

Universidad de Alcalá
Escuela Politécnica Superior

**MÁSTER UNIVERSITARIO EN INGENIERÍA
INDUSTRIAL**

Trabajo Fin de Máster

Aplicación del estándar IEC 61850 para
comunicaciones en microrredes eléctricas

ESCUELA POLITECNICA
SUPERIOR

Autor: Sergio Pérez Bachiller

Tutor: Carlos Santos Pérez

Cotutor: Fco. Javier Rodríguez Sánchez

2021

UNIVERSIDAD DE ALCALÁ
Escuela Politécnica Superior

MÁSTER EN INGENIERÍA INDUSTRIAL

Trabajo Fin de Máster

**Aplicación del estándar IEC 61850 para comunicaciones
en microrredes eléctricas**

Autor: Sergio Pérez Bachiller
Tutor/es: Carlos Santos Pérez
Fco. Javier Rodríguez Sánchez

TRIBUNAL:

Presidente: Pedro Martín Sánchez

Vocal 1º: Fernando Cruz Roldán

Vocal 2º: Carlos Santos Pérez

CALIFICACIÓN:

FECHA:

De nuevo y como siempre, a mis padres, a mi hermano y a Elena...

AGRADECIMIENTOS

Para empezar, agradecer a mi tutor Carlos, y a mi cotutor Fco. Javier, por la oportunidad, el apoyo y la ayuda que me han proporcionado para el desarrollo de este trabajo.

En segundo lugar, quería agradecer al grupo de investigación GEISER, especialmente al Proyecto Helios, por brindarme una oportunidad de realizar este trabajo unido a un grupo de investigación, donde he aprendido, me he formado y me ha dado pie a nuevos retos profesionales. Quería agradecer especialmente el trato recibido por parte tanto de cada uno de los profesores que integran este grupo, como de los compañeros que he tenido en el mismo.

Quería agradecer también a todos aquellos que empezaron como compañeros de carrera, y que hoy en día son grandes amigos, los cuales han hecho más ameno cada uno de los cursos realizados. También quería agradecer especialmente a todo el resto de mis amigos que como durante la mayor parte de mi vida, han estado siempre dispuestos a ofrecerme una parte de su tiempo para desconectar, descansar y vivir momentos inolvidables cuando más lo necesitaba.

Por último y posiblemente el agradecimiento más especial, me gustaría dárselo a mi familia, especialmente a mis padres y a mi hermano, los cuales han estado a mi lado siempre, ofreciéndome todo el cariño y el apoyo necesario en cada una de las decisiones que he ido tomando a lo largo de mi vida hasta llegar a este punto, gracias a ellos el recorrido y superar los momentos más difíciles ha sido mucho más sencillo. Como no, junto a mis padres y mi hermano el agradecimiento especial también es para ella, Elena, quien ha confiado ciegamente en mí, en cada uno de los momentos vividos a lo largo de este camino, estando a mi lado y apoyándome incondicionalmente en los momentos más difíciles y en alguna de las decisiones más importantes de mi vida, gracias.

RESUMEN

Este trabajo Fin de Máster (TFM) desarrolla un sistema de comunicaciones escalable para una microrred (MG) eléctrica bajo el estándar de comunicaciones IEC 61850. La MG donde se implementan las comunicaciones es una unidad de generación de recursos distribuidos (DER) basada en un sistema fotovoltaico, por lo que para su modelado este trabajo se basa en la sección del estándar IEC 61850-4-720. Para la implementación del sistema completo se hace uso de un PC, el cual actúa como Cliente/Publicador y una Raspberry Pi, la cual actúa como Servidor/Suscriptor, ambos sistemas se comunican vía Ethernet.

El Cliente/Publicador es el encargado de realizar la gestión externa de la planta, representando el valor de las variables reales de la planta para informar del estado de las mismas al usuario. Por otro lado, el Cliente/Publicador también es el encargado de enviar los comandos necesarios para modificar el estado de la planta. Estas órdenes de apertura se envían a través de la mensajería GOOSE debido a los requisitos de tiempos que presenta la apertura y el cierre de los interruptores de la planta.

Por otro lado, el Servidor/Suscriptor se encarga de la gestión del modelo de datos IEC 61850, de modo que obtiene el valor de las variables reales del sistema y actualiza el modelo de datos bajo el estándar IEC 61850. Asimismo, gestiona los comandos enviados vía GOOSE desde el Cliente/Suscriptor, realiza la configuración de la unidad DER, y supervisa el estado del canal de comunicaciones.

Palabras Clave: Microrred eléctrica, IEC 61850, Cliente, Publicador, Servidor, Suscriptor, GOOSE.

ABSTRACT

This work develops a scalable communications system for an electrical microgrid (MG) under the IEC 61850 communications standard. The MG where the communications are implemented is a distributed resources generation unit (DER) based on a photovoltaic system, so for its modelling this work is based on the section of the IEC 61850-4-720 standard. For the implementation of the complete system, we use a PC, which acts as Client/Publisher and a Raspberry Pi, which acts as Server/Subscriber, both systems communicate via Ethernet.

The Client/Publisher is in charge of the external management of the plant, so that it is responsible for representing the value of the real variables of the plant to report the status of the plant to the user. On the other hand, the Client/Publisher is also in charge of sending the necessary commands to open and close the plant switches to modify the plant status. These opening commands are sent through GOOSE messaging due to the timing requirements presented by the opening and closing of the plant switches.

On the other hand, the Server/Subscriber is in charge of the management of the IEC 61850 data model, so that it obtains the value of the actual system variables and updates the data model under the IEC 61850 standard. On the other hand, it manages the commands sent via GOOSE from the Client/Subscriber, performs the opening and closing of the actual switches, and monitors the status of the communication channel.

Keywords: Microgrid, IEC 61850, Client, Publisher, Server, Subscriber, GOOSE.

ÍNDICE

AGRADECIMIENTOS	7
RESUMEN.....	9
ABSTRACT	11
ÍNDICE DE FIGURAS	15
ÍNDICE DE TABLAS	18
GLOSARIO DE ACRÓNIMOS Y SIMBOLOS.....	19
CAPÍTULO 1: INTRODUCCIÓN	21
1.1.1 Contexto del Trabajo Fin de Máster.....	23
1.1.2 Objetivos del trabajo.....	27
1.1.3 Estructura del documento.....	27
CAPÍTULO 2: BASE TEÓRICA IEC 61850:.....	29
2.1.1 Introducción al IEC 61850	31
2.1.2 Estructura del estándar IEC 61850	35
2.1.3 Función del estándar IEC 61850	37
2.1.4 Nodo Lógico.....	37
2.1.5 Dispositivo Lógico.....	39
2.1.6 Equipo Electrónico Inteligente.....	40
2.1.7 PICOM	40
2.1.8 Tipos de mensajes.....	41
2.1.9 Requerimientos de comunicaciones	45
2.1.1 Modo Test.....	51
2.1.2 Modelo de Datos	55
2.1.1 Seguridad del sistema.....	56
CAPÍTULO 3: DESCRIPCIÓN DEL MATERIAL UTILIZADO	59
3.1.1 Ordenador basado en el sistema operativo Linux	61
3.1 Biblioteca libiec61850.....	61
3.1.1 Tarjeta de desarrollo Raspberry Pi	63
3.1.2 Microcontrolador tipo ESP-32	64
3.1.3 Software de programación para microcontroladores.....	65
3.1.4 Software de programación para tarjetas de desarrollo.....	66
3.1.5 Unidad de generación distribuida	67

CAPÍTULO 4: DESARROLLO DEL SISTEMA.....	73
4.1.1 Valoración de la implementación del estándar en una placa computadora o en un microcontrolador.....	75
4.1.2 Modelado de la unidad de generación distribuida bajo el estándar IEC 61850.....	76
4.1.3 Requerimientos de comunicaciones IEC 61850 en la unidad de generación distribuida.....	90
4.1.4 Sistema de sincronización para medida de tiempos	94
4.1.5 Implementación del modelo basado en el IEC 61850.....	95
4.6.1 Programación Cliente/Publicador (PC).....	97
4.6.2 Programación Servidor/Suscriptor (Raspberry Pi)	107
CAPÍTULO 5: PRUEBAS EXPERIMENTALES.....	115
5.1.1 PRUEBAS ENTRE LAS RASPBERRY PI Y EL PC.....	117
5.1.2 PRUEBAS CON CLIENTE IEC 61850 COMERCIAL.....	123
5.1.3 MEDIDAS DE TIEMPOS SOBRE REQUISITOS GOOSE CON NTP.	130
CAPÍTULO 6: CONCLUSIONES Y TRABAJOS FUTUROS	137
CAPÍTULO 7: PRESUPUESTO.....	141
7.1.1 Coste del material utilizado.....	143
7.1.2 Costes directos del sistema	144
7.1.3 Costes indirectos del sistema	144
7.1.4 Honorarios Facultativos	145
7.1.5 Costes del proyecto.....	146
CAPÍTULO 8: BIBLIOGRAFÍA.....	147
Bibliografía	149

ÍNDICE DE FIGURAS

Figura 1: Unidad de generación básica DER HELIOS.....	25
Figura 2: Virtual Power Plant [6].....	26
Figura 3: Incorporación del vehículo eléctrico.....	26
Figura 4: Subestación convencional y subestación digital.....	32
Figura 5: Deslocalización de funciones.	34
Figura 6: Símbolo IEC 61850.....	34
Figura 7: Modelo conceptual de un sistema completo IEC 61850.....	35
Figura 8: Organización de los conceptos básicos.	36
Figura 9: Bloque LN.....	39
Figura 10: PICOM.....	41
Figura 11: Tipos de mensajes IEC 61850.	41
Figura 12: Esquema de transmisión de mensajes GOOSE [12].....	44
Figura 13: Definición tiempo de transferencia en sistemas IEC 61850.....	46
Figura 14: Tiempo de transferencia para señal binaria.....	47
Figura 15: Niveles y lógicas de interfaces en SAS.....	48
Figura 16: Datos utilizados para la recepción de señales de simulación [16].	52
Figura 17: Ejemplo de señales de entrada en modo test [16].	53
Figura 18: Ejemplo modo test [16].	53
Figura 19: Definición del comportamiento de un LD (LN Beh) [20].....	54
Figura 20: Modelo de datos.....	55
Figura 21: SCL Navigator.	56
Figura 22: Raspberry Pi 3 Model B+.....	63
Figura 23: Arduino IDE.....	66
Figura 24: Geany.	66
Figura 25: Ubicación planta DER.	67
Figura 26: Esquema planta HELIOS.....	68
Figura 27: Paneles solares planta Helios.....	69
Figura 28: Regulador DC-DC planta Helios.....	69
Figura 29: Baterías planta Helios.	70
Figura 30: Equipo de fuerza planta Helios.....	70
Figura 31: Cargas planta Helios.....	71

Figura 32: Digitalización planta Helios según IEC 61850.....	90
Figura 33: Definición de PICOMs para la actuación de interruptores.....	92
Figura 34: Definición de PICOM para la lectura de variables.....	92
Figura 35: Conexión PC-Raspberry Pi.....	93
Figura 36: Ubicación NTP en la pila de protocolos.....	94
Figura 37: Jerarquía NTP.....	95
Figura 38: Flujograma de sistema de comunicaciones basadas en el estándar IEC 61850.....	97
Figura 39: Flujograma programa del Cliente/ Publicador.....	99
Figura 40: Estructura referencia.....	102
Figura 41: DO ECPTType [3].....	103
Figura 42: Valores del modelo de la planta.....	104
Figura 43: Flujograma programa del Servidor/Suscriptor.....	108
Figura 44: GOOSE en el suscriptor.....	118
Figura 45: Pérdida del canal de comunicaciones.....	119
Figura 46: Extracción de los LD del servidor.....	120
Figura 47: Modelo de datos IEC 61850.....	121
Figura 48: Estado de los interruptores de la planta a ON.....	122
Figura 49: Estado de los interruptores de la planta a OFF.....	122
Figura 50: Orden GOOSE de cierre de todos los interruptores de la planta.....	122
Figura 51: Orden GOOSE de apertura de todos los interruptores de la planta.....	122
Figura 52: Logo OMICROM.....	123
Figura 53: Configuración IP ordenador.....	124
Figura 54: Detección automática de Raspberry Pi con IEDSCOUT.....	125
Figura 55: Detección de LD de la Raspberry Pi a través del IEDSCOUT.....	126
Figura 56: Detección de LN de la Raspberry Pi a través del IEDSCOUT.....	126
Figura 57: Detección de DO de la Raspberry Pi a través del IEDSCOUT.....	127
Figura 58: Detección de DA de la Raspberry Pi a través del IEDSCOUT.....	127
Figura 59: Conexión Online del IEDSCOUT con la Raspberry Pi.....	128
Figura 60: Valores en tiempo real de DA de la Raspberry Pi a través del IED SCOUT.	129
Figura 61: Introducción de la marca de tiempo en el dataset del GOOSE.....	131
Figura 62: Análisis de tiempos 1.....	132

Figura 63: Análisis de tiempos 2.....	132
Figura 64: Análisis de tiempos 3.....	133
Figura 65: Análisis de tiempos 4.....	133
Figura 66: Análisis de tiempos 5.....	134

ÍNDICE DE TABLAS

Tabla 1: Capas OSI.....	42
Tabla 2: Clases para tiempos de transferencia.....	49
Tabla 3: Requerimientos mensajes tipo 1A.....	49
Tabla 4: Requerimientos mensajes tipo 1B.....	49
Tabla 5: Requerimientos mensajes tipo 2.....	50
Tabla 6: Requerimientos mensajes tipo 3.....	50
Tabla 7: Requerimientos <i>mensajes</i> tipo 4.....	50
Tabla 8: Requerimientos mensajes tipo 5.....	51
Tabla 9: Requerimientos mensajes tipo 6.....	51
Tabla 10: Costes del material utilizado.....	143
Tabla 11: Costes personal.....	144
Tabla 12: Costes indirectos del sistema.....	144
Tabla 13: Coste por honorarios facultativos.....	145
Tabla 14: Coste del proyecto.....	146

GLOSARIO DE ACRÓNIMOS Y SIMBOLOS

DER	Recursos Energéticos Distribuidos
MG	Microgrid
EMS	Energy Management System
VPP	Plantas virtuales de potencia
TFM	Trabajo Fin de Máster
VLAN	Red de Área Local Virtual
LAN	Red de Área Local
LD	Dispositivo Lógico
LN	Nodo Lógico
DO	Objeto de Dato
DA	Atributo de Dato
IED	Dispositivo Electrónico Inteligente
PICOM	Pices of Information COMunication
OSI	Open Systems Interconnection
SV	Sampled Value
GOOSE	Generic Object-Oriented System Event

MMS	Manufacturing Message Specification
SAS	Sistemas de Automatización de Subestaciones
SBC	Ordenador de Placa Simple
IDE	Entorno de Desarrollo Integrado
GUI	Constructor de Interfaz Gráfica
BTS	Estación Base
ANN	Redes Neuronales Artificiales
MPPT	Control de Punto de Máxima Potencia
CC	Corriente Continua
NTP	Network Time Protocol

CAPÍTULO 1: INTRODUCCIÓN

1.1.1 Contexto del Trabajo Fin de Máster.

En los últimos años se ha producido una creciente expansión de los Recursos Energéticos Distribuidos (DER de sus siglas en inglés, *Distributed Energy Resources*) [1]. Esto se debe principalmente a dos factores: el avance en la electrónica de potencia y el desarrollo de fuentes de energía renovable, todo ello fomentado por la inminente necesidad de reducir las emisiones de gases de efecto invernadero. Por ejemplo, en España está previsto un vertiginoso crecimiento de la solar fotovoltaica hasta alcanzar los 14,6 GW instalados en 2022, a un ritmo de 3 a 4 GW por año [2].

El aumento del número de sistemas DER interconectados a la red eléctrica no solo supone un enorme reto, sino una gran oportunidad. Países de todo el mundo reconocen los beneficios tanto económicos, como sociales y para el medioambiente que supone la integración de la tecnología DER en el sistema eléctrico [3].

Por otro lado, se espera un crecimiento exponencial de estas tecnologías debido a objetivos más ambiciosos para el futuro, como son los indicados en la Agenda2030 [4]:

- Al menos, un 40% menos de emisiones de gases de efecto invernadero en comparación con 1990.
- Al menos, un 27% de energías renovables en el consumo de energía.
- Al menos un 27% de mejora de la eficiencia energética.

Todo esto, ha generado cambios en la red eléctrica incorporando conceptos como el de *microgrid* (MG) o el de *prosumer*. Las MG se definen como un grupo descentralizado de fuentes de generación eléctrica y cargas conectadas entre sí y a la vez conectada a la red eléctrica convencional. Por otro lado, el nuevo concepto de *prosumer* define a un usuario como consumidor y productor de energía eléctrica al mismo tiempo [5].

Todos estos avances y desarrollos en el sistema eléctrico están cambiando el enfoque tradicional de los sistemas de energía, fomentando la introducción de un gran número de generadores en la red de distribución. A diferencia del sistema tradicional que seguía una estructura centralizada, basada en un pequeño número de generadores de gran potencia, conectados directamente a la red de transmisión.

Este cambio de filosofía en la red de energía eléctrica supone un reto a la hora de controlar y garantizar la estabilidad del sistema eléctrico, y más teniendo en cuenta el gran peso que toman los sistemas de generación de energía con carácter estocástico como son las energías renovables [6]. Para ello, se desarrollan sistemas que dotan de "inteligencia" a la red. Estos sistemas de gestión de la energía

distribuida, EMS, (*Energy Management System*), logran rendimientos comparables a los obtenidos en modo centralizado como presenta el trabajo desarrollado en [7].

Este reto para la correcta gestión de la red de energía eléctrica conduce a la necesidad de una operación coordinada de todos los sistemas y equipos presentes en la misma, con el fin de mantener la estabilidad del sistema de potencia (equilibrio entre la generación y el consumo de energía) [8]. Observando la necesidad de una buena comunicación entre los diferentes dispositivos y los diferentes sistemas que componen sistema eléctrico, algunos autores como [9] consideran ahora la red de comunicación como la parte más importante de la automatización de los sistemas eléctricos, especialmente en la red eléctrica. Siendo los estándares de comunicación, imprescindibles para dicha transición [10].

