

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA



**DISEÑO E IMPLEMENTACIÓN DE UN LABORATORIO VIRTUAL
BASADO EN UNA METODOLOGÍA PARA EL ESTUDIO DE
PRUEBAS DE CONFIGURACIÓN DE PROTOCOLOS EN EQUIPOS
VIRTUALES BAJO EL ESTÁNDAR IEC61850**

PRESENTADO POR:

JOEL ARNOLDO LÓPEZ ALVARADO

MIGUEL ENRIQUE VÁSQUEZ CERÓN

PARA OPTAR AL TÍTULO DE:

INGENIERO ELECTRICISTA

CIUDAD UNIVERSITARIA, FEBRERO 2021

UNIVERSIDAD DE EL SALVADOR

RECTOR:

MSC. ROGER ARMANDO ARIAS ALVARADO

SECRETARIA GENERAL:

MSC. FRANCISCO ANTONIO ALARCÓN SANDOVAL

FACULTAD DE INGENIERÍA Y ARQUITECTURA

DECANO:

PHD. EDGAR ARMANDO PEÑA FIGUEROA

SECRETARIO:

ING. JULIO ALBERTO PORTILLO

ESCUELA DE INGENIERÍA ELÉCTRICA

DIRECTOR:

ING. ARMANDO MARTÍNEZ CALDERÓN

UNIVERSIDAD DE EL SALVADOR
FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA DE INGENIERÍA ELÉCTRICA

Trabajo de Graduación previo a la opción al Grado de:

INGENIERO ELECTRICISTA

Título:

**DISEÑO E IMPLEMENTACIÓN DE UN LABORATORIO VIRTUAL
BASADO EN UNA METODOLOGÍA PARA EL ESTUDIO DE
PRUEBAS DE CONFIGURACIÓN DE PROTOCOLOS EN EQUIPOS
VIRTUALES BAJO EL ESTÁNDAR IEC61850**

Presentado por:

**JOEL ARNOLDO LOPEZ ALVARADO
MIGUEL ENRIQUE VASQUEZ CERON**

Trabajo de Graduación Aprobado por:

Docente Asesor:

ING. HUGO MIGUEL COLATO RODRÍGUEZ

SAN SALVADOR, FEBRERO DE 2020

Trabajo de Graduación Aprobado por:

Docente Asesor:

A handwritten signature in black ink, appearing to read 'Hugo M. Colato R.', written in a cursive style.

ING. HUGO MIGUEL COLATO RODRÍGUEZ

NOTA Y DEFENSA FINAL

En esta fecha, Lunes 14 de diciembre de 2020, en la Sala de Lectura de la Escuela de Ingeniería Eléctrica, a las 5:00 p.m. horas, en presencia de las siguientes autoridades de la Escuela de Ingeniería Eléctrica de la Universidad de El Salvador:

1. Ing. Armando Martínez Calderón
Director


Firma

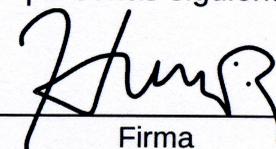

Firma



2. MSc. José Wilber Calderón Urrutia
Secretario

Y, con el Honorable Jurado de Evaluación integrado por las personas siguientes:

- MSC. HUGO MIGUEL COLATO RODRIGUEZ
(Docente Asesor)


Firma

- ING. NUMA POMPILO JIMENEZ CORTEZ


Firma

- MSC. LUIS ROBERTO CHEVEZ PAZ


Firma

Se efectuó la defensa final reglamentaria del Trabajo de Graduación:

DISEÑO E IMPLEMENTACION DE UN LABORATORIO VIRTUAL BASADO EN UNA METODOLOGIA PARA EL ESTUDIO DE PRUEBAS DE CONFIGURACION DE PROTOCOLOS EN EQUIPOS VIRTUALES BAJO EL ESTANDAR IEC61850

A cargo de los Bachilleres:

- JOEL ARNOLDO LÓPEZ ALVARADO

- MIGUEL ENRIQUE VÁSQUEZ CERÓN

Habiendo obtenido en el presente Trabajo una nota promedio de la defensa final: 9.2

(nueve punto dos)

AGRADECIMIENTOS

En este trabajo de graduación dejo plasmados mis más sinceros agradecimientos principalmente a Dios Padre todopoderoso y mi Madre celestial por la inteligencia, la gracia y la sabiduría que me han regalado y sobre todo por su fidelidad a cada momento.

A mis padres Rosa Cerón y Miguel Vásquez dos seres ejemplares con los que estoy eternamente agradecido, por cada consejo, regaño y amor brindado, por guiarme siempre por el camino del bien, por su gran esfuerzo y sacrificio, quienes hicieron posible que yo llegara a culminar mi carrera universitaria.

A mi amada Abigail Montano quien ha estado conmigo desde que inicié mi carrera y me ha brindado su amor y apoyo incondicional en los momentos buenos y malos, quien siempre ha confiado y ha tenido fe en mí en que puedo llegar y lograr mis metas.

A mi gran y hermosa familia que me han brindado su apoyo de diferentes maneras a lo largo de mi trayectoria y vida entera, especialmente a mis hermanas Tatiana, Jacqueline y Douglas que siempre me han apoyado en cada paso importante de mi vida.

A cada catedrático que fue responsable de mi formación profesional y quienes me guiaron a través del camino y conocimiento de la ingeniería, en especial al Ing. Hugo Colato quien fue mi mentor, por la enorme labor y ayuda brindada que hicieron posible la finalización de este proyecto.

A mi compañero y amigo de tesis Joel López, por su confianza y esfuerzo para que pudiéramos finalizar este proyecto que inicio de la nada, que a pesar de todas las dificultades que se nos presentaron siempre se mostró firme y seguro, siempre buscó la manera de salir adelante.

A mis amigos Kevin Vargas y Javier Mejía dos personas que conozco desde hace mucho tiempo, que a pesar de los altos y bajos que vivimos en la universidad siempre mostraron su apoyo y amistad sincera.

A mis amigos Jorge Rivera, Alberto Cano, Harold Romero y Fidel Castro, no puedo dedicaros un párrafo a cada uno, pero os tengo mucho que agradecer, siempre fueron mi equipo con los que cursé las materias más difíciles de la universidad, quienes gracias a sus esfuerzo y dedicación siempre logramos salir adelante a pesar de cualquier dificultad, y a todos aquellos amigos que hice a lo largo de mi carrera, gracias por su amistad.

Sin más, queda aquí plasmado mi nombre y este proyecto de grado como fiel testimonio de mi paso por las aulas de mi querida alma mater, mudos testigos de mi esfuerzo y sacrificio que viví en mi larga carrera de ingeniería eléctrica.

MIGUEL ENRIQUE VÁSQUEZ CERÓN

AGRADECIMIENTOS

Agradecer en primer lugar a Dios Jehová por darme fortaleza y poder ver su obrar en los momentos más difíciles y donde más lo necesité.

Agradezco infinitamente a mis padres por darme la oportunidad de estudiar y apoyarme para poder lograr la finalización de mi carrera, agradezco sus concejos y paciencia para estar siempre dispuesto a ayudarme en todo lo que pudieran.

A mi familia por no dudar ni un segundo en brindarme su mano cuando la necesité, mis hermanas Xenia y Eunice, a mi tío Balmore por haberme dedicado recursos, tiempo y sus conocimientos.

A mis compañeros de carrera y amigos Alonso, Laura y William, con quienes compartí momentos buenos y malos pero que a pesar de las dificultades siempre estuvieron ahí para brindarme su ayuda, sin la cual me hubiera sido mucho más difícil finalizar mi meta, también agradecer a las amistades que hice a lo largo de la carrera a quienes aprecio mucho y que llevare siempre en mi memoria.

A mi amigo y compañero de tesis Miguel Cerón, con el cual emprendimos este proyecto y a pesar de las dificultades nunca se rindió en seguir y se preocupó por dejar un trabajo bien hecho y que tuviera impacto en el país.

A los excelentes docentes que me brindaron su conocimiento y me formaron como profesional, agradezco sus concejos y esfuerzo por dar lo mejor a su alcancé por una educación de calidad, en especial agradezco a mi asesor de tesis Ing. Hugo Colato por brindarnos su tiempo aun en tiempos muy difíciles y que nos impulsó a continuar con este trabajo.

JOEL ARNOLDO LOPEZ ALVARADO

CONTENIDO

Contenido de tablas.....	i
Contenido de figuras.....	i
Glosario.....	v
Introducción.....	1
Justificación.....	2
CAPÍTULO I.....	3
1.0 Introducción.....	4
1.1 El estándar IEC 61850.....	4
1.1.1 Estructura de la norma.....	5
1.1.2 Arquitectura de una subestación con el estándar IEC61850.....	8
1.1.3 Modelo de datos.....	11
1.1.4 Nodos Lógicos.....	13
1.1.5 Clases de Datos Comunes (CDC).....	21
1.1.6 Restricciones Funcionales (FC).....	22
1.1.7 Servicios de comunicación.....	23
1.1.8 Data sets o Conjuntos de Datos.....	23
1.1.9 Reportes de Bloques de Control (URCB/BRCB).....	25
1.1.10 Lenguaje SCL.....	28
1.2 Equipos y software en una infraestructura de simulación/emulación en un laboratorio de aprendizaje de la norma IEC61850.....	31
1.2.1 61850 ICD EDITOR (Editor de archivos ICD).....	31
1.2.2 ReLab OPC Console (cliente).....	36
1.2.3 IEDexplorer (Cliente).....	41
1.2.4 Goose Sender/Receiver (Suscriptor/Publicador GOOSE).....	47
1.2.5 Infotech 61850 Relay (Dispositivo virtual).....	50
1.2.6 Simulator Bay (Dispositivo virtual).....	52
1.2.7 Wireshark (Analizador de red).....	54
1.2.8 Elipse Power (SCADA).....	57
1.2.9 VNCViewer (Cliente VNC).....	63

1.2.10 Libiec61850 (librería servidor/cliente IEC61850)	64
1.2.11 Librería libmodbus-3.1.6	69
1.2.12 Librería WiringPi.....	70
1.2.13 Sistema de Protección SEL 351	71
1.2.14 Raspberry PI 3B	73
1.2.15 Switch de comunicación	74
CAPÍTULO II.....	76
2.0 Introducción.....	77
2.1 CONFIGURACIÓN DE RELÉS SIMULADOS	78
2.1.1 Relé INFOTECH	78
2.1.2 Relé Simulator Bay Axon Group	86
2.2 CONFIGURACIÓN DE RELÉS EMULADOS	92
2.2.1 Equipo IEDRaspberry y sus componentes	92
2.2.1.1 Interfaz física	93
2.2.1.2 Sistema operativo	94
2.2.1.3 Aplicación para servidor IEC61850	94
2.2.2 Gateway IEC61850/Modbus	101
2.2.2.1 Interfaz física	101
2.2.2.2 Sistema operativo	103
2.3 CONFIGURACIÓN DEL POWERELIPSE SCADA.....	110
CAPÍTULO III.....	126
3.0 Introducción.....	127
3.1 Escenario 1, esquema de actuación de protecciones mecánicas de un transformador de potencia.....	128
3.1.1 Descripción y Funcionamiento.....	128
3.1.2 Esquema de protecciones.....	129
3.1.3 Configuración de los equipos	129
3.1.4 Procedimiento descrito en pasos para realizar la prueba.....	129

3.1.5 Evento #1, actuación de protecciones mecánicas del transformador de potencia	132
3.1.6 Evento #2, Restablecimiento de condiciones iniciales después de una falla.....	134
3.2 Escenario 2, disparo de barra acelerada (protección por sobre corriente instantánea)	135
3.2.1 Descripción y Funcionamiento.....	135
3.2.2 Esquema de protecciones.....	136
3.2.3 Configuración de los Equipos.	136
3.2.4 Procedimiento en pasos para realizar la prueba:	137
3.2.5 Evento #1, falla en el alimentador de salida.....	140
3.2.6 Evento #2, falla en la barra.....	143
3.3 Escenario 3, simulación de un SAS en niveles de estación y bahía	148
3.3.2 Procedimiento descrito en pasos para realizar la prueba.	150
3.3.3 Evento #1, protecciones mecánicas de un transformador de potencia	152
3.3.4 Evento #2, protección de barra acelerada.	155
CONCLUSIONES.....	160
RECOMENDACIONES.....	161
REFERENCIAS.	162
ANEXOS	163
Anexo A. Código básico servidor IEC61850.....	163
Anexo B. Código IEDRASPBERRY	165
Anexo C. Código convertidor IEC61850/Modbus.....	170
Anexo D. Instalación Raspbian desde S.O Windows.....	176
Anexo E. Software, simuladores y equipos que se entregan para uso y estudio del estándar a la Escuela de Ingeniería Eléctrica	177
Anexo F. Contenido del CD	178

Contenido de tablas

Tabla 1. Funciones de nodos lógicos.	19
Tabla 2. CDC definidas en el estándar IEC61850.....	21
Tabla 3. Restricciones Funcionales de IEC61850.....	22
Tabla 4. Tags entradas digitales IEDRASPBerry.	98
Tabla 5. Salidas digitales IEDRASPBerry.....	99
Tabla 6. Configuración puerto 5 relé SEL 351.	102
Tabla 7. Tags para la comunicación de SCADA.....	112
Tabla 8. Tags asociados a los elementos del modelo eléctrico.	116

Contenido de figuras

Figura 1. Estructura del estándar IEC 61850.	8
Figura 2. Arquitectura de un SAS en IEC61850.....	9
Figura 3. Interfases de comunicación.....	10
Figura 4. Modelo de datos para virtualización de dispositivos del estándar IEC61850.....	12
Figura 5. Concepto del modelo de datos.	13
Figura 6. Nodo Lógico del Interruptor de Potencia.....	20
Figura 7. Ejemplo de modelado de datos.	22
Figura 8. Dataset de posición de un XCBR.....	24
Figura 9. Bloque de transmisión de datos.	25
Figura 10. Modelo de transmisión de Datos Reportes con búfer y sin búfer.	25
Figura 11. Descripción general de los mensajes y servicios de IEC61850.....	26
Figura 12. Modelo OSI y el Estándar IEC61850.	28
Figura 13. Proceso de ingeniería en subestaciones.	29
Figura 14. Proceso de configuración de archivos SCL.	30
Figura 15. 61850 ICD Editor.....	31
Figura 16. Creación de un ICD.	32
Figura 17. Creación de un Dispositivo Lógico.....	33
Figura 18. Archivo ICD.	33
Figura 19. Creación de un Nodo Lógico.	34
Figura 20. Creación de DataSet.....	35
Figura 21. Creación de un Bloque de Control.....	36
Figura 22. ReLab OPC Console.....	37
Figura 23. Pestaña dashboard.....	38
Figura 24. Creación de un Driver en Relab.....	38
Figura 25. Selección del driver para un dispositivo IEC61850.....	39
Figura 26. Configuración del Driver IEC61850.....	39
Figura 27. Configuración TCP/IP.	40
Figura 28. IED conectado a Relab OPC.....	40
Figura 29. Mapeado de Tags del IED.	41
Figura 30. Interfaz gráfica IEDexplorer.	42
Figura 31. Configuración de IEDexplorer.....	42
Figura 32. Vista de Datos del IED.	43
Figura 33. Ventana de monitoreo de atributos.	44
Figura 34. Ventana de envío de comando MMS.....	44
Figura 35. Goose Explorer.	45

Figura 36. GOOSE sender.....	46
Figura 37. Visualización de Reportes.	46
Figura 38. GOOSE sender.....	47
Figura 39. GOOSE Receiver.....	48
Figura 40. Archivo SCL de mensajes GOOSE importado.....	49
Figura 41. Simulación de envío de mensajes GOOSE.....	49
Figura 42. Búsqueda de mensajes GOOSE.....	50
Figura 43. Simulador RELE IEC61850.....	51
Figura 44. Goose Publisher.....	52
Figura 45. Remote Protection Simulator GOOSE.....	52
Figura 46. Simulator Bay.....	53
Figura 47. Tráfico de Datos en Wireshark.....	54
Figura 48. Trama Sampled Value capturado en Wireshark.....	55
Figura 49. Trama GOOSE capturado en Wireshark.....	55
Figura 50. Selección de adaptadores de Red en Wireshark.....	56
Figura 51. Selección de Protocolo bajo análisis.....	56
Figura 52. Tráfico de datos de mensajes GOOSE.....	57
Figura 53. Software Elipse Power.....	58
Figura 54. Creación de un nuevo proyecto en Elipse Power.....	58
Figura 55. Configuración del modelo eléctrico.....	59
Figura 56. Configuración del modelo de la Subestación.....	59
Figura 57. Importando modelo eléctrico hacia el SCADA.....	60
Figura 58. Importando modelo eléctrico de subestación.....	61
Figura 59. Sincronización del modelo eléctrico en pantalla principal SCADA.....	62
Figura 60. Creación de Driver OPC.....	62
Figura 61. Configuración de servidor OPC.....	63
Figura 62. Ventana de configuración de VNCViewer.....	64
Figura 63. Estructura de servidor IEC61850 de la librería.....	64
Figura 64. Archivos necesarios para ejecutar servidor bajo el estándar IEC61850.....	66
Figura 65. Distribución de pines Raspberry Pi 3B.....	71
Figura 66. Rele SEL 351.....	72
Figura 67. Funciones del Relé de Protección SEL351.....	73
Figura 68. Pines GPIO Raspberry pi 3B.....	74
Figura 69. Router Claro A7600 A1.....	75
Figura 70. Red de comunicación para implementación de escenarios.....	77
Figura 71. Modelo de Datos del Rele Infotech.....	78
Figura 72. Atributos y Objetos de Datos del LN Obj1XCBR1.....	80
Figura 73. DataSet, Reportes y Mensajes GOOSE del Relé INFOTECH.....	81
Figura 74. Habilitar mensajes GOOSE.....	82
Figura 75. Datos del mensaje GOOSE gcb1.....	83
Figura 76. Mensaje GOOSE del software GOOSE Publisher.....	84
Figura 77. Parámetros GOOSE para el disparo del interruptor INFOTECH.....	84
Figura 78. Envío de mensaje GOOSE desde el GOOSE Sender.....	85
Figura 79. Habilitar Reportes Almacenados y No Almacenados.....	86
Figura 80. Configuración de comunicación del Relé AxonGroup.....	87
Figura 81. Modelo de Datos del Relé Simulator Bay de Axon Group.....	87
Figura 82. DataSet, Reportes y Mensajes GOOSE del Relé Simulator Bay.....	88

Figura 83. Desactivando envío de mensajes GOOSE.	89
Figura 84. Mensaje GOOSE del Software Remote Proteccion Simulator.	90
Figura 85. Parámetros GOOSE para el disparo del interruptor INTA_XCBR1.	91
Figura 86. Envío de mensaje GOOSE desde el GOOSE Sender.	91
Figura 87. Estructura IEDRASPBerry.	92
Figura 88. Elementos y diagrama electrico IEDRaspberry.	93
Figura 89. Modelo de Datos del IEDRaspberry.	95
Figura 90. Configuración de ICD IEDRaspberry.	96
Figura 91. Configuración de DataSet del ICD IEDRaspberry.	97
Figura 92. Configuración de mensajes GOOSE en el ICD IEDRaspberry.	97
Figura 93. ICD IEDRaspberry Pi.	98
Figura 94. Esquema de Configuración SEL351 bajo el estándar IEC61850.	101
Figura 95. Modelo de Datos del Relé SEL351.	104
Figura 96. Configuración ICD SEL351.	105
Figura 97. Configuración de Dataset del relé SEL351.	106
Figura 98. Configuración de mensaje GOOSE del Relé Sel351.	106
Figura 99. ICD del Relé Sel351.	107
Figura 100. Listado de registro Modbus de salidas del Relé SEL351.	109
Figura 101. Modelo Eléctrico de un SAS.	110
Figura 102. Drivers IEC61850 para la comunicación de IEDs simulados/emulados.	111
Figura 103. Tags mapeados en ReLab OPC.	113
Figura 104. Importando Tags desde el servidor Relab OPC.	114
Figura 105. Elementos del modelo eléctrico en PowerElipse.	114
Figura 106. Indicadores del transformador.	115
Figura 107. Asignación de Tag para la posición del interruptor I10.	116
Figura 108. Tag de posición para el interruptor I10.	117
Figura 109. Configuración de tags para los parámetros eléctricos del interruptor I10.	118
Figura 110. Propiedades del indicador 71Q.	118
Figura 111. Asignación de Tags al indicador 71Q.	119
Figura 112. Configuración de TAG para el indicador 71Q.	120
Figura 113. Asignación de evento del indicador 71Q.	120
Figura 114. Configuración del evento de activación de indicador 71Q.	121
Figura 115. Configuración de Alarmas en PowerElipse SCADA.	122
Figura 116. Creacion de Alarmas en PowerElipse SCADA.	122
Figura 117. Configuración de Alarma para disparo del Relé 1 AxonGroup.	123
Figura 118. Configuración de alarma digital para el disparo del Relé 1 de AxonGroup.	123
Figura 119. Configuración de alarma analógica para niveles altos de corriente del relé INFOTECH.	124
Figura 120. Asignación de panel para visualización de alarmas.	125
Figura 121. Diagrama general para los escenarios de pruebas.	127
Figura 122. Esquema de Protección de transformador.	128
Figura 123. Mensaje GOOSE enviado por las protecciones del transformador.	129
Figura 124. Esquema comunicaciones escenario #1.	130
Figura 125. Verificación inicio interruptor INFOTECH.	130
Figura 126. Ejecución de convertidor IEC61850/Modbus.	131
Figura 127. Ejecución de convertidor IEC61850/Modbus.	132
Figura 128. Detección de disparo de protección mecánica de TX.	133

Figura 129. HMI AcSELerator Quickset.	133
Figura 130. Mensaje GOOSE enviado por IEDRASPBERRY.....	134
Figura 131. Comando MMS de reseteo en alarma Gateway IEC61850/Modbus.	135
Figura 132. Esquema de Protección de barra simple.	136
Figura 133. Esquema de comunicación del escenario 2.....	137
Figura 134. Verificación inicio interruptor INFOTECH.....	138
Figura 135. Ejecución de relés virtuales Simulator Bay e Infotech	139
Figura 136. Monitoreo a través de GOOSE del relé INFOTECH.....	139
Figura 137. Simulación de una falla en los alimentadores de salida.....	140
Figura 138. Monitoreo a través de reportes del estado del Relé AxonGroup.	141
Figura 139. Simulación de una falla a la salida de I11.....	142
Figura 140. Simulación de una falla a la salida de I12.....	143
Figura 141. Simulación de una falla en la barra utilizando el relé INFOTECH.	144
Figura 142. Mensaje GOOSE para el cambio de estado del Relé INFOTECH.	145
Figura 143. Simulación de una falla en la barra (alimentador de entrada I10).	145
Figura 144. Envío de mensaje GOOSE del IEDRaspberry a los relés de bahía.....	146
Figura 145. Disparo de los relés de bahía a través de protocolo GOOSE.....	146
Figura 146. Mensaje GOOSE enviado por la IEDRaspberry.....	147
Figura 147. Simulación de un SAS niveles de Estación y Bahía.	149
Figura 148. Modelo eléctrico en SCADA	150
Figura 149. SCADA de la subestación de barra simple	152
Figura 150. Protecciones mecánicas del transformador supervisado a través de SCADA.	153
Figura 151. Protección del transformador 63B supervisado a través de SCADA.....	154
Figura 152. Condiciones de sobre corriente en I11.....	155
Figura 153. Interruptor I11 disparado por sobre corriente instantánea	156
Figura 154. Sobre corriente en la barra simulado por el relé INFOTECH.....	157
Figura 155. Disparo de I10 por falla en la barra	158
Figura 156. Proteccion de barra por sobre corriente instantánea	159
Figura 157. Pantalla de selección sistema operativo.....	176

Glosario

SEP	Un sistema eléctrico de potencia es una red de componentes eléctricos instalados para suministrar, transferir y usar energía eléctrica. Un ejemplo de un sistema de potencia es la red que proporciona energía a un área extendida.
SAS	Substation Automation System, provee protección, control, automatización, monitoreo y capacidades de comunicación como parte de una solución exhaustiva de control y monitoreo de subestaciones
SCADA	Supervisory Control And Adquisition, es un concepto que se emplea para realizar un software para ordenadores que permite controlar y supervisar procesos industriales.
IED	Intelligent Electronic Devices, Reles de protección, controladores de bahía, controladores, etc. Se conocen como IED en el entorno IEC61850. Reciben comandos y envían datos y valores medidos a través de la red.
GOOSE	Los mensajes GOOSE se utilizan para transmitir rápidamente información crítica en el tiempo: en este caso, la información no se reconoce, sino que se distribuye en forma de multidifusión.
MMS	Manufacturing Message Specification, desarrollado específicamente para aplicaciones industriales; sirve para el intercambio de datos en ambientes de producción. El protocolo define mensajes de comunicación transferidos entre controladores, así como entre la estación de ingeniería y el controlador.
SV	Sampled Values, los valores muestreados se utilizan para transmitir valores medidos desde transformadores de corriente y tensión convencionales o no convencionales, esto se logra mediante este protocolo
MU	Merging Unit, las MU enlazan los transformadores de corriente y tensión con el bus de procesos (red de comunicación con datos de tiempo real).
Dispositivos lógicos	Está compuesto por un conjunto de nodos lógicos y servicios que están relacionados. Se asocian directamente con un dispositivo real. Por ejemplo, un interruptor, un seccionador, o una protección, etc. O sea, a partir de este objeto es que se puede modelar cualquier equipo de la subestación.

Introducción

El estándar IEC61850 viene a revolucionar a los Sistemas de Automatización de Subestaciones (SAS) los cuales son los responsables de la protección, monitoreo y control de equipos eléctricos en una subestación. El estándar incorpora el uso de nodos lógicos para resolver problemas relacionados con la interoperabilidad y la intercambiabilidad en sistemas de subestaciones. Además, incluye protocolos que actúan con mayor rapidez dentro de los sistemas de comunicación (GOOSE y Sample Value) y reducen el cableado de cobre en una subestación.

Teniendo en cuenta la importancia del estándar este documento en su primer capítulo aborda la investigación del estándar mediante la síntesis de la norma, se incluye información e instrucciones de uso de softwares y una librería bajo el estándar que permite el montaje de un laboratorio mediante simuladores, IEDs emulados, clientes virtuales entre otras herramientas. Se incluye en dicho capítulo información y configuraciones del software ElipsePower (SCADA) funcionando bajo el estándar.

El capítulo dos aborda la configuración de parámetros de los software y equipos utilizado para el montaje de las pruebas de laboratorio. Por parte de los emuladores se desarrolló un IED de bajo costo y con funciones básicas, utilizando una Raspberry Pi corriendo una distribución Linux y ejecutando código en C con las funciones de la librería “libiec61850” para el manejo de los protocolos del estándar y la librería “WiringPi” para el manejo de los pines GPIO de las Raspberry Pi, ambas librerías bajo licencia OpenSource, se ha llamado a este dispositivo “IEDRASPBERRY”. También se desarrolla un Gateway IEC61850/Modbus para incorporar un RELE 351 del fabricante SEL en los escenarios de pruebas, ya que el dispositivo con el que se cuenta trabaja con el protocolo Modbus.

El capítulo tres aborda la ejecución de escenarios de pruebas en simuladores y emuladores que permiten el estudio del estándar y la configuración de protocolos MMS y GOOSE. Entre los softwares de simulación utilizados se incluyen interruptores de potencia, protecciones por sobre corriente, medidores y envío de información en tiempo real en mensajes GOOSE.

Justificación

El protocolo IEC61850 se ha convertido en un estándar en la mayoría de los dispositivos inteligentes que se utilizan en subestaciones eléctricas; en nuestro país son pocas las subestaciones donde ha sido aplicado el estándar, sin embargo, su uso se incrementa con el paso del tiempo y con la llegada de las redes inteligentes (SmartGrid) a futuro el sector eléctrico tendrá que migrar a dicho estándar, de tal forma que la investigación y documentación de pruebas en simuladores, emuladores y equipos representa un avance en el desarrollo de estas nuevas tecnologías en el país.

Por lo que surge la necesidad de preparar a futuros profesionales del sector electricidad o carreras afines en las nuevas tecnologías de subestaciones, incluyendo en su plan de estudio muchos de los temas desarrollados en este documento, por tanto, el presente documento considerarse una aportación de conocimiento técnico para ser retomado por la planta docente del sector de ingeniería eléctrica en el país.

CAPÍTULO I

Marco Conceptual

1.0 Introducción

En el presente capítulo se muestra la investigación del estándar IEC61850, sus conceptos, ventajas, uso actual en subestaciones, como está conformada, ejemplos de su uso entre otros temas. También incluye la investigación de software y librerías para poder realizar pruebas a nivel académico de forma virtual mediante simuladores, IEDs virtuales, servidores, clientes y otros. Se listan las características de algunos equipos necesarios para el montaje del laboratorio.

1.1 El estándar IEC 61850

Las subestaciones eléctricas juegan un papel muy importante dentro de un SEP y están formadas por un conjunto de equipos eléctricos responsables de cambiar los niveles de tensión en todo el sistema eléctrico, por ello es indispensable que las subestaciones también cuenten con equipos de monitoreo, control y protección que ayuden a mitigar fallas en el menor tiempo posible garantizando la confiabilidad del sistema, es así, que surge la necesidad de tener un sistema automatizado en una subestación.

Los Sistemas de Automatización de Subestaciones (SAS) cumplen con el objetivo de supervisar la mayor cantidad de elementos, simplificar los procesos de ingeniería y pruebas y disminuir el uso de cable de control. En las últimas décadas estos sistemas han ido evolucionando junto con la nueva generación de equipos de protección y control, denominados Dispositivos Electrónicos Inteligentes (IED's),

Los IED's son los responsables de recibir datos y medidas generados en los equipos de una subestación y pueden emitir ordenes de disparos a los equipos de protección ante la presencia de anomalías, por lo que la comunicación en estos dispositivos debe estar bien estructurada y diseñada para el correcto intercambio de datos, es así que el desarrollo de un protocolo de comunicación confiable, robusto e interoperable se vuelve indispensable para el correcto funcionamiento en un SAS, como solución a los problemas de comunicación surgieron protocolos propietarios y libres como DNP3, MODBUS, IEC61870 101, 102, 104, etc.

Durante mucho tiempo estos protocolos han sido utilizados en sistemas de automatización, y aunque han tenido logros favorables en el campo de los sistemas de comunicación, el inconveniente de dichos protocolos es su complejidad en términos de subestación y configuración del instrumento, además presentan dificultad para identificar las ingenierías de control y protección, por otro lado no logran garantizar por completo que IED's de distintos proveedores se comuniquen entre sí (interoperabilidad), a esto se suma también la llegada de las redes inteligentes (SmartGrid) ya que las nuevas tecnologías sobrepasan las capacidades de red para las que estos protocolos fueron diseñados originalmente.

En 1994 el grupo "Substation Control and Protection Interfaces" del comité técnico TC57, adscrito a IEC elaboraron los primeros objetivos para lo que sería el futuro del estándar en sistemas de comunicaciones en subestaciones eléctricas. Tan solo 10 años después en

conjunto con *Institute of Electrical and Electronic Engineers (IEEE)* y *Electric Power Research Institute (EPRI)* en el proyecto *Utility Communication Architecture (UCA)*, se llegó a un acuerdo para generar una norma de aceptación mundial, la cual denominaron “*IEC61850 Redes de Comunicaciones y Sistemas en Subestaciones*”, publicada en el año 2004.

IEC 61850 es un estándar internacional para la comunicación de protección, control y medida en los sistemas de automatización de subestaciones (SAS), El estándar surge de la necesidad de unificar protocolos tanto estandarizados (DNP3, Modbus etc.), como propietarios, con el fin de conseguir interoperabilidad entre fabricantes. Esto significa la capacidad de dos o más equipos electrónicos inteligentes (IEDs, intelligent electronic devices), de distintos fabricantes para intercambiar información entre ellos, y conseguir de este modo un funcionamiento y operación correcta.

La interoperabilidad prevista en el estándar IEC 61850 es mucho más que una simple transferencia de datos, también toma en cuenta el intercambio de información entre dos o más dispositivos similares. El receptor tiene que entender no solo la estructura de los datos (sintaxis), sino que también deberá comprender su significado, es decir la semántica basada en los atributos de los datos recibidos en la comunicación.

Para lograr el objetivo principal de interoperabilidad la norma establece una serie de aspectos importantes los cuales se describen a continuación:

- Una estructura formal y jerárquica para el Sistema de Automatización.
- Dos redes particulares de comunicación (red de estación + red del proceso).
- Una serie de parámetros de calidad del sistema (confiabilidad, seguridad, disponibilidad, otros).
- Un innovativo modelo de datos basado en artificios abstractos (dispositivos lógicos, nodos lógicos, etc.) para implementar las funcionalidades del sistema.
- Tres servicios particulares de comunicación (MMS, GOOSE y Sample Values).
- Un método exhaustivo para realizar la ingeniería del sistema mediante un conjunto de archivos normalizados (Lenguaje de configuración SCL) aplicando un grupo de herramientas de Ingeniería.

1.1.1 Estructura de la norma

La norma IEC 61850 está estructurada en 14 partes provenientes de 10 capítulos los cuales se describen brevemente a continuación:

- ✓ **IEC 61850-1 Introducción y vista general:**
 - Presenta el alcance de aplicabilidad de la norma en los Sistemas de Automatización de las Empresas Eléctricas *Power Utility Automation Systems (PUAS)*.

- Define la comunicación entre IEDs en la subestación y los requerimientos relacionados del sistema.
 - Introducción general de la norma y sus partes.
- ✓ **IEC 61850-2 Glosario:** Presenta las definiciones y acrónimos utilizados en el contexto de la automatización de las subestaciones.
- ✓ **IEC 61850-3 Requerimientos generales:** Especifica en forma general los requerimientos de la red de comunicaciones incluyendo calidad (confiabilidad, disponibilidad, mantenimiento, seguridad e integridad de los datos) así como las directrices para las condiciones ambientales (temperatura, humedad, presión, condiciones mecánicas y sísmicas, polución y corrosión) y servicios auxiliares.
- ✓ **IEC 61850-4 Administrador de proyectos y sistemas:** Describe en forma resumida el proceso de ingeniería (requerimientos, diseño, parametrización y funciones), las herramientas de soporte, las especificaciones de los IEDs e interfaces, el ciclo de vida del sistema considerando la actualización a nuevas versiones y la discontinuación de IEDs, así como el aseguramiento de la calidad desde el desarrollo hasta el fin del sistema de automatización.
- ✓ **IEC 61850-5 Requerimientos de comunicación:**
- En este apartado se estandariza la comunicación entre IEDs y define los requerimientos del sistema los cuales están basados en los siguientes contenidos:
- Nodos Lógicos (LogicalNode, LN)
 - Enlaces lógicos de comunicación
 - PICOM (Descripción de la transferencia de información en una conexión lógica determinada entre dos nodos lógicos)
 - Relación entre LN y PICOMs
 - Requerimientos de Desempeño
- ✓ **IEC 61850-6 Configuración del lenguaje para la comunicación de IEDs en subestaciones eléctricas:**
- Esta parte de la norma especifica el lenguaje de descripción de configuración de los IED's en una subestación *Substation Configuration Description Language* (SCL), especifica el formato del archivo para describir la comunicación relacionada a la configuración y parámetros de los IED's, configuraciones de los sistemas de comunicación, estructura del patio de maniobra y la relación entre ellos.
- ✓ **IEC 61850-7 Estructura de comunicación básica:**
- Este apartado contiene el núcleo de la norma IEC61850 y se divide en 4 secciones las cuales se describen a continuación:

- **IEC 61859-7-1 Principios y modelos**
Contiene los principios y modelos de comunicaciones, además presenta una introducción a todo el contenido de IEC 61850-7-X.
 - **IEC 61850-7-2 ASCI (Abstract Communication Service Interfase)**
Contiene la descripción del ACSI, las especificaciones del abstract communication service y el modelo de la estructura del database de los dispositivos.
 - **IEC 61850-7-3 Clase de datos comunes**
Contiene la descripción de los Common data Classes y los atributos relacionados a ellos.
 - **IEC 61859-7-4 Clases de nodos lógicos compatibles y clases de datos.**
Contiene la definición de cada una de las clases de nodos lógicos y sus data classes; todos los nodos lógicos están compuestos de data classes
- ✓ **IEC 61850-8 Servicios de mapeo para comunicación específica (Bus de estación):**
Contiene el mapeo de los servicios comúnmente usados para comunicaciones dentro de toda la subestación, dependiendo su uso se caracterizan en dos protocolos, los cuales son:
- MMS (Unicast) utilizado para servicios principales (SCADA, HMI, etc.)
 - GOOSE (Multicast) para disparos y mensajes rápidos y prioritarios.
- ✓ **IEC 61850-9 Servicios de mapeo para comunicación específica (Bus de procesos):** Describe la forma de implementar los servicios relacionados con el intercambio de información entre equipos de medición primaria, como son transformadores de intensidad (TC) y transformadores de tensión (TP). Este intercambio emplea valores muestreados (Sampled Values), transmitidos en tiempo real (del orden de microsegundos) que guardan un fuerte compromiso con su alineación en el tiempo, con el fin de que el dispositivo de medición o de protección secundario, sea capaz de producir las señales de los TC y TP entregadas de forma directa, es decir, de forma tradicional.
- ✓ **IEC 61850-10 Pruebas de Conformidad:**
Posee los métodos para las pruebas de conformidad de los dispositivos utilizados en los sistemas de automatización de subestaciones, los datos que deben ser medidos dentro de los equipos, de acuerdo con los requisitos definidos en IEC 61850-5, certificación de los servicios de prueba y los requerimientos para la validación de equipos de prueba.

En forma gráfica se presenta en la figura 1 la estructura del estándar IEC 61850:

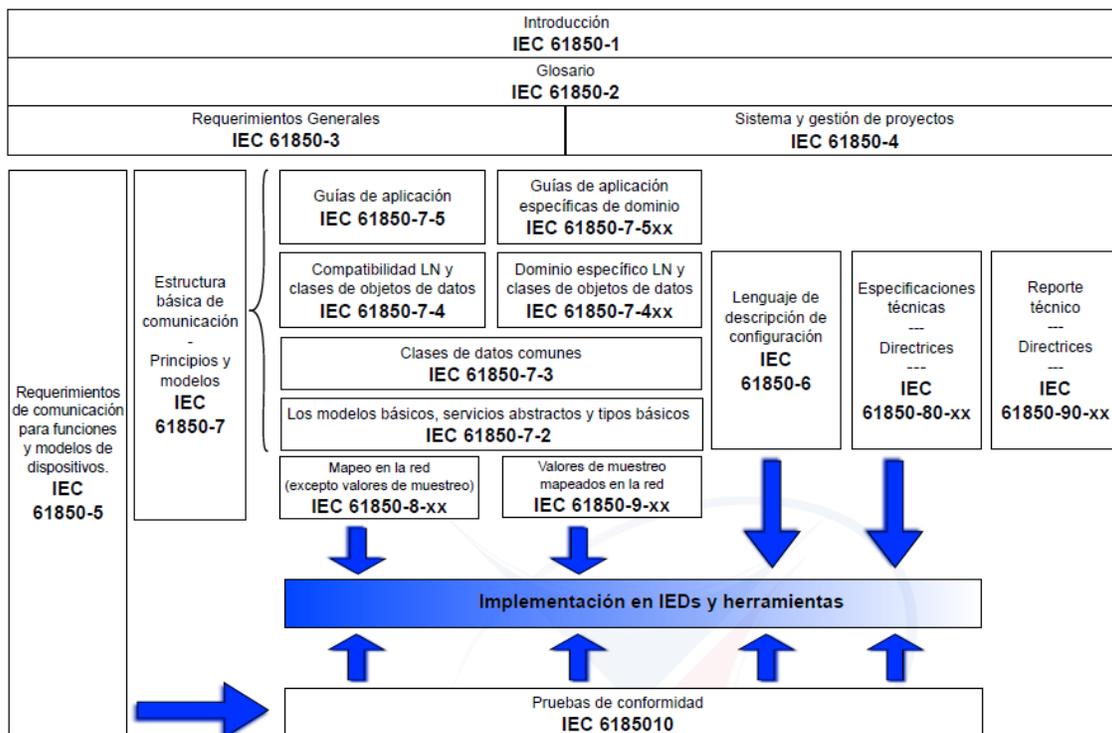


Figura 1. Estructura del estándar IEC 61850.

1.1.2 Arquitectura de una subestación con el estándar IEC61850

El estándar IEC 61850 divide el SAS en tres niveles, por lo cual, todas las tareas que se requieren para la operación, control, monitoreo y protección, de una subestación, además de las tareas necesarias para el mantenimiento de la misma, deben estar lógicamente localizadas en uno o más de los tres niveles definidos en el estándar, los cuales son:

- 1) **Nivel de Proceso.** Es el que tiene mayor interacción con los equipos primarios de la subestación; contiene todas las funciones encargadas de transportar la información no procesada del sistema de potencia, como las medidas adquiridas por los transformadores de instrumentación, el estado de los equipos de maniobra de la subestación o las señales de mando de estos. En este nivel típicamente son situados las Merging Unit y sensores inteligentes.
- 2) **Nivel de bahía.** Se ubican las funciones que principalmente usan información de una bahía en especial de la subestación, además de actuar sobre los equipos de esta misma. Consta de las unidades de protección, de monitoreo y de control de cada una de las bahías.
- 3) **Nivel de Estación.** De este nivel hace parte el computador de la subestación en el cual se almacena la base de datos con toda la información de maniobras, eventos, alarmas, tendencias, etc. de la subestación y la interfaz remota de comunicación. Es el centro de control y supervisión de la subestación (SCADA).

El nivel de estación tiene dos clases de funciones de estación: funciones de estación relacionadas con el proceso, las cuales usan la información de la subestación para actuar sobre los equipos primarios de una bahía o de la subestación completa, y funciones de estación relacionadas con la interfaz, las cuales relacionan el SAS con la interfaz hombre máquina (IHM) del mismo.

En la figura 2 se muestra de forma ilustrativa los instrumentos y equipos que se encuentran en cada uno de los niveles antes mencionados.

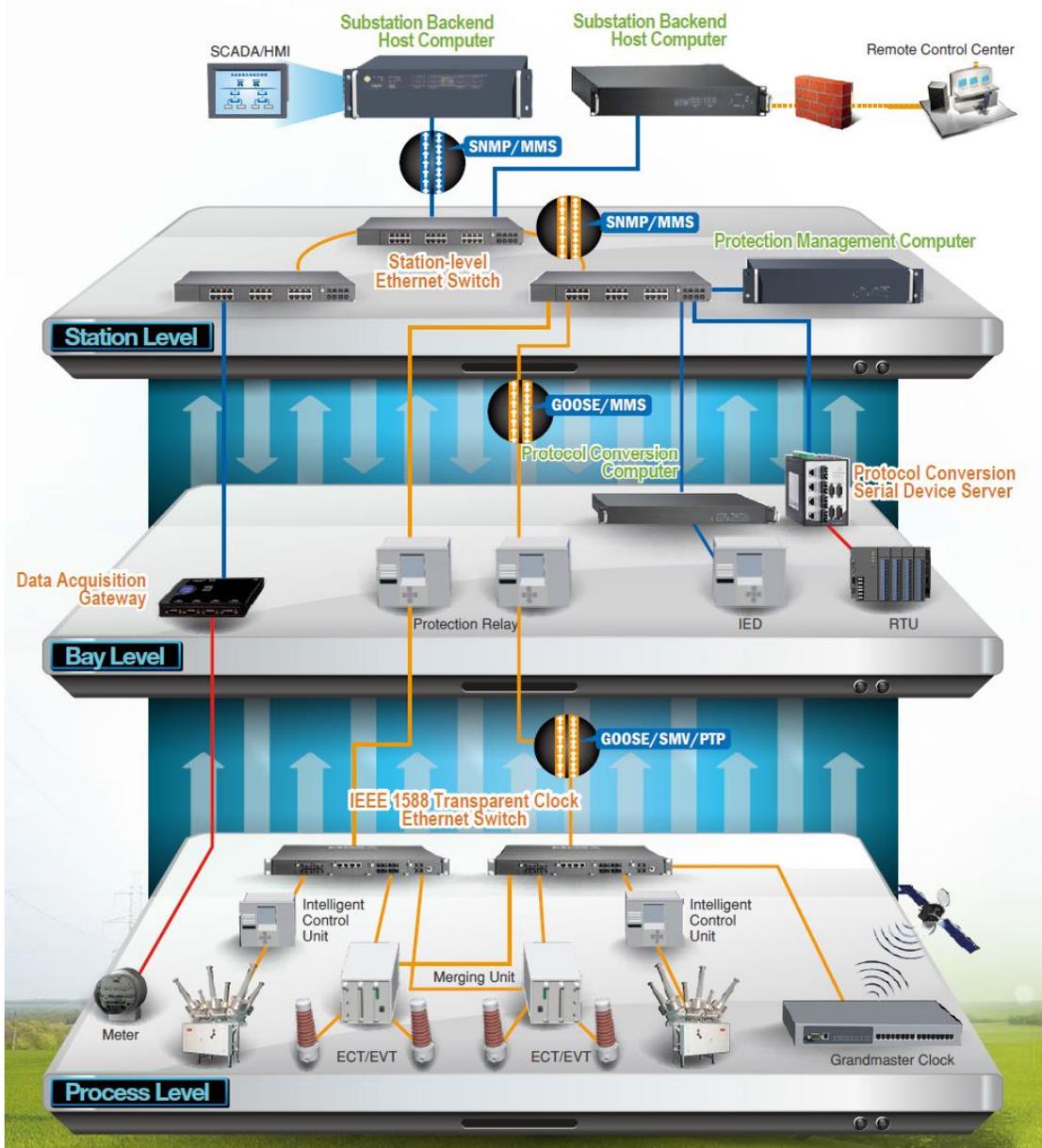


Figura 2. Arquitectura de un SAS en IEC61850.

En la figura 2 se observa los equipos de un SAS físicamente instalados en los distintos niveles, en el nivel proceso encontramos actuadores, sensores, transformadores de instrumentación etc. que envían información a equipos instalados en patio llamados “Merging Unit”.

Las “Merging Unit (MU)” han surgido de la necesidad de cumplir con los requerimientos de la norma pues dichos equipos reciben las mediciones y datos de sensores que se encuentran en patio y de forma digital envían dicha información a los equipos ubicados en el nivel bahía a través de ethernet o fibra óptica (Sampled Values y GOOSE), reduciendo de una manera significativa el exceso de cables.

Los equipos del nivel de bahía consisten en las unidades de protección y control o supervisión. Mientras que los de nivel de estación son el ordenador de la estación con la base de datos, el puesto de operador, interfaces para comunicación remota, etc.

Entre estos niveles y dentro de los mismos existen interfaces lógicas de comunicaciones que conforman la red de comunicaciones del SAS. Estas interfaces están representadas en la figura 3 estando los números 2 y 10 fuera del alcance de la norma.

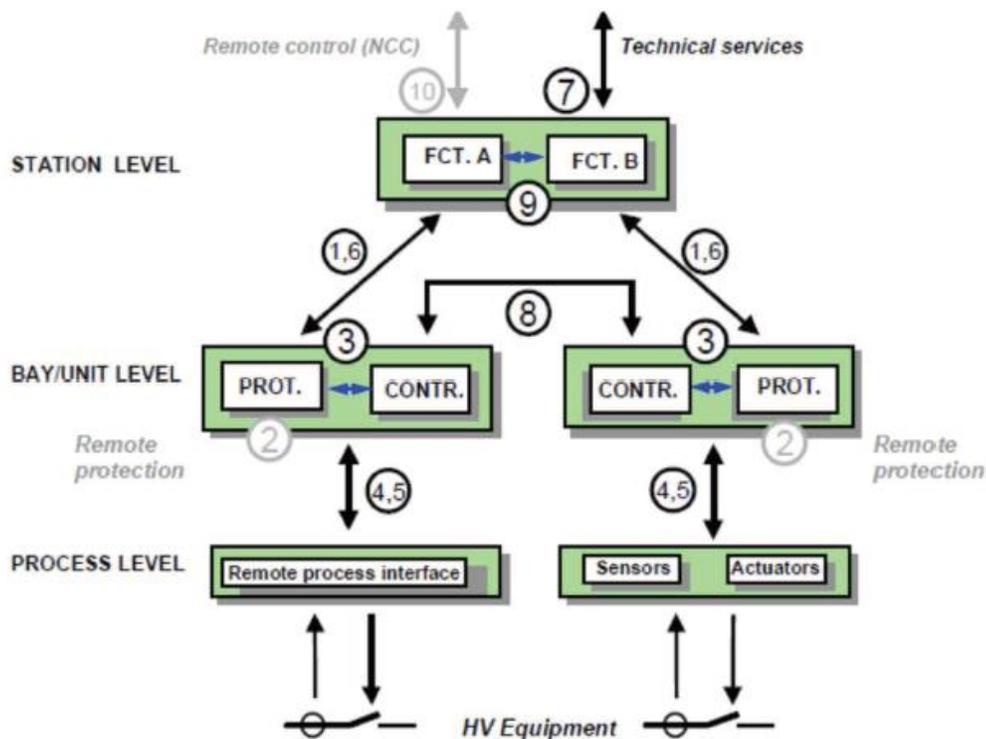


Figura 3. Interfaces de comunicación.

Cada una de las interfaces mostradas representan una función en el sistema de comunicación de un SAS, estos se definen como:

- Interfaz 1: Intercambio de datos de protección entre los niveles de bahía y estación.
- Interfaz 2: Intercambio de datos protección entre el nivel bahía y protección remota.
- Interfaz 3: Intercambio de datos dentro del nivel de bahía.
- Interfaz 4: Intercambio de datos instantáneos entre el nivel de proceso y el de bahía.
- Interfaz 5: Intercambio de datos de control entre los niveles de proceso y bahía.
- Interfaz 6: Intercambio de datos de control entre los niveles de bahía y estación.
- Interfaz 7: Intercambio de datos del nivel de estación y el puesto remoto de ingeniería.
- Interfaz 8: Intercambio de datos directo entre de bahías.
- Interfaz 9: Intercambio de datos dentro del nivel de estación.
- Interfaz 10: Intercambio datos de control entre equipos de subestación y centro control remoto.

La arquitectura de un SAS planteado en la norma muestra también dos redes LAN o buses de comunicación que son la base fundamental para el mejoramiento de la comunicación en las subestaciones actuales, estos reciben el nombre de BUS de proceso y BUS de estación:

- **BUS de estación o inter bahías:** Esta red es la que establece comunicación entre equipos del nivel de bahía y equipos de estación, en el interactúan las interfaces IF3, IF1, IF6, IF9 e IF8, el envío de información en este bus se hace a través de protocolos GOOSE y MMS.
- **BUS de proceso:** El bus de proceso es la nueva red que introduce la norma a los sistemas de comunicación en subestaciones y es un concepto de tecnología E/S distribuidas para sistemas de protección y control, establece la comunicación entre equipos de patios y equipos de bahía, las interfaces IF4 y la IF5 son las que interactúan en esta red enviando información a través de protocolos GOOSE y Sampled Values (SV).

1.1.3 Modelo de datos

El modelo de datos planteado en la norma IEC61850, surge de la dificultad de interpretar las ingenierías de control de la subestación diseñadas con el uso de protocolos propietarios u otros protocolos libres, ya que éstos no cuentan con una estandarización que faciliten la identificación de las configuraciones, limitando su mantenimiento y ajuste a personal que no se encuentra al tanto de todo el diseño de la misma. Es por esto que la norma IEC61850, definió la estructura de la subestación en objetos y funciones, logrando así definir el modelo de datos orientado a objetos, estandarizando la identificación de las estructuras del sistema automatizado de la subestación.

Lo que se pretende con la definición del modelo de datos es obtener una imagen virtual del mundo real, Así, se podrán modelar todos los objetos físicos reales (transformador, interruptor, seccionador, etc.) con todas sus propiedades, así como todas las funciones que deben ser desarrolladas dentro de una subestación. La norma establece un modelo de

información jerárquico, el cual permite virtualizar los dispositivos físicos y consta de cinco niveles los cuales son mostrados en la figura 4.

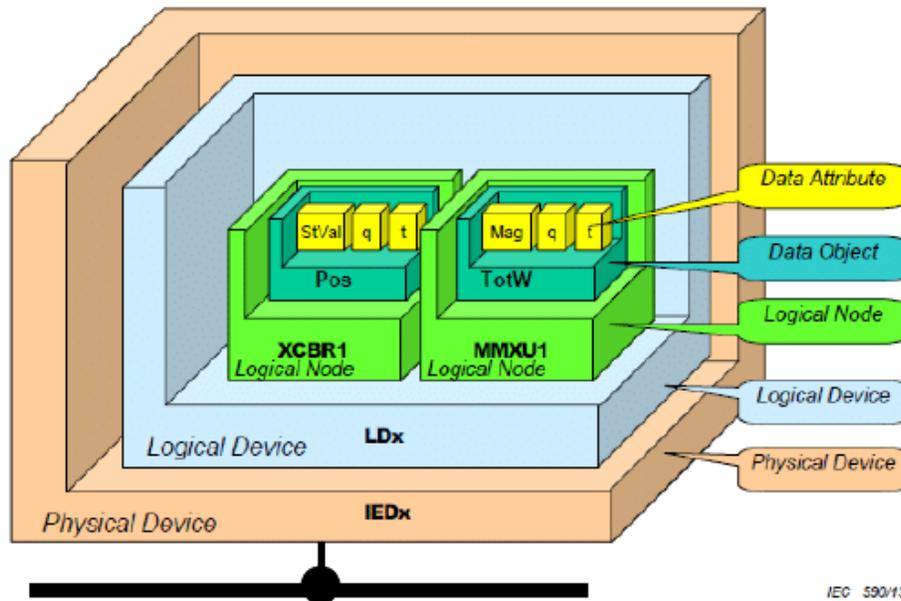


Figura 4. Modelo de datos para virtualización de dispositivos del estándar IEC61850.

Tal como se observa en la figura 4 existen cinco niveles que determinan el modelo de datos, en orden ascendente se tiene:

- **Nivel 1: Dispositivo Físico (IED)**, el cual posee un punto de conexión al mundo exterior a través de la red de comunicación.
- **Nivel 2: Dispositivo Lógico (LD)**, el cual está formado por un conjunto de nodos lógicos.
- **Nivel 3: Nodo Lógico (LN)**, este nivel corresponde al proceso de virtualización en el ejemplo el no lógico *XCBR* representa a un interruptor de potencia y *MMXU* una medición eléctrica.
- **Nivel 4: Objeto de Datos (DO)**, en este nivel se encuentra la información específica de un elemento, cada nodo lógico cuenta con objetos de datos los cuales están definidos en la norma, para el ejemplo *Pos* representa la posición del interruptor y *TotW* es una medida de la potencia activa.
- **Nivel 5: Atributos de Datos (DA)**, el ultimo nivel contiene información detallada del valor del objeto de datos, en el ejemplo se tiene los atributos *Stval* el cual describe el estado de la posición del interruptor y *Mag* el cual corresponde a la magnitud de la potencia activa, *q* y *t* son atributos de datos que corresponden a calidad y a estampas de tiempo respectivamente, como se puede notar estos DA son iguales en ambos objetos de datos por lo que la norma agrupa a dichos atributos en clases comunes de datos CDC.

La definición de este modelo de datos constituye el núcleo de la norma y atribuye al estándar dos de sus principales características: la posibilidad de auto descripción de los objetos y la posibilidad de intercambiar dispositivos de diferentes fabricantes siempre y cuando cumplan con las mismas funciones (interoperabilidad).

En la figura 5 se puede observar como el estándar IEC 61850 utiliza el concepto de virtualización en la creación del modelo de datos, modelando la información de equipos reales encontrados en las subestaciones, proporcionando una imagen del mundo físico al sistema de automatización de las subestaciones.

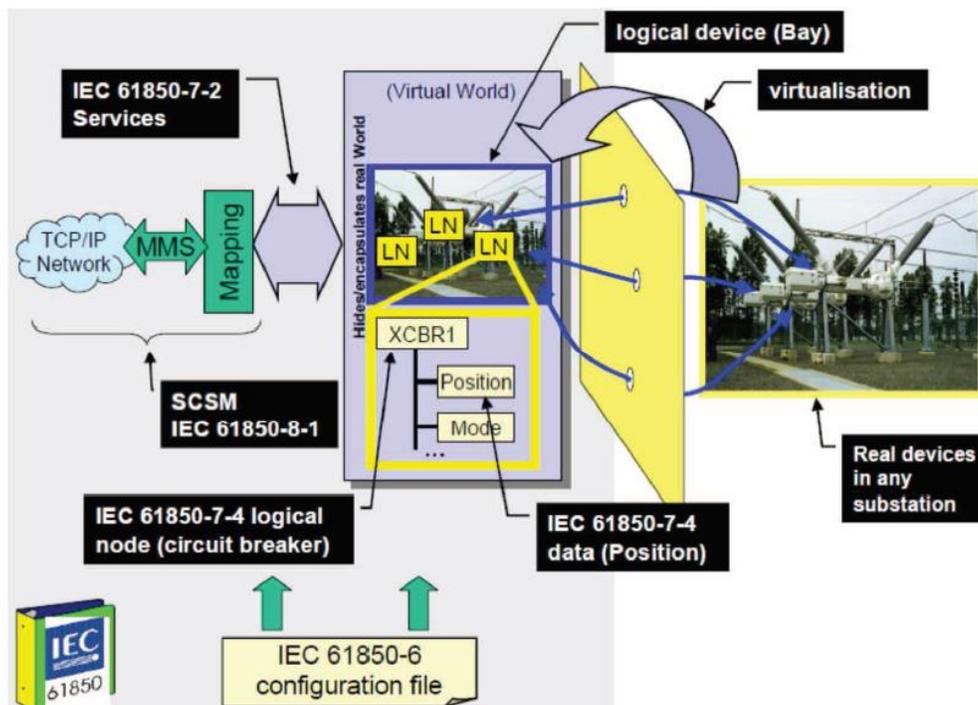


Figura 5. Concepto del modelo de datos.

Como se puede observar en la figura 5 la parte fundamental para que exista una virtualización del mundo real son los llamados *Nodos Lógicos*, los cuales representan funciones que intervienen en la automatización de una subestación, sin duda, son los pilares de la norma por lo que su estudio, será necesario para una mejor comprensión del estándar IEC61850.

1.1.4 Nodos Lógicos

Los nodos lógicos están ubicados en el tercer nivel de la jerarquía del modelo de datos y son una representación virtual de las funciones fundamentales dentro de un SAS, contienen agrupaciones predefinidas de objetos de datos que cumplen funciones específicas. En el estándar IEC61850-7-4 están definidos 19 grupos de nodos lógicos, según el propósito que desempeñan en el SAS, se asigna a cada nodo lógico un nombre que hace referencia al indicador del grupo al que pertenece, por ejemplo, un nodo lógico de protección iniciara con

la letra PXXX, A continuación, se presentan los grupos que se encuentran dentro de la norma según los niveles de la subestación:

- Nivel de Estación:
 - Grupo I: Interfaces y registros.
- Nivel de Posición o Bahía:
 - Grupo A: Control Automático.
 - Grupo C: Control.
 - Grupo F: Bloques Funcionales.
 - Grupo M: Medida.
 - Grupo P: Protección.
 - Grupo R: Relacionadas con protección.
 - Grupo S: Supervisión y monitorización.
- Nivel de Proceso:
 - Grupo T: Transformadores de instrumentación.
 - Grupo X: Interruptores.
 - Grupo Y: Transformadores de potencia.
 - Grupo Z: Transformadores de potencia secundaria.
- Nodos de sistema y otros:
 - Grupos L: Nodos lógicos del sistema.
 - Grupo W: Energía Eólica (Ed2).
 - Grupo G: Nodos lógicos genéricos.
 - Grupo K: Nodos lógicos para equipamiento mecánico y no eléctrico.
 - Grupo Q; Nodos lógicos para eventos de calidad de energía.
 - Grupo D: Fuentes de Energía Distribuida (Ed2).
 - Grupo H: Hidroeléctricas (Ed2).

Cada grupo describe las diferentes funciones en una subestación, en el apartado IEC61850 7-2 del estándar se definen las funciones mostradas en la tabla 1.

GRUPO L: NODO LÓGICO DEL SISTEMA	
Describen información específica del sistema, donde se incluye información de los dispositivos físicos que contienen dispositivos lógicos.	
NODO LÓGICO	DESCRIPCIÓN
LPHD	Información de dispositivo físico
LLN0	Logical node zero, Estado de un dispositivo lógico
LCCH	Supervisión de canal físico de comunicación
LGOS	Suscripción GOOSE
LSVS	Suscripción Sampled Values
LTIM	Administración del tiempo
LTMS	Supervisión del tiempo maestro
LTRK	Seguimiento del servicio
GRUPO A: NODO LÓGICO DE CONTROL AUTOMÁTICO	

Describen las funciones de control de procesos, que no están asociadas a la protección	
NODO LÓGICO	DESCRIPCIÓN
ANCR	Regulador de corriente de neutro
ARCO	Control de potencia reactiva
ARIS	Control de resistor
ASEQ	Secuenciador
ATCC	Controlador automático cambiador de tomas
AVCO	Control de voltaje
GRUPO C: NODO LÓGICO DE CONTROL	
Describen las funciones de control, asociado principalmente con los equipos de corte	
NODO LÓGICO	DESCRIPCIÓN
CALH	Manejo de alarmas
CCGR	Control del grupo de refrigeración
CILO	Interlocking
CPOW	Cambio de punto en onda
CSWI	Controlador de interruptor
CSYN	Controlador de sincronismo
GRUPO F: NODO LÓGICO DE BLOQUES FUNCIONALES	
Describe las funciones de nodos lógicos que se encuentran implícitas en la configuración de los dispositivos	
NODO LÓGICO	DESCRIPCIÓN
FCNT	Contador de nodo lógico
FCSD	Descripción de forma de onda
FFIL	Filtro genérico
FLIM	Limitación de salida de la función de control
FPID	Regulador de control PID
FRMP	Función de rampa
FSCC	Controlador de horario
FSCH	Horario
FSPT	Función de control de ajuste
FXOT	Acción de sobreumbral
FXUT	Acción de subumbral
GRUPO G: NODOS LÓGICOS GENÉRICOS	
Permiten representar funciones que no están disponibles en los nodos existentes	
NODO LÓGICO	DESCRIPCIÓN
GAPC	Genérico de control de proceso automático
GGIO	Genérico de proceso de entrada salida
GLOG	Genérico de registro
GSAL	Genérico de aplicación de seguridad

GRUPO I: NODO LÓGICO DE INTERFACE Y ARCHIVO	
Describe las funciones que determinan las interfaces y opciones de archivo en el sistema	
NODO LÓGICO	DESCRIPCIÓN
IARC	Archivo
IHMI	Interface humano-máquina
ISAF	Función de alarma de seguridad
ITCI	Interface de telecontrol
ITMI	Interface de telemonitoreo
ITPC	Interface de comunicación de teleprotección
GRUPO K: NODOS LÓGICOS PARA EQUIPOS PRIARIOS MECÁNICOS Y NO ELÉCTRICOS	
Describe los equipos no eléctricos del sistema y sus funciones	
NODO LÓGICO	DESCRIPCIÓN
KFAN	Ventilador
KFIL	Filtro
KPMP	Bombas
KTNK	Tanque
KVLV	Control de Válvula
GRUPO M: NODOS LÓGICOS DE MEDIDA	
Describe las funciones de medida que se presentan en los equipos destinados para ello	
NODO LÓGICO	DESCRIPCIÓN
MENV	Información Ambiental
MFLK	Medida de fluctuaciones
MFLW	Medida de flujo
MHAI	Medida de armónicos
MHAN	Medida de armónicos no relacionados con fases de Corriente alterna
MHET	Valores medidos en calor
MNTN	Medidas monofásicas
MMTR	Medidas trifásicas
MMXN	Medidas no relacionadas con fase de corriente alterna
MMXU	Medidas generales
MSQI	Medidas de secuencia
GRUPO P: NODOS LÓGICOS FUNCIONES DE PROTECCIÓN	
Describen las funciones de protección	
NODO LÓGICO	DESCRIPCIÓN
PDIF	Protección Diferencial
PDIR	Protección Direccional
PDIS	Protección de distancia
PDOP	Protección de sobrepotencial direccional
PDUP	Protección de su potencia direccional

PFRC	Protección de frecuencia
PHAR	Restricción de armónicos
PHIZ	Protección de falla de aislamiento
PIOC	Protección instantánea de sobre corriente
PMRI	Protección de inhibición de reinicio de motor
PMSS	Protección de supervisión del tiempo de inicio del motor
POPF	Protección de sobre factor de potencia
PPAM	Protección de medida de ángulo de fase
PRTR	Protección de rotor
PSCH	Esquema de protección
PSDE	Protección sensitiva direccional de falla a tierra
PTEF	Protección de falla transiente a tierra
PTHF	Protección de tiristor
PTOC	Protección temporizada de sobrecorriente
PTOF	Protección de sobrefrecuencia
PTOV	Protección de sobretensión
PTRC	Disparo general
PTTR	Protección de temperatura
PTUC	Protección de subcorriente
PTUF	Protección de subfrecuencia
PUPF	Protección de sub factor de potencia
PVOC	Protección de sobrecorriente controlada por tensión
PVPH	Protección voltios por Hertz
PZSU	Protección de velocidad cero o subvelocidad
GRUPO Q: NODOS LÓGICOS PARA EVENTOS DE CALIDAD DE POTENCIA	
Describe las funciones que permiten evidenciar los sucesos que afectan el correcto funcionamiento de la subestación	
NODO LÓGICO	DESCRIPCIÓN
QFVR	Variación de frecuencia
QITR	Transiente de corriente
QIUB	Variación de desbalance de fase
QVTR	Transiente de tensión
QVUB	Variación de desbalance de voltaje
QVVR	Variación de Voltaje
GRUPO R: NODOS LÓGICOS DE FUNCIONES RELACIONADAS CON PROTECCIÓN	
Describe las funciones que complementan las funciones de protección	
NODO LÓGICO	DESCRIPCIÓN
RADR	Registrador de eventos de canal análogo
RBDR	Registrador de eventos de canal binario
RBRF	Falla de interruptor

RDIR	Elemento Direccional
RDRE	Función de registrador de eventos
RDRS	Manejo de registrador de eventos
RFLO	Localizador de fallas
RMXU	Medidas diferenciales
RPSB	Detección y bloqueo de oscilaciones de potencia
RREC	Recierre
RSYN	Verificación de sincronismo

GRUPO S: NODOS LÓGICOS DE SUPERVISIÓN Y MONITOREO

Describen las funciones relacionadas con la supervisión y monitoreo de la subestación

NODO LÓGICO	DESCRIPCIÓN
SARC	Monitoreo y diagnóstico de arcos
SCBR	Supervisión de interruptor
SIMG	Supervisión de medio aislante, gas
SIML	Supervisión de medio aislante, líquido
SLTC	Supervisión del cambiador de tomas
SOPM	Supervisión de los mecanismos de operación
SPDC	Monitoreo y diagnóstico para descargas parciales
SPRS	Supervisión de presión
SPTR	Supervisión de transformador de potencia
SSWI	Supervisión de seccionador
STMP	Supervisión de Temperatura
SVBR	Supervisión de presión

GRUPO T: NODOS LÓGICOS DE TRANSFORMADORES DE INSTRUMENTOS

Describen las funciones asociadas a los transformadores de instrumentos

NODO LÓGICO	DESCRIPCIÓN
TANG	Sensor de ángulo
TAXD	Sensor de Desplazamiento axial
TCTR	Transformador de corriente
TDST	Sensor de distancia
TFLW	Sensor de Flujo de caudal
TFRQ	Sensor de frecuencia
TGSN	Sensor genérico
THUM	Sensor de Humedad
TLVL	Sensor de nivel
TMGF	Sensor de campo magnético
TMVM	Sensor de movimiento
TPOS	Indicador de posición
TPRS	Sensor de presión
TRTN	Trasmisor de velocidad angular

TSND	Sensor de presión sonora
TTMP	Sensor de temperatura
TTNS	Sensor de tensión mecánica
TVBR	Sensor de vibración
TVTR	Transformación de tensión
TWPH	Nivel de PH
GRUPO X: NODOS LÓGICOS DE EQUIPOS DE CORTE	
Describen los equipos de corte o maniobra presentes en la subestación	
NODO LÓGICO	DESCRIPCIÓN
XCBR	Interruptor
XSWI	Seccionador
GRUPO Y: NODOS LÓGICOS DE EQUIPOS DE POTENCIA	
Describen los equipos de potencia presentes en la subestación	
NODO LÓGICO	DESCRIPCIÓN
YEFN	Limitador de falla a tierra
YLTC	Cambiador de tomas
YPSH	Derivación de potencia
YPTR	Transformador de potencia
GRUPO Z: NODOS LÓGICOS PARA OTROS EQUIPOS DEL SISTEMA DE POTENCIA	
Describe los equipos y demás funciones que no están incluidas en los demás grupos	
NODO LÓGICO	DESCRIPCIÓN
ZAXN	Red auxiliar
ZBAT	Batería
ZBSH	Cojinetes
ZCAB	Cables de potencia
ZCAP	Banco de capacitores
ZCON	Convertidor
ZGEN	Generador
ZGIL	Línea aislada a gas
ZLIN	Línea aérea
ZMOT	Motor
ZREA	Reactor
ZRES	Resistor
ZRRC	Componente reactivo rotativo
ZSAR	Pararrayos
ZSCR	Rectificador controlado
ZSMC	Máquina sincrónica
ZTCF	Convertidor de frecuencia controlado por tiristor
ZTCR	Componente reactivo controlado por tiristor

Tabla 1. Funciones de nodos lógicos.

Los nodos lógicos y sus funciones descritos en la tabla 1 son utilizados dependiendo los requerimientos de una subestación, sin embargo, la norma establece el uso obligatorio de 2 nodos lógicos que deben estar presentes en cualquier dispositivo lógico estos son:

- **Nodo Lógico LPHD:** El cual contiene toda la información del dispositivo físico.
- **Nodo Lógico LLN0:** El cual contiene la configuración del dispositivo lógico, en este nodo se encuentra configurados los Dataset y Reportes.

Los nodos lógicos también presentaran objetos de datos que son de uso obligatorio (M), opcional (O) y Condicional (C), la norma define 4 DO que son de carácter mandatorio en un nodo lógico común (LPHD o LLN0) los cuales son **Mode, Behavior, Health y NamePlt**.

XCBR class					
Data object name	Common data class	Explanation			T M/O/C
CBOPCap	ENS	Circuit breaker operating capability			O
POWCap	ENS	Point on wave switching capability			O
MaxOpCap	INS	Circuit breaker operating capability when fully charged			O
Dsc	SPS	Discrepancy			O
Measured and metered values					
SumSwARs	BCR	Sum of switched amperes, resettable			O
Controls					
LocSta	SPC	Switching authority at station level			O
Pos	DPC	Switch position			M
BlkOpn	SPC	Block opening			M
BlkCls	SPC	Block closing			M
ChaMotEna	SPC	Charger motor enabled			O
Settings					
CBTmms	ING	Closing time of breaker			O
DPC class					
Attribute Name	Attribute Type	FC	TrgOp	Value/Value Range	M/O/C
DataName	Inherited from Data Class (see IEC 61850-7-2)				
DataAttribute					
<i>control and status</i>					
ctlVal	BOOLEAN	CO		off (FALSE) on (TRUE)	AC_CO_M
operTm	TimeStamp	CO			AC_CO_O
origin	Originator	CO, ST			AC_CO_O
ctlNum	INT8U	CO, ST		0..255	AC_CO_O
stVal	CODED ENUM	ST	dchg	intermediate-state off on bad-state	M
q	Quality	ST	qchg		M
t	TimeStamp	ST			M
stSeld	BOOLEAN	ST	dchg		AC_CO_O
<i>substitution</i>					
subEna	BOOLEAN	SV			PICS_SUBST
subVal	CODED ENUM	SV		intermediate-state off on bad-state	PICS_SUBST
subQ	Quality	SV			PICS_SUBST
subID	VISIBLE STRING64	SV			PICS_SUBST
<i>configuration, description and extension</i>					
pulseConfig	PulseConfig	CF			AC_CO_O
ctlModel	CtlModels	CF			M
sboTimeout	INT32U	CF			AC_CO_O
sboClass	SboClasses	CF			AC_CO_O
d	VISIBLE STRING255	DC		Text	O
dU	UNICODE STRING255	DC			O
cdcNs	VISIBLE STRING255	EX			AC_DLNDA_M
cdcName	VISIBLE STRING255	EX			AC_DLNDA_M
dataNs	VISIBLE STRING255	EX			AC_DLN_M

Figura 6. Nodo Lógico del Interruptor de Potencia.

1.1.5 Clases de Datos Comunes (CDC)

Según su funcionalidad los nodos lógicos contienen una lista de objetos (*nivel 4 “DO”*) con sus correspondientes atributos (*nivel 5 “DA”*). Para el modelado de estos atributos, los tipos de datos simples (enteros, boléanos, etc.), se organizan formando tipos de datos compuestos (calidad, escala), y estos, a su vez, se agrupan para formar los tipos de datos comunes *Common Data Classes (CDC)*, medida, datos controlables, información de estados, etc. La agrupación de estos últimos forma los nodos lógicos, tal como se observa en la figura 6.

