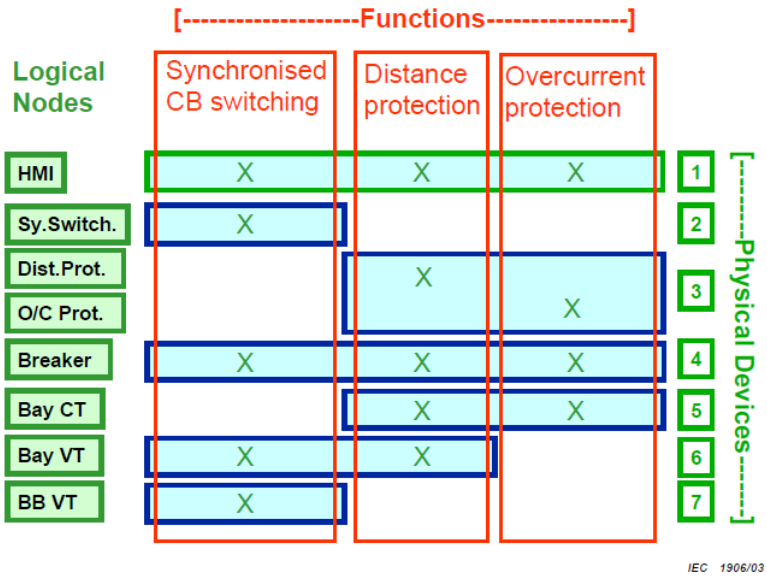


Figura 6: Clasificación de dispositivo físico y nodos lógicos por funciones, Fuente: International

Lo que nos muestra la figura es un ejemplo de algunas funciones y como estas son realizadas por nodos lógicos, los cuales son implementados por los dispositivos físicos que realizan las funciones, por ejemplo, el dispositivo número 1 es un computador que cae dentro del nodo lógico de tipo HMI (Human Machine Interface) y abarca las tres funciones de conmutación de disyuntor sincronizada, protección a distancia y protección de sobrecarga.

El estándar procede entonces a definir y agrupar todos los tipos de nodos lógicos, funciones y datos que se intercambian.

2.2.3 IEC 61850-6: Configuration language for communication in electrical substations related to IEDs. Esta es la parte más pertinente para fines de este trabajo, para representar los IDEs de un SAS, esta parte del estándar especifica un formato de archivo escrito en un lenguaje especial que describe la configuración y parámetros del dispositivo llamado Substation Configuration description Language (SCL) basado en XML, el propósito principal del formato es intercambiar las descripciones de capacidades de IDEs y descripciones del SAS entre las herramientas que los administran así como la posibilidad de intercambiarlo entre herramientas de diferentes fabricantes de manera compatible.

El SCL describe la estructura de sistema energético, que dispositivos se usan y como se conectan, su adyacente sistema de comunicación, así como la manera en la que se manipulan los datos para ser enviados, cada IDE y su descripción según IEC 61850-5, define los nodos lógicos y sus funciones.

A manera de resumen la estructura del lenguaje general es dada por el siguiente esquema en notación UML, existen ampliaciones y variaciones específicas conforme se revisan los estándares, pero una aproximación general está dada en la figura 7.

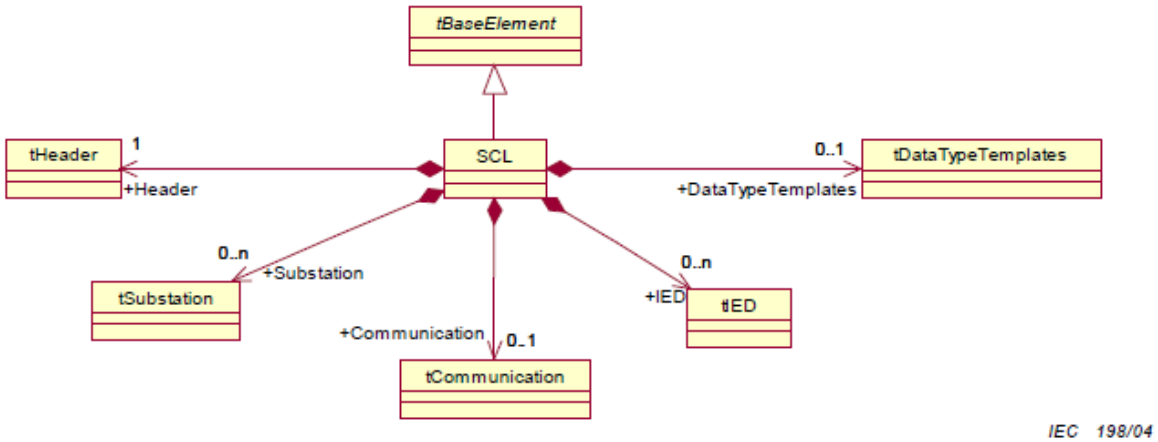


Figura 7: Modelo UML de un archivo SCL, Fuente: International Electrotechnical Commission, IEC 61850-6

El elemento *Header* sirve para identificar el archivo como un archivo de configuración SCL y cuál es su versión, el elemento *Substation* describe la subestación de la manera como sus dispositivos están comunicándose y el elemento *Communication* define la manera en la que se comunican los elementos de la subestación dentro de redes y subredes.

Cada IED presente en la subestación es descrito por el elemento IDE, sus nodos lógicos a que clase pertenece y los datos que maneja, aquí es donde este trabajo se va a centrar, porque este es un archivo de configuración especifico ICD (IED Capability Description) el cual describe los dispositivos que serán usados posteriormente.

2.2.4 IEC 61850-7: Basic communication structure for substation and feeder equipment and subparts. En esta parte del estándar se busca la unificación de las estructuras y modelos usados en el control de la subestación y sus comunicaciones, esta es otra parte extensa dividida en 4 sub-partes para los principios y modelos usados.

La estandarización de las comunicaciones viene dada por la estructura de la información y como se modela, define las interfaces que comunicaran los diferentes servicios a través de la subestación, define las clases de datos y su compatibilidad, esta parte es la que le da

interoperabilidad del sistema al estándar, por ello el estándar ha sido ampliado para el contexto de plantas hidroeléctricas, y energía distribuida entre otros, las especificaciones de todos estos modelos son muy extensas y específicas para cada tipo de aplicación por lo que no se profundizara mucho en esta parte.

2.2.5 IEC 61850-8 & 61850-9: Specific communication service mapping (SCSM). En estos dos capítulos se definen como se implementan los protocolos de comunicación para los servicios prestados por el software, es decir de qué manera se implementan estructuras de datos, nomenclaturas y los médelos de objetos definidos en los estándares anteriores para ser implementados.

El capítulo 8 provee las operaciones que permiten comunicación entre una variedad de dispositivos dentro de una subestación para lograr la compatibilidad deseada, hace esto dando información detallada de cómo se intercambian los mensajes y que protocolo usar dependiendo del tipo objeto.

El capítulo 9 se divide en diferentes implementaciones específicas para la comunicación en varios niveles del modelo OSI, por ejemplo, la sección 9-1 está definida para una conexión punto a punto unidireccional, pero debido a la poca practicidad de este ha sido descartado de la serie de estándares para dar lugar a tecnologías más modernas, mientras que la sección 9-2 implementa ISO/IEC 8802-3 para modelos definidos en IEC 61850-7-2.

La figura 8 muestra como diferentes tipos de mensajes viajan sobre los diferentes protocolos implementados, dependiendo de sus requisitos específicos, el estándar define 6 tipos de mensajes según su velocidad de transmisión siendo los de tipo 1 los más rápidos y los de tipo 6

los más lentos, como por ejemplo transferencia de archivos y por lo tanto tendrán requisitos diferentes en cuanto al protocolo a usar.

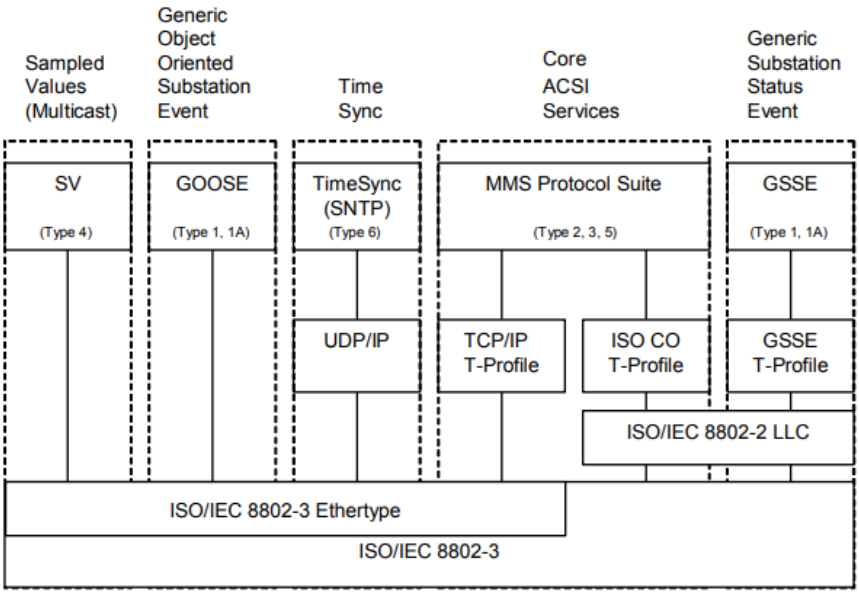


Figura 8: Funcionalidad de los perfiles usados sobre los protocolos de comunicación

2.2.6 IEC 61850-10: Conformance testing. Existen diferentes tipos de procesos involucrados en el diseño e implementación de un sistema completamente funcional, este capítulo busca que mediante pruebas se pueda verificar que se están cumpliendo con los requisitos y necesidades planteadas en el momento de diseño.

En la figura 9 se muestra el diagrama de flujo básico que se asegura, en una serie de pruebas post despliegue, que el sistema cumple con los requisitos, para fines de este trabajo de grado esta fuera del alcance considerar todos los factores de una implementación real de un SAS, para fines prácticos se implementaran las recomendaciones y guía para conectividad básica que puedan ser aplicables a los modelos finales.

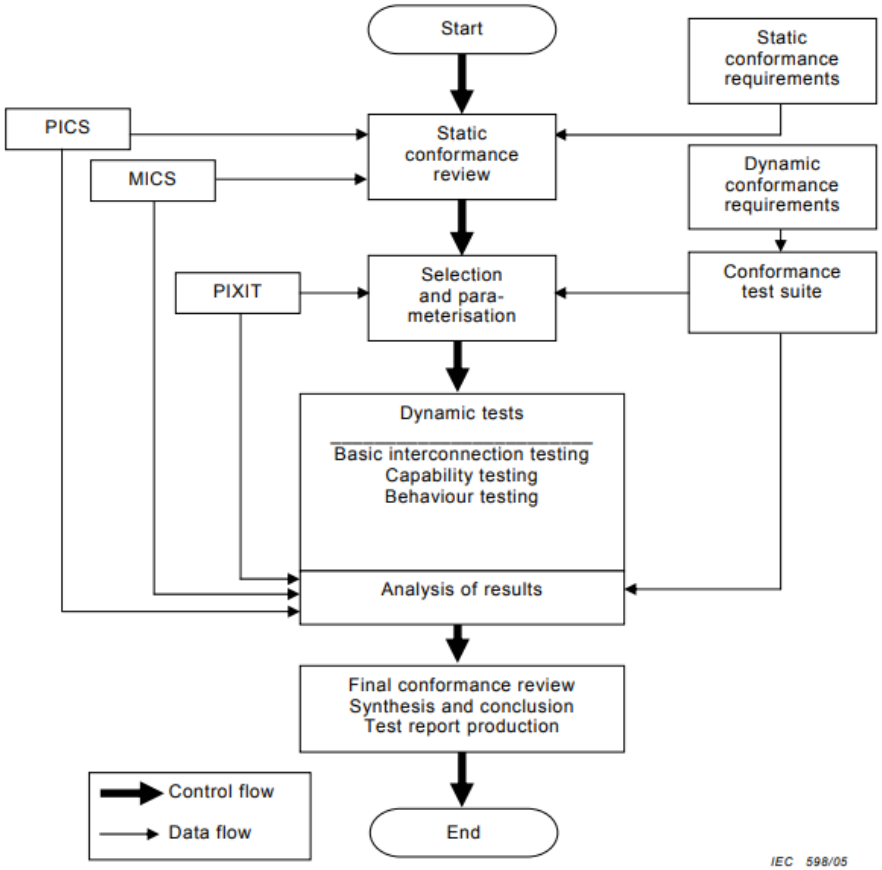


Figura 9: Grafica del proceso efectuado para pruebas de conformidad, Fuente: IEC 61850-10

2.3 Librería OpenIEC61850

El instituto Fraunhofer para sistemas de energía solar (ISE), Energy&Meteo Systems, OFFIS (Instituto para las tecnologías para la información) dentro del contexto del proyecto fundado por el gobierno alemán eTelligence buscaban usar diferentes enfoques de las tecnologías de información y comunicaciones para mejorar los sistemas proveedores de energía y habilitar la integración de sistemas de energía renovables.