Hoy en día, existen diferentes estándares de comunicación para la monitorización y el control de los sistemas eléctricos como, por ejemplo: DNP, IEC-60870-5-101, IEC 60870-5-104 o Modbus (TCP o RTU). Sin embargo, ninguno de ellos ha conseguido implantarse como estándar único en la mayor parte de los países, lo que dificulta la interoperabilidad de los sistemas de distintos fabricantes.

De esta necesidad surge el estándar IEC 61850, buscando la interoperabilidad entre los diferentes sistemas que componen la red eléctrica. La implementación de esta estándar abarca tanto las subestaciones eléctricas, como los diferentes sistemas de generación que componen la red eléctrica actual. Siendo, de especial importancia, las unidades DER, debido a su generación de energía con carácter estocástico.

En este Trabajo Fin de Máster se propone abordar esta línea de trabajo. Concretamente, se propone la implementación del estándar de comunicación IEC 61850, en la unidad de generación distribuida desarrollada en el proyecto *HELIOS*, por medio de una placa computadora o un microcontrolador.

HELIOS es un proyecto desarrollado por el grupo de investigación GEISER de la Universidad de Alcalá. El objetivo de este proyecto es abordar una de las grandes problemáticas de la red de eléctrica actual, la integración eficiente de las fuentes de generación distribuidas y renovables. Para ello, se implementa una DER con generación renovable basada en energía fotovoltaica, almacenamiento de energía, conexión a la red eléctrica y gestión de cargas (Figura 1).

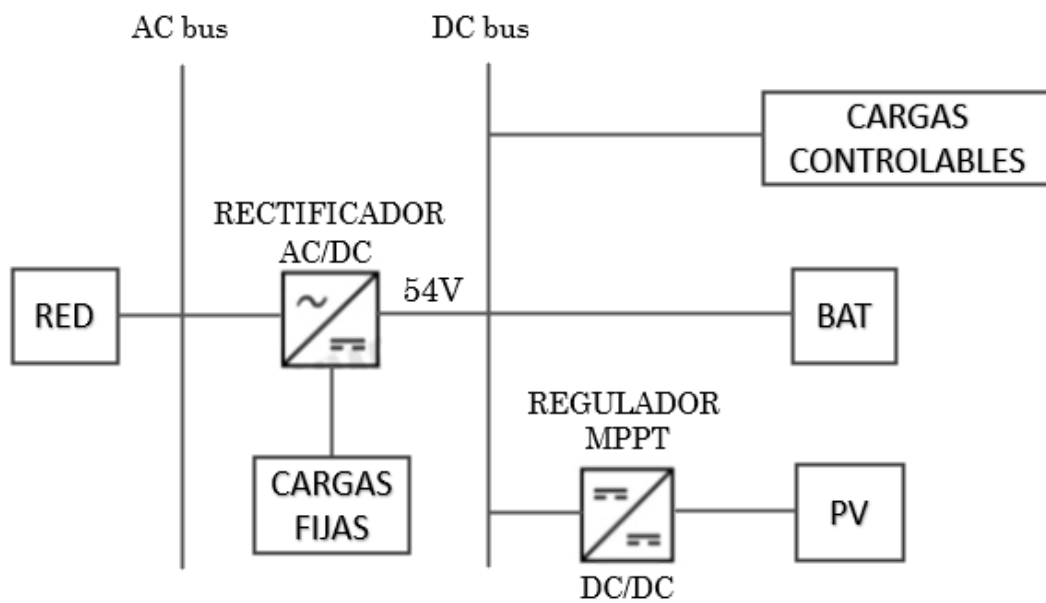


Figura 1: Unidad de generación básica DER HELIOS.

En este proyecto se ha implementado un sistema basado en energía fotovoltaica debido al crecimiento actual de esta energía renovable.

Para integrar estas fuentes de generación de energía de forma eficiente se propone utilizar el concepto de las plantas virtuales de potencia (VPP). Las VPP suponen un nuevo concepto de gestión energética que entrelaza diferentes fuentes de energía heterogéneas distribuidas en un área extensa [11]. Todas ellas se gestionan mediante un único sistema de control centralizado con la ayuda de las tecnologías de la información y el despliegue de dispositivos basados en el internet de las cosas, en los que las comunicaciones entre los distintos dispositivos juegan un papel vital [12].

De esta forma, se establece la posibilidad de instalar pequeños puntos de generación fotovoltaica en los tejados de los edificios y gestionar todos ellos como una planta de generación conjunta. En la Figura 2 se muestra una imagen del concepto de VPP.

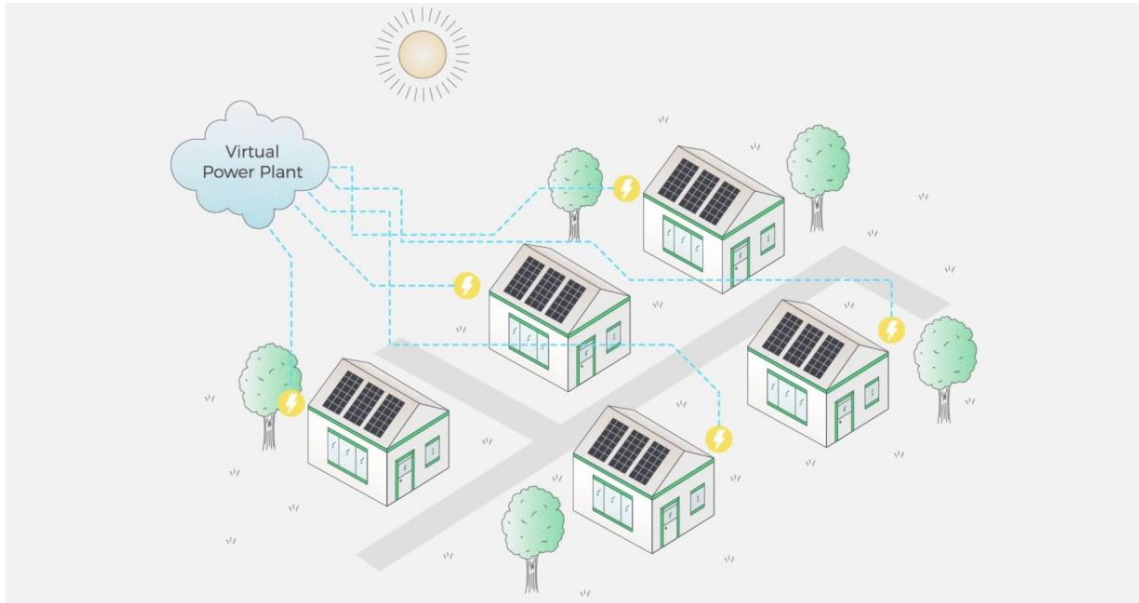


Figura 2: Virtual Power Plant [6].

Además, este modelo, gracias a su flexibilidad, puede verse fortalecido con el despliegue del vehículo eléctrico [13]. Ya que este, podría estar conectado a distintos puntos de la red gestionando la VPP con la carga y descarga de sus baterías Figura 3. Esta gestión se realizaría de forma óptima atendiendo en primer lugar a las necesidades del usuario y en segundo lugar a las necesidades de la instalación y la red.

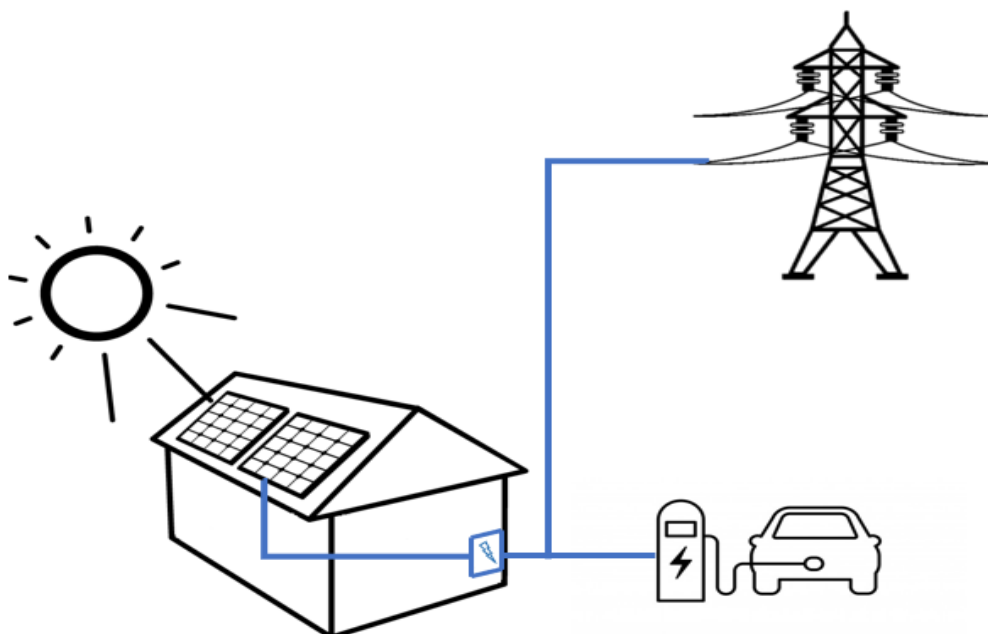


Figura 3: Incorporación del vehículo eléctrico.

1.1.2 Objetivos del trabajo.

El objetivo principal de este Trabajo Fin de Máster es permitir, tanto la interoperabilidad entre los diferentes sistemas de la unidad DER del proyecto Helios, como la interoperabilidad de la unidad como conjunto dentro de una MG, basándose en el estándar IEC 61850.

Para ello, se modela la unidad DER mediante nodos lógicos DER, basándose en la extensión del estándar IEC 61850-7-420. El modelo se introduce en una placa computadora, cuyo firmware gestiona y controla la unidad DER basándose en el protocolo IEC 61850.

De esta manera, se consigue tanto la interoperabilidad de los diferentes dispositivos y sistemas que forman la unidad DER, como de futuros dispositivos que se incorporen.

Por otro lado, permite el control y la gestión de la unidad completa bajo el estándar, con la finalidad de poder crear un EMS que permita gestionar dicha unidad DER dentro de una posible MG o *VPP*

1.1.3 Estructura del documento.

Este documento consta de 9 capítulos, en los cuales se pretende abordar cada uno de los aspectos que se han desarrollado en este TFM (Trabajo Fin de Máster) de manera secuencial hasta llegar al objetivo final.

El capítulo 1, aquí presente, expone una breve introducción del marco en el que está englobado el TFM. El capítulo 2 realiza una explicación de toda la base teórica necesaria para el desarrollo del TFM; en este capítulo se abordan conceptos relacionados con el estándar de comunicaciones IEC 61850. En el capítulo 3 se describe el material utilizado para implementar el proyecto, así como todas sus características. El proceso de desarrollo del sistema de comunicaciones basado en el estándar IEC 61850 paso a paso se desarrolla a lo largo del capítulo 4. En el capítulo 5 se recogen los resultados obtenidos una vez realizadas las pruebas. A continuación, en el capítulo 6, se expone las conclusiones y las posibles líneas futuras que se podrían abordar después de realizar el trabajo aquí presentado.

Posteriormente, se encuentra el presupuesto (capítulo 7). Y, por último, el capítulo 8 recoge toda la bibliografía usada a lo largo del desarrollo de este TFM.

CAPÍTULO 2: BASE TEÓRICA IEC 61850:

2.1.1 Introducción al IEC 61850

En este apartado se procede a realizar una breve introducción del estándar de comunicaciones IEC 61850.

El estándar de comunicaciones IEC 61850 es un estándar internacional de redes y sistemas de comunicaciones, que nació con el objetivo de la automatización del sistema eléctrico.

Este estándar aparece por la necesidad de definir los conceptos y los requerimientos necesarios para la digitalización de subestaciones eléctricas [14]. De esta forma, se buscaba una evolución de lo que se conocía como subestaciones convencionales (cableado), al nuevo concepto de subestaciones digitales (comunicaciones + automatización), consiguiendo (Figura 4):

- **Ahorro:** La reducción del cableado supone un ahorro de entre el 60% y el 80% del cableado. Además, este cambio a la hora implementar las subestaciones también supone un ahorro económico de entre el 7 y el 10%.
- **Espacio:** Los espacios se reducen de una manera notable. Esto también se ve afectado por la reducción del cableado.
- **Mantenimiento:** Como se tienen todos los equipos comunicados, se pueden realizar tareas de mantenimiento predictivo de manera remota de una forma mucho más precisa.
- **Seguridad:** Al eliminar la mayor parte del cableado a través del cual se llevaban las variables eléctricas hacia los dispositivos de medida, se reducen las posibilidades de que un operario se vea dañado por choque eléctrico al tocar una de estas mangueras.

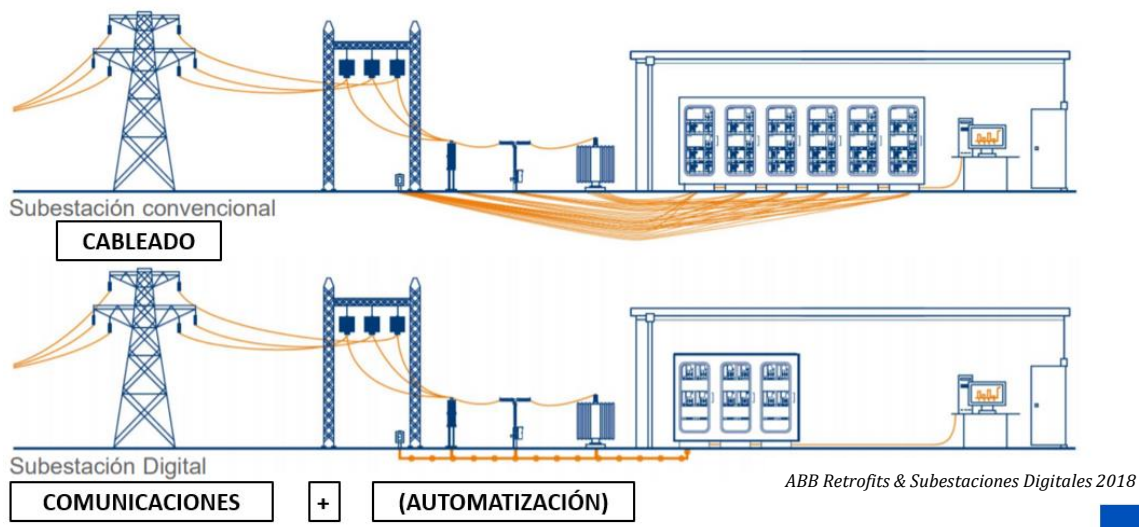


Figura 4: Subestación convencional y subestación digital.

Aunque el estándar de comunicaciones IEC 61850 se creó principalmente para su aplicación en entornos de subestaciones eléctricas, existen diferentes extensiones que abren un amplio abanico de aplicaciones para el mismo. Algunas de estas extensiones y normas complementarias son:

- **IEC 61850-7-410:**

Esta extensión define los modelos de información IEC 61850 para ser utilizados en el intercambio de información en centrales hidroeléctricas.

- **IEC 61400-25:**

Aunque en este caso no se trata de una extensión sino de una norma en sí, la norma IEC 61400-25 que define los requisitos de comunicaciones para aplicaciones de energía eólica se basa en los mismos principios que la norma IEC 61850.

- **IEC 61850-7-420:**

La extensión 7-420 de la norma describe la estructura básica de comunicación junto con los nodos lógicos referidos a Recursos de Energía Distribuidos (DER). Define los modelos de información IEC 61850 para ser utilizados en el intercambio de información con recursos de energía distribuida (DER). Comprende dispositivos de generación y dispositivos de almacenamiento, incluye motores, celdas de combustible, microturbinas, energía fotovoltaica y almacenamiento de energía.

Esta última extensión de la norma es donde se va a enmarcar el desarrollo de este Trabajo Fin de Máster.

Como se ha comentado anteriormente, el objetivo que se busca es poder controlar la introducción masiva de los sistemas DER en la red, y conseguir que estos trabajen de forma coordinada. Para lograr estos objetivos se requieren:

- **Equipos digitales:** Se necesita desarrollar equipos digitales que puedan llevar a cabo intercambios de información entre ellos y, a la vez, realizar acciones de manera coordinada.
- **Medio de Comunicación:** Estos equipos deberán intercambiar información a través de un medio de comunicación. Este medio de comunicación puede ser una red VLAN (red de área local virtual) a través de Ethernet.
- **Digitalización:** Se deben digitalizar todos los elementos presentes en una subestación, para poder hacer uso de ellos a través del sistema digital que gestiona la subestación.
- **Comportamiento uniforme y predecible:** Todos los equipos deben comportarse de una manera uniforme, es decir, independientemente de su función, esta debe realizarse del mismo modo. Y, por otro lado, se deben comportar de una forma predecible, ya que, dos equipos de dos fabricantes diferentes tienen que actuar de la misma forma (de la forma deseable) ante una misma orden.
- **Deslocalización de funciones:** Este es un concepto que no siempre se tiene del todo presente, pero a la hora de diseñar una subestación digital se debe pensar en funciones, no en equipos Figura 5. Por ejemplo, si se está pensando en proteger una línea, se deben pensar en funciones que protejan esa línea. Estas funciones pueden ser función de distancia, de fallo de interruptor o diferencial, pero no se debe pensar en equipos que protejan la línea, ya que una vez que se tengan definidas las funciones, se podrá pensar en equipos que incluso desarrollen diferentes funciones por sí solos.



Figura 5: Deslocalización de funciones.

- **Interoperabilidad semántica:** Uno de los conceptos más importantes a la hora de implementar una subestación digital. Con interoperabilidad semántica se entiende la capacidad de distintos equipos de diferentes fabricantes, que no solo presentan la capacidad de intercambiar información entre ellos, sino que existe un lenguaje definido y conocido entre ellos que permite a un usuario entender la información que se está intercambiando solamente con verla.

Todas estas necesidades se suplen gracias a la aparición y el desarrollo del estándar de comunicaciones IEC 61850 (Figura 6).