Los CDC dan información sobre parámetros de estado, parámetros de medida, información para ajustes, avisos, medidas, alarmas, información de protección y falla, configuraciones analógicas, etc. Los Atributos, constituyen la característica específica otorgada a cada CDC, refiriéndose a valores de estado (stVal), Valores de calidad (q), estampas de tiempo (t), etc.

El estándar IEC61850 7-3 define los CDC’s contenidos en la tabla 2.

Información de estado	SPS	Single Point Status
	DPS	Double Point Status
	INS	Integer Status
	ACT	Protection Activation info
	ACD	Activation Info Directional Protection
	SEC	Security Violation Counting
	BCR	Binary Counter Reading
Información de medidas	MV	Measurement Value
	CMV	Complex Measured Variable
	SAV	Sampled Value
	WYE	Phase to Ground
	DEL	Phase to Phase
	SEQ	Sequence
	HMW	Harmonic Value
	HWYE	Harmonic Value for WYE
	HDEL	Harmonic Value for DEL
Información de status controlable	SPC	Single Point Control
	DPC	Double Point control
	INC	Integer Status Control
	BSC	Binary Controlled Step Position Info
	ISC	Integer Controlled Step Position Info
Información de ajuste de estado	SPG	Single Point Setting
Información de descripción	DPL	Device Name Plate
	LPL	Logical Node Name Plate
	CSD	Curve Shape Description

Tabla 2. CDC definidas en el estándar IEC61850.

1.1.6 Restricciones Funcionales (FC)

Los FC (Functional Constraint), facilitan la identificación de los atributos presente en cada objeto de datos, agrupando los DA según su característica, esto permite en caso un cliente consulte la posición de un interruptor, el FC “ST” mostrara el estado del interruptor sin mayor dificultad, dado que permite conocer el estado del dispositivo en tiempo real (ver figura 6). En la tabla 3 se muestran los FC presentes en la norma.

FC	DESCRIPCIÓN
ST	Estado de dispositivos
MX	Valores de medida
CO	Comandos de Proceso, control
SP	Puntos de Ajuste,
SV	Sustitución
CF	Configuración
DC	Descripción
SG	Grupos Ajustables
SE	Valores editables
CB	Circuit Breaker
SP	Parámetros de Salida
Ex	Valores externos

Tabla 3. Restricciones Funcionales de IEC61850.

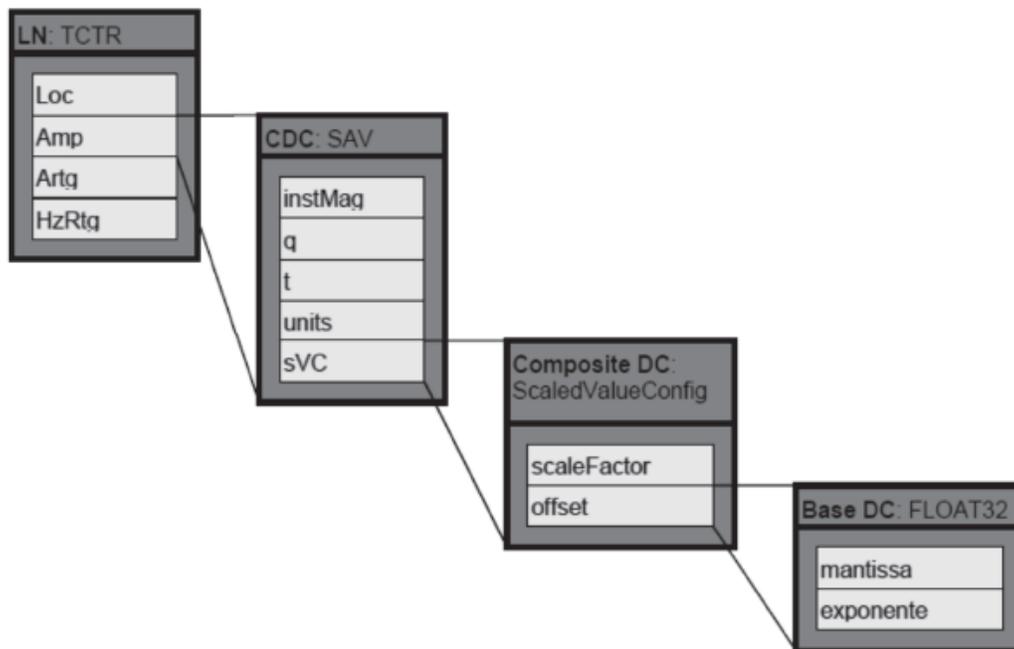


Figura 7. Ejemplo de modelado de datos.

Un nodo lógico tendrá agrupaciones que permitirán un orden en su estructura de datos, en la figura 7 se muestra un ejemplo del modelado de un nodo lógico de tipo transformador de intensidad (TCTR) formado por varios atributos. De estos atributos, Amp (valor instantáneo de intensidad) es una clase de datos compatible (CDC) de tipo SAV (Sampled Value) que está compuesto por un conjunto de atributos. El atributo sVC (escala) está formado por dos objetos (factor de escala y offset), este último construido con dos datos básicos (FLOAT32) que corresponden a una descripción (DC) dentro de una restricción funcional (FC).

La nomenclatura que un nodo lógico mostrara a un cliente en el momento de una consulta se muestra a continuación, tomando como ejemplo el nodo lógico de la figura 7 se tiene:

```
Rele1/TCTR$Amp$sVC$scaleFactor
```

Que se puede interpretar de la siguiente manera:

- Rele1: Nombre del dispositivo Lógico (LD)
- TCTR: Corresponde a un nodo lógico (LN) de transformador de instrumentación.
- Amp: Corresponde a la medida de corriente (DO).
- sVC: Atributo escala (DA)
- scaleFactor: Sub atributo de escala

1.1.7 Servicios de comunicación

El principal objetivo de la norma IEC 61850 es la automatización de las funciones de control y protección en una subestación. Para conseguirlo, es necesario que los elementos del sistema se comuniquen entre si intercambiando la información contenida en el modelo de datos. Así pues, los servicios de comunicación son los que especifican esta interacción, estructurada en el modelo de datos del apartado anterior.

Para que la información contenida en el modelo de datos y por ende en un nodo lógico pueda interactuar con el exterior y establecer comunicación con otros dispositivos, es necesario crear los servicios (Reportes, Goose, SV) que ayudaran a extraer la información desde un nodo lógico agrupando cada uno de sus datos en un conjunto de datos llamados Datasets.

1.1.8 Data sets o Conjuntos de Datos

IEC 61850 ha definido conjuntos de datos e informes de bloques de control para transmitir señales para propósitos de monitoreo, Una data set puede contener un único atributo de dato o diferentes datos que pertenezcan a uno o varios nodos lógicos. De este modo, una data set define un conjunto de referencias a datos.

Los data sets se utilizan en IEC 61850 para leer o escribir varios objetos o atributos de datos al mismo tiempo mediante un único mensaje. Los servicios de Reportes, GOOSE y SV se basan en el concepto de los data sets para la transmisión de información.

Los datos referenciados en una data set deben ser conocidos tanto por el servidor como por el cliente. En este caso, sólo se necesitará transmitir el nombre del Dataset y los valores de los datos, esto permite optimizar las comunicaciones.

La figura 8 muestra un Dataset que contiene información de los atributos de posición de un nodo lógico XCBR.

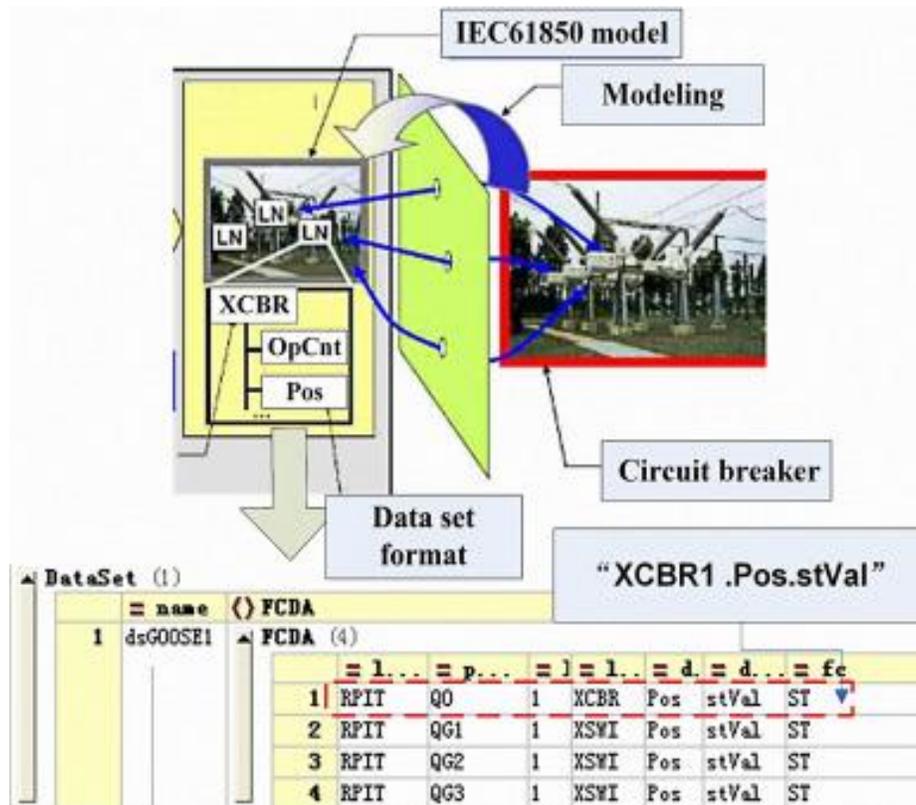


Figura 8. Dataset de posición de un XCBR.

Reglas generales para la configuración de Dataset:

- Todos los objetos de datos o sus atributos de datos se pueden seleccionar para un Dataset.
- Solo aquellos atributos de datos de un objeto de datos pueden ser seleccionados siempre y cuando tengan el mismo FC.
- Se pueden seleccionar objetos de datos con diferente FC para un Dataset. Por ejemplo, DO con FC =ST y DO con FC = MX pueden ser miembros en un conjunto de datos.
- Se puede seleccionar un solo atributo de datos cuando se especifica con una opción de activación. Por ejemplo, el atributo de datos stVal del objeto de datos Pos puede seleccionarse como miembro de un Dataset, porque se especifica con el cambio de datos de opción de disparo detectado (dchg), tal como está definido en la Figura anterior.

1.1.9 Reportes de Bloques de Control (URCB/BRCB)

Para poder transmitir las señales configuradas en un DataSet, debe haber un bloque de control de informe configurado para manejar y especificar cómo se transmiten los eventos a los clientes. Por tanto, para la transmisión de datos de un nodo lógico hacia el exterior se deberá crear un DataSet y posteriormente un Reporte de Bloque de Control. El diagrama de bloque mostrado en la figura 9 especifica el proceso para la comunicación y transmisión de datos.

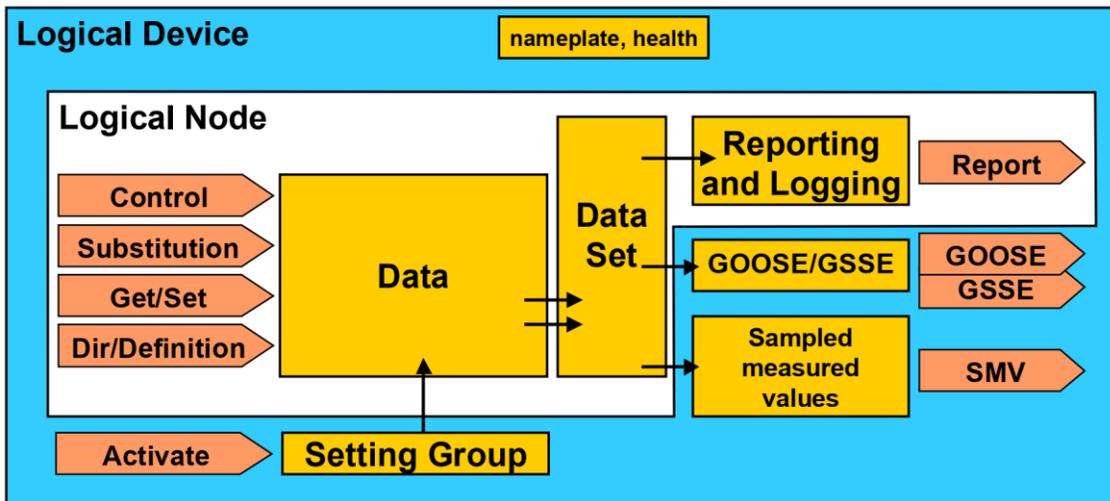


Figura 9. Bloque de transmisión de datos.

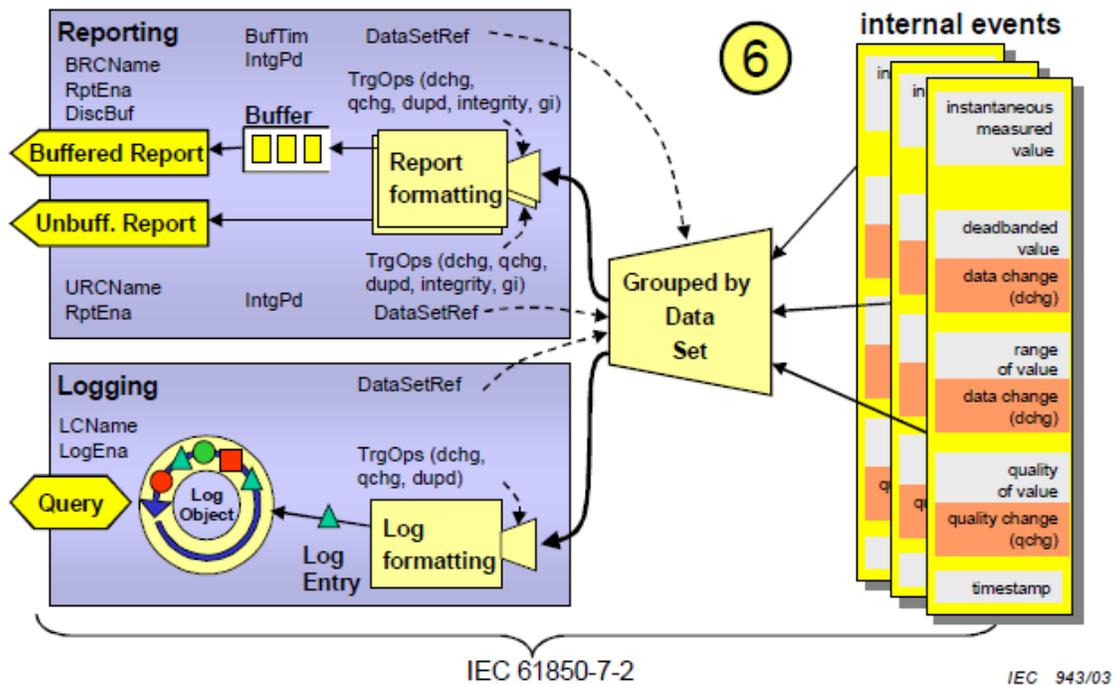


Figura 10. Modelo de transmisión de Datos Reportes con búfer y sin búfer.

Hay dos tipos de bloques de control de informes; *sin búfer* y *con búfer*. El bloque de control con búfer (informe almacenado) almacena los eventos durante una interrupción de comunicación, posteriormente cuando se reestablece la comunicación enviara los datos almacenados que fueron generados durante de la interrupción, seguidamente enviara los datos actuales, mientras que el bloque de control sin búfer (no almacenado) enviara únicamente los datos actuales al reestablecerse la comunicación, todos aquellos datos que fueron generados durante la interrupción no serán recuperados. La figura 10 muestra de manera ilustrativa el funcionamiento de los reportes con búfer y sin búfer.

Los modelos y servicios propios del estándar IEC 61850 se catalogan como modelos y servicios de tipo Abstract Communication Service Interface (ACSI por sus siglas en inglés) los cuales deben ser traducidos a modelos y servicios que empleen protocolos estándares, la norma emplea tres protocolos principales de comunicación, MMS, GOOSE y Sampled Values (Valores Muestreados) los cuales permiten el intercambio de información entre los distintos niveles de la subestación.

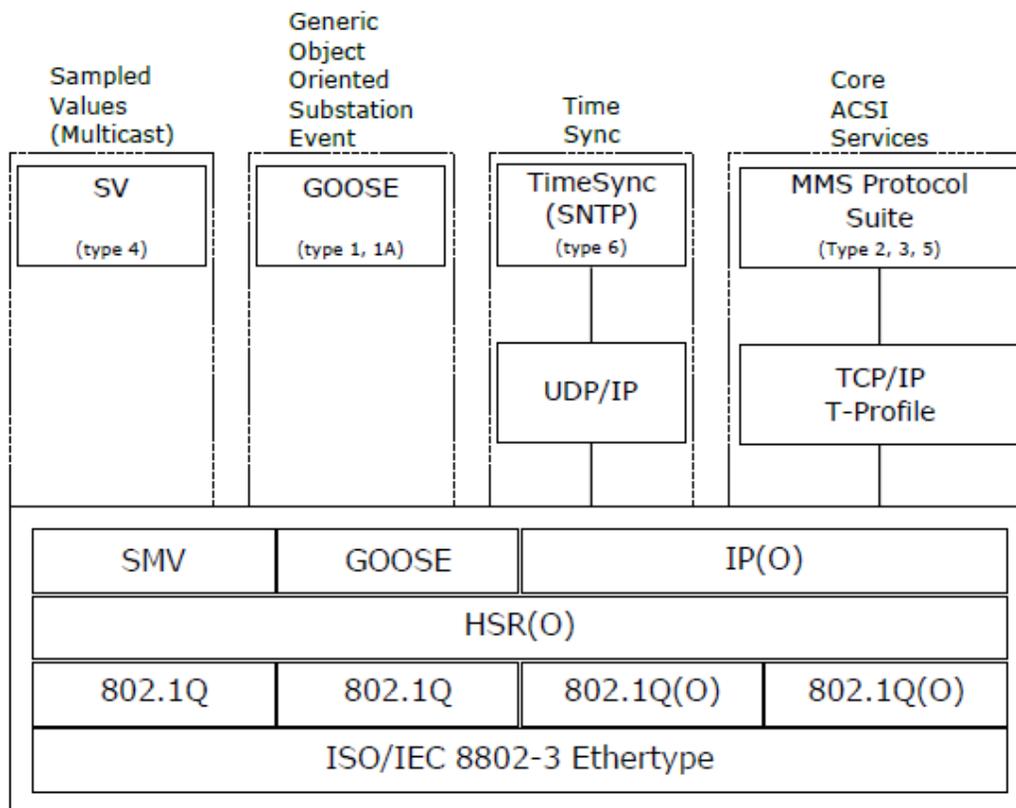


Figura 11. Descripción general de los mensajes y servicios de IEC61850.

En dichos protocolos existe una clasificación de mensajes los cuales dependerán de la naturaleza y prioridad que estos tengan en la red de comunicación, en la figura 11 se muestra esta clasificación y se describen a continuación:

- Tipo 1: Mensajes rápidos.
- Tipo 1A: Ordenes de disparos.
- Tipo 2: Mensajes de velocidad media.
- Tipo 3: Mensajes de velocidad baja.
- Tipo 4: Mensajes de valores análogos.
- Tipo 5: Mensajes de transferencia de archivos.
- Tipo 6: Mensajes de sincronización

El protocolo SNTP pertenece a servicios de sincronismos por lo que queda fuera del alcance de este documento. Los servicios presentes en la figura 11 se describen a continuación:

- **MMS (Manufacturing Message Specification):** La Especificación de Mensajes de Fabricación MMS, entrega los requerimientos necesarios para su transmisión en tiempo real de los datos que se definen en el IEC 61850-7 y que se asignan a protocolos especificados en IEC 61850-8, los cuales se basan en el modelo OSI que incluye TCP/IP y Ethernet para realizar también funciones de automatización. La correspondencia de los objetos del IEC 61850 a MMS determinan un método para transformar la información del modelo en un objeto variable y es el responsable de lograr la interoperabilidad a través de las funciones implementadas por distintos fabricantes.
- **Mensajería GOOSE:** La información que se debe gestionar y que tienen que dársele la mayor prioridad como por ejemplo una acción de disparo para protección, el accionamiento de un disyuntor, etc. Debe atenderse con la máxima rapidez y establecer por lo tanto un medio de comunicación de muy alto rendimiento, estas comunicaciones los brinda los entorno del IEC 61850, mediante GOOSE (Generic Object Oriented Substation Events), que corresponde a los mensajes que se transmiten, entre IED's esto es en forma horizontal en los niveles de bahía, e identifican a los eventos que se puedan dar en el proceso, se define en el IEC 61850-1, como una transferencia de datos en tiempo critico orientada a eventos.

El servicio GOOSE es la herramienta que se utiliza para transmitir datos que lleven información de estado, información de medidas, información de control, información que será distribuida a la red de tal forma que otros dispositivos hagan uso de la información enviada como mensajes multidifusión (multicast), esto es, no están dedicados a ningún dispositivo receptor en particular si no que lo reciben todos los dispositivos del bus de red, haciendo uso de la información el dispositivo que lo requiera, esta actividad de mensajería se realiza varias veces por lo que la probabilidad de que algún dispositivo haga uso de las misma es mayor.

- **Sampled Values (SV):** El servicio de transmisión SV, Sampled Values en inglés, es utilizado cuando se necesita transmitir señales analógicas de campo, tales como intensidad, tensión o cualquiera de sus derivados, utilizando comunicaciones

digitales. Las señales analógicas son muestreadas y transmitidas. Tanto el servicio GOOSE como SV se encapsulan sobre capa 2 del modelo OSI Ethernet por lo que la transmisión de estos protocolos en una red de comunicación será más rápida en comparación al MMS ya que este último como se ha mencionado anteriormente está basado en el modelo OSI tradicional y utiliza un modelo Cliente/Servidor para transferencia de datos.

La figura 12 se observa una comparación entre IEC61850 y el modelo OSI.

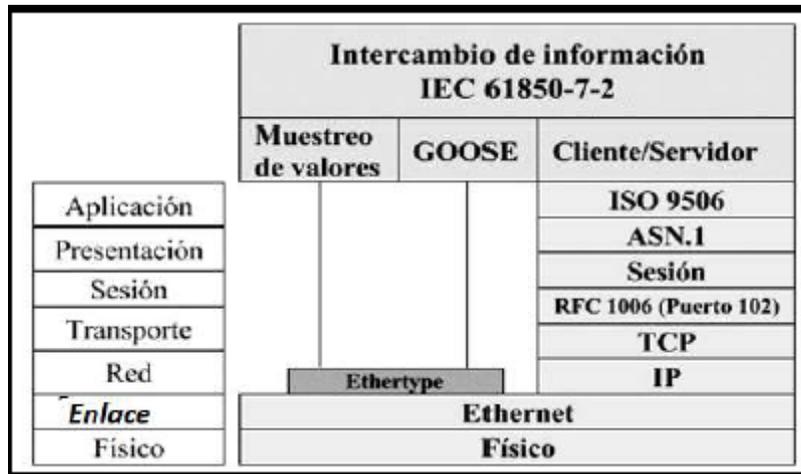


Figura 12. Modelo OSI y el Estándar IEC61850.

Como se ha descrito en este apartado los servicios o protocolos que trae consigo el estándar son los que permitirán una correcta comunicación en los niveles de un SAS por un lado tendremos los SV que serán los encargados de obtener y transmitir señales analógicas de los instrumentos en patio, a su vez contaremos con mensaje GOOSE encargados de transmitir señales digitales tanto de nivel de proceso como de nivel bahía, finalmente tendremos los servicios MMS que permitirán recibir y descifrar de una manera sencilla la información transmitida de nivel de bahía a nivel de estación.

1.1.10 Lenguaje SCL

Una de las principales ventajas de la norma IEC 61850 es la definición de un lenguaje completo de configuración a nivel de subestación. Este lenguaje, conocido como SCL (Substation Configuration Language), está basado en el estándar XML (extended mark up language) y permite describir las configuraciones y parámetros de comunicaciones de un IED y de la red IEC 61850 de la instalación.

Este lenguaje define un formato capaz de describir la ingeniería de un sistema de automatización de subestaciones, proporcionando una descripción estandarizada de:

- Funcionalidad del sistema de automatización.
- Estructura lógica de la comunicación del sistema.

- Relación entre los equipos y sus funciones de instrumentación.

El objetivo principal del lenguaje SCL es el intercambio interoperable de los datos de ingeniería en la subestación entre las herramientas de ingeniería de los distintos fabricantes. Este modelo también permite obtener una configuración automatizada de las funciones y de las comunicaciones, así como la comprobación del funcionamiento del sistema. Para poder proporcionar esta interoperabilidad es necesario:

- Una descripción formal del sistema de automatización de la subestación, incluyendo todos los enlaces de comunicación.
- Describir sin ningún tipo de ambigüedad las capacidades de los dispositivos IED's.
- Descripción de los servicios de comunicación aplicables.
- Descripción formal de la relación entre la instalación de distribución y los datos del sistema de automatización.

El lenguaje SCL establece cuatro archivos:

- 1) El archivo **SSD** (System Specification Description): En el cual constan las especificaciones del sistema y también describe el diagrama unifilar de la subestación y los LN requeridos.
- 2) El archivo **ICD** (IED Capability Description): Describen las capacidades de los IEDs, incluye también el soporte de la información de los LN y GOOSE.
- 3) El archivo **SCD** (Substation Configuration Description): Refieren también al detalle de la configuración de la subestacion. Contiene todos los IED's, una sección de configuración y una de subestación.
- 4) El archivo **CID** (Configured IED Description): Que detalla las configuraciones y descripciones completas de los IED, archivos que actuaran según los diferentes requerimientos por lo cual están construidos bajo los mismos formatos.

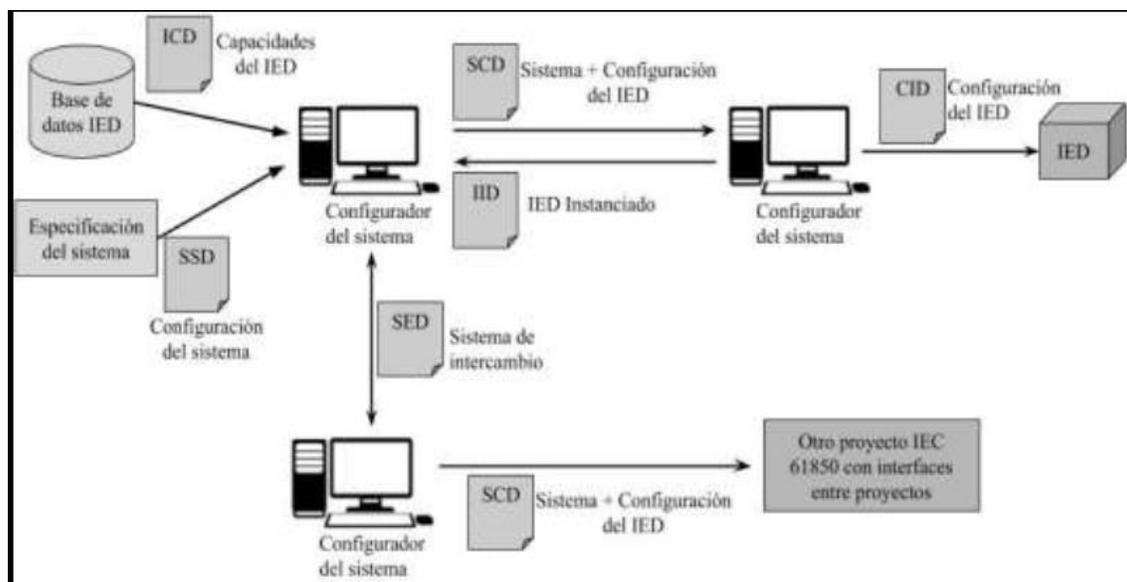


Figura 13. Proceso de ingeniería en subestaciones.

La figura 13 muestra el proceso de ingeniería para la configuración de una subestación y cada uno de sus dispositivos, en el cual al hacer la ingeniería de proceso obtenemos el archivo ICD de cada relé de protección el cual contiene la base de datos completa de variables análogas, digitales de control y medición de IEC 61850.

Posteriormente con una herramienta de configuración se crea el archivo SCD el cual contiene los archivos necesarios y la configuración para el mapeo cruzado de publicador y receptor.

Finalmente se obtiene el archivo CID, el cual contiene la configuración para los servicios proporcionados por IEC 61850 (Mensajes MMS, GOOSE y Sampled Values).

En resumen, el archivo ICD (contiene la información del relé) esta previamente instalado en el IED y con las herramientas de cada fabricante (Siemens, ABB, Schneider, etc.) se transforma a archivos SCD (contiene la información de equipos de la subestacion) para posteriormente descargar los archivos CID (configuración del relé) a los IED's correspondientes. La figura 14 muestra los pasos enumerados para el proceso de configuración de archivos SCL.

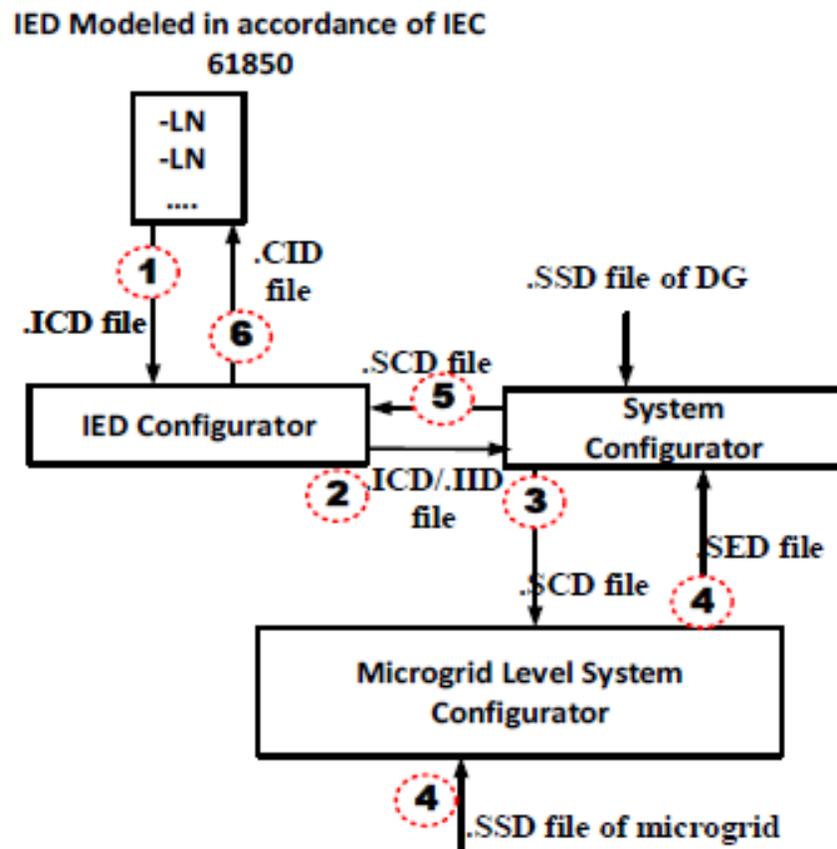


Figura 14. Proceso de configuración de archivos SCL.

1.2 Equipos y software en una infraestructura de simulación/emulación en un laboratorio de aprendizaje de la norma IEC61850

En esta sección del documento se recopila información de características y configuraciones básicas de softwares con licencia de prueba o gratuitos útiles para la configuración de escenarios de pruebas bajo el estándar IEC61850, entre los programas que se utilizan para propósitos educativos del aprendizaje de la norma se tienen: software de configuración de IED's, simuladores, IED's virtuales, clientes entre otros.

1.2.1 61850 ICD EDITOR (Editor de archivos ICD)

Software de prueba de la empresa Infotech, se utiliza para la creación y edición de archivos ICD, entre sus funciones están:

- Creación de archivos ICD.
- Adicionar o editar LD (Dispositivos lógicos).
- Adicionar o editar LN (Nodos lógicos).
- Adicionar o editar Datasets.
- Adicionar o editar Control Blocks.
- Establecer valores de atributos de datos.
- Edición de archivos en XML.
- Validación de archivo editado con esquema SCL.

En la figura 15 se observa la interfaz principal del software, en este espacio se crean los archivos ICD que incluyen dataset, dispositivos lógicos, nodos lógicos y bloques de control.

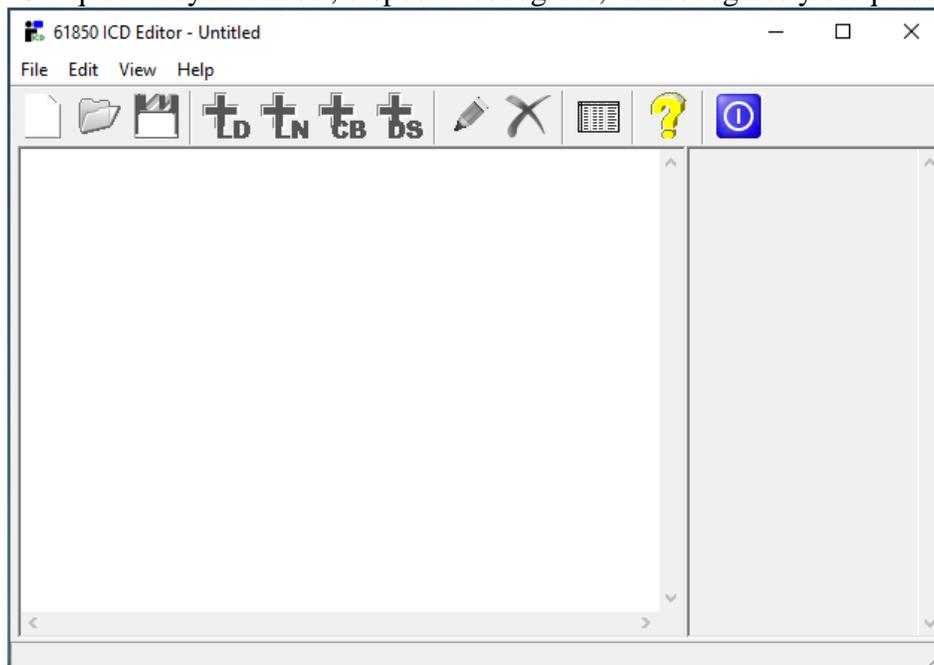


Figura 15. 61850 ICD Editor.

A continuación, se describen los elementos que se encuentran en la barra de herramientas:



Para la creación de un nuevo archivo ICD hacer clic en el botón “new”, lo cual abrirá la ventana mostrada en la figura 16.

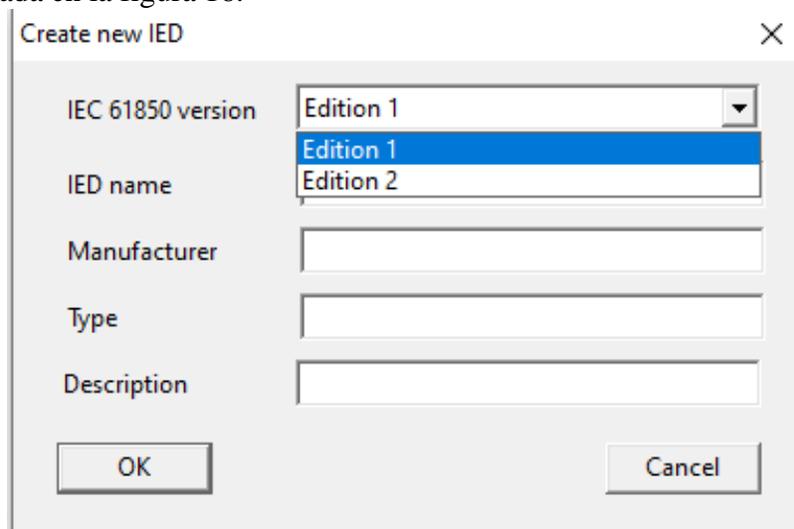


Figura 16. Creación de un ICD.

Tal como se observa en la figura anterior el software permite crear archivos ICD ya sea en edición 1 o edición 2 según las características del IED a configurar. También permite editar el nombre del IED, el fabricante, el tipo y una pequeña descripción, estos últimos ítems no son obligatorios y solo sirven para información general del dispositivo a configurar.

Luego de rellenar los campos de información del archivo ICD, hacer clic en OK, esto abrirá una nueva ventana la cual permite crear un dispositivo lógico tal como se muestra en la figura siguiente.

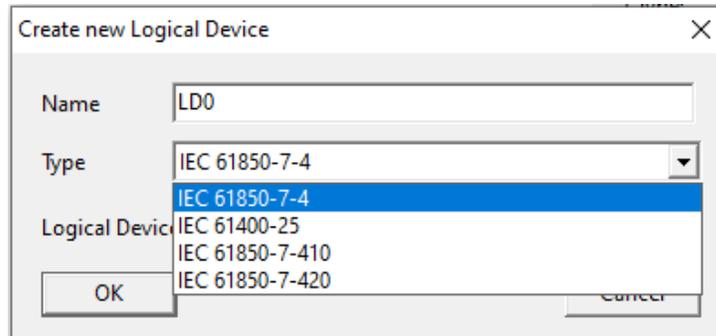


Figura 17. Creación de un Dispositivo Lógico.

El software permite crear dispositivos lógicos para diferentes aplicaciones, el presente documento únicamente está orientado a la configuración del estándar en subestaciones eléctricas por tanto en “Type” se selecciona IEC 61850-7-4 el cual corresponde a subestaciones eléctricas tal como se muestra en la figura 17.

Luego de asignar el nombre y el tipo del dispositivo lógico hacer clic en OK, finalmente se crea el nuevo archivo ICD el cual contiene en el dispositivo lógico, dos nodos lógicos obligatorios (LLN0 y LPHD1) tal como se muestra en la figura 18.

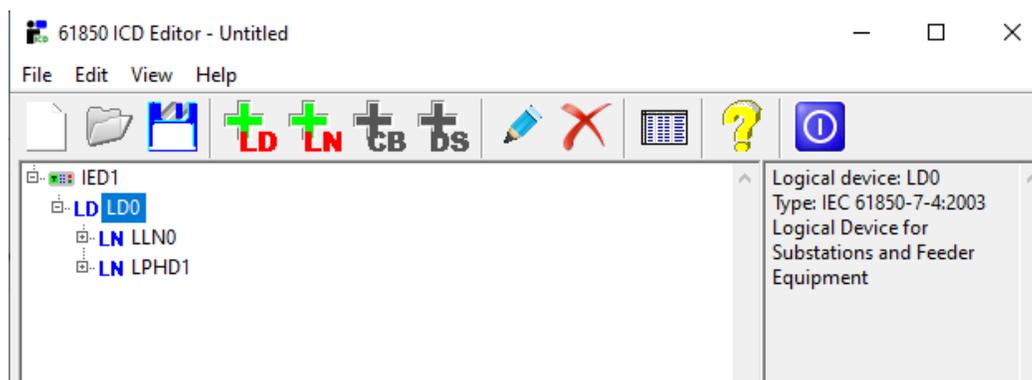


Figura 18. Archivo ICD.

En caso de agregar más dispositivos lógicos hacer clic en LD (Logical Device) el proceso a seguir para la creación de un dispositivo lógico es el que se ha descrito anteriormente. Si se

desea agregar más nodos lógicos hacer clic en LN (Logical Node), esto abrirá la ventana mostrada en la figura 19.

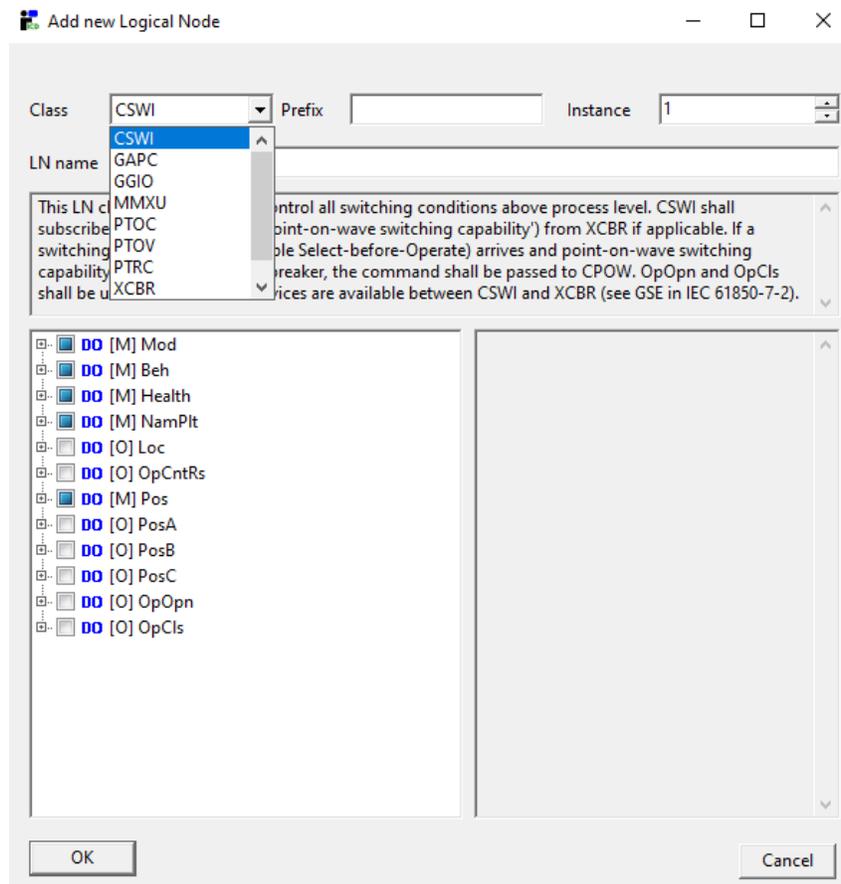


Figura 19. Creación de un Nodo Lógico.

Como se observa en la figura anterior el software solo permite crear 8 nodos lógicos en su versión de prueba, dependiendo de las necesidades del IED a configurar se selecciona el nodo lógico adecuado, se debe asignar también un nombre y un prefijo, el programa también permite crear un número de instancia, al terminar de editar el nodo lógico hacer clic en OK.

Para la creación de un Dataset se selecciona el nodo lógico LLN0 (nodo lógico asignado para la configuración de servicios de comunicación del IED), una vez ubicado en LLN0 hacer clic en DS, se abrirá la ventana de la figura 20.

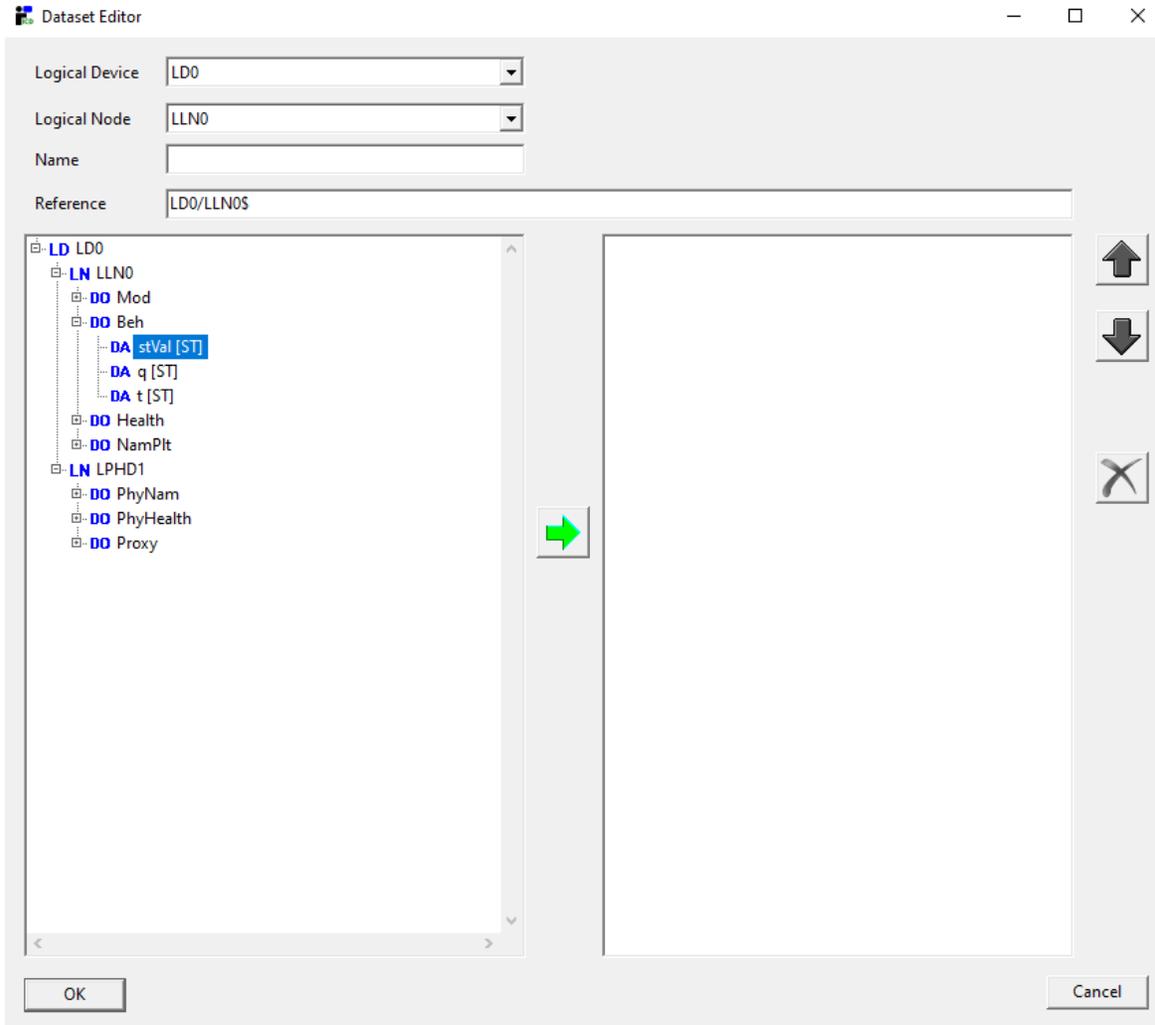


Figura 20. Creación de DataSet.

La imagen anterior muestra la ventana para la creación de un DataSet, después de asignarle un nombre, se debe seleccionar el atributo del nodo lógico que se colocara en el DataSet, una vez seleccionado el atributo hacer clic a la flecha que se encuentra al centro.

Para asignar más atributos al DS se repite el proceso mencionado en el párrafo anterior, finalmente hacer clic en OK.

Para la creación de un bloque de control hacer clic en CB, esto abrirá la ventana que se muestra en la figura 21, se debe tener en cuenta que al igual el DS, el bloque de control se configura en el nodo lógico LLN0.

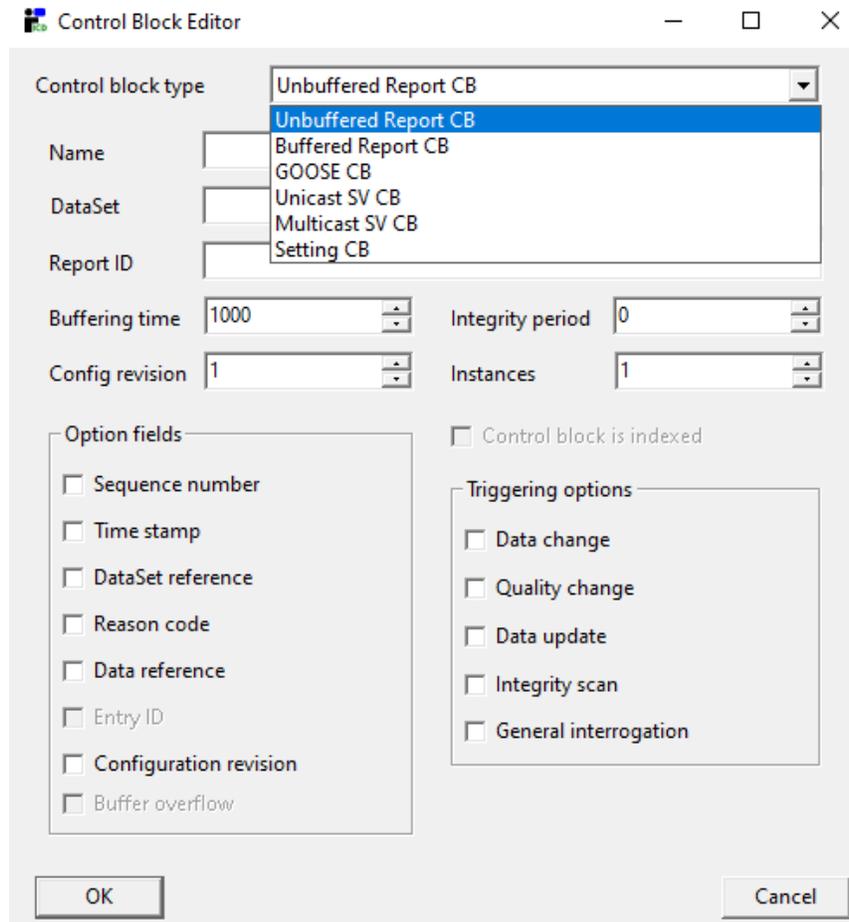


Figura 21. Creación de un Bloque de Control.

Tal como se muestra en la figura anterior el programa permite crear reportes búferizados y no búferizados, bloques de control GOOSE, Sampled Values y de configuración, en el capítulo 2 se desarrollan las configuraciones de un CB de tipo GOOSE.

1.2.2 ReLab OPC Console (cliente)

Software de la empresa Relabsoft, ofrece funcionalidades de comunicación a través de *Open Protocol Communication, OPC* (tecnología de comunicación con arquitectura cliente/servidor) para distintos protocolos como modbus, 101,104, MMS y IEC61850. Las funciones donde se aplica el estándar IEC 61850 son las siguientes:

- Mapeo de mensajes GOOSE de un IED.
- Envío y recepción de eventos, comandos, alarmas, indicaciones y mensajes mediante el protocolo GOOSE.
- Colección y procesamiento de datos MMS.
- Lectura y control de equipos a partir de archivos ICD, CID, SSD y SCD.

La figura 22 muestra la interfaz principal del software, sirviendo como cliente de un relé simulado, del lado izquierdo se encuentra el modelo de datos del IED, a la derecha se encuentra el mapeo de los TAGs (etiqueta o segmento de código) del relé simulado. En la parte inferior se encuentra una ventana de consola que brinda información acerca de las acciones realizadas, las conexiones y errores que ocurran.

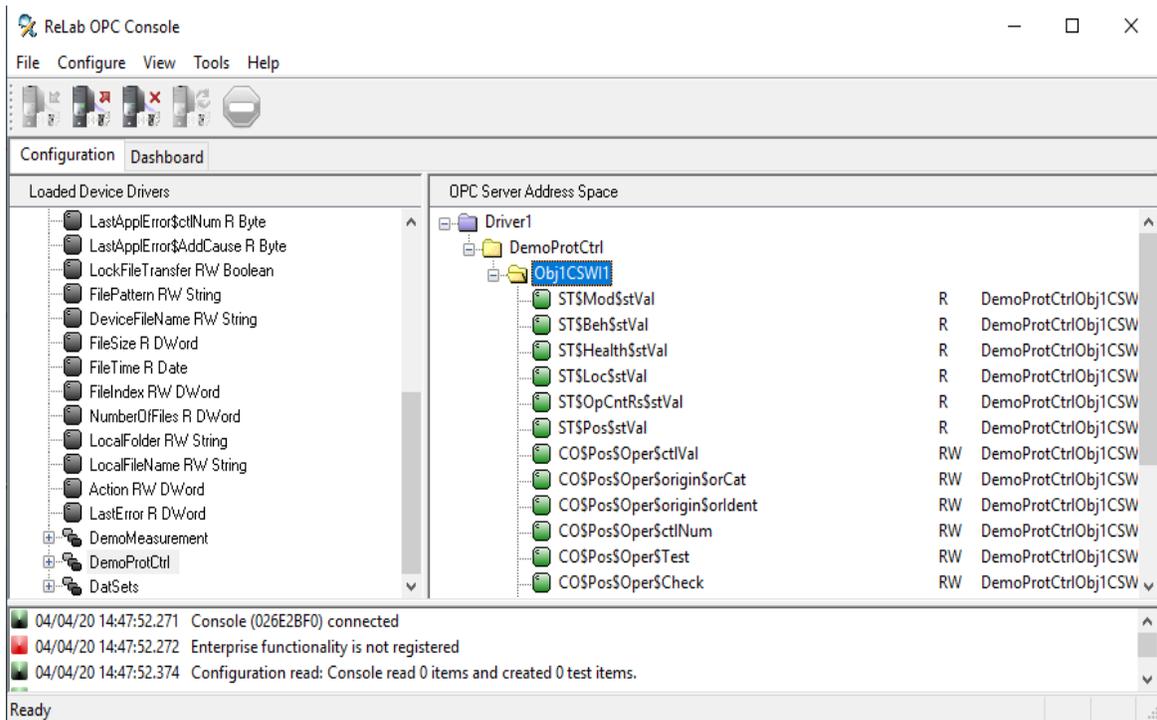


Figura 22. ReLab OPC Console.

La figura 23 muestra la pestaña dashboard en la que se encuentran los TAG que se han agregado, se visualiza su nombre completo, nombre del tag, valor, estampa de tiempo, calidad, tipo de acceso (solo lectura, o lectura y escritura), tipo de dato y descripción. Permite monitoreo en tiempo real para tipos de datos de lectura, y puede realizarse control en los casos de TAG's de tipo CO.

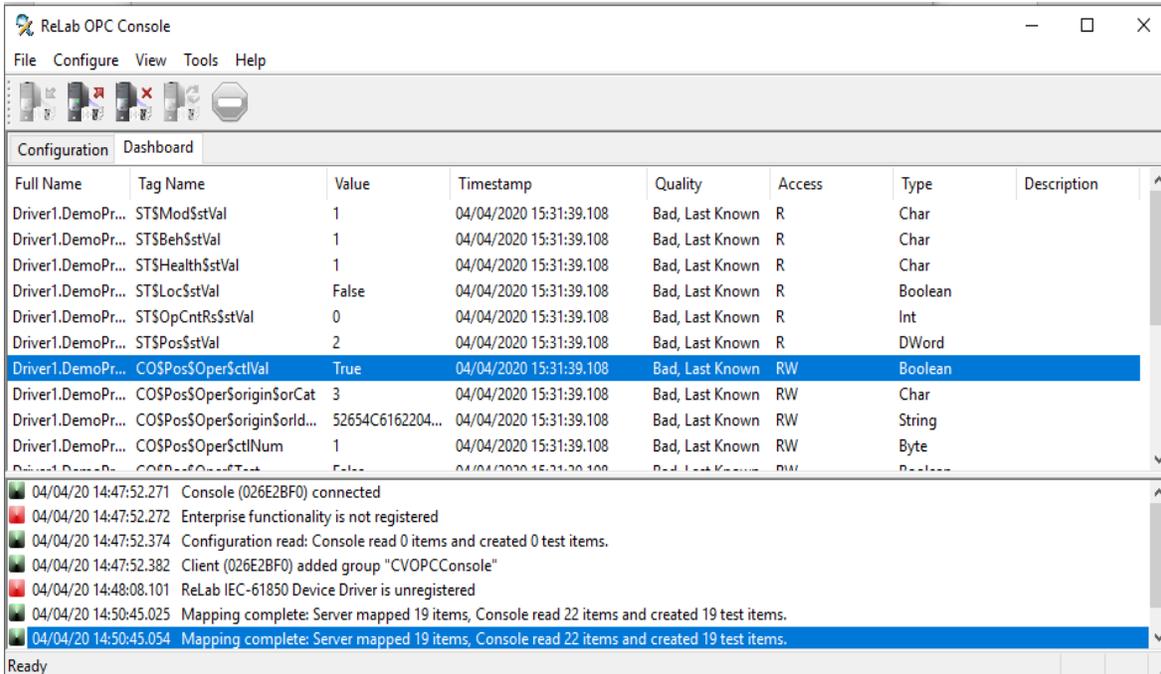


Figura 23. Pestaña dashboard.

Para la creación de un driver bajo el estándar IEC61850 hacer clic en la pestaña “configure” del software Relab OPC tal como se muestra en la figura 24, seguidamente hacer clic en *LoadDriver* y aparecerá la ventana mostrada en la figura 25.

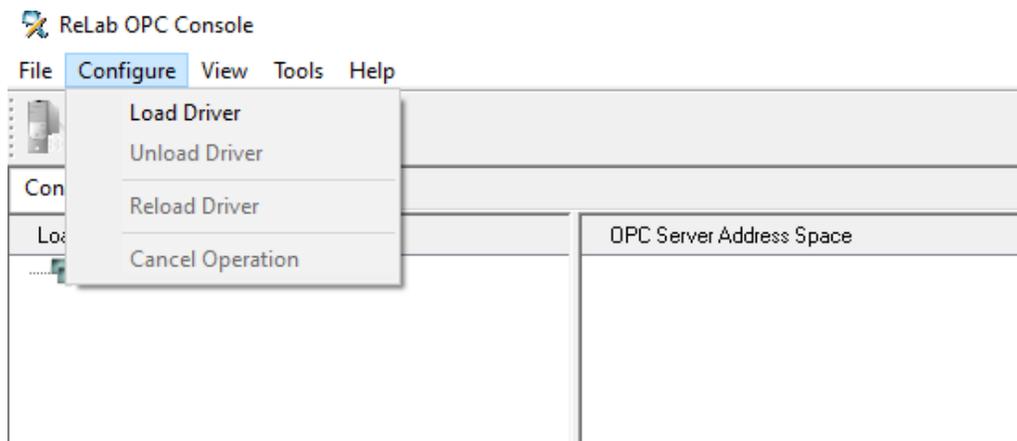


Figura 24. Creación de un Driver en Relab.

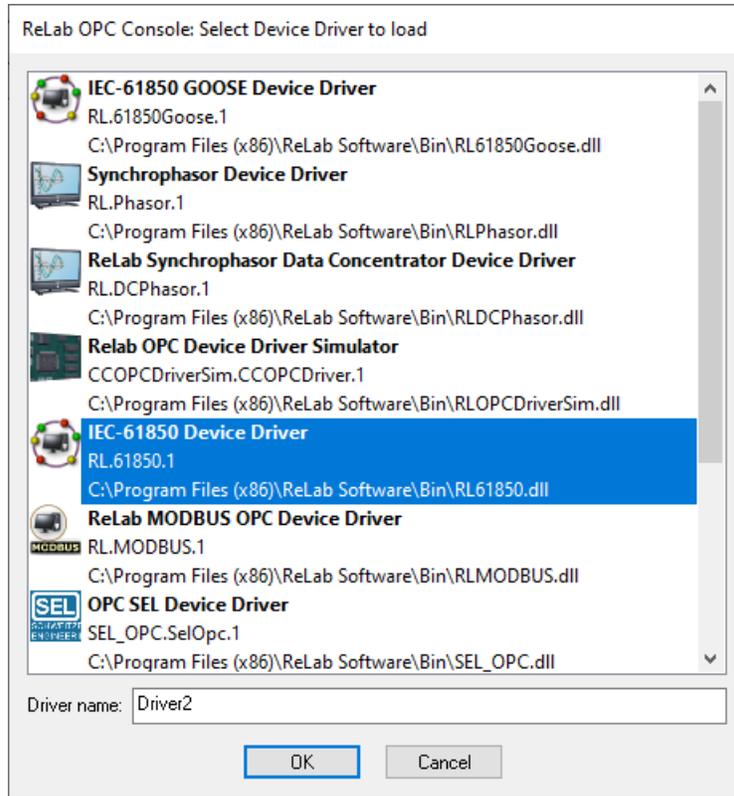


Figura 25. Selección del driver para un dispositivo IEC61850.

Como se observa en la figura 25 el software permite crear Drivers para diferentes protocolos, se selecciona el Driver para IEC61850 y se asigna un nombre a dicho driver “Driver2”, luego de nombrar el driver hacer clic en OK, seguidamente se abrirá la ventana mostrada en la figura 26.

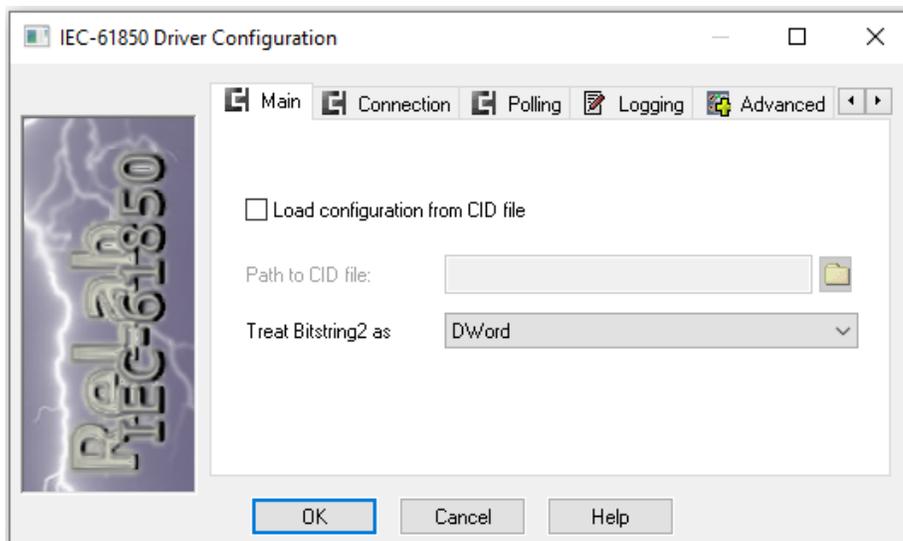


Figura 26. Configuración del Driver IEC61850.

Relab permite exportar archivos CID para visualizar los nodos lógicos, dataset y reportes del CID, para ello se hace clic en *Load configuration from CID file*, esto habilitará la opción para exportar el archivo CID, luego de exportar el archivo hacer clic en OK.

Si se desea leer los datos de un IED conectado a la red hacer clic en OK sin realizar ninguna modificación, aparecerá la ventana mostrada en la figura 27, se escribe la dirección IP del IED y el puerto TCP, el estándar define el puerto 102 por defecto, luego de ingresar los datos hacer clic en OK y el software buscará al dispositivo conectado en la red, una vez el software detecte el IED aparecerá el modelo de datos que contiene el dispositivo tal como se muestra en la figura 28.

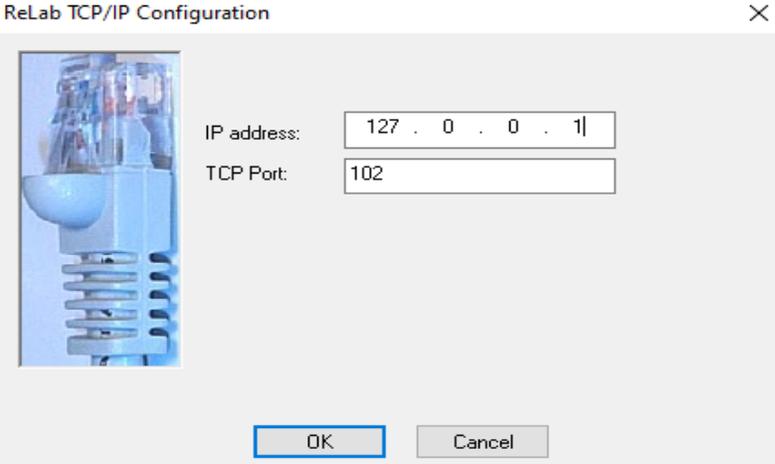


Figura 27. Configuración TCP/IP.

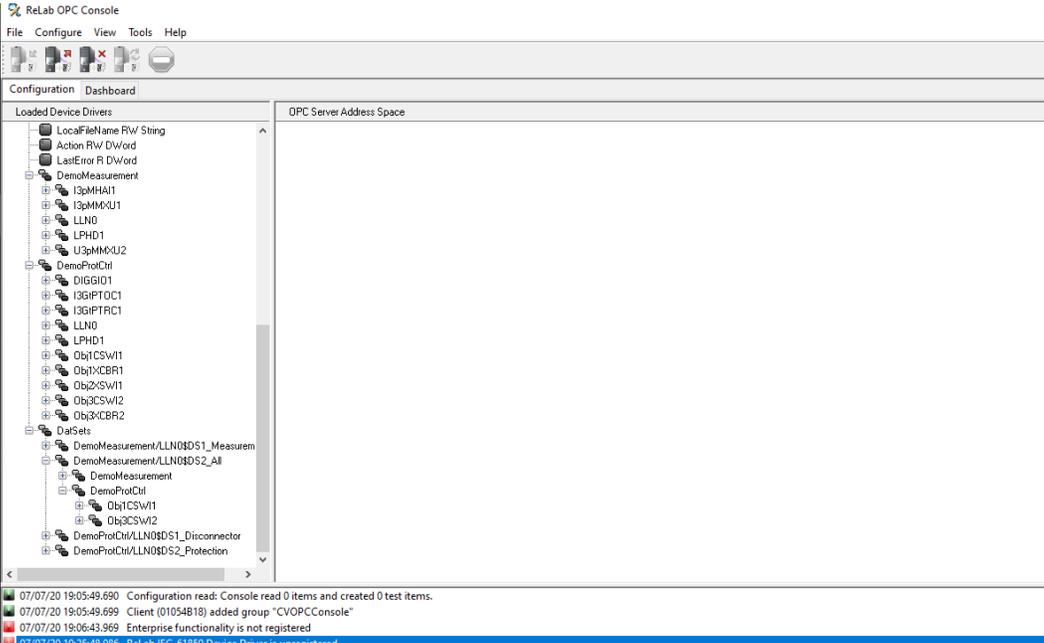


Figura 28. IED conectado a Relab OPC.

Para mapear los datos del IED seleccionar el nodo lógico que contiene los tags de interés, hacer clic derecho y seleccionar *Auto Map Recommended* para que los datos aparezcan en el Dashboard del Relab tal como se muestra en la figura 29.

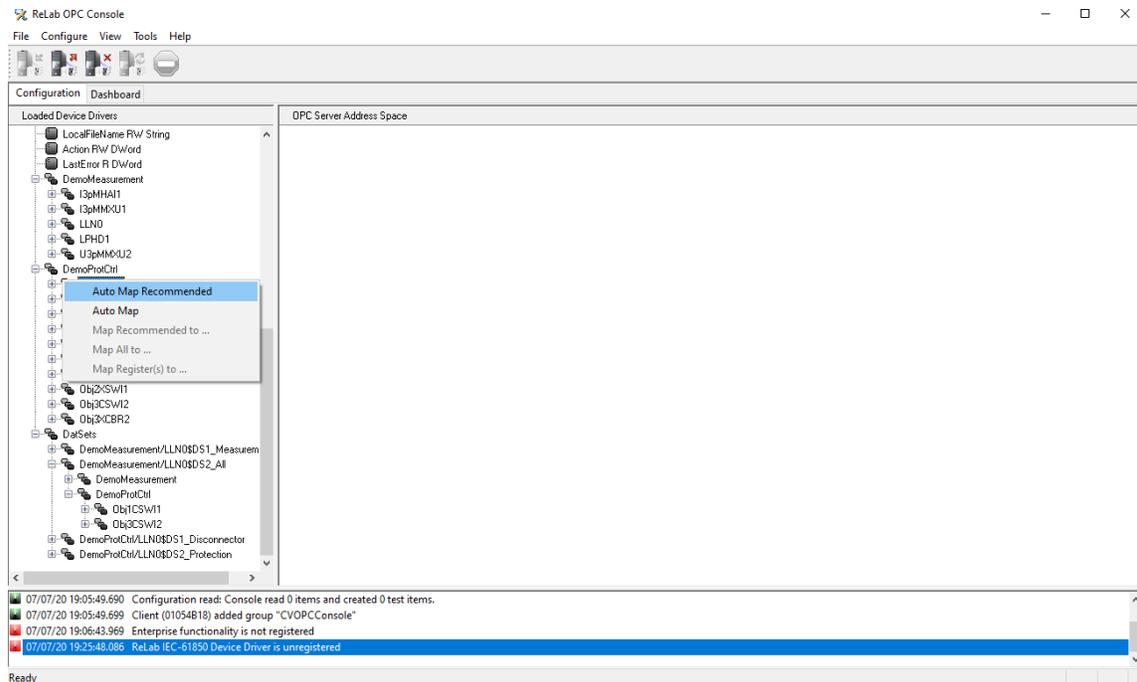


Figura 29. Mapeado de Tags del IED.

Para continuar agregando más tags al Dashboard se seguirá el mismo proceso descrito anteriormente, una vez se tengan los datos de interés se podrá visualizar en tiempo real el estado de los atributos del IED tal como se ha mostrado en la figura 23.

1.2.3 IEDexplorer (Cliente)

Software gratuito para pruebas y propósitos educativos, consiste en una herramienta cliente bajo el estándar IEC61850 utilizando el protocolo de comunicación MMS, el software permite la conexión con una interfaz amigable a los dispositivos simulados o conectados a la red. La figura 30 muestra la interfaz gráfica principal, del lado izquierdo en la pestaña IEDView (MMS) aparece el modelo de datos de los IED's conectados en la red, en el cual se podrán efectuar comandos de control o leer los estados de los atributos del IED.

En la ventana IEDview se visualizan los nodos lógicos, dataset, mediciones, comandos entre otros. Proporciona información como el nombre, valor actual y el tipo de dato, los valores monitoreados pueden ser leídos en tiempo real.

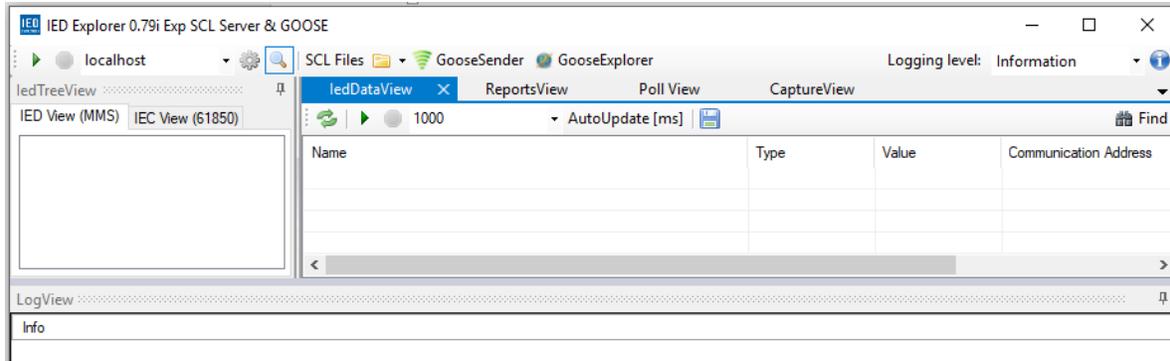


Figura 30. Interfaz gráfica IEDexplorer.

A continuación, se muestran las funciones utilizadas durante las pruebas realizadas en este documento.

1.2.3.1 Conexión a dispositivos IED's IEC61850

Para la conexión de un IED conectado a la misma red primero se deben configurar los parámetros del IED, la figura 31 muestra la ventana de configuración, donde se coloca la IP, el puerto TCP entre otros parámetros, dependiendo del IED puede requerir información de autenticación.

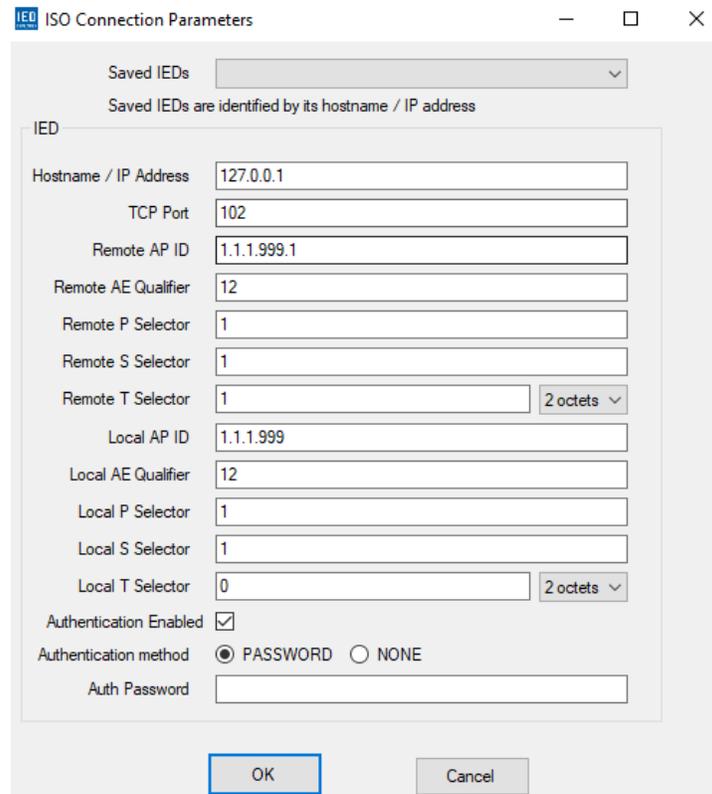


Figura 31. Configuración de IEDExplorer.

1.2.3.2 Monitoreo de atributos vía MMS y estructuras de árbol

Una vez configurados los parámetros de conexión, hacer clic al botón OK, el programa iniciara la conexión al IED mostrando en forma de diagrama de árbol su modelo de datos, así como sus nodos lógicos, objetos y atributos, tal como se muestra en la figura 32. En la ventana del lado izquierdo (IEDTreeView), se tienen dos opciones para la visualización del IED conectado.