Aquí empezó el desarrollo de la librería de código abierto OpenIEC61850 que implementa las funciones básicas de comunicaciones descritas en la serie de estándares IEC 61850, después de finalizado el proyecto el Instituto Fraunhofer da soporte activo a la librería como parte de su

solución OpenMUC para desarrollar sistemas de control y monitoreo, el cual está disponible para su descarga en https://www.openmuc.org/.

OpenIEC61850 permite crear aplicaciones que conectan dispositivos IDE dentro de una red local, los cuales están definidos en la norma IEC 61850 junto con ciertos servicios y funcionalidades.

En el documento javadoc disponible junto con la librería o en la página de documentación en línea, se puede encontrar la lista completa de clases implementadas disponibles en la librería para ser usadas, los métodos que implementa, atributos, argumentos y descripciones.

2.3.1 Marco de referencia. OpenIEC61850 es distribuido como parte de varias librerías que se centran en clases y funciones de Java que ejecutan la conexión y transmisión de datos, OpenIEC61850 es la principal desde la cual se definen los servicios disponibles para su implementación, también existen otras dependencias para usar ciertos protocolos de comunicación y servicios otorgados, principalmente la capacidad registrar los eventos y que ocurren en un dispositivo mediante el cual se comunica.

El desarrollo de una ampliación que implementa la librería llama a la instancia de una clase que implementa ya sea el cliente o el servidor, el servidor define un dispositivo dado.

Del mismo modo se llama a una función que implementa un cliente el cual se conecta mediante IP y puerto al servidor solicitando datos, la norma IEC 61850 define la conexión y las funciones que se pueden hacer en la misma, en su versión actual la librería permite:

- Acceder a los servicios para obtener la estructura de datos.
- Acceder y escribir valores de datos.
- Reportar eventos con o sin buffer.

- Control

En el capítulo 3 se explicará a profundidad como se implementó casa clase en este caso y la lógica usada, del mismo modo en el capítulo 4 se señalará en qué momento se usó algún servicio específico de la librería.

2.3.3 Comunicación de datos. La librería cubre la comunicación basado en un mapeo a MMS (Manufacturing Message Specification), un protocolo de comunicaciones descrito en la norma ISO 9506, este mapeo de servicios específicos se describe en la parte 8-1 del estándar que especifica la arquitectura de comunicación basada en capas OSI, dentro de la subestación.

La figura 10 muestra el modelo OSI que usa MMS y que es implementado en la librería mediante APIs (Application programming interface) de comunicación.

Application	Manufacturing Message Specification (MMS) – ISO/IEC 9506 Association Control Service Element (ACSE) – ISO 8649/8650
Presentation	Connection Oriented Presentation – ISO 8822/8823 Abstract Syntax Notation (ASN) – ISO 8824/8825
Session	Connection Oriented Session – ISO 8326/8327
Transport	ISO Transport over TCP – RFC-1006 Transmission Control Protocol (TCP) – RFC 793
Network	Internet Control Message Protocol (ICMP) – RFC 792 Internet Protocol (IP) – RFC 791
Link	IP Datagrams over Ethernet – RFC 894 MAC – Ethernet – ISO 8802-3
Physical	Ethernet

Figura 10: Capas de comunicación del protocolo MMS sobre TCP/IP, Fuente: Xelas Energy

Para implementar esta arquitectura la librería posee dos APIs, *josistack* que implementa las tres capas superiores y *jositransport* que implementa la capa de transporte, los cuales se encapsularan en datagramas IP que viajan sobre sobre Ethernet.

2.3.3 Software. OpenIEC61850 está escrito en Java, y también existe una versión alternativa en C. Basado en los estándares IEC 61850 la librería se ejecuta bajo la estructura cliente-servidor, en este contexto cuanto se ejecuta un servidor, este es el dispositivo IDE definido en un archivo ICD que describen el dispositivo y sus características. El cliente lee los datos del servidor mediante solicitudes en forma de un modelo de datos jerárquico.

El software el distribuido como un archivo comprimido *.tgz* el cual contiene, en la versión 1.41 usada en este trabajo:

- El código fuente de la implementación OpenIEC61850 y todas sus dependencias.
- Documentación del código fuente en java (javadoc).
- Licencias.
- Implementación de un cliente, servidor y una interfaz gráfica a manera de ejemplo.
- Implementación de Gradle para compilar e implementar versiones modificadas del software de manera automática.
- Carpetas con proyectos y archivos de pruebas para verificar funcionalidad.

Los archivos *.jar* ejecutables bajo la carpeta *build\libs-all* contienen todo el código fuente compilado, openiec61850-1.4.1.jar es la implementación central del cliente y servidor basados en el estándar los cual serán importados como referencia cuando se desarrolle una aplicación.

openiec61850-clientgui es una interfaz gráfica de usuario para el cliente que puede conectarse a un servidor además de leer y escribir datos.

3. Implementación

3.1 Herramientas y requerimientos del entorno de desarrollo

El fin de este trabajo es producir una implementación de funcionalidades básicas de una librería de código abierto. Inicialmente si se quiere ejecutar OpenIEC61850 siguiendo las instrucciones del desarrollador, el entorno de ejecución necesita únicamente JRE 7 o superior, adicionalmente si se desea modificar y compilar la herramienta se requiere mínimamente JDK 7 o superior, un editor de texto y Gradle si se desea recompilar modificaciones al código fuente, este último se descarga e instala automáticamente usando el Gradle Wrapper incluido en el paquete cuando se descarga OpenIEC61850 ejecutando el archivo gradle.bat, esto es para asegurar compatibilidad y robustez en las compilaciones.

En nuestro caso pese a que con los requisitos de desarrollo previamente mencionados son suficiente para lograr las metas, por conveniencia y funcionalidad se usaron una serie de herramientas adicionales, a continuación, se mencionan todas las herramientas usadas durante la construcción de software:

- JDK 1.8.0_162
- Eclipse IDE Oxygen 4.7.3a con WindowBuilder para diseño de interfaces.
- Gradle 4.4 y Gradle wrapper puede ser usado en integración con eclipse para construir software.
- OpenSCLConfigurator, un programa de código abierto bajo la licencia GNU para crear y modificar archivos SCL que describen IEDs.
- Openiec61850 1.4.1 con sus respectivas librerías referenciadas.

- Una implementación en Java para dirigir todos los resultados de la consola a la interfaz gráfica llamada MessageConsole.
- Windows 10 Home

3.2 Funcionalidades y requisitos

Las funcionalidades básicas del software son:

- Proporcionar una interfaz para acceder a las funciones de la librería OpenIEC61850 sin necesidad de compilar ni trabajar desde la línea de comandos.
- Crear o importar modelos de dispositivos según sea necesario.
- Visualizar y actualizar los datos entregados por un servidor para un dispositivo dado.
- Ver la conexiones y solicitudes realizadas a un servidor.

A final el software deberá ejecutarse, dar la opción para iniciar ya sea como cliente o servidor y proporcionar las opciones relacionadas, si se ejecuta como servidor, se deberá poder crear un nuevo archivo que defina un dispositivo e iniciar el servidor con el mismo archivo de configuración escuchando por conexiones dentro de la red de área local en un puerto especificado.

Si se ejecuta como cliente debe dar la posibilidad de conectarse a un servidor activo dentro de la red de área local, recuperar la estructura de datos y mostrar los valores actuales del servidor, también como escribir datos y modificar variables mediante la interfaz gráfica y ver los cambios reflejados en el lado del servidor.

3.3 Diseño

La vista general de un software que implementa código externo y agnóstico a su implementación esta mostrado en la figura 11.

Figura 11: Diagrama básico de la relación de la implementación software y la librería.

La manera en la que las clases Java son llamadas e implementadas en el software es descrito en la figura 12.

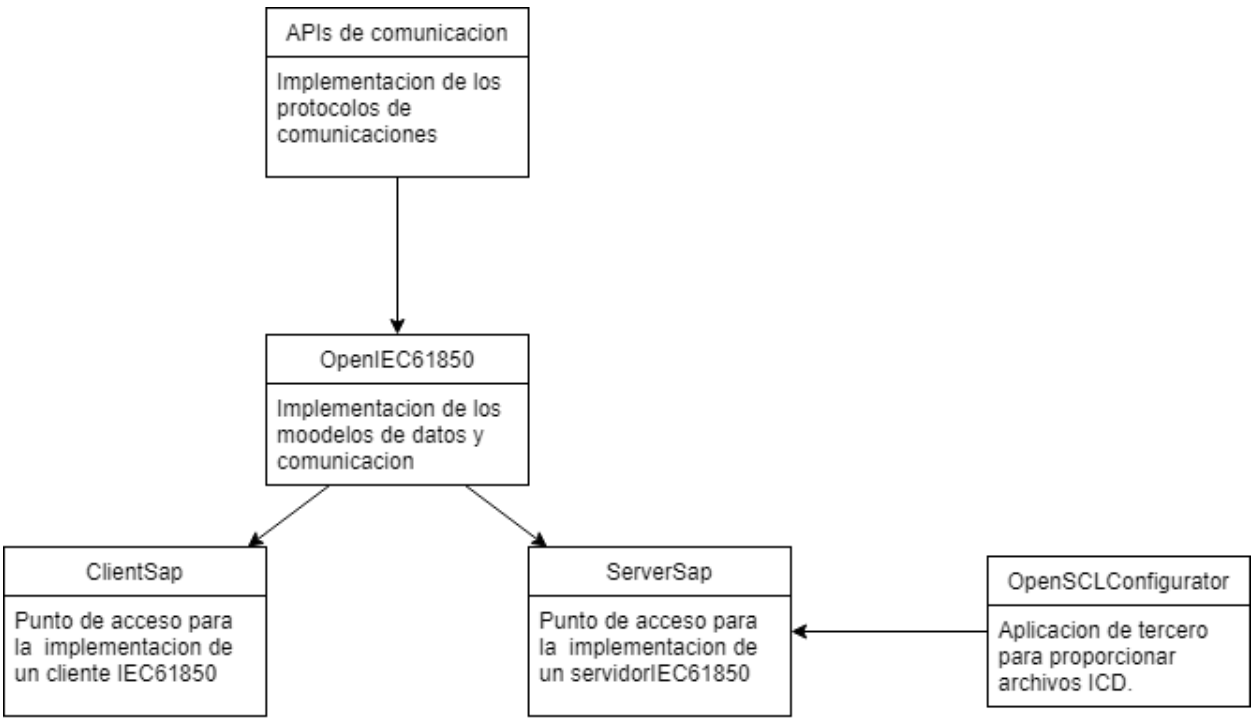


Figura 12: Diagrama descriptivo para los componentes del software

OpenIEC61850 implementa los siguientes tipos de nodos en el servidor:

- *ServerModel*
- *LogivalDevice*
- *LogicalNode*
- *FcDataObject*
- *Array*
- *ConstructedDataAttribute*
- *BasicDataAttribute*

Sin embargo, nos concentraremos solamente en *ServerModel* y en *BasicDataAttribute*, *ServerModel* guarda toda la estructura del servidor (El dispositivo definido en el archivo ICD), y es el que recuperamos en el cliente mediante una asociación de la capa de aplicación del modelo OSI para MMS mostrado en la figura 10.