Figura 6: Símbolo IEC 61850.

De este modo, gracias al estándar IEC 61850, es posible, mediante un modelo de datos orientado a objetos, describir los dispositivos presentes en la instalación, sus funciones e interfaces de comunicación entre ellos para llevar a cabo las diversas tareas de automatización [15].

De este modo, las características principales del estándar IEC 61850 son:

- El estándar IEC 61850, es un estándar orientado a objetos, de acuerdo con las necesidades presentes en cualquier sistema eléctrico, dividido en funciones y equipos que desempeñan esas funciones.
- Es un estándar que se definió con un objetivo claro, que sea perdurable ante nuevas tecnologías. Esto es así ya que la evolución y el desarrollo de

nuevos equipos y nuevas tecnologías en la actualidad es muy rápido, con lo que uno de sus objetivos a la hora de definirlo fue que no se quedase obsoleto ante la aparición nuevas tecnologías.

- Define una solución integrada para sistemas eléctricos digitales basado en el concepto de interoperabilidad semántica. De esta forma, el estándar define un lenguaje concreto para realizar el intercambio de información entre equipos dentro de la subestación. Permite intercambiar información entre diferentes equipos de diferentes fabricantes, y añade la característica de que hace posible comprender el intercambio de información que se está realizando, viendo la trama intercambiada.

Una vez definidos las características principales del estándar, se procede a exponer su estructura principal y definir cada una de las partes.

2.1.2 Estructura del estándar IEC 61850

El intercambio de información definido por el estándar IEC 61850 se basa principalmente en modelos de datos de información bien estructurados. Los modelos de información y los métodos modelados son la principal característica del estándar IEC 61850 [15].

El enfoque empleado para el modelado de la información se representa en la Figura 7.

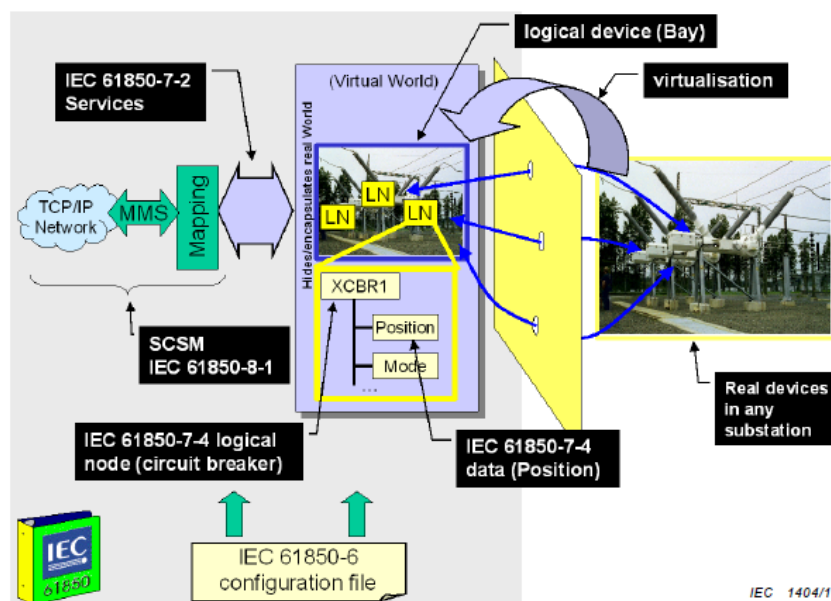


Figura 7: Modelo conceptual de un sistema completo IEC 61850.

Como se observa, una de las partes más importante del modelado es la virtualización del mundo real. Esta virtualización se basa en proporcionar una visión de las características más relevantes de un dispositivo real, que serán de interés y utilizadas para realizar el intercambio de información entre los diferentes equipos. Para ello, el estándar define varios conceptos que le permiten realizar esta tarea de forma estructurada.

Esta estructura se basa en organizar los elementos necesarios para poder realizar una función concreta en la planta DER. Para ello se dispone de un equipo físico al que se asocia el concepto de dispositivo lógico (LD), que a su vez está formado por unidades más pequeñas denominadas nodos lógicos (LN); estos, a su vez, están definidos a través de objetos de datos (DO) y sus atributos (DA). Esta estructura queda definida por la norma según la Figura 8.

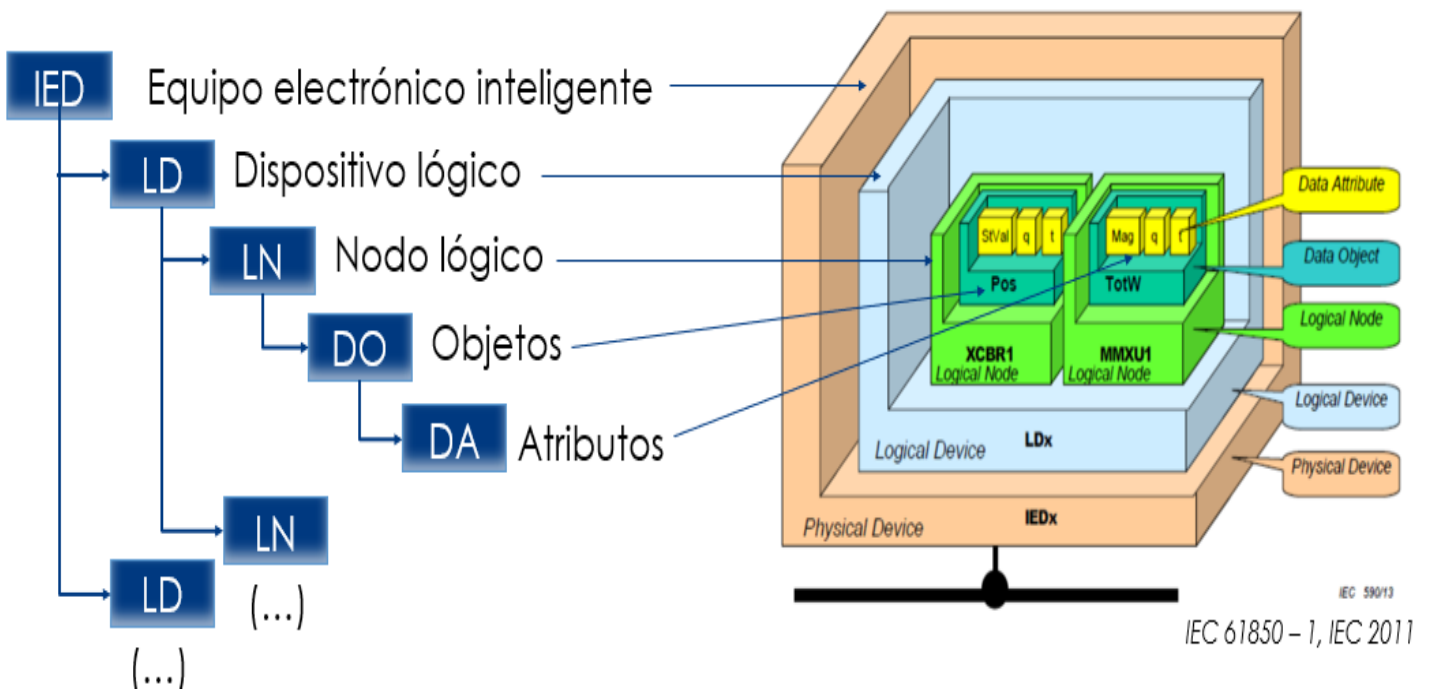


Figura 8: Organización de los conceptos básicos.

2.1.3 Función del estándar IEC 61850

El objetivo del estándar IEC 61850 es proporcionar interoperabilidad entre los dispositivos electrónicos inteligentes (IEDs) de diferentes fabricantes o, más precisamente, entre funciones que deben realizar los sistemas de automatización de los sistemas eléctricos, pero dichas funciones residen en equipos (dispositivos físicos) de diferentes fabricantes [16]. Dichas funciones interoperables, pueden ser las funciones que representan interfaces con el proceso, por ejemplo, interruptores, funciones de protección del sistema, o funciones de automatización del sistema DER. En la sección IEC 61850-7-1 se presentan ejemplos simples de funciones para describir los conceptos y métodos aplicados en la norma IEC 61850.

Por tanto, una función se puede definir como una tarea ejecutada por el sistema automático de potencia mediante funciones de aplicación. Para realizar esta tarea con un mayor grado de precisión, las funciones se descomponen en nodos lógicos.

2.1.4 Nodo Lógico

Como se ha comentado con anterioridad y como se describe en la norma IEC 61850-5, el enfoque de la norma es descomponer las funciones de aplicación en entidades más pequeñas, que se utilizan para intercambiar información. La granularidad viene dada por una distribución razonable de estas entidades en los IED. Estas entidades se denominan nodos lógicos (por ejemplo, para una representación virtual de una clase que representa un interruptor, el nombre de la clase estandarizada es XCBR). Los nodos lógicos se modelan y definen desde el punto de vista de la aplicación conceptual de la parte de la norma IEC 61850-5. Varios nodos lógicos construyen un dispositivo lógico LD (por ejemplo, una representación de una unidad de bahía) [16].

Por lo tanto, un nodo lógico se puede definir como la parte funcional más pequeña de una función que tiene capacidad de intercambiar información con el exterior para llevar a cabo su tarea. Existen nodos lógicos que se basan en funciones que se deben llevar a cabo en una subestación como, por ejemplo, nodos lógicos para funciones de protección:

- **Función diferencial (PDIF):** Este nodo ofrece la información necesaria para implementar una función de protección basada en diferencias de corrientes.
- **Función distancia (PDIS):** Este nodo ofrece la información necesaria para implementar la función de protección de distancia para proteger una línea eléctrica.
- **Fallo de interruptor (RBRF).** A través de este nodo se define la información necesaria para implementar la función de fallo de interruptor en el caso de que algún interruptor de la instalación no abra en el momento de la aparición de alguna falta o perturbación en el sistema.

O nodos lógicos para un sistema fotovoltaico integrado en una DER:

- **Características del array fotovoltaico (DPVA):** Este nodo ofrece información general sobre la composición del array fotovoltaico.
- **Control de la matriz fotovoltaica (DPVC):** Este nodo ofrece los atributos necesarios para poder fijar las consignas de control necesarias en un sistema fotovoltaico.

Estos nodos lógicos a su vez tienen internamente definidos por la norma objetos de datos (DO). Estos objetos de datos están definidos por la norma y representan las informaciones que contiene cada nodo lógico, información que puede ser de obligada aparición (M, *mandatory*), opcional (O, *optional*), obligatoria condicionada (X_X_M) u opcional condicionada (X_X_O). De esta forma, por ejemplo, un nodo lógico que representa características del array fotovoltaico contiene objetos de dato como:

- **Typ:** Tipo de montaje (Matriz, sub-matriz, serie, modulo, planta u otro).
- **ModulCnt:** Número de módulos por serie.
- **StrgCnt:** Número de series en paralelo por sub-matriz.
- **SubArrCnt:** Número de sub-matrices en paralelo por matriz.
- **ArrArea:** Área de la matriz.

A su vez estos objetos de datos están compuestos por los atributos que los definan, los cuales se basan en los tipos de dato que contiene cada objeto de dato.

De esta manera, un nodo lógico se puede ver como una función en un programa de código, al que se le introducen unas entradas y devuelve unas salidas (Figura 9) .



Figura 9: Bloque LN.

Los Nodos Lógicos para sistemas DER vienen definidos en la sección 7-420 del estándar: IEC 61850-7-420.

2.1.5 Dispositivo Lógico

Un dispositivo lógico es la entidad que agrupa un conjunto de funciones de automatización, protección o de otro tipo. Es decir, un LD está formado por diferentes LN. Un LD siempre se implementa en un IED; por lo tanto, los dispositivos lógicos no pueden contener nodos lógicos de diferentes IEDs [16].

De esta forma, el concepto de dispositivo lógico es un paso más en la forma de estructurar y virtualizar el mundo real por parte de la norma IEC 61850. Esto es así ya que el término dispositivo lógico va ligado con el concepto de deslocalización de funciones. Cuando se necesita automatizar una tarea o varias del sistema DER, en este caso, se deben definir las funciones necesarias para realizar dicha tarea. Una vez definidas las tareas, se deben buscar los nodos lógicos necesarios para definir las funciones, y posteriormente, se deberán agrupar dichos nodos lógicos de forma que muchos de ellos compartan información de entrada necesaria y puedan implementarse posteriormente en un mismo dispositivo electrónico inteligente (IED). Según este flujo, siempre se piensa en funciones, no en equipos necesarios para realizar tareas. Los equipos acaban siendo una consecuencia de la agrupación de dichas funciones definidas gracias a los nodos lógicos.

2.1.6 Equipo Electrónico Inteligente

Un IED, como su propio nombre indica es un dispositivo electrónico inteligente que ejecuta nodos lógicos y realiza intercambios de información con otros IEDs a través de los servicios de comunicaciones implementados en la propia subestación eléctrica. De esta forma el IED es la unidad más grande que contiene los dispositivos lógicos y a su vez los nodos lógicos.

2.1.7 PICOM

Un PICOM (*Pices of Information COMmunication*) es un concepto que se describe en la parte de la norma IEC 61850-5 y su función es especificar una transferencia de información entre dos nodos lógicos a través de los parámetros de comunicaciones necesarios como pueden ser requisitos de tiempo, rendimientos o prioridades.

Según [17], los PICOM introducidos por el CIGRE WG34.03 se utilizan para describir la información intercambiada entre los LN. Los componentes o atributos de un PICOM (Figura 10) son:

- Datos referentes al contenido de la información y su identificación según lo requieran las funciones (semántica).
- Conexión lógica que contiene la fuente lógica (nodo lógico emisor, fuente) y el sumidero lógico (nodo lógico receptor, sumidero).
- Tipo que describe la estructura de los datos, es decir, si es un valor analógico o binario, es un valor único o un conjunto de datos, etc.
- Rendimiento que maneja el tiempo de transmisión permitido (definido por la clase de rendimiento), la integridad de los datos y el método o causa de la transmisión (por ejemplo, periódica, impulsada por eventos, a petición).

El PICOM es un concepto clave para las especificaciones de requisitos y de diseño en un sistema digital basado en el estándar IEC 61850.

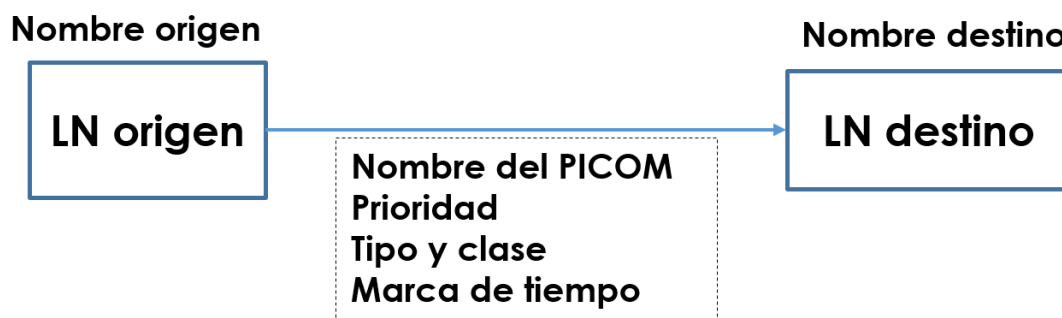
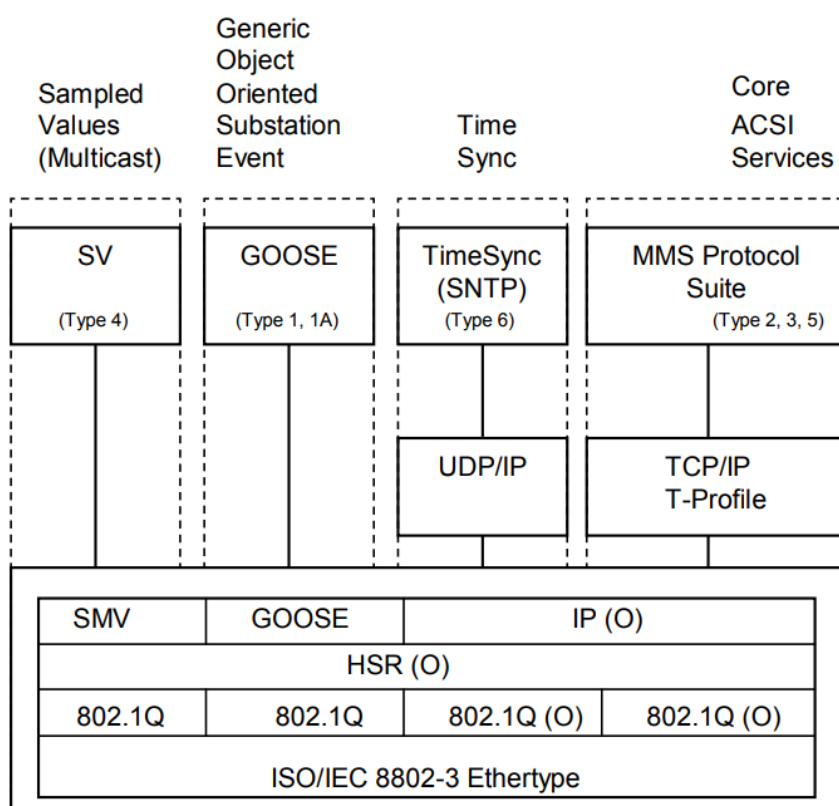


Figura 10: PICOM.

2.1.8 Tipos de mensajes

La norma define diferentes tipos de mensajes. Se selecciona un tipo de mensaje u otro en función de los requisitos especificados en el PICOM para realizar cada intercambio de información.

En la Figura 11 se observa los diferentes tipos de mensajes presentes en la norma IEC 61850, junto con la diferencia en cuanto al esquema de comunicaciones que presentan y su manera de realizar el intercambio de información.



(Type x) is the Message type and performance class defined in IEC 61850-5

IEC 811/11

Figura 11: Tipos de mensajes IEC 61850.