IEDView MMS, el cual muestra los datos del IED de manera ordenada, es decir, en vista MMS los reportes, dataset y datos del nodo lógico están separados según su categoría al que pertenecen, mientras que en IECView 61850 los datos aparecen como la norma lo establece, es decir, en esta opción se visualiza únicamente los nodos lógicos existentes en el IED, los reportes, datos y dataset se encuentran en el nodo lógico de configuración LLN0. Tal como se muestra en la figura 42.

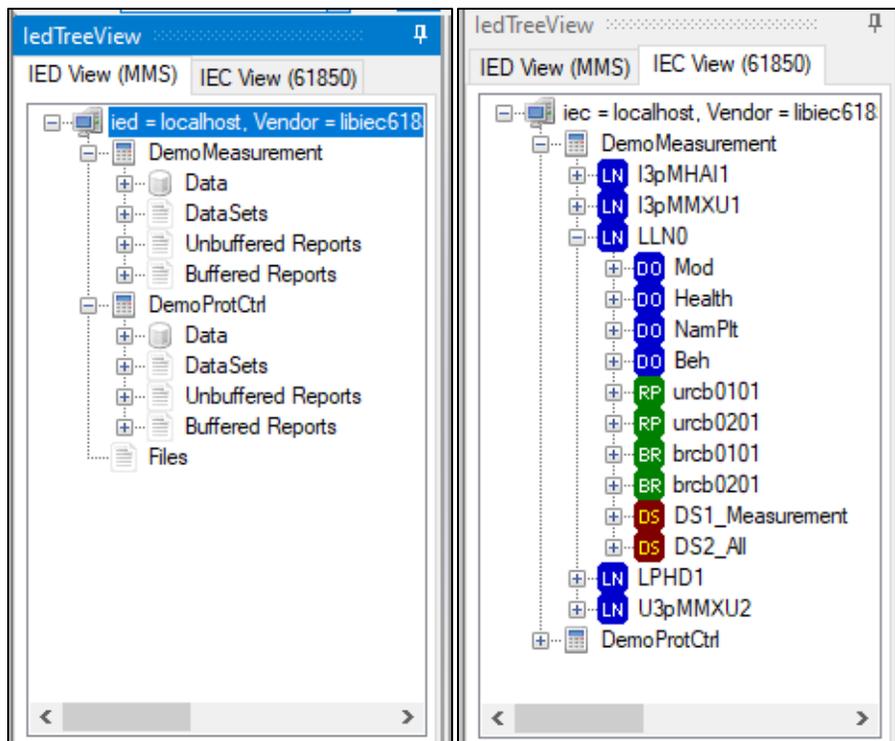


Figura 32. Vista de Datos del IED.

Para monitorear atributos, seleccionar en el diagrama de árbol el objeto o atributo que va a monitorearse y del lado derecho de la pantalla aparecerá el nombre del o los atributos, su tipo de dato y el valor, la figura 33 muestra un ejemplo y muestra subrayado en rojo el botón de auto actualizar y el intervalo de actualización.

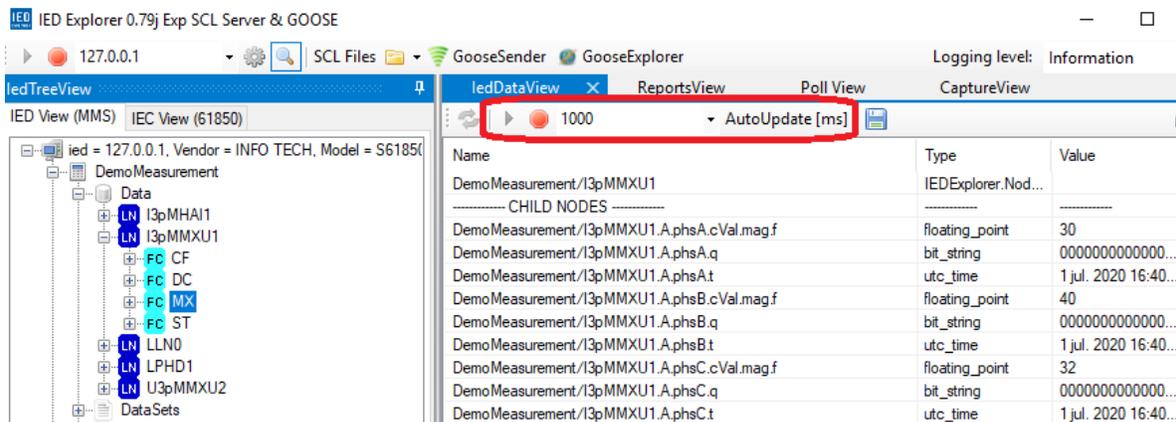


Figura 33. Ventana de monitoreo de atributos.

1.2.3.3 Envío de comandos MMS y Lectura/escritura de valores

La figura 34 muestra la ventana de comandos MMS, esta aparece al hacer clic derecho en el atributo al cual se desea enviar el comando, dicho atributo debe ser de tipo control (CO). Una vez abierta la ventana seleccionar el valor del comando y se verifica la sincronización del tiempo, también se selecciona en caso sea un comando Select Before Operate (SBO).

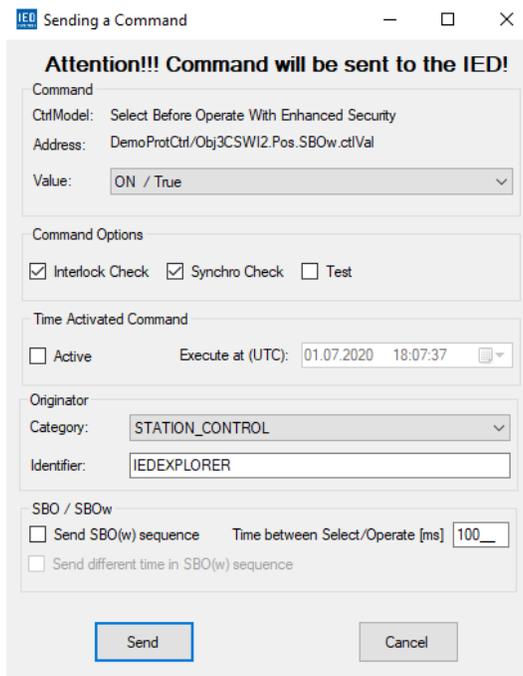


Figura 34. Ventana de envío de comando MMS.

Entre la información que se envía al IED está la categoría, la cual sirve para identificar de que parte de la subestación procede el comando, también se envía un identificador, esto para determinar si el comando es válido, entre las categorías están:

- BAY_CONTROL
- STATION_CONTROL
- REMOTE_CONTROL
- AUTOMATIC_BAY
- AUTOMATIC_STATION
- AUTOMATIC_REMOTE
- MAINTENANCE
- PROCESS

1.2.3.4 Explorar mensajes GOOSE

Para abrir un explorador GOOSE seleccionar la opción GooseExplorer ubicado en barra de herramientas (subrayado de color verde), se abrirá una pestaña donde se selecciona el adaptador de red a utilizar y se obtiene el número de IED's identificados, su respectiva dirección MAC y la cantidad de mensajes GOOSE recibidos, tal como se muestra seleccionado de color rojo en la figura 35. Una vez se reciban los mensajes GOOSE aparecerán organizados como diagramas de árbol (subrayado de color azul), del lado derecho encerrado en amarillo aparecerán los datos del mensaje, nombre, tipo y valor. En la parte inferior de la ventana se tiene un gráfico con el valor vs el tiempo del mensaje seleccionado.

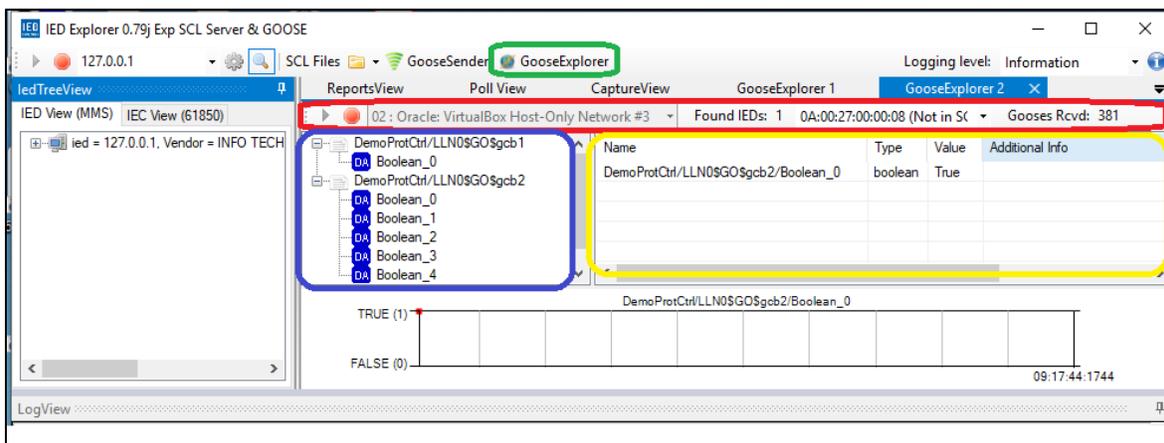


Figura 35. Goose Explorer.

Otra funcionalidad del software es exportar la configuración de los mensajes GOOSE como un archivo XML, esto se realiza al hacer clic derecho sobre el mensaje GOOSE o su conjunto, se selecciona la opción “export data from selected database” o “export goose including this dataset” y seleccionamos la ruta de destino del archivo XML.

1.2.3.5 Envió de mensajes GOOSE

La figura 36 muestra la ventana para el envío de mensajes GOOSE para abrir la aplicación presionar el botón enmarcado en verde, seleccionar el adaptador de red a utilizar y al ejecutarlo será posible ya sea crear un mensaje GOOSE desde cero o cargar las configuraciones GOOSE desde un archivo XML.

La figura 36 muestra enmarcado en azul los parámetros necesarios para el envío del mensaje, al presionar en Edit se podrá modificar los valores que se envían, y con el botón run/stop se comenzaran a enviar los mensajes.

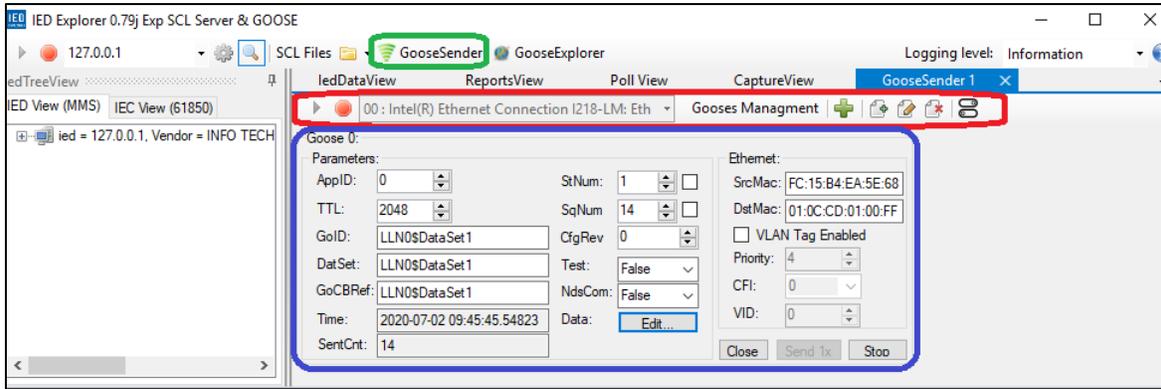


Figura 36. GOOSE sender.

Otra opción que incluye el software es la lectura de reportes con bufer o sin bufer, para visualizarlos estos deben ser habilitados desde IEDExplorer, Cuando los reportes sean habilitados serán mostrados en la sección ReportsView tal como se muestra en la figura 37, estos aparecen cuando exista un cambio de medida o estado.

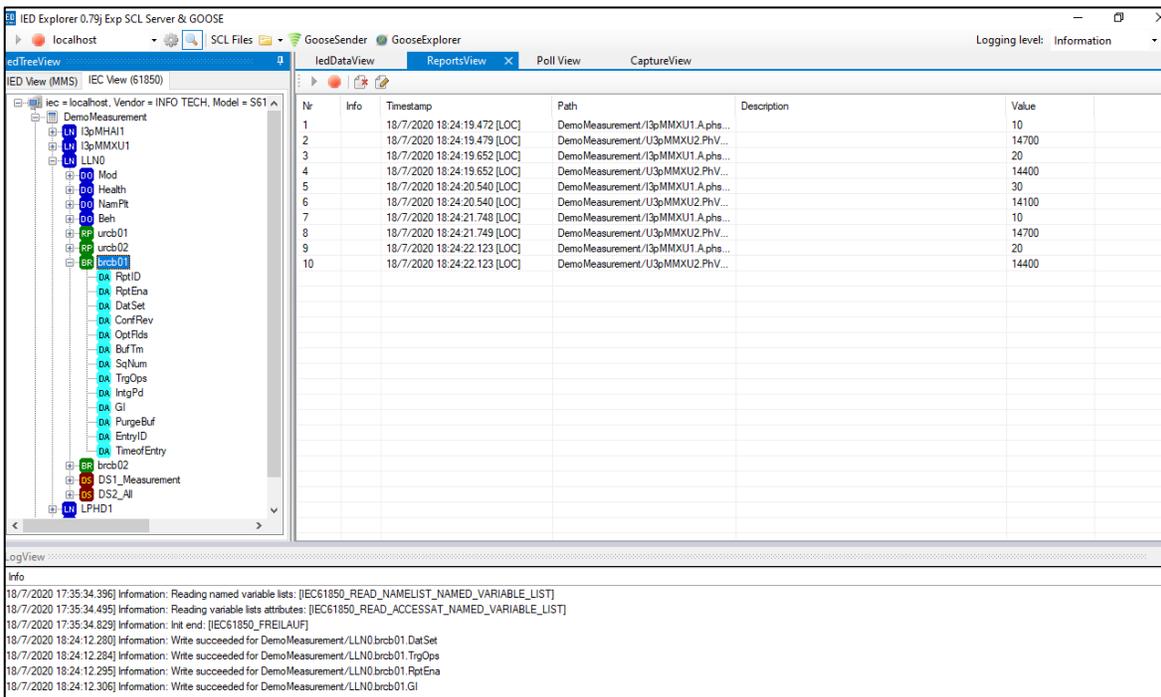


Figura 37. Visualización de Reportes.

1.2.4 Goose Sender/Receiver (Suscriptor/Publicador GOOSE)

Software de la empresa INFOTECH creado con propósitos educativos cuyo fin es el envío de mensajes GOOSE a través de una VLAN, tal como se observa en la figura 38 se configuran los parámetros del adaptador de red que ocupa, las direcciones MAC de fuente y destino, el número de VLAN, su prioridad, el App ID, el identificador de GOOSE, el nodo lógico y dataset al cual va dirigido el mensaje. En la parte inferior se observa un recuadro donde se encuentran los datos a ser enviados, este puede ser desde un valor bool, hasta un conjunto de valores a partir de una formula.

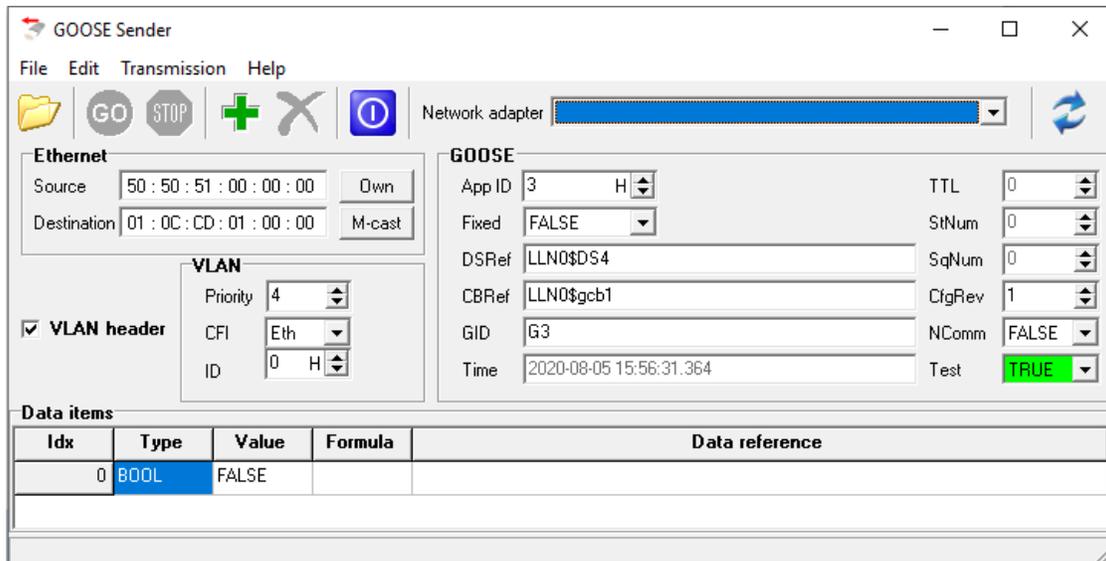


Figura 38. GOOSE sender.

Los valores enviados pueden ser captados por su programa dual GOOSE receiver de la misma empresa o pueden ser enviados a un equipo capaz de recibir mensajes GOOSE ya sea de forma simulada o un IED real.

Los elementos que se encuentran en la barra de herramientas del GOOSE Sender son descritos a continuación



Import: Importa archivos SCL que contengan mensajes GOOSE.

Start: Inicia la simulación para envío de mensajes GOOSE.

Stop: Detiene la simulación para el envío de mensajes GOOSE.

Add: Añade un nuevo ítem para mensajes GOOSE (Booleano, Entero, Calidad etc.).

Delete: Elimina ítem para mensajes GOOSE.



Exit: Cierra el programa GOOSE Sender.

Network adapter Adapter 1 MAC: 08:00:27:00:E0:E4

Selecciona la Red para envío de GOOSE.



Reload: Actualiza los adaptadores de Red.

La figura 39 muestra la interfaz del software complemento GOOSE receiver, se puede observar que ambos son configurados de la misma manera, sin embargo, este otro software monitorea los mensajes que se están enviando en la misma red.

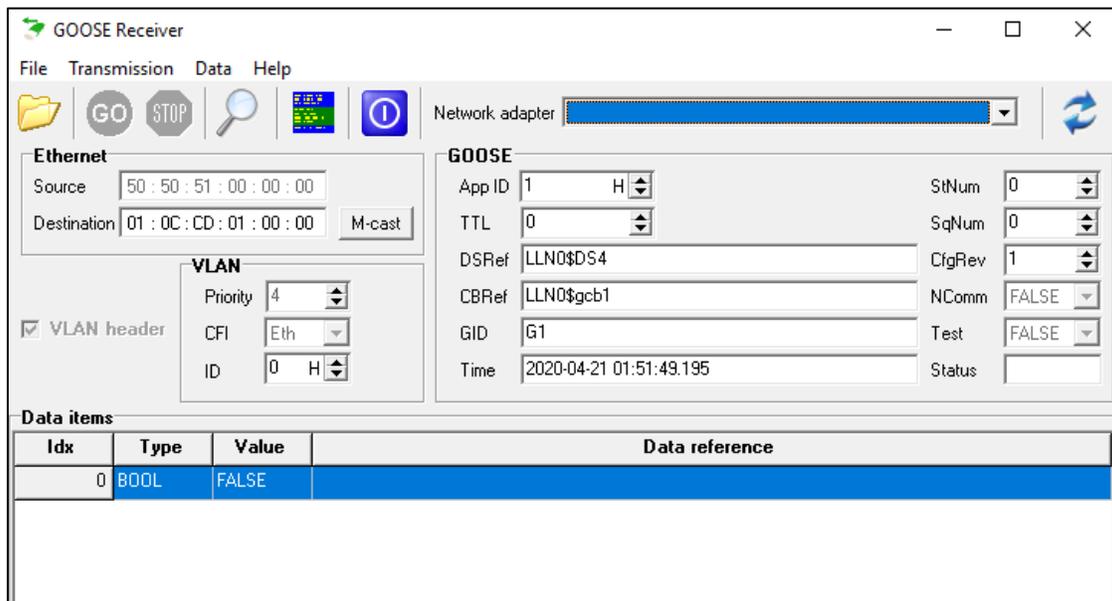


Figura 39. GOOSE Receiver.

Los elementos que se encuentran en la barra de herramientas del GOOSE Receiver son descritos a continuación:



Import: Importa archivos SCL que contengan mensajes GOOSE.



Start: Inicia la simulación para envío de mensajes GOOSE.



Stop: Detiene la simulación para el envío de mensajes GOOSE.



Detect Streams: Busca los mensajes GOOSE enviados en la Red.



Parser Window: Visualiza el tráfico de mensajes GOOSE en la Red.



Exit: Cierra el programa GOOSE Sender.

Para el envío de mensajes GOOSE se importa un archivo SCL o se configura directamente un mensaje desde el GOOSE Sender, la figura 40 muestra los mensajes GOOSE de un archivo importado, para realizar esta acción hacer clic en *Import* y seleccionar el archivo SCL que contiene los mensajes GOOSE.

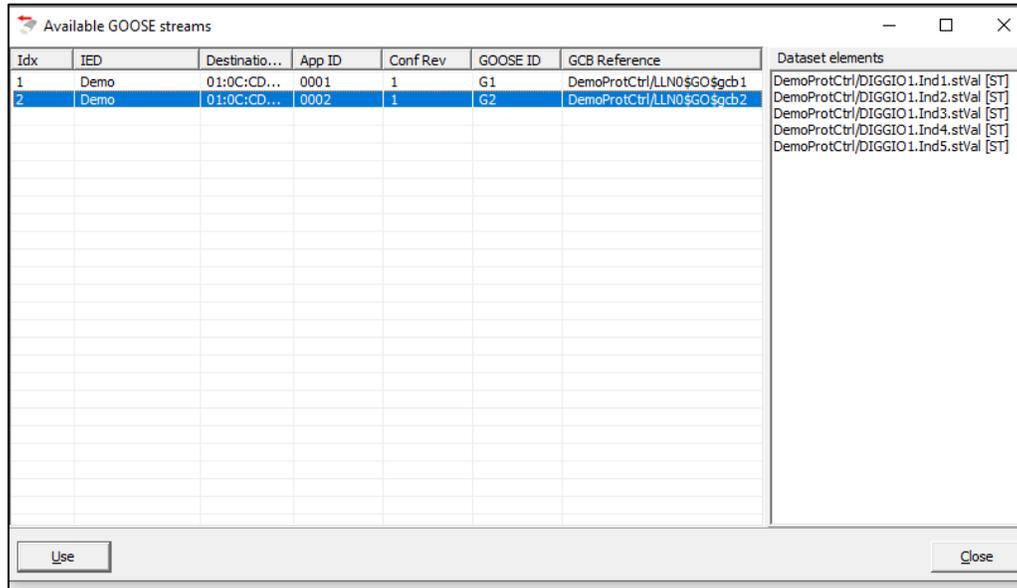


Figura 40. Archivo SCL de mensajes GOOSE importado.

Para simular un mensaje GOOSE se selecciona el mensaje que se desea simular desde el archivo importado y luego se hace clic en el botón Use, al realizar esta acción se mostrara en la ventana principal los atributos que contiene el mensaje, así como su configuración, tal como lo muestra la figura 41.

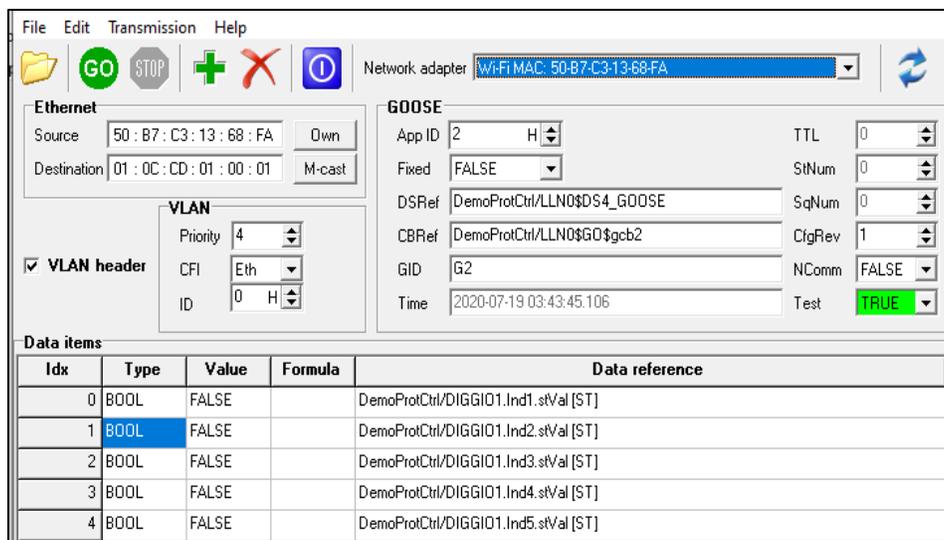


Figura 41. Simulación de envío de mensajes GOOSE.

Una vez se tengan los mensajes GOOSE a simular, seleccionar el adaptador de Red en el cual se estarán enviando dichos mensajes y hacer clic en *Start*, para la visualización de los mensajes enviados se utiliza el software GOOSE Receiver para ello se hace clic en *Detect Streams*, al realizar esta acción se mostrarán en una ventana todos los mensajes GOOSE que están siendo enviados por un publicador en la Red, tal como se muestra en la figura 42.

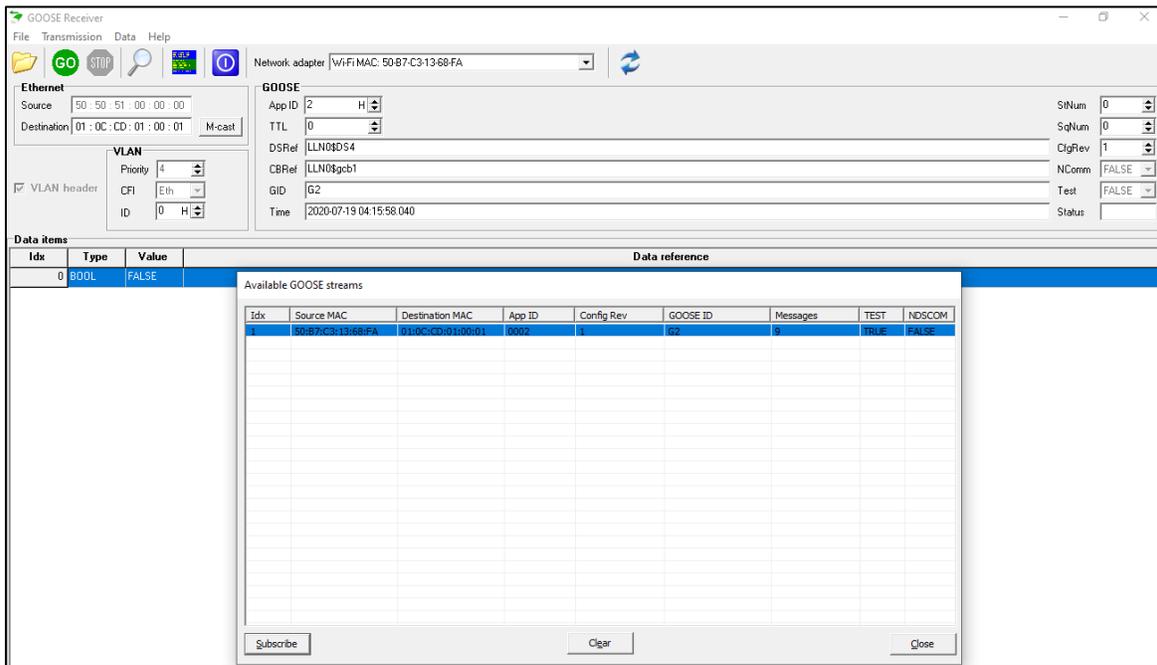


Figura 42. Búsqueda de mensajes GOOSE.

Luego de seleccionar el mensaje GOOSE que se desea visualizar, hacer clic en *Suscribe*, seguidamente aparecerá en la ventana principal los atributos y la configuración que contiene el mensaje GOOSE al cual se está suscrito, por último, dar clic en *Start*.

Es importante que el adaptador de Red sea el mismo que está configurado en el Goose Sender, los cambios realizados en cada atributo desde el GOOSE Sender se reflejan inmediatamente en el GOOSE Receiver.

1.2.5 Infotech 61850 Relay (Dispositivo virtual)

Software de simulación de bahía compuesto por interruptores y un seccionador, posee un relé de protección de sobre corriente bajo el estándar IEC61850 para la simulación de falla en un motor y medición de corriente en las tres fases del motor. El simulador cuenta con 5 indicadores, los cuales envían tramas de mensajes GOOSE, también cuenta con activación del indicador TRIP mediante mensajes GOOSE, además de un modo local y remoto para el control de los relés a través de MMS. La figura 43 muestra la interfaz del software.

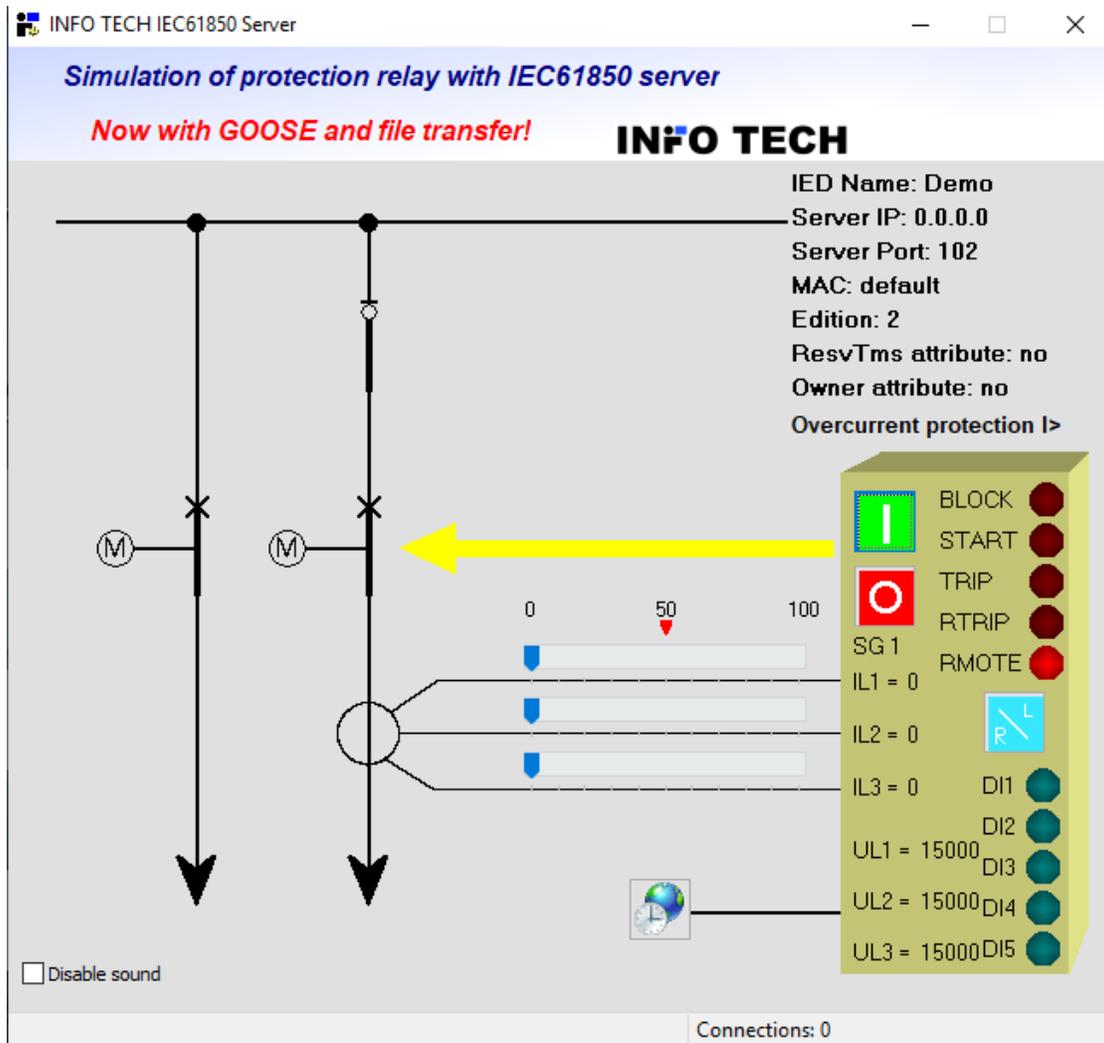


Figura 43. Simulador RELE IEC61850.

El software funciona como un servidor bajo el estándar IEC61850, el archivo ICD del relé simulado se encuentra en la ruta de instalación con las configuración y modelado del IED, dicho archivo puede ser simulado en otro servidor y funcionara de igual manera, pero sin interfaz gráfica, también es posible cambiar la dirección IP, puerto, dirección MAC entre otros parámetros al editar dichos archivos.

Para el envío de disparos del relé a través de mensajes GOOSE se utiliza el *software GOOSE Publisher*, este es instalado junto con el Simulador del Relé, para enviar un mensaje GOOSE al relé se debe configurar el adaptador de Red donde se enviará la trama de mensajes GOOSE y luego se hace clic en GO, tal como se muestra en la figura siguiente.



Figura 44. Goose Publisher.

1.2.6 Simulator Bay (Dispositivo virtual)

Es una herramienta interactiva de simulación para una bahía típica con soporte para comunicaciones IEC61850 / MMS. Simulator Bay cuenta con una herramienta que actúa como publicador/suscriptor del Relé simulado con el cual se pueden efectuar disparos vía GOOSE y visualizar los cambios de estado de los indicadores del Relé, la figura siguiente muestra la interfaz gráfica de esta herramienta.

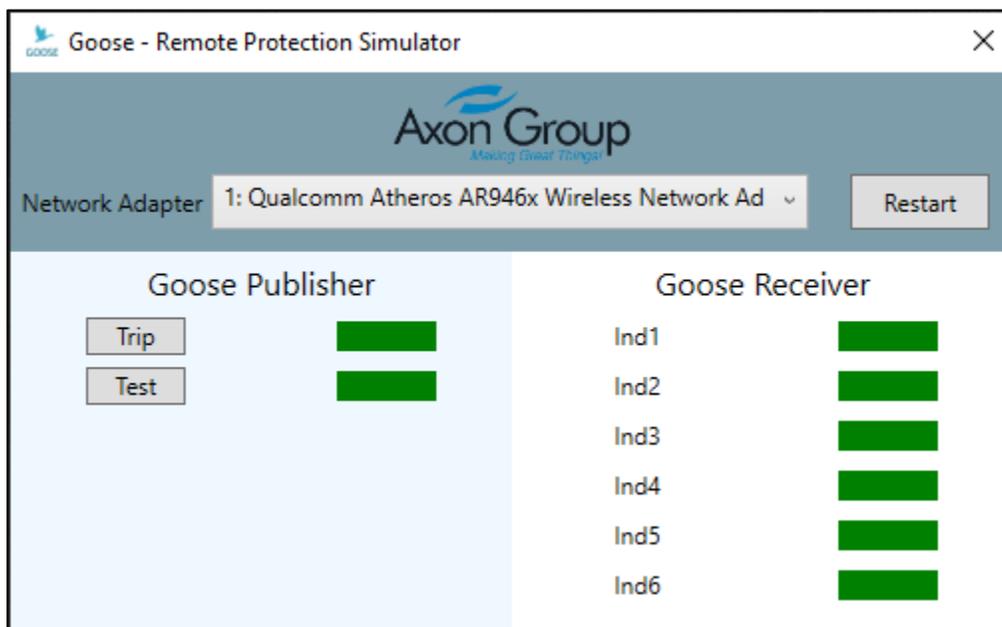


Figura 45. Remote Protection Simulator GOOSE.

La herramienta ofrece una interfaz interactiva en la que se podrá simular elementos comunes de una bahía en configuración barra con doble salida. Para ello, utiliza tres equipos: dos interruptores y un seccionador. Además, presenta un panel de medidas que se pueden ajustar en modelo de auto-simulación para la falla de un motor, posee también 6 indicadores los cuales enviaran tramas de mensajes GOOSE, la figura siguiente muestra la interfaz gráfica del software.

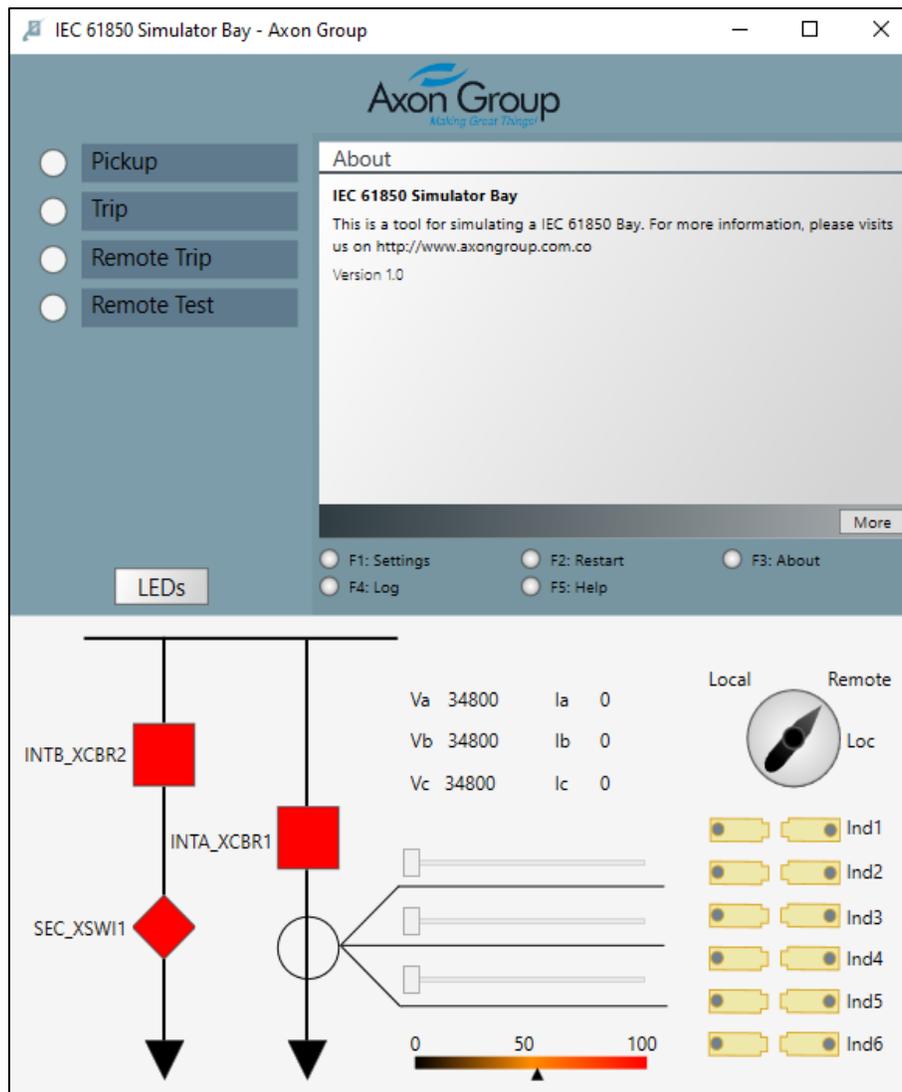


Figura 46. Simulator Bay.

El software también permite soportar operaciones de lectura/escritura sobre los atributos de los equipos, simular enclavamiento Local/Remoto y seccionador para validar comandos inválidos, también posee un log de eventos para visualizar cambios de conexión y comandos.

1.2.7 Wireshark (Analizador de red)

Es un software gratuito que permite analizar el tráfico en la red en tiempo real. La herramienta intercepta el tráfico y lo convierte en un formato legible para las personas. Esto hace que sea más fácil identificar qué tráfico está cruzando la red, con qué frecuencia y la latencia que hay entre ciertos saltos. Wireshark admite más de 2000 protocolos de red, incluidos entre ellos IEC61850, en la figura siguiente muestra la interfaz gráfica del software.

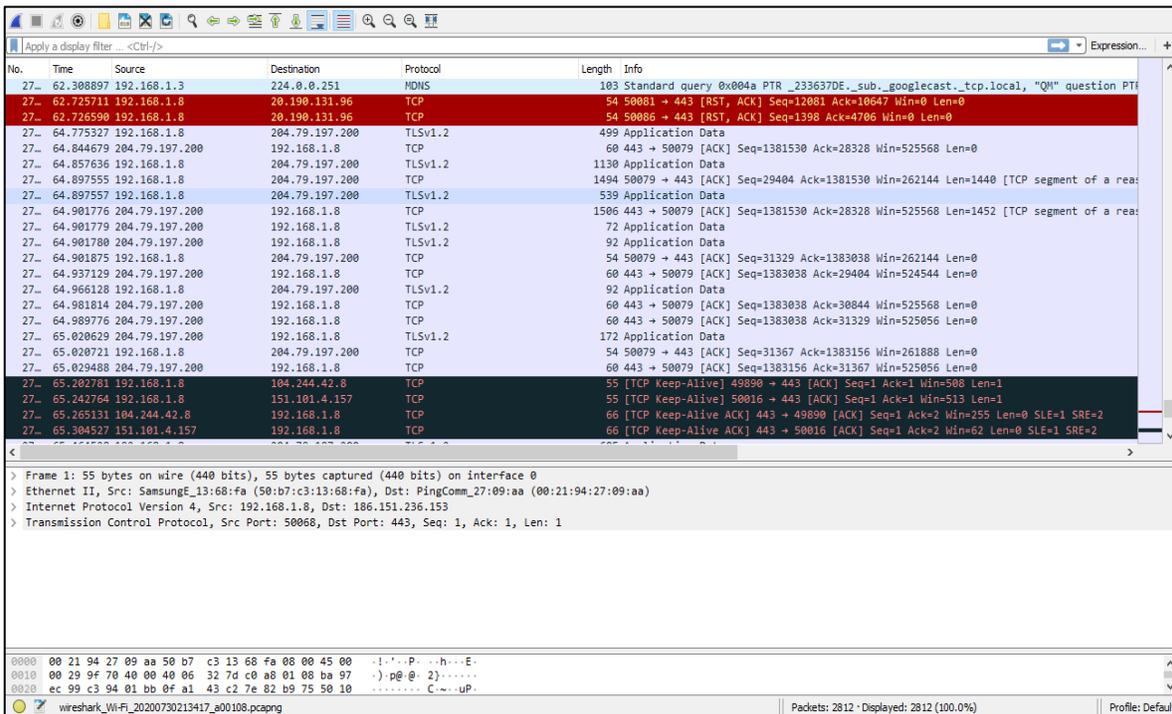


Figura 47. Tráfico de Datos en Wireshark.

Para propósitos de este documento, Wireshark se utiliza para analizar el tráfico de datos del protocolo GOOSE, también es posible analizar el tráfico del protocolo Sampled Value. La publicación de estos protocolos se realiza a través de tramas de multidifusión, un paquete de trama ethernet estará conformado por una MAC de origen, una MAC de destino y el Ethertype este último permite identificar el tipo de mensaje que se está transmitiendo, por ejemplo, para un mensaje **GOOSE** el Ethertype utilizado será **0x88B8**, en el caso de un mensaje **SV** el Ethertype utilizado es **0x88BA**.

En las siguientes ilustraciones se muestra la trama de un mensaje GOOSE y un mensaje SV capturado en Wireshark.

```

> Frame 1: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
▼ Ethernet II, Src: SamsungE_13:68:fa (50:b7:c3:13:68:fa), Dst: Iec-Tc57_04:00:02 (01:0c:cd:04:00:02)
  > Destination: Iec-Tc57_04:00:02 (01:0c:cd:04:00:02)
  > Source: SamsungE_13:68:fa (50:b7:c3:13:68:fa)
    Type: 802.1Q Virtual LAN (0x8100)
▼ 802.1Q Virtual LAN, PRI: 4, DEI: 1, ID: 0
  100. .... .... = Priority: Video, < 100ms latency and jitter (4)
  ...1 .... .... = DEI: Eligible
  .... 0000 0000 0000 = ID: 0
  Type: IEC 61850/SV (Sampled Value Transmission (0x88ba))
▼ IEC61850 Sampled Values

```

Figura 48. Trama Sampled Value capturado en Wireshark.

```

> Frame 3: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface 0
▼ Ethernet II, Src: SamsungE_13:68:fa (50:b7:c3:13:68:fa), Dst: Iec-Tc57_01:00:00 (01:0c:cd:01:00:00)
  > Destination: Iec-Tc57_01:00:00 (01:0c:cd:01:00:00)
  > Source: SamsungE_13:68:fa (50:b7:c3:13:68:fa)
    Type: 802.1Q Virtual LAN (0x8100)
▼ 802.1Q Virtual LAN, PRI: 4, DEI: 0, ID: 0
  100. .... .... = Priority: Video, < 100ms latency and jitter (4)
  ...0 .... .... = DEI: Ineligible
  .... 0000 0000 0000 = ID: 0
  Type: IEC 61850/GOOSE (0x88b8)
> GOOSE

```

Figura 49. Trama GOOSE capturado en Wireshark.

Como se logra apreciar en las ilustraciones anteriores las tramas aquí presentes están definidas como:

- **MAC Origen:** SamsungE_13:68:fa (50:b7:c3:13:68:fa)
- **MAC Destino:** Iec-Tc57_01:00:00 (01:0c:cd:01:00:00)
- **EtherType:**
 - IEC 61850/GOOSE (0x88b8)
 - IEC 61850/SV (Sampled Value Transmission (0x88ba)).

Para visualizar el tráfico de datos de mensajes GOOSE seleccionar el adaptador de red por el cual se estarán publicandando dichos mensajes, esto se logra en la ventana de inicio tal como lo muestra la figura 50, una vez seleccionado el adaptador de red, se abrirá una ventana mostrando el tráfico existente en la red.

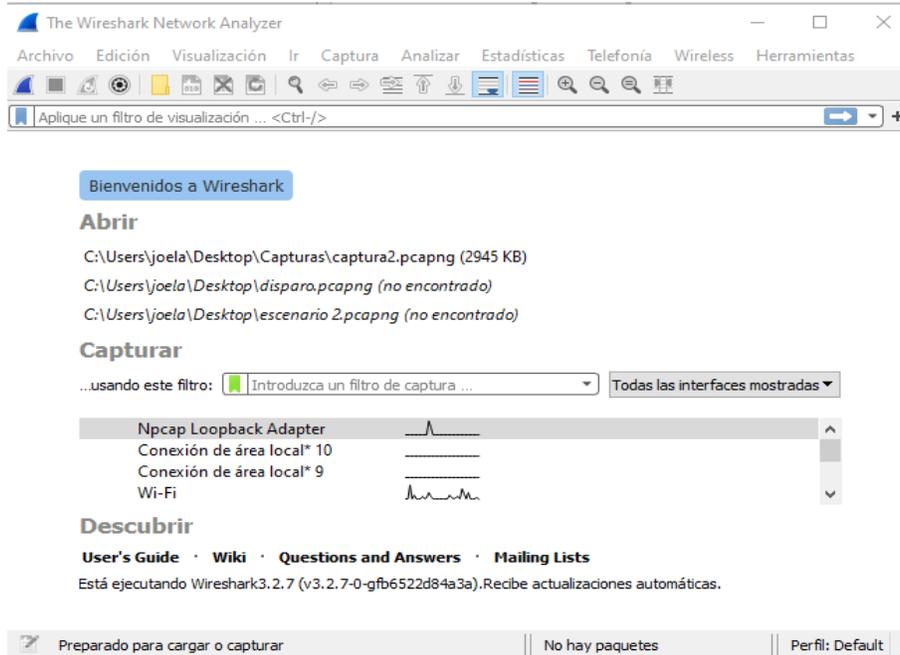


Figura 50. Selección de adaptadores de Red en Wireshark.

Para poder visualizar únicamente el tráfico de mensajes Goose hacer clic en “Expresión” ubicado al lado derecho de la barra de herramientas, seguidamente se abrirá una nueva ventana que mostrará todos los protocolos que puede analizar el software, ubicarse en “Search” y escribir el nombre del protocolo de interés. tal como se muestra en la figura 51.

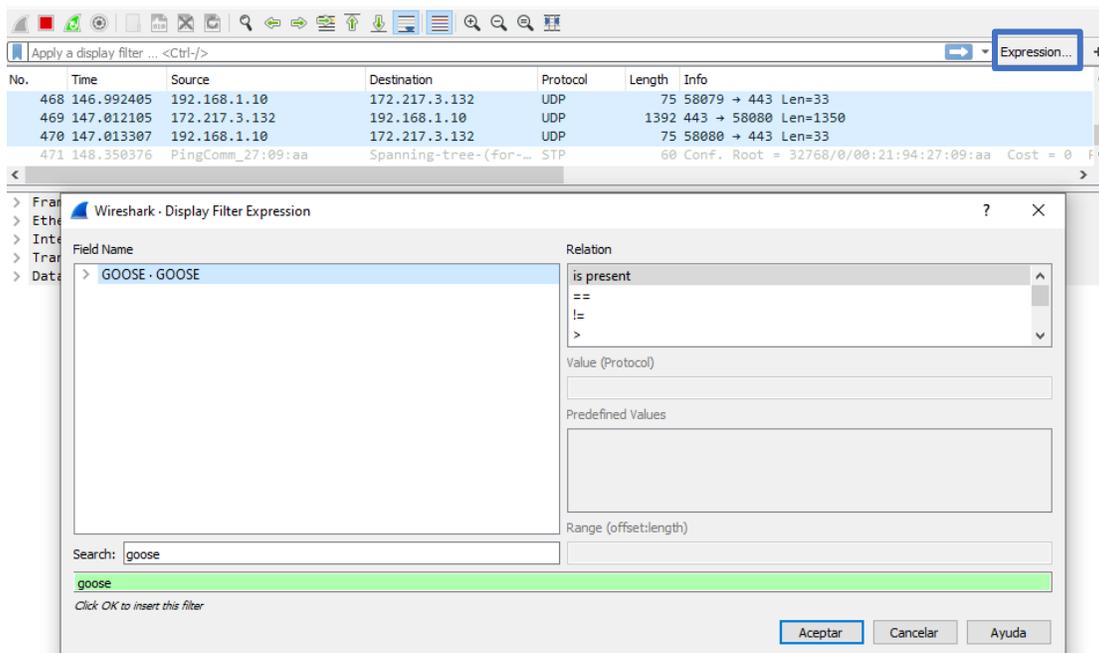


Figura 51. Selección de Protocolo bajo análisis.

Una vez se encuentre el protocolo deseado se selecciona y se hace clic en aceptar, esto permitirá ver únicamente el tráfico de datos de mensajes GOOSE tal como se muestra en la figura 52.

The screenshot shows a network traffic analysis tool interface. The top part is a table of captured packets, and the bottom part is a detailed view of a selected frame.

No.	Time	Source	Destination	Protocol	Length	Info
9	3.280982	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	93	
26	11.471998	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	94	
75	27.856018	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	94	
201	60.624006	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	94	
311	93.391975	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	94	
420	126.159996	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	94	
495	158.928035	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	94	
589	191.695981	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	94	
698	224.463986	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	94	
748	257.233042	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	94	
831	289.999990	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	94	
1163	322.767984	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	94	
1216	355.535005	SamsungE_e3:f6:c2	Iec-Tc57_01:00:02	GOOSE	94	

The detailed view of frame 201 shows the following structure:

- Frame 201: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0
- Ethernet II, Src: SamsungE_e3:f6:c2 (e8:03:9a:e3:f6:c2), Dst: Iec-Tc57_01:00:02 (01:0c:cd:01:00:02)
 - Destination: Iec-Tc57_01:00:02 (01:0c:cd:01:00:02)
 - Source: SamsungE_e3:f6:c2 (e8:03:9a:e3:f6:c2)
 - Type: 802.1Q Virtual LAN (0x8100)
- 802.1Q Virtual LAN, PRI: 4, DEI: 0, ID: 0
 - 100. = Priority: Video, < 100ms latency and jitter (4)
 - ...0 = DEI: Ineligible
 - ... 0000 0000 0000 = ID: 0
 - Type: IEC 61850/GOOSE (0x88b8)
- GOOSE
 - APPID: 0x0003 (3)
 - Length: 76
 - Reserved 1: 0x0000 (0)
 - Reserved 2: 0x0000 (0)
 - goosePdu

Figura 52. Tráfico de datos de mensajes GOOSE.

1.2.8 Elipse Power (SCADA)

Elipse Power es una herramienta para el desarrollo de aplicaciones de supervisión, control y análisis de sistemas de energía eléctrica, reúne en una sola plataforma, funcionalidades específicas para el desarrollo de aplicaciones para la automatización de subestaciones, proporcionando operación local y acceso remoto al centro de control, funcionando como una puerta de enlace de comunicación.

Para propósitos de este documento se utilizará Elipse Power como centro de control de la subestación, el cual simulará una SAS a niveles de bahía y de estación. La figura siguiente muestra la interfaz gráfica del software.

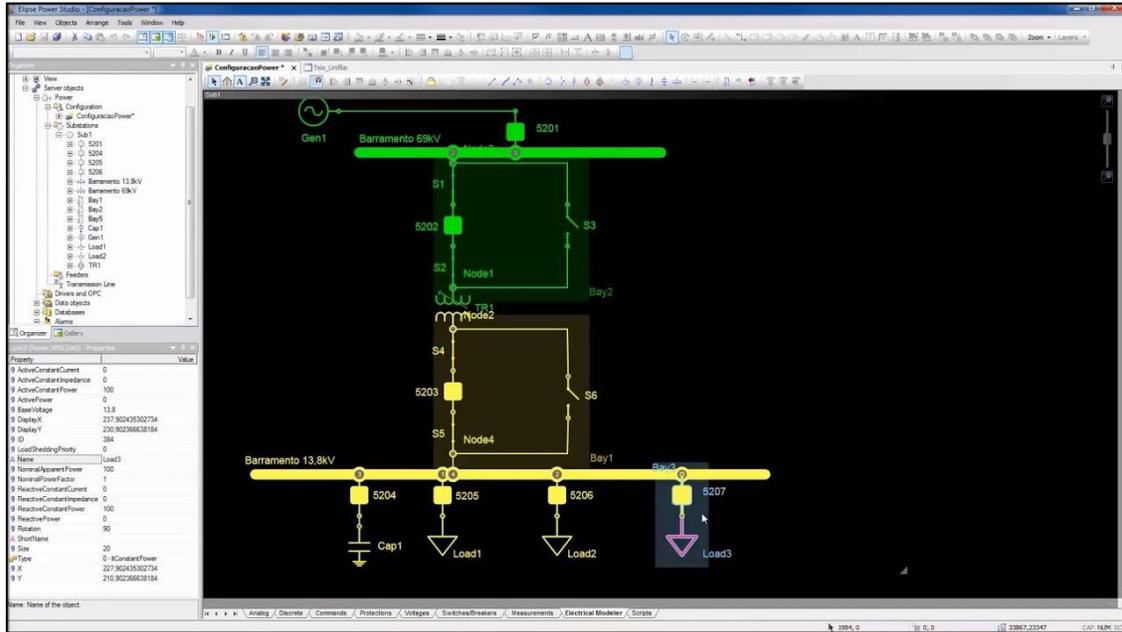


Figura 53. Software Elipse Power.

El diseño de un sistema SCADA quedan fuera de los alcances de este documento, por lo que únicamente se describen de manera breve algunas herramientas del software que serán útiles para la aplicación de uno de los escenarios de prueba planteado en la sección 3.3 del capítulo 3. Para la creación de un nuevo proyecto hacer clic al icono “nuevo” ubicado en la barra de herramientas, esto abrirá una ventana donde se agrega el nombre del proyecto y la dirección donde desea guardarse, al finalizar hacer clic en avanzar hasta llegar a la última ventana donde se debe hacer clic en “concluir”, la figura siguiente muestra lo descrito en este párrafo.

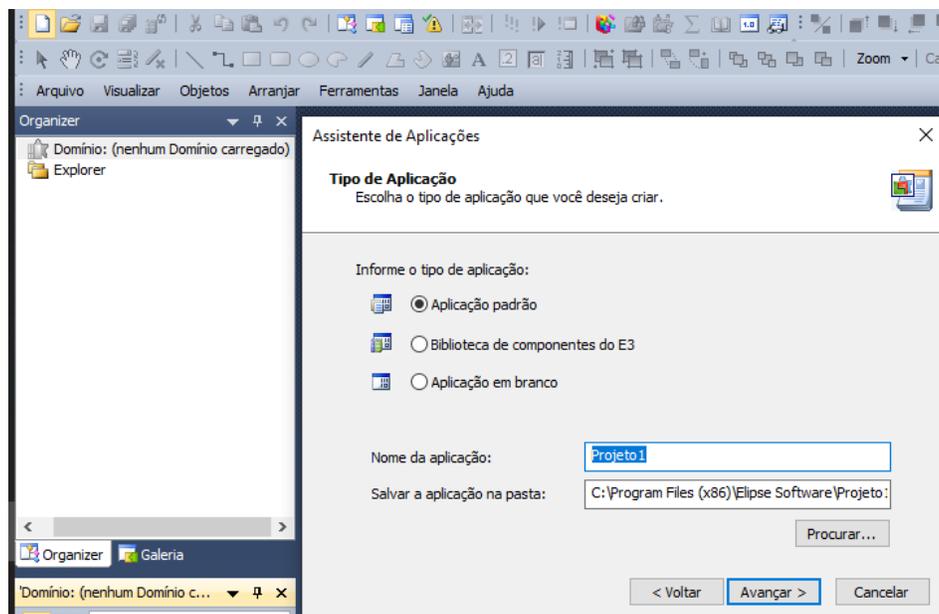


Figura 54. Creación de un nuevo proyecto en Elipse Power.

Una vez que se han configurado los parámetros para un nuevo proyecto se abre una nueva ventana, en la cual se configura el modelo eléctrico del proyecto, es posible modelar un sistema eléctrico de potencia completo, sin embargo, en este documento se utiliza el software para un proyecto de subestación, por lo que en la barra de herramientas de dicha ventana se presiona el icono de subestación y se arrastra hacia el área de trabajo tal como se muestra en la figura 55.

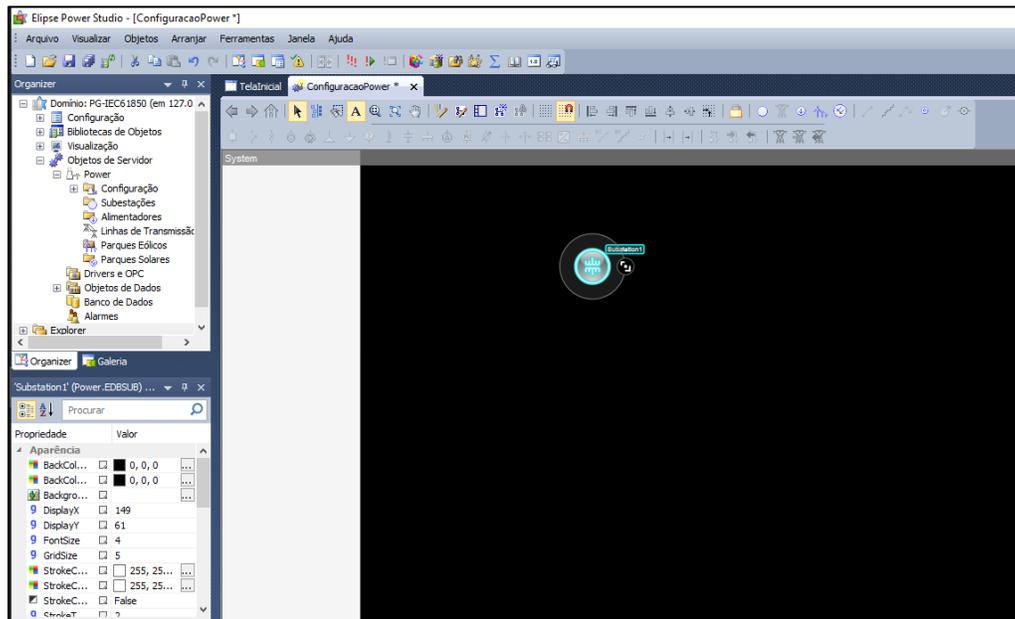


Figura 55. Configuración del modelo eléctrico.

Para la configuración del modelo de la subestación hacer doble clic sobre el elemento “subestación” que se colocó en el área de trabajo, se abrirá una nueva ventana donde se podrá diseñar el modelo de la subestación, tal como se muestra a continuación.

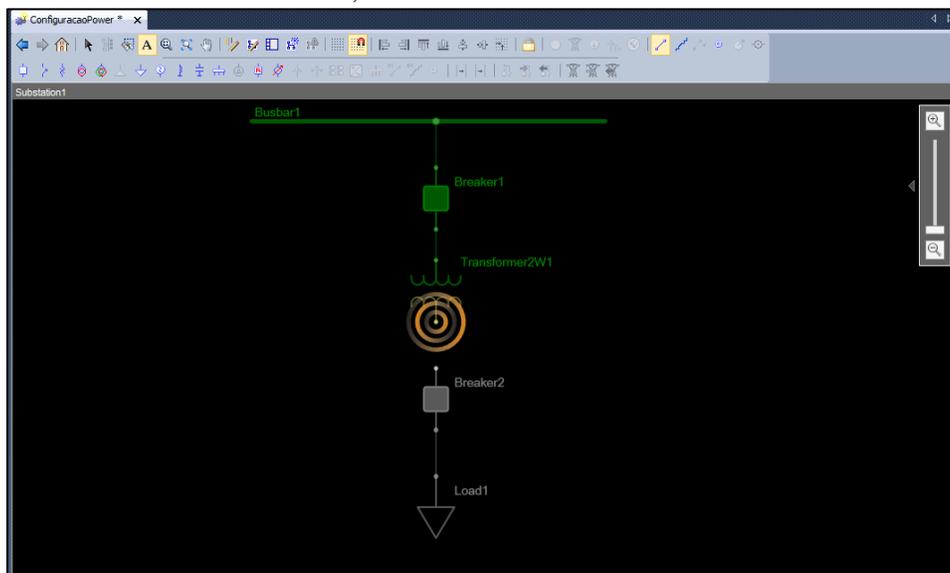


Figura 56. Configuración del modelo de la Subestación.

Cada uno de los elementos que se observan en la figura anterior han sido agregados desde la barra de herramientas de la ventana configuración del modelo eléctrico, para ello se debe seleccionar el equipo (Interruptores, transformadores, seccionadores etc.) que tendrá el modelo de subestación y arrastrar el elemento hasta el área de trabajo.

Es necesario importar el modelo eléctrico hacia la pantalla principal de control y supervisión (SCADA) para sincronizar los equipos del modelo eléctrico que han sido configurado y de esta manera poder asignar los Tags que tendrá cada elemento. Para ello en la pestaña “Organizer” que se encuentra en el lado izquierdo de la ventana principal del software hacer clic derecho sobre “ConfigurePower-Modelo Eléctrico-Importar para Tela”, tal como se muestra a continuación.

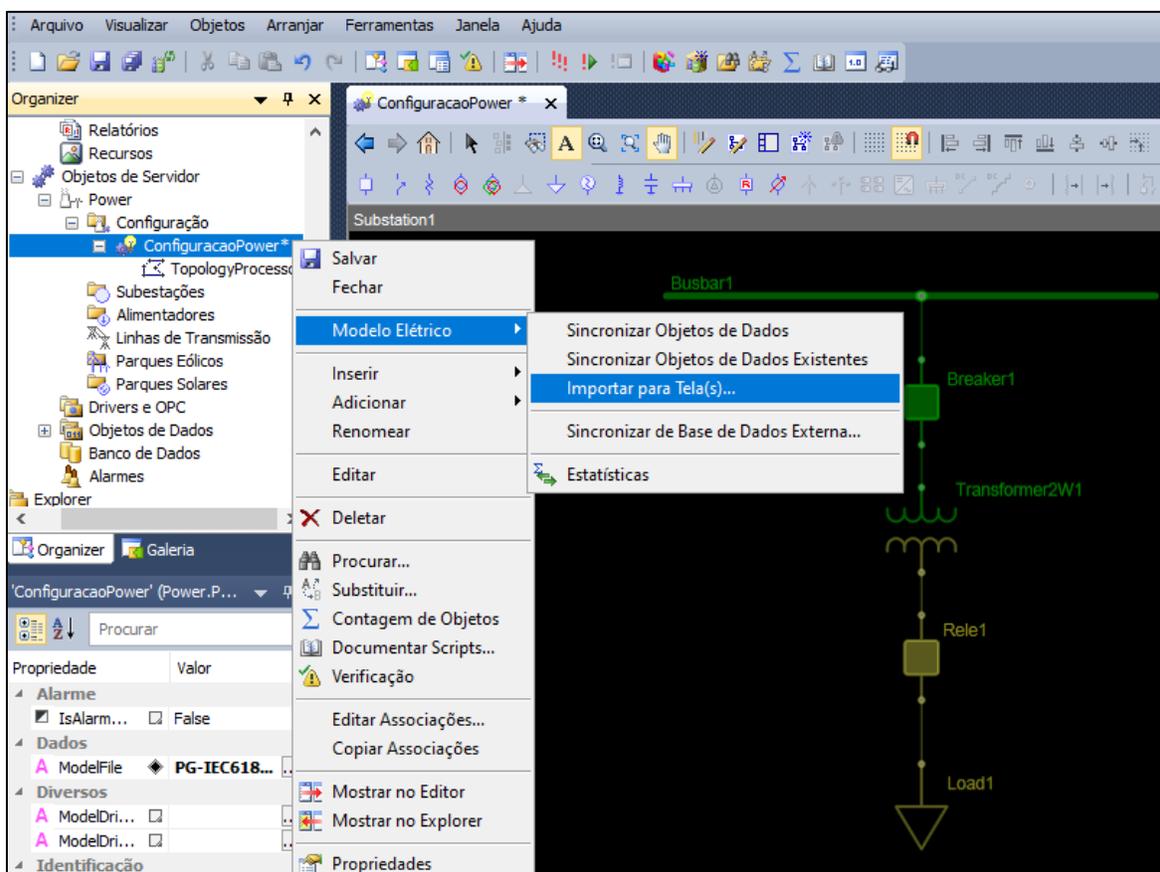


Figura 57. Importando modelo eléctrico hacia el SCADA.

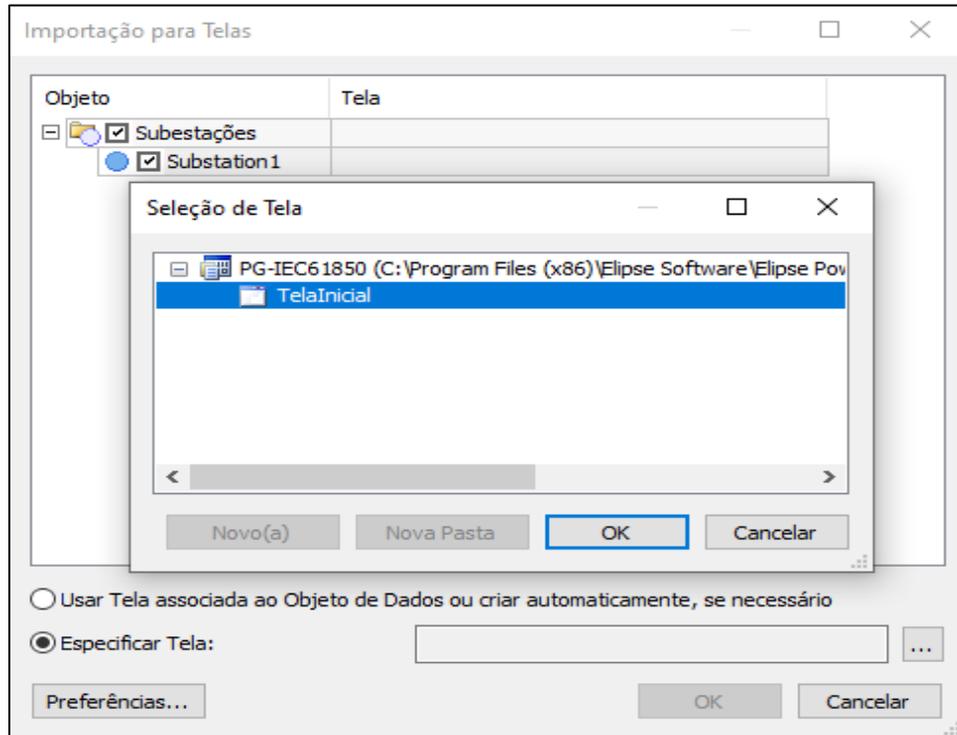


Figura 58. Importando modelo eléctrico de subestación.

Luego de seleccionar “Importar para Tela” aparecerá una nueva ventana como se muestra en la figura 58. Aquí se selecciona los objetos de datos del modelo eléctrico que se desea sincronizar y la pantalla a la cual se desea importar el archivo, por último, se hace clic en OK.

Una vez realizada esta acción aparecerán en la pantalla principal todos los elementos del modelo eléctrico, los cuales están sincronizados y se configuran desde la pestaña Subestación asignando los tags correspondientes a cada elemento para su supervisión en la pantalla principal SCADA.

La figura 59 muestra la pantalla principal del SCADA, desde la cual es posible configurar el entorno e interfaz gráfica que tendrá el SAS para su control y supervisión.

Para que ElipsePower pueda extraer los tags desde los servidores que trabajen bajo el estándar IEC61850, será necesario agregar un Driver OPC (Cliente) esto permite comunicarse con un servidor OPC (Relab) desde el SCADA, para crear un nuevo driver se selecciona *Driver OPC* ubicado en la pestaña “Organizer” y se hace clic derecho sobre él, esta acción abrirá un submenú en el cual se selecciona el item *Insertar driver de comunicación OPC*, seguidamente se abre un nuevo menú donde se selecciona el item *Nuevo Archivo* tal como se muestra en la figura 60. Finalmente debe agregarse un nombre al Driver y guardarlo en la dirección que aparece asignada por defecto la cual corresponde al proyecto en el cual se está trabajando.

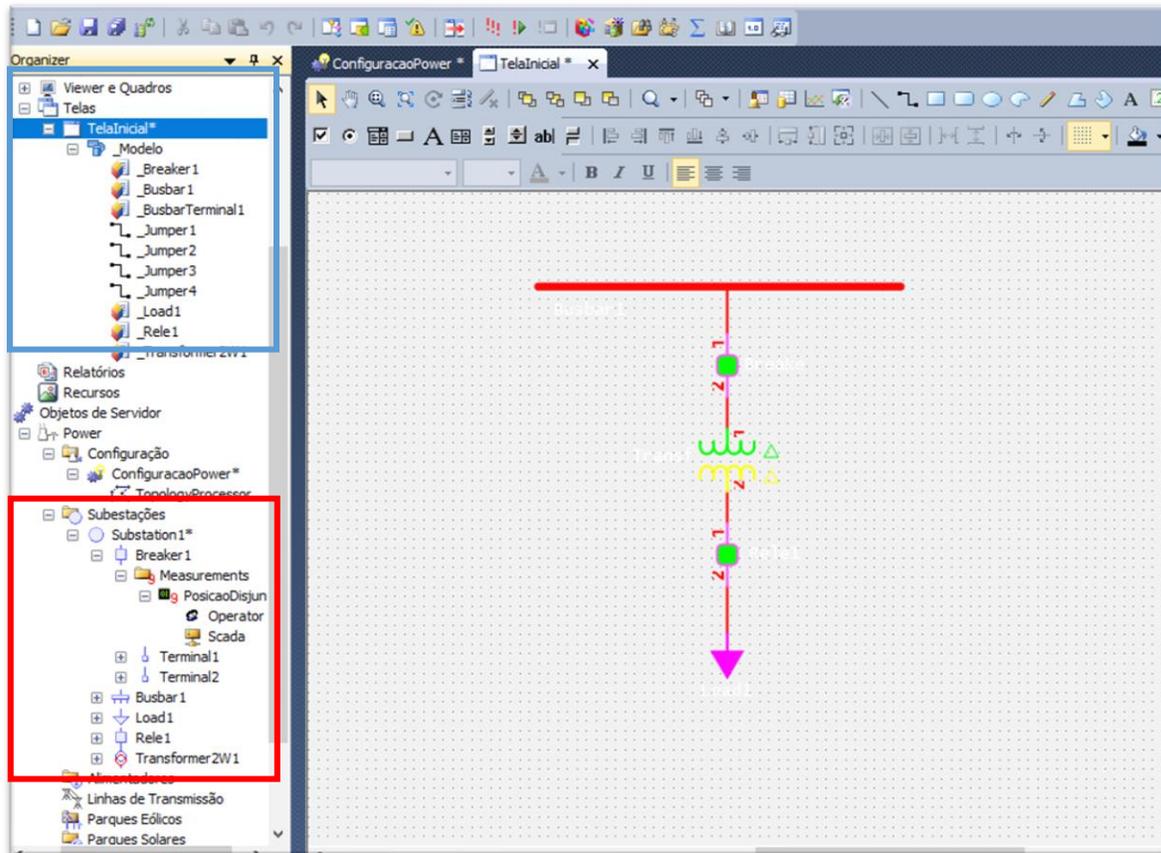


Figura 59. Sincronización del modelo eléctrico en pantalla principal SCADA.