Mientras que *BasicDataAttribute* es un tipo de dato simple que se puede leer y escribir en el servidor y el cual usaremos para enviar por la red y detectar cambios, definiendo así el comportamiento de un dispositivo, como por ejemplo la medición de un voltaje o el estado de un interruptor.

3.4 Licencias

Openiec61850 se distribuye bajo la licencia Apache 2.0, del mismo modo otras herramientas o código fuente de terceros pueden contener licencias de distribución o uso libre, el propósito de este software es meramente académico y los componentes usados que no fueron producidos por el autor, principalmente código e implementación específica, son de código abierto o distribución libre cumpliendo con la ley, en Colombia el software libre se rige por las leyes de los derechos de autor

y las libertades que da la licencia, en el código fuente del software y en la documentación adjunta se da reconocimiento a los autores originales del código y se señala si se hicieron cambios al mismo.

Los siguientes son los módulos usado con licencia, su autor y las licencias que llevan, si no se señala en específico se asume que es el autor.

- Openiec61850 y todas las dependencias con las que es distribuido: Fraunhofer ISE, Apache 2.0.
- Gradle: Gradle, Inc. Terms of Service, https://gradle.org/terms/
- OpenSCLConfigurator: Comisión Federal de Electricidad de México, GNU General Public License.

3.5 Procedimiento y código

A continuación, se describen los pasos para crear un proyecto, incluir dependencias, desarrollar y compilar.

Para empezar a desarrollar una aplicación en eclipse IDE creo un proyecto de tipo Java Project, es posible seleccionar otras opciones teniendo el miso resultado, pero este fue el procedimiento realizado.

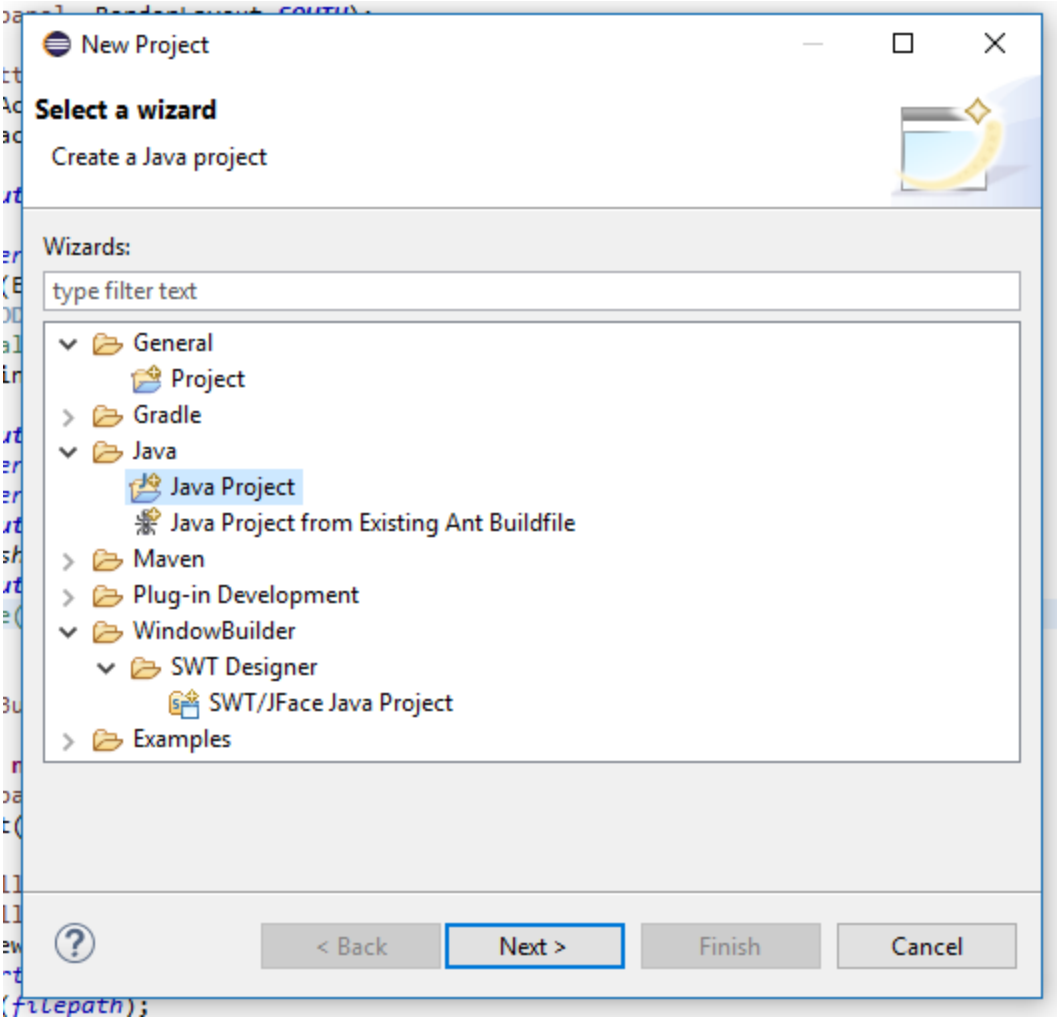
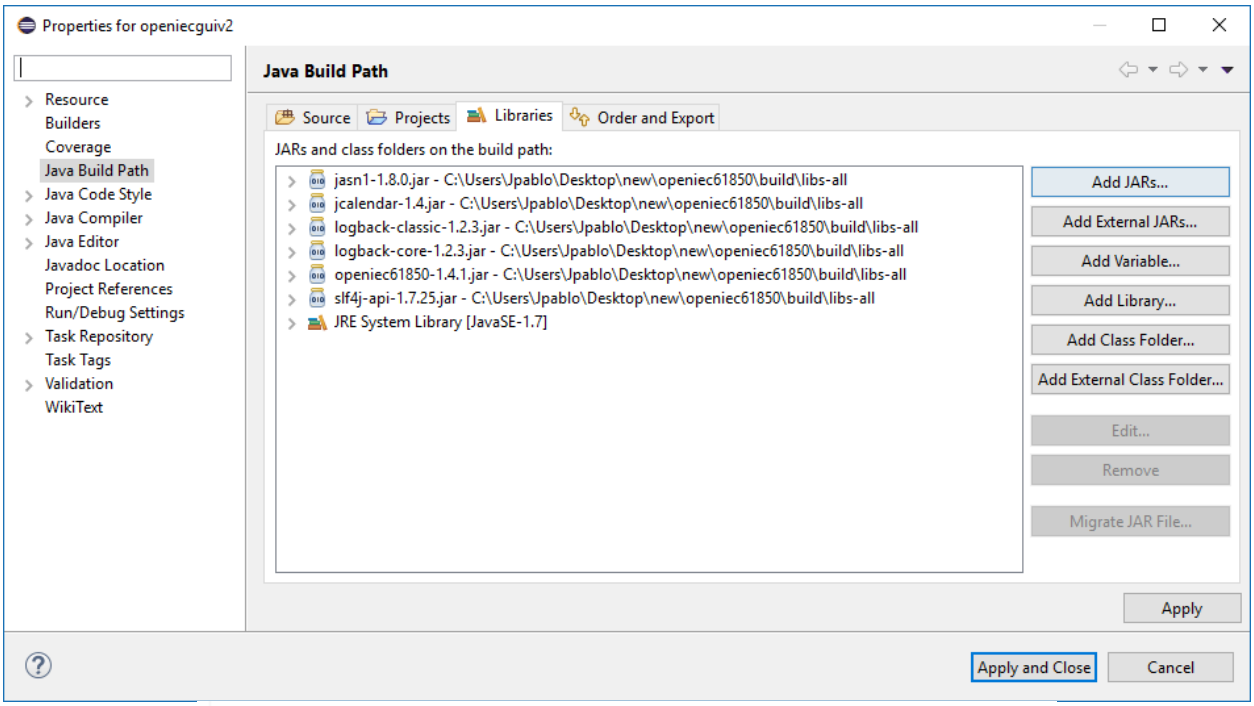


Figura 13: Creando un proyecto Java

Posteriormente si se desea trabajar con librerías se les debe hacer referencia en el proyecto, en este caso como librerías JAR externas, de esta manera cuando se creen nuevos archivos .java importando las clases usadas se puede llamar a la librería.




```
import org.openmuc.openiec61850.BasicDataAttribute;
import org.openmuc.openiec61850.SclParseException;
import org.openmuc.openiec61850.ServerEventListener;
import org.openmuc.openiec61850.ServerModel;
import org.openmuc.openiec61850.ServerSap;
```

Figura 14: Referenciando las librerías necesarias

Lo siguiente es la estructura del código fuente del software, para realizar la interfaz gráfica se usó la herramienta de eclipse *WindowBuilder* que puede ser descargada como un componente de Eclipse IDE, con esto se pueden realizar cambios a la interfaz gráfica sin tener que escribir código manualmente, y se usa la estructura de vistas *CardLayout* otorgada por la API de Java para manejar las que ventanas del programa se muestran dependiendo que las acciones y poder trabajar de manera modular.

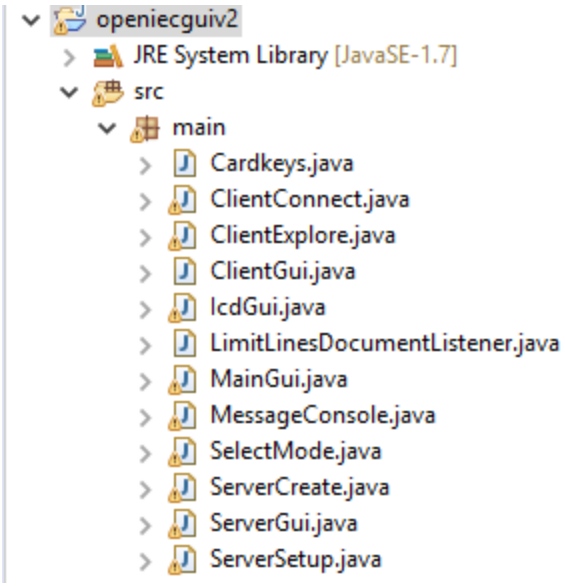


Figura 15: Clases Java implementadas

Por lo tanto, cuando en la clase que implementa y ejecuta un cliente IEC6150 se llama a la clase *ClientSap()* y mediante el método *associate()* dada una dirección IP y un puerto al cual conectarse, los cuales correspondan a un servidor IEC 61850 que este escuchando por conexiones se puede obtener un modelo de datos del servidor con *association.retrieveModel().*

Sobre este objeto de tipo *ServerModel* proporcionado por la librería se pueden navegar, consultar por información y escribir nueva información si el campo está habilitado para edición, en las siguientes partes de código, se crea una instancia de *ClientSap,* se inicia la asociación al servidor especificado y se obtiene el modelo del servidor mediante la conexión.

```
private void connect() {
    ClientSap clientSap = new ClientSap();


try {
    association = clientSap.associate(address, remotePort, null, null);
} catch (IOException e) {
    System.out.println("Error connecting to server: " + e.getMessage());
    return;
}
```

```
ServerModel serverModel;
try {
    serverModel = association.retrieveModel();
    association.getAllDataValues();
```

En el lado del servidor se necesita configurar un dispositivo inteligente que tenga capacidad de comunicar sus datos, el servidor carga un archivo de configuración que define al dispositivo y es habilitado mediante un puerto para recibir solicitudes entrantes, para implementar este comportamiento se llama a la clase *ServerSaps*, la cual se alimenta de la información del archivo SCL proporcionado y después de configurar el puerto se le indica al servidor que comience a escuchar por conexiones entrante con *serverSap.startListening()*.

```
List<ServerSap> serverSaps = null;
   try {
       serverSaps = ServerSap.getSapsFromSclFile(filepath);
   } catch (SclParseException e) {
       txtrIecServerStarted.append("Error parsing SCL/ICD file: " + e.getMessage());
       return;
   }
   serverSap = serverSaps.get(0);
   serverModel = serverSap.getModelCopy();
   serverSap.setPort(102);
   //System.out.println(serverModel);
   try {
       serverSap.startListening(new EventListener());
   } catch (IOException e) {
       // TODO Auto-generated catch block
       e.printStackTrace();
   }
   txtrIecServerStarted.append("server tree here");
```

El resultado de la interfaz gráfica funcional se muestra en las figuras 16 a 20.