Estos esquemas se basan en el modelo OSI (*Open Systems Interconnection*), creado por la organización internacional para la Estandarización (ISO) en 1984. El objetivo de este modelo es garantizar el intercambio de datos entre dos usuarios finales (ser humano, robot, software, etc.) estén donde estén y con independencia del hardware utilizado. Este modelo se organiza en capas funcionales que van incrementando los servicios conforme se sube de nivel:

CAPAS OSI	
Nivel Aplicación	Servicios de red para aplicaciones
Nivel Presentación	Representación de los datos
Nivel Sesión	Comunicación entre dispositivos de la red
Nivel Transporte	Conexión extrema a extremo y fiabilidad de los datos
Nivel Red	Determinación de ruta e IP (Direccionamiento lógico)
Nivel Enlace	Nivel de enlace de datos (Direccionamiento físico (MAC y LLC))
Nivel Físico	Señal y transmisión binaria

Tabla 1: Capas OSI.

Basándose en el modelo OSI, todos los tipos de mensajes presentes en el estándar IEC 61850 presentan una capa inferior común, la capa Ethernet. De ahí para arriba los diferentes mensajes presentan un esquema de comunicación diferente en función de los requisitos que se les exigen:

- **GOOSE:** La mensajería GOOSE (*Generic Object-Oriented System Event*) es la clave de la norma. Los requisitos de tiempos que garantiza están por debajo de los 10 ms, con lo que son mensajes rápidos que se utilizan para realizar acciones como la apertura o el cierre de un interruptor por disparo de una protección [18].

Por la necesidad de que sean tan rápidos, los mensajes GOOSE pasan de la capa Ethernet a la capa de aplicación directamente, sin introducir capas intermedias para conseguir mensajes lo más rápidos posibles.

Los mensajes GOOSE se basan en el concepto publicador/suscriptor, es decir, un publicador “publica” la información en una dirección *multicast* a la que debe estar suscrito el suscriptor que debe recibir esa información. Aun usando un esquema que no hace uso de la capa de direccionamiento IP (TCP/IP), el modelo de servicio GOOSE del IEC 61850 proporciona la posibilidad de una rápida y fiable distribución de los valores de los datos de entrada y salida [19]. Esto es así ya que usa sistemas de publicación periódica (bombardeo) para intentar garantizar que los mensajes lleguen a su destino. Este sistema funciona de forma que, ante un cambio de valor de algún elemento del *dataset* (se denomina *dataset* a la agrupación de diferentes datos en un mismo conjunto) enviado a través del GOOSE, ocurre el siguiente efecto reflejado en la Figura 12:

- El mensaje GOOSE se publica inmediatamente tras detectar el cambio con el *dataset* actualizado.
- Pasado un tiempo T_1 , el mensaje GOOSE se publica de nuevo. Pasado de nuevo un tiempo T_1 , el mensaje GOOSE se publica otra vez.
- Se van publicando retransmisiones aumentando el tiempo entre envíos hasta alcanzar el tiempo T_0 .

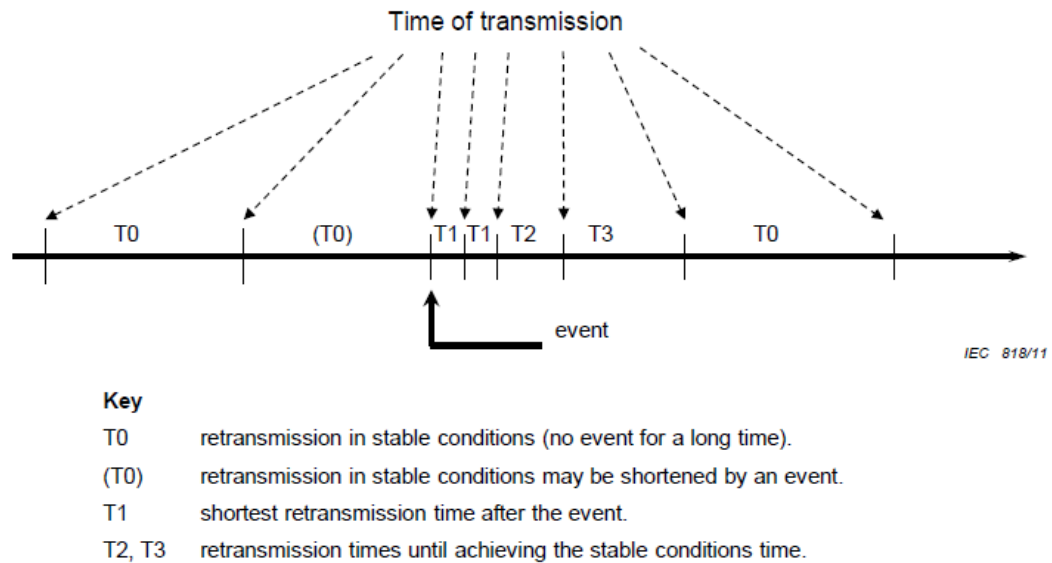


Figure 8 – Transmission time for events

Figura 12: Esquema de transmisión de mensajes GOOSE [12].

Debido al servicio de bombardeo con el objetivo de muestrear y garantizar el correcto estado del canal de comunicaciones, el servicio de mensajería GOOSE de la norma IEC 61850 define dos parámetros para poder distinguir si el mensaje recibido en un instante determinado es un mensaje GOOSE nuevo que debe ser procesado, o únicamente es un mensaje GOOSE de repetición cada T_0 para muestrear el canal de comunicaciones. Estos parámetros son el **stNum** y el **sqNum**:

- o **stNum**: Este valor muestra el número de secuencia del número de cambios detectado. Es decir, ante un cambio en el *dataset* (conjunto de datos) del mensaje GOOSE, el valor stNum se incrementa en un valor.
- o **sqNum**: Este valor muestra el número de secuencia del mensaje GOOSE dentro del actual stNum. Es decir, si una vez que se detecta el cambio de *dataset* y se incrementa el stNum, se repite el envío periódico del mismo *dataset*, por cada nuevo envío se incrementa en una unidad el valor sqNum hasta detectar un nuevo mensaje GOOSE donde se haya modificado el *dataset*, en este caso, se

reinicia a cero el valor del sqNum, y se incrementa nuevamente en una unidad el valor del parámetro stNum.

- **Sampled Value (SV):** Los valores muestreados son también un tipo de mensajes muy exigentes en cuanto a requerimientos de tiempos. Al igual que los mensajes GOOSE pasan de la capa Ethernet a la capa aplicación directamente.

La exigencia de este tipo de mensajes radica en la necesidad de una cadencia muy elevada a la hora de enviar los valores, en el caso de que estos procedan de un muestreo a una elevada frecuencia.

Al igual que los mensajes GOOSE los *Sampled Value* se envían a una dirección *multicast*.

- **Mensajes MMS:** Estos mensajes presentan una capa intermedia entre la capa Ethernet y la capa aplicación. En este caso se trata de una capa TCP/IP que introduce mayor fiabilidad a la hora de garantizar que el mensaje ha llegado al destino, pero a la vez introduce mayores tiempos de retardo. Los mensajes MMS (*Manufacturing Message Specification*) tienen un tiempo de retardo entono a 500ms.

A diferencia de los dos mensajes anteriores, estos mensajes se basan en el concepto Cliente/Servidor, de manera que, la información se envía a una dirección IP concreta, la del equipo servidor.

- **TimeSync:** Este tipo de mensajes informan sobre tiempos o marcas de sincronización. A diferencia de los MMS, en este tipo de mensajes se emplea el protocolo UDP/IP. La diferencia reside en que estos últimos suelen ser usados para sincronizar la mayoría de los equipos de un sistema por lo que para reducir la saturación de la red de comunicaciones se decide emplear la capa UDP/IP en vez de TCP/IP aun teniendo requisitos de tiempos semejantes a los mensajes MMS.

2.1.9 Requerimientos de comunicaciones

Se selecciona un servicio (tipo) de mensaje u otro en función de los requerimientos y las exigencias en cuanto a tiempos de transferencia de información de la aplicación.

El tiempo de transferencia completo t (Figura 13 y Figura 14), se especifica en la norma como el tiempo de transmisión completo de un mensaje incluyendo la gestión de la comunicación en ambos extremos (emisor, receptor). El tiempo se mide desde el momento en que el emisor pone el contenido de los datos de aplicación en la parte superior de su pila de transmisión (codificación y envío) hasta el momento en que el receptor extrae los datos de su pila de transmisión (recepción y decodificación). Estos procesos de codificación/decodificación pueden ser manejados por el procesador principal o por un procesador de comunicación dedicado (fuera del ámbito de la norma). Este requisito es independiente de la pila seleccionada en otras partes de la serie.

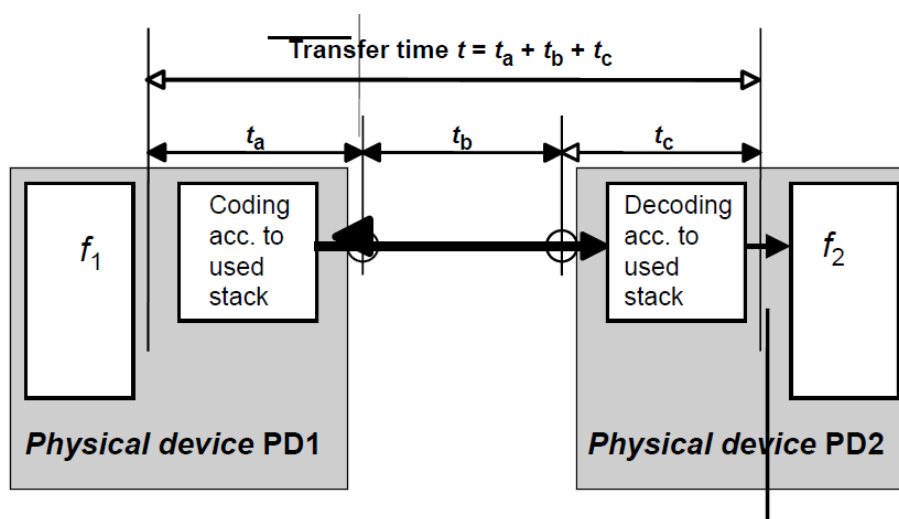


Figura 13: Definición tiempo de transferencia en sistemas IEC 61850.

Para definir el rendimiento de un sistema, además del tiempo de transferencia, es importante el tiempo de procesamiento. Un ejemplo es una función de protección distribuida que consiste en dos dispositivos físicos que tienen la corriente y/o el voltaje como entrada y el disparo como salida. Dado que la comunicación es importante, pero es solo una función auxiliar, el tiempo de transferencia será normalmente pequeño comparado con los tiempos de procesamiento: $t \ll t_{f1} + t_{f2}$.

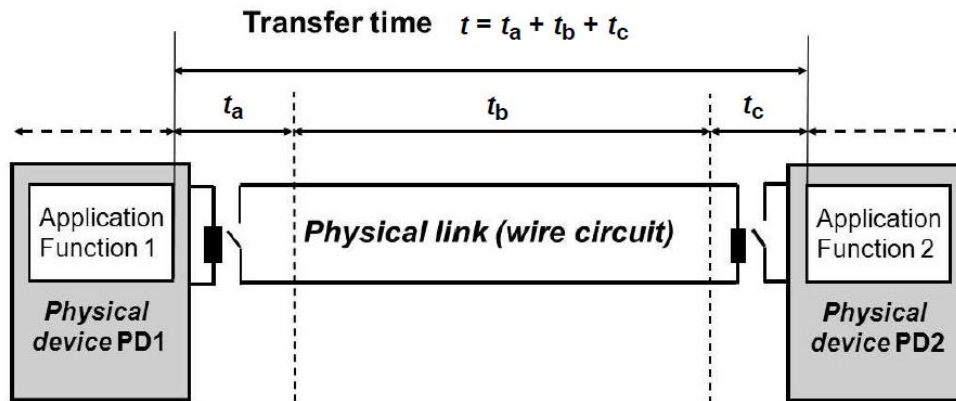


Figura 14: Tiempo de transferencia para señal binaria.

Los requisitos de comunicación vienen definidos por las interfaces que se seleccionan para realizar dichos intercambios de información. Esta parte viene definida en la sección 5 de la norma, IEC 61850-5.

Como se ha comentado, la norma IEC 61850 fue creada inicialmente con el objetivo de definir los requisitos necesarios para el desarrollo de sistemas de automatización de subestaciones (SAS). En este tipo de subestaciones, las operaciones que se desarrollan se sitúan en tres niveles lógicos diferentes: nivel de estación, nivel de bahía (posición) y nivel de proceso. Para dicha arquitectura, la norma IEC 61850 define once interfaces (Figura 15) que permiten la comunicación entre los tres niveles, entendiendo por interfaz cada parte de la organización lógica de las interacciones entre elementos del sistema.

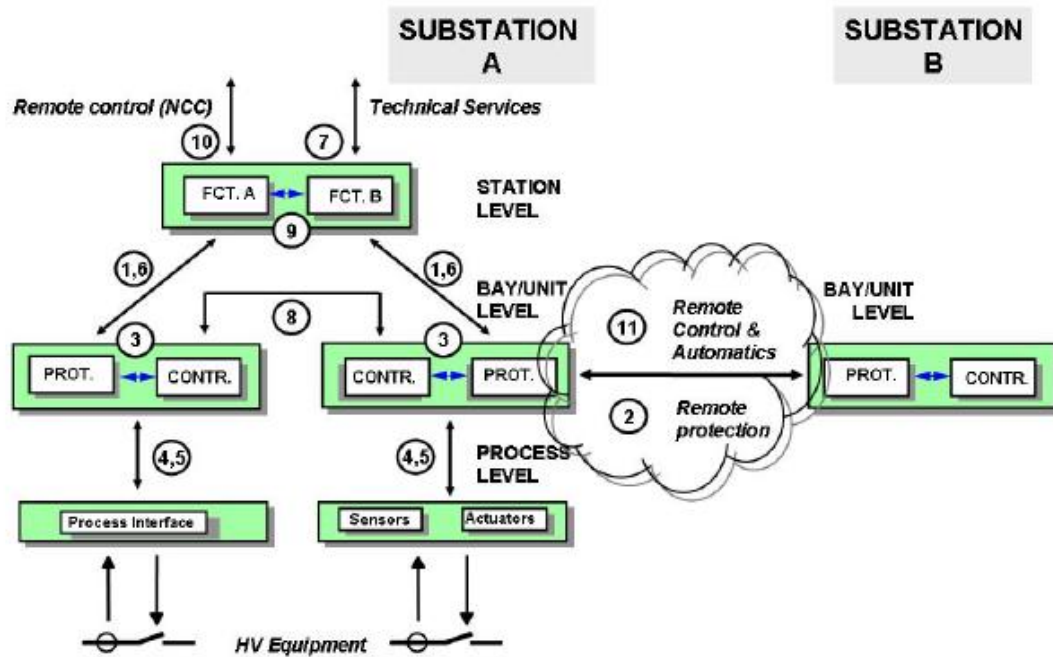


Figura 15: Niveles y lógicas de interfaces en SAS.

De esta forma, las diez interfaces que define la norma son:

- **IF1:** Intercambio de datos de protección entre nivel de bahía y nivel de estación.
- **IF2:** Intercambio de datos de protección entre nivel de bahía y protección remota.
- **IF3:** Intercambio de datos entre nivel de bahía.
- **IF4:** Intercambio de datos analógicos entre nivel de proceso y nivel de bahía.
- **IF5:** Intercambio de datos entre nivel de proceso y nivel de estación.
- **IF6:** Intercambio de datos entre nivel de bahía y nivel de estación.
- **IF7:** Intercambio de datos entre nivel de subestación y centro de control remoto.
- **IF8:** Intercambio de datos entre niveles de bahía, especialmente para funciones rápidas como “*interlocking*”.
- **IF9:** Intercambio de datos con nivel de estación.
- **IF10:** Intercambio de datos de control entre la subestación y un centro de control remoto.
- **IF11:** Intercambio de datos de control entre subestaciones.

Además de las interfaces, la parte de la norma IEC 61850-5 clasifica los requisitos de rendimiento (tiempos de transferencia) y asigna las clases a las interfaces. La norma define 7 clases para tiempos de transferencia, donde el menor tiempo de transferencia admitido por la norma es de 3 ms y equivale a la clase TT6. En la siguiente tabla se muestran todas las clases definidas por la norma:

Transfer time class	Transfer time [ms]	Application examples: Transfer of
TT0	>1 000	Files, events, log contents
TT1	1 000	Events, alarms
TT2	500	Operator commands
TT3	100	Slow automatic interactions
TT4	20	Fast automatic interactions
TT5	10	Releases, status changes
TT6	3	Trips, blockings

Tabla 2: Clases para tiempos de transferencia.

Por último, la norma define unos tipos de mensaje junto con sus clases de rendimiento y las interfaces típicas asociadas. La norma define 6 tipos de mensaje:

- **Tipo 1 – Mensajes rápidos (“Protección”)**

- **Tipo 1A “Trip”**

El disparo es el mensaje rápido más importante de la subestación, y los requerimientos de tiempos son:

Performance class	Requirement description	Transfer time		Typical for Interface (IF ^a)
		Class	ms	
P1	The total transmission time shall be below the order of a quarter of a cycle (5 ms for 50 Hz, 4 ms for 60 Hz).	TT6	≤ 3	3,5,8
P2	The total transmission time shall be in the order of half a cycle (10 ms for 50 Hz, 8 ms for 60 Hz).	TT5	≤ 10	2,3,11

^a Interfaces according to Figure 2.

Tabla 3: Requerimientos mensajes tipo 1A.

- **Tipo 1B “Otros”**

Otros mensajes rápidos:

Performance class	Requirement description	Transfer time		Typical for Interface (IF)
		Class	ms	
P3	The total transmission time shall be the order of one cycle (20 ms for 50 Hz, 17 ms for 60 Hz).	TT4	≤ 20	2,3,8,11

Tabla 4: Requerimientos mensajes tipo 1B.

- **Tipo 2 – Mensajes de velocidad media**

Se trata de mensajes en los que la hora en la que se originó el mensaje es importante, pero en los que la hora de transmisión es menos crítica.

Performance class	Requirement description	Transfer time		Typical for Interface (IF)
		Class	ms	
P4	The transfer time for automation functions is less demanding than protection type messages (trip, block, release, critical status change) but more demanding than operator actions.	TT3	≤ 100	2,3,8,9,11

Tabla 5: Requerimientos mensajes tipo 2.