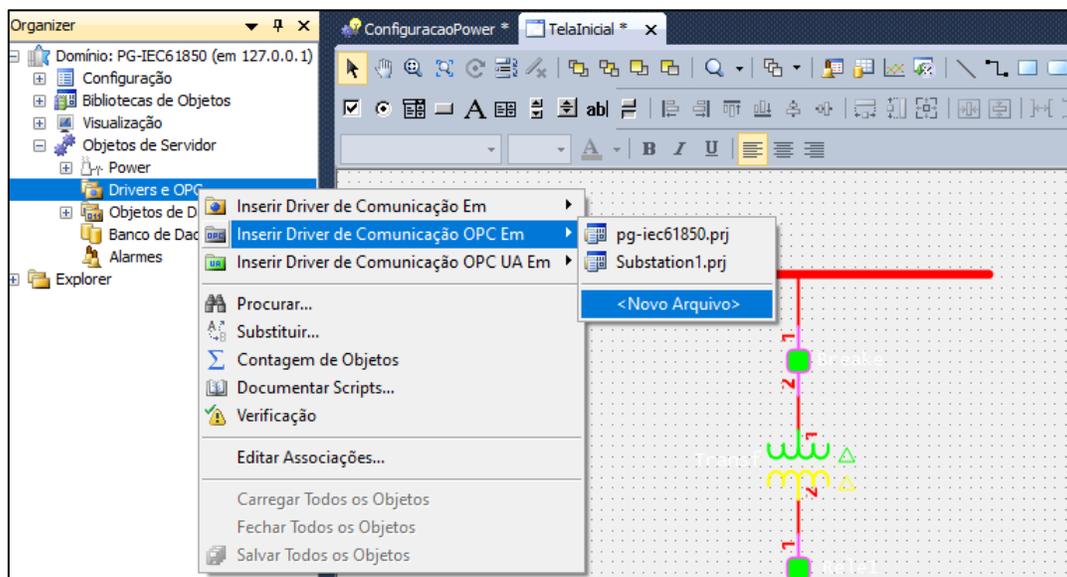


Figura 60. Creación de Driver OPC.

Luego de crear el Driver aparecerá la ventana que se muestra en la figura 61, se selecciona un servidor OPC, para este trabajo se utiliza el *CV.OPC.1* del software *ReLab OPC*, el cual se ha descrito en el apartado 1.2.2 por último se finaliza haciendo clic en OK. Para activar la comunicación entre el SCADA y Relab OPC se hace clic en activar comunicación (icono dentro de círculo rojo), al realizar esta acción aparecerán todos los tags que se han mapeado en el Dashboard del ReLab, se importarán aquellos tags que sean de interés en el control y supervisión del SAS.

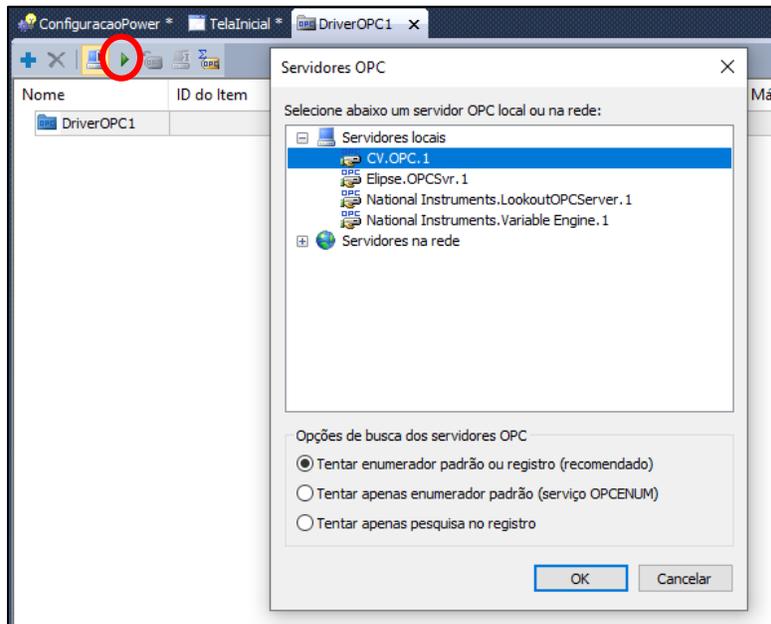


Figura 61. Configuración de servidor OPC.

1.2.9 VNCViewer (Cliente VNC)

VNC Viewer es simplemente un cliente de VNC. Esta tecnología se usa para poder controlar remotamente un dispositivo desde otro. Para realizar una conexión VNC se deben realizar diversos pasos previos. Por un lado, debe tenerse un servidor y un cliente. El servidor, el dispositivo que se quiere controlar, debe disponer de un software servidor de VNC. El cliente, el dispositivo desde el que se quiere acceder, debe disponer de un software cliente de VNC.

Para este proyecto se utiliza esta herramienta para controlar desde la PC local la Raspberry Pi y acceder a las librerías que ayudan a emular un IED. Para la conexión PC-Raspberry Pi, bastará con conocer la dirección IP que tiene la Raspberry, una vez se establece la conexión se ingresa el nombre de usuario y contraseña del dispositivo, la figura siguiente muestra la ventana de inicio del software donde se configuran estos parámetros.

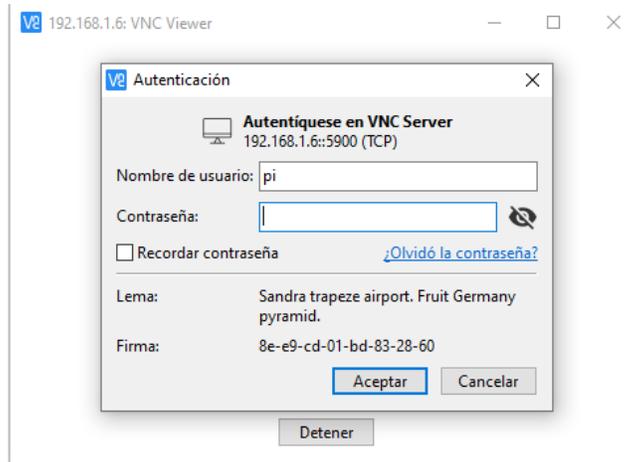


Figura 62. Ventana de configuración de VNCViewer.

1.2.10 Libiec61850 (librería servidor/cliente IEC61850)

Librería open source, escrita en C consiste en una API (Application Programming Interface) que puede ser instalada en Windows o en el sistema Linux, dicha API consiste en un cliente y servidor para mensajes MMS y publicador/suscriptor para mensaje GOOSE y Sample Values (SV).

El soporte para IEC61850 incluye la generación automática del modelo del dispositivo vía MMS en estructura de árbol aparte del modelo de datos bajo el estándar IEC61850. También la API provee soporte para el modelo de control IEC 61850, servicios log, modelo de ajuste de grupo, etc. El servidor IEC61850 soporta data sets, logeo y reportes, la estructura se muestra en la figura 63.

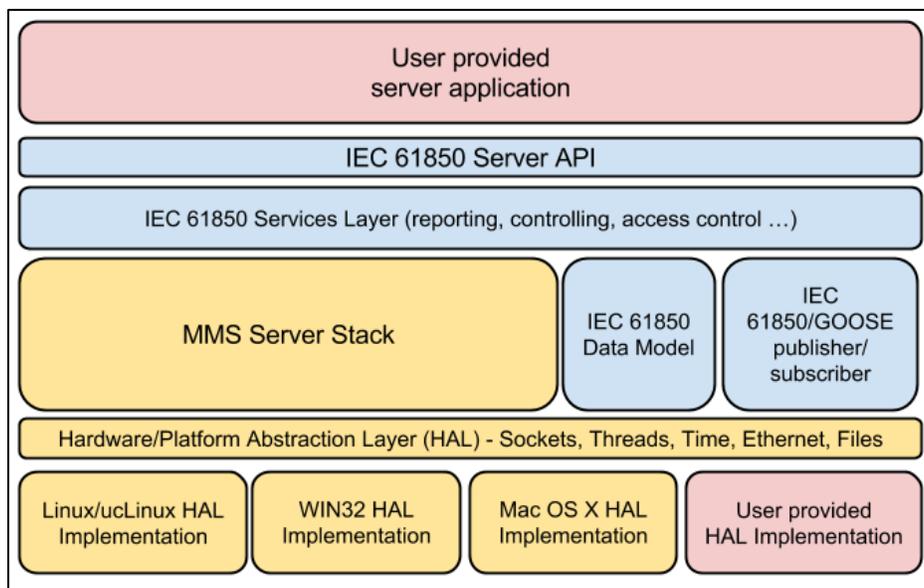


Figura 63. Estructura de servidor IEC61850 de la librería.

1.2.10.1 Instalación

Para la instalación en sistemas Linux se siguen los siguientes pasos:

1. Descarga de librería desde el sitio oficial:

<https://libiec61850.com/libiec61850/documentation/>

2. Instalación de ambiente de desarrollo bajo C++, para esto se ejecutan las siguientes líneas en la ventana de comandos:

```
$sudo apt-get build essential
```

3. Instalar la librería ejecutando en la terminal los siguientes comandos ubicándose en la carpeta principal de la librería:

```
$sudo make  
$sudo make install
```

1.2.10.2 Creación de servidor IEC61850 a partir de archivo ICD

La librería permite crear mediante una aplicación en C y a manera de servidor un archivo ICD, de esta manera la PC ejecuta el código y emula un IED con todas sus funcionalidades, en combinación con librerías de control de E/S en una placa de desarrollo con sistema Linux como Raspberry pi es posible emular un IED bajo el estándar.

Para la ejecución de un servidor bajo el estándar IEC61850, la librería necesita 6 archivos los cuales se muestran en la figura 64 y se describen a continuación:

- **Archivo CID:** Detalla las configuraciones y descripciones completas del IED,
- **Static_model.c y static_model.h:** Conversión que realiza la librería de la estructura del archivo ICD escrita en XML a código C, esto para lograr compatibilidad cuando se ejecute el código, dentro de estos archivos se encuentran todos los nodos lógicos, Dataset y configuración GOOSE pertenecientes al ICD.

Para realizar la conversión de un archivo ICD o CID a código C se utiliza una herramienta llamada Model Generator, el cual es una aplicación escrita en java, por lo que es necesario instalar java previo a utilizar la herramienta, para ello escribimos las siguientes líneas en la ventana de comandos:

```
$sudo apt-get update  
$sudo apt-install default jdk
```

La herramienta Model Generator está localizado en la carpeta tools/model_generator de la librería, en esta dirección se coloca el archivo ICD o CID y mediante la consola de comandos se ejecuta la siguiente instrucción.

```
$sudo java -jar genmodel.jar [nombre ICD] [ap - nombre de punto de acceso] [nombre de archivo de salida(static_model)] [prefijo del modelo]
```

- **Código principal:** Archivo que será compilado y ejecutado, todas las instrucciones y funciones que realice el IED estarán escritas en este archivo, el código principal hace posible la publicación y suscripción de mensajes GOOSE, ejecución de comandos MMS, activación de pines GPIO y lectura de sensores.
- **Makefile:** Contiene las instrucciones de compilación del código, en este se incluyen las librerías adicionales que se ocupan ya sea Modbus o WiringPI para la Raspberry.

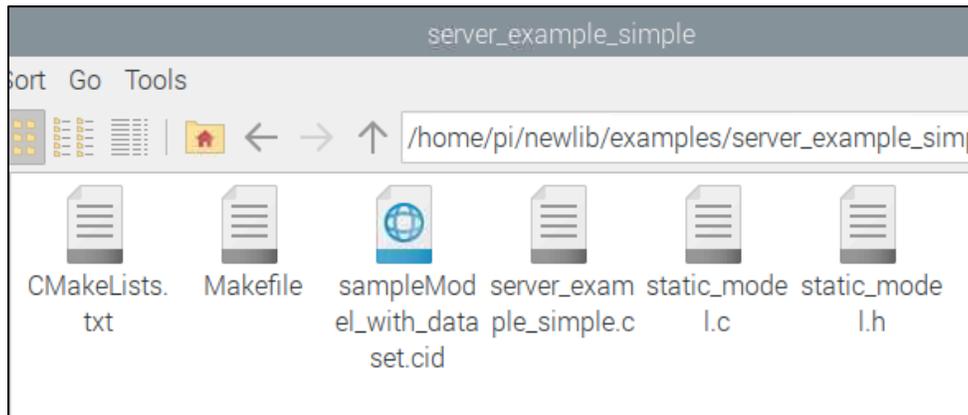


Figura 64. Archivos necesarios para ejecutar servidor bajo el estándar IEC61850.

1.2.10.3 Habilitar comandos MMS

Para habilitar comandos vía MMS se necesitan incluir algunas funciones para verificar el tipo de comando que ha recibido el servidor y verificar si es válido, dichas funciones son:

- 1) **checkHandler:** Verifica que el comando recibido sea de uno de los atributos para control disponibles, el siguiente ejemplo muestra la función para un LN de tipo GGIO y un objeto de tipo single point control status (SPCS), la función retorna un valor indicando si el control es aceptado o indefinido en caso contrario.

```
static CheckHandlerResult // Función para comprobar los
comandos MMS recibidos sean validos
checkHandler (ControlAction action, void* parameter, MmsValue*
ctlVal, bool test, bool interlockCheck)
{
    if (parameter == IEDMODEL_GenericIO_GGIO1_SPCS01)
        return CONTROL_ACCEPTED;
    return CONTROL_OBJECT_UNDEFINED;
}
```

- 2) **controlHandlerForBinaryOutput:** Extrae el valor solicitado por el comando MMS y actualiza el valor del atributo stValue por el del comando recibido, también actualiza el timestamp o marca de tiempo asociada a dicho objeto, el siguiente ejemplo muestra la función para un objeto de tipo SPCS01:

```

static ControlHandlerResult
controlHandlerForBinaryOutput (ControlAction action, void*
parameter, MmsValue* value, bool test)
{
    uint64_t timestamp = Hal_getTimeInMs ();
    if (parameter == IEDMODEL_GenericIO_GGIO1_SPCS01) {
        IedServer_updateUTCTimeAttributeValue (iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01_t, timestamp);
        IedServer_updateAttributeValue (iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01_stVal, value);
    }
    else
        return CONTROL_RESULT_FAILED;
}

```

- 3) **Main:** Función principal, hace el llamado a las funciones anteriores, esto se hace con la función *setControlHandler* y *setPerformCheckHandler* donde se colocan el nombre del servidor y atributo, llamando a la función de control, atributo. Tal como se muestra a continuación:

```

IedServer_setControlHandler (iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01,
(ControlHandler) controlHandlerForBinaryOutput,
IEDMODEL_GenericIO_GGIO1_SPCS01);
IedServer_setPerformCheckHandler (iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01, checkHandler,
IEDMODEL_GenericIO_GGIO1_SPCS01);

```

1.2.10.4 Habilitar publicador de mensajes GOOSE

Los mensajes GOOSE a ser enviados deberán definirse en primer lugar en el CID desde el que se crea el servidor. En el código debe incluirse el archivo *goose_publisher.h* al inicio del documento, y en la función “main” del código principal se configura según el ejemplo mostrado a continuación:

```

CommParameters gooseCommParameters;
gooseCommParameters.appId = 1000;
gooseCommParameters.dstAddress[0] = 0x01;
gooseCommParameters.dstAddress[1] = 0x0c;
gooseCommParameters.dstAddress[2] = 0xcd;
gooseCommParameters.dstAddress[3] = 0x01;
gooseCommParameters.dstAddress[4] = 0x00;
gooseCommParameters.dstAddress[5] = 0x01;
gooseCommParameters.vlanId = 0;
gooseCommParameters.vlanPriority = 4;

```

Como se observa en el ejemplo se tiene el *appId*, la dirección MAC del adaptador que enviara los mensajes, el ID de la VLAN a utilizar y su prioridad. En sistemas Linux la librería puede obtener automáticamente la dirección MAC en caso que no se incluyera entre los argumentos cuando se ejecuta el código, para esto se lee el fichero *stack_config.h* el cual contiene el

campo `CONFIG_ETHERNET_INTERFACE_ID` que contiene la MAC del adaptador ethernet si los hubiere.

Una vez establecidos los parámetros de la comunicación GOOSE, se establece la interfaz GOOSE, usando la función `IedServer_setGooseInterfaceIdEx` que lleva como parámetros: nombre del servidor IED, nodo lógico, descripción de mensajes, ID de interfaz física, a continuación, se muestra un ejemplo de uso:

```
IedServer_setGooseInterfaceIdEx (iedServer, IEDMODEL_GenericIO_LLNO,  
"valores_indicadores", ethernetIfcID);
```

Por último, se habilita el envío de mensajes GOOSE con la siguiente instrucción que necesita de parámetro el nombre del servidor IED:

```
IedServer_enableGoosePublishing(iedServer);
```

1.2.10.5 Habilitar suscripción de mensajes GOOSE

Para habilitar la suscripción de mensajes GOOSE es necesario incluir los archivos `goose_receiver` y `goose_subscriber` al inicio del código, en la función “main” se configuran los parámetros para la suscripción de la siguiente forma:

1. Creación de objeto receiver:

```
GooseReceiver receiver = GooseReceiver_create ();
```

2. Seleccionar el ID de la interfaz física a utilizar, en sistemas Linux esta instrucción automáticamente buscará la MAC del adaptador de red seleccionado.

```
GooseReceiver_setInterfaceId (receiver, "eth0");
```

3. Se crea el objeto subscriber el cual contiene los atributos que serán leídos en el adaptador de red seleccionado:

```
GooseSubscriber subscriber = GooseSubscriber_create  
("DemoProtCtrl/LLNO$GO$gcb2", NULL);
```

4. Se selecciona el AppId a utilizar:

```
GooseSubscriber_setAppId(subscriber, 2);  
GooseSubscriber_setListener(subscriber, gooseListener, NULL);  
GooseReceiver_addSubscriber(receiver, subscriber);
```

5. La siguiente instrucción inicia la lectura de los mensajes goose según los parámetros que se han seleccionado

```
GooseReceiver_start(receiver);
```

1.2.11 Librería libmodbus-3.1.6

Es una librería de software libre para enviar/recibir datos bajo el protocolo Modbus. Dicha librería está escrita en lenguaje C y soporta comunicaciones vía RTU (serial) y TCP (Ethernet).

A continuación, se muestra su forma de uso mediante un sencillo ejemplo:

```
#include <stdio.h>
#include <modbus.h> // incluimos la librería
int main(void) {
    modbus_t *mb; // creamos puntero para parametros modbus
    uint16_t tab_reg[32]; // Arreglo de enteros de 16bits
    mb = modbus_new_tcp("127.0.0.1", 1502); //añadimos parametros a
    puntero
    modbus_connect(mb); //realizamos conexión
    /* Lectura de 5 registros desde la dirección 0*/
    modbus_read_registers(mb, 0, 5, tab_reg);
    modbus_close(mb); //cerramos conexión
    modbus_free(mb); //liberamos puntero
}
```

El protocolo Modbus define diferentes tipos de datos y funciones para la lectura y escritura de datos, las cuales ya están definidas en la librería, algunas de ellas son:

- **modbus_read_bits ():** Lee el contenido de los bits de entrada *nb* de la dirección Modbus configuradas, el resultado se almacena en el arreglo *dest* con un dato signo de 8 bits. Este utiliza la función modbus de código 0x01(read coil status).

Sintaxis:

```
int modbus_read_bits(modbus_t *ctx, int addr, int nb, uint8_t *dest);
```

- **modbus_read_registers ():** Lee el contenido de los registros de valores (holding registers) de la dirección modbus seleccionada, El resultado se almacena en el arreglo *dest* como un valor de tipo word(16bits). Se utiliza la función modbus de código 0x03 (read holding registers).

Sintaxis: `int modbus_read_registers (modbus_t *ctx, int addr, int nb, uint16_t *dest).`

- **modbus_write_bit ():** La función escribe un bit único en la dirección *addr* del dispositivo remoto, dicho valor debe ser TRUE o FALSE, se utiliza la función Modbus 0x05 (force single coil).

Sintaxis: `int modbus_write_bit (modbus_t *ctx, int addr, int status).`

- **modbus_write_register ():** La función escribe un valor de tipo entero sin signo(16bits) en los registros de dirección *addr* del dispositivo remoto, utiliza la función Modbus 0x06 (preset single register).

1.2.12 Librería WiringPi

Librería escrita en C para el acceso de los pines GPIO de dispositivos SoC BCM2835 BCM2836 y BCM2837 usados en todas las versiones de Raspberry Pi, permite configurar los pines GPIO para escritura/lectura de señales, también pueden utilizarse los pines I2C, SPI y UART como pines de propósito general I/O cuando no son utilizados en sus modos de comunicación.

La librería viene instalada por defecto en el sistema operativo raspbian, para comprobar que este instalado usamos el siguiente comando en consola:

```
$gpio -v
```

En caso de no tener instalada la librería la instalamos con el siguiente comando en consola:

```
$sudo apt-get install wiringpi
```

1.2.12.1 Funciones básicas

- **wiringPiSetup (void):** Inicializa la librería y se asume que el programa utilizara la distribución de pines de la librería, tal como se muestra en la figura 65.
- **pinMode (int pin, int mode):** Configura el modo de funcionamiento del pin, ya sea INPUT, OUTPUT, PWM_OUTPUT (únicamente pin 1), GPIO_CLOCK (únicamente pin 18).
- **digitalWrite (int pin, int value):** Escribe el valor HIGH (1) o LOW (0) al pin indicado, el cual debe haber sido configurado como salida.
- **pwmWrite (int pin, int value):** Escribe el valor al registro PWM del pin indicado, la Raspberry Pi posee un pin PWM (pin 1) con un rango de 0 a 1024.
- **digitalRead (int pin):** Retorna el valor leído del pin indicado, con valores HIGH (1) o LOW (0).

Raspberry Pi GPIO Header

BCM	WiringPi	Name	Physical	Name	WiringPi	BCM
		3.3v	1	2	5v	
2	8	SDA.1	3	4	5V	
3	9	SCL.1	5	6	0v	
4	7	1-Wire	7	8	TxD	15 14
		0v	9	10	RxD	16 15
17	0	GPIO. 0	11	12	GPIO. 1	1 18
27	2	GPIO. 2	13	14	0v	
22	3	GPIO. 3	15	16	GPIO. 4	4 23
		3.3v	17	18	GPIO. 5	5 24
10	12	MOSI	19	20	0v	
9	13	MISO	21	22	GPIO. 6	6 25
11	14	SCLK	23	24	CE0	10 8
		0v	25	26	CE1	11 7
0	30	SDA.0	27	28	SCL.0	31 1
5	21	GPIO.21	29	30	0v	
6	22	GPIO.22	31	32	GPIO.26	26 12
13	23	GPIO.23	33	34	0v	
19	24	GPIO.24	35	36	GPIO.27	27 16
26	25	GPIO.25	37	38	GPIO.28	28 20
		0v	39	40	GPIO.29	29 21

Figura 65. Distribución de pines Raspberry Pi 3B.

A continuación, se muestra un ejemplo básico del uso de la librería:

```
#include <wiringPi.h> // incluimos librería
int main (void)
{
    wiringPiSetup (); //inicializamos librería
    pinMode (0, OUTPUT); // configuramos el pin 0 como salida
    for (;;)
    {
        digitalWrite (0,HIGH) ; delay (500); //cambiamos el estado del
pin 0 a HIGH
        digitalWrite (0, LOW); delay (500); //cambiamos el estado del pin
0 a LOW
    }
    return 0;
}
```

1.2.13 Sistema de Protección SEL 351

Para la realización de pruebas con equipos físicos se adquirió un Relé de la marca SEL mostrado en la figura 66, con capacidad para utilizar el estándar IEC61850, pero sin la instalación del mismo de fábrica.

El relé cuenta con una diversidad de protocolos de comunicación los cuales se listan a continuación:

- IEC61850 (Opcional) MMS y GOOSE. Hasta 6 sesiones MMS, mensajes GOOSE con 24 suscripciones y 8 publicaciones.
- Modbus con configuración de mapas basada en etiquetas (serial y Ethernet), hasta 3 sesiones.
- DNP3 de Nivel 2 con configuración de mapas basada en etiquetas (serial y Ethernet), hasta 6 sesiones.
- IEEE C37.118-2005 Protocolo de sincrofasores.
- Telnet y web server integrado en Ethernet.
- Dual – channel MIRRORED BITS protocolo privativo de SEL

Dicho equipo se encuentra en subestaciones reales en el nivel de bahía según el estándar, y se encarga de las funciones de protección que se observan en la figura 67.



Figura 66. Relé SEL 351.

Debido a problemas con el acuerdo de usuario final de SEL al ser un equipo comprado a terceros, no es posible coordinar la instalación sin pago del estándar en el relé, por lo que se optó por desarrollar un Gateway entre los protocolos del estándar IEC61850 y el protocolo modbus, para esto se utilizó las librerías opensource libiec61850 y libmodbus. Así se establece comunicación con los simuladores y clientes instaladas en las PC a utilizar.

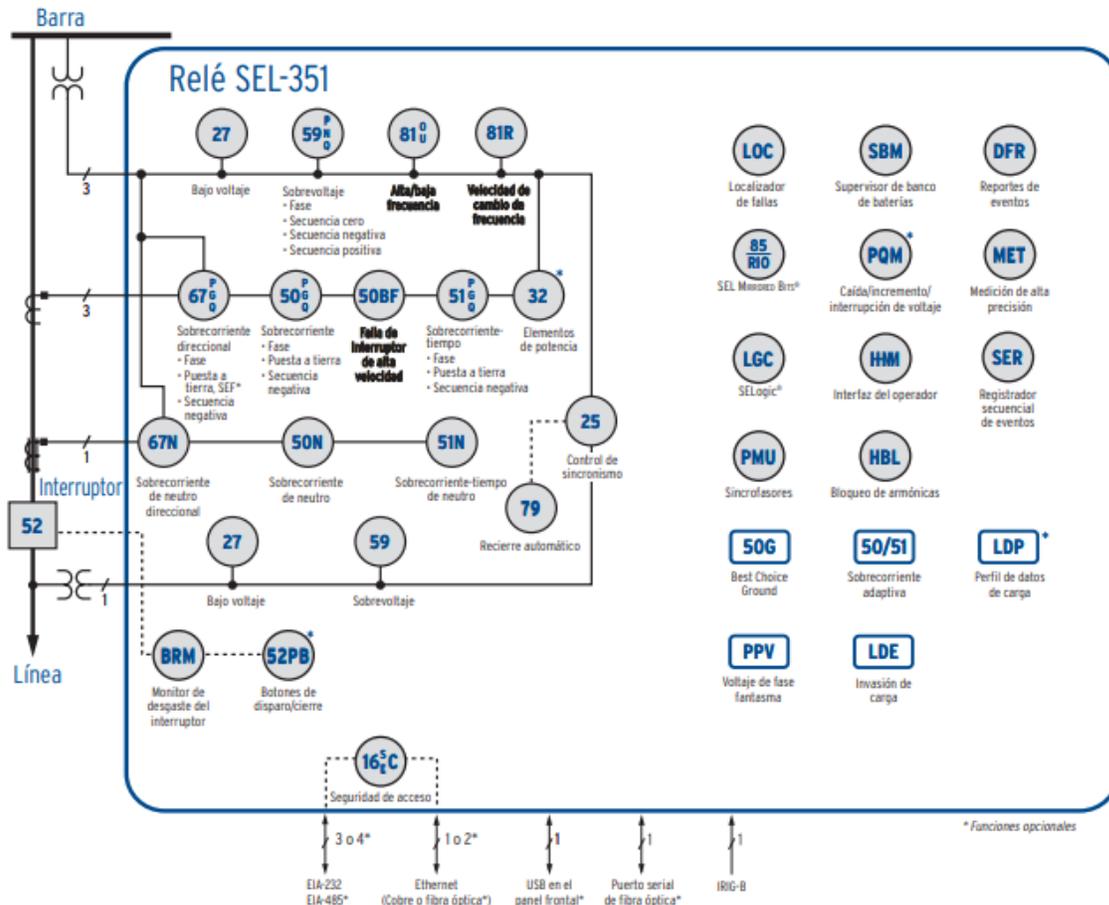


Figura 67. Funciones del Relé de Protección SEL351.

1.2.14 Raspberry PI 3B

Es una computadora de bajo costo y con un tamaño compacto, creada en febrero de 2012 por la Raspberry Pi Foundation con fines educativos. Ejecuta un sistema GNU/Linux y para los propósitos de este documento se desarrolló un programa descrito en el capítulo 2 el cual al ejecutarse en el OS Raspbian en la Raspberry funciona como un IED bajo el estándar IEC61850 y con la capacidad de controlar las entradas y salidas GPIO de la tarjeta.

Especificaciones técnicas de la placa Raspberry Pi 3B:

- CPU + GPU: **Broadcom BCM2837B0**, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
- RAM: **1GB** LPDDR2 SDRAM
- Wi-Fi + Bluetooth: 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac, Bluetooth 4.2, BLE
- Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps)
- GPIO de 40 pines
- HDMI+4 puertos USB 2.0

- Puerto CSI para conectar una cámara.
- Puerto DSI para conectar una pantalla táctil
- Salida de audio estéreo y vídeo compuesto
- Micro-SD
- Power-over-Ethernet (PoE)

En la figura siguiente se muestra la placa de desarrollo Raspberry pi con los pines GPIO que se utilizan para la creación de un IED emulado.

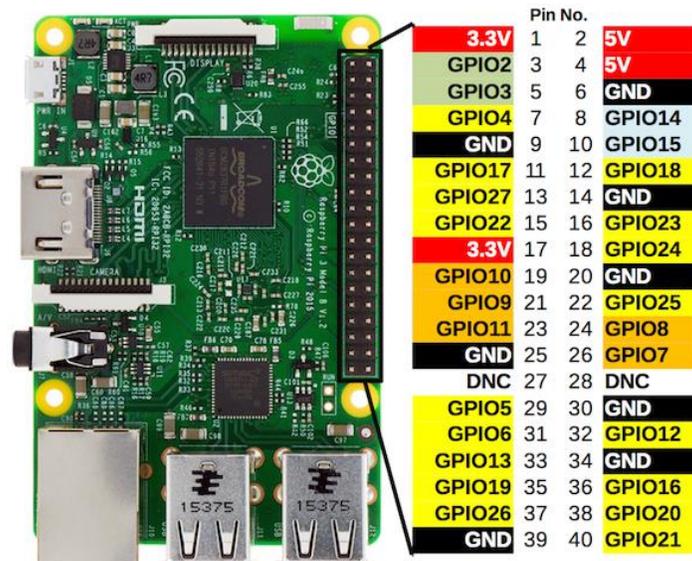


Figura 68. Pines GPIO Raspberry pi 3B.

1.2.15 Switch de comunicación

Para la creación de infraestructura de comunicación de red para el caso de un laboratorio con fines didácticos es necesario un switch que cumpla con los siguientes requisitos mínimos:

- Configuración de redes VLAN
- 8 puertos LAN 10/100BASE-TX
- Servidor DHCP, cliente DHCP, DHCP Relay
- Rutas estáticas y dinámicas (RIPv1 y RIPv2)
- DNS dinámico
- Soporte de IEEE 802.1d
- NAT/NAPT
- IEEE 802.1q etiquetado VLAN
- Soporte IPV6

En caso de un laboratorio de tipo profesional o montaje de equipo industrial es necesario que los equipos de la red de comunicación tengan compatibilidad con la norma, debido a que los tiempos de comunicación y confiabilidad son características de suma importancia, en contraste con la importancia de dichos parámetros para el montaje de equipos y software para propósitos didácticos.

Para propósitos de este documento se utiliza un router de uso residencial, el cual posee entre sus características la función de switch de 4 puertos, y con asignación de IP vía DHCP, modelo Claro A7600 A1, la figura 69 muestra una imagen del router.



Figura 69. Router Claro A7600 A1

CAPÍTULO II

Simuladores y emuladores para el desarrollo de escenarios
para pruebas bajo la norma IEC61850

2.0 Introducción

El presente capítulo describe la configuración de equipos y software utilizados para la implementación de escenarios de pruebas desarrollados en el capítulo 3 de este documento. La figura 70 muestra un esquema general de la comunicación entre simuladores y emuladores, el cual está basado en una Red LAN de tipo estrella en el cual los equipos simulados y emulados se comunicarán vía MMS y GOOSE, este esquema se segmenta para crear escenarios específicos para la configuración de protocolos del estándar.

El presente capítulo se ha distribuido en tres apartados los cuales se recomienda seguir de manera ordenada para la implementación correcta de los escenarios, estos apartados se enumeran a continuación:

1. **Configuración de relés simulados:** Se realiza la configuración de los Relés INFOTECH y Simulator Bay de Axon Group.
2. **Configuración de relés emulados:** Se realiza la configuración de los Relés SEL351 y Raspberry Pi.
3. **Configuración SCADA:** En este apartado se configura la interfaz gráfica del control y supervisión de los escenarios.

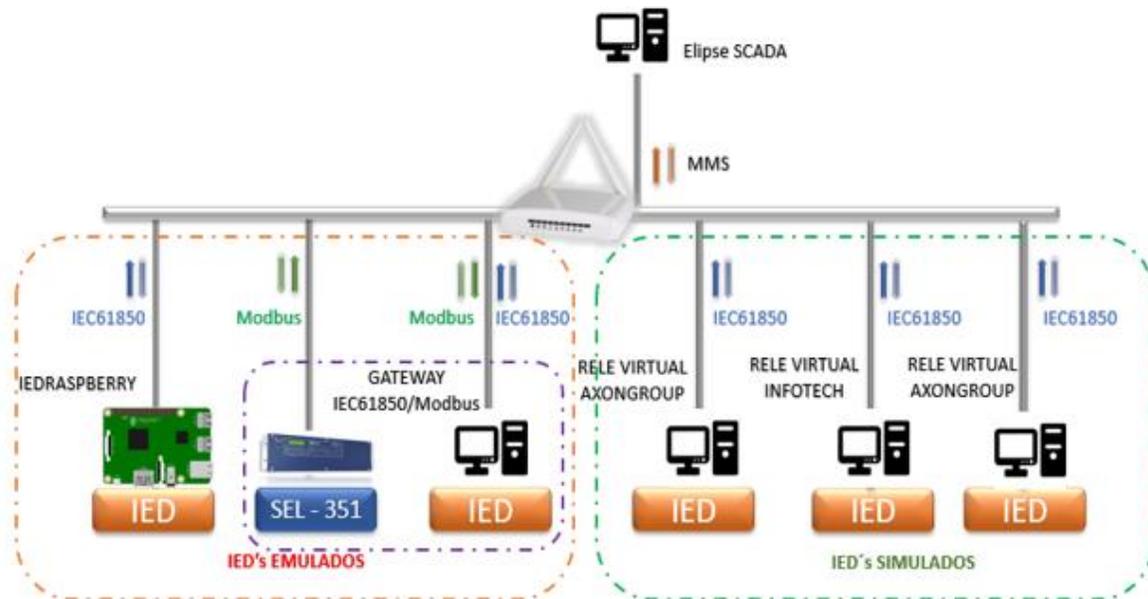


Figura 70. Red de comunicación para implementación de escenarios.

2.1 CONFIGURACIÓN DE RELÉS SIMULADOS

En esta sección se describen los parámetros configurados en los relés virtuales de la marca Infotech y Axon Group, para los escenarios de pruebas del capítulo 3 de este documento, se incluyen mensajes GOOSE para el disparo del relé y para el monitoreo del mismo con sus componentes.

2.1.1 Relé INFOTECH

Para poder obtener el modelo de datos del Relé Infotech y de esta manera efectuar las configuraciones correspondientes para la comunicación con los demás dispositivos, se utiliza el software IED Explorer, descrito en la sección 1.2.3 de este documento, se configuran por defecto los siguientes parámetros:

- IP: 127.0.0.1.
- Puerto: 502.

Una vez iniciado y configurado se podrá visualizar su modelo de datos tal como se muestra en la figura 71.

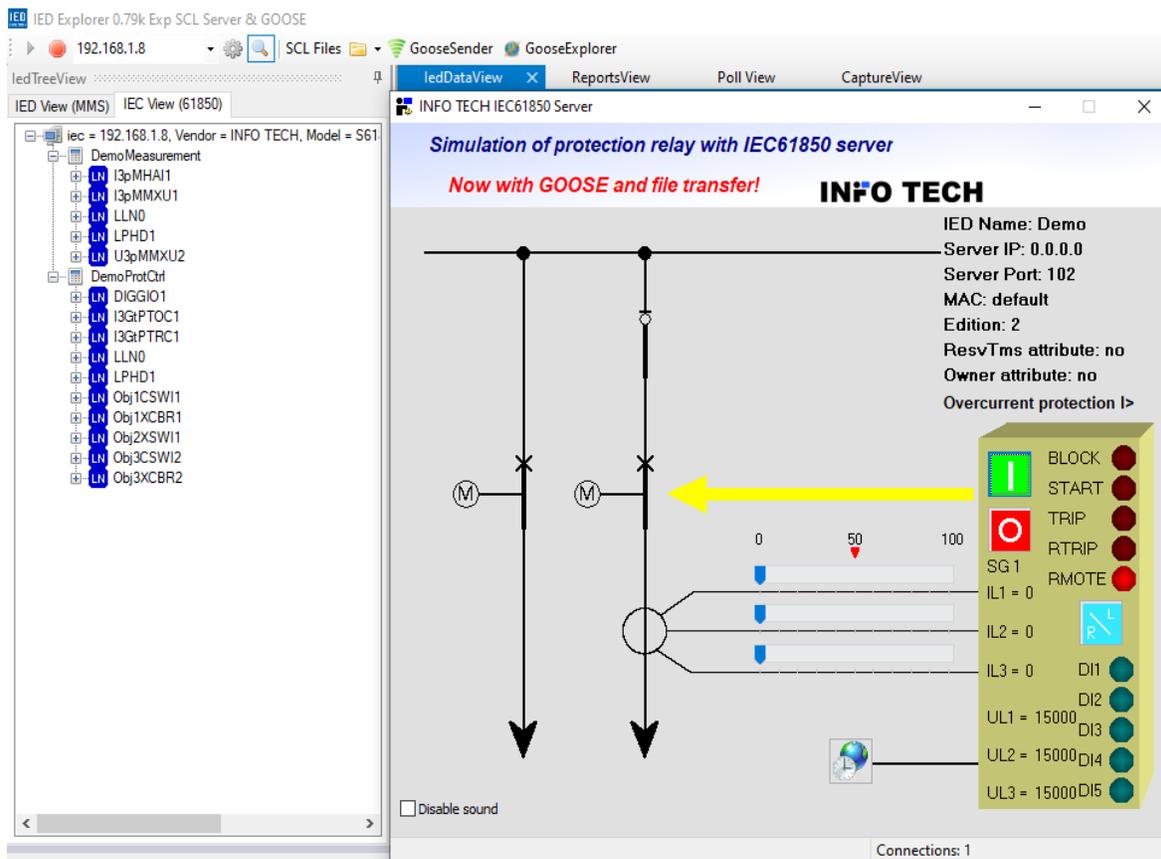


Figura 71. Modelo de Datos del Relé Infotech.

En el modelo de datos del Relé es posible observar que este posee dos dispositivos lógicos (LD) que corresponden a las medidas y protecciones del Relé Infotech los cuales son:

- 1) **LD DemoMeasurement:** posee cinco nodos lógicos (LN), dos de ellos son los que la norma ha establecido obligatorios *LLN0* y *LPHD* los cuales fueron explicados en el capítulo anterior, El *LN I3pMHAI1* representa las mediciones de los armónicos del Relé, el *LN I3pMMXU1* y *U3pMMXU* representa las mediciones de corriente (IL1, IL2, IL3) y voltaje (VL1, VL2, VL3) del Relé respectivamente.
- 2) El **LD DemoProtCtrl:** posee diez nodos lógicos, *LLN0* y *LPHD* los cuales son nodos no lógicos obligatorios, el *LN Obj1XCBR1* y *Obj3XCBR2* contienen la información de los interruptores de potencia presentes en el Relé, *Obj1CSWI1* y *Obj3CSWI2* contienen la información para el control de los interruptores de potencia, el *LN Obj2XSWI1* posee la información del seccionador del Relé.

Los *LN I3GtPTOC1* e *I3GtPTRC1* poseen las protecciones del Relé para disparos por sobre corriente y disparo general respectivamente, por último, se encuentra el nodo lógico *DIGGIO1* este LN contiene las entradas y salidas genéricas del Relé, se estará utilizando este nodo lógico para los indicadores presentes en el *Relé (DI1 ... DI5)*.

Como se muestra en la sección 1.1 de este documento IEC61850 posee un modelo de datos jerárquico por lo que cada uno de estos nodos lógicos tendrán dentro de ellos objetos de datos y atributos de datos que representarán las funciones o parámetros de cada uno de los elementos presentes en relé.

La figura 72 muestra el contenido del nodo lógico *Obj1XCBR1* (Interruptor 1) el cual posee objetos de datos (DO) y atributos de datos (DA), para el ejemplo se ha tomado el DO *Pos* el cual contiene la posición del interruptor y dentro del objeto de datos se encuentran los atributos de datos como el estado del interruptor (*stVal*), calidad de comunicación (*q*) y estampa de tiempo (*t*), tal como se puede observar en los DA aparecen entre paréntesis las restricciones funcionales (FC) a los cuales pertenecen. Para visualizar cualquiera de estos atributos posicionar en el DA, para el ejemplo se ha tomado el *stVal* del interruptor, lo cual permite observar el estado del interruptor en el *IEDDataView*, para este caso ya que el interruptor se encuentra abierto el valor del DA corresponde a 01.

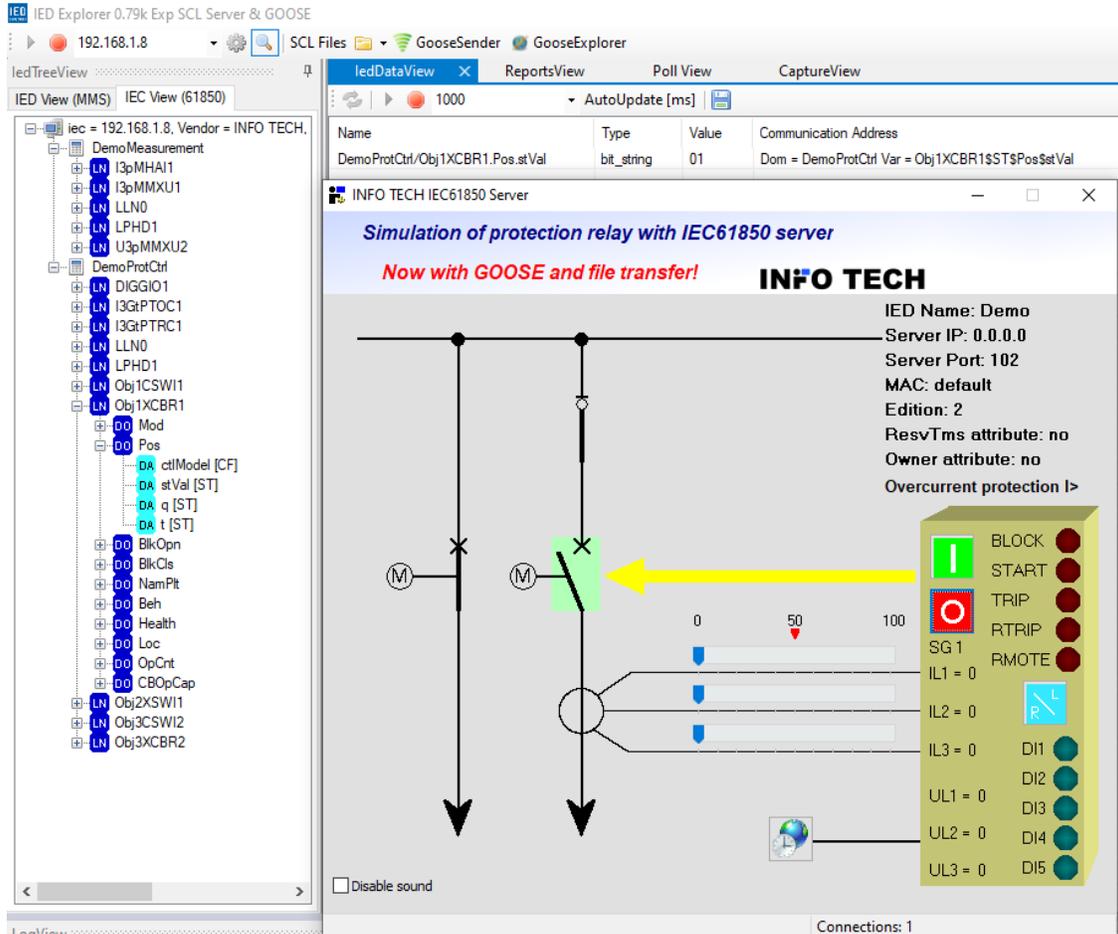


Figura 72. Atributos y Objetos de Datos del LN Obj1XCBR1.

2.1.1.1 Habilitar Mensajes GOOSE del Relé INFOTECH

El software del relé INFOTECH incorpora Reportes bufereados y no bufereados, así como bloques de control de mensajes GOOSE en su modelo de datos, como ya se explicó en el capítulo 1 estos mensajes y reportes se encuentran configurados en el nodo lógico LNN0 así mismo, estarán ubicados las configuraciones de los DataSet (DS). La figura siguiente muestra los DS, Reportes y mensajes GOOSE de los LD del Relé.

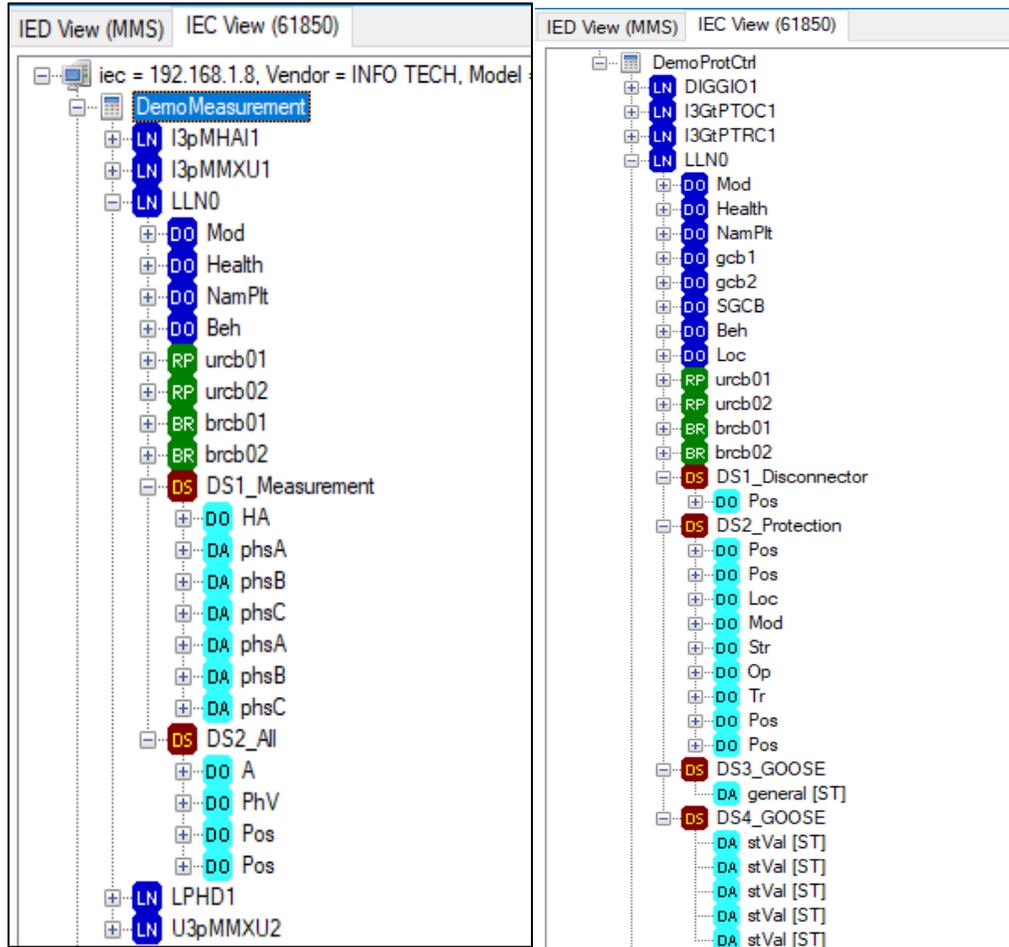


Figura 73. DataSet, Reportes y Mensajes GOOSE del Relé INFOTECH.

La figura 73 muestra los DataSet de los dispositivos lógicos de medidas y protección, en el cual el LD *DemoMeasurement* posee en su modelo de datos dos DataSet (*DS1_Measurement* y *DS2-All*) los cuales corresponden a las medidas de armónicos y medidas de corriente y voltajes).

El LD *DemoProtCtrl* posee cuatro DataSet (*DS1_Disconnector*, *DS2 Protection*, *DS3_GOOSE* y *DS4_GOOSE*), los cuales corresponden al estado del interruptor, atributos de la protección por sobre corriente, trigger del interruptor y estados de los indicadores, respectivamente.

Los dos dispositivos lógicos del Relé también contienen reportes bufereados (**brcb**) y no bufereados (**urcb**) los cuales están configurados con cada uno de los DataSet antes mencionados. El LD *DemoProtCtrl* además de contener reportes, posee dos mensajes goose **gcb1** el cual posee los datos contenidos en el *DS3_GOOSE* y **gcb2** que posee los datos del *DS4_GOOSE* estos mensajes pueden activarse o desactivarse desde el IEDExplorer.

Para propósitos de este documento únicamente se utiliza el mensaje **gcb1** el cual contiene el trigger del Interruptor 1, para la activación de dicho mensaje se debe seleccionar el control block (gcb1) y abrir su contenido, en el submenú que se despliega se selecciona GoEna (GO) y se hace clic derecho sobre él, seguidamente se selecciona la opción Write Data, al realizar estos pasos se abrirá la ventana mostrada en la figura 74, para activar el mensaje se digita “True” y finalmente se hace clic en OK.

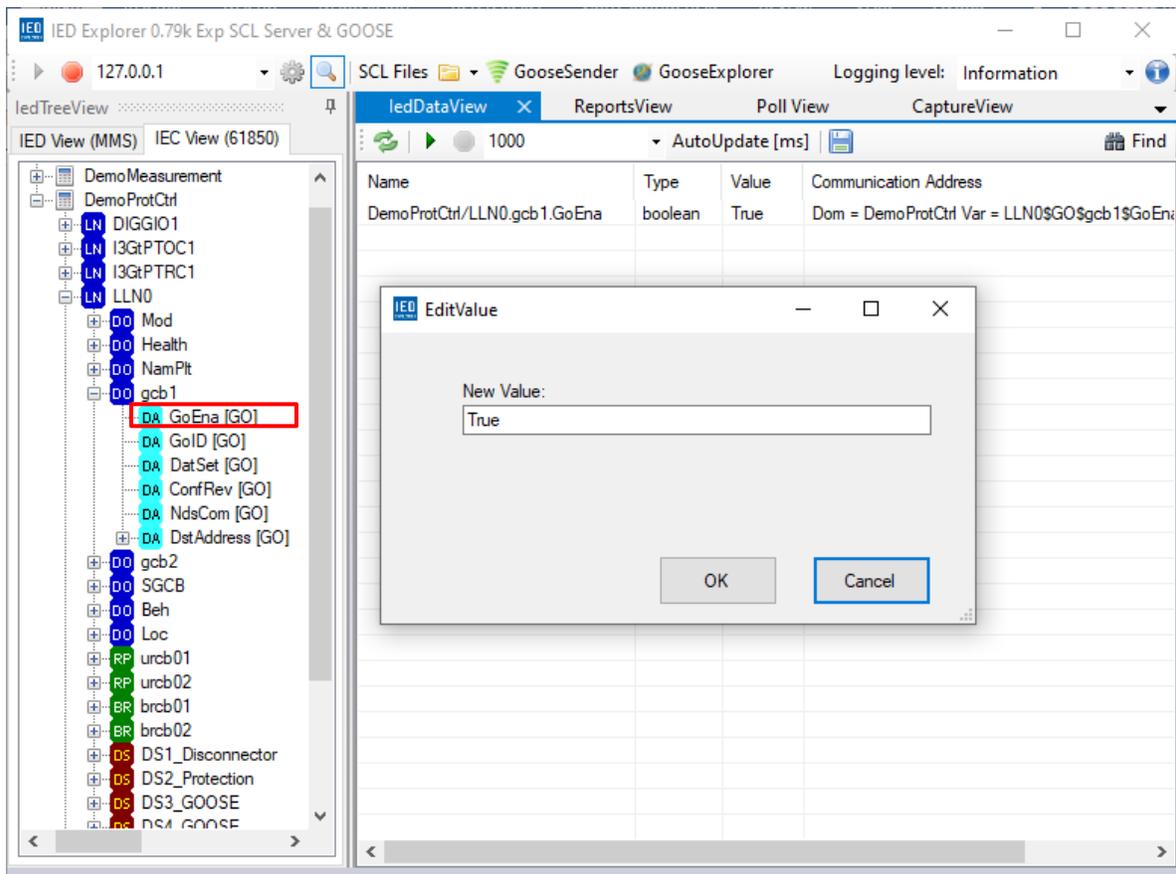


Figura 74. Habilitar mensajes GOOSE.

De esta forma quedaran habilitados los mensajes GOOSE de gcb1. Para **gcb2** (estados de los indicadores) se realizan los pasos antes descritos pero esta vez se digita la palabra false, esta acción desactiva el mensaje GOOSE para los indicadores.

Para poder observar el mensaje GOOSE del Relé INFOTECH que se ha habilitado se utiliza la herramienta GOOSE Receiver, para la suscripción del mensaje GOOSE se realizan los pasos descritos en el apartado 1.2.4 del capítulo 1, una vez suscritos al mensaje aparecerán los datos contenidos en él, tal como se muestra en la siguiente figura.

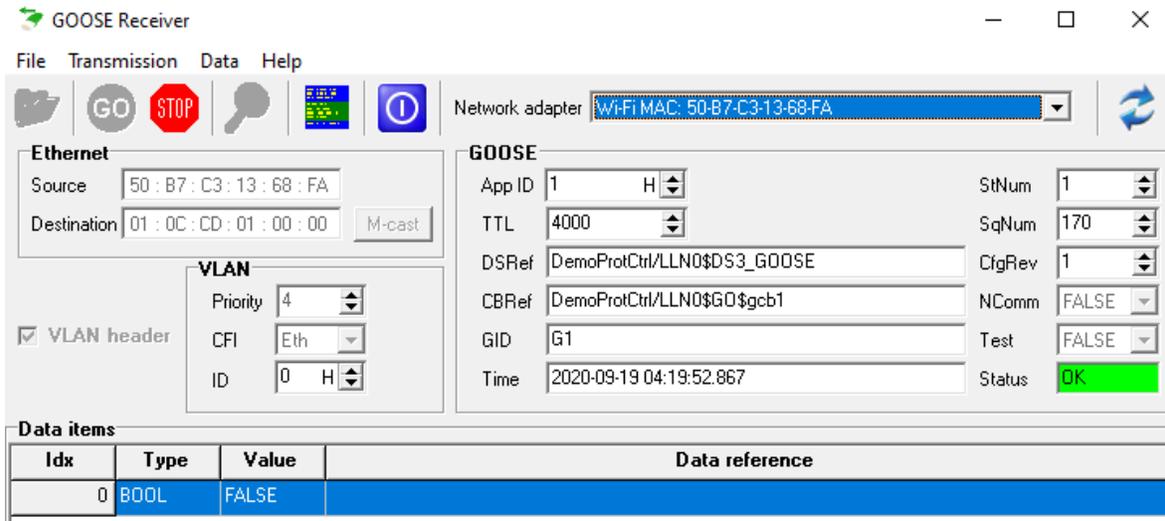


Figura 75. Datos del mensaje GOOSE gcb1.

Como se logra observar en la figura 75, el parámetro DSRef contiene el DataSet del mensaje GOOSE, CBRef contiene la trama de datos del mensaje GOOSE, el identificador del mensaje se encuentra en GID y el App ID se identifica como 1. Cada uno de estos parámetros son de utilidad para la configuración de un publicador desde el IEDRaspberry (Apartado 1.2.1), para propósitos de este documento dicho mensaje será el encargado de indicar a los otros simuladores cuando exista una falla en el relé INFOTECH y de esta manera permita realizar cambios en los demás dispositivos simulados.

También es necesario especificar el adaptador de red por donde se enviarán los mensajes, para algunos casos también será necesario especificar la MAC de destino.

El Relé INFOTECH cuenta con el GOOSE Publisher (ver apartado 1.2.5 capítulo 1), este envía un mensaje GOOSE para el disparo del interruptor 1, sin embargo, no se visualiza en el modelo de datos del Relé ya que ha sido programado internamente por el fabricante, pero es posible obtener sus parámetros utilizando el GOOSE Receiver. Para ello se presiona el botón “Go” ubicado en el programa GOOSE Publisher, al realizar esta acción se publicará el mensaje GOOSE en la red, la suscripción del mensaje se realiza a través del GOOSE Receiver el nuevo mensaje GOOSE tiene un AppId 3 y GoID G3, el cual corresponde al mensaje GOOSE enviado por el software, la figura 76 muestra lo descrito anteriormente.

Available GOOSE streams

Idx	Source MAC	Destination MAC	App ID	Config Rev	GOOSE ID	Messages	TEST	NDSCOM
1	50:B7:C3:13:68:FA	01:0C:CD:01:00:00	0001	1	G1	440	FALSE	FALSE
2	50:B7:C3:13:68:FA	01:0C:CD:01:00:02	0003	1	G3	32	FALSE	FALSE

GOOSE Publisher

GOOSE Publisher

INFO TECH

Network adapter: Wi-Fi MAC: 50-B7-C3-13-68-FA

Go
 Stop
 About

■

■

Running
 ■

Figura 76. Mensaje GOOSE del software GOOSE Publisher.

Luego que se identifica el mensaje se hace clic en el botón Subscribe, esta acción mostrara los parámetros contenidos en el mensaje GOOSE.

La figura 77 muestra los datos del mensaje para el cual el App ID tiene un valor de 3, el DataSet de Referencia es LLN0\$DS1, el CRef es LLN0\$gcb1 y el GID es igual a G3, para este caso también se configura la MAC del Relé INFOTECH (01:0C:CD:01:00:02).

GOOSE Receiver

File Transmission Data Help

Network adapter: Wi-Fi MAC: 50-B7-C3-13-68-FA

Ethernet

Source: 50 : B7 : C3 : 13 : 68 : FA

Destination: 01 : 0C : CD : 01 : 00 : 02

VLAN

VLAN header

Priority: 4

CFI: Eth

ID: 0

GOOSE

App ID: 3

TTL: 16384

DSRef: LLN0\$DS1

CRef: LLN0\$gcb1

GID: G3

Time: 2020-09-19 05:24:17.735

StNum: 1

SqNum: 13

CfgRev: 1

NComm: FALSE

Test: FALSE

Status: OK

Data items

Idx	Type	Value	Data reference
0	BOOL	FALSE	
1	BOOL	FALSE	

Figura 77. Parámetros GOOSE para el disparo del interruptor INFOTECH.

Conocidos los parámetros del mensaje GOOSE es posible activar o desactivar los disparos del Interruptor de potencia desde un software o dispositivo que publique mensajes GOOSE, como por ejemplo el GOOSE Sender, IEDRapsberry PI o el Relé SEL 351. La Figura 78 muestra la configuración de dicho mensaje desde el GOOSE Sender.

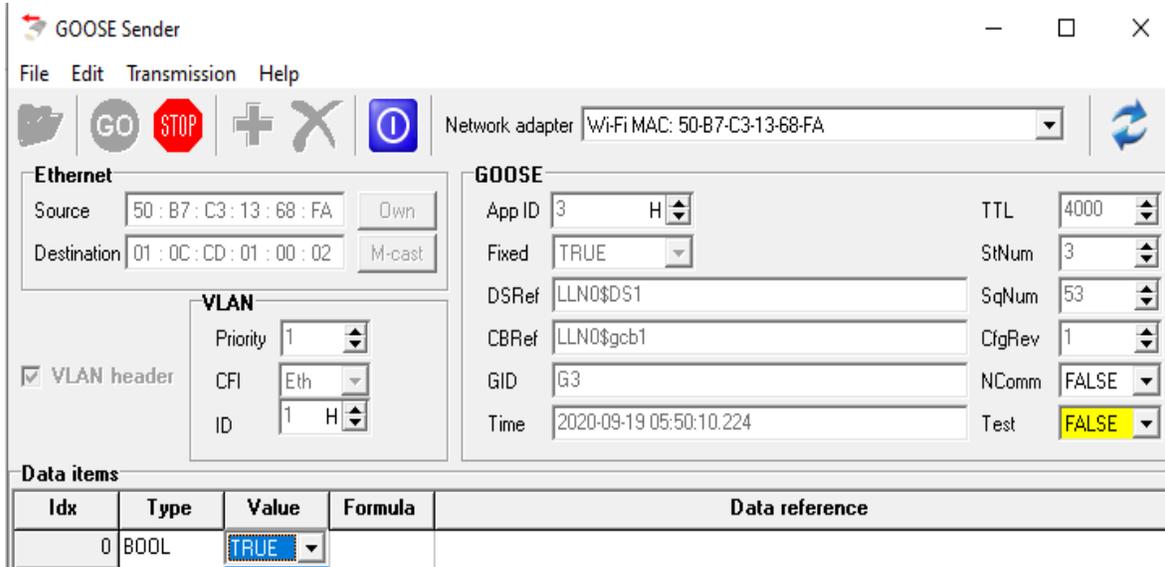


Figura 78. Envío de mensaje GOOSE desde el GOOSE Sender.

2.1.1.2 Habilitar Reportes del Relé INFOTECH

Para visualizar los reportes que contiene el Relé es necesario habilitarlos desde el IEDExplorer tal como se hizo con los mensajes GOOSE, para ello se selecciona el reporte que se desea habilitar y se hace clic derecho sobre él, en el menú que se despliega se selecciona *Configure RCB*, esta acción abrirá una nueva ventana la cual se muestra en la figura 79.

Para el ejemplo se ha tomado el reporte **ucrb02**, el cual contiene las medidas de corriente del relé INFOTECH, la ventana **Configure and Actívale RCB** muestra cada uno de los parámetros que contiene el reporte, se debe ubicar el item RptEna (Report Enable) y seleccionar RptEna, luego se hace clic en Send y de esta forma quedara habilitado el reporte. Los reportes que se habiliten pueden ser visualizados en la sección ReportsView, tal como se explicó en la sección 1.2.3.5.

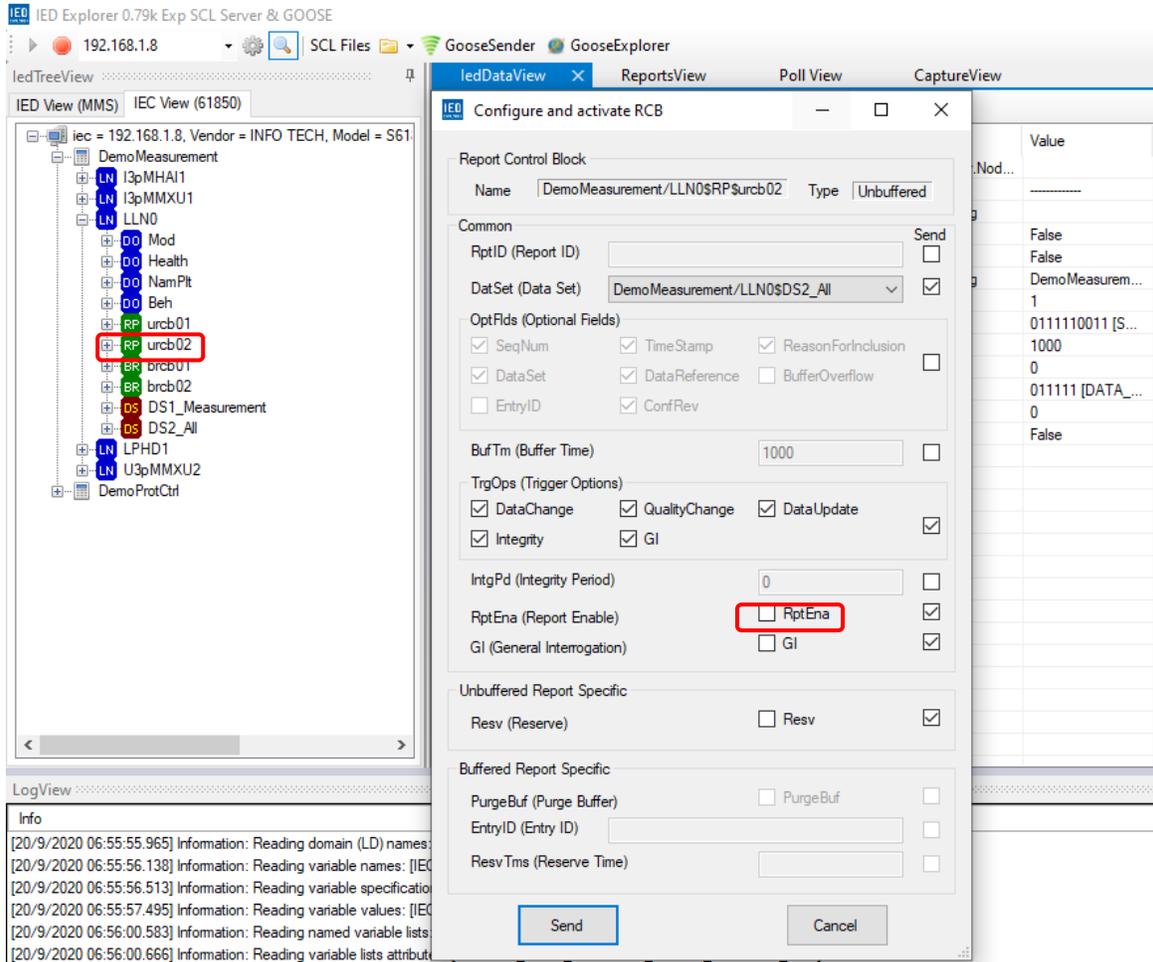


Figura 79. Habilitar Reportes Almacenados y No Almacenados.

2.1.2 Relé Simulator Bay Axon Group

Para la configuración del Relé Simulator Bay de Axon Group será necesario conocer su modelo de datos, esto se realizará utilizando el programa IED Explorer tal como se hizo con el Relé del apartado anterior, a diferencia del Relé INFOTECH el Relé Axon Group permite modificar el puerto de comunicación y el adaptador de red, para los escenarios que se estarán presentando en el capítulo 3 será necesario configurar al Relé Axon Group en el puerto de comunicación 103 y el adaptador de red en Ethernet esto se realiza desde la opción Settings, una vez configurados estos parámetros se hace clic en Restart, tal como se muestra en la figura siguiente.

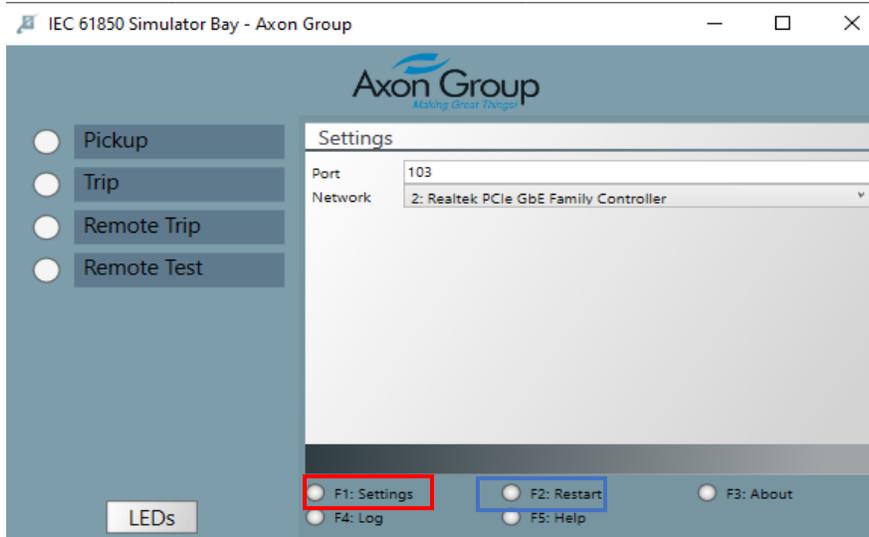


Figura 80. Configuración de comunicación del Relé AxonGroup.

Una vez el Relé ha sido configurado con los parámetros de comunicación correctos, es posible ver su modelo de datos con el IED Explorer el cual se muestra en la figura 81.

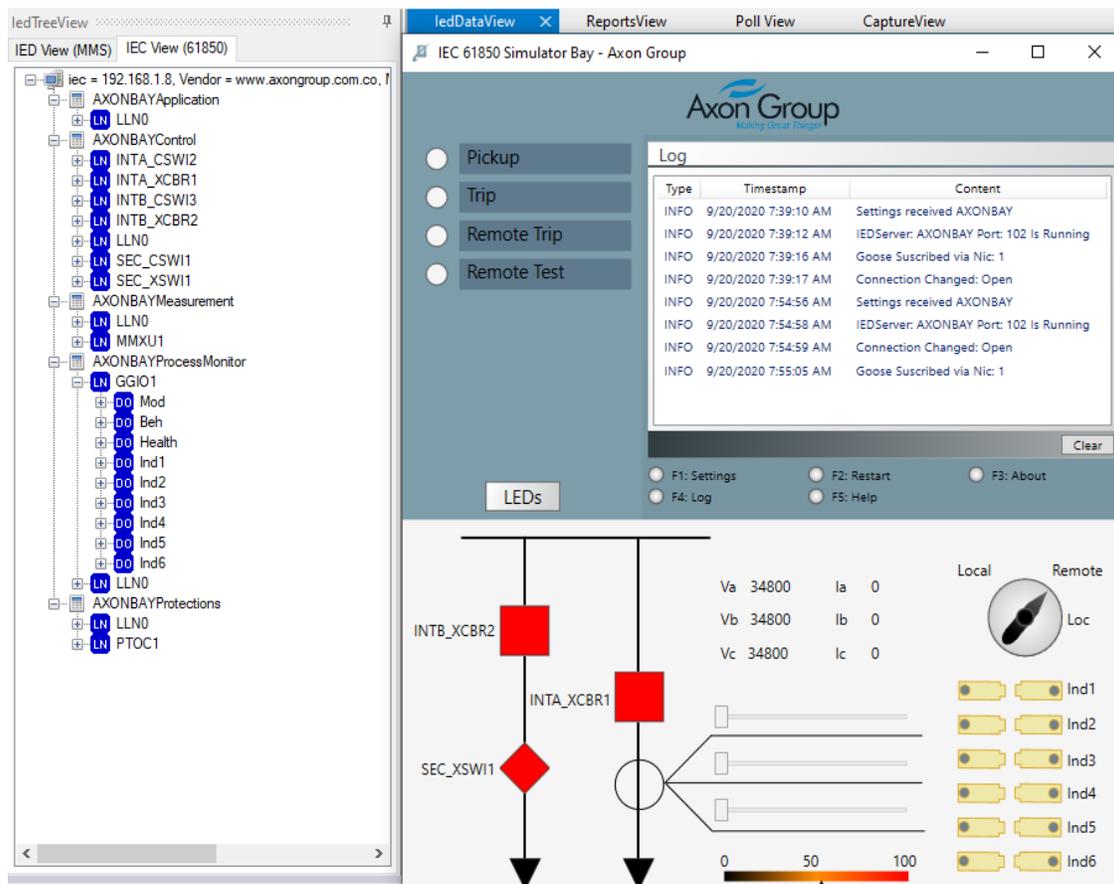


Figura 81. Modelo de Datos del Relé Simulator Bay de Axon Group.

En el costado izquierdo de la figura anterior se observa el modelo de datos del Relé Axon Group este posee cinco dispositivos lógicos (LD) los cuales son:

- 1) **AXONBAYApplication:** Este dispositivo contiene toda la configuración de los DataSet, Reportes y Mensajes GOOSE del Relé, a diferencia del relé INFOTECH el relé Axon Group ubica todas sus configuraciones en un único Dispositivo Lógico.
- 2) **AXONBAYControl:** Posee los datos de los interruptores y seccionadores del relé, así como los comandos de control de cada de ellos.
- 3) **AXONBAYMeasurement:** contiene los datos de medidas de voltaje y corriente del Relé AxonGroup.
- 4) **AXONBAYProcessMonitor:** Posee los datos de los indicadores y el LD AXONBAYProtections posee los datos de las protecciones del Relé.

Para poder visualizar los Dataset, Reportes y Mensajes GOOSE del Relé AxonGroup se debe ubicar en el nodo lógico 0 (LLN0) del LD *AXONBAYApplication*, los cuales se muestran en la figura 82.