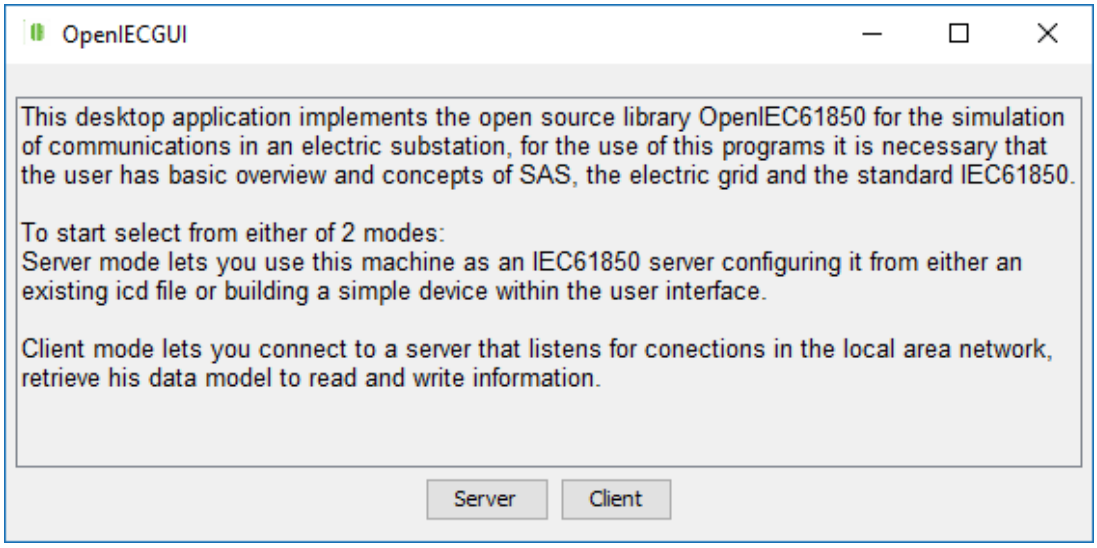


Figura 16: Ventana inicial del software con opciones cliente y servidor

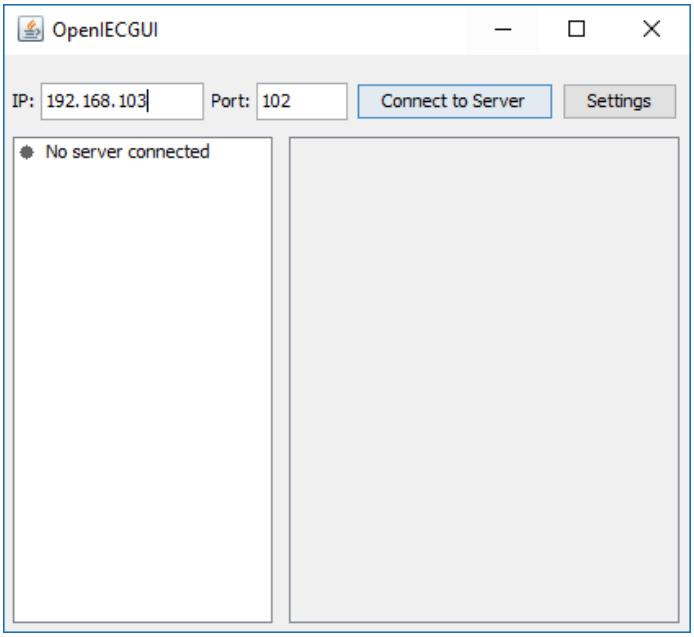


Figura 17: Conectándose a un servidor de IP y puerto conocidos que este escuchando

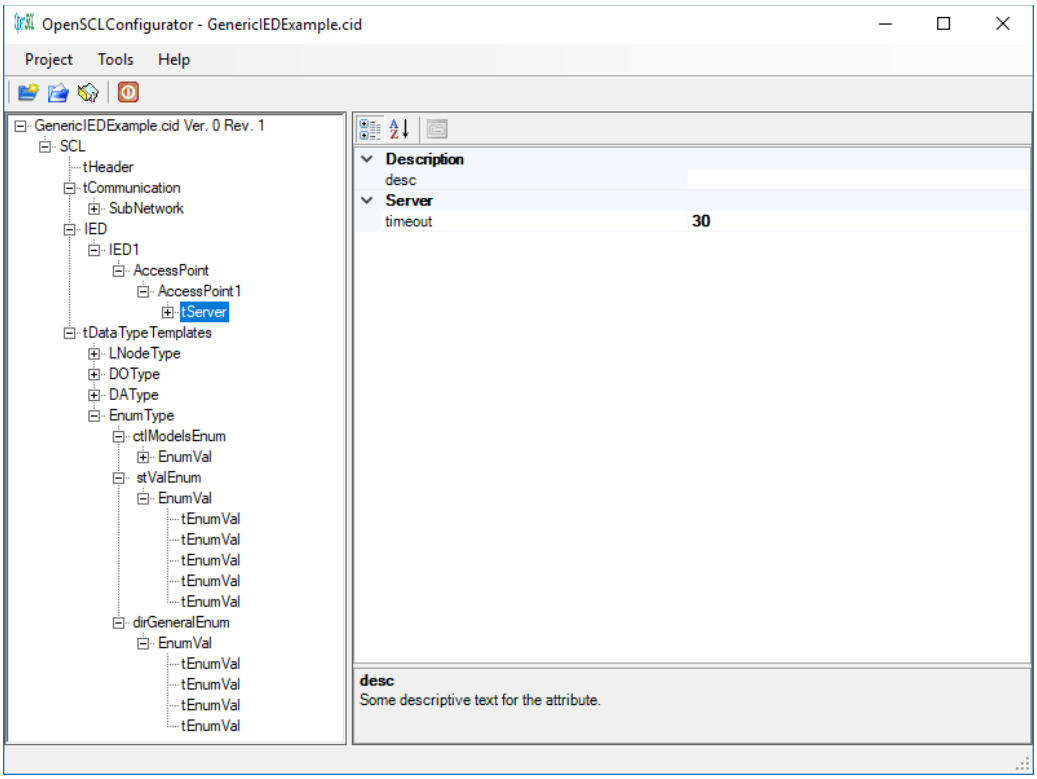


Figura 18: Herramienta de distribución libre usado para poder editar archivos SCL

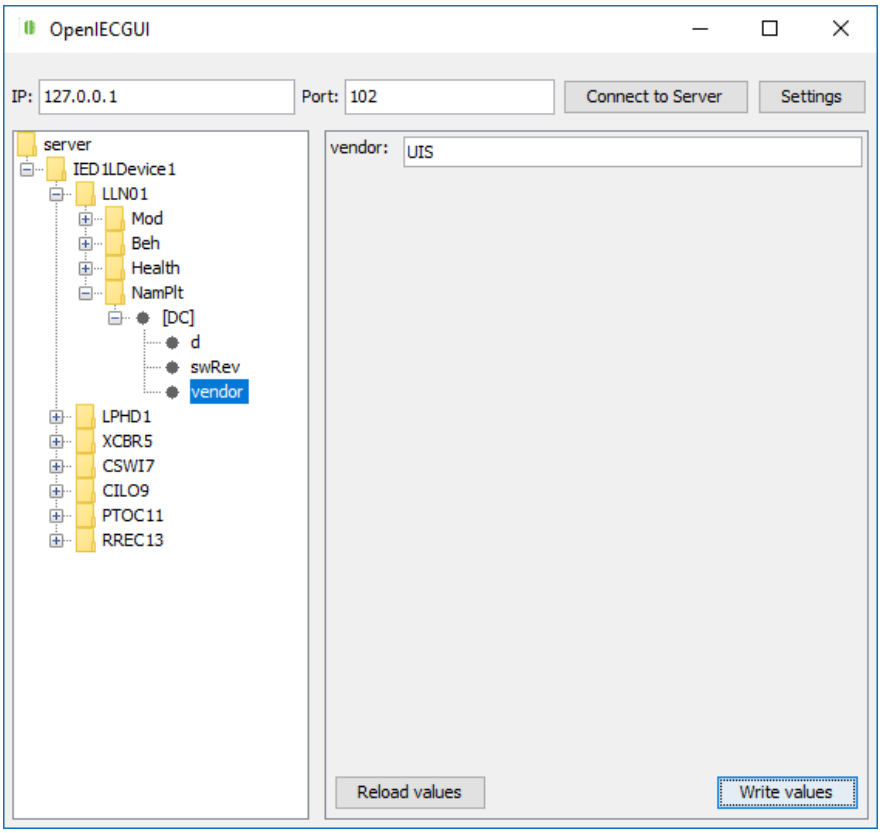


Figura 19: Navegando la estructura de datos del servidor y escribiendo nuevos datos

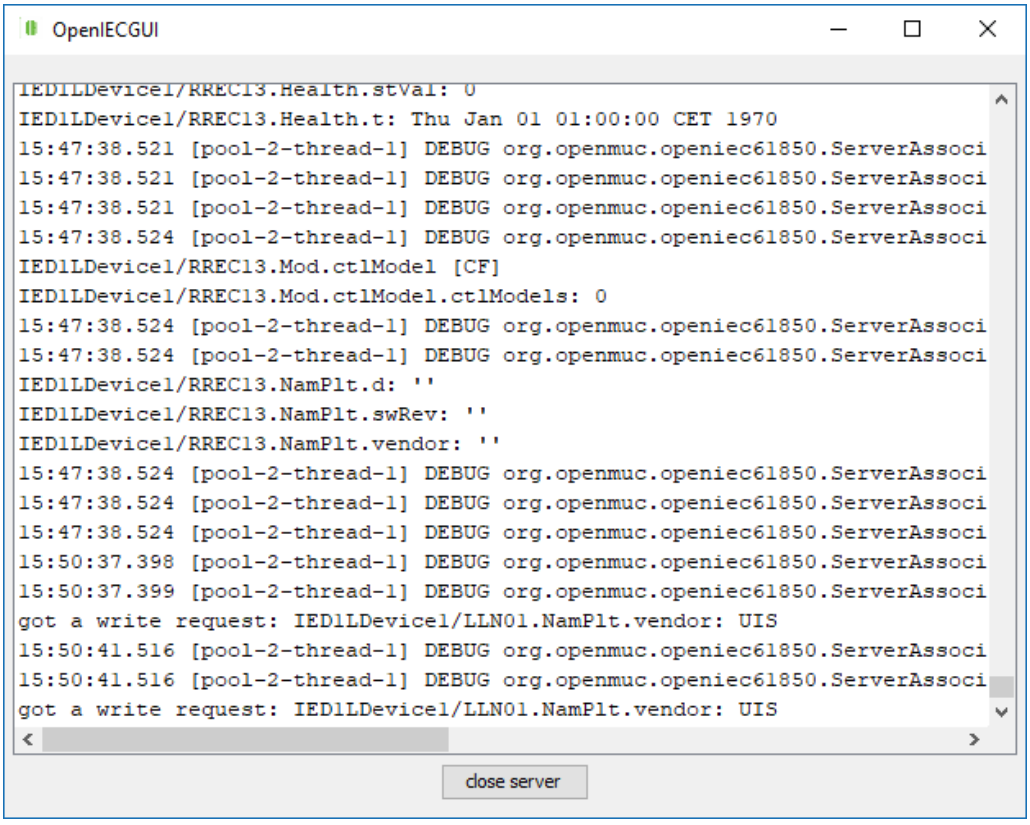


Figura 20: Servidor activo escuchando por conexiones y recibiendo solicitudes de escritura

4. Modelo y Simulación de comunicación de datos

Para implementar el software creado en los capítulos anteriores se generan los casos de uso en los que se puedan aprovechar las funcionalidades del mismo, debido a que la librería OpenIEC61850 se encuentra limitada en cuando a la implementación de funcionalidades del estándar IEC 61850 y a que este trabajo de grado se concentra en cómo se transmiten los datos en este tipo específico de software, los modelos propuestos no comprenderán una subestación completamente realizada, pero una porción de las partes que permiten observar intercambio de datos.