- **Tipo 3 – Mensajes de velocidad baja**

Este tipo incluye mensajes completos que pueden requerir ser etiquetados en el tiempo.

Performance class	Requirement description	Transfer time		Typical for Interface (IF)
		Class	ms	
P5	The total transmission time shall be half the operator response time of ≥ 1 s regarding event and response (bidirectional)	TT2	≤ 500	1, 3, 4, 5, 6, 7, 8, 9, 10
P6	The total transmission time shall be in line with the operator response time of ≥ 1 s regarding unidirectional event	TT1	≤ 1 000	1, 3, 4, 5, 6, 7, 8, 9, 10

Tabla 6: Requerimientos mensajes tipo 3.

- **Tipo 4 – Mensajes de datos en bruto (“Samples”)**

Este tipo de mensajes incluye los datos de salida de la digitalización:

Performance class	Requirement description	Transfer time		Typical for Interface (IF)
		Class	ms	
P7 ^a	Delay acceptable for protection functions using these samples	TT6	≤ 3	4,8
P8 ^b	Delay acceptable for other functions using these samples	TT5	≤ 10	2,4,8
^a equivalent to P1.				
^b equivalent to P2.				

Tabla 7: Requerimientos *mensajes* tipo 4.

- **Tipo 5 – Funciones de transferencia de ficheros**

Este tipo de mensajes es usado para la transferencia de largos ficheros de datos.

Performance class	Requirement description	Transfer time		Typical for Interface (IF)
		Class	ms	
P9	Transfer times for files are not critical. Typically, files with process data are used either for post-mortem analysis or for off-line statistics. Files with configuration data require a careful installation and check process. Therefore, no quick operator action of about 1 s is requested. Therefore, 10 000 ms fit very well the file transfer requirements.	TT0	≤ 10 000	1, 4, 5, 6,7,10

Tabla 8: Requerimientos mensajes tipo 5.

- **Tipo 6 – Mensajes de comando y transferencia de ficheros con control de acceso**

Este tipo de mensajes es usado para transferir órdenes de control.

Performance class	Requirement description	Transfer time		Typical for Interface (IF)
		Class	ms	
P10 ^a	Type 3.P5 message with access control: The total transmission time shall be half the operator response time of ≥ 1 s regarding event and response (bidirectional)	TT2	≤ 500	1,3, 4, 5, 6, 7, 8, 9, 10
P11 ^b	Type 3.P6 message with access control: The total transmission time shall be in line with the operator response time of ≥ 1 s regarding unidirectional event	TT1	≤ 1 000	1, 3, 4, 5, 6, 7, 8, 9, 10
P12 ^c	Type 5 message with access control: Transfer times for files are not critical. Typically, the time requirements are in the order of the operator response time (≥ 1 s) or of archives for post-mortem analysis (>> 1 s).	TT0	≤ 10 000	1, 4, 5, 6,7,10
^a equivalent to P5. ^b equivalent to P6. ^c equivalent to P9.				

Tabla 9: Requerimientos mensajes tipo 6.

2.1.1 Modo Test

Una de las funcionalidades que presenta y describe la norma es el modo de pruebas. Esta funcionalidad es interesante tenerla en cuenta antes de iniciar el diseño de un sistema basado en comunicaciones IEC 61850.

La parte del modo de pruebas se describe en la parte de la norma IEC 61850-7-1, concretamente en la sección 7.8 “Data used for testing” [16]. Con el objetivo de permitir la realización de pruebas funcionales, de puesta en marcha o incluso de mantenimiento, los sistemas basados en IEC 61850 deben ofrecer las siguientes facilidades:

- A **nivel de IED**, la opción de recibir señales simuladas en vez de la señal real. Respecto a este punto y observando la Figura 16, la norma expone que un dispositivo lógico puede procesar información en modo simulado o no. Para ello, si el IED tiene activo el dato *Sim.stVal* del LN LPHD1, este recibe y procesa la información con el bit de simulación activo prioritariamente. En el caso de que el dato *Sim.stVal* del LN LPHD1 se encuentre en "False", el IED no procesa información con el bit de simulación activo. Que el dato *Sim.stVal* este activo no implica que el IED solo procese información en modo simulado, de hecho, en la Figura 16 se observa como los GOOSE 2 y GOOSE 3 no tienen el bit de simulación activo por lo que sigue procesándose el mensaje actual; sin embargo, por la parte del GOOSE 1 recibe un mensaje simulado y otro no, en este caso al estar el IED con el dato *Sim.stVal* activo, solo procesa el mensaje GOOSE 1 simulado.

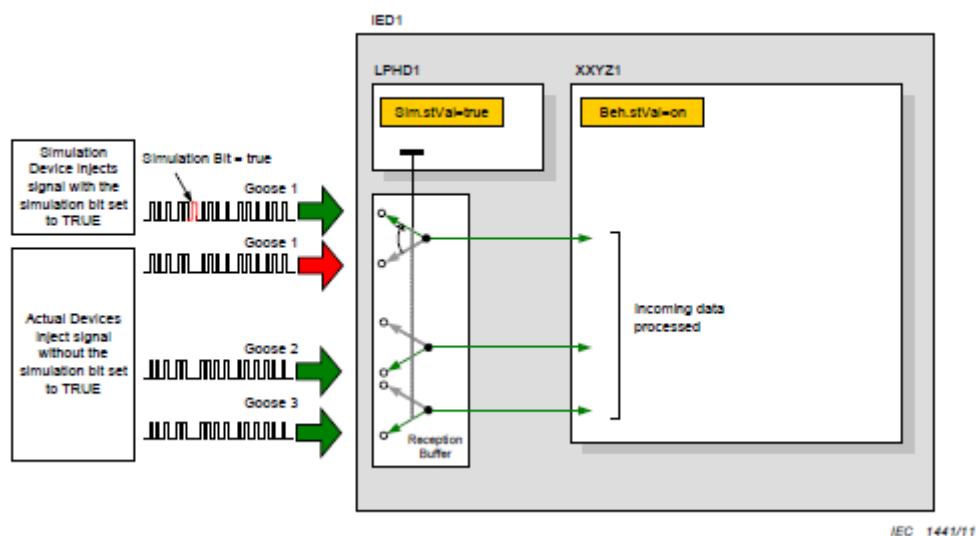


Figura 16: Datos utilizados para la recepción de señales de simulación [16].

- A **nivel de LN (Nodo Lógico)**, la opción de recibir señales de entrada en modo test en vez de las señales reales. Al igual que en el caso anterior, en el caso de que el LN tenga activo el atributo "testEna" procesa la información con el modo test activo. Esta parte es muy útil ya que como al igual que se observa en la Figura 17, puede ser de utilidad para realizar unas pruebas puntuales tener la opción de recibir la información simulada a través de una entrada digital (*GGIO1*) en vez del valor recibido real desde otro LN.

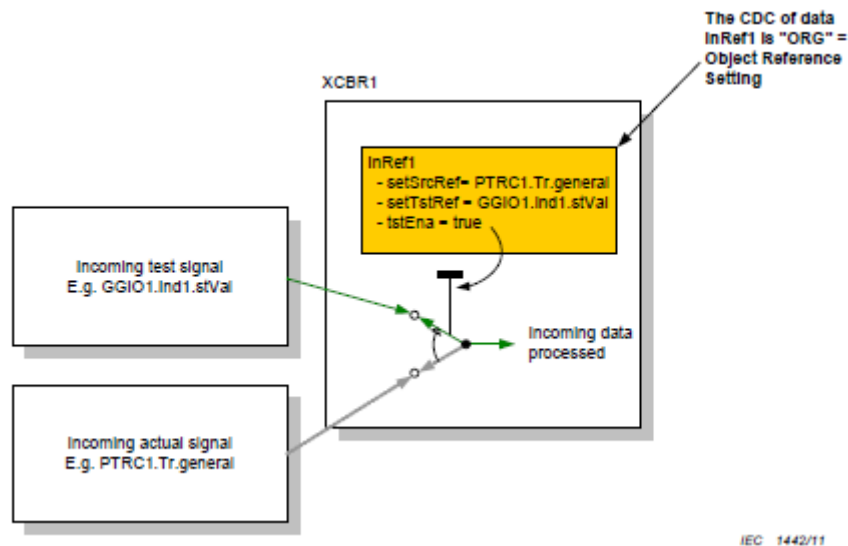


Figura 17: Ejemplo de señales de entrada en modo test [16].

- **A nivel de LN (Nodo Lógico)**, la opción de modificar el estado de una función o un grupo de funciones a modo test (Figura 18). Concretamente un nodo lógico puede ponerse en uno de los siguientes estados [20]:
 - **ON:** Equipo en servicio y operativo. Admite información con el bit de test desactivado.
 - **TEST:** Equipo en pruebas. Publica toda la información con el bit de calidad test activado. Relés de salida operativos. Admite información con el bit de test en cualquier estado.
 - **ON-BLOCKED:** ON pero con los relés de salida bloqueados.
 - **TEST-BLOCKED:** TEST con los relés de salida bloqueados.
 - **OFF:** Equipo fuera de servicio, a nivel de servicios IEC 61850.

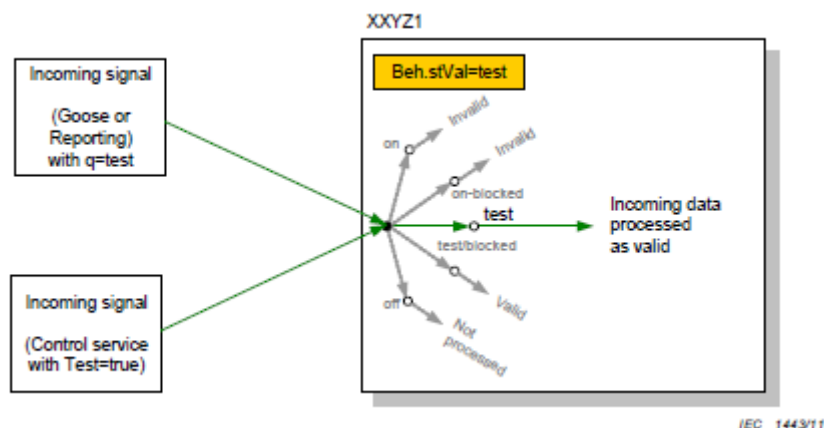


Figura 18: Ejemplo modo test [16].

Dado que el dispositivo lógico controla todos los nodos lógicos que lo forman, el modo del dispositivo lógico ("LDMode" = LLN0.Mod) y el modo de un nodo lógico específico ("LNMode" = XXXX.Mod) están relacionados. El comportamiento de un nodo lógico es, por tanto, una combinación de LLN0.Mod y XXXX.Mod y está descrito en "LNBeh" = XXXX.Beh (Figura 19). Este objeto de datos es de sólo lectura y tiene los mismos valores posibles que Mod. El valor se determina según la siguiente tabla [20]:

LNMode XXXX.Mod	LDMode LLN0.Mod	LNBeh (read only) XXXX.Beh	LNBeh Value
on	on	on	1
on	on-blocked	on-blocked	2
on	test	test	3
on	test/blocked	test/blocked	4
on	off	off	5
on-blocked	on	on-blocked	2
on-blocked	on-blocked	on-blocked	2
on-blocked	test	test/blocked	4
on-blocked	test/blocked	test/blocked	4
on-blocked	off	off	5
test	on	test	3
test	on-blocked	test/blocked	4
test	test	test	3
test	test/blocked	test/blocked	4
test	off	off	5
test/blocked	on	test/blocked	4
test/blocked	on-blocked	test/blocked	4
test/blocked	test	test/blocked	4
test/blocked	test/blocked	test/blocked	4
test/blocked	off	off	5
off	on	off	5
off	on-blocked	off	5
off	test	off	5
off	test/blocked	off	5
off	off	off	5

Figura 19: Definición del comportamiento de un LD (LN Beh) [20].

2.1.2 Modelo de Datos

La digitalización o virtualización de la subestación se debe realizar a través de un lenguaje y un modelo de datos que pueda ser introducido e interpretado por los diferentes IEDs, para saber de qué equipos consta la subestación y con qué equipos se deben realizar los intercambios de información.

Para ello, aparece el modelo de datos. El modelo de datos describe en SCL (*Structure Control Language*) los equipos por los que está compuesta la subestación de una forma estructurada según la Figura 8.

De esta forma, un modelo de datos real queda como en la Figura 20:

```

Nombre del Equipo electrónico inteligente <IED name="PRUEBA" type="" manufacturer="" configVersion="1.0">
  <AccessPoint name="AccessPoint">
    <Server>
      <Authentication />
Dispositivo Lógico (LD) <LDevice desc= "LD1" inst="Management" ldName="Management">
  <LN0 lnType="LLN0_TYPE" lnClass="LLN0" inst="" />
  <LN lnType="LPHD_TYPE" lnClass="LPHD" inst="1" />
Nodos Lógicos (LN) <LN lnType="DCRP_TYPE" lnClass="DCRP" inst="1" prefix="" />
  <LN lnType="DOPR_TYPE" lnClass="DOPR" inst="1" prefix="" />
  <LN lnType="DOPM_TYPE" lnClass="DOPM" inst="1" prefix="" />
  <LN lnType="DPST_TYPE" lnClass="DPST" inst="1" prefix="" />
  .....
</LDevice>

```

Figura 20: Modelo de datos.

Para la creación del modelo de datos existen herramientas como SCL NAVIGATOR de la empresa TRIANGLE MICROWORKS [21] que permiten crear este modelo de una forma automática una vez que se definen los elementos por los que está compuesto el modelo de datos deseado (Figura 21).

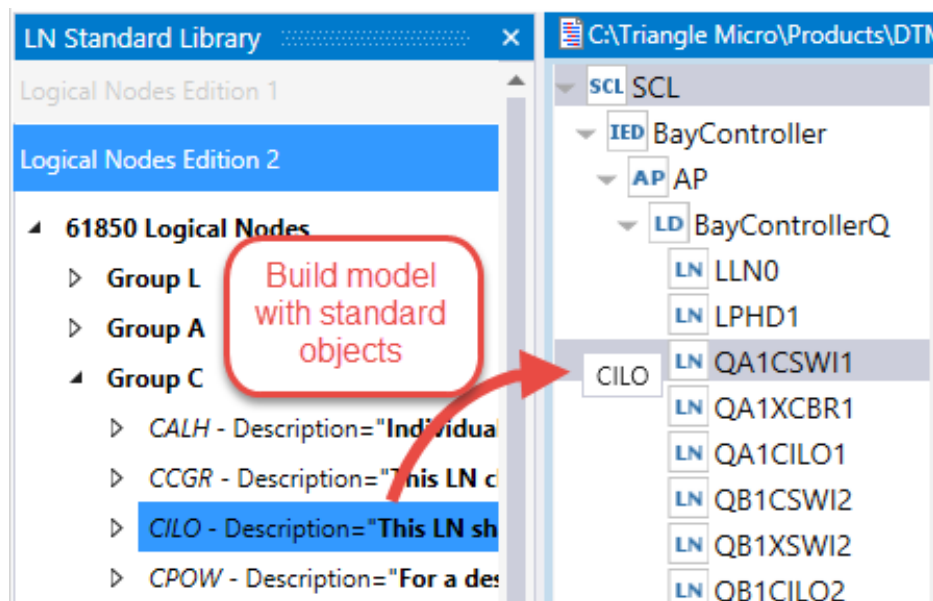


Figura 21: SCL Navigator.

2.1.1 Seguridad del sistema

Aunque este trabajo no aborda la seguridad del sistema, hoy en día en el desarrollo e implementación de sistemas digitales en sistemas de energía, la ciberseguridad toma una importancia vital [22].

Anteriormente, debido a que los protocolos de comunicación en sistemas eléctricos eran muy especializados y diferentes, se asumía que el enfoque de “seguridad por oscuridad” (*security by obscurity*) era suficiente. Sin embargo, con la estandarización de los protocolos de comunicación como es el caso del IEC 61850, esta presunción ya no es válida [23]. De esta necesidad aparece el IEC 62351. La norma IEC 62351 al igual que el IEC 61850 es un estándar desarrollado por el grupo de trabajo 15 (WG15) del comité técnico de la Comisión Electrotécnica Internacional 57 (IEC TC57). Este estándar se publica para proporcionar recomendaciones de seguridad para diferentes protocolos de comunicación en sistemas eléctricos [24]. La norma se desarrolla con el objetivo del manejo de la seguridad del TC57 serie de protocolos, incluyendo:

- IEC 60870-5
- IEC 60870-6
- **IEC 61850**
- IEC 61970
- IEC 61968

En la norma IEC 62351 se especifican diferentes objetivos de seguridad incluyendo la autenticación de la transferencia de datos a través de firmas digitales, asegurando solo el acceso autenticado, prevención de escuchas ilegales, prevención de reproducción y suplantación de identidad, y detección de intrusos. Además, el IEC 62351 discute sobre la vulnerabilidad de ciberseguridad de la comunicación IEC 61850 y los esquemas recomendados para mitigarlos.

Aunque la norma ofrezca una lista de pautas y desarrolle un marco para la operación segura de un sistema basado en comunicaciones IEC 61850, su implementación en escenarios de operaciones reales, como en la gestión de plantas DER, está abierta a interpretación [23]. Por lo tanto, a la hora de implementar un sistema como el descrito en este trabajo, se deben seguir las pautas expuestas en la norma IEC 62351, especialmente en las partes:

- IEC 62351-4: Seguridad para cualquier perfil, incluido MMS.
- IEC 62351-6: Seguridad para perfiles IEC 61850.

CAPÍTULO 3: DESCRIPCIÓN DEL MATERIAL UTILIZADO

En este capítulo se procede a realizar una descripción del entorno de desarrollo empleado en este trabajo.

3.1.1 Ordenador basado en el sistema operativo Linux

El ordenador que se ha utilizado es un ASUS con procesador Intel Core i7-4710HQ a 4.20 GHz de 64 bits y 8GB de memoria RAM.