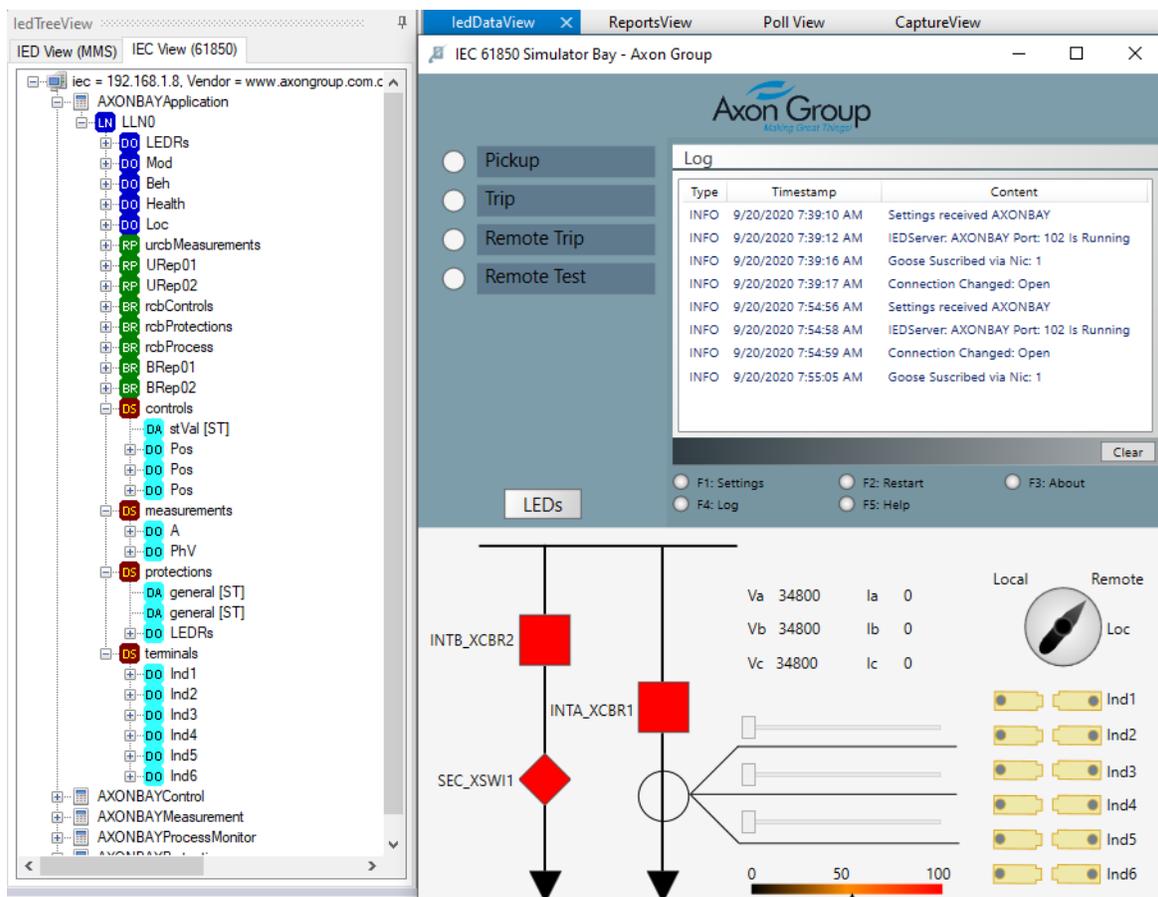


Figura 82. DataSet, Reportes y Mensajes GOOSE del Relé Simulator Bay.

Tal como se muestra en la figura anterior el Relé AxonGroup posee cuatro DataSet los cuales son:

- 1) **Controls:** Posee las posiciones de los Interruptores y seccionadores del Relé.
- 2) **Measurements:** El cual contiene las medidas de voltaje y corriente.
- 3) **Protecciones:** Este DS contiene los estados del trigger y de las protecciones.
- 4) **Terminals:** el cual posee el estado de los indicadores.

El relé incluye reportes bufereados y no bufereados los cuales corresponden a cada uno de los DS antes mencionados, para habilitarlos se realizan los mismos pasos que se describieron el apartado del Relé INFOTECH.

Como se ha podido observar en el modelo de datos del relé AxonGroup los nodos lógicos (LN) presentes en los dispositivos lógicos, son idénticos al del relé INFOTECH esto es lo que cumple con el primer objetivo de la norma IEC61850 el cual es la interoperabilidad de los dispositivos.

2.1.4.1 Habilitar Mensajes GOOSE del Relé Simulator Bay

El Relé Simulator Bay de Axon Group únicamente tiene un mensaje GOOSE el cual representa al estado de los indicadores, para propósitos de este documento dicho mensaje no se estará utilizando por lo que lo se desactiva para no saturar la Red con mensajes GOOSE innecesarios. Para deshabilitar este mensaje se seguirán los pasos descritos en el apartado del Relé INFOTECH, para el caso se escribe la palabra False tal como se muestra en la figura 83.

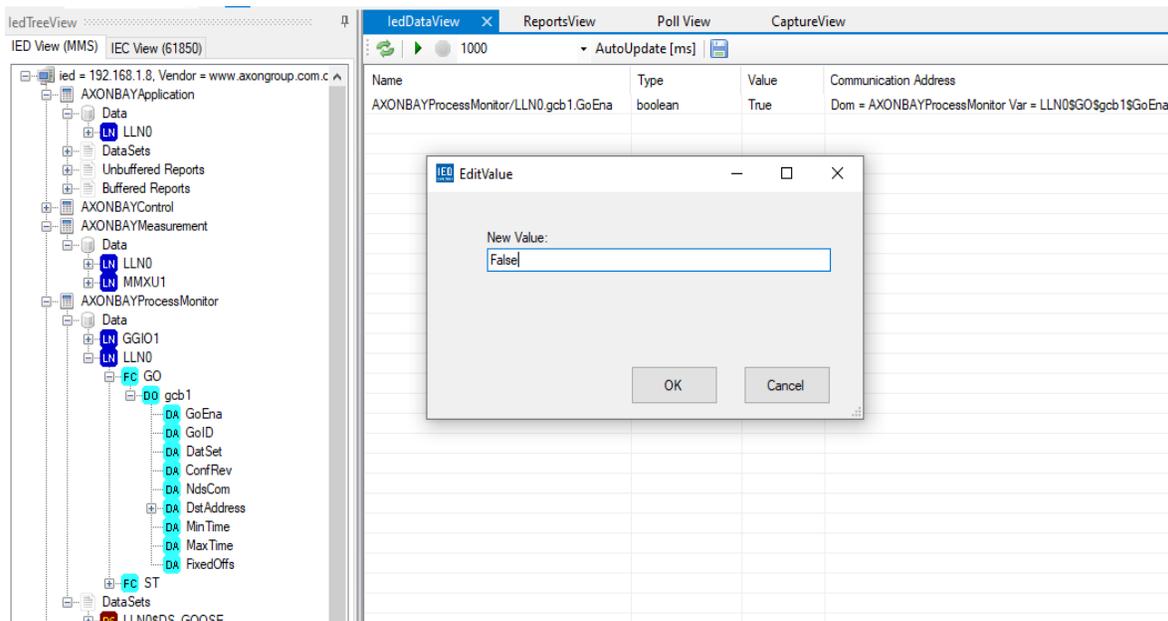


Figura 83. Desactivando envío de mensajes GOOSE.

El Relé Simulator Bay también posee un publicador de mensajes GOOSE el cual envía mensajes para disparos del interruptor de potencia “INTA_XCBR1”, este mensaje GOOSE al igual que el del Relé INFOTECH no se puede visualizar en el modelo de datos del relé ya que ha sido programado internamente por el fabricante, por lo cual se utiliza el software GOOSE Receiver para conocer los parámetros con los que ha sido programado el mensaje GOOSE, para ello se inicia el programa *Remote Protection Simulator* y se selecciona el adaptador de Red (Etherneth) donde se enviara el mensaje GOOSE, al efectuar esta acción se observa el mensaje en el GOOSE Receiver, tal como se muestra en la figura siguiente.

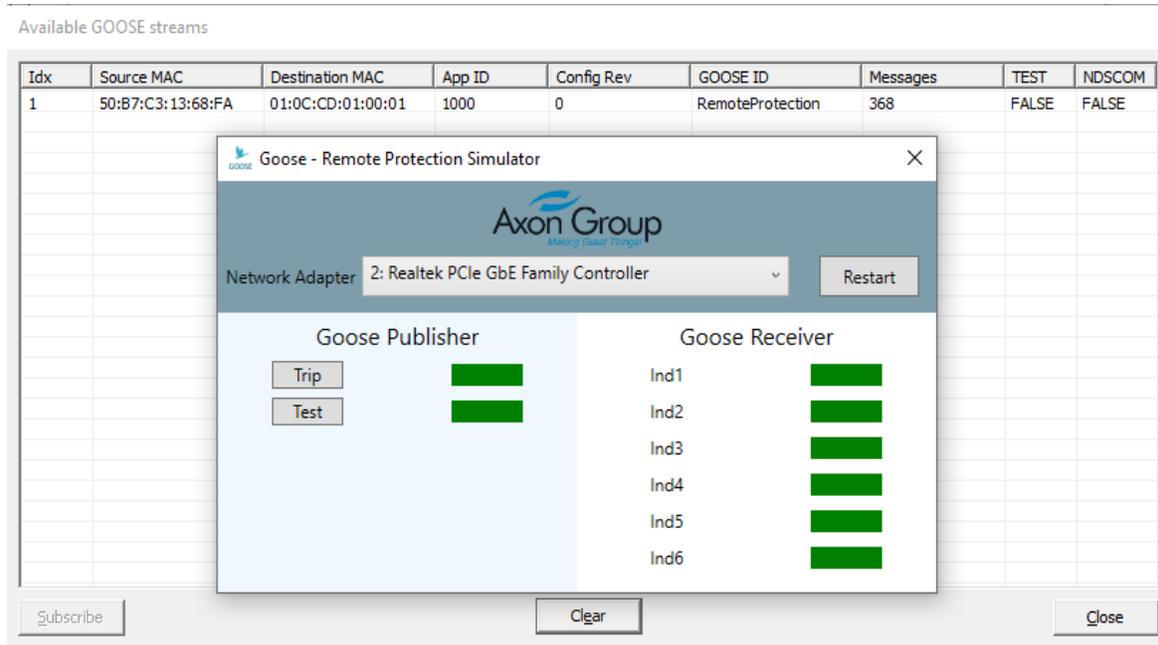


Figura 84. Mensaje GOOSE del Software Remote Proteccion Simulator.

Luego de identificar el mensaje GOOSE del publicador, se selecciona y se hace clic en Subscribe, esto permite visualizar los parámetros del mensaje en el GOOSE Receiver.

La figura 85 muestra los datos del mensaje para el cual el App ID tiene un valor de 1000, el DataSet de referencia es:

```
REMOTEProtections/LLN0$DS1_GOOSE
```

El CRef es:

```
REMOTEProtections/LLN0$GO$gcb1
```

El GID es *RemoteProtection*, la MAC de destino o del Relé Simulator Bay será 01:0C:CD:01:00:01.

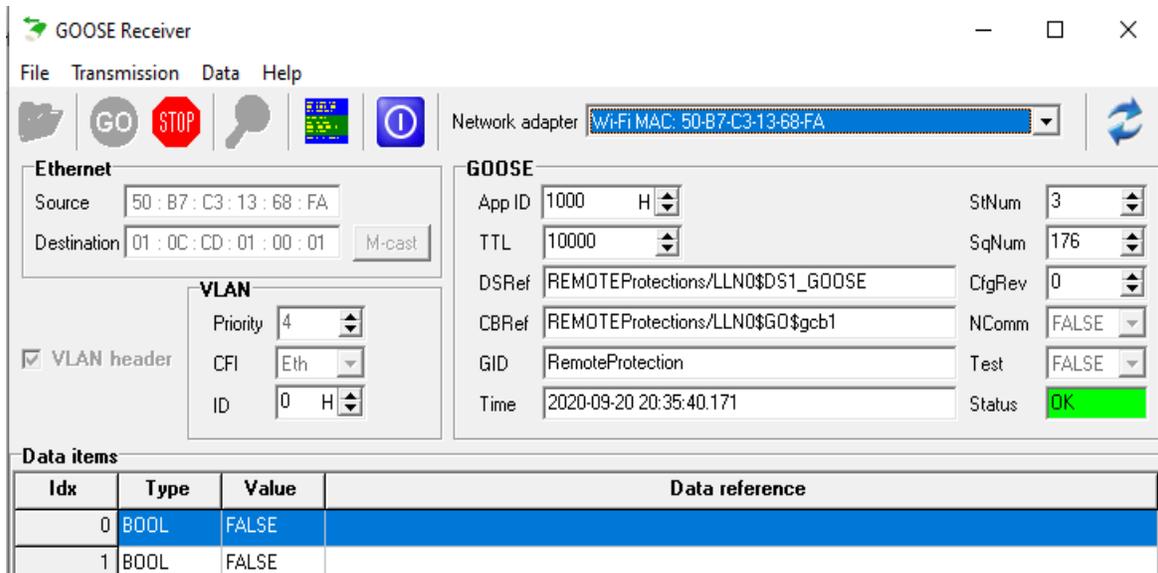


Figura 85. Parámetros GOOSE para el disparo del interruptor INTA_XCBR1.

Una vez se conozca los parámetros del mensaje GOOSE es posible activar o desactivar los disparos del Interruptor de potencia desde un software o dispositivo que publique mensajes GOOSE, como por ejemplo el GOOSE Sender, la Raspberry PI o el Rele SEL 351.

La Figura 86 muestra la configuración de dicho mensaje desde el GOOSE Sender.

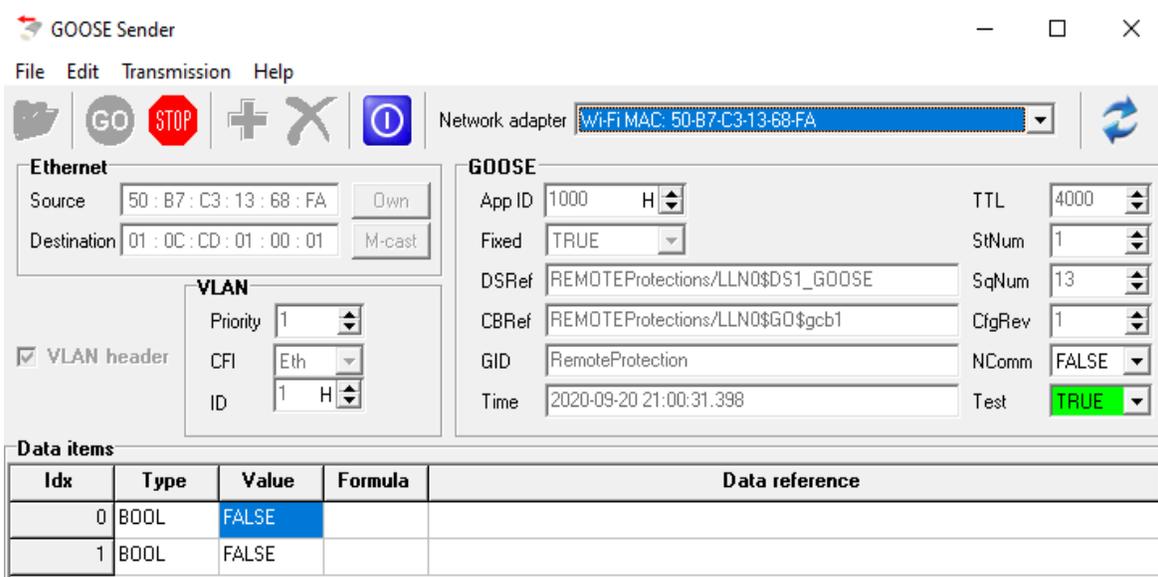


Figura 86. Envío de mensaje GOOSE desde el GOOSE Sender.

2.2 CONFIGURACIÓN DE RELÉS EMULADOS

En esta sección se aborda la programación y configuración de emuladores de IEDs bajo el estándar, específicamente utilizando la librería libiec61850 la cual permite crear un servidor emulando un dispositivo en una PC y mediante el control de entradas/salidas tener funciones básicas para la realización de pruebas con señales e indicadores físicos, los dispositivos emulados en este documento son el equipo IEDRaspberry y el Gateway IEC61850/Modbus.

2.2.1 Equipo IEDRaspberry y sus componentes

Para la realización de pruebas utilizando equipos físicos se optó por el desarrollo de un IED de bajo costo utilizando software y hardware de licencia libre, para esto se utilizó la placa de desarrollo Raspberry PI 3B+, en la cual se ejecuta el OS GNU/Linux Raspbian y donde a partir de la librería libiec61850 se emula un servidor IEC61850 con la mayoría de las funcionalidades de un IED, en el capítulo 3 se utiliza este IED para emular un dispositivo el cual monitorea mediante sus entradas digitales las protecciones mecánicas de un transformador de potencia y manda señales de disparo GOOSE a otros IEDs de la red. Las características principales del IEDRaspberry son:

- 4 entradas digitales.
- 5 salidas digitales con relevadores NA.
- Puerto Ethernet de comunicación.
- Servidor IEC61850.
- Suscripción/publicación de mensajes GOOSE.
- Envío/Recepción de comandos MMS.

La figura 87 muestra de forma gráfica los componentes del IEDRaspberry.

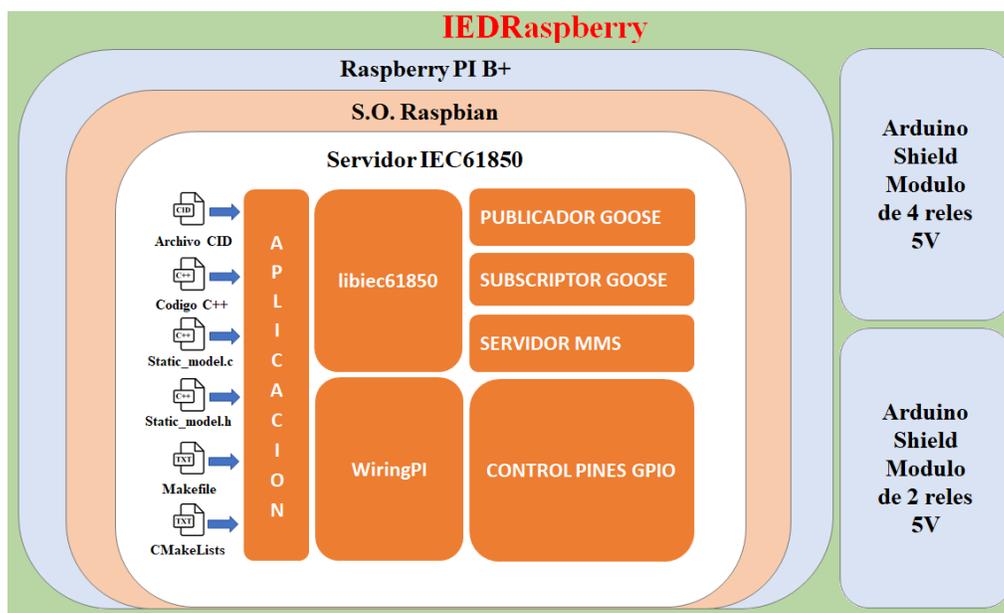


Figura 87. Estructura IEDRASPBERRY.

2.2.1.1 Interfaz física

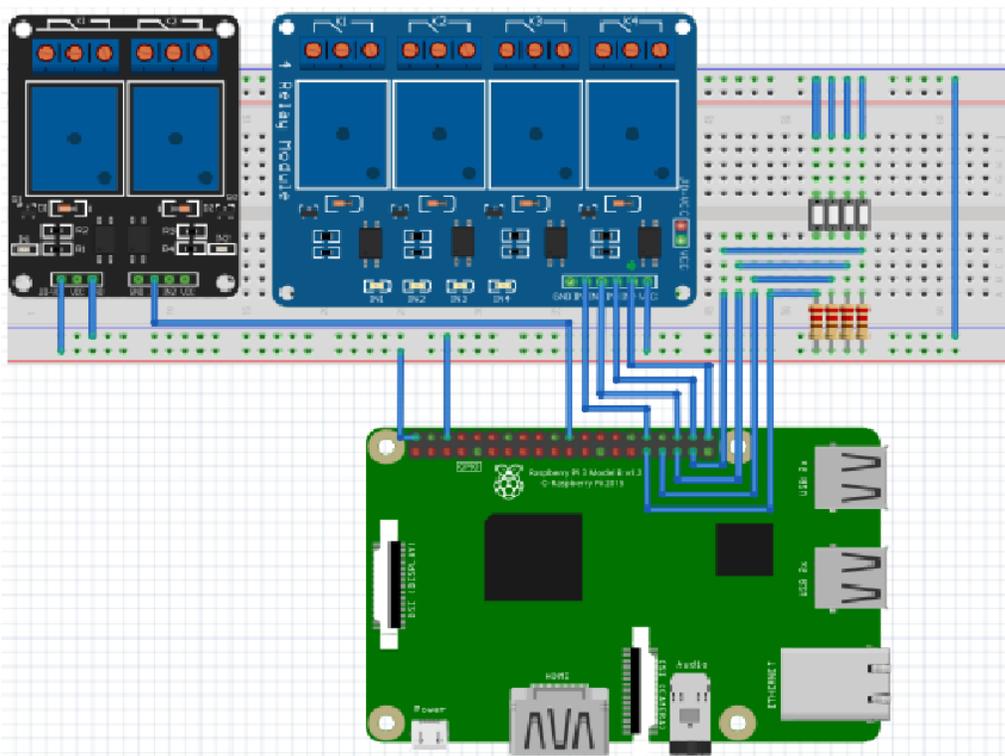
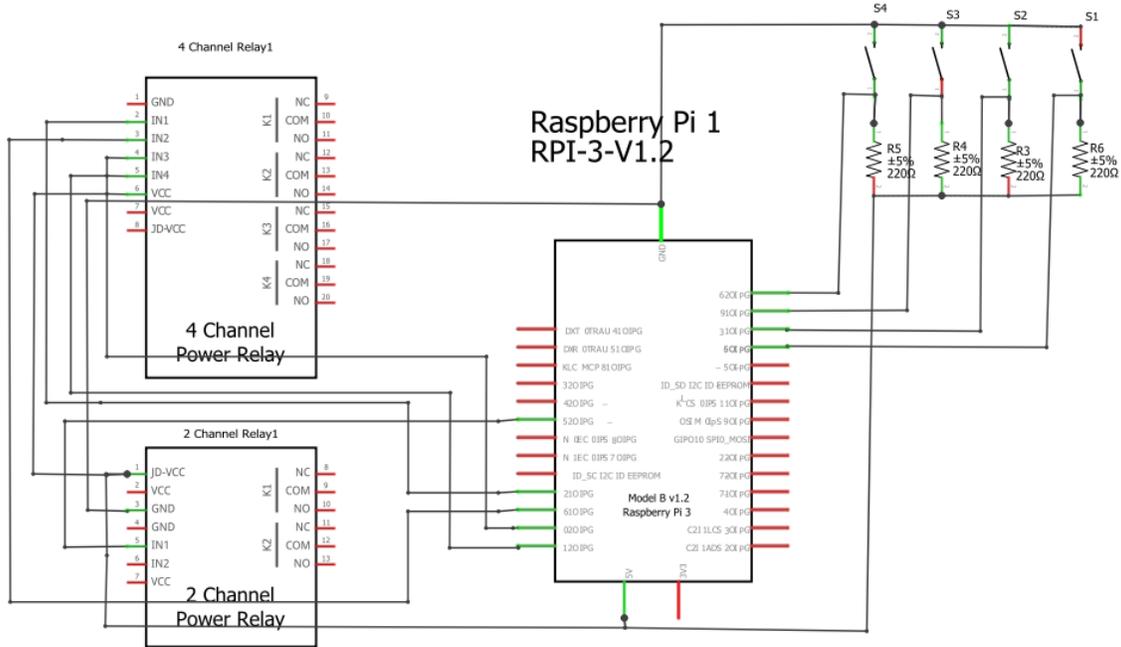


Figura 88. Elementos y diagrama eléctrico IEDRaspberry.

El IEDRaspberry consiste en una placa Raspberry pi 3B, como se muestra en la sección 1.2.14 de este documento, se utilizan 4 entradas digitales en sus pines GPIO para la lectura

de estado y su publicación vía MMS y mensaje GOOSE. Estos están configurados como Indicadores de un nodo lógico de tipo uso general o GGIO, y están conectados a los pines GPIO 22 – 25.

Incluye también 5 salidas de pines GPIO (26-29,6) conectados a un shield de relays con leds, esto con el fin de indicar la actuación de alguna de las entradas que monitorean protecciones. La figura 88 muestra los componentes y el diagrama eléctrico del IEDRASPBerry.

2.2.1.2 Sistema operativo

Se utiliza raspbian como sistema operativo, sin embargo, podría utilizarse cualquier sistema operativo compatible con la placa Raspberry Pi, y que exista el grupo de paquetes para desarrollo bajo C++. Para la descarga e instalación se siguen los pasos de la página oficial de Raspberry: <https://www.raspberrypi.org/downloads/>, y que han sido incluidos en el anexo D de este documento.

La instalación del ambiente de desarrollo C++ en el sistema se realiza iniciando una terminal bash y ejecutando los siguientes comandos:

```
$apt-get -y update
$apt-get -y install build-essential
```

Adicionalmente es necesario la instalación de la librería libiec61850, los pasos de instalación se encuentran en la sección 1.2.10. La librería WiringPi está instalada por defecto en Raspbian, por lo que solo es necesaria su instalación en caso se utilicé otro S.O.

2.2.1.3 Aplicación para servidor IEC61850

El IEDRaspberry ejecuta una aplicación escrita en lenguaje C que permite el montaje de un servidor virtual bajo el estándar IEC61850, dicho servidor se observa en la red como un IED y mediante las funciones de la librería se logra la lectura de comandos, el envío de datos vía MMS, suscripción GOOSE y demás funciones que ejecute el IED. Para el desarrollo de la aplicación se utilizan 2 librerías, libiec61850 para la comunicación bajo protocolos del estándar IEC61850 y WiringPi para el control de entradas/salidas de los pines GPIO de la raspberry pi, la instalación, configuración y compilación de dichas librerías se realiza tal como se muestra en la sección 1.2.10 y sección 1.2.12 respectivamente

En el anexo B se encuentra el código principal comentado donde se encuentra la programación del comportamiento del IEDRaspberry. Dicha aplicación se segmenta en 5 tareas principales:

1) Creación de servidor IEC61850

Consiste en el código básico de la librería, el cual se muestra en el anexo A en el cual a partir del modelo en C en los archivos “static_model” se carga en el código

ejecutándose un servidor IEC61850 el cual pueden visualizarse mediante un cliente como el IEDExplorer, en la figura 89 se muestra el modelo de Datos de la IEDRaspberry que se ha configurado en el archivo CID.

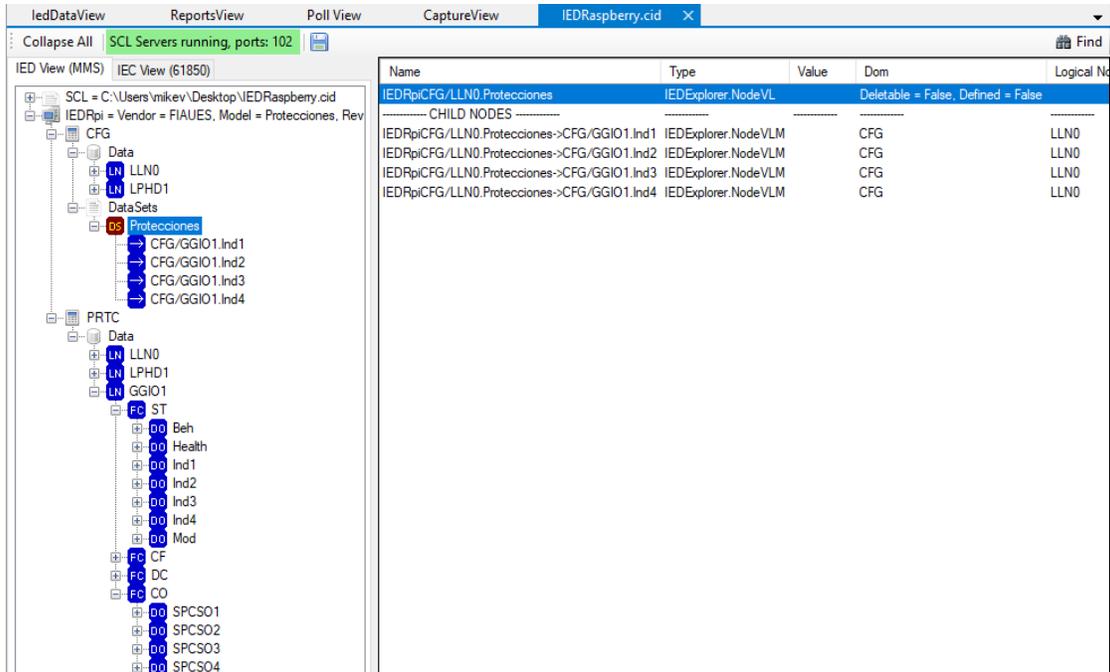


Figura 89. Modelo de Datos del IEDRaspberry.

Como se ha descrito en la sección 1.1 para todo IED bajo el estándar es necesario crear un archivo ICD el cual contendrá el modelo de datos de la IEDRaspberry. Para propósitos de este documento se utiliza el software ICD Editor (ver sección 1.2.1) con el nombre *IEDRpi*, se agrega dos dispositivos lógicos *CFG* el cual contiene la configuración de DataSet y Mensajes GOOSE y *PRTC* el cual contiene el nodo lógico *GGIO* que simula entradas de protecciones de un transformador y salidas de estados de protecciones, tal como se muestra a continuación.

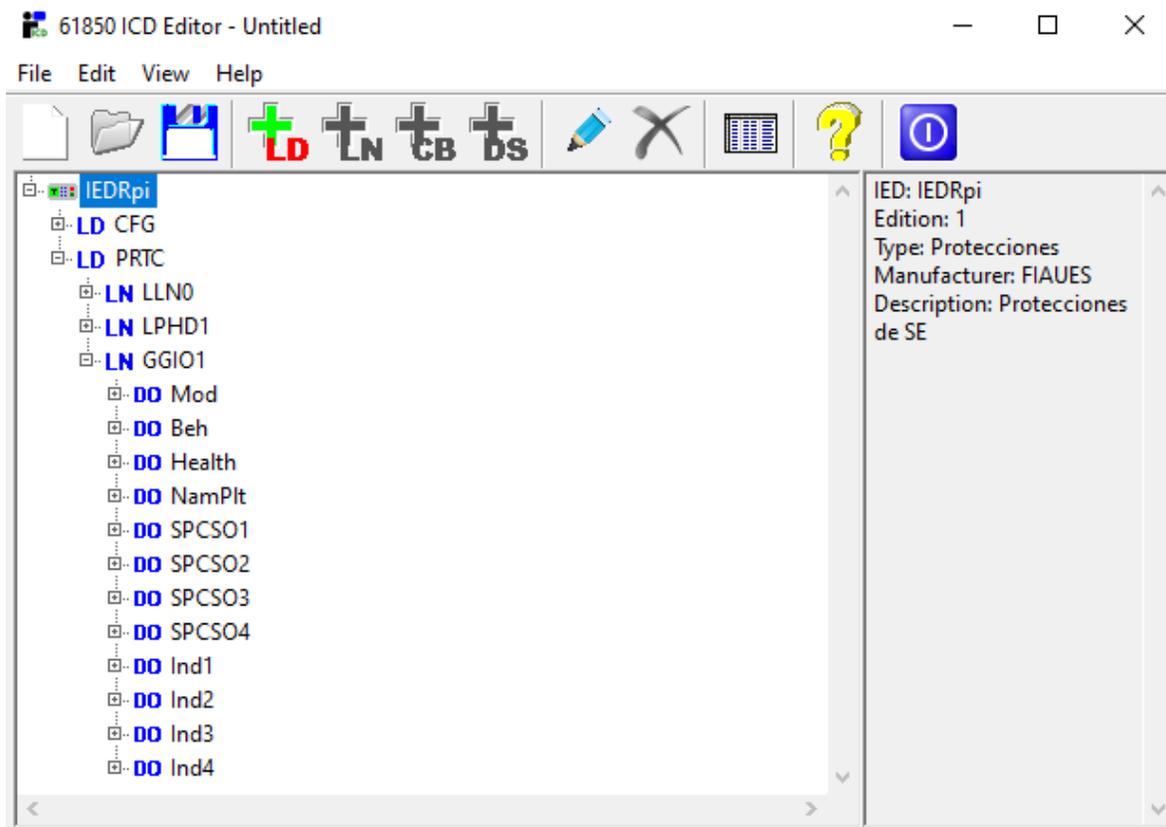


Figura 90. Configuración de ICD IEDRaspberry.

Como se muestra en la figura 90 el nodo lógico GGIO contiene objetos de datos obligatorios y opcionales los cuales son SPCS el cual describe las salidas de la Raspberry Pi y el DO Ind los cuales son indicadores que simulan las protecciones del transformador.

Para configurar los mensajes GOOSE de las protecciones del transformador es necesario crear un conjunto de datos (DS) que contendrá la información de los atributos de datos que es necesario mapear, para ello se utiliza el nodo lógico cero del LD CFG y se creó un DS (ver apartado 1.2.1) con nombre protecciones, en este DataSet se agrega la información de estados de los indicadores tal como se muestra en la figura 91. También se utiliza un bloque de control de tipo GOOSE (ver apartado 1.2.1) con el nombre PTRX, tal como se muestra en la figura 92.

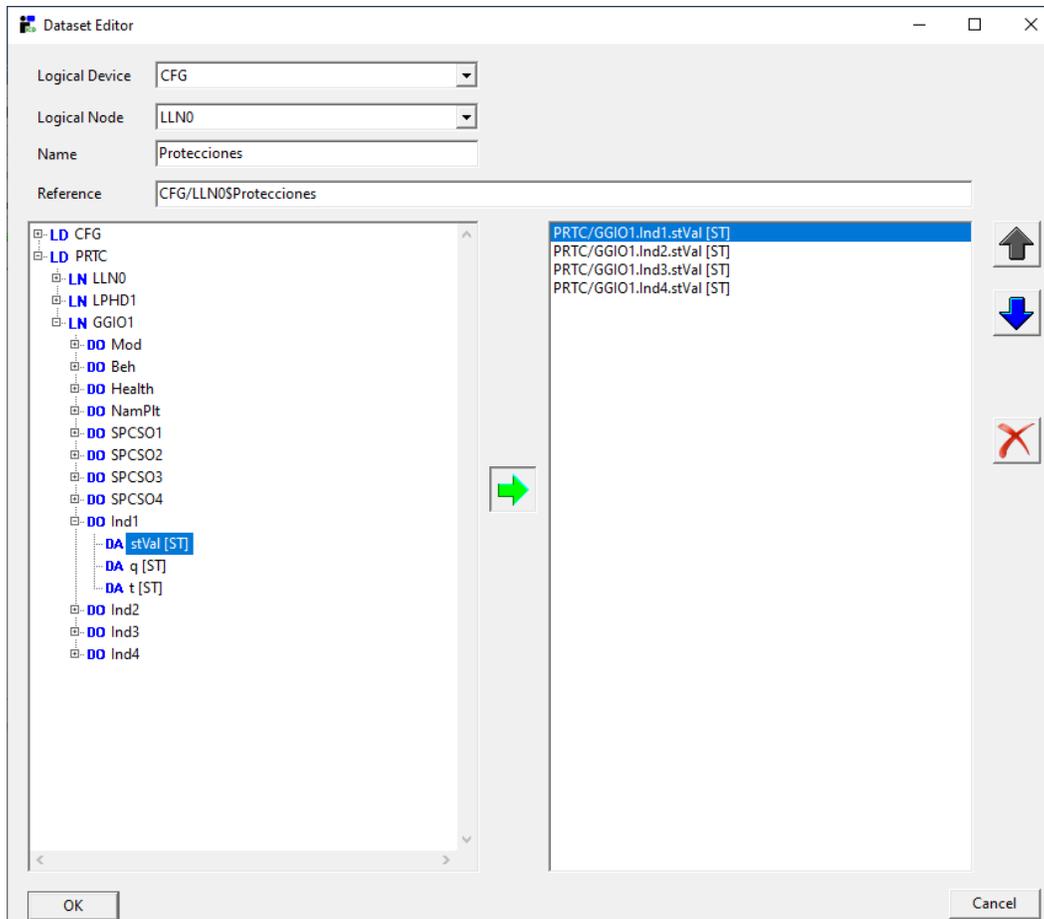


Figura 91. Configuración de DataSet del ICD IEDRaspberry.

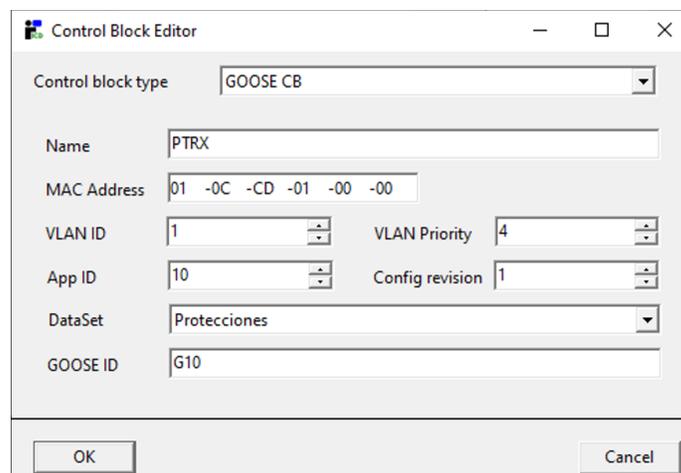


Figura 92. Configuración de mensajes GOOSE en el ICD IEDRaspberry.

La figura 93 muestra la configuración del mensaje GOOSE, para el cual se utiliza el DS (Protecciones), y se asigna una MAC de tipo Multicast, el App ID y el GOOSE

ID del mensaje es 10 y G10 respectivamente, también se designa una prioridad de 4 (muy alta) en el mensaje GOOSE. Luego de realizar la configuración de parámetros se hace clic en OK y de esta manera se creará un DS y un mensaje GOOSE en el ICD tal como se muestra en la Figura 92. Finalmente se guarda el archivo bajo la extensión CID.

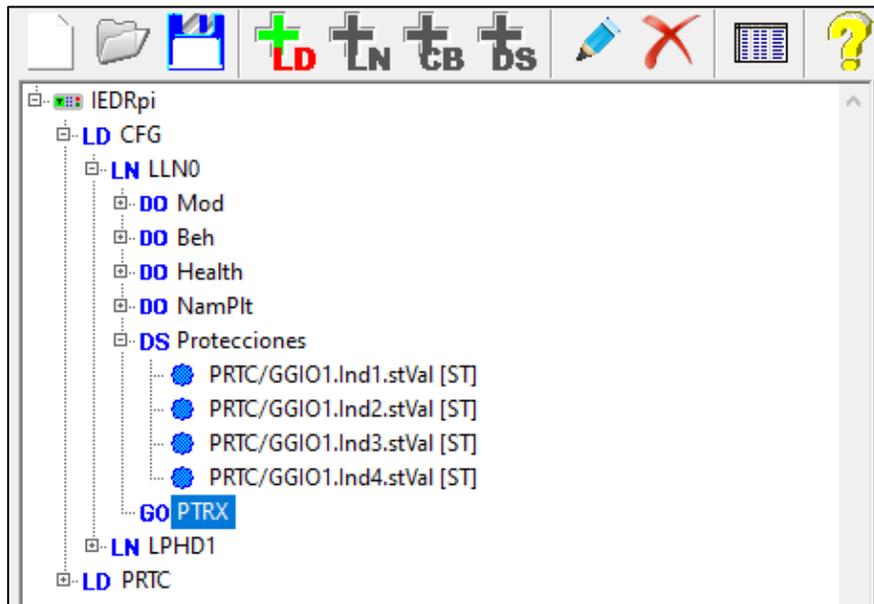


Figura 93. ICD IEDRaspberry Pi.

2) **Monitoreo de entradas digitales de pines GPIO de la Raspberry.** Mediante la librería WiringPi se obtiene la lectura de los pines GPIO, en este caso se utiliza los pines 22 - 25 para el escenario 1 del capítulo 3 donde se monitorean las protecciones mecánicas de un transformador de potencia. La configuración de la librería y lectura/escritura de la librería se realiza tal como se muestra en la sección 1.2.12. La tabla 4 muestra los tags correspondientes a las entradas monitoreadas y su correspondiente indicador.

PIN GPIO	LD	LN	Tags	Descripción
22	PRTC	GGIO	ST\$Ind1\$stVal	Indicador 1
23	PRTC	GGIO	ST\$Ind2\$stVal	Indicador 2
24	PRTC	GGIO	ST\$Ind3\$stVal	Indicador 3
25	PRTC	GGIO	ST\$Ind4\$stVal	Indicador 4

Tabla 4. Tags entradas digitales IEDRASPBERRY.

Los pines utilizados para salidas se muestran en la tabla 5 y son utilizados para indicar físicamente el estado en que se encuentran las entradas mediante un LED y una salida de relé para conectar alguna alerta sonora o visual. Se configura también una salida

para monitorear el mensaje GOOSE que envía el interruptor de INFOTECH en la red, se utiliza el pin 9 y posee indicador LED y salida de relé.

PIN GPIO	LD	LN	Tags	Descripción
26	PRTC	GGIO	CO\$SPCSO1\$ctVal	Relé 1
27	PRTC	GGIO	CO\$SPCSO2\$ctVal	Relé 2
28	PRTC	GGIO	CO\$SPCSO3\$ctVal	Relé 3
29	PRTC	GGIO	CO\$SPCSO4\$ctVal	Relé 4

Tabla 5. Salidas digitales IEDRASPBERRY.

3) **Suscripción a mensaje GOOSE para monitoreo de estado de interruptor virtual INFOTECH.** El IEDRASPBERRY tiene la función de monitorear el estado del interruptor virtual de INFOTECH mediante la suscripción al mensaje GOOSE el cual se ha habilitado en el apartado 2.1.1, la configuración de la suscripción GOOSE se realiza utilizando funciones de la librería libiec61850 tal como se muestra en la sección 1.2.10 con los siguientes parámetros:

- Adaptador de red: Ethernet (eth0)
- Tipo de publicación: Multicast
- AppID:1
- CRef: "DemoProtCtrl/LLN0\$GO\$gcb1"
- GOOSEID: "G1"
- DataSet: "DemoProtCtrl/LLN0\$DS3_GOOSE"
- Tipo de dato: Booleano
- VLANID: 0
- Prioridad: 4

4) **Publicación de mensajes GOOSE con parámetros actualizados del IED.** Los mensajes GOOSE son configurados como se muestra en la sección 1.2.10 y para el funcionamiento de los escenarios del capítulo 3 se utilizan los siguientes mensajes GOOSE y sus parámetros:

Estado de las entradas digitales de 4 pines GPIO, para simulación de relés de protección del transformador, parámetros:

- Adaptador de red: Ethernet (eth0)
- Tipo de publicación: Multicast
- AppID:5
- CRef: "simpleIOGenericIO/LLN0\$GO\$IndGoose"
- GOOSEID: "simpleIOGenericIO"
- DataSet: "simpleIOGenericIO/LLN0\$DS1"
- Tipo de dato: Booleano

- VLANID: 0
- Prioridad: 4

Disparo de interruptores virtuales de AxonGroup en caso de falla en la barra de escenario de capítulo 3, parámetros:

- Adaptador de red: Ethernet (eth0)
- Tipo de publicación: Multicast
- AppID: 1000=4096 hexadecimal
- CRef: "REMOTEProtections/LLN0\$GO\$gcb1"
- GOOSEID: "RemoteProtection"
- DataSet: "REMOTEProtections/LLN0\$DS1_GOOSE"
- Tipo de dato: Booleano
- VLANID: 0
- Prioridad: 4

Para la compilación de la aplicación se utiliza un archivo Makefile con todas las instrucciones que van desde el llamado a los archivos con el archivo CID convertido a código C, hasta aplicaciones en java para generar el modelo del IED a crearse como servidor, se tomó de base el archivo Makefile del ejemplo basic_io de la librería, con el agregado de incluir la librería wiringPi al inicio y denominarla EXTRA_LIBS en el archivo Makefile.

Para compilar ejecutar el comando: “make” en una terminal bash ubicada en la carpeta principal de la aplicación, a continuación, se presenta el contenido del archivo makefile:

```
EXTRA_LIBS=-lwiringPi -lm
LIBIEC_HOME=../..
PROJECT_BINARY_NAME = server_example_simple
PROJECT_SOURCES = server_example_simple.c
PROJECT_SOURCES += static_model.c

PROJECT_ICD_FILE = sampleModel_with_dataset.cid

include $(LIBIEC_HOME)/make/target_system.mk
include $(LIBIEC_HOME)/make/stack_includes.mk

all: $(PROJECT_BINARY_NAME)
include $(LIBIEC_HOME)/make/common_targets.mk

model: $(PROJECT_ICD_FILE)
    java -jar $(LIBIEC_HOME)/tools/model_generator/genmodel.jar
$(PROJECT_ICD_FILE)
$(PROJECT_BINARY_NAME): $(PROJECT_SOURCES) $(LIB_NAME)
    $(CC) $(CFLAGS) $(LDFLAGS) -o $(PROJECT_BINARY_NAME)
$(PROJECT_SOURCES) $(INCLUDES) $(LIB_NAME) $(LDLIBS) $(EXTRA_LIBS)
clean:
    rm -f $(PROJECT_BINARY_NAME)
```

2.2.2 Gateway IEC61850/Modbus

Debido a que el relé de protección SEL 351 con el que se cuenta no posee el estándar IEC61850, se creó un Gateway virtual IEC61850 a modbus el cual nos permite crear un servidor IEC61850 que emula un IED el cual contiene varias funciones básicas del relé SEL351 que se utiliza en los escenarios del capítulo 3. Dicho Gateway funciona bajo el OS Xubuntu en una máquina virtual, con adaptador de red de tipo puente configurado. La figura 94 muestra un esquema del proceso que realiza el servidor IEC61850.

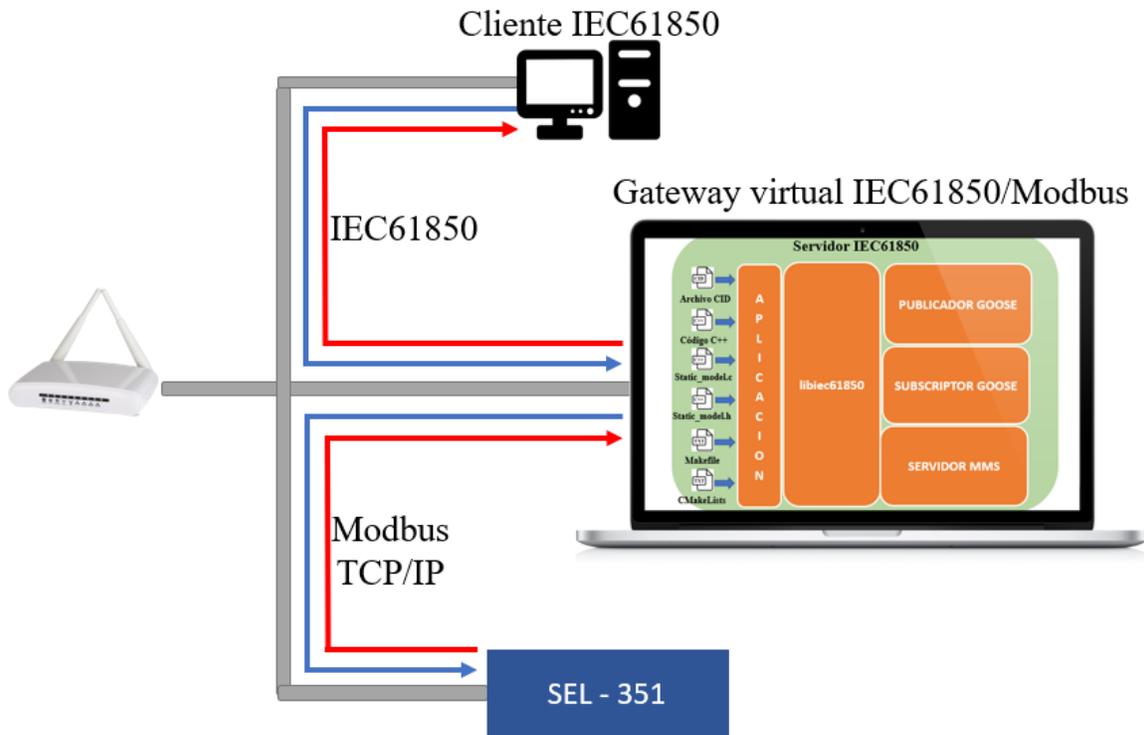


Figura 94. Esquema de Configuración SEL351 bajo el estándar IEC61850.

2.2.2.1 Interfaz física

Para los propósitos de este documento se utiliza un relé de protección de la marca SEL, modelo 351 tal como se muestra información en la sección 1.2.13. Dicho relé es utilizado para escenarios de pruebas en el capítulo 3 en donde actúa ante comandos de disparos enviados desde otro IED. Este relé cuenta con el protocolo Modbus TCP/IP para recibir comandos de disparos, es en esto donde funciona el Gateway IEC61850/Modbus, cuya función es enviar vía modbus los disparos recibidos en su servidor en mensajes GOOSE. El relé SEL351 cuenta con 5 puertos de comunicación, de estos el puerto 5 es ethernet y es el que se utiliza en este documento. La tabla 6 se muestra la configuración guardada para el funcionamiento del relé con el Gateway.

Puerto 5		
Nombre	Descripción	Valor
EPORT	Enable Port	Y
IPADDR	Device IP Address (zzz.yyy.xxx.www)	192.168.1.80
SUBNETM	Subnet Mask (zzz.yyy.xxx.www)	255.255.255.0
DEFRTR	Default Router (zzz.yyy.xxx.www)	192.168.1.1
ETCPKA	Enable TCP Keep-Alive	Y
KAIDLE	TCP Keep-Alive Idle Range (seconds)	10
KAINTV	TCP Keep-Alive Interval Range (seconds)	10
KACNT	TCP Keep-Alive Count Range	5
NETMODE	Operating Mode	FAILOVER
FTIME	Failover Time-out (seconds)	1.00
NETPORT	Primary Net Port	B
ETELNET	Enable Telnet	Y
MAXACC	Maximum Access Level	C
TPORT	Telnet Port	23
AUTO	Send Auto Messages to Port	N
FASTOP	Fast Operate Enable	Y
EFTPSERV	Enable FTP	Y
FTPUSER	FTP User Name	FTPUSER
FTPCBAN	FTP Connect Banner(254 char; NA to NULL)	FTP SERVER
FTPIDLE	FTP Idle Timeout (min)	5
EHTTP	Enable HTTP Server	Y
HTTPACC	HTTP Maximum Access Level	2
HTTPPORT	TCP/IP Port	80
HTTPIDLE	HTTP Web Server Timeout (min)	10
FWFPC	Firmware Upgrade Front Panel Confirmation	Y
EDNP	Enable DNP Sessions	0
ESNTP	Enable SNTP Client	OFF
EMODBUS	Enable Modbus	1
MODIP1	IP Address (zzz.yyy.xxx.www)	192.168.1.60
MTIMEO1	Modbus Session Time-out (seconds)	15

Tabla 6. Configuración puerto 5 relé SEL 351.

2.2.2.2 Sistema operativo

Para la ejecución del Gateway IEC61850 es necesario una PC con S.O GNU/Linux, para los propósitos de este documento se utiliza la distribución Xubuntu, debido a su poco consumo de recursos y su interfaz gráfica agradable al usuario, se instaló el S.O en una máquina virtual, esto con el fin de utilizar en una misma PC softwares simuladores de dispositivos virtuales que funcionan bajo el S.O Windows. Se instala en la distro GNU/Linux el ambiente de desarrollo para el lenguaje C, esto se logra mediante las siguientes instrucciones en una consola bash:

```
$sudo apt-get -y update
$sudo apt-get -y install build-essential
```

Adicionalmente para el correcto funcionamiento del Gateway es necesario la instalación de las librerías libiec61850 y libmodbus, la instalación se muestra en las secciones 1.2.10 y 1.2.11 de este documento.

2.2.2.3 Aplicación para creación de servidor IEC61850

La aplicación que realiza función de comunicación entre protocolos utiliza dos librerías, libiec61850 y libmodbus (ver secciones 1.2.10 y 1.2.12). Se crea un servidor IEC61850 el cual posee algunas de las funciones del relé de protección SEL, las cuales se utilizan en los escenarios del capítulo 3 las salidas de relé han sido configuradas como objetos GGIO SPCS (Single point control) y pueden ser activadas mediante mensajes de disparo de tipo GOOSE, también ha sido configurado para suscribirse a los mensajes GOOSE que envíe el IEDRASPERRY con los estados de sus entradas para la toma de decisión de disparos en caso exista una falla.

En el anexo C se encuentra el código principal comentado de la aplicación en lenguaje C, donde se encuentra el llamado de funciones y la toma de decisiones configuradas. El código se segmenta en 5 tareas principales:

- 1) **Creación de servidor IEC61850.** Consiste en el código básico de la librería, el cual se muestra en el anexo A y que a partir del modelo del IED dentro del archivo ICD se carga dicho modelo como un servidor IEC61850 emulando en la red la existencia del IED, el cual puede visualizarse mediante un cliente como IEDEXplorer, en la figura 99 se muestra el modelo de Datos de la IEDSEL351 que se ha configurado en el archivo CID.

Name	Type	Value	Dom	Logical Node	Var Path
----- CHILD NODES -----					
CFG	IEDExplorer.NodeLD		CFG		
IEDSEL351CFG/LLN0.Beh.stVal	Enum (Integer)		CFG	LLN0	\$Beh\$stVal
IEDSEL351CFG/LLN0.Beh.q	Quality		CFG	LLN0	\$Beh\$q
IEDSEL351CFG/LLN0.Beh.t	Timestamp		CFG	LLN0	\$Beh\$t
IEDSEL351CFG/LLN0.Health.stVal	Enum (Integer)		CFG	LLN0	\$Health\$stVal
IEDSEL351CFG/LLN0.Health.q	Quality		CFG	LLN0	\$Health\$q
IEDSEL351CFG/LLN0.Health.t	Timestamp		CFG	LLN0	\$Health\$t
IEDSEL351CFG/LLN0.Mod.stVal	Enum (Integer)		CFG	LLN0	\$Mod\$stVal
IEDSEL351CFG/LLN0.Mod.q	Quality		CFG	LLN0	\$Mod\$q
IEDSEL351CFG/LLN0.Mod.t	Timestamp		CFG	LLN0	\$Mod\$t
IEDSEL351CFG/LLN0.Mod.ctilModel	Enum (Integer)		CFG	LLN0	\$Mod\$ctilModel
IEDSEL351CFG/LLN0.NamPtt.vendor	VisString255		CFG	LLN0	\$NamPtt\$vendor
IEDSEL351CFG/LLN0.NamPtt.swRev	VisString255		CFG	LLN0	\$NamPtt\$swRev
IEDSEL351CFG/LLN0.NamPtt.d	VisString255		CFG	LLN0	\$NamPtt\$d
IEDSEL351CFG/LLN0.NamPtt.IdNs	VisString255	IEC 61850-7-4:2003	CFG	LLN0	\$NamPtt\$IdNs
IEDSEL351CFG/LPHD1.PhyHealth.stVal	Enum (Integer)		CFG	LPHD1	\$PhyHealth\$stVal
IEDSEL351CFG/LPHD1.PhyHealth.q	Quality		CFG	LPHD1	\$PhyHealth\$q
IEDSEL351CFG/LPHD1.PhyHealth.t	Timestamp		CFG	LPHD1	\$PhyHealth\$t
IEDSEL351CFG/LPHD1.Proxy.stVal	BOOLEAN		CFG	LPHD1	\$Proxy\$stVal
IEDSEL351CFG/LPHD1.Proxy.q	Quality		CFG	LPHD1	\$Proxy\$q
IEDSEL351CFG/LPHD1.Proxy.t	Timestamp		CFG	LPHD1	\$Proxy\$t
IEDSEL351CFG/LPHD1.PhyNam.vendor	VisString255		CFG	LPHD1	\$PhyNam\$vendor

Figura 95. Modelo de Datos del Relé SEL351.

Como se ha descrito en la sección 1.2.1 para que el relé SEL351 ejecute funciones bajo el estándar IEC61850 es necesario crear un ICD, para ello creamos un IED en el software ICD Editor (ver sección 1.2.5) el cual se llama *IEDSEL351*, el IED tendrá en su modelo de datos dos dispositivos lógicos los cuales son *CFG* el cual contiene la configuración de Dataset y Mensajes GOOSE del Relé SEL351 y *PTRC* el cual posee un nodo lógico GGIO con 7 objetos de datos SPCSO que representan las salidas del relé y 6 indicadores que representas las entradas, tal como se muestra a continuación.

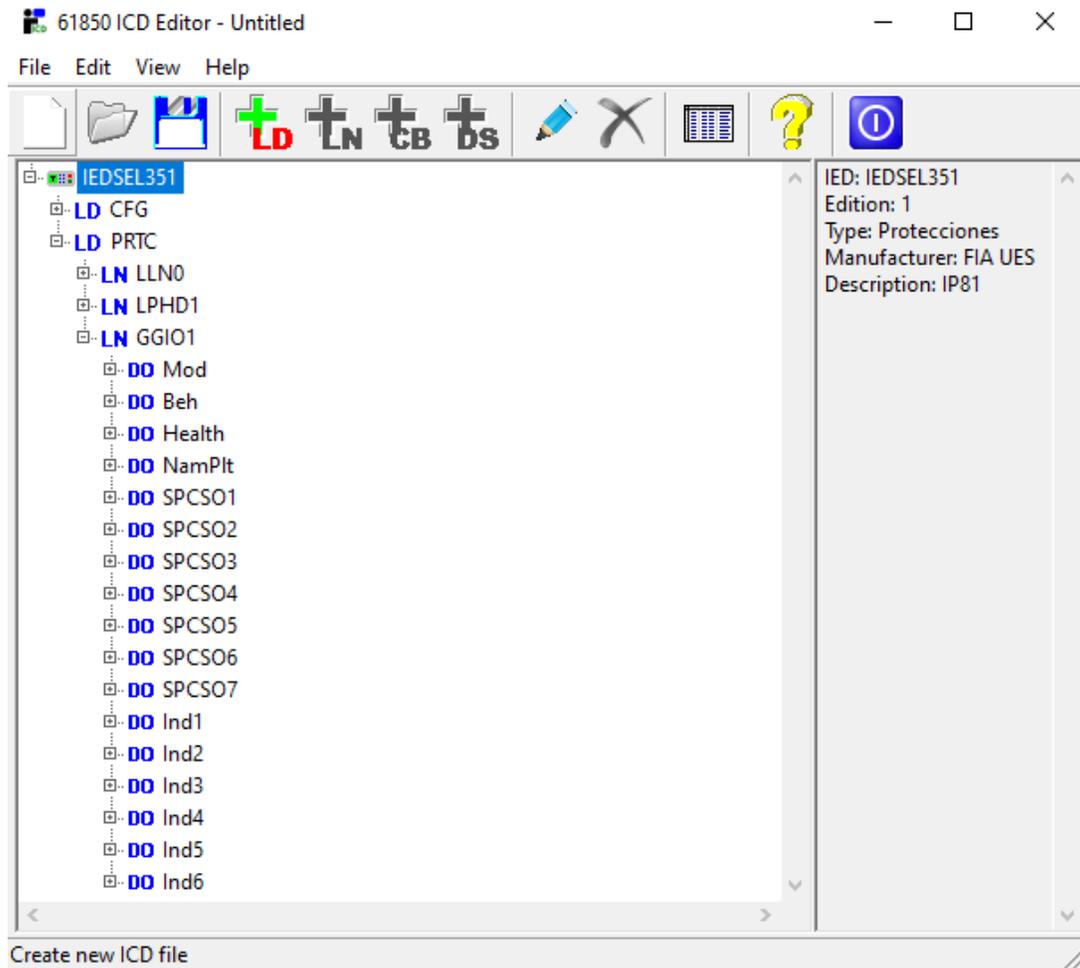


Figura 96. Configuración ICD SEL351.

Para el envío de mensajes GOOSE del relé Sel351 se configura un Dataset (ver apartado 1.2.1) que contenga el estado de los indicadores (entradas) al cual llamaremos *In351*, luego se configuran los mensajes GOOSE. La figura 95 muestra la configuración del Dataset.

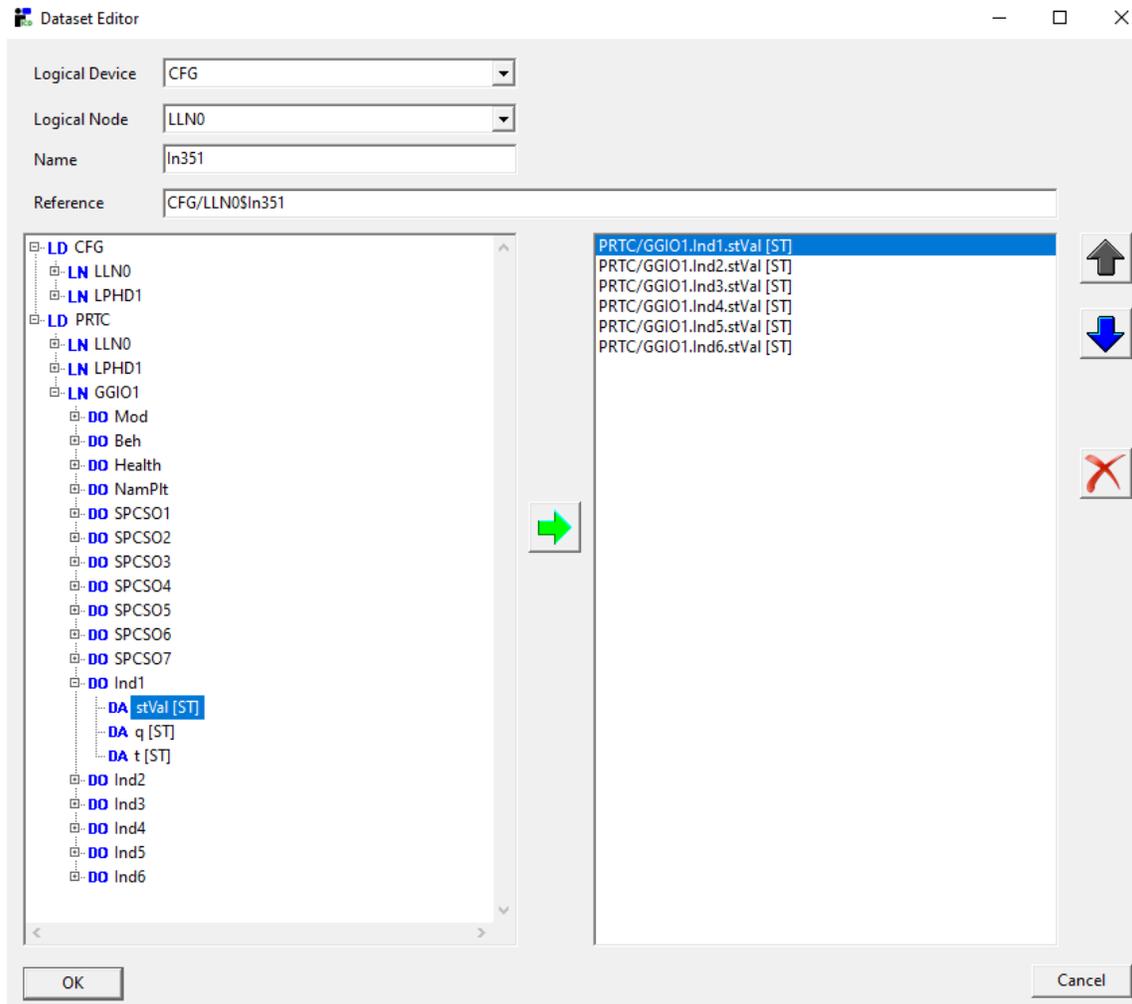


Figura 97. Configuración de Dataset del relé SEL351.

Una vez que se ha creado el Dataset In351 se configura un bloque de control de tipo GOOSE con nombre *InputSEL*, tal como se muestra en la figura 97.

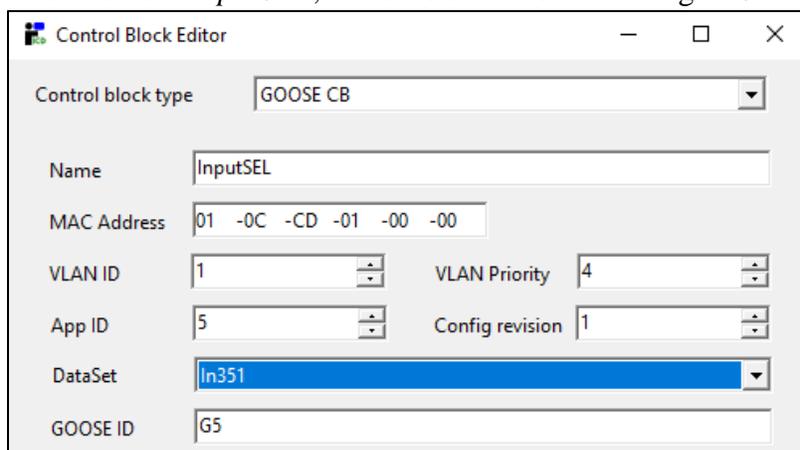


Figura 98. Configuración de mensaje GOOSE del Relé Sel351.

La figura 97 muestra los parámetros que tiene el mensaje GOOSE el cual se configura con el DS que se crea previamente, se selecciona un MAC de tipo multicast y App ID 5, se coloca una prioridad de 4 y un GOOSE ID con nombre G5. Una vez configurado el mensaje GOOSE se guarda el archivo con extensión CID. La figura 98 muestra el archivo ICD del Relé SEL351.

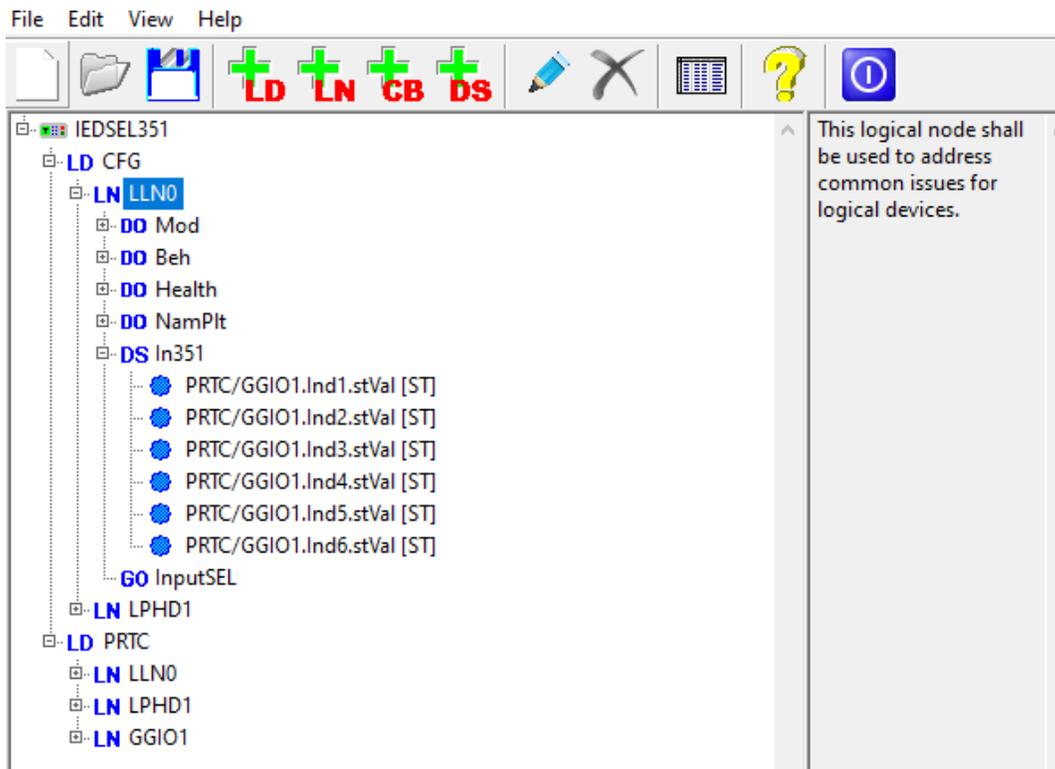


Figura 99. ICD del Relé Sel351.

- 2) **Respuesta a comandos MMS desde clientes.** La configuración para recepción de comandos vía MMS se realiza tal como se muestra en la sección 1.2.12. A continuación se muestra el comando MMS que es utilizado en uno de los escenarios del capítulo 3.

```
LLN0/GGIO/SPCS01/ctlValue
```

El comando anterior es utilizado para reestablecer el relé SEL a su estado normal después de que se detecte una falla a través de las entradas del IEDRASPBERRY y que hayan sido normalizadas las condiciones.

- 3) **Suscripción a mensajes GOOSE.** El relé SEL monitorea mediante mensajes GOOSE las entradas del IEDRASPBERRY, esto debido a que representan la actuación de protecciones por lo que debe ser disparados también los interruptores

asociados al relé SEL, la configuración se realizó tal como se mostró en la sección 1.2.10 con los siguientes parámetros:

- Adaptador de red: Ethernet (eth0)
- Tipo de publicación: Multicast
- AppID:5
- CRef: “simpleIOGenericIO/LLN0\$GO\$IndGoose”
- GOOSEID: “simpleIOGenericIO”
- DataSet: “simpleIOGenericIO/LLN0\$DS1”
- Tipo de dato: Booleano
- VLANID: 0
- Prioridad: 4

4) Publicación de mensajes GOOSE. Como parte de las funciones del relé programadas para funcionar en los escenarios del capítulo 3, debe hacer el disparo del interruptor conectado a una de sus salidas y el disparo vía mensaje GOOSE del interruptor virtual, para ello se configura la publicación de mensajes GOOSE tal como se mostró en la sección 1.2.10 y con los siguientes parámetros:

- Adaptador de red: Ethernet (enp0s3)
- Tipo de publicación: Unicast
- MAC de destino: 01:0C:CD:01:00:02
- AppID:3
- CRef: “LLN0\$gcb1”
- GOOSEID: “G3”
- DataSet: “LLN0\$DS1”
- Tipo de dato: Booleano
- VLANID: 0
- Prioridad: 4

5) Escritura de registros Modbus del Relé SEL- 351. Para él envío de los comandos de activación de las salidas del relé se utilizó la función modbus_write_bit (), y se configura tal como se mostró en la sección 1.2.11 utilizando los registros Modbus mostrados en la figura 100 obtenidos del manual del relé y que funcionan para activar las salidas del relé durante 1 segundo.

Coil Address in Decimal	Coil Address in Hex	Function Code Supported	Coil Description	Coil Function	Duration
0	0	1,5	OUT101 ^a	Pulse	1 second
1	1	1,5	OUT102 ^a	Pulse	1 second
2	2	1,5	OUT103 ^a	Pulse	1 second
3	3	1,5	OUT104 ^a	Pulse	1 second
4	4	1,5	OUT105 ^a	Pulse	1 second
5	5	1,5	OUT106 ^a	Pulse	1 second
6	6	1,5	OUT107 ^a	Pulse	1 second
7	7	1,5	ALRMOUT ^b	Pulse	1 second

Figura 100. Listado de registro Modbus de salidas del Relé SEL351.