Como parte de la ingeniería normal en la práctica, descrita generalmente en el capítulo 4, del estándar IEC 61850, el diseño de un SAS para una subestación se da desde el nivel lógico y

funcional en base a los IDE que serán implementados y estos son integrados a la subestación teniendo en cuenta los requerimientos y situaciones específicos de la red eléctrica.

En el caso de una implementación software, una planeación completa incluye todos los modelos definidos para los archivos SCL descritos en el capítulo 6 del estándar IEC 61850, la lista completa de archivos usados, ICD, SSD, SCD, CID, IID y SED es usada en el proceso para llegar a descripción y configuración detallada de un dispositivo real proporcionado por un fabricante, el fabricante usualmente proporciona el ICD que solo describe las capacidades del dispositivo, es tarea del ingeniero entonces incluir el dispositivo en el diseño y descripción de subestación en otros archivos como SCD donde se especifican todas las funcionalidades de la subestación en cuestión.

Para fines de este trabajo se alimentará al software con archivos de configuración de IDE básicos que describan una funcionalidad específica para un dispositivo concreto que pueda cambiar sus valores en el tiempo para así poder capturar el envió de datos por medio de reportes a través de la red, de esta manera se simula una pequeña subestación con cierta cantidad de dispositivos que cumplen una función.

4.1 Modelo propuesto

Se plantea una serie de dispositivos representado con configuraciones diferentes en archivos ICD que se cargarán en el programa y seguirán la topología mostrada en la figura 21.

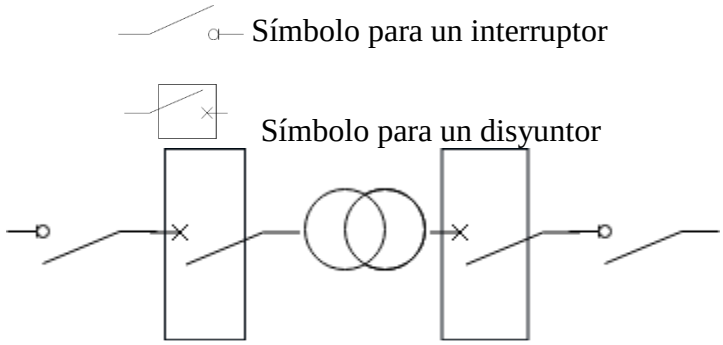


Figura 21: Transformador dentro de una línea de barra sencilla

Para representar los dispositivos dentro de una red propuesta cada dispositivo corresponderá a un nodo lógico para fines de sencillez, el transformador por ejemplo ejecutara la función definida por su nodo lógico LN0 para medir el voltaje, mientras interruptores tendrán un valor booleano para abrir o cerrar el circuito y el disyuntor tendrá funciones de monitoreo.

4.2 Recursos para el modelado de las comunicaciones en la subestación

La prueba se realiza de dos maneras, mediante una red física con 2 computadores portátiles en una red de área local privada, y mediante una red virtual, ejecutando máquinas virtuales que comparte una subred dentro del mismo adaptador, en ambos casos se espera que los datos se transmitan sobre los mismos protocolos debido a que las máquinas virtuales son agnósticas a los dispositivos virtuales.

Para fines de la simulación y la simplicidad otorgada por usar OpenIEC61850, una implementación de este tamaño en hardware físico se verá como lo demuestra la figura 22.

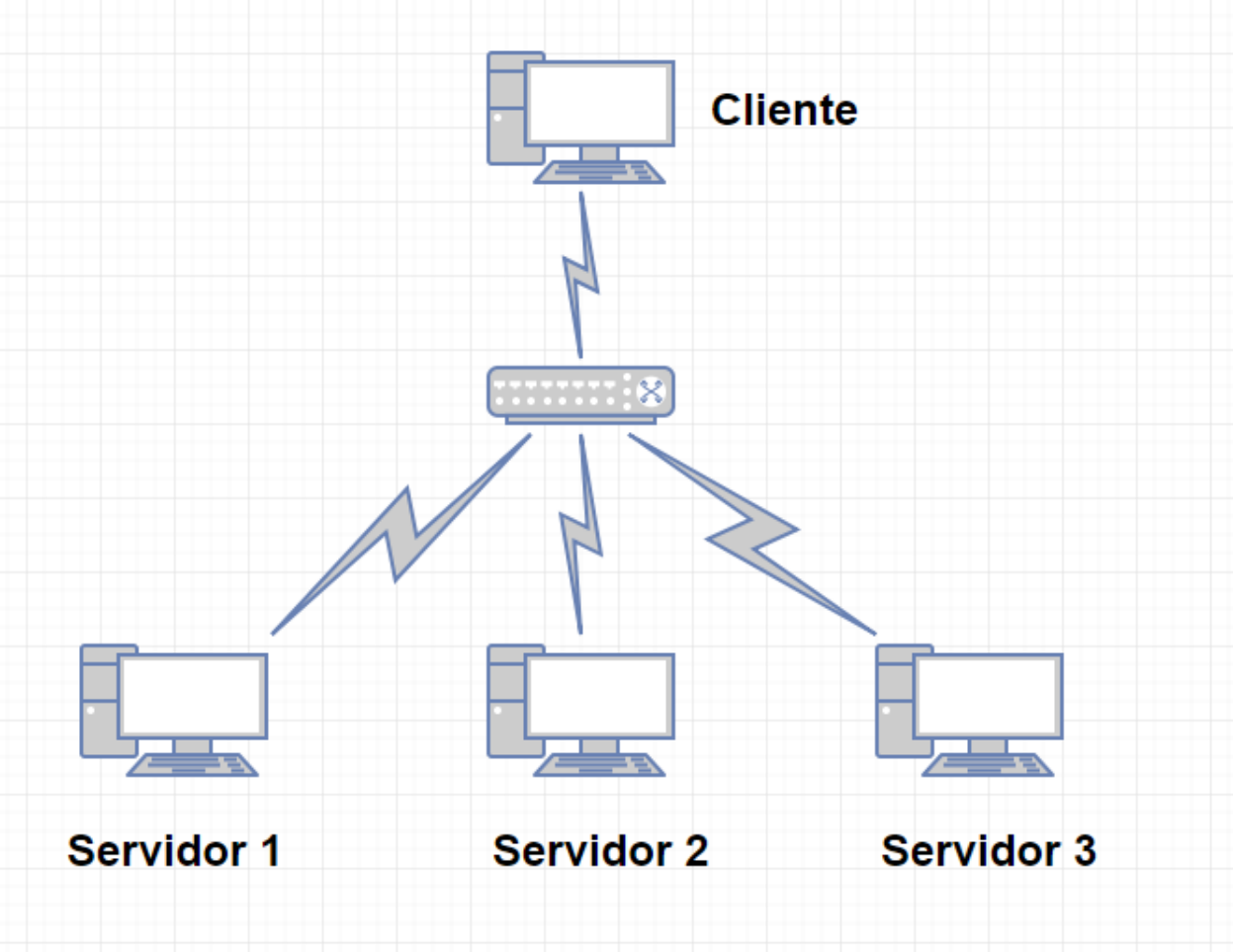


Figura 22: Topología física implementada

Donde cada servidor ejecutará una instancia de *ServerSap()* proporcionada con la librería OpenIEC61850 y estará cargada con un archivo ICD describiendo un dispositivo especifico y escuchando por solicitudes entrantes.

El cliente entonces se conectará a cada uno de los servidores para solicitar datos, simulando así una central de control que monitorea los dispositivos.

Todo esto se realiza dentro de una red de área local, los datos se mueven sobre un protocolo TCP/IP y una conexión Ethernet cableada, se espera observar la implementación MMS por parte de la librería y la lectura y escritura de datos correcta por parte del software, este tipo de comunicaciones difiere de las redes de datos residenciales y empresarial en las capas de aplicación

que implementan protocolos específicos para este fin o estándares que fueron descartados hace tiempo, por ejemplo en las capas menores durante sus primeras versiones MMS implementaba Token Ring, en el dominio de la subestación los dispositivos se controlan sobre la red local, a su vez la subestación es un nodo en una red de área amplia (WAN), que comprende toda la red eléctrica administrada por un operador.

En una subestación física la topología regular para el ejemplo previamente mostrado se podría ver representado como en la figura 23.

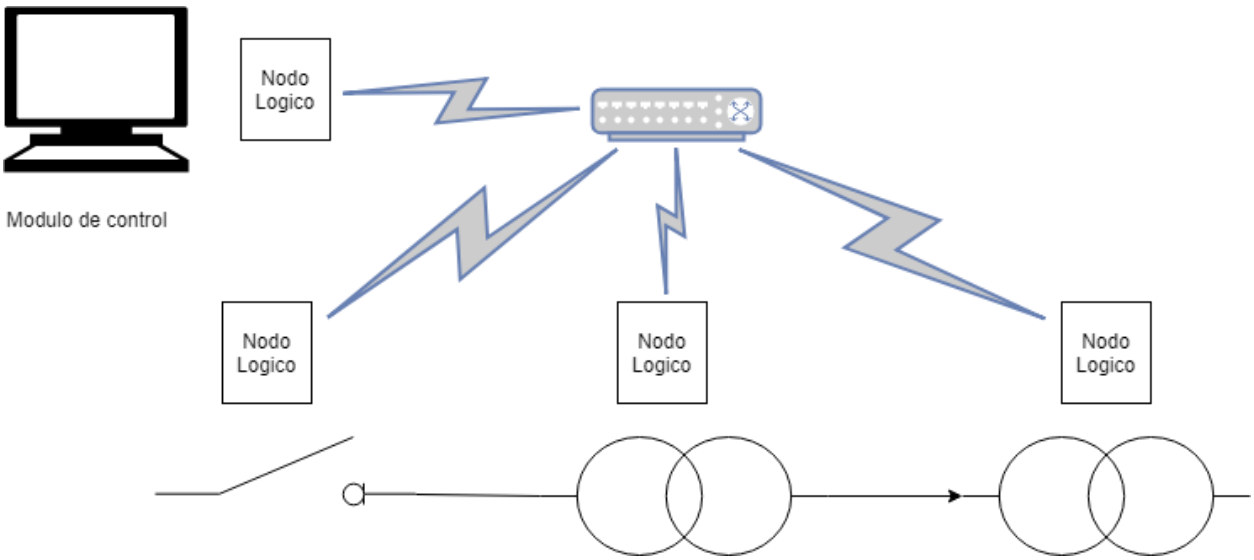


Figura 23: Los nodos lógicos son los que comunican sus datos en este caso

El software de captura de datos sobre protocolos en la red Wireshark será usado para monitorear la red, detectar las conexión y solicitudes realizada y analizar los datagramas capturados, del mismo modo se busca si es posible ver los datos de la capa de aplicación enviados ya sea por cliente o servidor como una brecha de seguridad en algún punto de la implementación.

Si regresamos a la figura 10 donde se muestran los protocolos que implementa MMS, buscaremos encontrar cuales de los protocolos diferentes a TCP/IP implementa la librería sobre este entorno.

4.3 Procedimiento

Iniciando los servidores se está usando la subred por defecto y el puerto 102 en cada uno de ellos, se carga el archivo *sample-model.icd* modificado para que muestre ciertos datos deseables como una frecuencia de reportes más baja, para poder capturar comunicaciones de manera más frecuente.