Para facilitar el uso de las librerías y las comunicaciones necesarias para implementar el estándar de comunicaciones IEC 61850, el sistema operativo bajo el que trabaja el ordenador es Ubuntu 20.04.1 LTS. LTS significa soporte a largo plazo, (cinco años, hasta abril de 2025, con actualizaciones gratuitas de seguridad y mantenimiento garantizadas).

3.1 Biblioteca libiec61850

La biblioteca libiec61850 [25] proporciona una implementación de IEC 61850 además del protocolo MMS en lenguaje C. También proporciona soporte para la comunicación dentro de la subestación a través de GOOSE. El objetivo de este proyecto es proporcionar una implementación que sea portátil y que pueda ejecutarse en sistemas empujados y microcontroladores. También se proporciona un conjunto de ejemplos sencillos que se pueden utilizar como punto de partida para aplicaciones propias. La biblioteca también contiene un contenedor .NET para permitir su uso en lenguajes de alto nivel como C #.

Esta implementación se ejecuta en sistemas empujados, sistemas Linux integrados, así como en computadoras de escritorio que ejecutan Linux, Windows o MacOS.

Se proporcionan las siguientes características:

- Pila de protocolo ISO completa sobre TCP / IP.
- Implementación estática del modelo IED mediante la generación de código C a partir del archivo SCL.
- Creación dinámica del modelo IED mediante llamadas a API o mediante archivo de configuración.
- Servicio de asociación (con autenticación de contraseña).
- Servicios de lectura y escritura para variables MMS simples y complejas.

- Servicios de exploración de modelos (GetServerDirectory, GetDeviceDirectory...).
- Servicios de conjuntos de datos, incluida la creación y eliminación dinámicas de conjuntos de datos.
- API de cliente y servidor IEC 61850.
- API de cliente MMS simple (p. Ej., Para usarse con TASE.2).
- Servicio de informes con búfer y sin búfer.
- Servicio de registro.
- Soporte para configurar servicios de control de grupo.
- Todos los modelos de control IEC 61850 son compatibles con el servidor y el cliente.
- Código de suscriptor y editor de GOOSE que también se puede utilizar de forma independiente.
- Código de suscriptor y editor IEC 61850-9-2 (Sampled Values) que se puede utilizar de forma independiente.
- Capa de abstracción de hardware e implementaciones para POSIX (Linux), WIN32, sistemas de estilo BSD (Mac OS X, FreeBSD).
- Herramienta de conversión para traducir archivos SCL a modelos de IED estáticos.
- Herramienta de conversión para traducir archivos SCL a archivos de configuración del servidor.
- Todos los servicios de archivos (exploración de directorios, getFile, setFile, deleteFile ...
- También es adecuado para su uso en sistemas integrados con limitaciones de recursos.
- API de cliente C # para aplicaciones que se ejecutan en .NET y Mono.
- Listo para la certificación UCA de los dispositivos de la edición 1 y la edición 2.

3.1.1 Tarjeta de desarrollo Raspberry Pi

Para la implementación del servidor IEC 61850 se hace uso de un de una Raspberry Pi 3 Model B+ (Figura 22). Raspberry Pi es un micro-ordenador, ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo coste desarrollado en Reino Unido por la *Raspberry Pi Foundation*. Aunque la primera idea era desarrollar una placa de bajo coste al alcance de todo el mundo, con el principal objetivo de que fuese usada para la enseñanza de informática en las escuelas, ha acabado siendo más popular de lo esperado y se usa para aplicaciones más complejas, incluso en la robótica.



Figura 22: Raspberry Pi 3 Model B+.

En este trabajo se hace uso concretamente del modelo Raspberry Pi 3 Model B+, este modelo fue lanzado en marzo de 2018 con las siguientes características:

- 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).

3.1.2 Microcontrolador tipo ESP-32

El microcontrolador ESP-32 es un chip SoC (Sistema en chip, del inglés *system on chip*) de bajo costo y consumo de energía, con tecnología Wi-Fi y Bluetooth.

- Procesador:
 - CPU: microprocesador de 32-bit Xtensa LX6 de doble núcleo (o de un solo núcleo), operando a 160 o 240 MHz y rindiendo hasta 600 DMIPS
 - Co-procesador de ultra baja energía (ULP)
- Memoria: 520 KiB SRAM
- Conectividad inalámbrica:
 - Wi-Fi: 802.11 b/g/n
 - Bluetooth: v4.2 BR/EDR y BLE
- Interfaces periféricas:
 - 12-bit SAR ADC de hasta 18 canales
 - 2 × 8-bit DACs
 - 10 × sensores de tacto (sensores capacitivos GPIOs)
 - 4 × SPI
 - 2 × interfaces I²S
 - 2 × interfaces I²C
 - 3 × UART
 - Controlador host SD/SDIO/CE-ATA/MMC/eMMC
 - Controlador esclavo SDIO/SPI

- Interfaz Ethernet MAC con DMA dedicado y soporte para el protocolo IEEE 1588 Precision Time Protocol
- Bus CAN 2.0
- Controlador remoto infrarrojo (TX/RX, hasta 8 canales)
- Motor PWM
- LED PWM (hasta 16 canales)
- Sensor de efecto Hall
- Pre-amplificador analógico de ultra baja potencia
- Seguridad:
 - Soporta todas las características de seguridad estándar de IEEE 802.11, incluyendo WFA, WPA/WPA2 y WAPI
 - Arranque seguro
 - Cifrado flash
 - 1024-bit OTP, hasta 768-bit para clientes
 - Criptografía acelerada por hardware: AES, SHA-2, RSA, criptografía de curva elíptica (ECC), generador de números aleatorios (RNG)
- Administración de energía:
 - Regulador interno de baja caída
 - Dominio de poder individual para RTC
 - Corriente de 5 μ A en modo de suspensión profundo
 - Despierta por interrupción de GPIO, temporizador, medidas de ADC, interrupción por sensor de tacto capacitivo

3.1.3 Software de programación para microcontroladores

Para la programación del microcontrolador ESP-32, se hace uso del entorno de desarrollo Arduino IDE (Figura 23). Arduino IDE (*Integrated Development Environment* o Entorno de Desarrollo Integrado) es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además, Arduino, incorpora herramientas software que permite a los programadores desarrollar y cargar programas ya compilados para el microcontrolador ESP-32 en la memoria flash a través del puerto serie, lo cual resulta de gran ayuda en el desarrollo del programa. Además de ser un entorno

bastante sencillo e intuitivo, permite al usuario la integración de librerías en el IDE que contienen programas ya desarrollados y que facilitan la programación del ESP-32.



Figura 23: Arduino IDE.

3.1.4 Software de programación para tarjetas de desarrollo.

Para la compilación y el desarrollo del programa basado en el estándar IEC 61850 a través de la librería libiec61850, se hace uso del editor de texto Geany (Figura 24). Geany es un editor de texto pequeño y ligero de interfaz gráfica de usuario (GUI) basado en Scintilla y bibliotecas gráficas GTK. Incluye características básicas de IDE (Entorno de desarrollo integrado), y está diseñado para tener tiempos de carga reducidos, con poca dependencia en paquetes o bibliotecas externas en Linux.



Figura 24: Geany.

3.1.5 Unidad de generación distribuida

La planta DER ha sido implementada en la terraza oeste de la planta tercera de la Escuela Politécnica Superior de Alcalá, ubicada en el Campus Científico-Tecnológico de la misma (Figura 25).



Figura 25: Ubicación planta DER.

La planta DER implementada en el proyecto Helios tiene como objetivo ofrecer una solución para la alimentación y la gestión de sistemas BTS (*Base Transceiver Station* o estación base). Las BTSs son instalaciones de radio para la comunicación media, baja o alta bidireccional. Normalmente se usan para conectar radios de baja potencia, como por ejemplo teléfonos móviles.

Para ello, la planta Helios (Figura 26) ofrece una MG basada en una instalación fotovoltaica y almacenamiento inteligente implementando métodos para producir de manera eficiente pronósticos de energía fotovoltaica precisos intradía utilizando información disponible y gratuita [26]. Además de estos métodos el proyecto Helios estudia enfoques de pronóstico de irradiancia híbrido, orientados especialmente para cuando se trabaja con conjuntos de irradiancia históricos exigüos o inexistentes. Este enfoque se basa en redes neuronales artificiales (ANN) [27].

La planta DER consta de un sistema fotovoltaico con control de punto de máxima potencia (MPPT) y un bus de corriente continua (CC) conectado al rectificador de la BTS y a las cargas fijas. Este bus de CC actúa como bus principal para la MG al que se conectan distintos elementos [28].

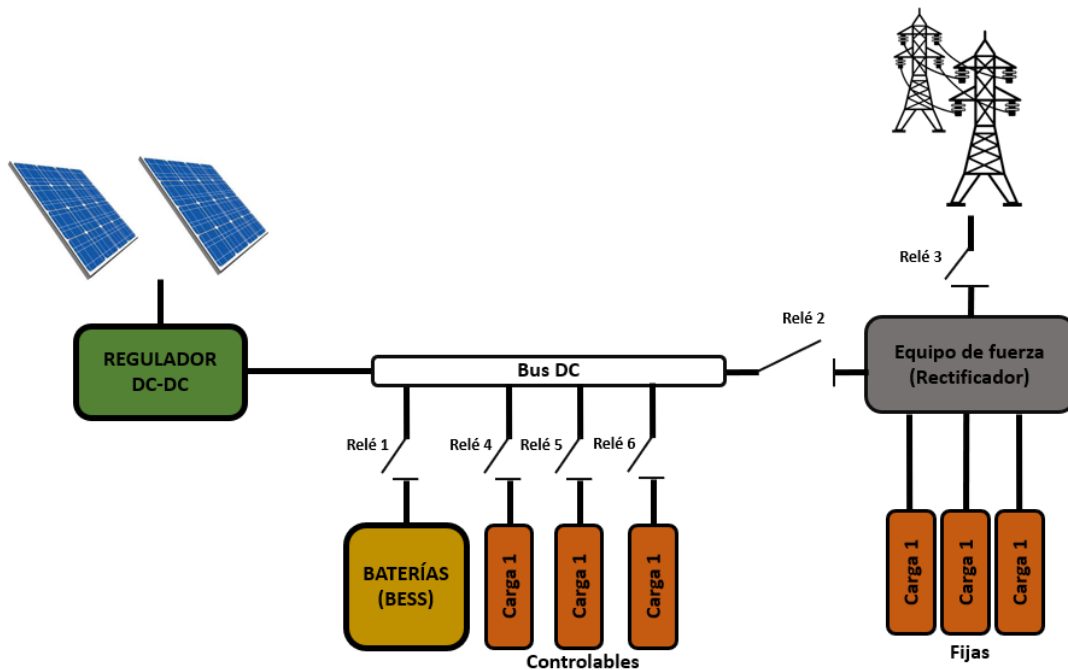


Figura 26: Esquema planta HELIOS.

Como se observa en la Figura 26 el sistema de baterías (BESS) y las cargas no críticas (estas son las cargas controlables ya que se permite su conexión y desconexión para la regulación de la planta) se conectan directamente al bus de CC sin regulador DC/DC. Este es el diseño adoptado por la mayoría de las empresas con el objetivo de maximizar la rentabilidad a corto plazo, ya que un diseño más sofisticado tecnológicamente aumentaría considerablemente los costes de la instalación. Por otro lado, se conecta al bus CC a través de un regulador DC/DC el sistema fotovoltaico y el equipo de fuerza (rectificador) el cual a su vez está conectado con las cargas fijas (BTS) y la red eléctrica.

A continuación, se procede a definir las características principales de cada elemento presente en la planta DER:

- **Paneles solares:** Cada panel solar produce $330 W_p$, que, con un total de 9 paneles instalados, ofrecen $2970 W_p$. Se han instalado con orientación Sur 0° e inclinación 30° (por motivos estructurales). Se disponen en 3 filas de 3 paneles en serie (marcado por el voltaje impuesto por el regulador) En meses de invierno envía una potencia máxima de 2.1 kW al sistema.



Figura 27: Paneles solares planta Helios.

- **Regulador DC-DC:** El regulador envía P_{solar} al bus DC. La selección del equipo se debe a aspectos económicos ($3.2 kW \approx 200 \text{ €}$) además de cumplir con los requerimientos necesarios del sistema y de monitorización del mismo.

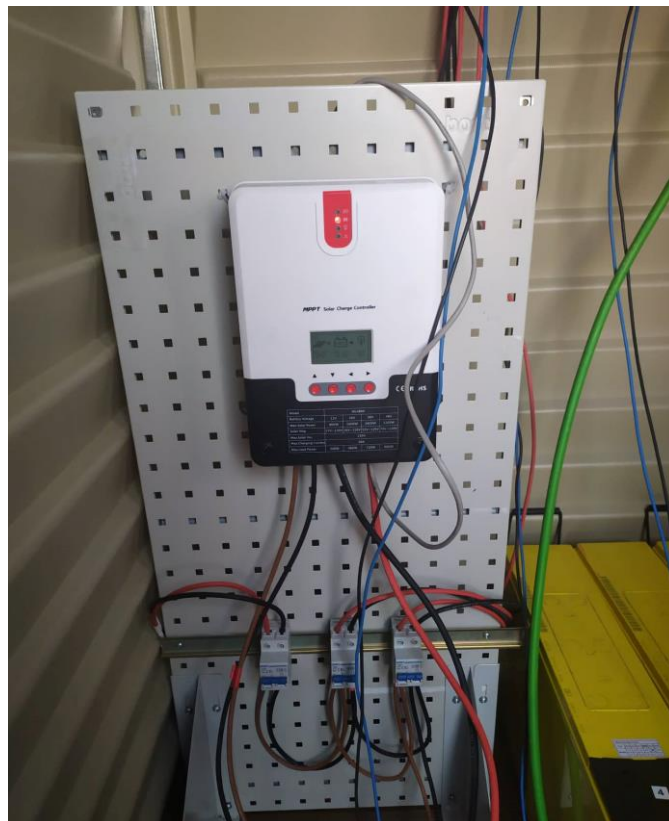


Figura 28: Regulador DC-DC planta Helios.

- **Baterías:** Las baterías instaladas presentan una capacidad de 190 Ah. Con un total de 4 baterías serie, ofrecen 9120 Wh.



Figura 29: Baterías planta Helios.

- **Equipo de fuerza:** El equipo instalado es un Huawei APM30H [29] y consta de varios rectificadores PSU4850A (6kW) [30].

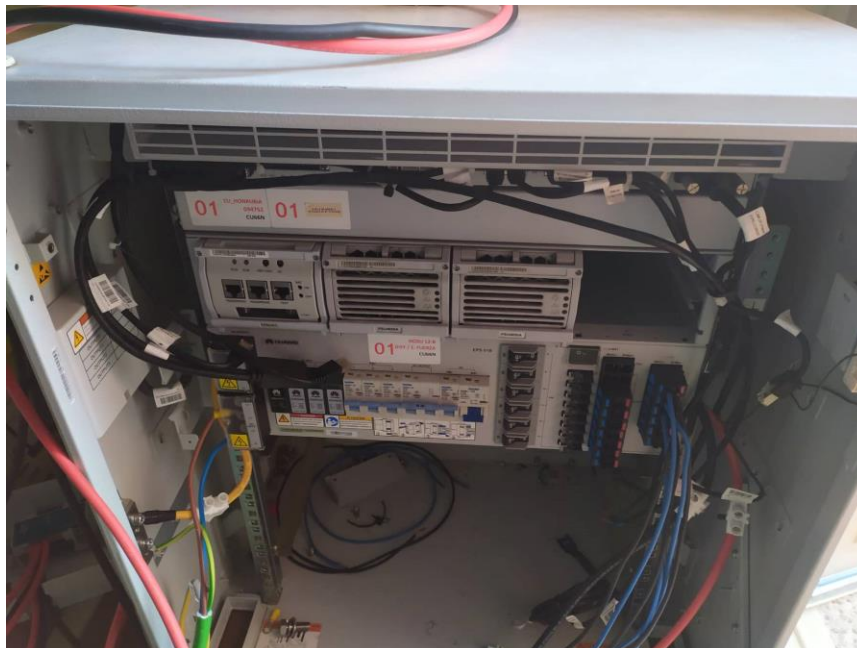


Figura 30: Equipo de fuerza planta Helios.

- **Cargas:** Con el fin de emular las BTS, se hace uso de unas resistencias eléctricas como cargas. Cada carga solicita 1000 W a 55 V con un total de 6 cargas, se solicitan 6000 W.



Figura 31: Cargas planta Helios.

CAPÍTULO 4: DESARROLLO DEL SISTEMA

Para llevar a cabo la implementación del estándar IEC 61850 en la planta DER desarrollada en el proyecto HELIOS, se debe digitalizar la planta DER a través de los requisitos presentes en el estándar IEC 61850.

Posteriormente, una vez digitalizada la planta a través del IEC 61850 se deben implementar las comunicaciones entre dispositivos mediante los requisitos presentes en dicho estándar. Para ello se valoran dos alternativas, la implementación en una placa computadora Raspberry Pi o en un microcontrolador ESP-32.

4.1.1 Valoración de la implementación del estándar en una placa computadora o en un microcontrolador

Debido a que las comunicaciones presentes en el estándar IEC 61850 se basan en protocolos Cliente-Servidor y Publicador-Suscriptor implementados atendiendo al modelo OSI, se requiere disponer de con una librería donde se implemente dichos protocolos según el estándar IEC 61850.

En la literatura se encuentran diferentes librerías entre las que se destacan la librería de código abierto libIEC61850 [31] basada en lenguaje de programación C y la librería OpenIEC61850 [32] basada en lenguaje de programación java.

En este trabajo se opta por realizar la implementación del estándar IEC 61850 a través de la librería libIEC61850, ya que permite la implementación del estándar IEC 61850 a través de código abierto basado en lenguaje C, compilable tanto para una Raspberry Pi como para el ESP-32.