Otros parámetros utilizados para la comunicación son los siguientes:

- Dirección IP: 192.168.1.80
- Puerto: 502
- Función modbus: 05h
- No de registro modbus: 0

Para la compilación del código se utiliza un archivo Makefile con todas las instrucciones que van desde el llamado a los archivos con el archivo CID convertido a código C, hasta aplicaciones en java para generar el modelo del IED a crearse como servidor, se tomó de base el archivo Makefile del ejemplo server_example_control de la librería, con el agregado de incluir la librería libmodbus al inicio y denominarla LIBPQ-LIBS en el archivo Makefile. Para compilar se ejecuta el comando: make en una terminal bash ubicada en la carpeta del programa principal.

```
LIBPQ-INCLUDE=`pkg-config --cflags libmodbus`
LIBPQ-LIBS=`pkg-config --libs libmodbus`
LIBIEC_HOME=../..
PROJECT_BINARY_NAME = server_example_control
PROJECT_SOURCES = server_example_control.c
PROJECT_SOURCES += static_model.c
PROJECT_ICD_FILE = simpleIO_control_tests.cid
include $(LIBIEC_HOME)/make/target_system.mk
include $(LIBIEC_HOME)/make/stack_includes.mk
all: $(PROJECT_BINARY_NAME)
include $(LIBIEC_HOME)/make/common_targets.mk
model: $(PROJECT_ICD_FILE)
    java -jar $(LIBIEC_HOME)/tools/model_generator/genmodel.jar
$(PROJECT_ICD_FILE)
$(PROJECT_BINARY_NAME): $(PROJECT_SOURCES) $(LIB_NAME)
    $(CC) $(CFLAGS) $(LDFLAGS) -o $(PROJECT_BINARY_NAME)
$(PROJECT_SOURCES) $(INCLUDES) $(LIB_NAME) $(LDLIBS) $(LIBPQ-
INCLUDE) $(LIBPQ-LIBS)
clean:
    rm -f $(PROJECT_BINARY_NAME)
```

2.3 CONFIGURACIÓN DEL POWERELIPSE SCADA

El modelo eléctrico que se utiliza en el SCADA es el que se muestra en la figura 101, el cual está configurado con los tags de los dispositivos simulados y emulados, para ello se debe importar el modelo eléctrico hacia la pantalla principal de control y supervisión del SCADA (ver apartado 1.2.8)

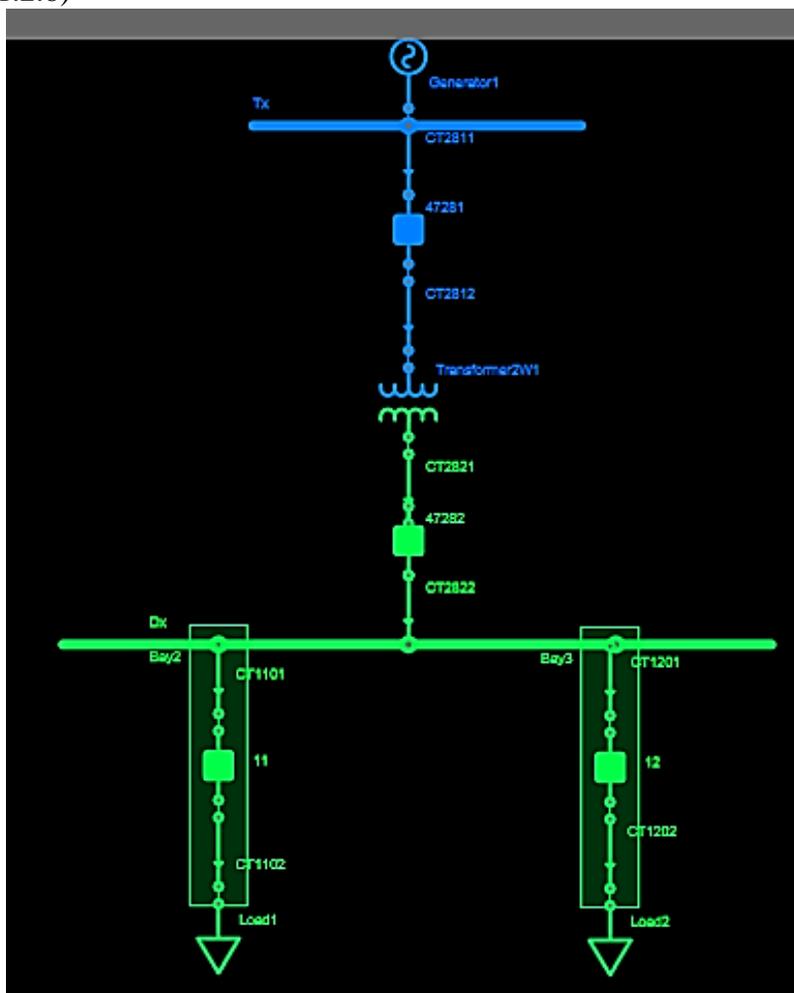


Figura 101. Modelo Eléctrico de un SAS.

El esquema presentado en la figura 101, representa una subestación de Distribución de barra simple el cual está compuesto por dos bahías, un transformador trifásico conexión delta-estrella aterrizada 46/23 kV, interruptores de potencias y cuchillas de entrada y salida.

La comunicación del SCADA con los IEDs simulados/emulados se realiza a través de un servidor OPC, en este proyecto se utiliza el software Relab OPC el cual permite comunicar los IEDs con IEC61850.

En el Relab se crean 5 Drivers IEC61850 (Para la configuración de un Driver, ver apartado 1.2.2) que estarán comunicándose con cada uno de los IEDs presentes en las pruebas, los cuales se muestran en la Figura 102.

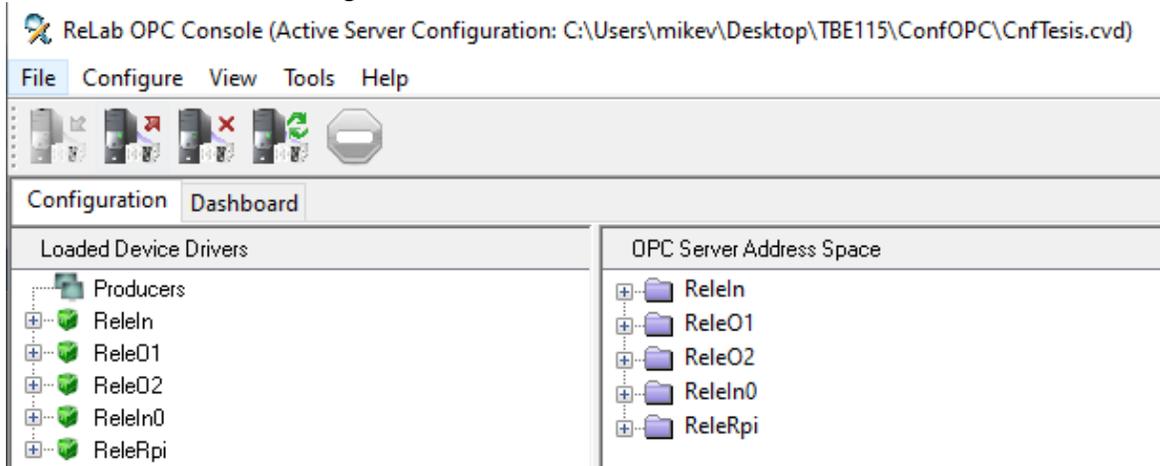


Figura 102. Drivers IEC61850 para la comunicación de IEDs simulados/emulados.

La figura 102 muestra cinco drivers IEC61850 los cuales son:

- ReleIn: Establece la comunicación con el Relé de INFOTECH
- ReleO1: Establece la comunicación con el primer Relé Simulator Bay de AxonGroup.
- ReleO2: Establece la comunicación con el segundo Relé Simulator Bay de AxonGroup.
- ReleIn0: Establece la comunicación con el Relé SEL351.
- ReleRpi: Establece la comunicación con el Relé RaspberryPi.

Cada uno de estos Driver posee el modelo de datos de los IEDs simulados y emulados, para propósito de este documento únicamente se mapean aquellos Tags que sean de interés en la supervisión y control del SCADA, los cuales se presentan en la Tabla 7.

Driver	LD	LN	Tags	Descripción
ReleIn	DemoMeasurement	I3pMMXU1	MX\$A\$phsA\$cVal\$mag\$f	Corriente Fase A
ReleIn	DemoMeasurement	I3pMMXU1	MX\$A\$phsB\$cVal\$mag\$f	Corriente Fase B
ReleIn	DemoMeasurement	I3pMMXU1	MX\$A\$phsC\$cVal\$mag\$f	Corriente Fase C
ReleIn	DemoProtCtrl	Obj1XCBR1	ST\$Pos\$stVal	Posición del Interruptor 1
ReleIn	DemoProtCtrl	I3GtPTRC1	ST\$Tr\$general	Triger del Interruptor 1
ReleIn0	simpleOGenericIO	GGIO1	CO\$SPCSO1\$Oper\$ctVal	Control de Disparo para Interruptor SEL351

ReleIn0	simpleOGenericIO	GGIO1	ST\$SPCSO1\$stVal	Posición del Interruptor SEL351
ReleO1	AXONBAYMeasurement	MMXU1	MX\$A\$phsA\$cVal\$mag\$f	Corriente Fase A
ReleO1	AXONBAYMeasurement	MMXU1	MX\$A\$phsB\$cVal\$mag\$f	Corriente Fase B
ReleO1	AXONBAYMeasurement	MMXU1	MX\$A\$phsC\$cVal\$mag\$f	Corriente Fase C
ReleO1	AXONBAYProtections	PTOC1	ST\$Str\$general	Triger del Interruptor 1 activado por Sobre corriente
ReleO1	AXONBAYControl	INTA_XCBR1	ST\$Pos\$stVal	Posición del Interruptor 1
ReleO2	AXONBAYMeasurement	MMXU1	MX\$A\$phsA\$cVal\$mag\$f	Corriente Fase A
ReleO2	AXONBAYMeasurement	MMXU1	MX\$A\$phsB\$cVal\$mag\$f	Corriente Fase B
ReleO2	AXONBAYMeasurement	MMXU1	MX\$A\$phsC\$cVal\$mag\$f	Corriente Fase C
ReleO2	AXONBAYProtections	PTOC1	ST\$Str\$general	Triger del Interruptor 1 activado por Sobre corriente
ReleO2	AXONBAYControl	INTA_XCBR1	ST\$Pos\$stVal	Posición del Interruptor 1
ReleRpi	simpleOGenericIO	GGIO1	ST\$Ind1\$stVal	Protección del transformador 49T
ReleRpi	simpleOGenericIO	GGIO1	ST\$Ind2\$stVal	Protección del transformador 63T
ReleRpi	simpleOGenericIO	GGIO1	ST\$Ind3\$stVal	Protección del transformador 63B
ReleRpi	simpleOGenericIO	GGIO1	ST\$Ind4\$stVal	Protección del transformador 71Q

Tabla 7. Tags para la comunicación de SCADA.

Una vez mapeado cada uno de los tags presentes en la tabla 7, es posible visualizarlos en el Dashboard del ReLab OPC en tiempo real, tal como se muestra a continuación.

Full Name	Tag Name	Value	Timestamp	Quality	Access	Type
ReleIn.DemoMeasurement.I3pMMXU1.MXSA\$phsA\$CVal\$magSf	MXSA\$phsA\$CVal\$magSf	0	20/09/2020 22:07:17.655	Bad, Last Known	R	Float
ReleIn.DemoMeasurement.I3pMMXU1.MXSA\$phsB\$CVal\$magSf	MXSA\$phsB\$CVal\$magSf	0	20/09/2020 22:07:17.655	Bad, Last Known	R	Float
ReleIn.DemoMeasurement.I3pMMXU1.MXSA\$phsC\$CVal\$magSf	MXSA\$phsC\$CVal\$magSf	0	20/09/2020 22:07:17.655	Bad, Last Known	R	Float
ReleIn.DemoProtCtrl.Obj1XCBR1.ST\$Pos\$stVal	ST\$Pos\$stVal	0	20/09/2020 22:07:17.655	Bad, Last Known	R	DWord
ReleIn.DemoProtCtrl.I3GtPTRC1.ST\$Tr\$general	ST\$Tr\$general	False	20/09/2020 22:07:17.655	Bad, Last Known	R	Boolean
ReleO1.AXONBAYMeasurement.MMXU1.MXSA\$phsA\$CVal\$magSf	MXSA\$phsA\$CVal\$magSf	0	20/09/2020 22:07:29.774	Bad, Last Known	R	Float
ReleO1.AXONBAYMeasurement.MMXU1.MXSA\$phsB\$CVal\$magSf	MXSA\$phsB\$CVal\$magSf	0	20/09/2020 22:07:29.774	Bad, Last Known	R	Float
ReleO1.AXONBAYMeasurement.MMXU1.MXSA\$phsC\$CVal\$magSf	MXSA\$phsC\$CVal\$magSf	0	20/09/2020 22:07:29.774	Bad, Last Known	R	Float
ReleO1.AXONBAYProtections.PTOC1.ST\$Str\$general	ST\$Str\$general	False	20/09/2020 22:07:29.774	Bad, Last Known	R	Boolean
ReleO1.AXONBAYControl.INTA_XCBR1.ST\$Pos\$stVal	ST\$Pos\$stVal	0	20/09/2020 22:07:29.774	Bad, Last Known	R	DWord
ReleO2.AXONBAYMeasurement.MMXU1.MXSA\$phsA\$CVal\$magSf	MXSA\$phsA\$CVal\$magSf	0	20/09/2020 22:08:25.980	Bad, Last Known	R	Float
ReleO2.AXONBAYMeasurement.MMXU1.MXSA\$phsB\$CVal\$magSf	MXSA\$phsB\$CVal\$magSf	0	20/09/2020 22:08:25.980	Bad, Last Known	R	Float
ReleO2.AXONBAYMeasurement.MMXU1.MXSA\$phsC\$CVal\$magSf	MXSA\$phsC\$CVal\$magSf	0	20/09/2020 22:08:25.980	Bad, Last Known	R	Float
ReleO2.AXONBAYProtections.PTOC1.ST\$Str\$general	ST\$Str\$general	False	20/09/2020 22:08:25.980	Bad, Last Known	R	Boolean
ReleO2.AXONBAYControl.INTA_XCBR1.ST\$Pos\$stVal	ST\$Pos\$stVal	0	20/09/2020 22:08:25.980	Bad, Last Known	R	DWord
ReleIn0.simpleIOGenericO.GGIO1.CO\$SPCSO1\$Oper\$ctlNum	CO\$SPCSO1\$Oper\$ctlNum	0	20/09/2020 22:08:57.253	Bad, Last Known	RW	Byte
ReleIn0.simpleIOGenericO.GGIO1.CO\$SPCSO1\$Oper\$ctlVal	CO\$SPCSO1\$Oper\$ctlVal	False	20/09/2020 22:08:57.253	Bad, Last Known	RW	Boolean
ReleIn0.ST\$SPCSO1\$stVal	ST\$SPCSO1\$stVal	False	20/09/2020 22:08:57.253	Bad, Last Known	R	Boolean
ReleRpi.simpleIOGenericO.GGIO1.ST\$Ind1\$stVal	ST\$Ind1\$stVal	False	20/09/2020 22:09:28.279	Bad, Last Known	R	Boolean
ReleRpi.simpleIOGenericO.GGIO1.ST\$Ind2\$stVal	ST\$Ind2\$stVal	False	20/09/2020 22:09:28.279	Bad, Last Known	R	Boolean
ReleRpi.simpleIOGenericO.GGIO1.ST\$Ind3\$stVal	ST\$Ind3\$stVal	False	20/09/2020 22:09:28.279	Bad, Last Known	R	Boolean
ReleRpi.simpleIOGenericO.GGIO1.ST\$Ind4\$stVal	ST\$Ind4\$stVal	False	20/09/2020 22:09:28.279	Bad, Last Known	R	Boolean

Figura 103. Tags mapeados en ReLab OPC.

Para importar los Tags mapeados se crea un Driver OPC Cliente en ElipsePower SCADA (ver apartado 1.2.8), luego se ejecuta el Driver OPC y se importan todos los tags que se encuentran en el Dashboard ReLab. En la figura 104 se muestran los drivers que anteriormente se habían creado desde ReLab.

Para agregar los tags al GrupoOPC del Driver Protecciones bastara con seleccionar el driver que se desea configurar desde el SCADA y arrastrarlo al GrupoOPC, luego se hace clic en OK, al realizar esta acción el driver creado en SCADA se comunicara con el ReLab OPC extrayendo la información en tiempo real de aquellos tags que se encuentra en el dashboard.

Las configuración y asociaciones de los Tags a los elementos del modelo eléctrico se realizan desde el ítem subestaciones ubicados en el PowerElipse. La figura 105 muestra cada uno de los elementos presentes en el modelo eléctrico.

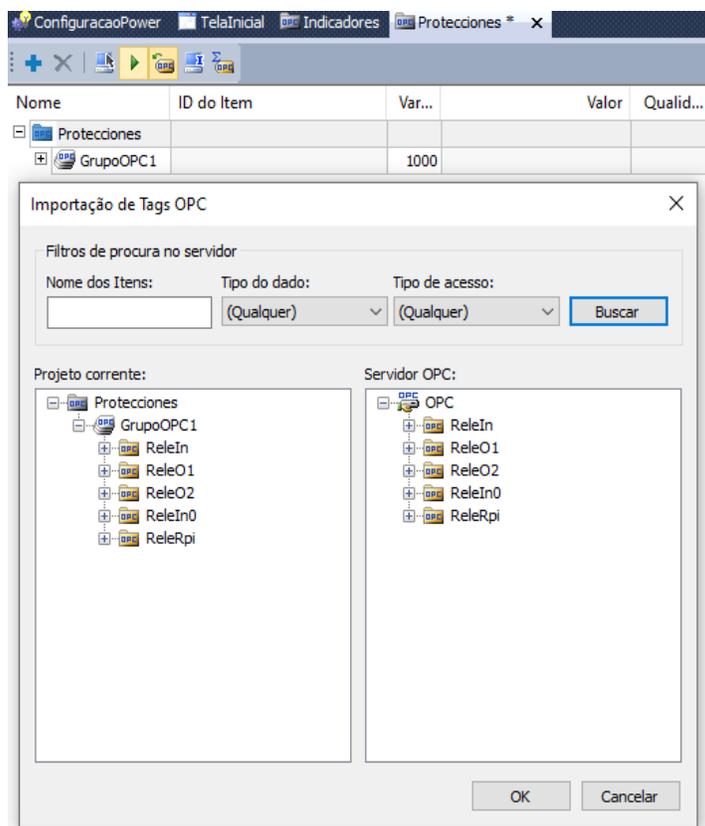


Figura 104. Importando Tags desde el servidor Relab OPC.

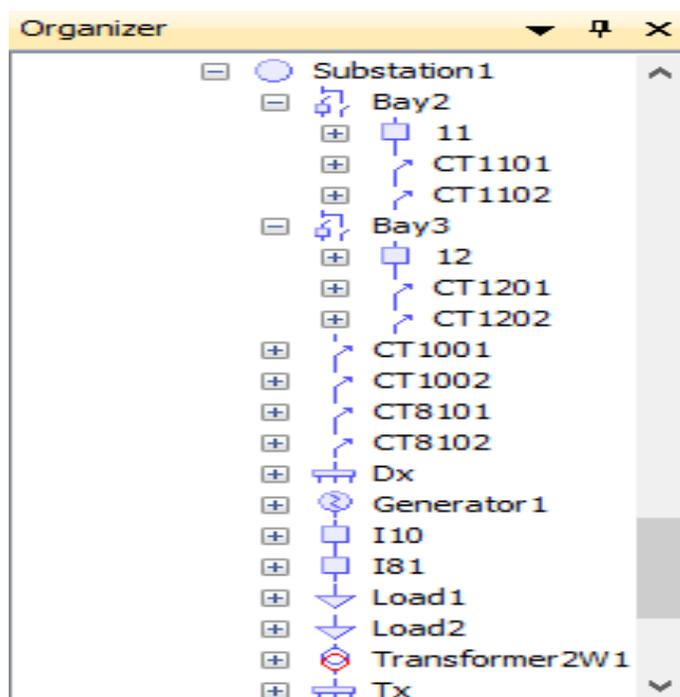


Figura 105. Elementos del modelo eléctrico en PowerElipse.

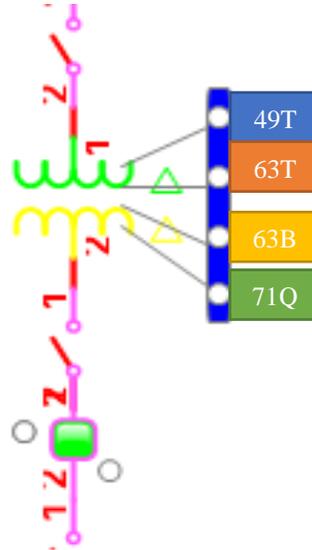


Figura 106. Indicadores del transformador.

Como se logra apreciar en la figura 106 en la pantalla principal del SCADA también se agregan indicadores y alarmas, las cuales están asociados a los disparos de los interruptores y las protecciones del transformador.

2.3.1.1 Asignación de Tags a los elementos e indicadores del modelo eléctrico

Los tags que serán asociados a cada uno de los elementos del modelo eléctrico y de los indicadores del modelo SCADA se muestran en la tabla 8, por ser una versión de prueba PowerElipse solo permite importar 20 tags en un Driver por lo que únicamente los elementos que se asociarán a los tags serán: indicadores de protección del transformador, estados del interruptor y medidas de corriente.

Driver	LD	LN	Tags	Elemento
ReleIn	DemoMeasurement	I3pMMXU1	MX\$A\$phsA\$cVal\$mag\$f	I10 (Corriente fase A)
ReleIn	DemoMeasurement	I3pMMXU1	MX\$A\$phsB\$cVal\$mag\$f	I10 (Corriente fase B)
ReleIn	DemoMeasurement	I3pMMXU1	MX\$A\$phsC\$cVal\$mag\$f	I10 (Corriente fase C)
ReleIn	DemoProtCtrl	Obj1XCBR1	ST\$Pos\$stVal	I10
ReleIn	DemoProtCtrl	I3GtPTRC1	ST\$Tr\$general	Indicador I10
ReleIn0	simpleOGenericIO	GGIO1	CO\$SPCSO1\$Oper\$ctVal	Control I81
ReleIn0	simpleOGenericIO	GGIO1	ST\$SPCSO1\$stVal	I81
ReleO1	AXONBAYMeasurement	MMXU1	MX\$A\$phsA\$cVal\$mag\$f	I11 (Corriente Fase A)
ReleO1	AXONBAYMeasurement	MMXU1	MX\$A\$phsB\$cVal\$mag\$f	I11 (Corriente Fase B)

ReleO1	AXONBAYMeasurement	MMXU1	MX\$A\$phsC\$cVal\$mag\$f	I11 (Corriente Fase C)
ReleO1	AXONBAYProtections	PTOC1	ST\$Str\$general	Indicador I11
ReleO1	AXONBAYControl	INTA_XCBR1	ST\$Pos\$stVal	I11
ReleO2	AXONBAYMeasurement	MMXU1	MX\$A\$phsA\$cVal\$mag\$f	I12 (Corriente Fase A)
ReleO2	AXONBAYMeasurement	MMXU1	MX\$A\$phsB\$cVal\$mag\$f	I12 (Corriente Fase B)
ReleO2	AXONBAYMeasurement	MMXU1	MX\$A\$phsC\$cVal\$mag\$f	I12 (Corriente Fase C)
ReleO2	AXONBAYProtections	PTOC1	ST\$Str\$general	Indicador I12
ReleO2	AXONBAYControl	INTA_XCBR1	ST\$Pos\$stVal	I12
ReleRpi	simpleOGenericIO	GGIO1	ST\$Ind1\$stVal	Indicador 49T
ReleRpi	simpleOGenericIO	GGIO1	ST\$Ind2\$stVal	Indicador 63T
ReleRpi	simpleOGenericIO	GGIO1	ST\$Ind3\$stVal	Indicador 63B
ReleRpi	simpleOGenericIO	GGIO1	ST\$Ind4\$stVal	Indicador 71Q

Tabla 8. Tags asociados a los elementos del modelo eléctrico.

Para poder asociar un tag a un elemento del modelo eléctrico, se selecciona el elemento que se desea configurar, para el ejemplo se toma el interruptor I10, el cual cuenta con medidas de corriente y posición para su monitoreo en SCADA, se selecciona el tipo de dato del interruptor que se desea configurar y se hace clic en el símbolo “tres puntos” de la sección tag, tal como se muestra en la siguiente figura.

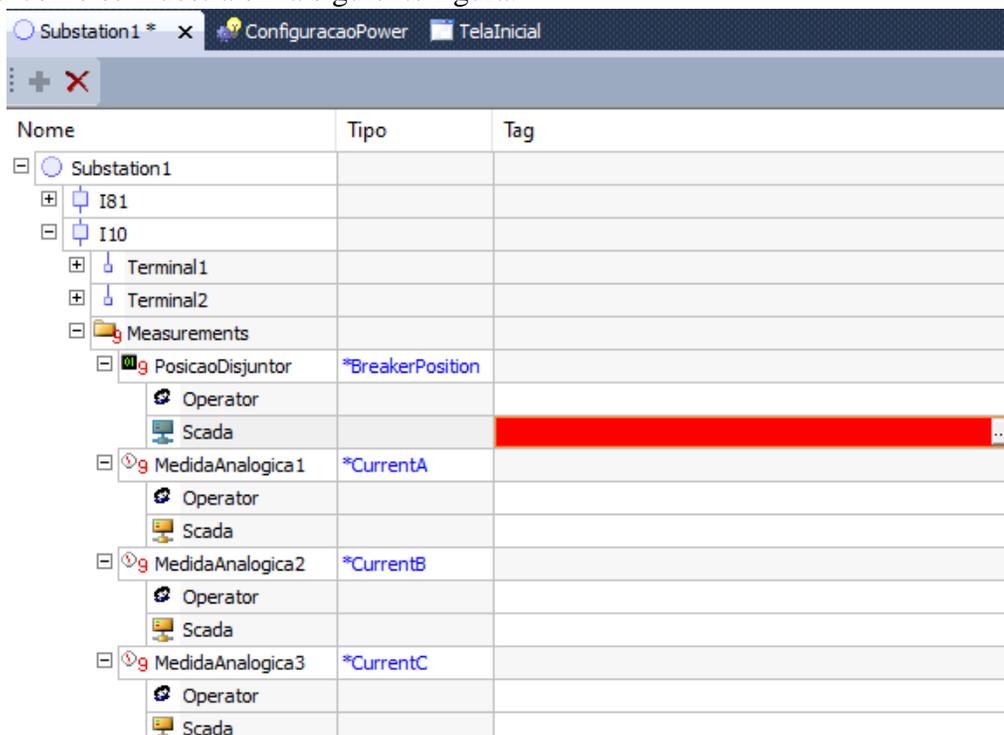


Figura 107. Asignación de Tag para la posición del interruptor I10.

Seguidamente aparece una nueva ventana donde se selecciona el Driver donde se encuentra el tag de posición del interruptor I10, se selecciona el tag como *Value* y hace clic en *Colar*, tal como se muestra en la figura 108.

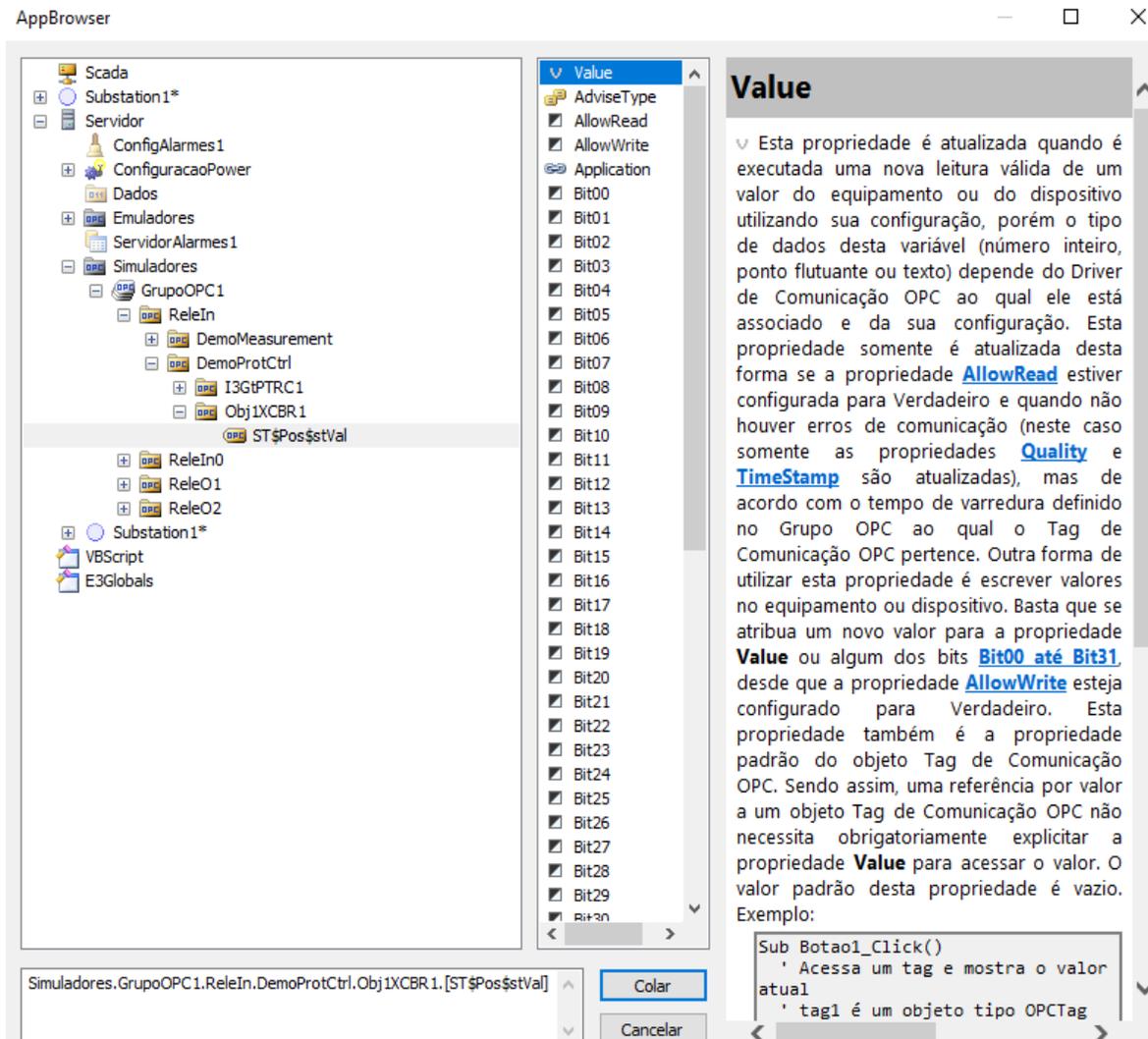


Figura 108. Tag de posición para el interruptor I10.

Para asociar los tags de las mediciones de corriente del interruptor I10 se seguirán los pasos antes mencionados, la figura 109 muestra la configuración de tags del interruptor I10 para su posición y mediciones de corriente.

Substation1		
I81		
I10		
Terminal1		
Terminal2		
Measurements		
PosicaoDisjuntor	*BreakerPosition	
Operator		
Scada		Simuladores.GrupoOPC1.ReleIn.DemoProtCtrl.Obj1XCBR1.[ST\$Pos\$stVal].Value
MedidaAnalogica1	*CurrentA	
Operator		
Scada		Simuladores.GrupoOPC1.ReleIn.DemoMeasurement.I3pMMXU1.[MX\$A\$phsA\$scVal\$mag\$].Value
MedidaAnalogica2	*CurrentB	
Operator		
Scada		Simuladores.GrupoOPC1.ReleIn.DemoMeasurement.I3pMMXU1.[MX\$A\$phsB\$scVal\$mag\$].Value
MedidaAnalogica3	*CurrentC	
Operator		
Scada		Simuladores.GrupoOPC1.ReleIn.DemoMeasurement.I3pMMXU1.[MX\$A\$phsC\$scVal\$mag\$].Value

Figura 109. Configuración de tags para los parámetros eléctricos del interruptor I10.

Para la configuración de los interruptores restantes se siguen los mismos pasos que se realizaron en el interruptor I10, se utiliza la tabla 8 como guía para la asignación de tags de cada elemento.

De manera similar se realizan la asignación de tags para los indicadores, para el ejemplo se toma el indicador de bajo/alto nivel de aceite del transformador (71Q), para ello se selecciona el indicador 71Q y se hace clic derecho sobre él, seguidamente se selecciona el ítem propiedades, esto abrirá una nueva ventana la cual se muestra en la siguiente figura.

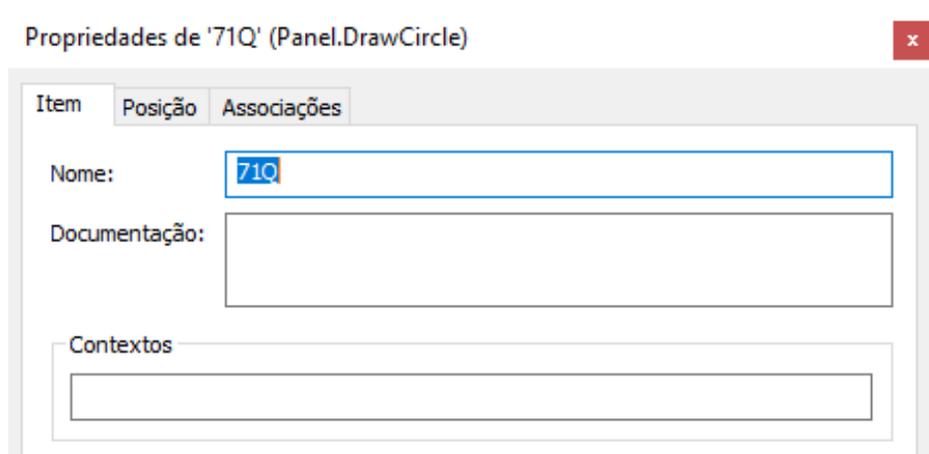


Figura 110. Propiedades del indicador 71Q.

En la ventana propiedades del indicador se selecciona la pestaña con nombre “Associacoes”, luego se selecciona el ítem *ForegrounColor* y se hace clic sobre el símbolo “tres puntos” en la sección *Fonte*, tal como se muestra en la figura siguiente.

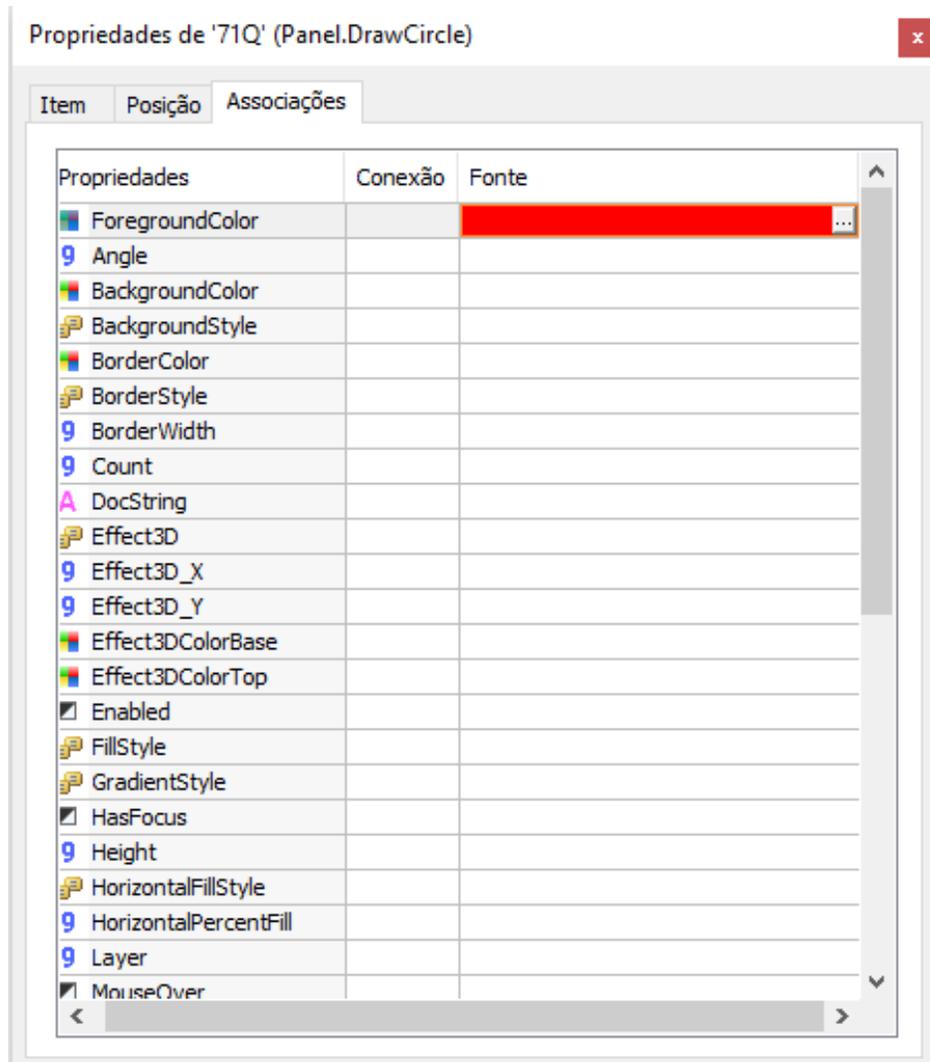


Figura 111. Asignación de Tags al indicador 71Q.

Luego de efectuar los pasos descritos en el párrafo anterior se abrirá una nueva ventana donde se selecciona el Driver y tag que contiene la información del indicador que se desea configurar, se selecciona *Valué* y se hace clic en *Colar*, tal como lo muestra la figura 112.

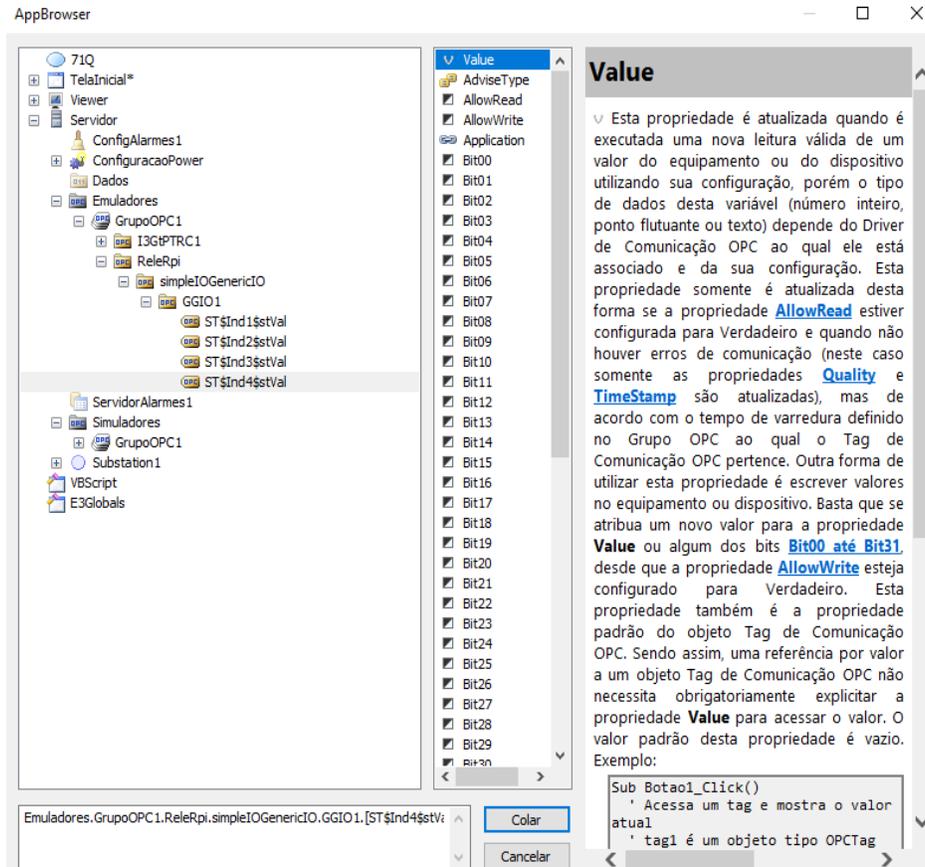


Figura 112. Configuración de TAG para el indicador 71Q.

Una vez que se ha configurado el tag del indicador 71Q se debe asignar un evento para poder visualizar en el SCADA cuando este sea activado, para ello se selecciona el item *ForegroundColor* y en la sección *Conexao* de la ventana propiedades, se selecciona la opción *Conexao por Tabela*, tal como se muestra en la figura 113.

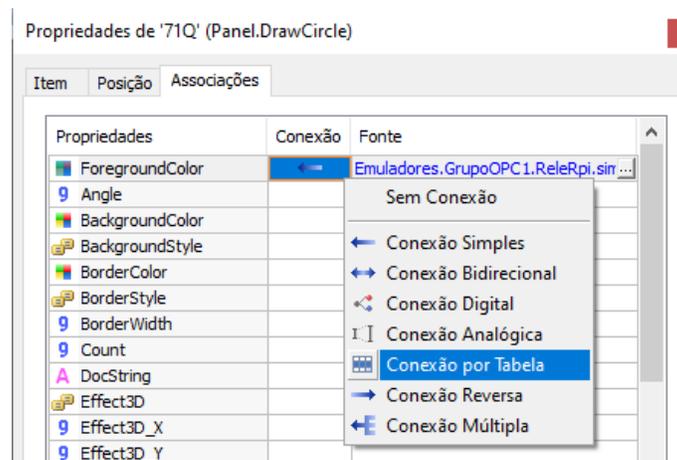


Figura 113. Asignación de evento del indicador 71Q.

Luego se abrirá una nueva ventana donde se agrega un nuevo evento para el cual se configura los valores Min. y Max. con un valor de 0 y 1 respectivamente, se asignará un color intermitente al indicador (rojo-amarillo), esto se realizará seleccionando la opción *pisca*, tal como se puede observar en la figura 114.

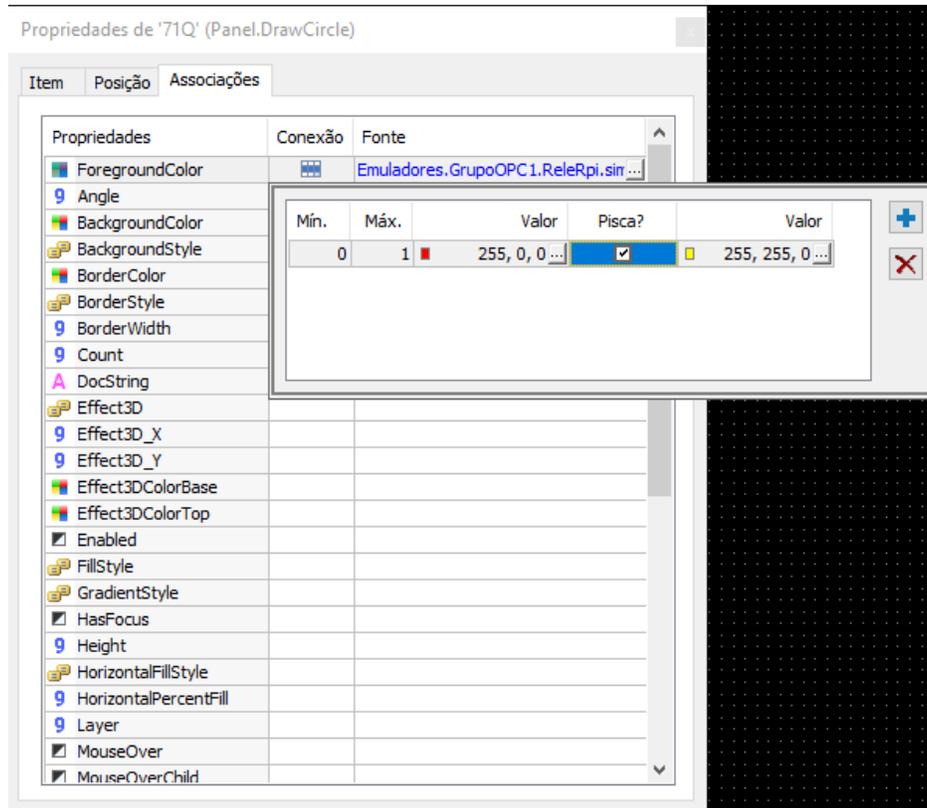


Figura 114. Configuración del evento de activación de indicador 71Q.

Esta configuración permitirá mostrar en el SCADA al indicador bajo un color intermitente (rojo-amarillo) cuando este sea activado.

2.3.1.2 Creación de Alarmas en PowerEllipse

Para la creación de una alarma en elipse SCADA es necesario crear una configuración de alarma la cual se realiza desde la sección *alarmes*, en dicha sección se selecciona la opción *configAlarmes1*, tal como se muestra en la figura siguiente.

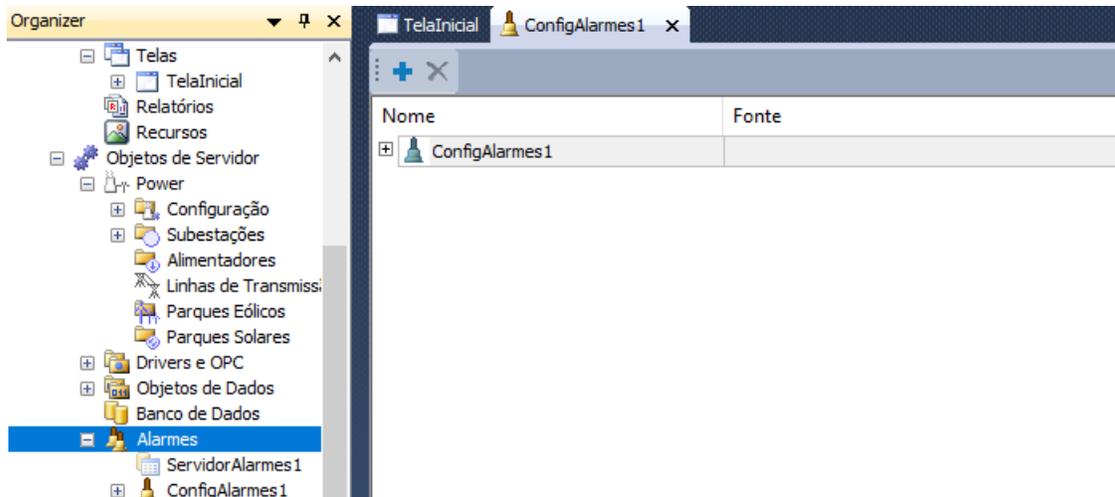


Figura 115. Configuración de Alarmas en PowerElipse SCADA.

En *configAlarmes1* se hace clic derecho y seguidamente aparecerá un submenú donde se selecciona el tipo de alarmas que se pueden configurar tal como se muestra en la figura 116.

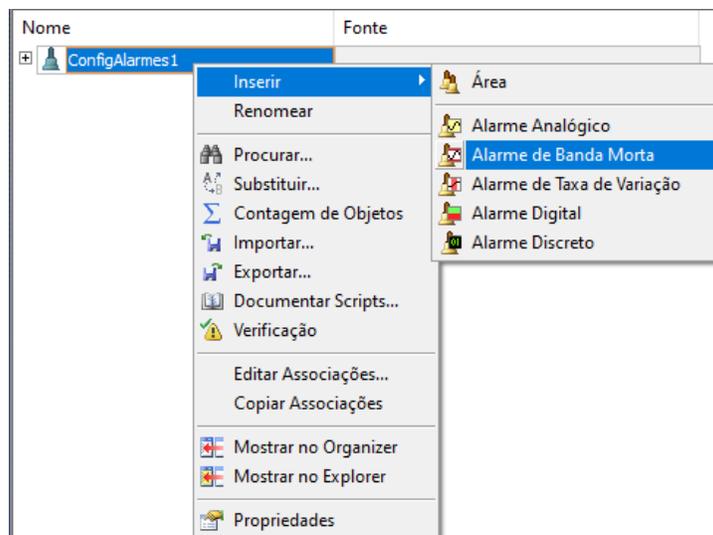


Figura 116. Creación de Alarmas en PowerElipse SCADA.

Para propósitos de este documento únicamente se utilizarán alarmas digitales y análogas, que se activarán por disparos de los interruptores y protecciones del transformador o por niveles altos de corrientes.

Para asignar un tag a una alarma se seguirán los mismos pasos que anteriormente se han descrito en los apartados de indicadores y elementos del modelo eléctrico, una vez se ha asignado un tag a la alarma se selecciona y se hace clic derecho sobre él, al efectuar esta acción se despliega el submenú mostrado en la figura anterior donde se selecciona el ítem propiedades, seguidamente aparece la ventana que se muestra en la figura 117.

Para el ejemplo se configura una alarma digital cuyo tag ha sido asignado al disparo por sobre corriente del relé AxonGroup.

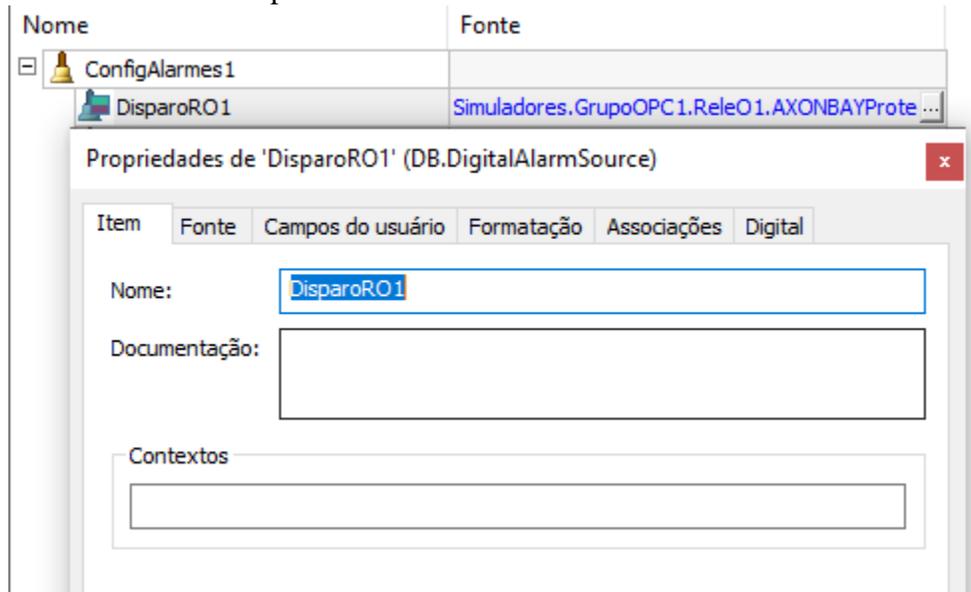


Figura 117. Configuración de Alarma para disparo del Relé 1 AxonGroup.

Luego de asignar un nombre a la alarma (DisparoRO1) se selecciona la sección *Digital*, aquí se indica el texto que se desea que aparezca en condiciones normales (ReleO1 en buen estado) y en condiciones de falla (Sobre corriente en ReleO1), así como la importancia de la alarma (Alta), al finalizar la configuración se habilita la alarma, tal como se muestra en la figura siguiente.

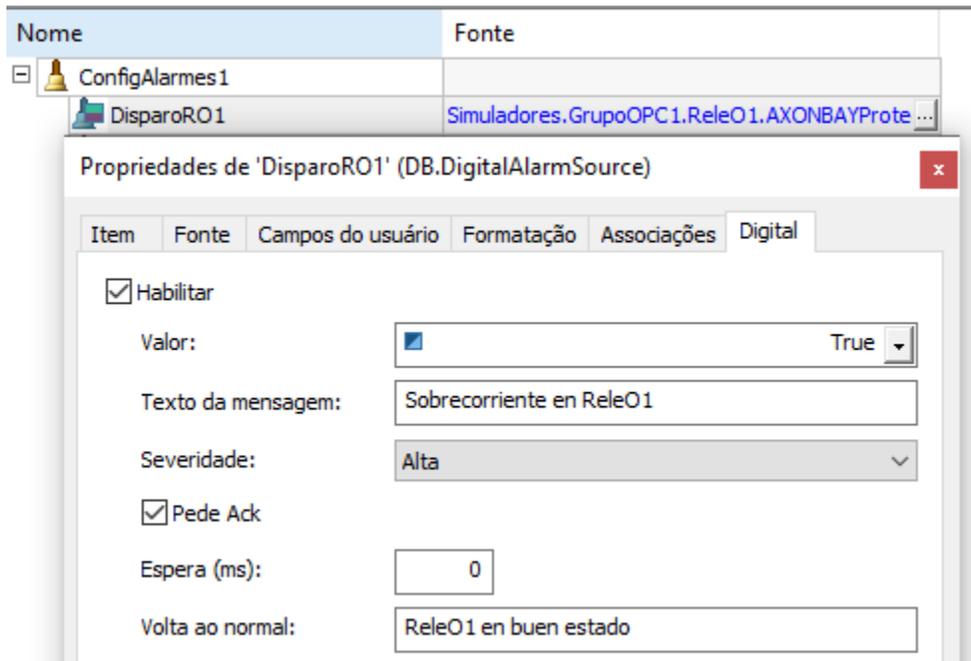


Figura 118. Configuración de alarma digital para el disparo del Relé 1 de AxonGroup.

Para la configuración de alarmas digitales por disparo por sobre corriente de un relé y para las protecciones del transformador se seguirán cada uno de los pasos que se han descrito anteriormente.

Para la configuración de alarmas analógicas se crea una alarma análoga desde el submenú mostrado en la figura 116, luego de asignar un tag y un nombre a la alarma se selecciona la sección *Analógico* de la ventana propiedades tal como se muestra en la figura 118. Para el ejemplo se configura la corriente de la fase A del relé INFOTECH.

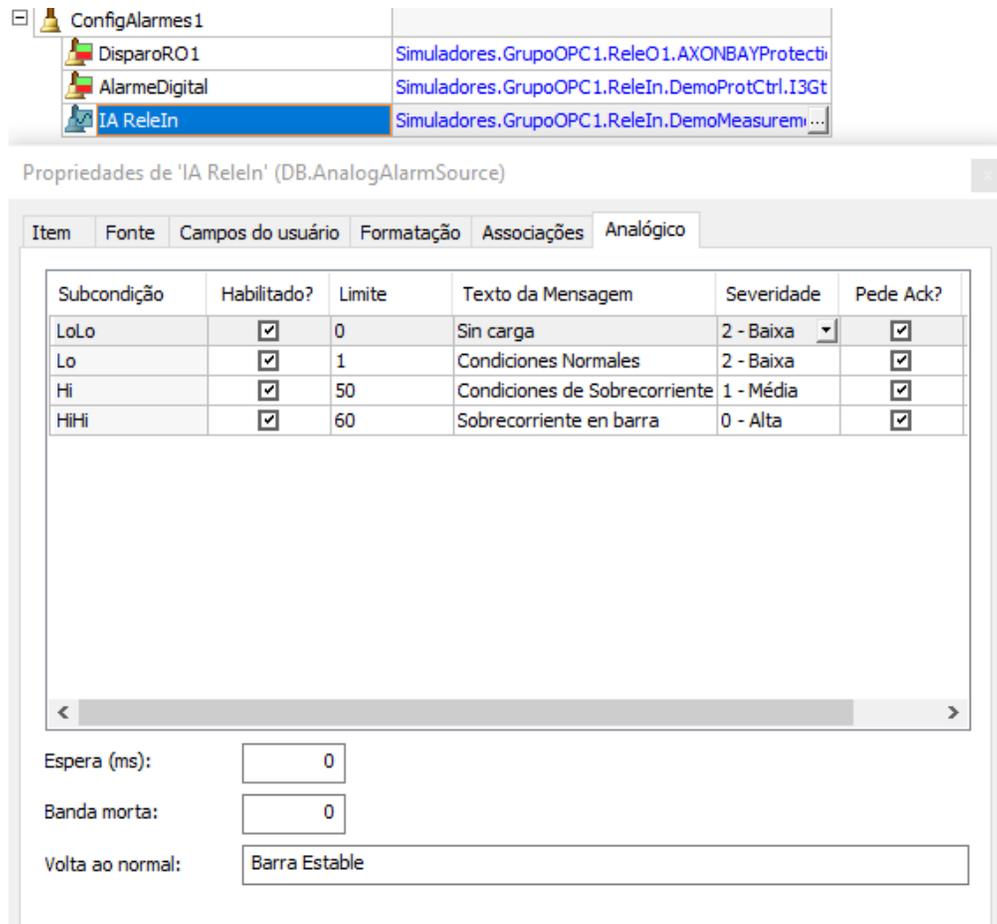


Figura 119. Configuración de alarma analógica para niveles altos de corriente del relé INFOTECH.

En la figura 119 se presenta cuatro condiciones de niveles de corriente LoLo, Lo, Hi, HiHi los cuales se clasifican según su importancia, para el caso se considera con valor muy bajo (Lolo) el nivel de corriente 0A para el cual mostrará un mensaje de alerta “Sin carga”, para un nivel de corriente de 1A se considera un nivel bajo (Lo) para el cual se mostrará un mensaje “Condiciones Normales”.

Se considera un nivel alto (Hi) y de importancia media a los niveles de corrientes entre 50A y 60A en este caso se mostrará un mensaje de alerta “Condiciones de sobre corriente”, para valores mayores a 60A se considera un nivel muy alto (HiHi) y de importancia alta para el cual se mostrará un mensaje de alerta “Sobre corriente en la barra”. De esta manera se configuran las alarmas analógicas para medidas de corriente.

Las alarmas analógicas relacionadas a las fases restantes del relé y de los relés restantes se configurarán siguiendo los pasos que se han descrito en los párrafos anteriores.

Finalmente se debe asignar en la pantalla principal un resumen de alarmas para poder visualizar todas las alarmas que se han creado, para ello en la pantalla principal del SCADA y en barra de herramientas se selecciona el icono *E3Alarm (Sumário de Alarmes)* y se arrastra hacia el área de trabajo, luego se configura el tamaño que tendrá el panel, tal como se presenta en la figura 120.

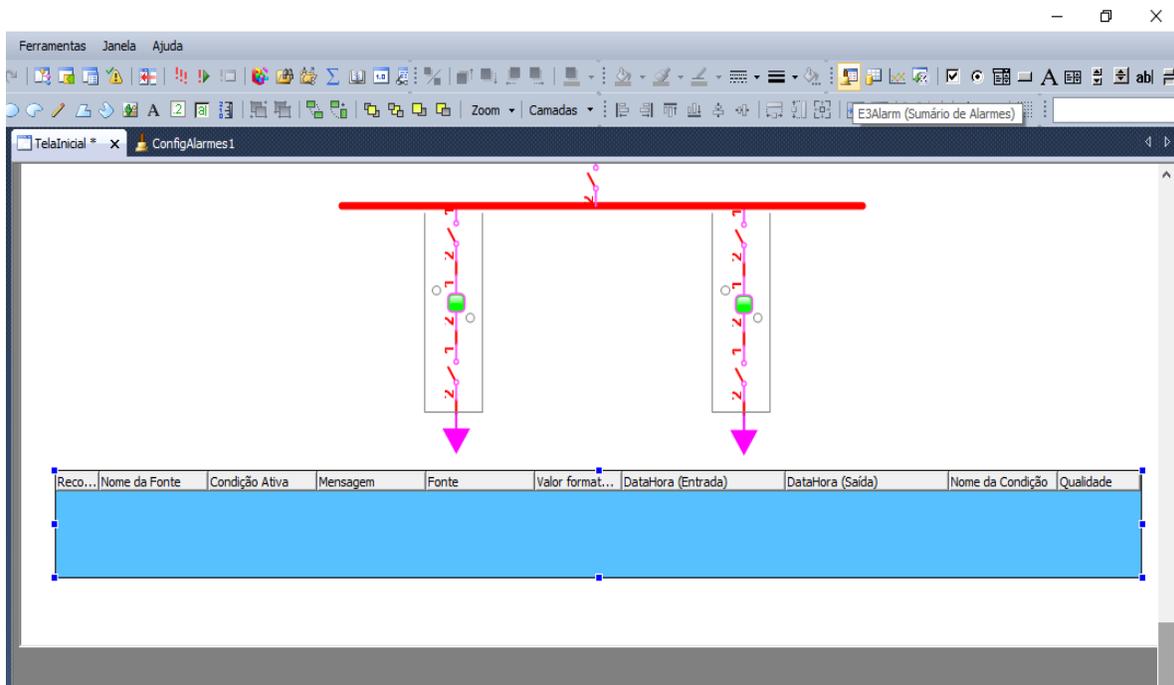


Figura 120. Asignación de panel para visualización de alarmas.

CAPÍTULO III

Escenarios para pruebas de un SAS bajo el estándar
IEC61850

3.0 Introducción

En el presente capítulo se describen tres escenarios de pruebas para una subestación de barra simple en el cual los IED's encargados de las protecciones son dispositivos simuladores y emuladores revisados y configurados en el capítulo anterior.

La figura 121 muestra el modelo de una subestación de distribución de barra simple el cual posee un transformador delta/estrella aterrizada 46/23 kV en su entrada y dos bahías en su salida.

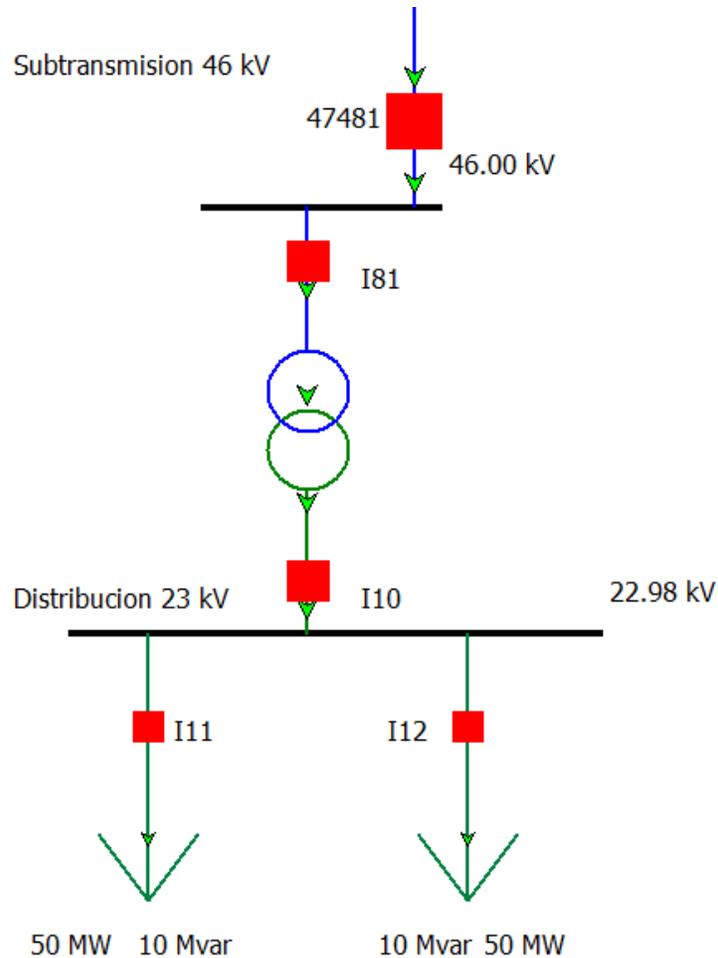


Figura 121. Diagrama general para los escenarios de pruebas.

Los escenarios que se presentan en este capítulo han sido determinados a partir del diagrama general presentado en la figura anterior, los cuales son:

- **Escenario 1:** Protecciones de un transformador de Potencia.
- **Escenario 2:** Protección de barra simple por sobre corriente instantánea.
- **Escenario 3:** Simulación de niveles de bahía y estación (SCADA).

3.1 Escenario 1, esquema de actuación de protecciones mecánicas de un transformador de potencia

El objetivo en este escenario es simular la actuación de protecciones mecánicas de un transformador, enviando mensajes GOOSE para el disparo de los interruptores I81 e I10, despejando la falla en el transformador.

3.1.1 Descripción y Funcionamiento

El escenario se muestra en la figura 122 y consiste en una subestación de distribución. Las protecciones eléctricas del I81 las controla el relé SEL – 351 el cual está conectado a la red de comunicación que se muestra en la figura 70 del capítulo 2 y cuyos elementos funcionan bajo el estándar IEC61850. También se incluye el IEDRASPBerry el cual tiene la función de convertir las señales de 4 pines GGIO mostrados en la figura 88 que representa las protecciones mecánicas de un transformador de potencia, las cuales son:

- 49T: Relé de sobrepresión
- 63T: Relé buchholz
- 63B: Relé de presión súbita
- 71Q: Nivel Alto/Bajo de aceite

Las señales que envían dichas protecciones mecánicas son convertidas a mensajes GOOSE y son monitoreadas por los interruptores I81 y el interruptor INFOTECH, se utiliza el software Wireshark ejecutándose en modo run para visualizar el mensaje GOOSE enviado por cada una de las protecciones, un ejemplo de estos mensajes GOOSE puede observarse en la figura 130.

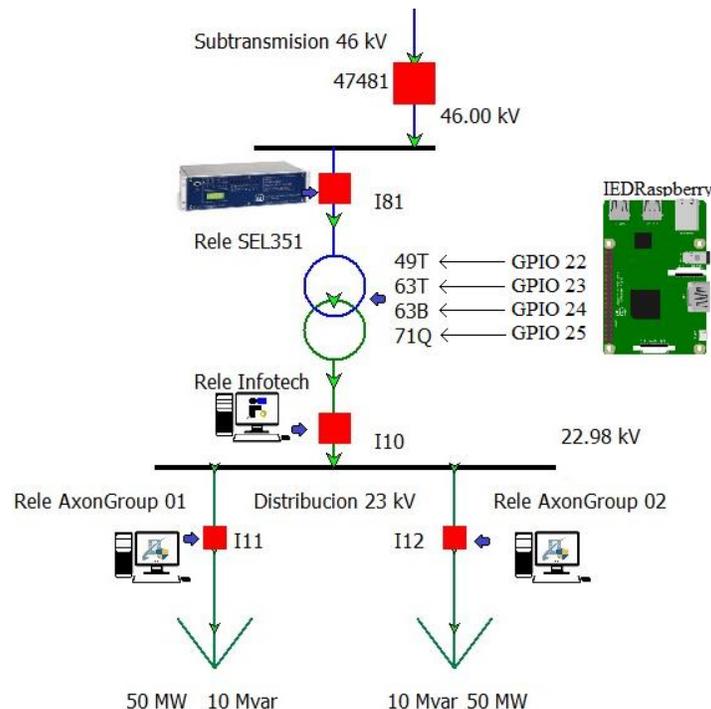


Figura 122. Esquema de Protección de transformador

3.1.2 Esquema de protecciones

Las protecciones mecánicas del transformador son condiciones en las cuales se debe aislar el transformador por lo que se debe abrir los interruptores del lado de sub transmisión (I81) y de distribución (I10) mostrados en la figura 122.

Cuando el IEDRaspberry envía un mensaje GOOSE simulando el disparo de una protección del transformador, el relé asociado al I81 abre el interruptor enviando a su vez un mensaje GOOSE al relé asociado a I10 para su apertura, esta acción despeja completamente la falla en el transformador.

3.1.3 Configuración de los equipos

El CID del IEDRaspberry ha sido configurado para el envío de mensajes GOOSE como respuesta del disparo de cualquiera de las protecciones simuladas del transformador, su configuración se muestra en el apartado 2.2.1 y puede visualizarse con el programa GooseExplorer tal como se muestra en la figura 123. (ver apartado 1.2.3.4 para la configuración del GooseExplorer)



Figura 123. Mensaje GOOSE enviado por las protecciones del transformador

El convertidor IEC61850/Modbus ha sido configurado para suscribirse a los mensajes GOOSE que envía el IEDRASPBerry y como respuesta al disparo de alguna de las protecciones mecánicas del transformador se enciende el relé 1 de su bornera para el comando de disparo del interruptor I81 y el relé 2 para indicación de disparo del interruptor INFOTECH el cual será disparado mediante un mensaje GOOSE.

3.1.4 Procedimiento descrito en pasos para realizar la prueba

- 1) Conectar la red de comunicación de los equipos tal como se muestra en la figura 124.

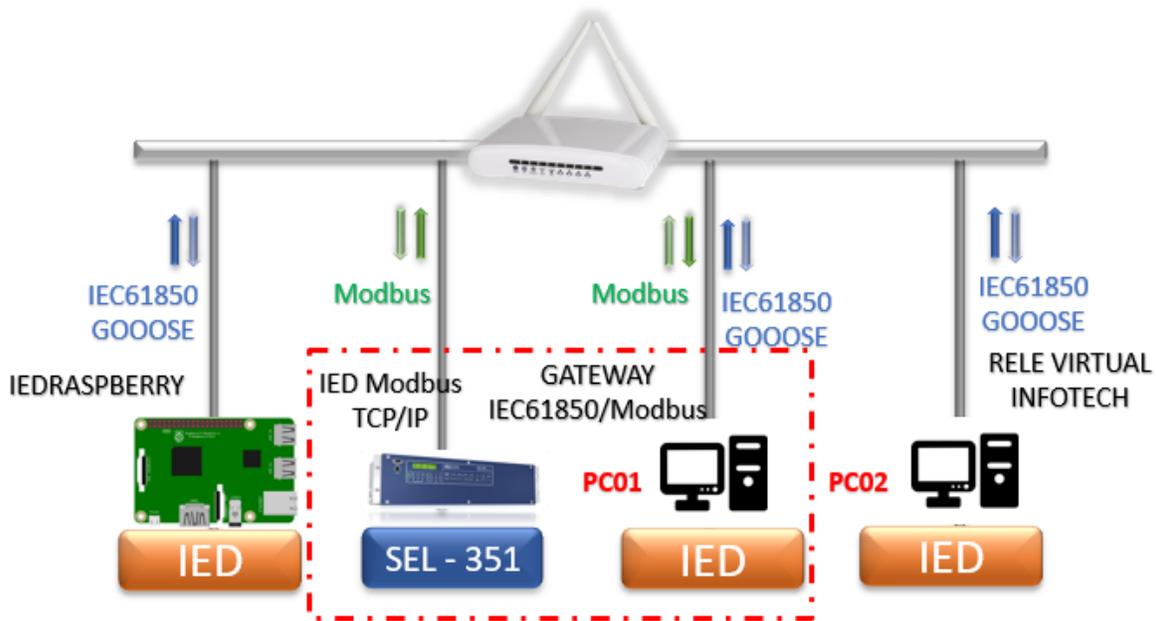


Figura 124. Esquema comunicaciones escenario #1

- 2) Iniciar el relé virtual de INFOTECH en la PC-02 y verificar mediante el software IEDexplorer instalado en la misma PC que el servidor IEC61850 este inicializado y recibiendo datos GOOSE de la posición del interruptor y el estado de los 5 indicadores del interruptor, también se verifica con la herramienta Goose Explorer que el relé esté conectado a través del adaptador de red Ethernet, el resultado se muestra en la figura 125.

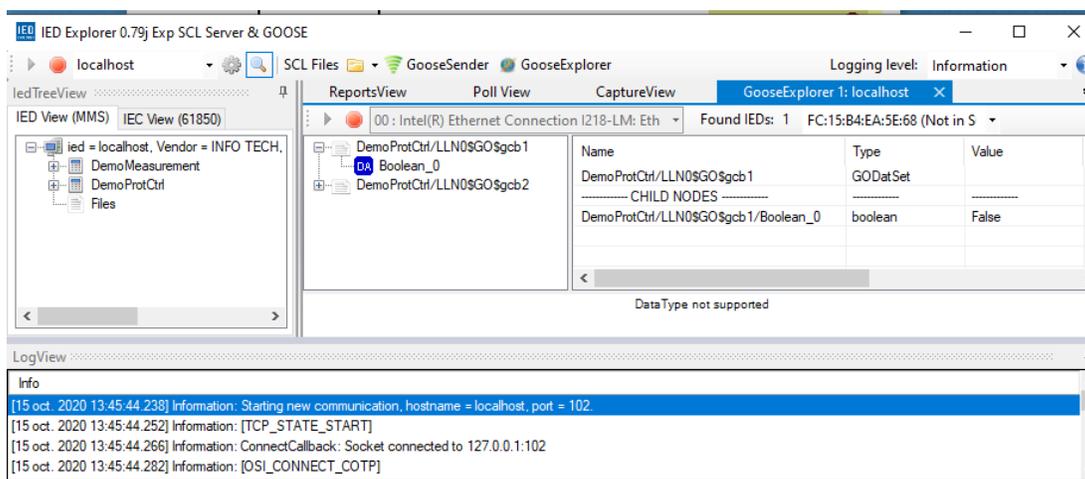


Figura 125. Verificación inicio interruptor INFOTECH.

- 3) Iniciar máquina virtual de Gateway IEC61850/Modbus en la PC-01, ingresar a una terminal bash y ejecutar el siguiente comando para cambiar a la ubicación donde se instaló la librería como se muestra en la sección 1.2.10.

```
cd /home/libiec61850-1.4/examples/rele_sel
```

Compilar y ejecutar la aplicación cuyo código fuente se encuentra en la misma ubicación /home/libiec61850-1.4/examples/rele_sel, mediante la instrucción:

```
$make; sudo ./server_example_control
```

La figura 126 muestra el resultado, en la terminal aparecen la dirección IP del relé SEL y el puerto utilizado, también aparece el número de evento GOOSE, la estampa de tiempo y tiempo en que ha estado recibiendo mensajes. Hasta no recibir nuevos mensajes GOOSE de parte del IEDRASPBerry no sea actualizaren los valores en la consola.

La última línea se refiere al estado de los mensajes GOOSE indicando las 4 entradas en el IEDRASPBerry, el quinto dígito se refiere a la banderilla de disparo del relé, la cual no permite que vuelva a dispararse el relé SEL mientras existan condiciones de alarma. Dicha alarma se resetea siempre que las condiciones de disparo en el transformador hayan sido normalizadas y se envíe un comando MMS de nivel alto al SPCS01 del convertidor.

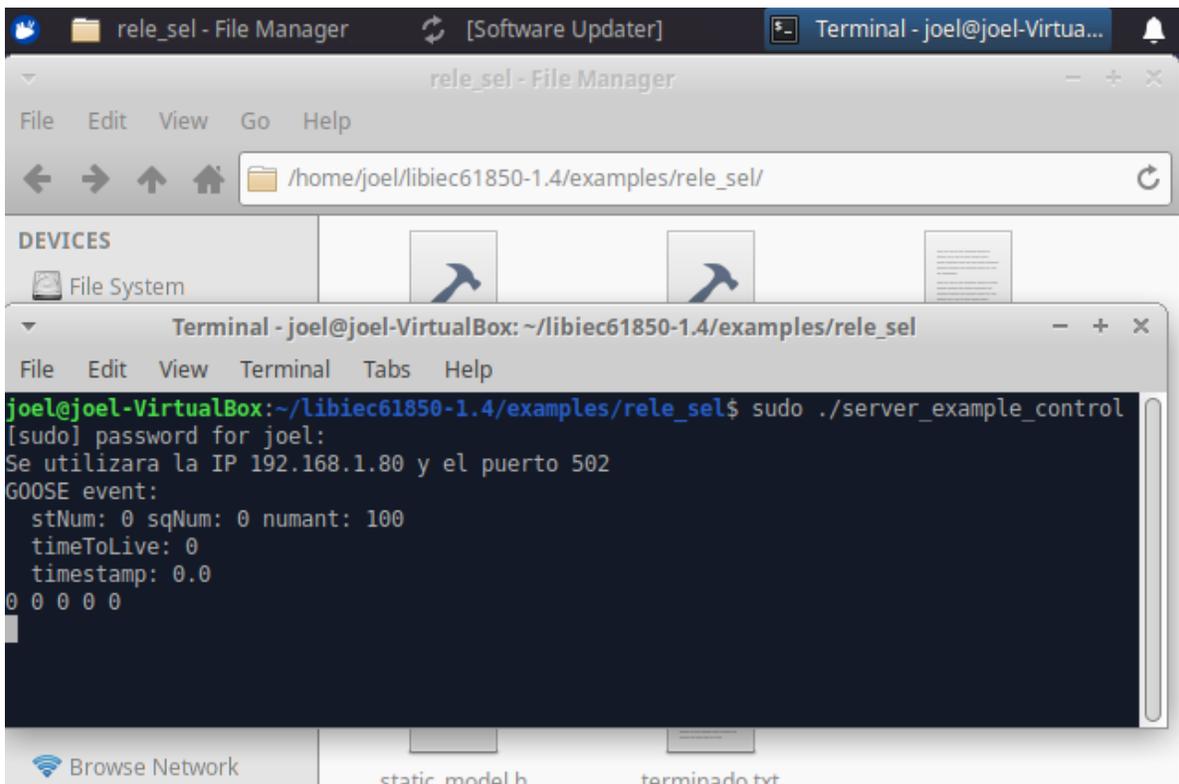


Figura 126. Ejecución de convertidor IEC61850/Modbus.