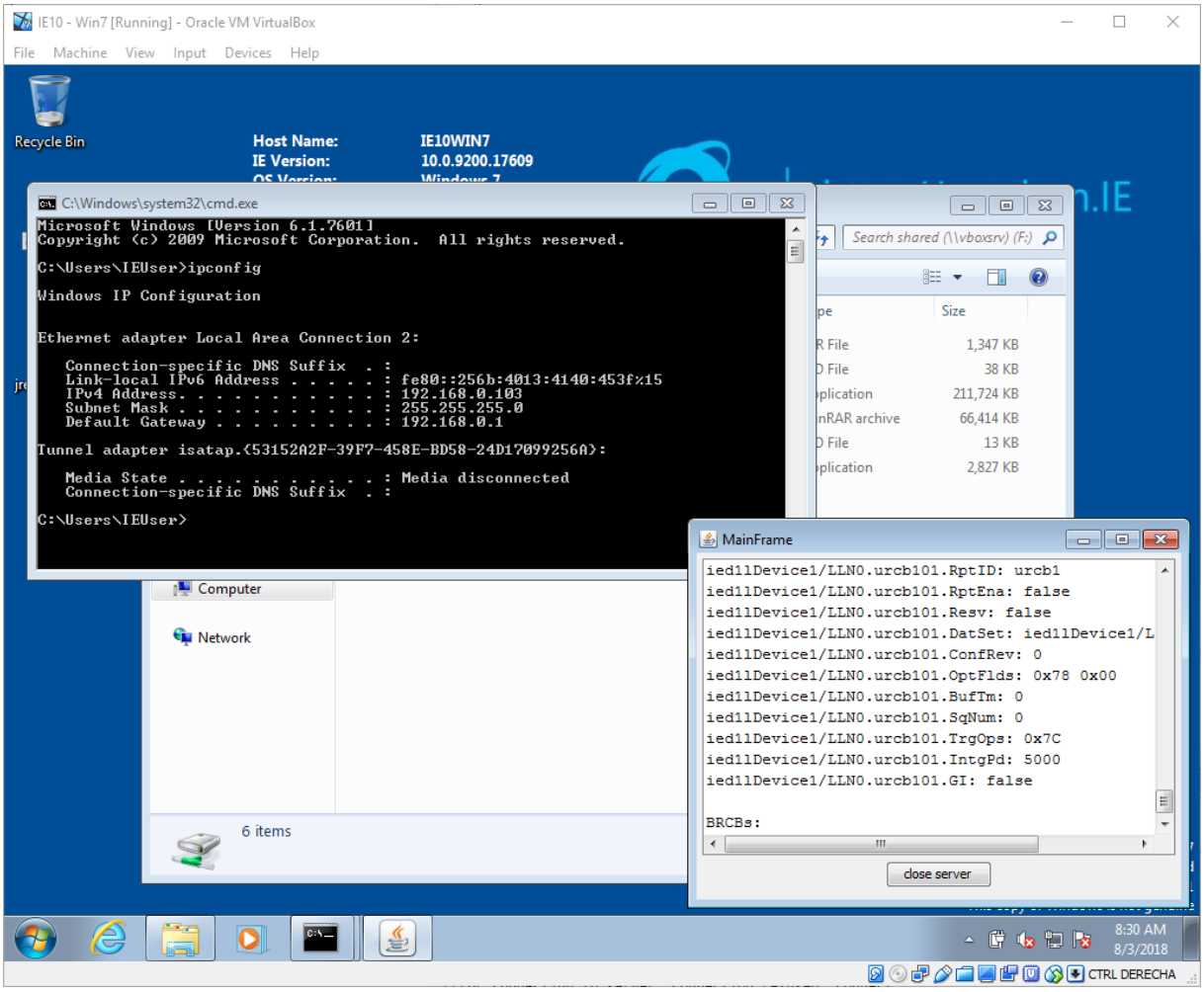


Figura 24: Máquina virtual ejecutando un servidor con su IP local.

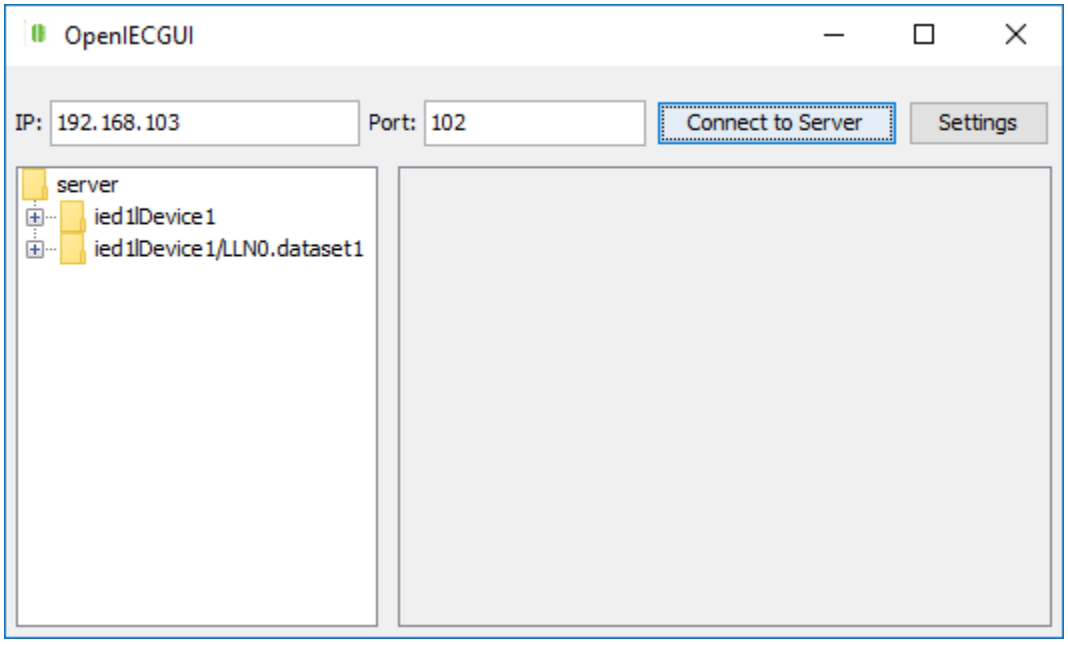


Figura 25: Conectándose a uno de los servidores usando el gui-client parte de la librería.

Aquí se ejecuta Wireshark para empezar a capturar las conexiones, una observación es que el filtro por el cual se muestran las conexiones es el protocolo de la cada de presentación, de agrega entonces la opción de solo mostrar las conexiones que involucren al servidor, esto para mejorar la visibilidad.

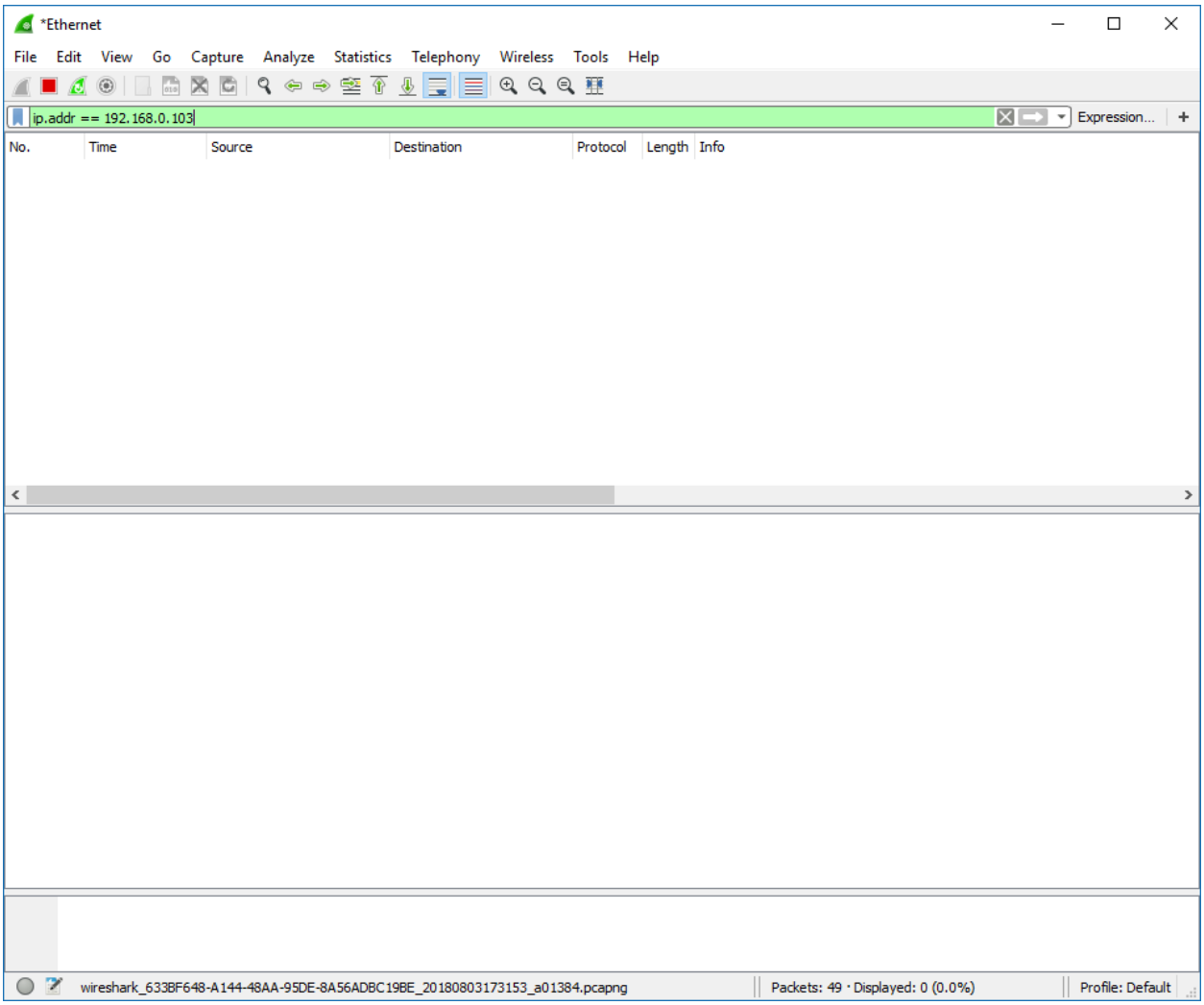


Figura 26: Filtro para las conexiones

Tan pronto como se conectan el cliente y el servidor en cuestión, esto es se formó una asociación en la API de la librería, se pueden observar el envío de datos cuando el cliente le solicita al servidor una copia de su modelo con todos los valores.