Debido a que la librería libIEC61850 ha sido testeada en las siguientes plataformas:

- Linux/x86
- Linux/PPC (iBook with LinuxMintPPC)
- Linux/MIPS32 (Carambola board)
- Linux/ARM (BeagleBone – Cortex-A9, Raspberry Pi – ARM11)
- ucLinux/ARM7 (Wago Linux Fieldbus coupler)
- Windows 7/MinGW
- Windows 7/VisualStudio 2010/2012
- Mac OS X 10.9

La compilación en la Raspberry Pi es directa, ya que esta ejecuta el sistema operativo Raspberry Pi OS (anteriormente llamado Raspbian), una adaptación del sistema operativo GNU/Linux basado en Debian. Esta distribución ya ha sido previamente testeada con la librería.

Sin embargo, la compilación de esta librería para el ESP-32, requiere un arduo esfuerzo tanto en tiempo como en modificación de la librería, debido a que dicha librería ha sido desarrollada e implementada para ser ejecutada en sistemas Linux integrados, así como en computadoras de escritorio que ejecutan Linux, Windows o MacOS. Para compilarse en un microcontrolador como el ESP-32, es necesario desarrollar ciertas funciones de comunicaciones requeridas para la implementación del estándar IEC 61850. Por ello, y teniendo en cuenta que el objetivo principal de este trabajo es la implementación del estándar de comunicaciones IEC 61850 para la planta DER, en este trabajo se opta por una implementación en la Raspberry Pi.

4.1.2 Modelado de la unidad de generación distribuida bajo el estándar IEC 61850

El primer paso para el desarrollo de este trabajo es el modelado de la planta DER a través del estándar de comunicaciones IEC 61850. A este paso es al que se le denomina la digitalización de la planta, es decir la traducción de los elementos y las variables físicas presentes en la instalación en parámetros y valores digitalizados.

Para ello, en primer lugar, se definen los equipos a modelar presentes en la instalación. Observando la Figura 26 donde se representa un esquema de la planta Helios, se procede a definir los diferentes subgrupos presentes en la planta, esto servirá para definir posteriormente el modelo IEC 61850 de la misma. Estos grupos serán:

- **AC:** Esta parte engloba los parámetros en relación con la conexión con la red eléctrica en corriente alterna. El elemento principal a modelar dentro de este grupo sería el rectificador:
 - **Rectificador:** Equipo de fuerza para acoplar el bus de continua con la red eléctrica.

- **Fotovoltaica:** En este grupo se englobaría toda la información referida a los paneles fotovoltaicos y a las medidas eléctricas de dicha etapa.
- **Baterías:** En esta etapa se englobaría la información relacionada con las baterías y con las medidas eléctricas de las mismas.
- **Cargas controlables y cargas no controlables:** Se diferencian en dos grupos debido a que con unas se permitirá realizar gestión de la demanda y con las no controlables no. Por ello se deben modelar como dos elementos diferentes.
- **Estación meteorológica:** Debido a que la fuente de generación principal de la planta es la energía solar es de suma importancia parametrizar y modelar la información meteorológica.
- **Gestión de la planta:** Se define una parte general para la gestión de la planta, donde se definirá el estado de la planta, es decir el modo de funcionamiento y por ende el estado de los interruptores presentes en la misma.

Una vez realizada la primera definición de los grupos de interés para modelarlos bajo el estándar IEC 61850, se procede a realizar el modelado a través de las partes del estándar IEC 61850-4 y más concretamente el IEC 61850-4-720.

Para ello, se definen los siguientes dispositivos lógicos (LD) como grupos principales para estructurar el IED principal que se pretende modelar, la Raspberry Pi:

- **LD MNGT:** En este dispositivo lógico se engloban todas las funcionalidades necesarias para la gestión del modo de operación de la planta global. Algunas de las funcionalidades son:
 - Información y definición de los parámetros de gestión de la planta.
 - Apertura y cierre de los diferentes interruptores de la planta, para que trabaje en los diferentes modos en función de la generación solar y el precio de la electricidad.
- **LD ACGRID:** Este dispositivo lógico engloba todo lo necesario para la conexión de la planta con la red AC. En este caso:
 - Rectificador.
 - Medidas analógicas.

- **LD PV:** En este dispositivo lógico se engloban todas las características y toda la información sobre la generación mediante las placas fotovoltaicas:
 - Información sobre la matriz de paneles instalada.
 - Información sobre el regulador MPPT.
- **LD BESS:** Sistema de almacenamiento de energía en baterías:
 - Información sobre los parámetros más característicos del sistema de baterías usado.
- **LD PLOAD:** Dispositivo lógico donde se recoge información de las cargas prioritarias conectadas en el sistema. Principalmente información sobre variables eléctricas.
- **LD NPLOAD:** Este dispositivo lógico al igual que el anterior, recoge la información de las cargas conectadas en el sistema, pero en este caso se unifican únicamente las cargas no prioritarias.
- **LD WEATHERSTATION:** Dispositivo lógico donde se recoge la información relevante sobre las variables del tiempo:
 - Temperatura.
 - Humedad.
 - Viento.
 - Irradiancia.

Una vez definidos los dispositivos lógicos, se definen los Nodos Lógicos necesarias para cada LD para modelar la Planta:

- **LD MNGT:**
 - **LLNO:** Contiene información de los dispositivos lógicos.
 - **LPHD:** Contiene información de los dispositivos físicos.
 - **DCRP:** Características corporativas de la planta DER en el ECP. Incluye: Propiedad, operación, autoría, obligaciones contractuales y permisos, ubicación e identificadores de todos los dispositivos DER conectados directa o indirectamente en el ECP.
 - **DOPR:** Características operacionales en el ECP, incluye tipo de dispositivos DER, tipo de conexión, modos de operación, clasificación

combinada de todas las DER en el ECP, límites de operación del sistema de potencia en el ECP.

- **DOPM:** Modo de operación del DER en el punto de conexión. Este LN puede ser usado para fijar los modos de operación disponibles junto con los modos de operación actuales.
 - **DPST:** Estado actual en cada ECP, incluye estado de conexión de la planta DER, alarmas.
 - **DCCT:** Parámetros de despacho económico para las operaciones DER.
 - **DSCC:** Control de los programas de energía y servicios auxiliares.
 - **DSCH:** Programa de la planta DER para proporcionar energía o los servicios auxiliares.
 - **CSWI1:** Control del interruptor → Relé 1.
 - **XCBR1:** Interruptor de circuito → Relé 1.
 - **CSWI2:** Control del interruptor → Relé 2.
 - **XCBR2:** Interruptor de circuito → Relé 2.
 - **CSWI3:** Control del interruptor → Relé 3.
 - **XCBR3:** Interruptor de circuito → Relé 3.
 - **CSWI4:** Control del interruptor → Relé 4.
 - **XCBR4:** Interruptor de circuito → Relé 4.
 - **CSWI5:** Control del interruptor → Relé 5.
 - **XCBR5:** Interruptor de circuito → Relé 5.
 - **CSWI6:** Control del interruptor → Relé 6.
 - **XCBR6:** Interruptor de circuito → Relé 6.
- **LD PV**
 - **LLN0:** Contiene información de los dispositivos lógicos.
 - **LPHD:** Contiene información de los dispositivos físicos.
 - **DPVM:** Información sobre el módulo PV.
 - **DPVA:** Características del array PV. Ofrece información general del array PV.
 - **DPVC:** Control de la matriz PV. Usado para maximizar la potencia de salida de la matriz.
 - **MMDC:** Medidas de las variables eléctricas DC.

- **LD BESS**
 - **LLNO**: Contiene información de los dispositivos lógicos.
 - **LPHD**: Contiene información de los dispositivos físicos.
 - **ZBAT**: Características del sistema de baterías.
 - **ZBTC**: Carga del sistema de baterías.
 - **MMDC**: Medidas de las variables eléctricas DC.

- **LD PLOAD**
 - **LLNO**: Contiene información de los dispositivos lógicos.
 - **LPHD**: Contiene información de los dispositivos físicos.
 - **MMDC**: Medidas de las variables eléctricas DC.

- **LD NPLOAD**
 - **LLNO**: Contiene información de los dispositivos lógicos.
 - **LPHD**: Contiene información de los dispositivos físicos.
 - **MMDC**: Medidas de las variables eléctricas DC.

- **LD ACGRID**
 - **LLNO**: Contiene información de los dispositivos lógicos.
 - **LPHD**: Contiene información de los dispositivos físicos.
 - **ZRCT**: Características del rectificador.
 - **MMXU**: Medidas de las variables eléctricas AC.

- **LD WEATHERSTATION**
 - **LLNO**: Contiene información de los dispositivos lógicos.
 - **LPHD**: Contiene información de los dispositivos físicos.
 - **MMET**: Información meteorológica.

Una vez definidos los nodos lógicos que se desean incluir en el modelado, se deben analizar los objetos de datos (DO) que estos incluirán. Se debe tener en cuenta que los DO presentes en un LN pueden ser:

- **M (Mandatory)**: La presencia de este tipo de objetos de datos dentro del nodo lógico es obligatoria.

- **O (Optional):** La presencia de este tipo de objetos de datos dentro del nodo lógico es opcional.
- **C (Conditional):** La presencia de este tipo de objetos de datos dentro del nodo lógico es condicional, es decir depende de algún requisito su presencia.

Según esta información y los requisitos presentes para el modelado de la planta, los objetos de datos incluidos han sido:

- **LN DCRP**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **PlntObSelf:** Propósito/obligaciones de la planta en el ECP – True = funciona de forma pasiva o siempre que sea posible (fotovoltaica).
 - **PlntObBck:** Propósito/obligaciones en el ECP – True = Para respaldo (backuo).

- **LN DOPR**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **ECPTType:** Tipo de ECP:
 - 0: No aplicable / Desconocido.
 - 1: Conexión de una única DER a una carga local
 - 2: Conexión del grupo de DERs al EPS local, sirviendo una carga local.
 - 3: Conexión del EPS local con la carga local al EPS de la zona (PCC).
 - 4: Conexión del EPS local sin carga local al EPS de la zona (PCC).
 - 99: Otro.
 - **ECPID:** Identificación del ECP.

- **ECPNomWRtg:** Nominal, mínimo, y máximo agregado de watos DER en el ECP.
- **ECPNomVLev:** Nominal, mínimo, y máximo nivel de voltaje en el ECP.
- **LN DOPM**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **OpModPM:** Modos de operación – Impulsado por la fuente de energía solar, por lo que el nivel de generación depende de la disponibilidad de esa fuente de energía.
 - **OpModConW:** Modo de operación – watos constantes.
 - **OpModConV:** Modo de operación – Voltaje constante.
 - **OpModExIm:** Modo de operación – Exportación/ importación constante.
 - **OpModIsl:** Modo de operación – Isla en el ECP.
- **LN DPST**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **OpTms:** Tiempo de operación desde la puesta en marcha.
 - **ECPConn:** Conexión de la planta DER en el ECP:
 - **True:** Conectado eléctricamente al ECP.
 - **False:** No conectado eléctricamente al ECP.
 - **TotWh:** Total de watos-hora al ECP desde el último reset.
- **LN DCCT**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).

- **Mod:** Modo.
- **Currency:** Código de 3 caracteres de la moneda ISO 4217 (978).
- **OpCost:** Coste operacional marginal por hora.
- **OpWCost:** Coste operacional marginal por kWh.
- **StrCost:** Coste de la puesta en marcha de la DER.
- **StopCost:** Coste de parada de la DER.

- **LN DSCC**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **ActWSchSt:** Indicación de qué programa de energía está activo – Schedule 0 indica no programa.
 - **ActAncSchdSt:** Indicación de que programa de servicios auxiliares está activo – Schedule 0 indica no programa.
 - **ActWSchd:** Activar un programa de energía específico, usando TimeActivatedOperate para establecer la hora de inicio de los programas usando el tiempo relativo y si la hora de inicio está en el futuro. ctrVal: 0 = desactivar, 1 = activar.
 - **ActAncSchd:** Activar un programa de equipos auxiliares específico, usando TimeActivatedOperate para establecer la hora de inicio de los programas usando el tiempo relativo y si la hora de inicio está en el futuro. ctrVal: 0 = desactivar, 1 = activar.

- **LN DSCH**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **SchdSt:** Indica que el programa ha sido activado.
 - **SchdId:** Identidad no nula del programa.

- **SchdTyp:** Tipo de programa, identificador del modo de funcionamiento en el que se utiliza el programa:
 - **0:** No aplicable / Desconocido.
 - **1:** Energía.
 - **2:** Reserva de contingencia “spinning”.
 - **3:** Reserva suplementaria de contingencia.
 - **4:** Reserva de emergencia.
 - **5:** Reserva de emisiones.
 - **6:** Balance energético.
 - **7:** Potencia reactiva.
 - **8:** Black start.
 - **9:** Isla emergencia.
 - **99:** Otro.
- **SchdCat:** Categoría del programa:
 - **0:** No aplicable / Desconocido.
 - **1:** Regular.
 - **2:** BackUp.
 - **3:** Emergencia.
 - **4:** Mantenimiento.
 - **99:** Otro.
- **SchdVal:** Significado del parámetro val en el SCA o SCR:
 - **0:** No aplicable / Desconocido.
 - **1:** Potencia activa.
 - **2:** Potencia Reactiva.
 - **3:** Factor de potencia.
 - **4:** Voltaje.
 - **5:** Precio por la potencia activa.
 - **6:** Precio por la potencia reactiva.
 - **7:** Calor.
 - **99:** Otro.
- **LN CSWI**
 - **NamPlt:** Placa de identificación.

- **Beh:** Comportamiento.
- **Health:** Salud (estado).
- **Mod:** Modo.
- **LocKey:** Local o remoto.
- **Loc:** Control local del comportamiento.
- **OpOpn:** Operación “Apertura del interruptor”.
- **SelOpn:** Selección “Apertura del interruptor”.
- **OpCls:** Operación “Cierre del interruptor”.
- **SelCls:** Selección “Cierre del interruptor”.
- **OpCntRs:** Contador de operaciones reseteable.
- **Pos:** Posición.

- **LN CSWI**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **LocKey:** Local o remoto.
 - **Loc:** Comportamiento de control local.
 - **OpCnt:** Contador de maniobras.
 - **Pos:** Posición del interruptor.
 - **BlkOpn:** Apertura del bloque.
 - **BlkCls:** Cierre del bloque.

- **LN DPVM**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **AVCrv:** Índice en el punto activo de la curva AV.
 - **MdulCfgTyp:** Tipo de configuración del módulo PV:
 - **0:** Desconocido/ no aplicable.
 - **1:** Flate plate.

- **2:** Concentrado.
- **99:** Otro.
- **MdulAVCrv:** Curva Amp-Volt del módulo en STC.
- **MdulWRtg:** Módulo de potencia nominal en vatios pico STC.
- **MdulW200Rtg:** Módulo de potencia nominal en vatios pico a 200 W/m².
- **MaxMdulV:** Voltaje del módulo a máxima potencia en STC.
- **MaxMdulA:** Corriente del módulo a máxima potencia en STC.
- **MdulOpnCctV:** Voltaje del módulo en circuito abierto.
- **MdulSrtCctA:** Corriente del módulo en cortocircuito.
- **MdulWTmpDrt:** Porcentaje de reducción de la potencia en función de la temperatura (potencia/temperatura) por encima de 25°C.
- **MdulATmpDrt:** Porcentaje de reducción de la corriente en función de la temperatura (corriente/temperatura) por encima de 25°C.
- **MdulVTmpDrt:** Porcentaje de reducción de la tensión en función de la temperatura (tensión/temperatura) por encima de 25°C.
- **MdulAgeDrt:** Porcentaje de envejecimiento del módulo con el tiempo.
- **LN DPVA**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **Typ:** Tipo de montaje:
 - **0:** Desconocido/ no aplicable.
 - **1:** Matriz.
 - **2:** Sub-matriz.
 - **3:** Serie.
 - **4:** Modulo.
 - **5:** Planta.
 - **99:** Otro.
 - **ModulCont:** Número de módulos por serie
 - **StrgCnt:** Número de series en paralelo por sub-matriz.

- **SubArrCnt:** Número de sub-matrices en paralelo por matriz.
- **ArrArea:** Área de la matriz.
- **GrnConn:** Tipo de conexión a tierra:
 - **0:** Desconocido/ no aplicable.
 - **1:** Puesta a tierra positiva.
 - **2:** Puesta tierra negativa.
 - **3:** Sin puesta a tierra.
 - **99:** Otro.
- **ArrWRtg:** Potencia nominal del array (wattios pico – wattios p).
- **Tilt:** Montaje de inclinación fija - grados desde la horizontal (puede ser ajustado por estación).
- **Azi:** Azimut de montaje – grados desde el norte verdadero.

- **LN DPVC**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **CtrModSt:** Estado del modo de control de la matriz.
 - **ArrModCtr:** Modo seleccionado para el control de la potencia de salida de la matriz:
 - **0:** Desconocido/ no aplicable.
 - **1:** Seguimiento del punto de máxima potencia (MPPT).
 - **2:** Control del límite de potencia.
 - **3:** Limite de la corriente DC.
 - **4:** Control del voltaje de la matriz.
 - **99:** Otro.

- **LN MMDC**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.

- **Watt:** Potencia.
- **Amp:** Corriente.
- **Vol:** Voltaje.

- **LN ZBAT**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **OpTmh:** Tiempo de operación.
 - **TestRsl:** Resultado de las pruebas de batería.
 - **BatHi:** Batería alta (voltaje o carga - sobrecarga).
 - **BatLo:** Batería baja (voltaje o carga).
 - **Vol:** Voltaje de baterías.
 - **VolChagRte:** Tasa de cambio del voltaje de baterías.
 - **Amp:** Corriente de drenaje de la batería.
 - **BatTest:** Start del test de batería.
 - **LoBatVal:** Valor de alarma de batería baja.
 - **HiBatVal:** Valor de alarma de batería alta.

- **LN ZBTC**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **ChaV:** Voltaje de carga.
 - **ChaA:** Corriente de carga.
 - **BatChaTyp:** Tipo de carga de batería:
 - **0:** Desconocido/ no aplicable.
 - **1:** Voltaje constante.
 - **2:** Corriente constante.
 - **99:** Otro.
 - **ChaCrv:** Curva de carga.

- **ReChArTe:** Tasa de recarga.
- **BatChAPwr:** Potencia requerida para la carga de la batería.