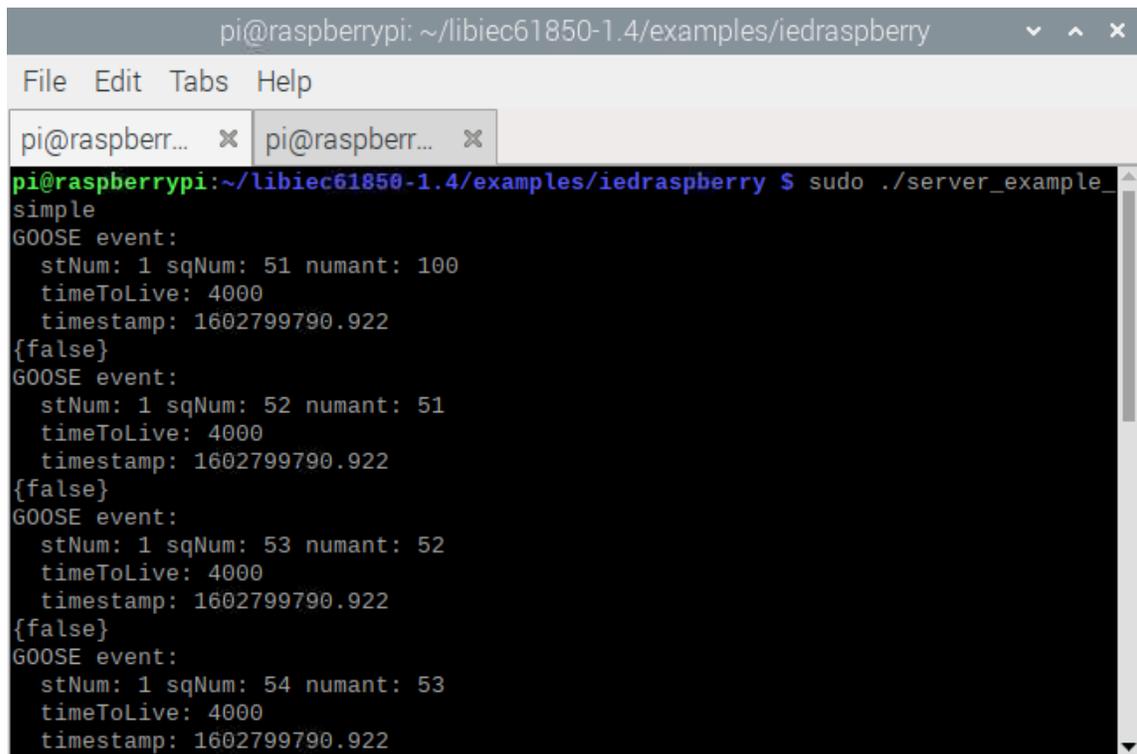
- 4) Acceder vía VNC viewer al IEDRaspberry, ingresar la dirección IP, el usuario(pi) y la contraseña (1234), tal como se muestra en el apartado 1.2.9. Ingresar a una terminal bash y ejecutar el siguiente comando para ubicarse en la carpeta de la aplicación:

```
$cd /home/pi/libiec61850-1.4/examples/iedraspberry
```

Compilar y ejecutar la aplicación mediante la instrucción:

```
$make; sudo ./server_example_control
```

En la terminal se observa el mensaje GOOSE del estado del interruptor INFOTECH utilizado en el escenario 2 al que está suscrito el IEDRASBPERRY también aparece el número de evento GOOSE y la estampa de tiempo. Hasta no recibir nuevos mensajes GOOSE de parte del relé INFOTECH no se actualizan los datos en la terminal, la figura 127 muestra el resultado de este paso.



```
pi@raspberrypi: ~/libiec61850-1.4/examples/iedraspberry
File Edit Tabs Help
pi@raspberr... x pi@raspberr... x
pi@raspberrypi:~/libiec61850-1.4/examples/iedraspberry $ sudo ./server_example_
simple
GOOSE event:
  stNum: 1 sqNum: 51 numant: 100
  timeToLive: 4000
  timestamp: 1602799790.922
{false}
GOOSE event:
  stNum: 1 sqNum: 52 numant: 51
  timeToLive: 4000
  timestamp: 1602799790.922
{false}
GOOSE event:
  stNum: 1 sqNum: 53 numant: 52
  timeToLive: 4000
  timestamp: 1602799790.922
{false}
GOOSE event:
  stNum: 1 sqNum: 54 numant: 53
  timeToLive: 4000
  timestamp: 1602799790.922
```

Figura 127. Ejecución de convertidor IEC61850/Modbus.

3.1.5 Evento #1, actuación de protecciones mecánicas del transformador de potencia

La figura 128 muestra la activación de una de las entradas del IEDRASBPERRY, dato recibido mediante la suscripción GOOSE del convertidor IEC61850/Modbus. Como consecuencia se envía vía modbus el comando de activación de la salida #1 del relé SEL por 1 segundo tal como se muestra en la HMI del software AcSELerator Quickset mostrada en la figura 129, también se enciende la banderilla en representación de “ALARMA” y que se

observa como el quinto bit en la consola con el valor de “1”. La activación de dicha banderilla no permite la normalización del estado del relé aun cuando la condición de fallo sea reestablecida.

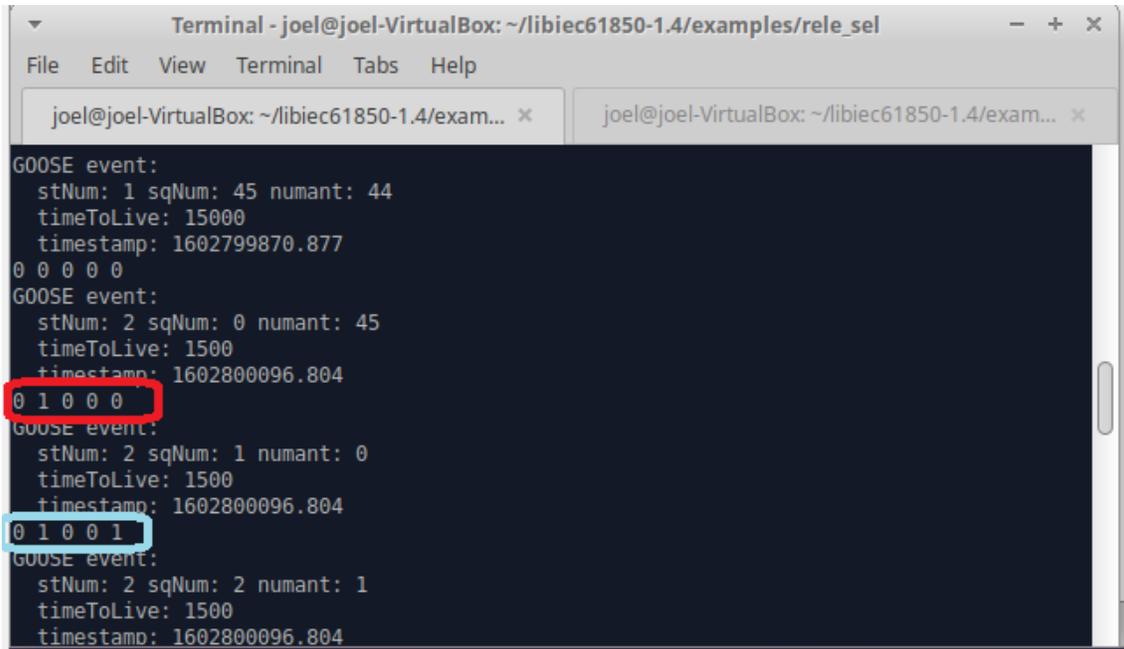


Figura 128. Detección de disparo de protección mecánica de TX.

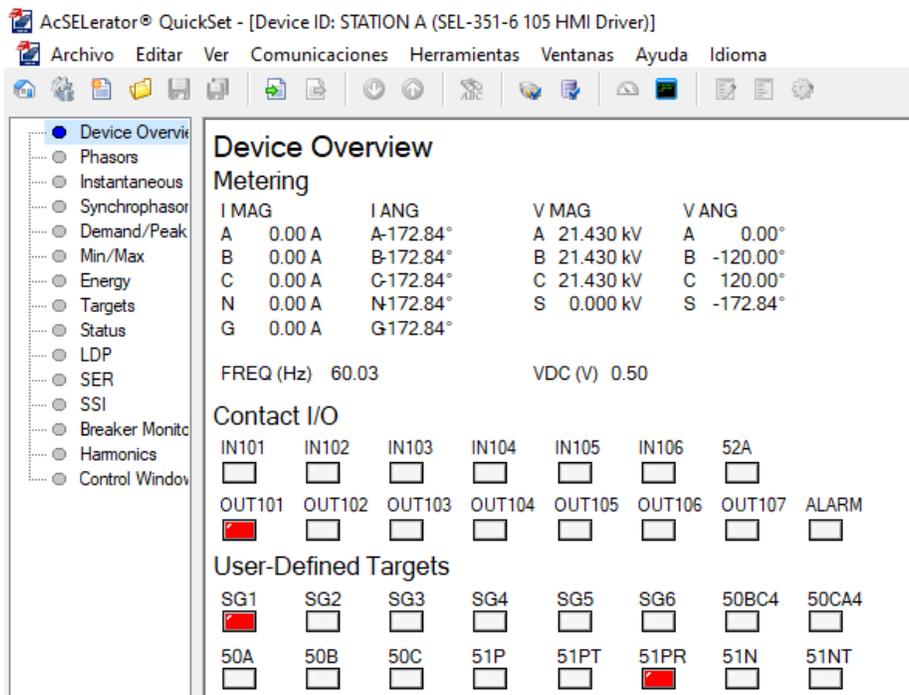


Figura 129. HMI AcSElerator Quickset.

En la figura 130 se utilizó Wireshark para analizar los mensajes GOOSE que envía el IEDRASPBerry del estado de sus entradas al convertidor IEC61850/Modbus:

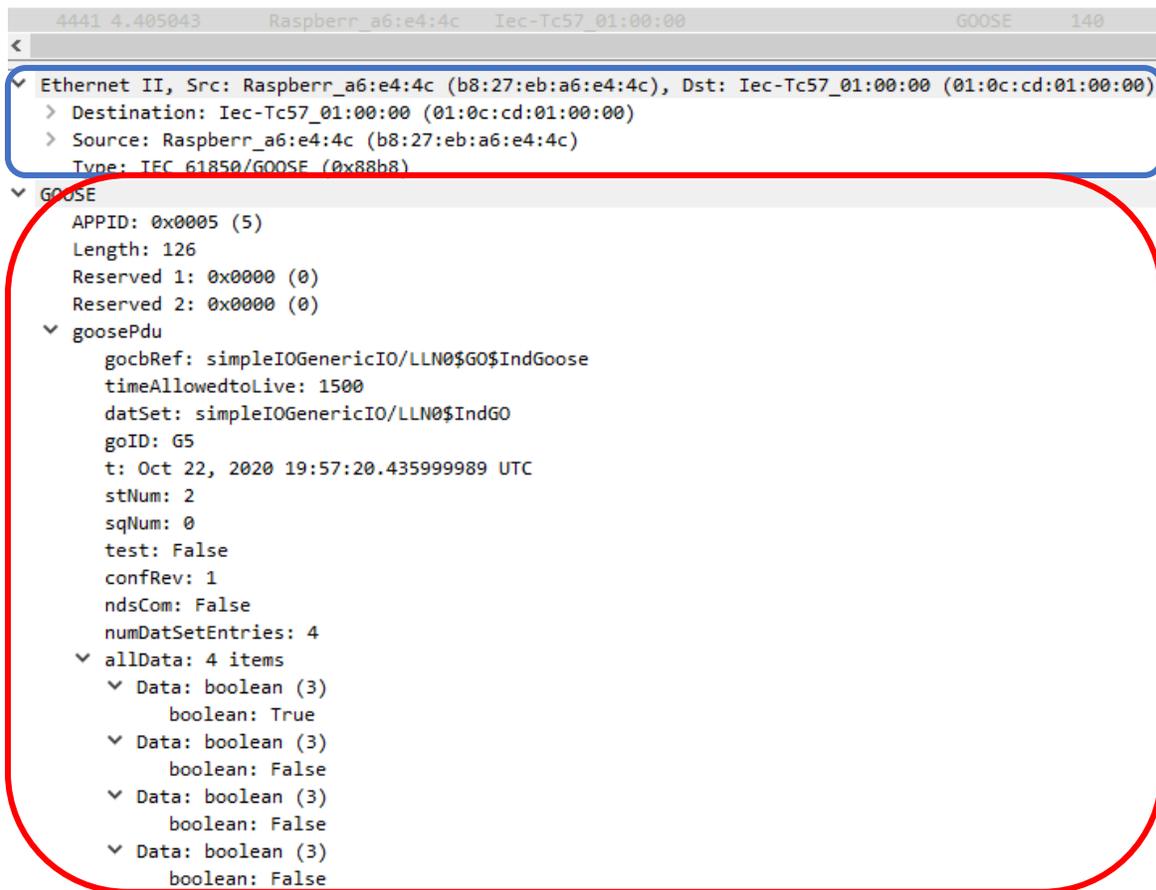


Figura 130. Mensaje GOOSE enviado por IEDRASPBerry.

De la figura anterior es posible identificar los siguientes parámetros del mensaje:

- Encerrado en color azul se encuentran los parámetros de la **Fuente** (Raspberr_a6:e4:4c) y **Destino** (Iec-Tc57_01:00:01) del mensaje GOOSE, también es posible observar la MAC del **publicador** (b8:27:eb:e6:e4:4c) y del **suscriptor** (01:0c:cd:01:00:02) y el tipo de mensaje (GOOSE) e Ethertype (0x88b8).
- Encerrado en color rojo se encuentra el contenido del mensaje GOOSE **APPID**, **GoRef**, **DataSet**, **GoID**, estampa de tiempo (**t**), cambios de estado en el DS (**stNum**), veces en que se ha enviado el mensaje GOOSE (**SqNum**), modo de prueba del IED (**test**) y el estado del dato del mensaje GOOSE booleano (**Data**)

3.1.6 Evento #2, Restablecimiento de condiciones iniciales después de una falla

Luego de que la falla en el transformador (simulada por el IEDRaspBerry) ha sido despejada, es necesario resetear la banderilla de alarma del convertidor IEC61850/Modbus para que el

dispositivo SEL351 regrese a condiciones normales. Para esto se envía un comando False al atributo “ctlVal” del objeto “SPCS01” del convertidor. Tal como se muestra en la figura 131.

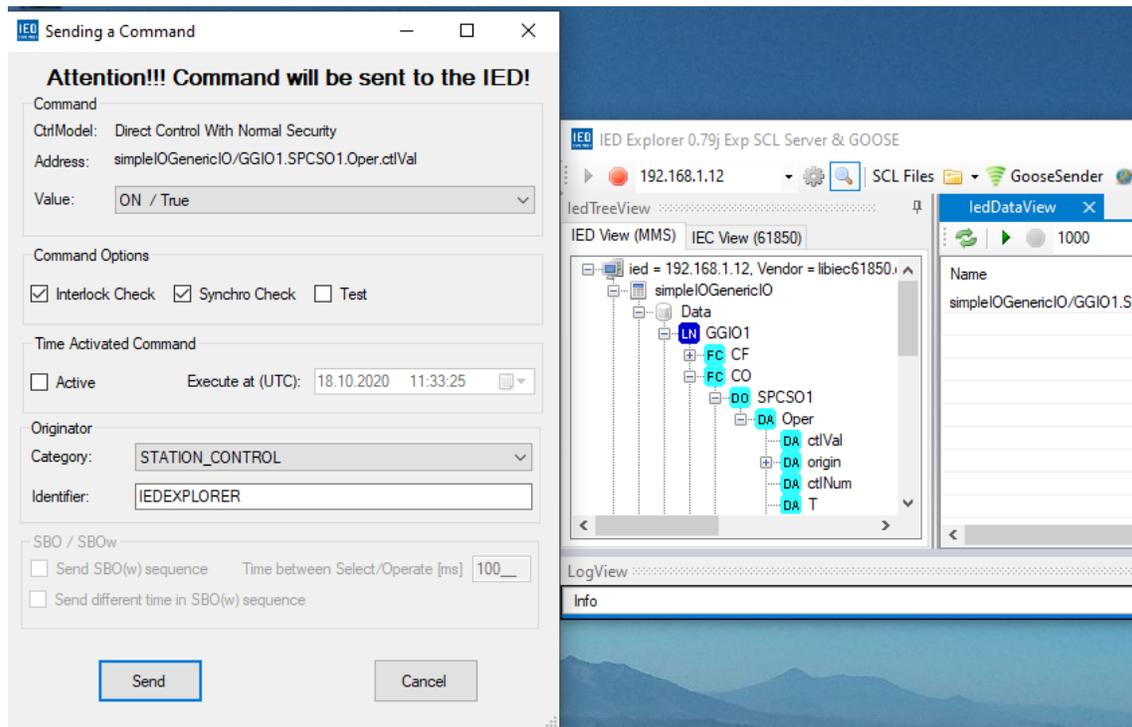


Figura 131. Comando MMS de reseteo en alarma Gateway IEC61850/Modbus.

3.2 Escenario 2, disparo de barra acelerada (protección por sobre corriente instantánea)

El objetivo en este escenario es simular una protección de barra simple por sobre corriente instantánea aplicando mensajería GOOSE, utilizando software de simulación.

3.2.1 Descripción y Funcionamiento

El escenario de prueba se muestra en la figura 132 el cual consiste en la protección de una barra simple por sobre corriente instantánea, para el cual los relés de AxonGroup representa la protección de las bahías y el relé INFOTECH del alimentador principal.

En este escenario se utiliza la mensajería GOOSE para el intercambio de señales de protección en cual el relé INFOTECH está monitoreando el estado de los relés de AxonGroup, de esta manera determinará si la falla es en los alimentadores o en la barra. En este escenario también se utiliza el software Wireshark para el tráfico de mensajes GOOSE.

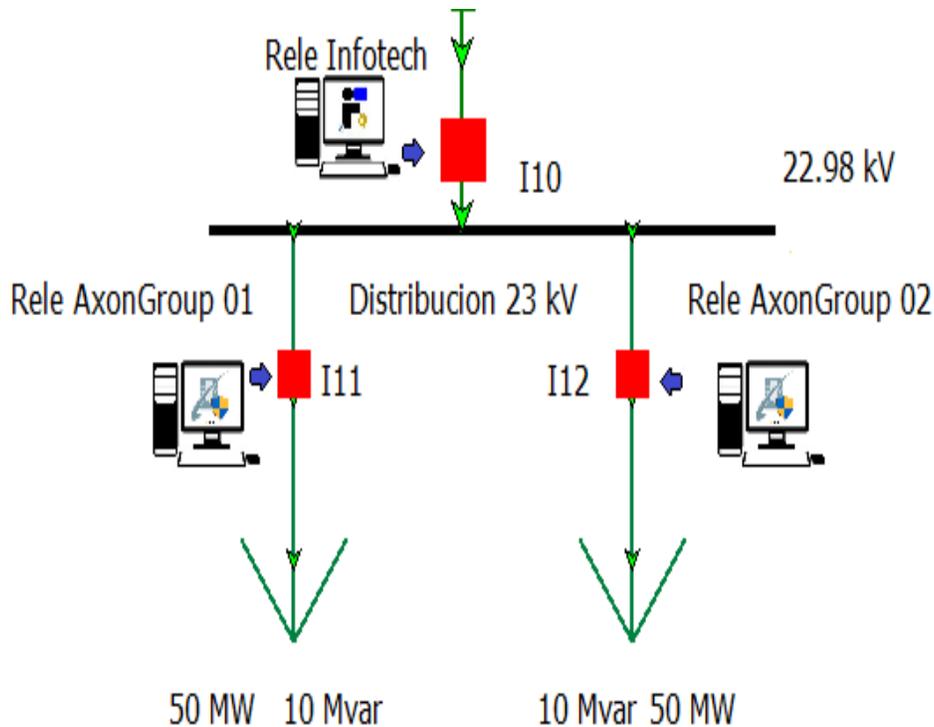


Figura 132. Esquema de Protección de barra simple.

3.2.2 Esquema de protecciones

El esquema de protección que se presenta en este escenario se basa en dos tipos de comportamiento según la falla:

- 1) **Falla externa (bahía):** En el caso que la falla ocurra en los alimentadores de bahía los relés AxonGroup serán los encargados de despejar dicha falla en su circuito, garantizando el suministro de energía en los demás alimentadores, para ello se simula una falla con el software correspondiente.
- 2) **Falla en la barra:** En el caso que la falla ocurra en la barra, la protección encargada de despejar dicha falla será el relé de INFOTECH el cual después de disparar su protección envía un mensaje GOOSE a los relés de AxonGroup, de esta manera dejara despejada completamente la falla en la barra.

3.2.3 Configuración de los Equipos.

En este escenario se configura un mensaje GOOSE para el disparo de los relés de AxonGroup para el caso en que exista una falla en la barra.

Inicialmente se configura el mensaje GOOSE a transmitir (ver la descripción del apartado 2.1.4.1 para la configuración del mensaje GOOSE), el contenido del mensaje para el disparo de los relés de bahía es similar a:

- **AppID:** 1000=4096 hexadecimal
- **CBRef:** "REMOTEProtections/LLN0\$GO\$gcb1"
- **GOOSEID:** "RemoteProtection"
- **DataSet:** "REMOTEProtections/LLN0\$DS1_GOOSE"
- **Tipo de dato:** Booleano
- **Prioridad:** 4

Para monitorear el estado del Simulator Bay de AxonGroup se configura el tag de Pickup del software de esta manera el relé INFOTECH determina cuando existe una falla externa a él, el tag es el siguiente:

Tag Pickup relé Simulator Bay: AXONBAYProtections\$PTOC1\$ST\$Str\$general

3.2.4 Procedimiento en pasos para realizar la prueba

- 1) Conectar los dispositivos al router como muestra en la figura 133.

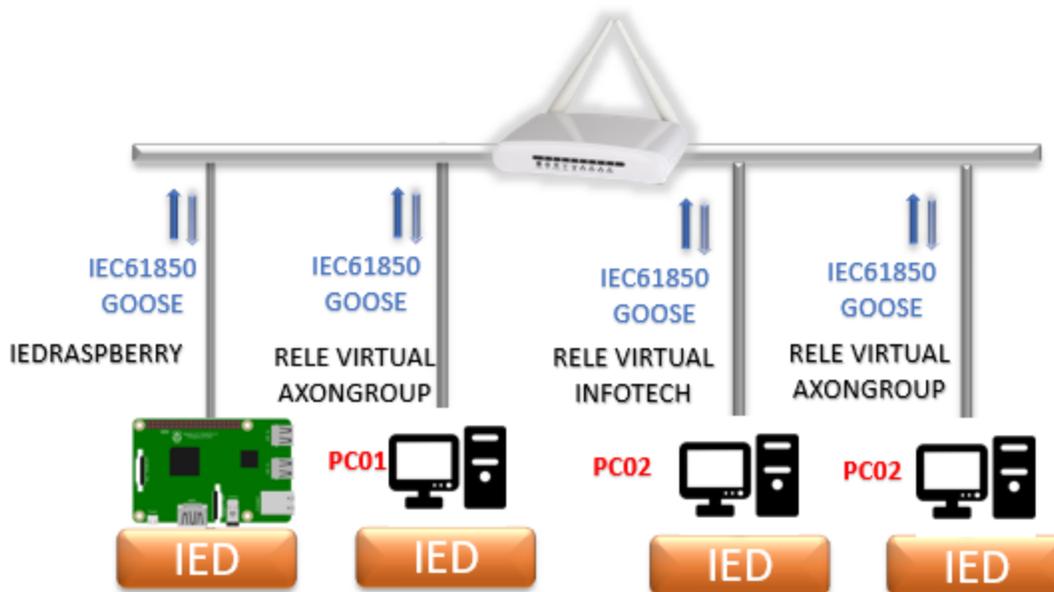


Figura 133. Esquema de comunicación del escenario 2.

- 2) Iniciar el relé virtual de INFOTECH en la PC-02 y verificar mediante el software IEDexplorer instalado en la misma PC que el servidor IEC61850 este inicializado y recibiendo datos GOOSE de la posición del interruptor y el estado de los cinco indicadores del interruptor, también se verifica con la herramienta Goose Explorer que el relé esté conectado a través del adaptador de red Ethernet, el resultado se muestra en la figura 134.

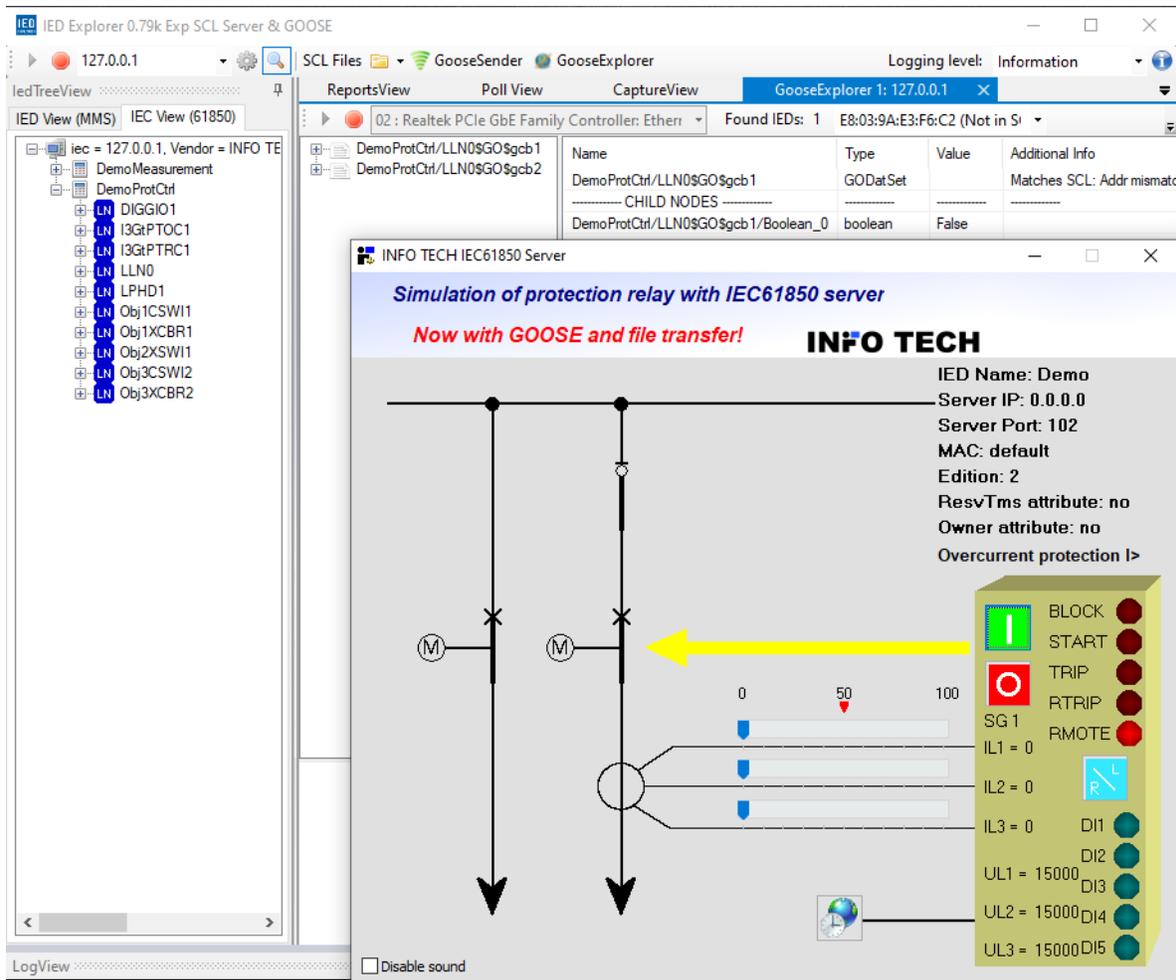


Figura 134. Verificación inicio interruptor INFOTECH.

- 3) Iniciar el relé virtual #1 Simulator Bay en la PC – 01.
- 4) Iniciar el relé virtual #2 de Simulator Bay en la PC- 02, cambiar en el apartado settings a 103 el número de puerto y verificar este seleccionado el adaptador de red ethernet (ver apartado 2.1.2), a continuación, presionar F2 para reiniciar, esto se realiza debido a que el relé virtual de Infotech utiliza el puerto 102 por defecto y existe conflicto entre ambos relés.
- 5) Verificar mediante el software IEDexplorer la conexión de ambos servidores IEC61850 al local host, verificar de igual manera el funcionamiento del relé virtual #1 ingresando la IP de la PC-01. El resultado esperado se muestra en la figura 135.

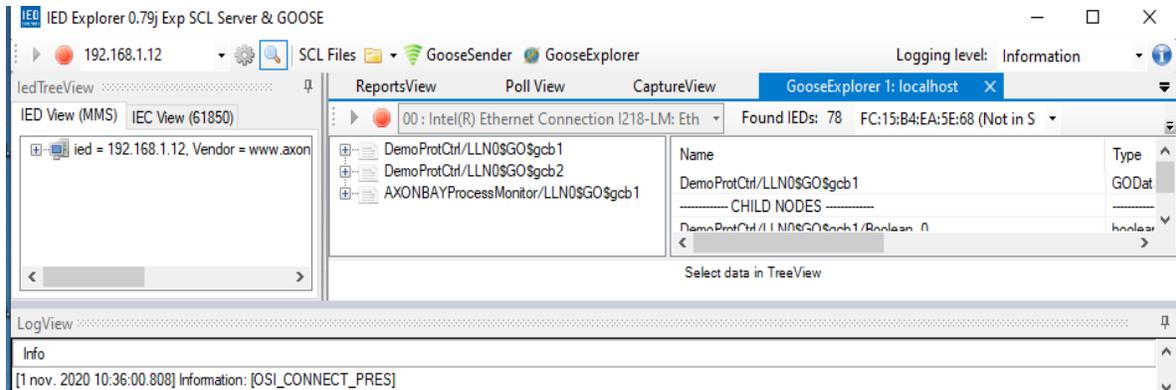


Figura 135. Ejecución de relés virtuales Simulator Bay e Infotech

- 6) Acceder vía VNC viewer al IEDRaspberry, ingresar la dirección IP, el usuario(pi) y la contraseña (1234). Ingresar a una terminal bash y ejecutar el siguiente comando para ubicarse en la carpeta de la aplicación, según se instalo en la sección 1.2.10:

```
$cd /home/pi/libiec61850-1.4/examples/iedraspberry
```

Compilar y ejecutar la aplicación cuyo código fuente se encuentra en la misma carpeta mediante la instrucción:

```
$make; sudo ./server_example_control
```

El dispositivo IEDRaspberry monitorea el estado del relé INFOTECH a través del protocolo GOOSE tal como se muestra en la figura 136, permitiendo realizar el disparo en los relés de bahía al existir una falla en la barra.

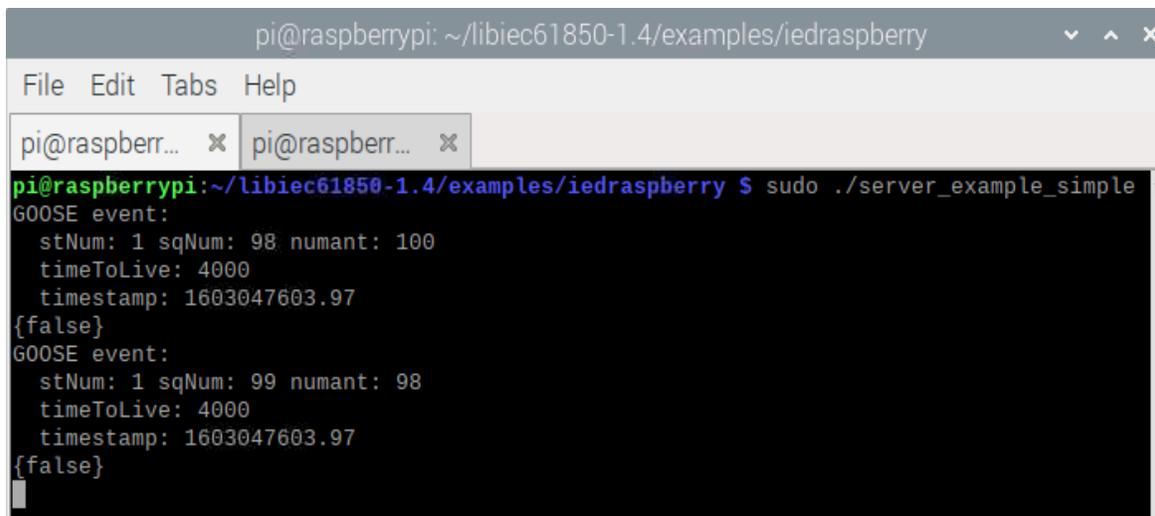


Figura 136. Monitoreo a través de GOOSE del relé INFOTECH.

3.2.5 Evento #1, falla en el alimentador de salida

Al existir una falla a la salida de los alimentadores, los interruptores que tienen que operar son los relés simulados de AxonGroup. El relé INFOTECH solo está informado de lo que sucede aguas abajo pero no efectúa un disparo, los encargados de despejar las fallas son los relés correspondientes al interruptor I11 e I12.

Para simular una falla en los alimentadores de salida, se aumenta la corriente de una de las fases del relé AxonGroup para el ejemplo se aumenta la corriente en fase B, el tag que se está monitoreando indica cuando el pickup del relé ha cambiado de estado tal como se muestra en la figura siguiente.

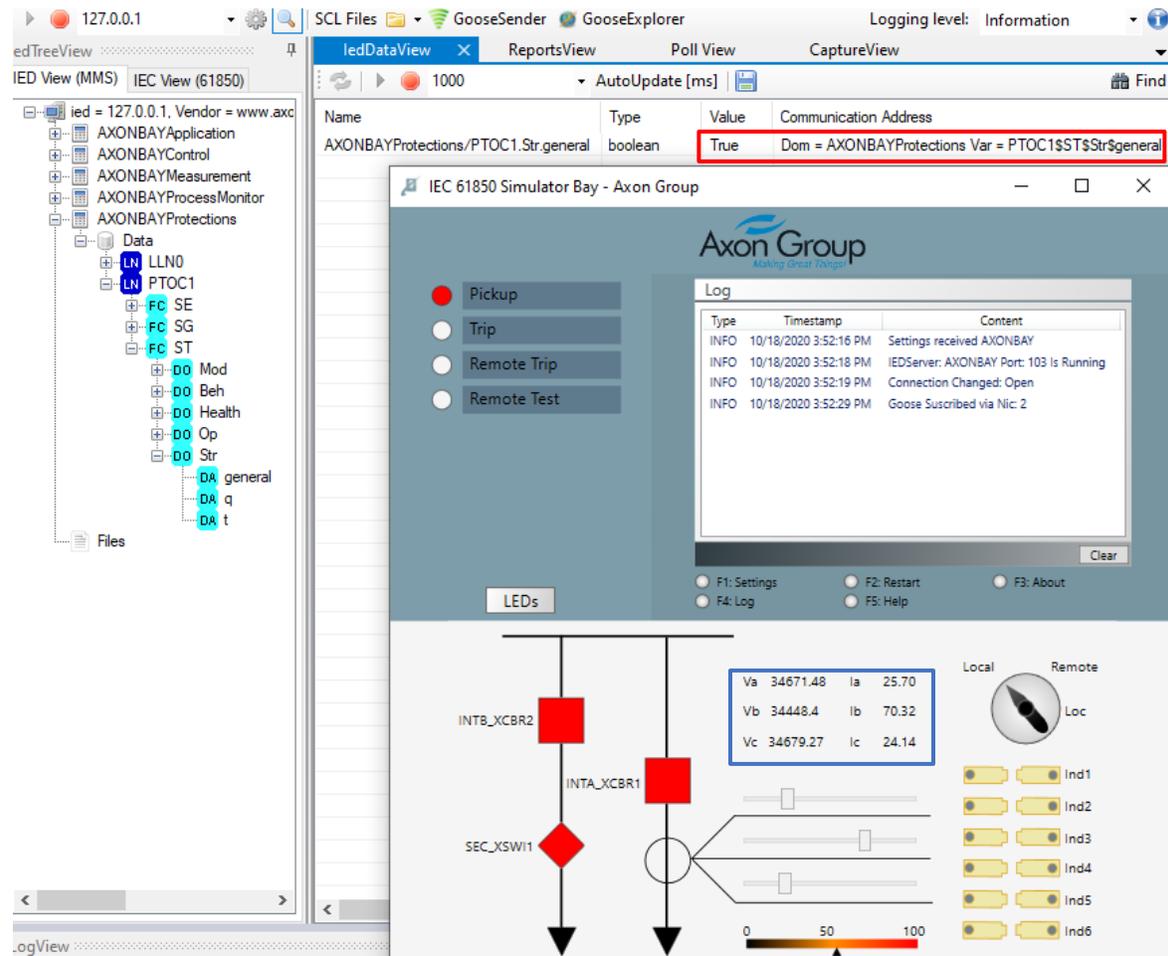


Figura 137. Simulación de una falla en los alimentadores de salida.

El relé AxonGroup también cuenta con reportes que han sido programados por el fabricante, dentro de ellos se encuentran reportes que monitorean el estado del relé, puede visualizarse dichos reportes al habilitarlos, tal como se ha mostrado en el apartado 2.1.1.2. La figura 138 muestra la información del estado Relé AxonGroup a través de reportes.

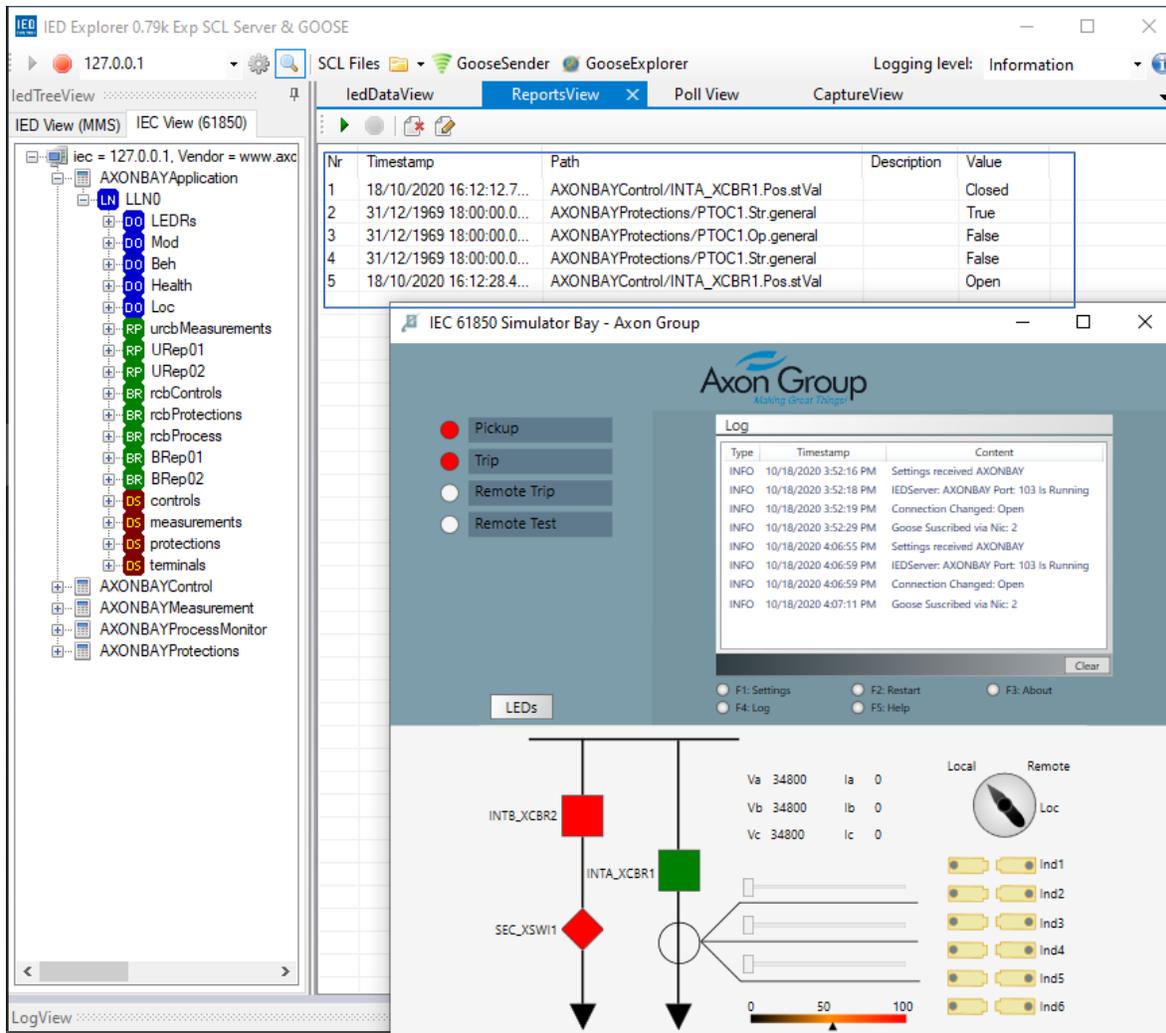


Figura 138. Monitoreo a través de reportes del estado del Relé AxonGroup.

Como se observa en la figura anterior al existir una sobrecorriente en cualquiera de las fases del interruptor (INTA_XCBR1), inmediatamente su protección realiza un disparo al interruptor del relé Axongroup, en la figura 138 se logra apreciar el cambio de estado del interruptor a través de reportes, de esta manera se está simulando una falla en la salida de los interruptores I11 e I12, en caso de falla en cualquiera de los dos se genera la siguiente condición:

- **Falla en I11:**

1. **IED INFOTECH (I10):** Bloqueado (No opera).
2. **IED Simulator Bay AxonGroup 01 (I11):** Activado (Opera despejando la falla).
3. **IED Simulator Bay AxonGroup 02 (I12):** Bloqueado (No opera)

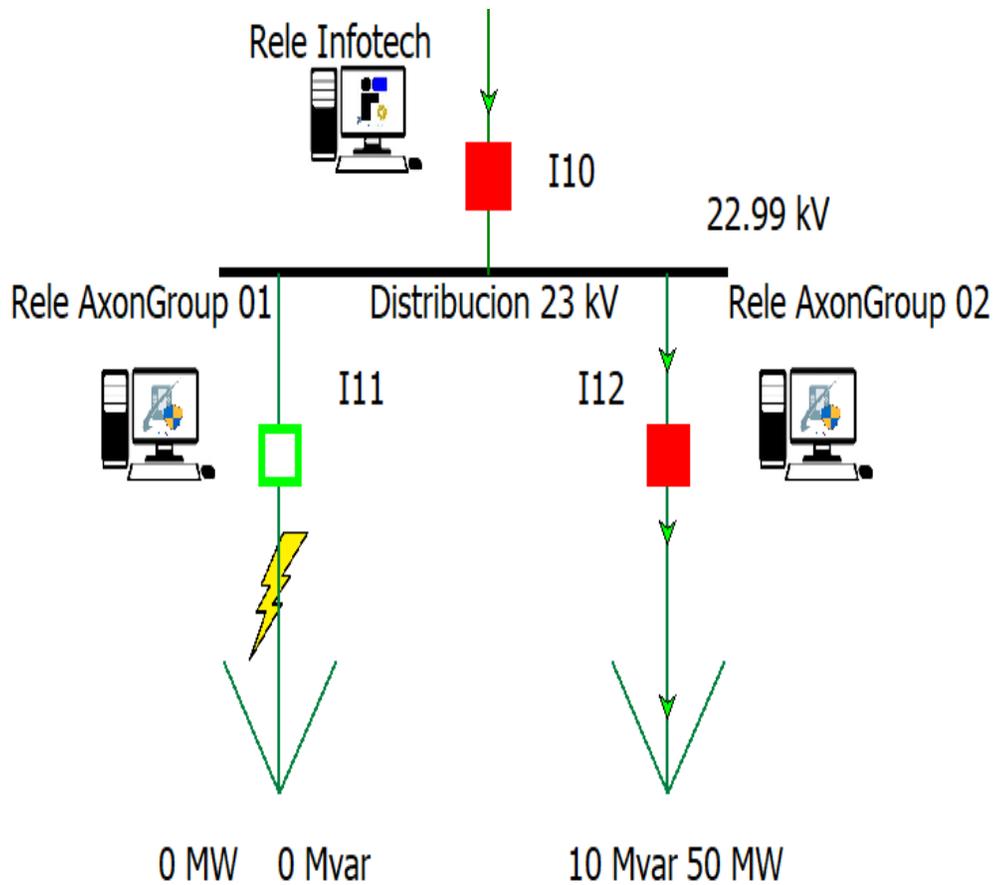


Figura 139. Simulación de una falla a la salida de I11.

Como se muestra en la figura 139, al existir una falla en el alimentador de salida I11, el relé que debe operar para despejar dicha falla es el relé de AxonGroup 01, los relés restantes siguen operando con normalidad.

- **Falla en I12:**

1. **IED INFOTECH (I10):** Bloqueado (No opera).
2. **IED Simulator Bay AxonGroup 01 (I11):** Bloqueado (No opera).
3. **IED Simulator Bay AxonGroup 02 (I12):** Activado (Opera despejando la falla).

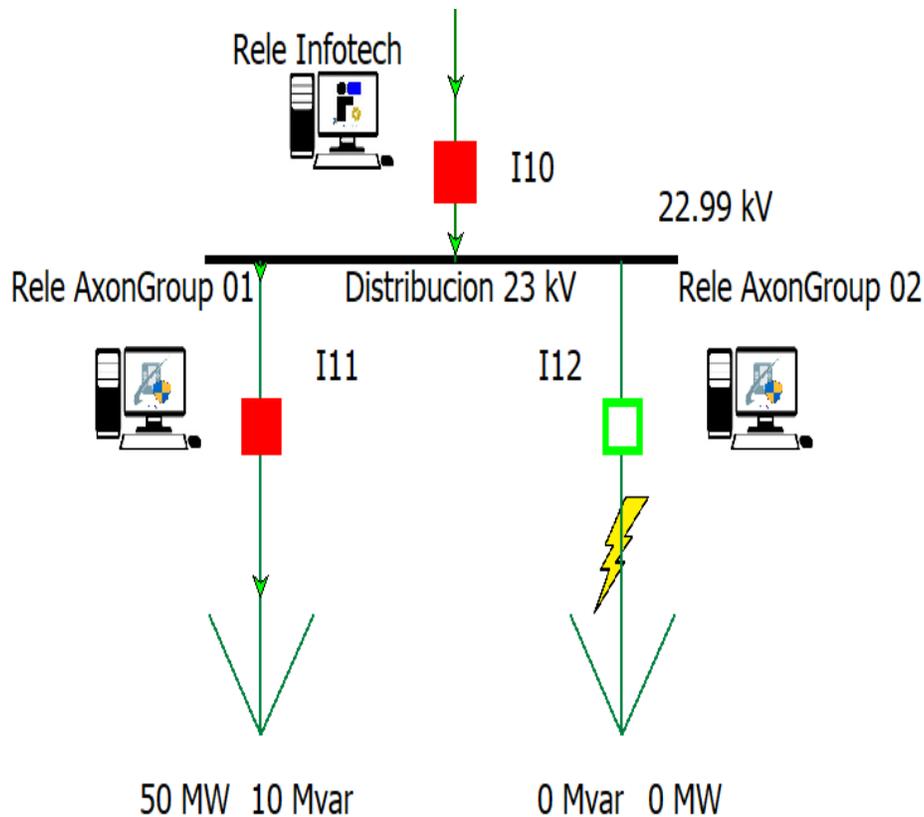


Figura 140. Simulación de una falla a la salida de I12.

Como se muestra en la figura 140, al existir una falla en el alimentador de salida I12, el relé que debe operar para despejar dicha falla será el relé de AxonGroup 02, los relés restantes seguirán operando con normalidad.

3.2.6 Evento #2, falla en la barra

Al ocurrir una falla en la barra (alimentador de entrada I10), el relé encargado de despejar dicha falla es el relé INFOTECH, luego de efectuar su apertura este envía inmediatamente un mensaje GOOSE a los relés de AxonGroup para despejar totalmente la falla en la barra.

Para simular una falla en la barra se aumenta la corriente de cualquiera de las fases del interruptor del relé INFOTECH. Para el ejemplo se aumenta la corriente en la fase A, tal como se muestra en la figura 141.

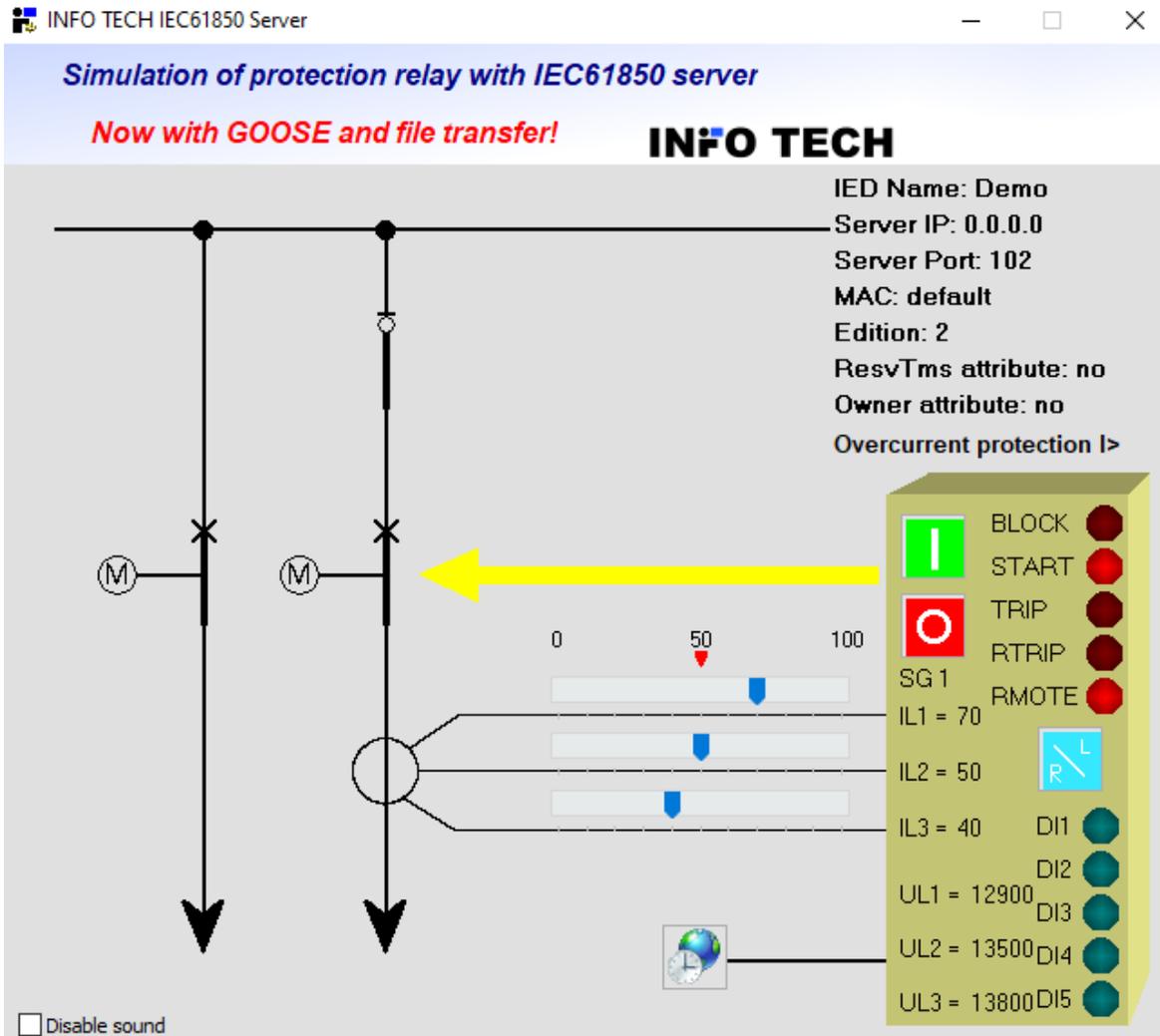


Figura 141. Simulación de una falla en la barra utilizando el relé INFOTECH.

Al aumentar la corriente se inicia el tiempo de disparo de la protección del relé, el disparo del TRIP del relé INFOTECH está siendo monitoreado por el IEDRaspberry (ver figura 136), cuando se detecte un cambio en el estado del relé a través de mensajes GOOSE el IEDRaspberry enviará un nuevo mensaje GOOSE que provocará el disparo inmediato de los relés de bahía (Simulator Bay AxonGroup).

La figura 142 muestra el cambio del TRIP a través de mensaje GOOSE del relé INFOTECH. Este genera un cambio inmediato por el disparo del relé (TRUE), pero luego de haber operado regresa a su normalidad (FALSE).

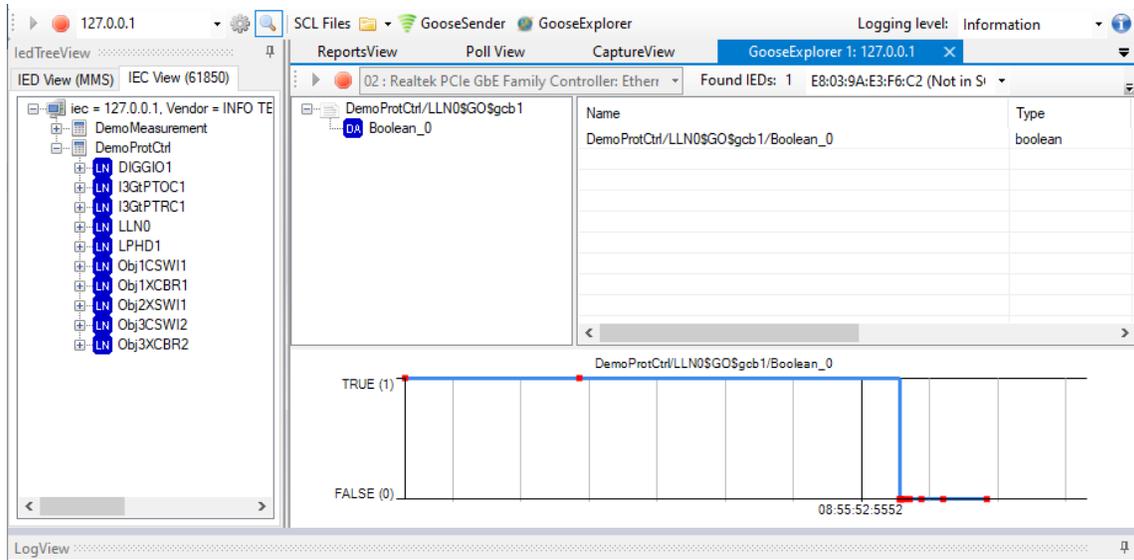


Figura 142. Mensaje GOOSE para el cambio de estado del Relé INFOTECH.

Si se presenta una falla en la barra (alimentador de entrada I10) se genera la siguiente condición:

- 1) **IED INFOTECH (I10):** Activado (Opera despejando la falla, adicionalmente envía un GOOSE hacia los relés de AxonGroup).
- 2) **IED Simulator Bay AxonGroup 01 (I11):** Opera a través de protocolo GOOSE.
- 3) **IED Simulator Bay AxonGroup 02 (I12):** Opera a través de protocolo GOOSE

La figura 143 muestra el escenario de operación de los relés cuando existe una falla en la barra.

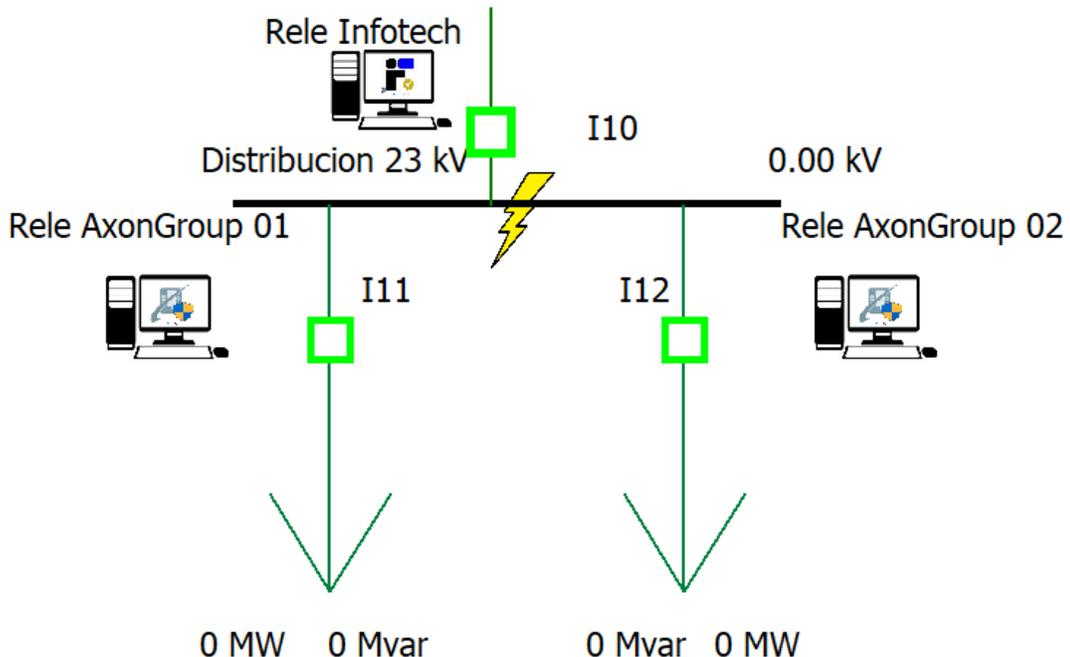


Figura 143. Simulación de una falla en la barra (alimentador de entrada I10).

En el esquema de la figura anterior el relé que opera inicialmente es el relé INFOTECH, el IEDRaspberry responde enviando un mensaje GOOSE para que las protecciones de los alimentadores de salida también operen tal como se muestra en las figuras 144 y 145.

```

pi@raspberrypi: ~/libiec61850-1.4/examples/iedraspberry
File Edit Tabs Help
pi@raspberr... x pi@raspberr... x
GOOSE event:
  stNum: 2 sqNum: 6 numant: 5
  timeToLive: 128
  timestamp: 1603048580.697
{true}
GOOSE event:
  stNum: 2 sqNum: 7 numant: 6
  timeToLive: 256
  timestamp: 1603048580.697
{true}
GOOSE event:
  stNum: 2 sqNum: 8 numant: 7
  timeToLive: 512
  timestamp: 1603048580.697
{true}
GOOSE event:
  stNum: 2 sqNum: 9 numant: 8
  timeToLive: 1024
  timestamp: 1603048580.697
{true}

```

Figura 144. Envío de mensaje GOOSE del IEDRaspberry a los relés de bahía

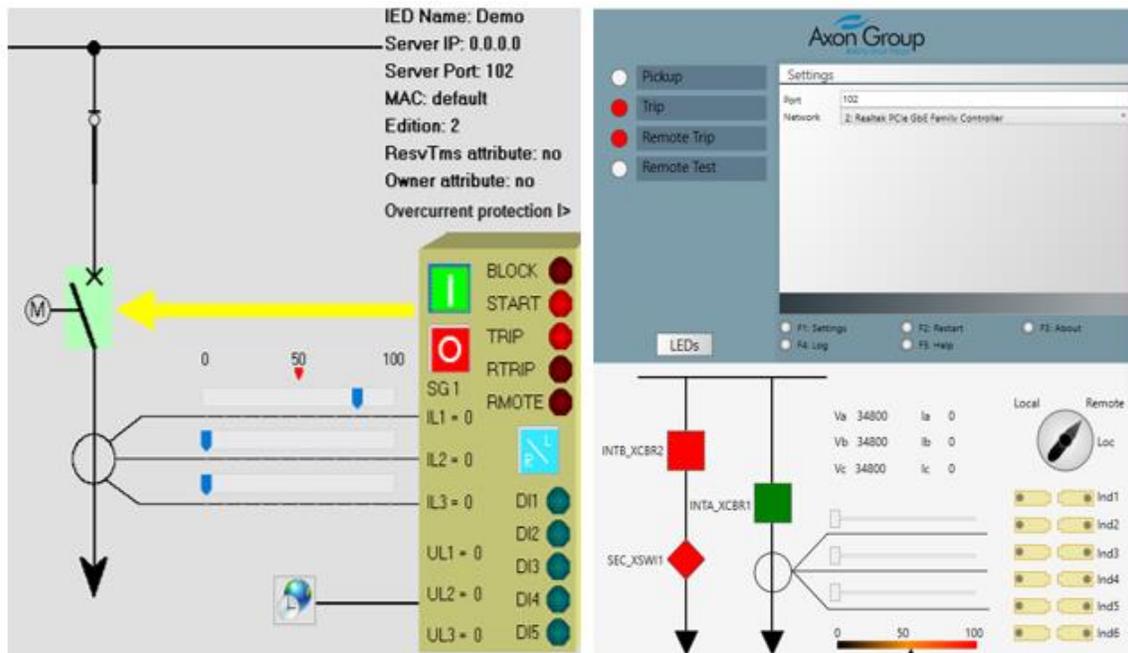


Figura 145. Disparo de los relés de bahía a través de protocolo GOOSE.

En la figura 146 se muestra el interruptor INTA_XCBR1 del relé SimulatorBay de AxonGroup el cual ha sido disparado por un mensaje GOOSE (Remote Trip).

No.	Time	Source	Destination	Protocol	Length	Info
4807	5.504308	Raspberr_a6:e4:4c	Iec-Tc57_01:00:01	GOOSE	146	
4808	5.505470	Raspberr_a6:e4:4c	Iec-Tc57_01:00:01	GOOSE	146	
48...	5.505875	HewlettP_ea:5e:68	Iec-Tc57_01:00:00	GOOSE	124	
4810	5.505884	HewlettP_ea:5e:68	Iec-Tc57_01:00:00	GOOSE	124	

```

<
> Frame 4809: 124 bytes on wire (992 bits), 124 bytes captured (992 bits) on interface 0
< Ethernet II, Src: HewlettP_ea:5e:68 (fc:15:b4:ea:5e:68), Dst: Iec-Tc57_01:00:00 (01:0c:cd:01:00:00)
  > Destination: Iec-Tc57_01:00:00 (01:0c:cd:01:00:00)
  > Source: HewlettP_ea:5e:68 (fc:15:b4:ea:5e:68)
  Type: 802.1Q Virtual LAN (0x8100)
< 802.1Q Virtual LAN, PRI: 4, DEI: 0, ID: 0
  100. .... .... = Priority: Video, < 100ms latency and jitter (4)
  ...0 .... .... = DEI: Ineligible
  .... 0000 0000 0000 = ID: 0
  Type: IEC 61850/GOOSE (0x88b8)
< GOOSE
  APPID: 0x0001 (1)
  Length: 106
  Reserved 1: 0x0000 (0)
  Reserved 2: 0x0000 (0)
  < goosePdu
    gocbRef: DemoProtCtrl/LLN0$G0$gcb1
    timeAllowedtoLive: 8
    datSet: DemoProtCtrl/LLN0$D53_GOOSE
    goID: G1
    t: Oct 18, 2020 19:07:25.789499998 UTC
    stNum: 2
    sqNum: 2
    test: False
    confRev: 1
    ndsCom: False
    numDatSetEntries: 1
  < allData: 1 item
    < Data: boolean (3)
  
```

Figura 146. Mensaje GOOSE enviado por la IEDRaspberry

La figura 146 también muestra la información del mensaje GOOSE utilizando el software Wireshark, en el cual es posible distinguir los siguientes parámetros:

- **Fuente** (Raspberr_a6:e4:4c) y **Destino** (Iec-Tc57_01:00:01) del mensaje GOOSE, también es posible observar la MAC del **publicador** (b8:27:eb:e6:e4:4c) y del **suscriptor** (01:0c:cd:01:00:01) y el tipo de mensaje (GOOSE) e Ethertype (0x88b8).


```

< Ethernet II, Src: HewlettP_ea:5e:68 (fc:15:b4:ea:5e:68), Dst: Iec-Tc57_01:00:00 (01:0c:cd:01:00:00)
  > Destination: Iec-Tc57_01:00:00 (01:0c:cd:01:00:00)
  > Source: HewlettP_ea:5e:68 (fc:15:b4:ea:5e:68)
  Type: 802.1Q Virtual LAN (0x8100)
      
```
- Propiedades de la VLAN: **Prioridad** (4) definido por la norma e **ID** de la VLAN (0).


```

< 802.1Q Virtual LAN, PRI: 4, DEI: 0, ID: 0
  100. .... .... = Priority: Video, < 100ms latency and jitter (4)
  ...0 .... .... = DEI: Ineligible
  .... 0000 0000 0000 = ID: 0
  Type: IEC 61850/GOOSE (0x88b8)
      
```

- Contenido del mensaje GOOSE: **APPID**, **GoRef**, **DataSet**, **GoID**, estampa de tiempo (**t**), cambios de estado en el DS (**stNum**), veces en que se ha enviado el mensaje GOOSE (**SqNum**), modo de prueba del IED (**test**) y el estado del dato del mensaje GOOSE booleano (**Data**)

```

  v GOOSE
    APPID: 0x1000 (4096)
    Length: 132
    Reserved 1: 0x0000 (0)
    Reserved 2: 0x0000 (0)
    v goosePdu
      gocbRef: REMOTEProtections/LLN0$G0$gcb1
      timeAllowedtoLive: 0
      datSet: REMOTEProtections/LLN0$DS1_GOOSE
      goID: RemoteProtection
      t: Oct 18, 2020 19:02:55.905999958 UTC
      stNum: 1
      sqNum: 232308
      test: False
      confRev: 1
      ndsCom: False
      numDatSetEntries: 1
      v allData: 1 item
        v Data: boolean (3)
          boolean: True

```

3.3 Escenario 3, simulación de un SAS en niveles de estación y bahía

Este escenario tiene como objetivo establecer una comunicación entre los niveles de estación (SCADA) y bahía (IED's) bajo el estándar IEC61850 en un entorno simulado.

3.3.1 Descripción y funcionamiento

Para la comunicación de los dispositivos mostrados en la figura 147 de este escenario se crea una red LAN de tipo estrella, la figura 147 muestra la conexión de cada uno de ellos y los protocolos que interactúan en cada nivel (GOOSE, MMS, Modbus), a través del software Elipse SCADA se supervisa y controla los dispositivos ubicados en el nivel bahía, en este escenario se presenta de manera esquemática los escenarios mostrados en los apartados anteriores, por lo tanto es posible visualizar los disparos de los interruptores y alarmas ocasionados por protecciones de sobre corriente y protecciones del transformador.

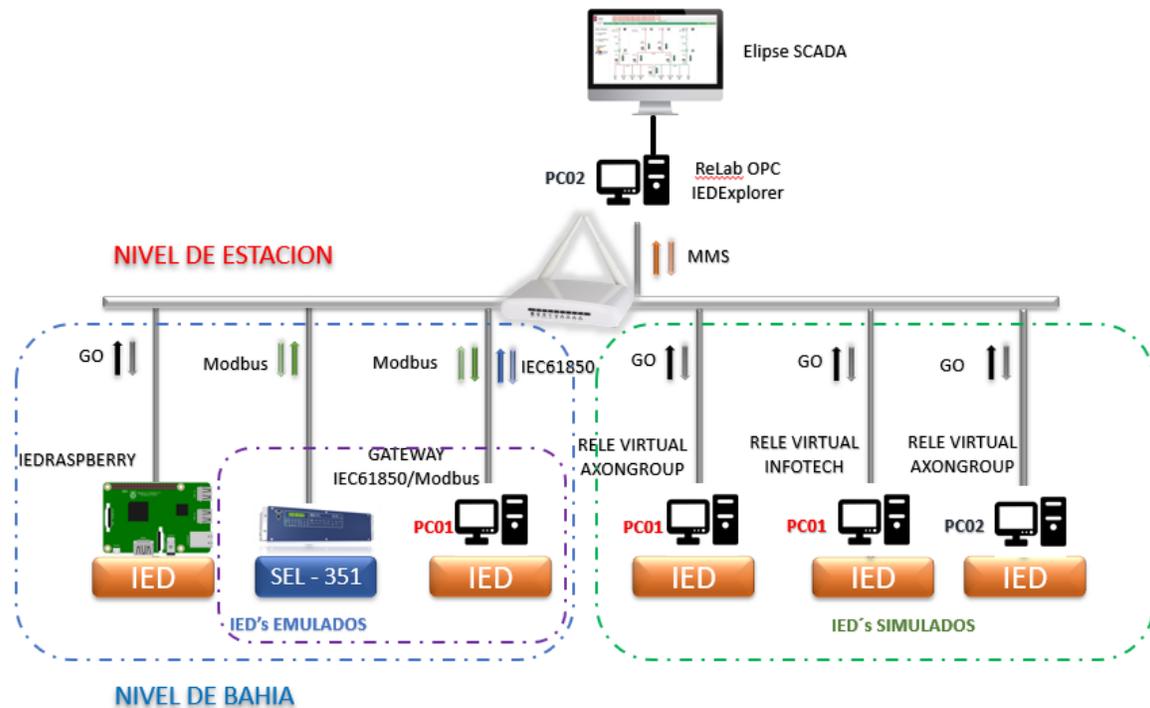


Figura 147. Simulación de un SAS niveles de Estación y Bahía.

El router utilizado en este escenario posee únicamente cuatro puertos ethernet por lo que dificulta la conexión de más dispositivos, por tanto, se emplean dos máquinas físicas (PC01 y PC02) en el cual se ejecutan varios programas a la vez, los puertos restantes se utilizan para la conexión del IEDRaspberry y el relé SEL351.

En la figura 148 se muestra el modelo eléctrico del SCADA, el interruptor I81 representa al relé SEL351, I10 representa al relé INFOTECH y los interruptores I11 e I12 representan a los relés de AxonGroup. El transformador posee una barra con indicadores que representan sus protecciones (relé térmico, relé presión, relé Buchholz y relé de nivel de aceite) las cuales se activan desde el IEDRaspberry.

También se incluye un panel de alarmas en la parte inferior que muestra los disparos de los interruptores por sobre corriente y la activación de las protecciones del transformador, la comunicación a nivel de SCADA se realiza bajo el protocolo Cliente/Servidor MMS.