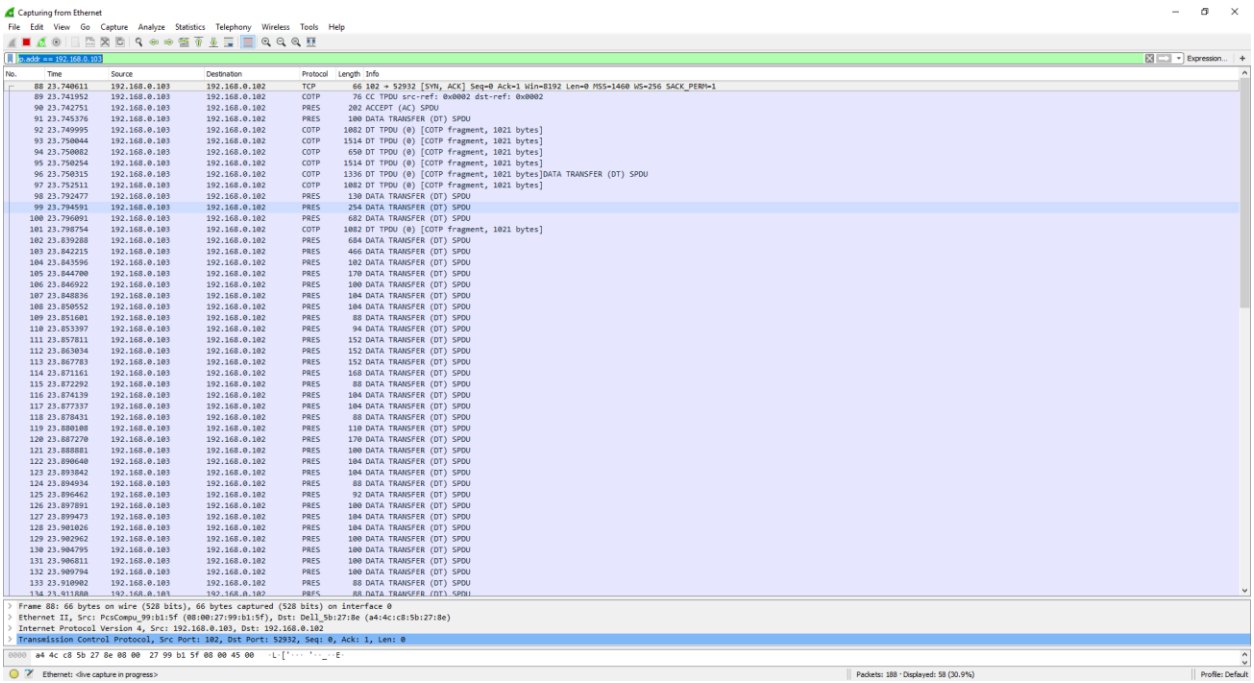


Figura 27: Servidor enviando datos

Aquí podemos apreciar la manera en la que se ejecuta la conexión, conforme lo dice la teoría, los datagramas se mueven sobre TCP/IP y luego pasan sobre las implementaciones de MMS dadas por los protocolos ISO 8073, ISO 8327-1, ISO 8823, otro elemento a resaltar el uso de TPKT, que simplemente encapsula todo lo que se va a transportar sobre TCP.

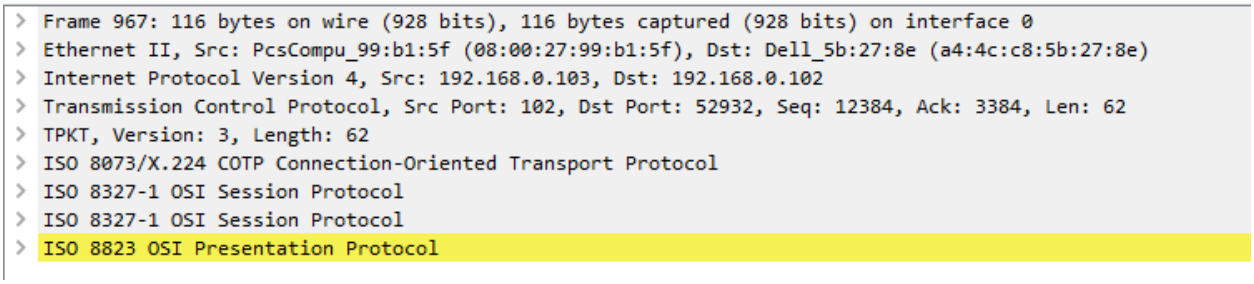


Figura 28: Capas de protocolos presentes cuando el servidor envía datos.

```
):34:00.804 [pool-4-thread-1] DEBUG org.openmuc.openiec61850.ServerAssociation - sub BDA is:ied1lDevice1/DSCH1.SchdAbsTm.va
):35:21.132 [pool-4-thread-1] DEBUG org.openmuc.openiec61850.ServerAssociation - Got a GetDataSetValues request.
):35:21.132 [pool-4-thread-1] DEBUG org.openmuc.openiec61850.ServerAssociation - Got a GetDataSetValues request.
):37:49.163 [pool-4-thread-1] DEBUG org.openmuc.openiec61850.ServerAssociation - Got a Write request
):37:49.163 [pool-4-thread-1] DEBUG org.openmuc.openiec61850.ServerAssociation - Got a Write request
):37:49.163 [pool-4-thread-1] DEBUG org.openmuc.openiec61850.ServerAssociation - Got a SetDataValues request.
):37:49.163 [pool-4-thread-1] DEBUG org.openmuc.openiec61850.ServerAssociation - Got a SetDataValues request.
```

Figura 29: Servidor recibe solicitudes para implementar los servicios, GetDataSetValue, Write y SetDataValues

Aunque se detecta un poco de retraso en comparación con pruebas previas se puede ver que el servidor procesa las solicitudes y actualiza los datos.

```
Got a GetDataValues request for node: ied1lDevice1/LLN0.urcb102.IntgPd: 2500
Got a GetDataValues request for node: ied1lDevice1/LLN0.urcb102.IntgPd: 2500
```

Wireshark · Packet 3533 · Ethernet

```
> Frame 3533: 88 bytes on wire (704 bits), 88 bytes captured (704 bits) on interface 0
> Ethernet II, Src: PcsCompu_99:b1:5f (08:00:27:99:b1:5f), Dst: Dell_5b:27:8e (a4:4c:c8:5b:27:8e)
> Internet Protocol Version 4, Src: 192.168.0.103, Dst: 192.168.0.102
> Transmission Control Protocol, Src Port: 102, Dst Port: 52932, Seq: 12475, Ack: 3538, Len: 33
> TPKT, Version: 3, Length: 33
> ISO 8073/X.224 COTP Connection-Oriented Transport Protocol
> ISO 8327-1 OSI Session Protocol
> ISO 8327-1 OSI Session Protocol
v ISO 8823 OSI Presentation Protocol
  v user-data: fully-encoded-data (1)
    v fully-encoded-data: 1 item
      v PDV-list
          presentation-context-identifier: 3
        v presentation-data-values: single-ASN1-type (0)
          v Dissector is not available
            v [Expert Info (Warning/Undecoded): Dissector is not available]
                [Dissector is not available]
                [Severity level: Warning]
                [Group: Undecoded]
> VSS-Monitoring ethernet trailer, Source Port: 0
```

```
0000  a4 4c c8 5b 27 8e 08 00  27 99 b1 5f 08 00 45 00   ·L·['··· '··_··E·
0010  00 49 01 21 40 00 80 06  77 70 c0 a8 00 67 c0 a8   ·I·!@··· wp···g··
0020  00 66 00 66 ce c4 b7 08  bf 1e 9f 6e c7 6e 50 18   ·f·f···· ···n·nP·
0030  00 fe 89 c7 00 00 03 00  00 21 02 f0 80 01 00 01   ········ ·!······
0040  00 61 14 30 12 02 01 03  a0 0d a1 0b 02 01 34 a4   ·a·0···· ······4·
0050  06 a1 04 86 02 09 c4 00                           ········
```

Figura 30: Datagrama de la capa de presentación donde los datos del usuario están encriptados

La figura 30 mostraba que de hecho existe algo de seguridad heredada en el protocolo de comunicación pese a que no es explícitamente descrito durante los estándares, este el datagrama que carga todos los intercambios de datos entre clientes y servidores.

Del mismo modo se prueban diferentes configuraciones de ICD con distintos tipos de datos y en la figura 31 se muestra lo más resaltado.

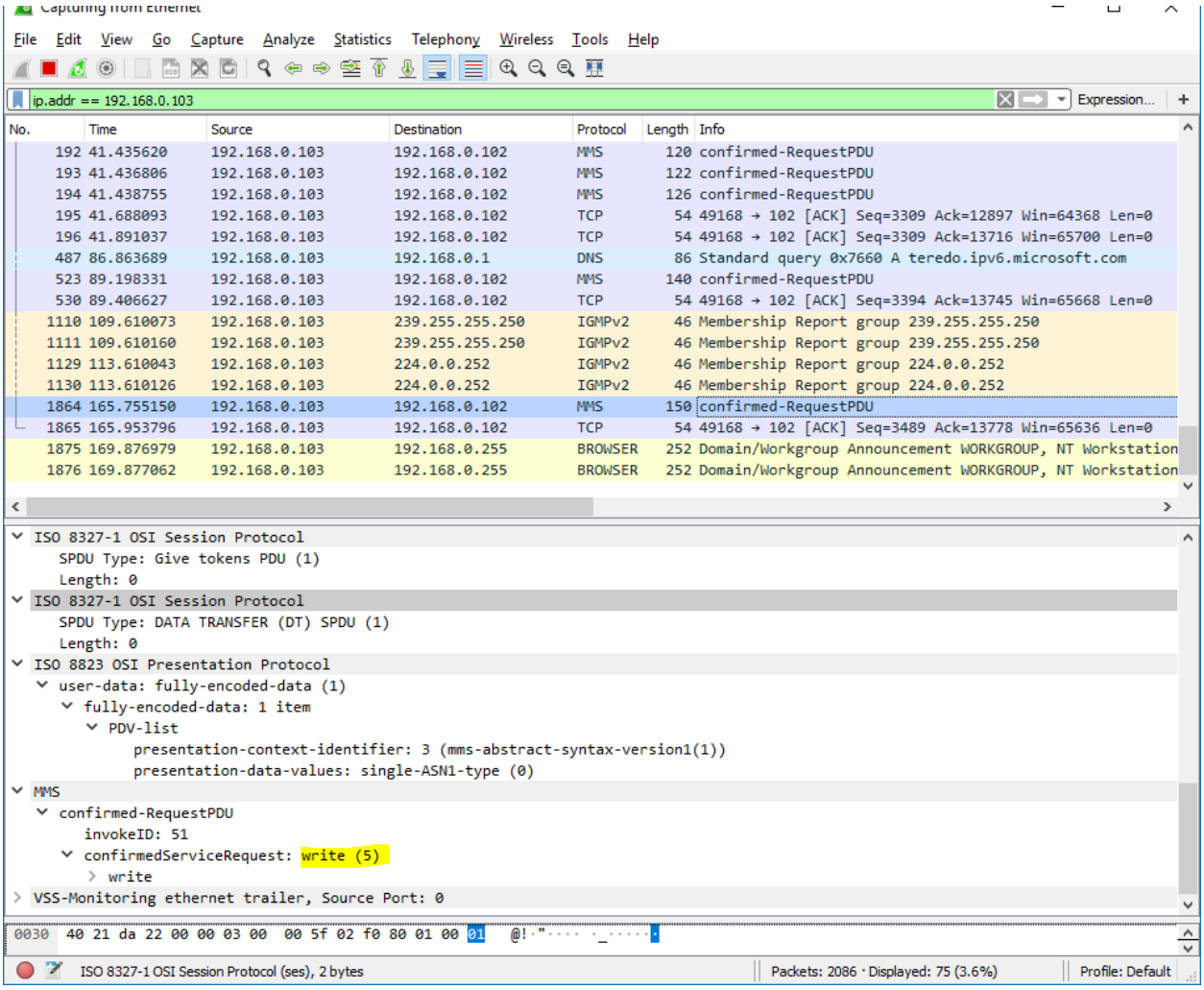


Figura 31: Captura desde el servidor muestra los valores y solicitudes efectuadas sobre MMS.

Si se está capturando el tráfico desde el servidor y existe más de un cliente conectado a la vez, de hecho, se pueden observar los mensajes MMS junto con los valores de los datos y el servicio de la solicitud efectuada.

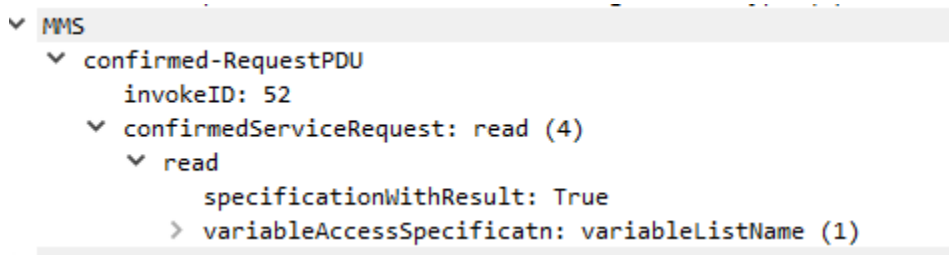


Figura 32: Solicitud y valor visibles en él datagrama.

Cabe resaltar que después de un poco de investigación se encontró que la presencia de VSS-Monitoring ethernet tráiler, es debido a un bug de Wireshark, por lo tanto, es ignorado.

5. Resultados

Después de revisar el proceso de las conexiones, cuando se envía al servidor un cambio de datos para un valor dentro de su nodo lógico LN, esto se ejecuta como cualquier otra llamada a un servicio, el cliente entonces envía una solicitud para obtener los datos de nuevo, esto fue para comprobar que los datos si están siendo actualización, aquí es cuando en este ambiente de red se nota un retraso en la velocidad del servidor, diferente a cuando se hicieron algunas pruebas con archivos vacíos y en el mismo computador los tiempo eran casi inmediatos, se efectuó una prueba de ping hacia el servidor, y no hay diferencia notable.

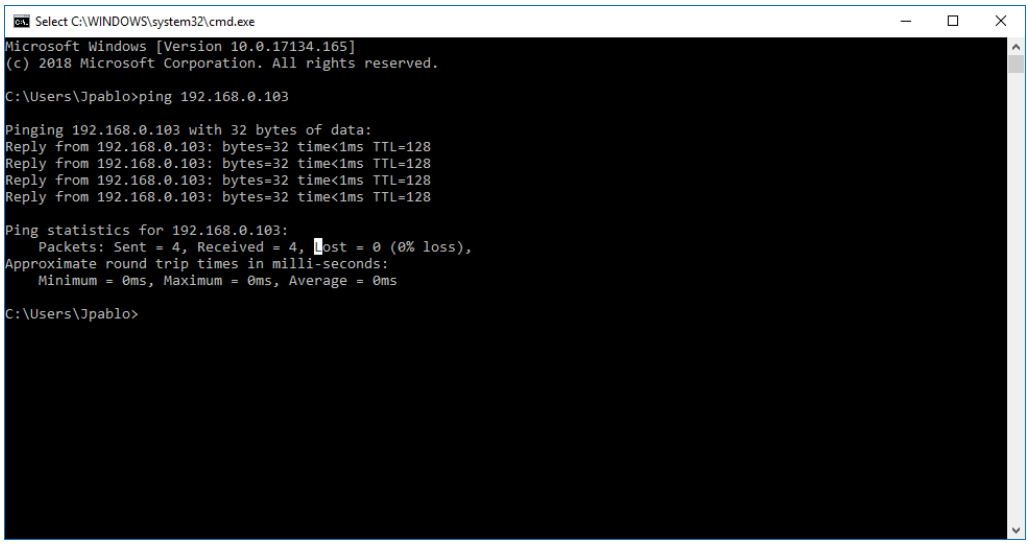


Figura 33: Ping desde el cliente a el servidor

Esto quiere decir que los retrasos probablemente son una combinación de procesamiento por parte del servidor para atender a las solicitudes y el tráfico regular de la red, por la que viajan todos los valores del servidor incluso si no han cambiado, esto es por diseño, debido a que en la práctica se implementara en un sistema de alta disponibilidad y el monitoreo el crucial.

5.1 Seguridad

Como se observa en la figura 30 cuando se captura el envío de datos en el cliente, los datos dentro del datagrama de la presentación están encriptados, después de investigación acerca de este, las capas de sesión y de presentación efectúan tarea de encriptación básicas y la capa de sesión es la encargada de seguridad, pero parece que esto solo está presente cuando se monitorea un cliente, figura 31, ya que es posible acceder no solo a los datos, si no saber qué operación se estaba realizando.

Esto es una falla de seguridad, pero la gravedad depende del contexto en el que se maneje la implementación, desde la perspectiva software la seguridad solo estaba dada por las librerías y APIs implementadas, las cuales no otorgan gran protección a no ser que se esté buscando explícitamente encriptar datos y denegar accesos no autorizados, el cual no era el caso.

Revisado el estándar IEC 61850 se encontró que no contempla directrices ni aspectos específicos para la seguridad en las comunicaciones a nivel de software. Igualmente, la librería tampoco contempla esos aspectos, por lo tanto, las soluciones prácticas deberán tener esto en cuenta junto con procesos a nivel de proyecto para garantizar seguridad en un sistema tan sensible como lo es la red eléctrica.

5.2 Estado de la librería OpenIEC61850

La figura 34 muestra las definiciones de servicios para cada clase definida en el capítulo 7 de la serie de estándares, en contraste OpenIEC61850 en su versión más reciente implementa los siguientes servicios:

- Todos los servicios de tipo GetDirectory/GetDataDefinition
- GetDataValues/SetDataValues
- Todos los de escribir datos
- Unbuffered reporting
- Buffered reporting (solo cliente)
- Todos los servicios de Control

Lo cual es una implementación amplia, y más que suficiente parar cubrir los casos de uso básicos, pero si se es necesario alguna funcionalidad o servicio en específico que no esté

implementado eso dependerá el contexto especifico, por ejemplo, en el transcurso de este trabajo solo se usaron los servicios relaciones con GetDataValues/SetDataValues específicamente.

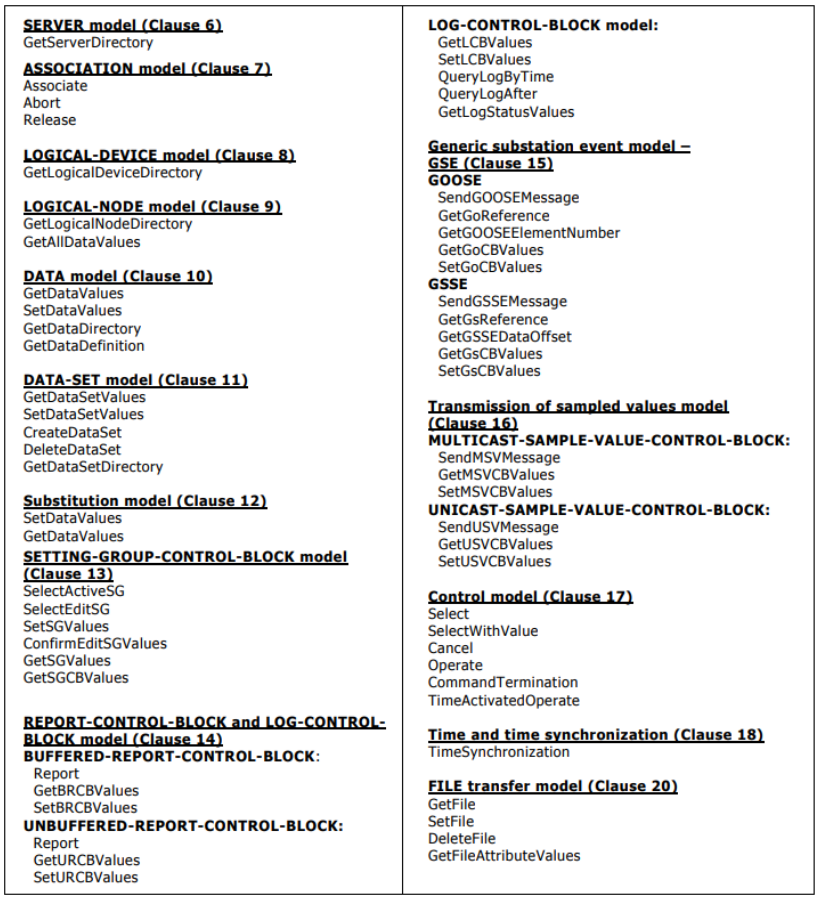


Figura 34: Lista completa de clases y sus servicios en IEC61850, Fuente: IEC61850-7-2

6. Conclusiones

Después de investigar la literatura con respecto a los sistemas software que implementan normas como IEC 61850 se encontró que existe una gran cantidad de herramientas licenciadas, y estas junto con el conocimiento de las implementaciones tienden a estar reservadas solo para sectores específicos de la industria, no existe mucho código abierto que explore estos temas y su aprendizaje puede ser costoso.

El desarrollo de herramientas software que sean diseñadas específicamente para dar soporte al diseño y desarrollo de sistemas complejos como lo son los SAS, depende mucho de conocimiento técnico de muy bajo nivel, la abstracción proporcionada por la implementación de APIs que se adhieran a los estándares facilita capacitación y progreso en la industria por fuera del sector privado.

La librería OpenIEC61850 proporciona herramientas que permiten efectuar experimentación con sistemas de control y dispositivos variados, provee funcionalidades básicas y sin embargo sería deseable que proporcionara herramientas específicas para definir IEDs, como por ejemplo configuradores, aun así, la iniciativa de mantener una librería de código abierto permite que el usuario implemente sus propias soluciones del mismo modo que las funcionalidades mejoran.

El desempeño y seguridad de la librería y al mismo tiempo del software desarrollado que la implementó, estuvo perjudicado por las variables del entorno en el que fueron ejecutados, como todo sistema que se ejecute en un ambiente de red, si las capas de nivel más bajo presentan bajo rendimiento y brechas en la seguridad, el impacto será visto en las capas superiores, aunque la seguridad en este contexto no es una prioridad debido a que en la práctica las subestaciones eléctricas estarán escudadas de medidas de seguridad físicas y lógicas más allá de los protocolos que maneja.

7. Recomendaciones

En la ejecución del proyecto se encontró que posterior a la existencia de la IEC 61850 aparece la norma IEC 61351 que pretende desarrollar los aspectos de seguridad complementarios

para la IEC 61850, sería beneficioso abordar este tema desde una perspectiva de seguridad más fuerte, implementando e investigando este estándar.

Este trabajo fue una introducción al aspecto software de la implementación de este tipo de normas, previamente se había efectuado un trabajo con una perspectiva eléctrica que uso las mismas herramientas, es recomendable seguir profundizando estos temas que son interdisciplinarios debido al valor académico que proveen.

Del mismo modo es posible seguir el desarrollo de herramientas que implementen la librería o cualquiera de las otras soluciones de la suite OpenMUC, como el software resultado de este trabajo se usó para ciertos casos de uso, se podrían proponer aplicaciones y mejoras desde la perspectiva logia y estética, del mismo modo se pueden crear herramientas independientes que puedan ser usadas por la comunidad universitaria para abordar el tema.

Referencias Bibliográficas

Acevedo, F, Calderon, L, y González, J. (2013). *Modelado y simulación de subestaciones: Análisis indirecto de las variables eléctricas en estado estable y transitorio* (tesis de pregrado). Universidad Industrial de Santander, Bucaramanga, Colombia.

Fraunhoufer ISE. (2018). *OpenMUC.* Recuperado de https://www.openmuc.org/

Hunt, R. (2016). *IEC61850 Overview and Application.* Recuperado de http://sites.ieee.org/houston/

International Electrotechnical Commission. (2003-2017). *IEC 61850:2018 SER Communication networks and systems for power utility automation - ALL PARTS.* Recuperate de http://www.iec.ch/smartgrid/standards/

International Electrotechnical Commission. (2018). *Electropedia: The World's Online Electrotechnical Vocabulary*. Recuperado de http://www.electropedia.org/iev/iev.nsf/6d6bdd8667c378f7c12581fa003d80e7?OpenForm

International Telecommunication Union. (2002). *SERIES X: DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS.* Recuperado de https://www.itu.int/en/Pages/default.aspx

LX Innovations. (2011). *Intelligence Embedded Power Grids.* Recuperado de https://lxinnovations.wordpress.com/

Pollock, Z. (2017). *DERs and Grid Optimization – The Next Phase of the Grid Modernization Journey.* Recuperado de https://www.renewableenergyworld.com/index.html

Spell, B. (2015). *Pro Java 8 Programming.* 3ra Ed. Apress

The Apache Software Foundation. (2004). *Apache License 2.0.* Recuperado de https://www.apache.org/licenses/LICENSE-2.0

United Stated Department of Agriculture. (2001). *Design guide for rural substations.* Recuperado de https://www.rd.usda.gov/

University of Calgary. (2017). *Energy Education.* Recuperado de https://energyeducation.ca/encyclopedia/