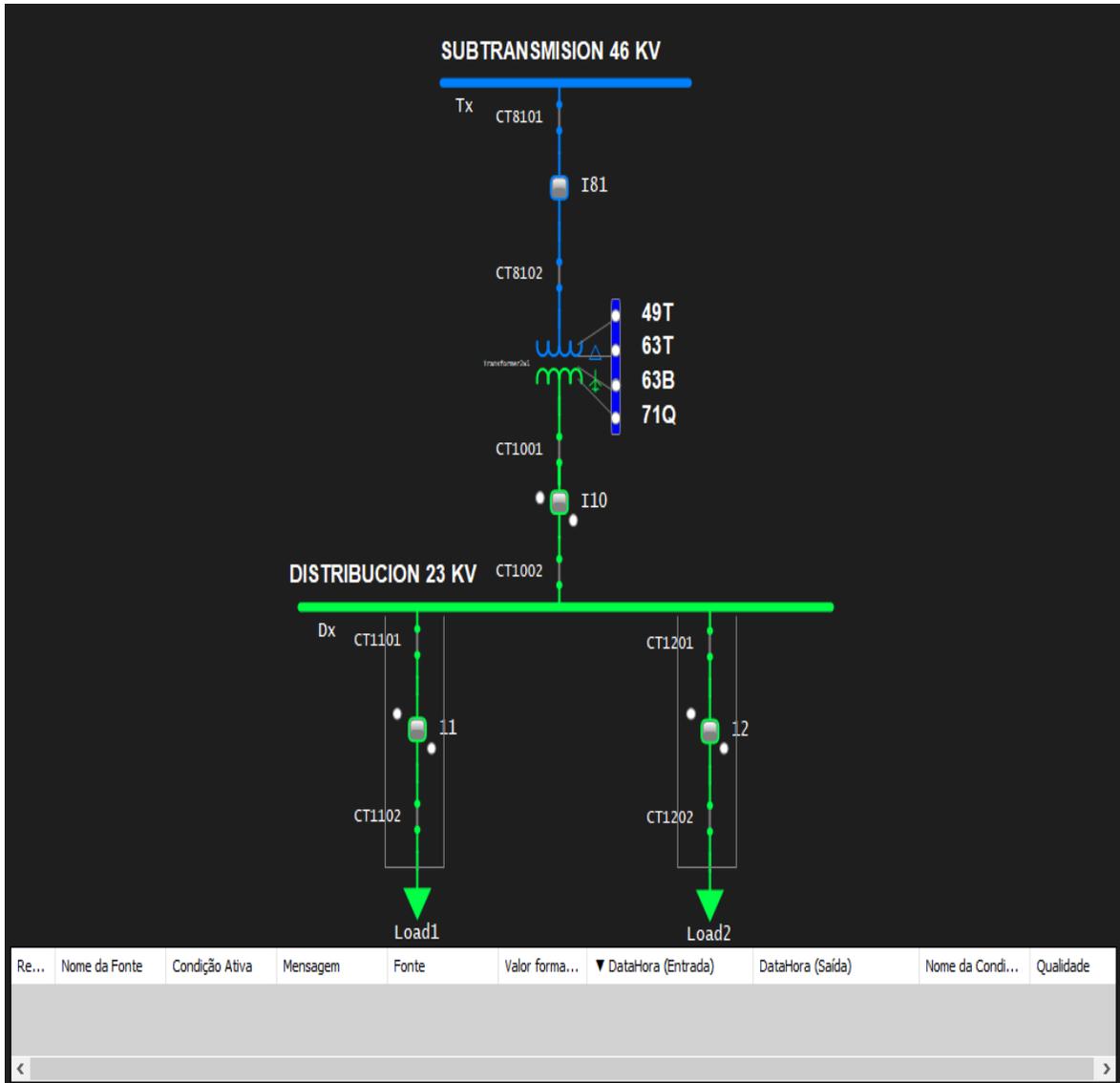


Figura 148. Modelo eléctrico en SCADA

3.3.2 Procedimiento descrito en pasos para realizar la prueba

- 1) Realizar una red LAN con los dispositivos emulados y simulados tal como se muestra en la figura 147, el **IEDRaspberry** y el **relé SEL351** se conectan directamente a un puerto ethernet cada uno, en los puertos restantes se conectan dos máquinas físicas (PC01 y PC02) donde se ejecutan los distintos softwares de simulación los cuales se distribuyen de la siguiente manera:
 - i. PC01: **Máquina Virtual** (se ejecuta el Gateway IEC61850/Modbus), **relé INFOTECH** (puerto 102) y **relé AxonGroup 1** (puerto 103).

- ii. PC02: **Relé AxonGroup 2** (puerto 102), **IEDEplorer**, **ReLab OPC y SCADA**.
- 2) En el software Relab OPC se configura los drivers asignando la dirección IP y puertos de los dispositivos emulados y simulados (ver apartado 1.2.2). El direccionamiento IP de cada dispositivo se realiza automáticamente por DHCP.
 - 3) Mapear en el software Relab OPC los tags que se utilizan en el SCADA (ver tabla 8), los cuales se pueden visualizar en el dashboard del software tal como se muestra en la figura 103 del capítulo 2.
 - 4) En el software elipse SCADA se realiza la comunicación entre Relab OPC y un driver OPC del SCADA (ver apartado 2.3).
 - 5) Se configura los tags de los dispositivos emulados y simulados, asignándolos a cada elemento del modelo eléctrico del SCADA (ver apartado 2.3.1.1).
 - 6) Se establece comunicación con el Gateway IEC61850 a través de IEDEplorer para el control del disparo del relé SEL351 activado por las protecciones del transformador, lo que permite regresar a condiciones normales el sistema de subestación simulado (ver apartado 3.1.6).
 - 7) Se ejecuta el dominio del SCADA, permitiendo supervisar los IED's que se encuentra en el nivel de bahía, cuando se establece la comunicación con los IED's el SCADA muestra el estado de cada interruptor, alarmas, indicadores y corrientes de la subestación simulada tal como se muestra en la figura 149.

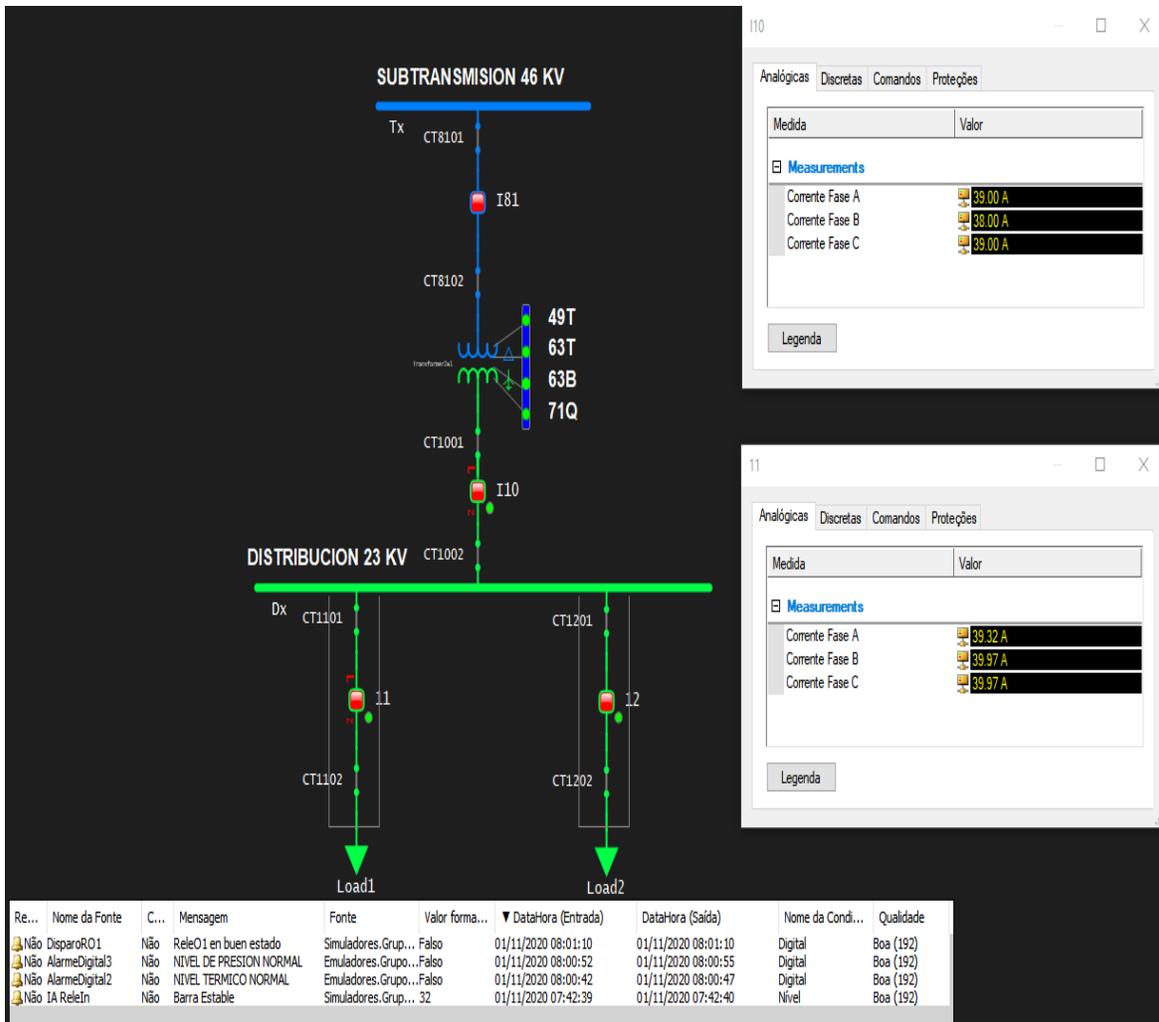


Figura 149. SCADA de la subestación de barra simple

El sistema SCADA mostrado en la figura 149 corresponde a una subestación de barra simple y emplea los escenarios mostrados en los apartados 3.1 y 3.2, a continuación, se presentan los resultados de las pruebas de cada uno de los escenarios utilizando el software Elipse SCADA.

3.3.3 Evento #1, protecciones mecánicas de un transformador de potencia

Como se ha descrito en el apartado 3.1, las actuaciones de las protecciones del transformador se realizan a través del IEDRaspberry, en el sistema SCADA es posible observar cuál de las protecciones fue activada con su respectiva alarma y disparos de los interruptores correspondientes, tal como se muestra en la figura 150.

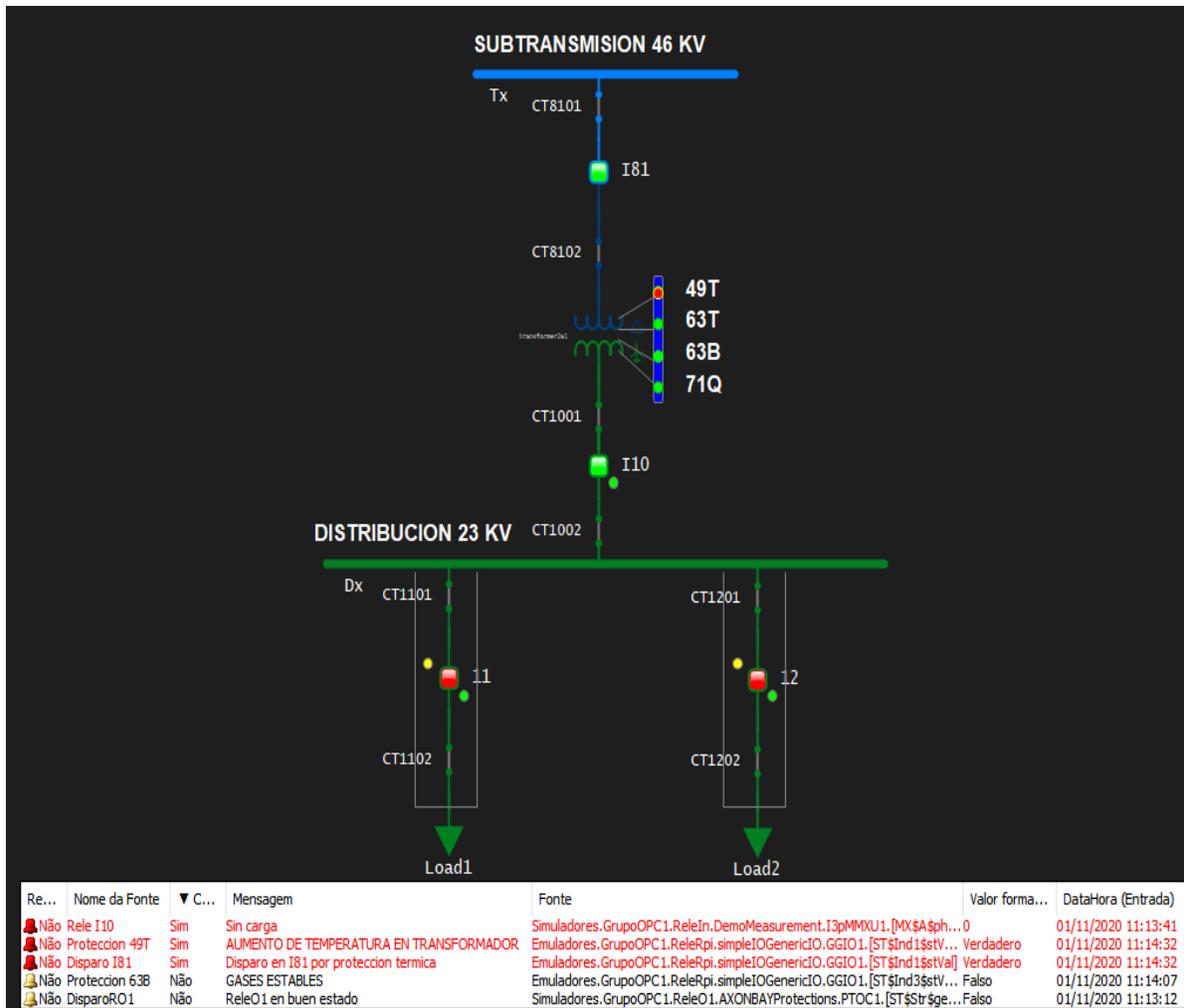


Figura 150. Protecciones mecánicas del transformador supervisado a través de SCADA.

Como se observa en la figura anterior la protección de temperatura (49T) ha sido activada, esto ocasiona un disparo en los interruptores I81 e I10 el cual cumple con el funcionamiento de las protecciones descrito en el apartado 3.1.

En el panel de alarmas es posible observar las alarmas que se han activado al ocurrir este evento, las cuales se describen a continuación:

- Relé I10: Esta alarma se activa al no detectar carga en el interruptor I10.
- Protección 49T: Indica cual ha sido la protección y el motivo por el cual ha sido activada (Aumento de temperatura en el transformador).
- Disparo I81: Esta alarma indica el motivo por el cual se ha disparado el I81.

También es posible observar en el panel de alarmas el tag correspondiente de cada una de ellas, el valor que poseen y la estampa de tiempo.

Los interruptores I11 e I12 se mantienen activos con un indicador en la parte superior (punto amarillo), que indica que el interruptor I10 está disparado. También es posible observar en el SCADA el flujo de energía.

Este mismo evento ocurrirá al activarse cualquiera de las protecciones del transformador, en la figura siguiente se muestra el caso para la protección del relé Buchholz (aumento de gases en el transformador) en el cual es posible identificar el motivo por cual se ha disparo el interruptor I81.

Para regresar el sistema de la subestación a condiciones normales se ejecutan los pasos mostrados en el apartado 3.1.6 desde el IEDExplorer.

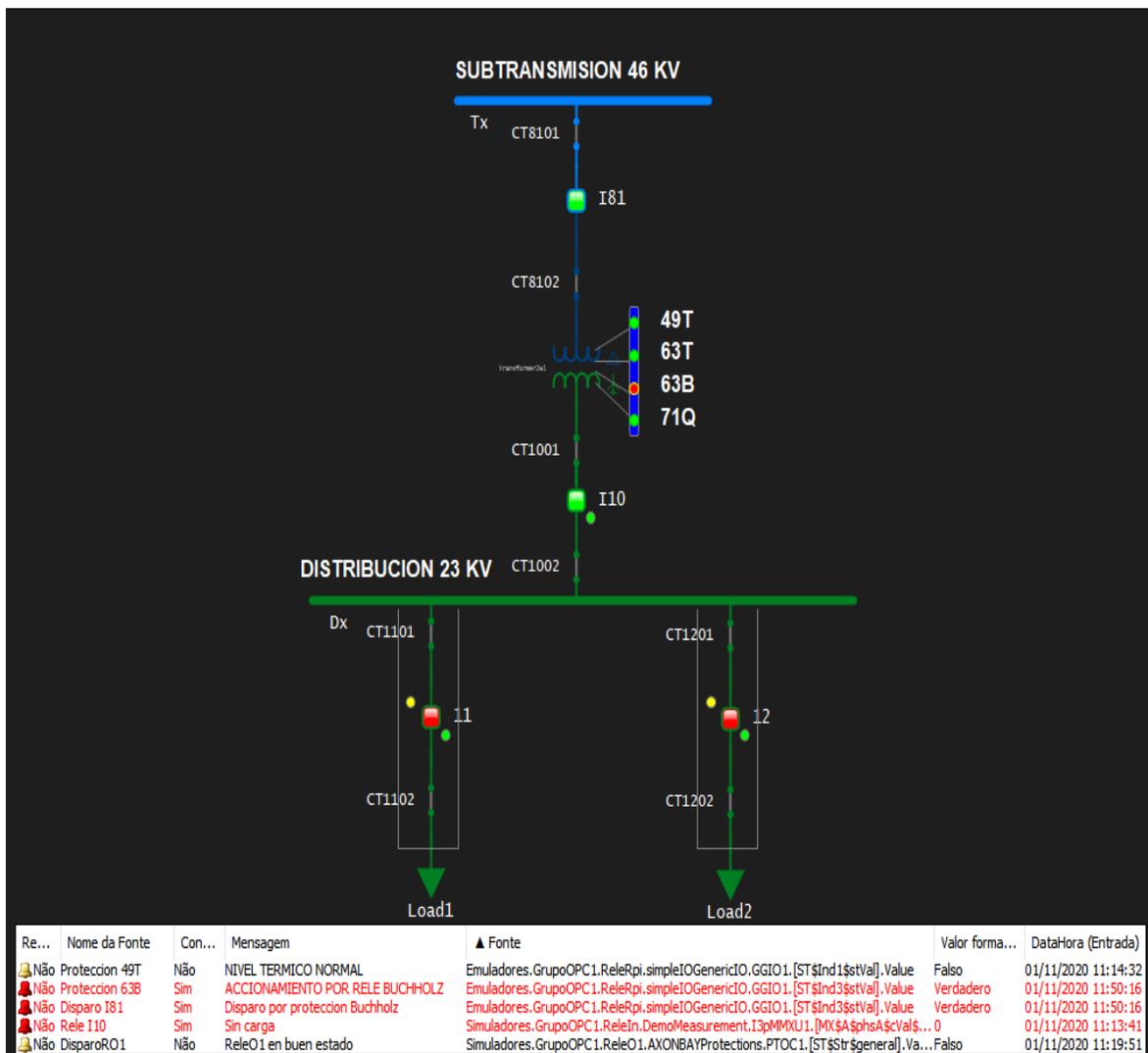


Figura 151. Protección del transformador 63B supervisado a través de SCADA

3.3.4 Evento #2, protección de barra acelerada

La protección de barra acelerada o por sobre corriente instantánea se ha descrito en el apartado 3.2, su funcionamiento se visualiza mejor desde el SCADA donde se muestra los eventos descritos en los apartados 3.2.5 (Protección alimentador de salida) y 3.2.6 (Protección alimentador de entrada), en la figura siguiente se muestra el evento cuando existe una falla en la salida del alimentador I11.

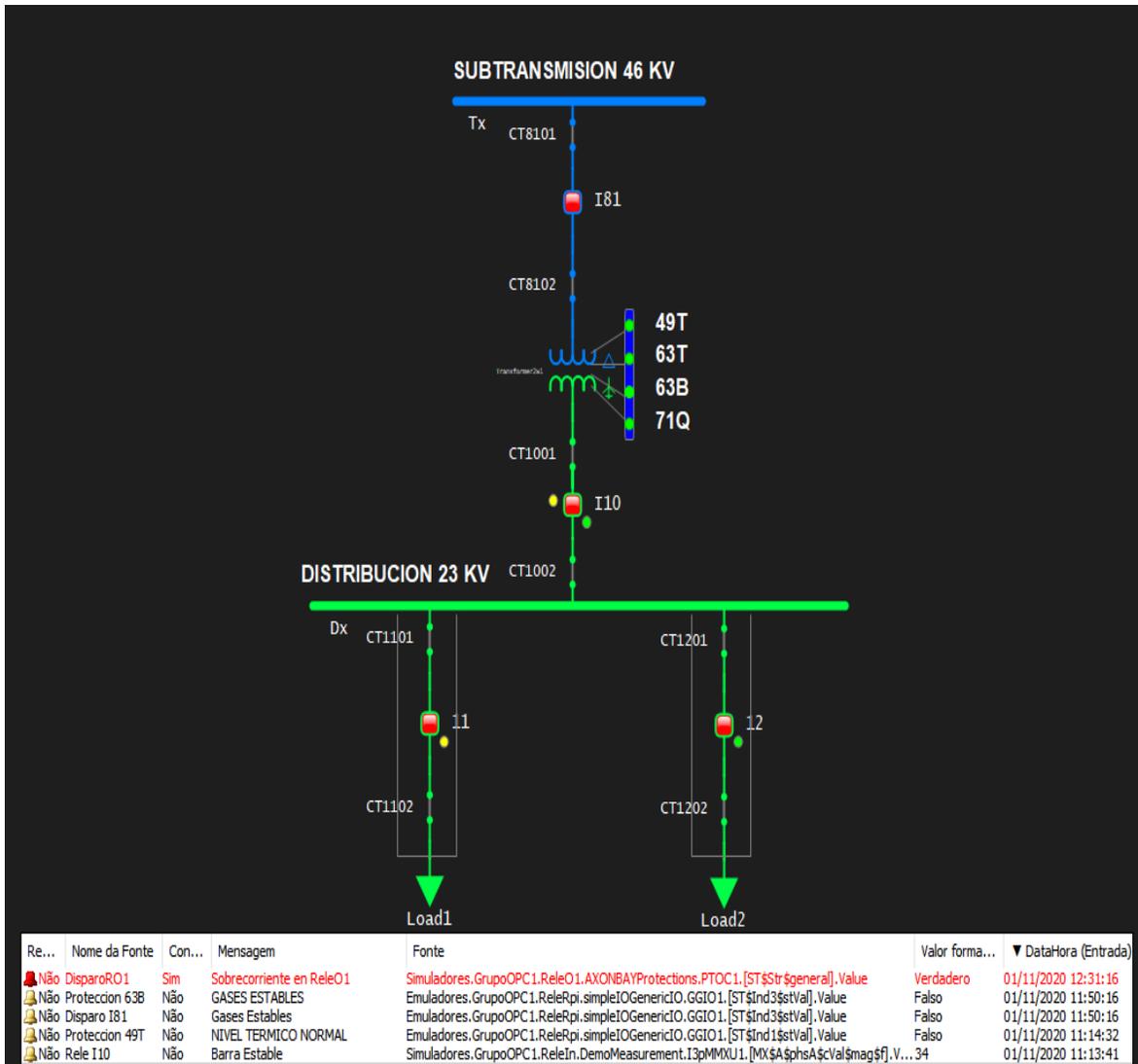


Figura 152. Condiciones de sobre corriente en I11

Al existir una sobre corriente en I11 (simulado con relé AxonGroup), se activa una alarma indicando que el estado de I11 (ReleO1) se encuentra en condiciones de sobre corriente, el interruptor I10 detectara la falla, pero no efectuara un disparo, en su lugar solo muestra un

indicador de color amarillo en la parte superior, lo que indica que la falla se encuentra en uno de los alimentadores de salida.

La figura 153 muestra las condiciones después de que el relé AxonGroup ha efectuado el disparo, en el panel inferior se activa una nueva alarma indicando que I11 está sin carga, el flujo de energía representado en el SCADA se muestra con un color menos intenso a la salida del interruptor I11 lo que indica que el interruptor está abierto.

Las condiciones que se han descrito en los párrafos anteriores se repiten para el caso en el que exista una sobre corriente en I12.

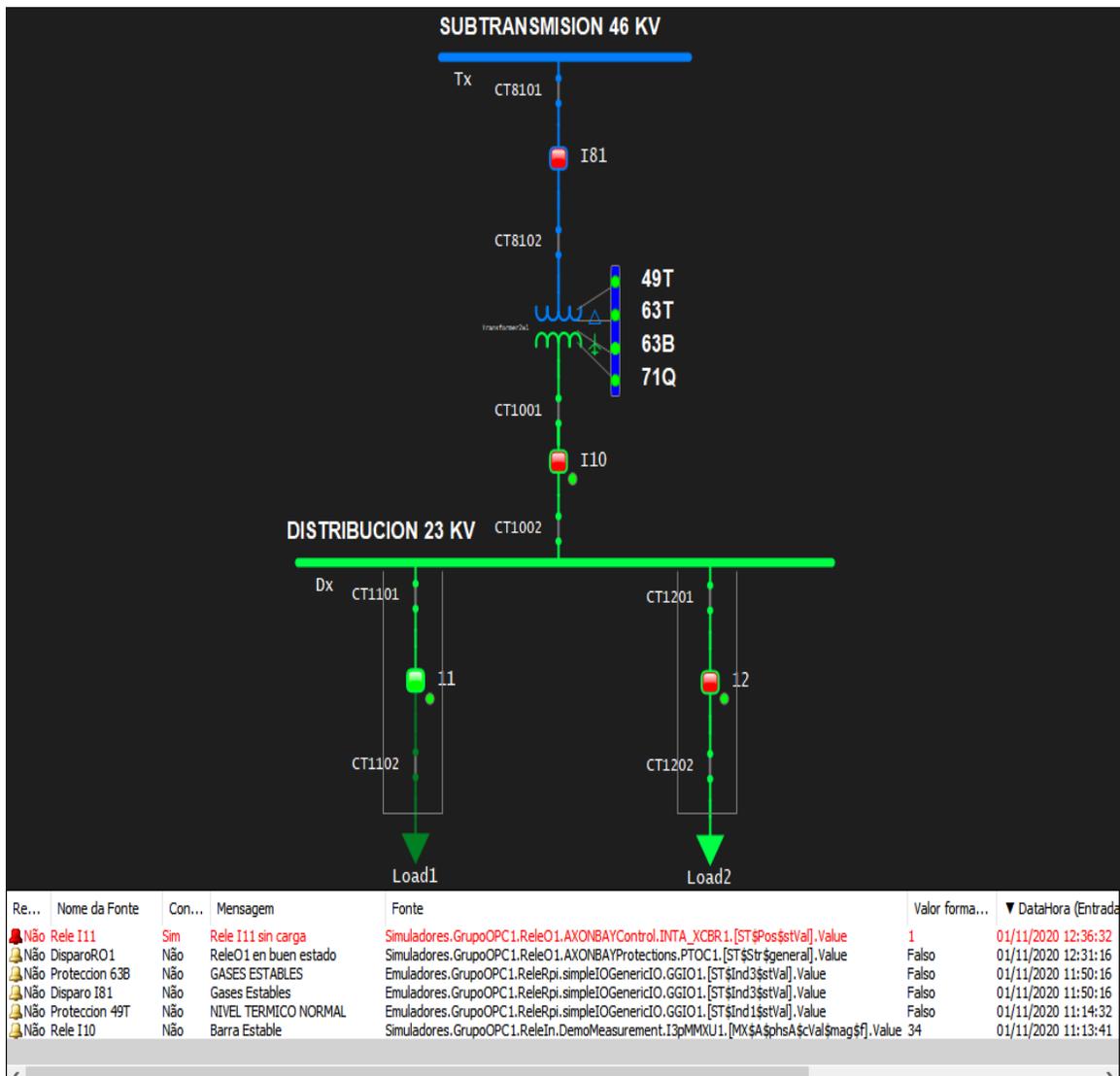


Figura 153. Interruptor I11 disparado por sobre corriente instantánea

Cuando existe una sobre corriente en la barra, la protección del interruptor I10 actúa primero, instantáneamente envía un mensaje GOOSE a los interruptores de los alimentadores de salida (ver apartado 3.2). La figura 154 muestra este evento en el SCADA.

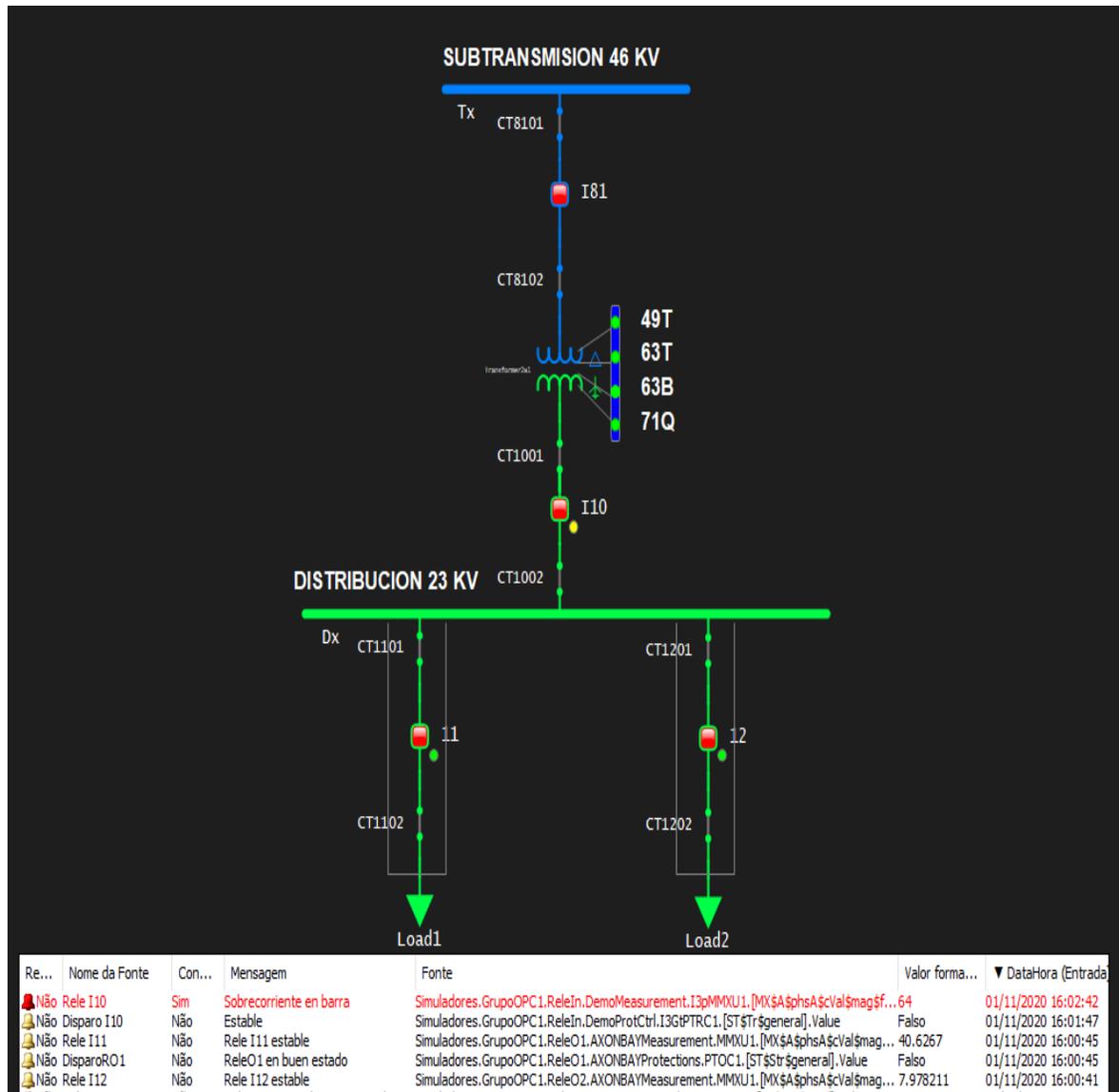


Figura 154. Sobre corriente en la barra simulado por el relé INFOTECH

Como se observa en la figura anterior al existir una falla en la barra (simulada por el relé INFOTECH) se activa una alarma la cual indica que existe una sobre corriente en la barra, los interruptores I11 e I12 se mantienen estables hasta el momento en que la protección de I10 se activa.

La figura siguiente muestra el evento una vez que I10 es disparado por su protección, se activa una nueva alarma indicando que ha ocurrido un disparo por sobre corriente en la barra inmediatamente se envía un mensaje GOOSE a los interruptores de salida, el indicador de color amarillo en la parte superior de los interruptores de salida, muestra que ha ocurrido una falla en el alimentador principal.

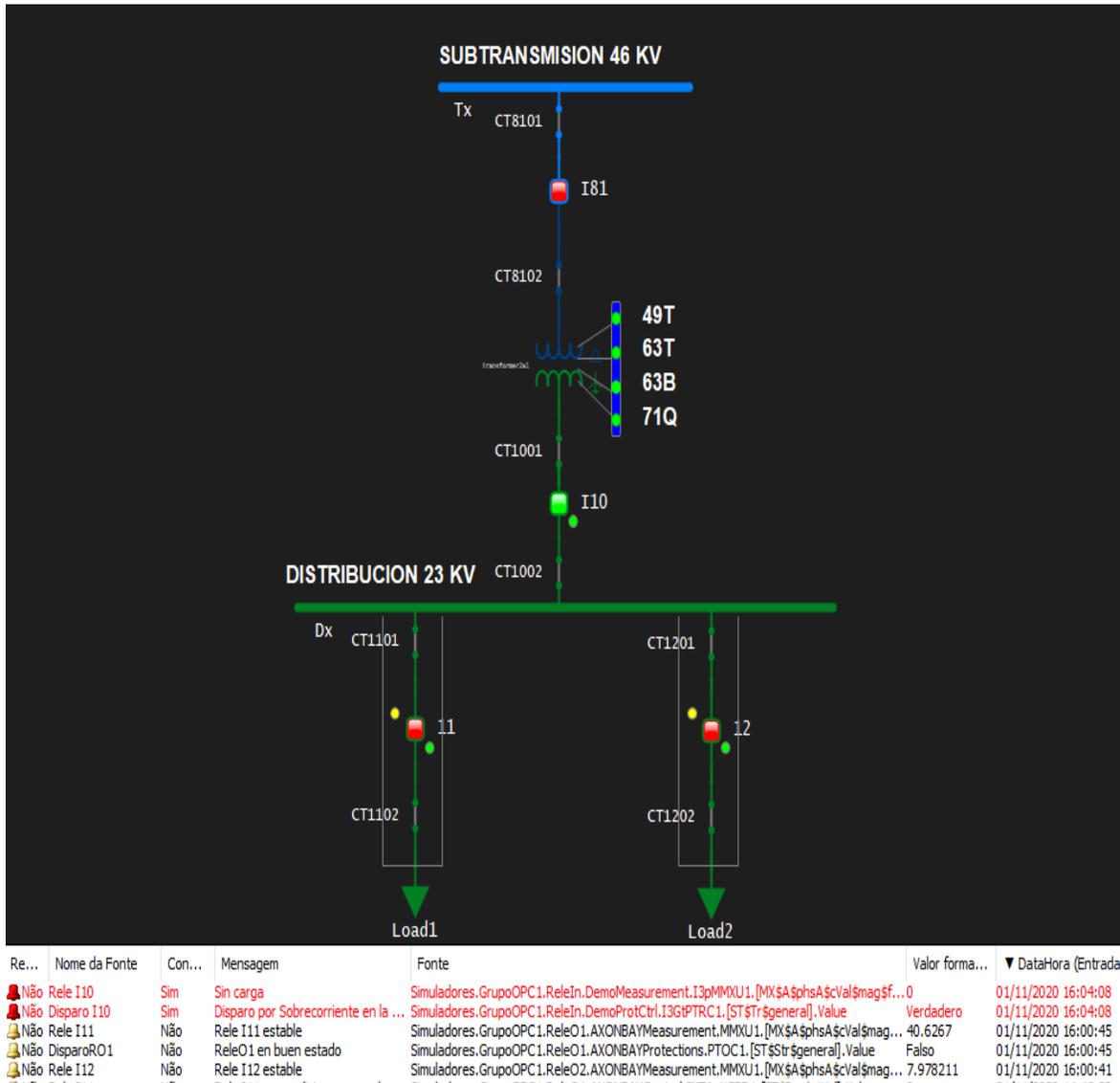


Figura 155. Disparo de I10 por falla en la barra

Finalmente se activa la protección de los interruptores de salida simulados por los relés de AxonGroup despejando totalmente la falla en la barra, luego de ocurrir este evento se activarán nuevas alarmas indicando que los interruptores de I10, I11 e I12 se encuentran sin carga tal como se muestra en la figura 156.

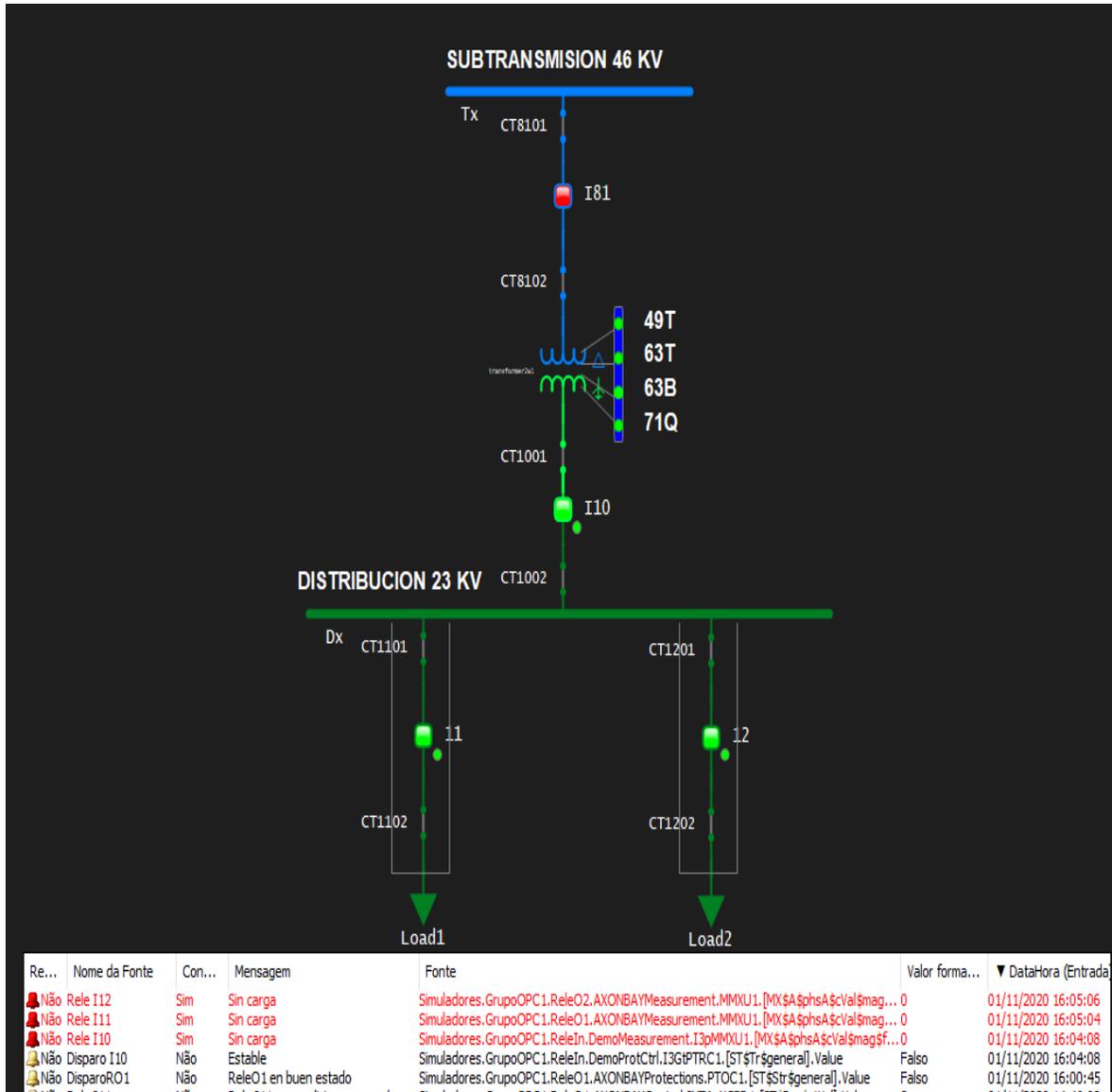


Figura 156. Protección de barra por sobre corriente instantánea

CONCLUSIONES

- La implantación de estándares como el IEC61850 en las subestaciones automatizan muchos procesos, considerando esto una ventaja para el personal que realiza la instalación, configuración y mantenimiento en las subestaciones debido a la interoperabilidad definida en la norma entre equipos de diferentes fabricantes, que permite un fácil reemplazo de los equipos.
- La interoperabilidad entre equipos de distintos fabricantes se logra fácilmente para el caso de disparos de protecciones mediante el protocolo GOOSE, debido a que se utilizan los mismos parámetros de configuración y la única diferencia entre fabricantes es el manejo de su software propietario de configuración.
- Actualmente el país se encuentra en la estandarización bajo IEC61850 de forma gradual en subestaciones de transmisión y distribución, por lo que es importante incluir el estudio del estándar en la formación de los nuevos profesionales del país.
- Debido a los elevados costos de equipos profesionales utilizados en subestaciones bajo el estándar IEC61850, el uso de dispositivos virtuales y la emulación de dispositivos mediante la librería libiec61850 es la opción más indicada para la introducir el estándar IEC61850 en el desarrollo de prácticas de laboratorio.

RECOMENDACIONES

- Presentar a empresas de generación, transmisión y distribución la propuesta de laboratorio expuesto en este documento con el fin de que pueda existir un asocio para la donación de equipos al laboratorio, oportunidades de pasantías y para la capacitación sobre el estándar IEC61850 tanto de estudiantes como de personal docente.
- Proponer la inclusión del tema Smart Grid y el estándar IEC61850 en el contenido de materias electivas impartidas en la EIE/UES, para preparar a los estudiantes en las nuevas tecnologías que a mediano plazo podrían llegar a ser predominantes en los sistemas eléctricos del país.
- Para la mejora de tiempos de retardo y la desaturación de la red se recomienda la utilización de switches de comunicación bajo el estándar IEC61850 y configurando redes virtuales (VLANS).
- Para futuros trabajos que aborden el estándar IEC61850, se recomienda incluir el uso de la librería libiec61850 que en conjunto con la placa Raspberry es posible emular IEDs de bajo costo tal y como se ha desarrollado en el presente documento.
- Se recomienda la investigación de simuladores y emuladores que trabajen bajo el protocolo de valores muestreados (SV) que es parte de la pila de protocolos del estándar IEC61850.
- Se recomienda tomar de ejemplo el escenario dos de este documento para desarrollar nuevos escenarios, utilizando los nuevos métodos de protección por protocolo GOOSE.
- Para futuros trabajos relacionados con IEC61850 se recomienda a las autoridades de la escuela obtener una edición más actualizada del estándar.

REFERENCIAS

1. Cediell, J. y Chaparro, H. (2010). *Aspectos generales en la automatización de una subestación eléctrica y empleo del protocolo de comunicación IEC61850*, Universidad Industrial de Santander.
2. Bernal, J., Herrera, N. y Guevara, J. (2017). *Aplicación del estándar IEC61850 en los sistemas de protecciones y mediciones eléctricas en subestaciones de alta tensión*, Universidad de El Salvador.
3. Yépez, J. y Granda, K. *Implementación de mensajería GOOSE bajo la norma IEC61850 en relés SEL para esquemas de protección de barra*, Universidad Politécnica Salesiana.
4. Infotech. *Guía de Usuario IEC61850 Avenue 2.1 Substation Communication Tool*. [online] Available at: <https://www.infotech.pl/iec-61850/libraries/tip-iec-61850-browser-61850-avenue-tip> [Accesed 14 Dic. 2020].
5. Elipse. *Manual de usuario Elipse Power SCADA*. [online] Available at: <https://www.elipse.com.br/downloads/?cat=80&key=&language=ptbr> [Accesed 14 Dic. 2020].
6. Axon group. *Manual de usuario-AT61*. [online] Available at: <https://www.axongroup.com.co/downloads/> [Accesed 14 Dic. 2020].
7. Sajad Amjadi, M. (2016). *Managing IEC61850 GOOSE Messaging in Multi-vendor Zone Substations*, Victoria University.
8. Moreno, A. y Pallares, V.(2016). *Estudio con modelo de datos para la automatización en redes eléctricas inteligentes*, Universidad de Cordoba.
9. *IEC61850 Communication networks and systems in substations ED.1 (2003). Technical Report*.
10. Michael Zillgith, *Documentación libiec61850*. [online] Available at: <https://libiec61850.com/libiec61850/documentation/> [Accesed 14 Dic. 2020].
11. Gordon Henderson. *WiringPi reference*. [online] Available at: <http://wiringpi.com/reference/> [Accesed 14 Dic. 2020].
12. Github. *Libmodbus documentation*. [online] Available at: <https://libmodbus.org/documentation/>

ANEXOS

Anexo A. Código básico servidor IEC61850

```
/*
*****
*Autores: Joel Arnaldo Lopez Alvarado, Miguel Enrique Vasquez Ceron
*Versión: 1.0
*Licencia: GPL v2
*Se ha retomado del ejemplo "server_example_simple" de la librería
*libieec61850-1.4
*Copyright 2013 Michael Zillgith
*Descripción: El presente código contiene las instrucciones básicas para
*la creación de un servidor bajo el estándar IEC61850, dicho servidor podrá
*ser explorado desde cualquier cliente bajo el protocolo, el código fue
*comentado al español para una mejor comprensión de los parámetros y
*funciones utilizadas, las cuales se encuentra una descripción en la sección
*1.2.10
*Para la ejecución de código ejecutar el siguiente comando en una consola
*bash ubicada en la carpeta del código:
*$sudo ./nombre_del_archivo.c [nombre adaptador de red]
*En caso se utilice el adaptador eth0 no escribir argumento del nombre
*adaptador de red
*****
#include "ieec61850_server.h"
#include "hal_thread.h"
#include <signal.h>
#include <stdlib.h>
#include <stdio.h>
#include "static_model.h"
/* importamos modelo del dispositivo IEC 61850 creado de archivo SCL */
extern IedModel iedModel;
static int running = 0;
void sigint_handler(int signalId)//funcion para detener la ejecucion del
servidor
{
    running = 0;
}
int main(int argc, char** argv) {
    int tcpPort = 102;// Puerto a utilizar
/*En la terminal de ejecucion incluimos argumento para seleccionar
adaptador de red
en caso contrario se tomara por defecto el adaptador "eth0"*/
    if (argc > 1) {
        tcpPort = atoi(argv[1]);
    }
/*creación de modelo a partir de archivos SCL
    IedServer iedServer = IedServer_create(&iedModel);
/*Servidor MMS comenzara a recibir conexiones de clientes IEC61850*/
    IedServer_start(iedServer, tcpPort);
/*En caso exista error en la creación del servidor se detendra el programa*/
    if (!IedServer_isRunning(iedServer)) {
printf("Inicializacion de servidor fallida! Saliendo.\n");

```

```
IedServer_destroy(iedServer);
exit(-1);
}
running = 1;
signal(SIGINT, sigint_handler);
while (running) { //función repetitiva
    Thread_sleep(1);
}
/* Se detiene servidor MMS */
IedServer_stop(iedServer);
/* Limpia - Libera todos los recursos*/
IedServer_destroy(iedServer);
}
```

Anexo B. Código IEDRASPBERRY

```
/*
*Autores: Joel Arnoldo Lopez Alvarado, Miguel Enrique Vasquez Ceron
*Versión: 1.0
*Licencia: GPL v2
*Se ha retomado de los ejemplos "server_example_basic_io",
*"server_example_goose" y "goose subscriber" de la librería *libiec61850-
*1.4
*Copyright 2013 Michael Zillgith
*Descripción: El presente código contiene las instrucciones para la
*creación del servidor IEC61850 que se monta en el IEDRaspberry en el cual
*se actualiza en tiempo real el estado del IED, de igual forma contiene la
*publicación GOOSE del estado de sus entradas digitales y el comportamiento
*a seguir al recibir comandos de otro IED vía suscripción GOOSE. Las
*funciones y parámetros utilizados se explican en la sección 1.2.10 y 2.2.1
*respectivamente.
*
*Para la ejecución de código ejecutar el siguiente comando en una consola
*bash ubicada en la carpeta del código:
*$sudo ./nombre_del_archivo.c [nombre adaptador de red]
*En caso se utilice el adaptador eth0 no escribir argumento del nombre
*adaptador de red
*
*****/
#include "goose_receiver.h"
#include "goose_subscriber.h"
#include "iec61850_server.h"
#include "hal_thread.h"
#include <signal.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <wiringPi.h>
#include <stdbool.h>
#include <stdint.h>
#include "mms_value.h"
#include "goose_publisher.h"
#include "static_model.h"
/* importamos modelo del dispositivo IEC 61850 creado de archivo SCL */
extern IedModel iedModel;

static int running = 0;
static IedServer iedServer = NULL;

static int valor[5]; // variable para almacenar el estado de los LEDs
static unsigned int numanterior = 100; // variable para control de lectura
de mensajes goose

void sigint_handler(int signalId)//funcion para detener la ejecucion del
servidor
{
    running = 0;

```

```

}
/* Función para lectura de mensajes de subscripcion GOOSE*/
void
gooseListener(GooseSubscriber subscriber, void* parameter)
{
    /*Condicion para detener impresion de mensajes al no recibir nuevos
    mensajes GOOSE*/
    if (numanterior != GooseSubscriber_getSqNum(subscriber)){
        /*impresion de datos importantes del mensaje GOOSE*/
        printf("GOOSE event:\n");
        printf("          stNum:   %u   sqNum:   %u   numant:   %u   \n",
GooseSubscriber_getStNum(subscriber),
          GooseSubscriber_getSqNum(subscriber),numanterior);
        printf("          timeToLive:           %u\n",
GooseSubscriber_getTimeAllowedToLive(subscriber));
        //obtenemos e imprimimos estampa de tiempo de publicador
        uint64_t timestamp = GooseSubscriber_getTimestamp(subscriber);
        printf("          timestamp:  %u.%u\n", (uint32_t) (timestamp / 1000),
(uint32_t) (timestamp % 1000));
        // obtenemos mensaje goose y lo almacenamos en un buffer de tipo char
        MmsValue* values = GooseSubscriber_getDataSetValues(subscriber);
        char buffer[1024];
        char *p;
        //imprimimos en consola mensaje goose recibido
        MmsValue_printToBuffer(values, buffer, 1024);
        /* condiciones para clasificar mensajes GOOSE recibidos y cambiar
        estado de LEDs*/
        if (strcmp(buffer,"{true}") == 0){
            valor[0] = 1;}
        else {
            valor[0] = 0;}
        }
        numanterior = GooseSubscriber_getSqNum(subscriber);
    }
}
int main(int argc, char** argv) {
    int tcpPort = 102;// Puerto a utilizar
    //creación de modelo a partir de archivos SCL
    IedServer = IedServer_create(&iedModel);
    /*Servidor MMS comenzara a recibir conexiones de clientes MMS server
    will be instructed to start listening to client connections. */
    IedServer_start(IedServer, 102);

    /*INICIO CONFIGURACION PARAMETROS PARA SUSCRIPCION GOOSE */
    GooseReceiver receiver = GooseReceiver_create();
    //seleccionamos adaptador de red
    GooseReceiver_setInterfaceId(receiver, "eth0");
    //Configuramos nombre de mensaje GOOSE a leer y AppId
    GooseSubscriber subscriber =
GooseSubscriber_create("DemoProtCtrl/LLN0$GO$gcb1", NULL);
    GooseSubscriber_setAppId(subscriber, 1);
    //Configuramos funcion que manejara los mensajes GOOSE
    GooseSubscriber_setListener(subscriber, gooseListener, NULL);
    GooseReceiver_addSubscriber(receiver, subscriber);
}

```

```

GooseReceiver_start(receiver); //Inicio de recepcion de mensajes GOOSE
//FIN SUSCRIPCION GOOSE

//INICIALIZACION DE PARAMETROS PUBLICADOR GOOSE PARA DISPARO AXON
GROUP

char* interface;
interface = "eth0"; //Seleccion de adaptador de red ethernet
//Inicializamos los parametros goose y configuramos appid, dirección
MAC y VLAN
CommParameters gooseCommParameters;
gooseCommParameters.appId = 4096 ;
gooseCommParameters.dstAddress[0] = 0x01;
gooseCommParameters.dstAddress[1] = 0x0c;
gooseCommParameters.dstAddress[2] = 0xcd;
gooseCommParameters.dstAddress[3] = 0x01;
gooseCommParameters.dstAddress[4] = 0x00;
gooseCommParameters.dstAddress[5] = 0x01;
gooseCommParameters.vlanId = 0;
//creamos objeto tipo GOOSE PUBLISHER
GoosePublisher publisher =
GoosePublisher_create(&gooseCommParameters, interface);

if (publisher) { //En caso no exista error en la creación del objeto
//Agregamos los parametros GOOSE ID, Data set.
GoosePublisher_setGoCbRef(publisher,
"REMOTEProtections/LLN0$GO$gcb1");
GoosePublisher_setGoID(publisher, "RemoteProtection");
GoosePublisher_setConfRev(publisher, 1);
GoosePublisher_setDataSetRef(publisher,
"REMOTEProtections/LLN0$DS1_GOOSE");
}
//FIN DE INICIALIZACION DE PARAMETROS PUBLICADOR GOOSE PARA DISPARO
AXON GROUP

//INICIO ENVIO DE DATOS GOOSE DE ENTRADAS DIGITALES IEDRASPBERRY
// si incluimos argumento en la ejecucion del código se tomara como
nombre de interfaz de red
if (argc > 1) {
char* ethernetIfcID = argv[1];
printf("Utilizando interfaz GOOSE: %s\n", ethernetIfcID);

/* set GOOSE interface for all GOOSE publishers (GCBs) */
IedServer_setGooseInterfaceId(iedServer, ethernetIfcID);
}
//En caso contrario se tomara el adaptador de red Eth0
if (argc > 2) {
char* ethernetIfcID = argv[2];
printf("Utilizando inferfaz GOOSE para
GenericIO/LLN0.gcbAnalogValues: %s\n", ethernetIfcID);
/*Configura interfaz GOOSE para un publicador particular(GCB)*/
IedServer_setGooseInterfaceIdEx(iedServer,
IEDMODEL_GenericIO_LLN0, "valores_indicadores", ethernetIfcID);

```

```

    }
    /*FIN DE ENVIO GOOSE*/

    if (!IedServer_isRunning(iedServer)) {
        printf("Inicializacion de servidor fallida! Saliendo.\n");
        IedServer_destroy(iedServer);
        exit(-1);
    }
    IedServer_enableGoosePublishing(iedServer); //Comienza a publicar
mensajes GOOSE
    wiringPiSetup (); // Inicializamos librería WiringPi
    /*Configuramos pines GPIO a utilizar*/
    pinMode (29, OUTPUT);
    pinMode (28, OUTPUT);
    pinMode (27, OUTPUT);
    pinMode (26, OUTPUT);
    pinMode (25, OUTPUT);
    pinMode (21, INPUT);
    pinMode (22, INPUT);
    pinMode (23, INPUT);
    pinMode (24, INPUT);
    running = 1;
    signal(SIGINT, sigint_handler);
while (running) { // Función repetitiva

bool led = valor[0] != 0; //convertimos variable valor en booleano
LinkedList dataSetValues = LinkedList_create(); //creamos objeto para
almacenar datos
LinkedList_add(dataSetValues, MmsValue_newBoolean(led)); //añadimos al
objeto valores MMS de tipo bool
    if (GoosePublisher_publish(publisher, dataSetValues) == -1)
    { //publicación de mensaje GOOSE
        printf("Error enviando mensaje!\n");
    }
    //Eliminamos objeto que almaceno los datos
    LinkedList_destroyDeep(dataSetValues, (LinkedListValueDeleteFunction)
MmsValue_delete);
    Thread_sleep(1);
    //obtenemos estampa de tiempo para la actualizacion de valores
    uint64_t timeStamp = Hal_getTimeInMs();
    Timestamp iecTimestamp;
    Timestamp_clearFlags(&iecTimestamp);
    Timestamp_setTimeInMilliseconds(&iecTimestamp, timeStamp);
    Timestamp_setLeapSecondKnown(&iecTimestamp, true);
    //actualizamos pin 29 con valor leído de suscripcion GOOSE
    digitalWrite(29, valor[0]);
    /*Leemos el valor de los pines GPIO de los Dipswitch y actualizamos su
    corespondiente atributo */

    IedServer_updateUTCTimeAttributeValue(iedServer, IEDMODEL_GenericIO_GGIO1_
Ind1_t, timeStamp);
    IedServer_updateBooleanAttributeValue(iedServer,
    IEDMODEL_GenericIO_GGIO1_Ind1_stVal, digitalRead(21));

```

```

IedServer_updateUTCTimeAttributeValue(iedServer,IEDMODEL_GenericIO_GGIO1_
Ind2_t, timeStamp);
IedServer_updateBooleanAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_Ind2_stVal, digitalRead(22));

IedServer_updateUTCTimeAttributeValue(iedServer,IEDMODEL_GenericIO_GGIO1_
Ind3_t, timeStamp);
IedServer_updateBooleanAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_Ind3_stVal, digitalRead(23));

IedServer_updateUTCTimeAttributeValue(iedServer,IEDMODEL_GenericIO_GGIO1_
Ind4_t, timeStamp);
    IedServer_updateBooleanAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_Ind4_stVal, digitalRead(24));
//Actualizamos salidas para relés para indicar estado de las entradas
    digitalWrite(28,digitalRead(21));
    digitalWrite(27,digitalRead(22));
    digitalWrite(26,digitalRead(23));
    digitalWrite(25,digitalRead(24));
/*Llamado a funcion para lectura de subscripcion GOOSE*/
    gooseListener(subscriber,NULL);
}
/* Se detiene servidor MMS */
IedServer_stop(iedServer);
/* Limpia - Libera todos los recursos*/
IedServer_destroy(iedServer);
}

```

Anexo C. Código convertidor IEC61850/Modbus

```
/*
*Autores: Joel Arnoldo Lopez Alvarado, Miguel Enrique Vasquez Ceron
*Versión: 1.0
*Licencia: GPL v2
*Se ha retomado de los ejemplos "server_example_basic_io",
*"server_example_goose" y "goose subscriber" de la librería *libiec61850-
*1.4
*Copyright 2013 Michael Zillgith
*Descripción: El presente código contiene las instrucciones para la
*creación de un servidor virtual IEC61850 que se monta en una PC y hace las
*funciones de gateway entre el protocolo modbus y los protocolos del
*estándar IEC61850 MMS y GOOSE, contiene también el comportamiento al
*recibir instrucciones mediante las suscripciones GOOSE a otros equipos en
*la subestación. Las funciones y parámetros utilizados se explican en la
*sección 1.2.10 y 2.2.2 *respectivamente.
*
*Para la ejecución de código ejecutar el siguiente comando en una consola
*bash ubicada en la carpeta del código:
*$sudo ./nombre_del_archivo.c [nombre adaptador de red]
*En caso se utilice el adaptador eth0 no escribir argumento del nombre
*adaptador de red
*
*****/
#include "goose_receiver.h"
#include "goose_subscriber.h"
#include "iec61850_server.h"
#include "hal_thread.h"
#include <signal.h>
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <modbus.h>
#include <string.h>
#include "static_model.h"
#include <stdbool.h>
#include <stdint.h>
#include "mms_value.h"
#include "goose_publisher.h"
#define SERVIDOR 0x01 //Direccion asignada al servidor modbus
modbus_t *ctx=NULL; //Crea puntero
/* importamos modelo del dispositivo IEC 61850 creado de archivo SCL */
extern IedModel iedModel;
static int running = 0;
static IedServer iedServer = NULL;
static int valor[5]; //Variable para almacenar datos de suscripcion GOOSE
static unsigned int numanterior = 100; //Variable para control de impresion
en consola
void sigint_handler(int signalId) //funcion para detener la ejecucion del
servidor
{
    running = 0;

```

```

}
static CheckHandlerResult
checkHandler(ControlAction action, void* parameter, MmsValue* ctlVal, bool
test, bool interlockCheck)
{// Función para comprobar los comandos MMS recibidos sean validos

    if (parameter == IEDMODEL_GenericIO_GGIO1_SPCS01)
        return CONTROL_ACCEPTED;

    if (parameter == IEDMODEL_GenericIO_GGIO1_SPCS02)
        return CONTROL_ACCEPTED;

    if (parameter == IEDMODEL_GenericIO_GGIO1_SPCS03)
        return CONTROL_ACCEPTED;

    if (parameter == IEDMODEL_GenericIO_GGIO1_SPCS04)
        return CONTROL_ACCEPTED;

    return CONTROL_OBJECT_UNDEFINED;
}
static ControlHandlerResult
controlHandlerForBinaryOutput(ControlAction action, void* parameter,
MmsValue* value, bool test)
{// Función para cambio de estado de objetos SPCS por comando MMS
    uint64_t timestamp = Hal_getTimeInMs();// obtiene la estampa de tiempo
/*Condiciones para actualizar valor segun atributo al que se envio comando*/
/* Se utilizara el objeto SPCS01 para resetear el relé en caso ser
reestablezcan condiciones*/
    if (parameter == IEDMODEL_GenericIO_GGIO1_SPCS01) {
        /*actualizacion de estampa de tiempo*/
        IedServer_updateUTCTimeAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01_t, timestamp);
        /*actualizacion de valor segun comando recibido*/
        IedServer_updateAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01_stVal, value);
        //usamos funcion get boolean para obtener booleano a partir de dato
MMS
        valor[4] = MmsValue_getBoolean(value);
        valor[4] = valor[4] != 1;//negamos valor obtenido
    }
    else
        return CONTROL_RESULT_FAILED;
    return CONTROL_RESULT_OK;
}
/* Función para lectura de mensajes de subscripcion GOOSE*/
void
gooseListener(GooseSubscriber subscriber, void* parameter)
{
    /*Condicion para detener impresion de mensajes al no recibir nuevos
mensajes GOOSE*/
    if (numanterior != GooseSubscriber_getSqNum(subscriber)){
        /*impresion de datos importantes del mensaje GOOSE*/
        printf("GOOSE event:\n");
    }
}

```

```

printf("          stNum:   %u   sqNum:   %u   numant:   %u   \n",
GooseSubscriber_getStNum(subscriber),
        GooseSubscriber_getSqNum(subscriber),numanterior);
printf("                                timeToLive:           %u\n",
GooseSubscriber_getTimeAllowedToLive(subscriber));
//obtenemos e imprimimos estampa de tiempo de publicador
uint64_t timestamp = GooseSubscriber_getTimestamp(subscriber);
printf("          timestamp:  %u.%u\n", (uint32_t) (timestamp / 1000),
(uint32_t) (timestamp % 1000));
// obtenemos mensaje goose y lo almacenamos en un buffer de tipo char
MmsValue* values = GooseSubscriber_getDataSetValues(subscriber);
char buffer[1024];
char *p;
//imprimimos en consola mensaje goose recibido
MmsValue_printToBuffer(values, buffer, 1024);
/* condiciones para convertir en numeros enteros mensajes GOOSE
recibidos */
if      ((strcmp(buffer,"{true,false,false,false}") ==
0) || (strcmp(buffer,"{true,false,false,true}") ==
0) || (strcmp(buffer,"{true,false,true,false}") ==
0) || (strcmp(buffer,"{true,false,true,true}") ==
0) || (strcmp(buffer,"{true,true,false,false}") ==
0) || (strcmp(buffer,"{true,true,false,true}") ==
0) || (strcmp(buffer,"{true,true,true,false}") ==
0) || (strcmp(buffer,"{true,true,true,true}") == 0)){
    valor[0] = 1;}

else {
    valor[0] = 0;}
if      ((strcmp(buffer,"{false,true,false,false}") ==
0) || (strcmp(buffer,"{false,true,false,true}") ==
0) || (strcmp(buffer,"{false,true,true,false}") ==
0) || (strcmp(buffer,"{false,true,true,true}") ==
0) || (strcmp(buffer,"{true,true,false,false}") ==
0) || (strcmp(buffer,"{true,true,false,true}") ==
0) || (strcmp(buffer,"{true,true,true,false}") ==
0) || (strcmp(buffer,"{true,true,true,true}") == 0)){
    valor[1] = 1;}

else {
    valor[1] = 0;}
if      ((strcmp(buffer,"{false,false,true,false}") ==
0) || (strcmp(buffer,"{false,false,true,true}") ==
0) || (strcmp(buffer,"{false,true,true,false}") ==
0) || (strcmp(buffer,"{false,true,true,true}") ==
0) || (strcmp(buffer,"{true,false,true,false}") ==
0) || (strcmp(buffer,"{true,false,true,true}") ==
0) || (strcmp(buffer,"{true,true,true,false}") ==
0) || (strcmp(buffer,"{true,true,true,true}") == 0)){
    valor[2] = 1;}

else {
    valor[2] = 0;}

```

```

if      ((strcmp(buffer,"{false,false,false,true}")      ==
0)|| (strcmp(buffer,"{false,false,true,true}")          ==
0)|| (strcmp(buffer,"{false,true,false,true}")          ==
0)|| (strcmp(buffer,"{false,true,true,true}")           ==
0)|| (strcmp(buffer,"{true,false,false,true}")         ==
0)|| (strcmp(buffer,"{true,false,true,true}")          ==
0)|| (strcmp(buffer,"{true,true,false,true}")          ==
0)|| (strcmp(buffer,"{true,true,true,true}") == 0)){
    valor[3] = 1;}
else {
    valor[3] = 0;}
printf("%i          %i          %i          %i          %i
%i\n",valor[0],valor[1],valor[2],valor[3],valor[4]);
    }
    numanterior = GooseSubscriber_getSqNum(subscriber);
}
int main(int argc, char** argv) {
    int tcpPort = 102;
//creación de modelo a partir de archivos SCL
    IedServer = IedServer_create(&iedModel);
/*Llamado a función para cambiar valor a atributos SPCS
por recibir comando MMS*/
    IedServer_setControlHandler(iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01,
        (ControlHandler) controlHandlerForBinaryOutput,
        IEDMODEL_GenericIO_GGIO1_SPCS01);
    //INICIO CONEXION MODBUS
    uint16_t tab_reg[32]; //Crea un arreglo unsigned long int 16-bit para
colocar los datos
    int socket;
    int rc,rct;
    int puerto;
    int nb;
    char *ip="192.168.1.80"; // IP A UTILIZAR
    struct timeval old_response_timeout;
    struct timeval response_timeout;
//validacion de la ip y puerto
    puerto=502;
    printf("Se utilizara la IP %s y el puerto %d\n",ip,puerto);

/*INICIO CONFIGURACION PARAMETROS PARA SUSCRIPCION GOOSE */
    GooseReceiver receiver = GooseReceiver_create();
//seleccionamos adaptador de red
    GooseReceiver_setInterfaceId(receiver, "enp0s3");
//Configuramos nombre de mensaje GOOSE a leer y AppId
    GooseSubscriber
        subscriber
        =
GooseSubscriber_create("simpleIOGenericIO/LLN0$GO$IndGoose", NULL);
    GooseSubscriber_setAppId(subscriber, 5);
//Configuramos función que manejara los mensajes GOOSE
    GooseSubscriber_setListener(subscriber, gooseListener, NULL);
    GooseReceiver_addSubscriber(receiver, subscriber);
//Inicio de recepción de mensajes GOOSE
    GooseReceiver_start(receiver);

```

```

/*Servidor MMS comenzara a recibir conexiones de clientes MMS*/
IedServer_start(iedServer, 102);
    if (!IedServer_isRunning(iedServer)) {
        printf("Starting server failed! Exit.\n");
        IedServer_destroy(iedServer);
        exit(-1);
    }
    running = 1;
    signal(SIGINT, sigint_handler);
//INICIALIZACION DE PARAMETROS PUBLICADOR GOOSE PARA DISPARO AXON GROUP
    char* interface;
    interface = "enp0s3";//Seleccion de adaptador de red
//Inicializamos los parametros goose y configuramos appid, dirección MAC,
VLAN y prioridad
    CommParameters gooseCommParameters;
    gooseCommParameters.appId = 3;
    gooseCommParameters.dstAddress[0] = 0x01;
    gooseCommParameters.dstAddress[1] = 0x0c;
    gooseCommParameters.dstAddress[2] = 0xcd;
    gooseCommParameters.dstAddress[3] = 0x01;
    gooseCommParameters.dstAddress[4] = 0x00;
    gooseCommParameters.dstAddress[5] = 0x02;
    gooseCommParameters.vlanId = 0;
    gooseCommParameters.vlanPriority = 4;
//creamos objeto tipo GOOSE PUBLISHER
    GoosePublisher publisher =
GoosePublisher_create(&gooseCommParameters, interface);
    if (publisher) {
//Agregamos los parametros GOOSE ID, Data set.
        GoosePublisher_setGoCbRef(publisher, "LLN0$gcb1");
        GoosePublisher_setGoID(publisher, "G3");
        GoosePublisher_setConfRev(publisher, 1);
        GoosePublisher_setDataSetRef(publisher, "LLN0$DS1");
    }

    IedServer_enableGoosePublishing(iedServer);//Comienza a publicar
mensajes GOOSE
while (running) {
    Thread_sleep(100);
//evaluamos si se recibe alguna activacion de protecciones de la
IEDRASPERRY para disparar el relé
    if ((valor[0] == 1 || valor[1] == 1 || valor[2] == 1 || valor[3] ==
1) && valor[4] == 0){
        valor[4] = 1;
        ctx = modbus_new_tcp(ip,puerto); //Usando contexto TCP, ip y puerto
comunicacion modbus
        modbus_set_slave(ctx, SERVIDOR);
        if (ctx == NULL) {
            fprintf(stderr, "No pudo ubicar el contexto libmodbus\n");
            return -1;
        }
        if (modbus_connect(ctx) == -1) {

```

```

    fprintf(stderr, "La conexión fallo: %s\n", modbus_strerror(errno));
    modbus_free(ctx);
    return -1;
}
//Mandamos la orden de disparo de una de las salidas del relé vía
modbus
modbus_write_bit(ctx,0,valor[4]);
modbus_free(ctx);//liberamos recursos utilizados para la
comunicacion
}
/*Enviamos disparo vía GOOSE de interruptores de lado de alta y baja
de TX de potencia*/
bool disparo = valor[4]!=0;
LinkedList dataSetValues = LinkedList_create();
LinkedList_add(dataSetValues, MmsValue_newBoolean(disparo));
    if (GoosePublisher_publish(publisher, dataSetValues) == -1) {
        printf("Error sending message!\n");
    }
    LinkedList_destroyDeep(dataSetValues,
(LinkedListValueDeleteFunction) MmsValue_delete);
    IedServer_updateAttributeValue(iedServer,
IEDMODEL_GenericIO_GGIO1_SPCS01_stVal, MmsValue_newBoolean(disparo != 1));
    gooseListener(subscriber,NULL);
}
/* Se detiene servidor MMS */
IedServer_stop(iedServer);
/* Limpia - Libera todos los recursos*/
IedServer_destroy(iedServer);}

```

Anexo D. Instalación Raspbian desde S.O Windows

- 1) Descargar e instalar Raspberry Pi Imager desde el siguiente link: <https://www.raspberrypi.org/software/>
- 2) Abrir el programa y elegir tarjeta micro SD donde se instala raspbian y en la versión del sistema operativo a instalar seleccionar raspbian, tal como se muestra en la figura 157.

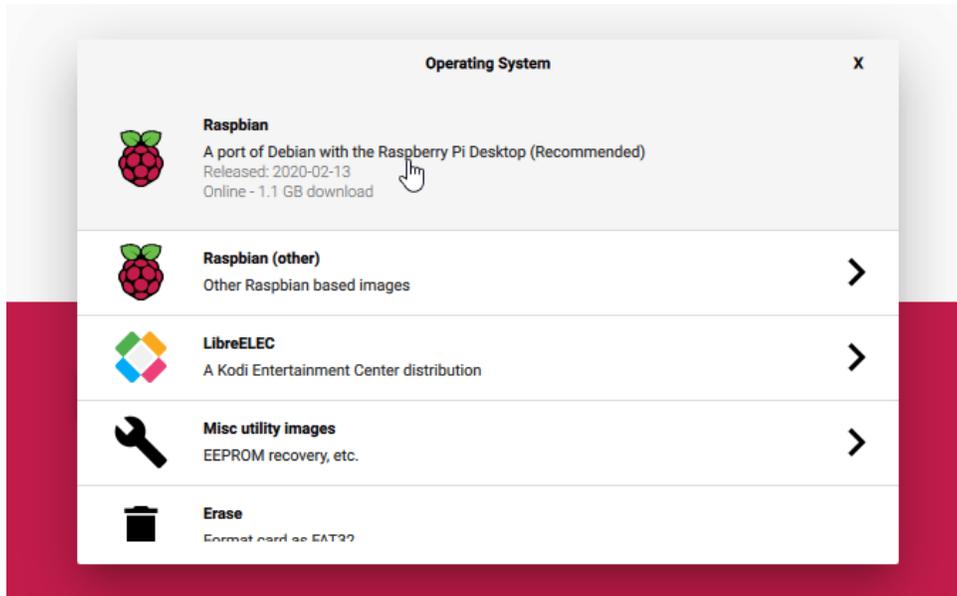


Figura 157. Pantalla de selección sistema operativo.

- 3) Una vez seleccionado hacer clic sobre el botón “write”, se descarga e instala el sistema operativo en la tarjeta SD
- 4) Colocar la tarjeta micro SD en la ranura respectiva de la raspberry pi.
- 5) Instalar pantalla en puerto HDMI, teclado y mouse.
- 6) Conectar cable de alimentación para encender raspberry pi.

Anexo E. Software, simuladores y equipos que se entregan para uso y estudio del estándar a la Escuela de Ingeniería Eléctrica

Equipos y Software	Descripción
IEDExplorer	Software Cliente/Servidor IEC61850
ReLab OPC	Software Cliente/Servidor OPC/IEC61850
Wireshark	Software analizador de datos de red
Eclipse Power	Software SCADA
ICD Editor	Software de edición de archivos ICD
GOOSE Sender	Software para en el envío de mensajes GOOSE
GOOSE Explorer	Software para la recepción de mensajes GOOSE
Simulator Bay (AxonGroup)	Software simulador de una bahía
Simulator Bay (Infotech)	Software simulador de una bahía
IEDRaspberry	IED Emulado con librería IEC61850
SEL351	Relé de protección

Anexo F. Contenido del CD

- i. Documento de proyecto de graduación
- ii. Norma IEC61850 Ed.1
- iii. Instaladores de software simulación
- iv. Instaladores de software cliente/servidor
- v. Librería IEC61850