- **LN ZRCT**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **IsoTyp:** Tipo de aislamiento:
 - **0:** Desconocido/ no aplicable.
 - **1:** Transformador de frecuencia de potencia aislado.
 - **2:** Hi transformador de frecuencia aislado.
 - **3:** Sin aislamiento, puesta a tierra.
 - **4:** Sin aislamiento, aislamiento de la fuente DC.
 - **99:** Otro.
 - **VRegTyp:** Tipo de regulación de voltaje:
 - **0:** Desconocido/ no aplicable.
 - **1:** Regulación de salida: voltaje fijo.
 - **2:** Regulación de salida: Voltaje variable.
 - **3:** Filtrado de salida: Dependiente de la carga.
 - **4:** Sin regulación y sin filtrado.
 - **99:** Otro.
 - **VRegTyp:** Tipo de regulación de voltaje:
 - **1:** Fase única.
 - **2:** Dos fases.
 - **3:** Tres fases.

- **LN ZRCT**
 - **NamPlt:** Placa de identificación.
 - **Beh:** Comportamiento.
 - **Health:** Salud (estado).
 - **Mod:** Modo.
 - **EnvTmp:** Temperatura ambiente.

- **EnvHum:** Humedad.
- **DffInsol:** Irradiancia difusa.
- **DctInSol:** Irradiancia normal directa.
- **HorInsol:** Irradiancia total horizontal.
- **HorWdDir:** Dirección del viento horizontal.
- **HorWdSpd:** Velocidad del viento horizontal.
- **EnvPres:** Presión barométrica.

De esta forma, la digitalización de la planta resumida, sin introducir los LN obligatorios quedaría según la Figura 32.

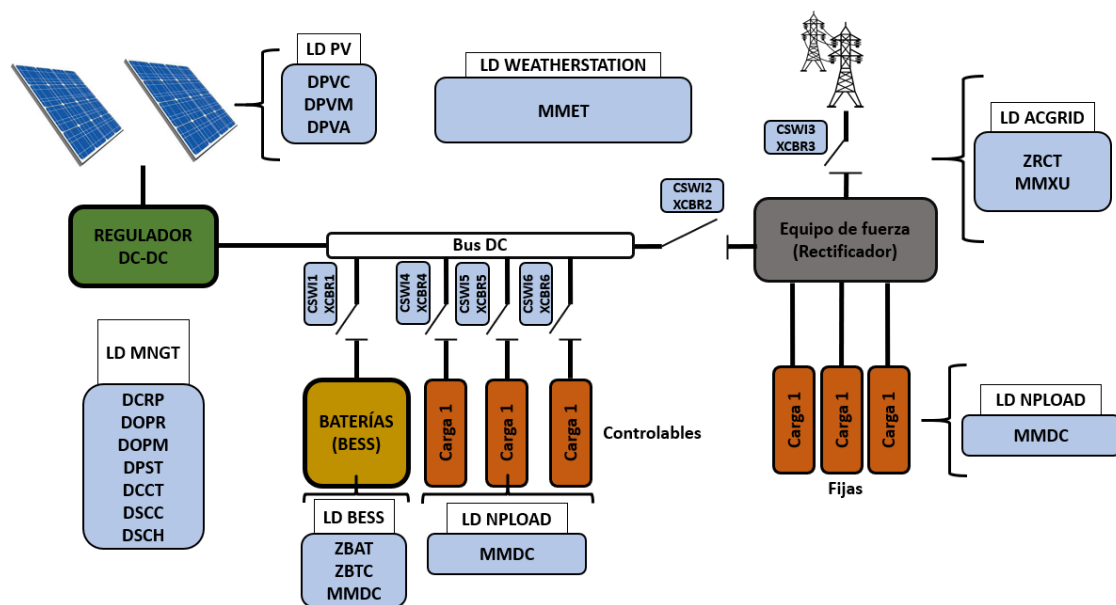


Figura 32: Digitalización planta Helios según IEC 61850.

4.1.3 Requerimientos de comunicaciones IEC 61850 en la unidad de generación distribuida

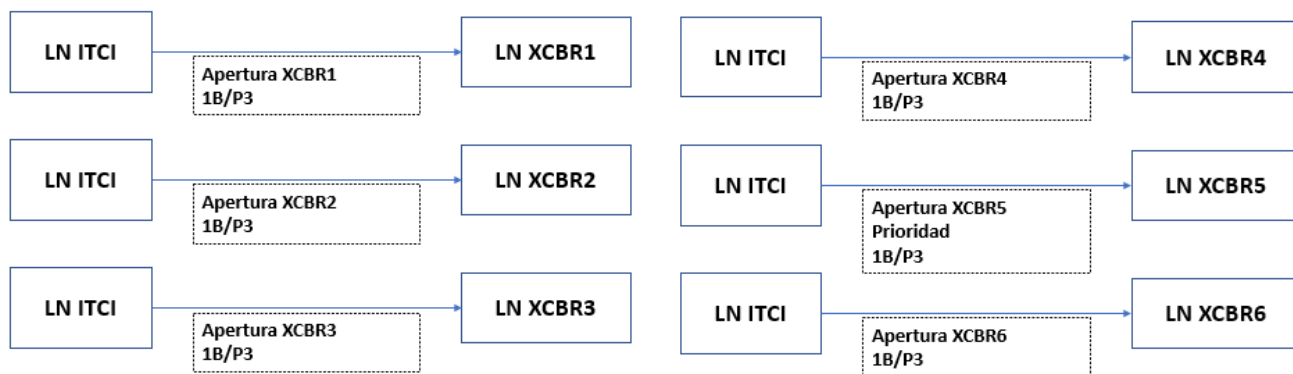
Una vez digitalizada la planta, se procede al desarrollo de un programa basado en comunicaciones cliente-servidor y publicador-suscriptor con el fin de actualizar y realizar las comunicaciones en base al estándar de comunicaciones IEC 61850.

Para ello, se hace uso de un PC que actúa como cliente/publicador, conectado a una Raspberry Pi que trabaja como servidor/suscriptor.

Antes de implementar el sistema se deben definir los requerimientos de comunicaciones presentes en la planta. Para ello, tal y como se describe en el apartado 2.1.9, se deben analizar las interfaces presentes en dicho sistema, junto con las necesidades y los requerimientos de tiempos necesarios para poder seleccionar los tipos de tiempos y definir los PICOM.

Observando las diferentes interfaces de comunicación presentes en la norma, la interfaz que más se ajusta para los intercambios de información presentes en este sistema es la interfaz 10. Además, la norma expone como recomendación que para sistemas DER, como el presente trabajo, se haga uso de interfaz 10 para comunicar con un nivel superior para el control. El PC (Cliente/Publicador) se considera un sistema para el control remoto, mientras que la Raspberry Pi, se considera a nivel de subestación.

Respecto a los tipos de mensaje, se selecciona para el intercambio de las órdenes de apertura y cierre de los interruptores del sistema el tipo 1B con un rendimiento P3. Se seleccionan los mensajes tipo 1B para la apertura y cierre de interruptores, debido a que esta maniobra se considera crítica en el sistema por lo que se le exigen requerimientos de tiempos más exigentes. No se seleccionan los mensajes tipo 1^a debido a que estos mensajes están orientados para sistemas donde se realizan funciones de protección. En este sistema la apertura y cierre de los interruptores solo implican maniobras de control de la planta. Para los intercambios de información para la adquisición de las variables del sistema se selecciona el tipo 3 con un rendimiento P5, ya que estos mensajes presentan requerimientos de tiempos menos exigentes y según la norma, este tipo de mensajes son los empleados para la transmisión de mensajes donde se informa a operadores del estado del sistema. De esta forma los PICOM quedan:



- PICOM para la transmisión órdenes de apertura y cierre de los 6 interruptores del sistema, Figura 33.

Figura 33: Definición de PICOMs para la actuación de interruptores.

- PICOM para la transmisión del valor de las variables del sistema, Figura 34.

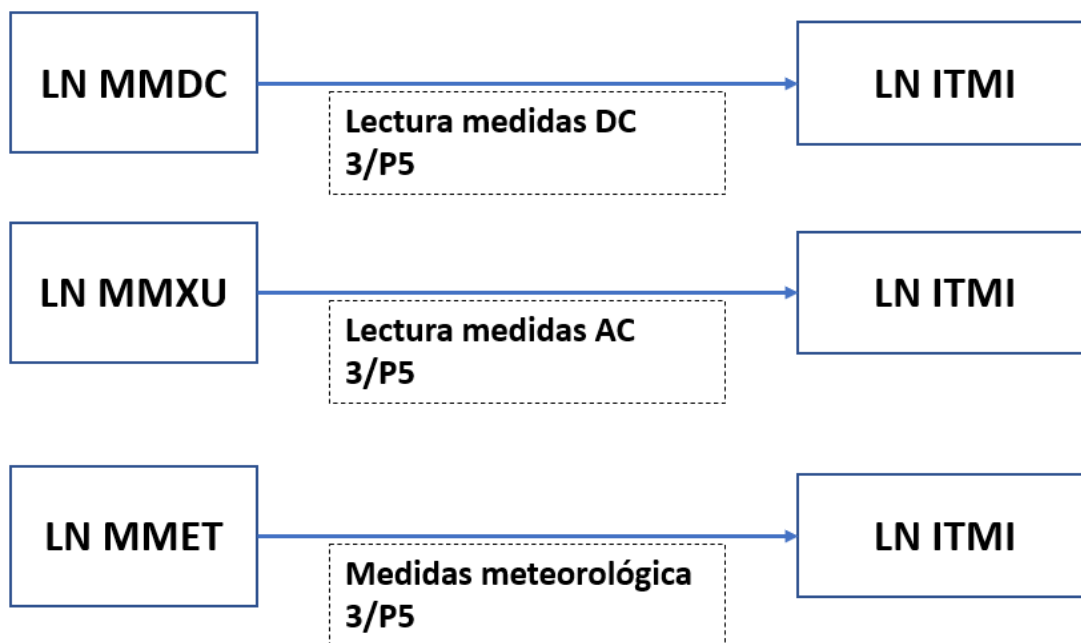


Figura 34: Definición de PICOM para la lectura de variables.

Observando los requerimientos de comunicaciones, y más en concreto los tiempos máximos definidos para los intercambios de información, se considera necesario hacer uso del servicio de comunicaciones GOOSE para la transmisión de los PICOM para la apertura y cierre de los interruptores. Esta decisión se toma debido a que

para estas transmisiones se fijan tiempos de respuesta por debajo de los 20ms. Por otro lado, se decide implementar el servicio de comunicaciones MMS para la transmisión de las variables físicas del sistema, debido a que los requerimientos de esta transmisión de información son menos exigentes encontrándose en rangos por debajo de los 500ms.

Una vez definidos los requerimientos de comunicaciones, se procede a definir el medio físico. En este tipo de aplicaciones, donde las comunicaciones se basan en el estándar de comunicaciones IEC 61850, normalmente se hace uso de una de las siguientes dos tecnologías de canales físicos de comunicaciones: cable Ethernet o fibra óptica. Debido a que la distancia de cable es pequeña, a la baja carga de flujo de comunicaciones y a la facilidad y el coste de la implementación, en este trabajo se decide implementar las comunicaciones entre el PC y la Raspberry mediante una conexión vía cable Ethernet. Para permitir la comunicación entre ambos equipos, se configuran las IPs y las máscaras de ambos equipos de forma que se cree una red de área local (LAN) donde se puedan comunicar ambos (Figura 35).

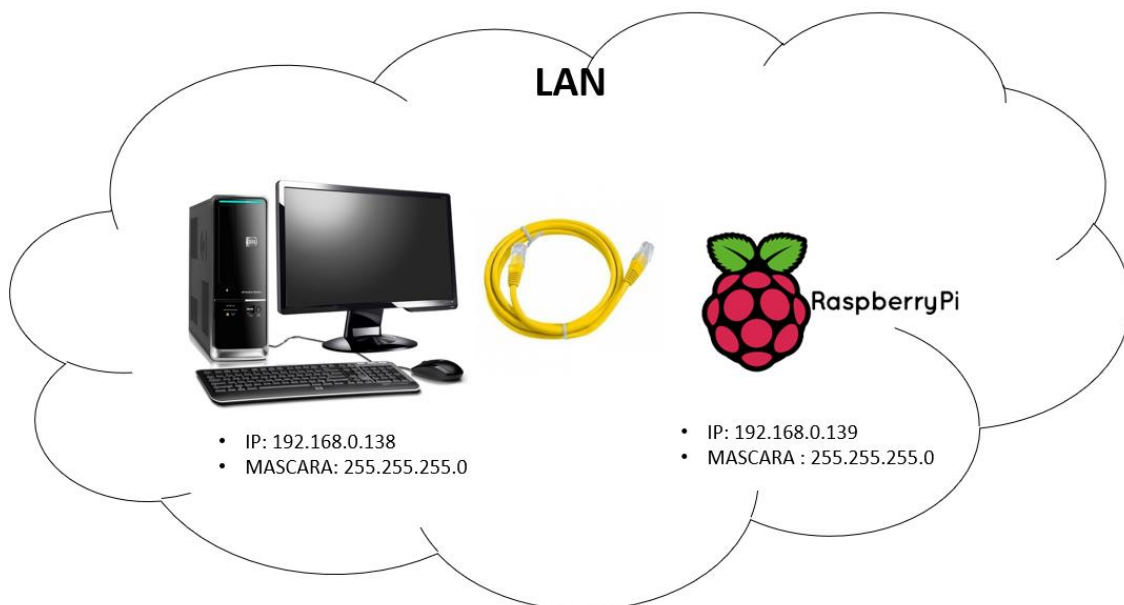


Figura 35: Conexión PC-Raspberry Pi.

En el caso que se deseara introducir un nuevo equipo en el sistema, por ejemplo, un nuevo servidor PC2, se debería introducir un switch y conectar todos los equipos a través de dicho switch además de configurar el mismo con una dirección IP y la máscara adecuada.

4.1.4 Sistema de sincronización para medida de tiempos

Debido a los requisitos de comunicaciones expuestos en los apartados anteriores, y con el objetivo de verificar si el sistema desarrollado en este trabajo cumple con las especificaciones definidas, se decide implementar un protocolo de sincronización entre el cliente y el servidor para, posteriormente, verificar los tiempos de transmisión de la información.

Aprovechando la implementación de red en la que se basa el estándar IEC 61850 trabajando con las capas OSI y aprovechando la misma red Ethernet implementada para la mensajería, se decide implementar el protocolo de sincronización NTP (*Network Time Protocol*). NTP es un protocolo de internet común para la sincronización de los relojes de sistemas informáticos en redes locales y globales a través del enrutamiento de paquetes en redes con latencia variable. NTP utiliza UDP (Figura 36) como su capa de transporte, usando el puerto 123. Además, está diseñado para resistir los efectos de la latencia variable.

Ubicación en la pila de protocolos	
Aplicación	NTP
Transporte	UDP
Red	IP

Figura 36: Ubicación NTP en la pila de protocolos.

El funcionamiento de NTP difiere de la mayoría de los otros protocolos. NTP no sincroniza todos los relojes conectados, establece una jerarquía de servidores de tiempo y clientes. Cada nivel de esta jerarquía (Figura 37) se denomina estrato, donde el estrato 0 constituye el nivel más alto [33].

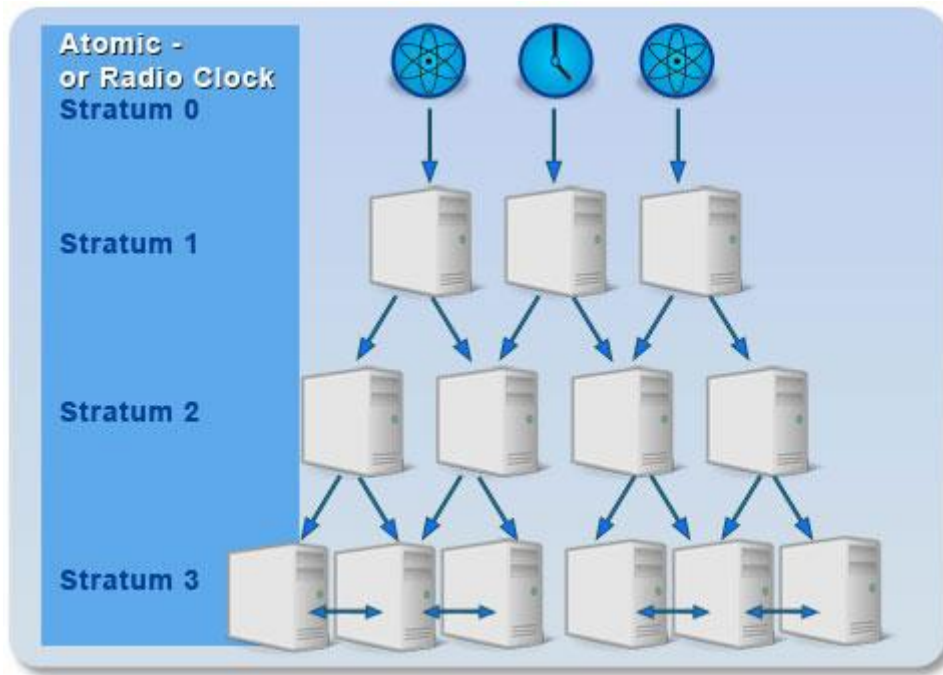


Figura 37: Jerarquía NTP.

De esta forma este trabajo implementa una jerarquía de sincronización donde la Raspberry se sincroniza como cliente y el PC Linux como servidor NTP. Para ello, se instala el paquete con las librerías NTP en ambos sistemas, y posteriormente se configura la Raspberry Pi para que tome como fuente de sincronización el PC Linux mediante NTP. Para ello se debe modificar el archivo *ntp.conf* seleccionando a través del direccionamiento IP del ordenador PC Linux a este como fuente maestra de sincronización.

4.1.5 Implementación del modelo basado en el IEC 61850.

Como se ha comentado en el apartado 4.1.3 el sistema presente en este trabajo se basa en un cliente/publicador y un servidor/suscriptor.

De esta forma, una vez definidos los requerimientos del sistema, se decide implementar en el PC un cliente/publicador que se encarga de realizar las siguientes tareas:

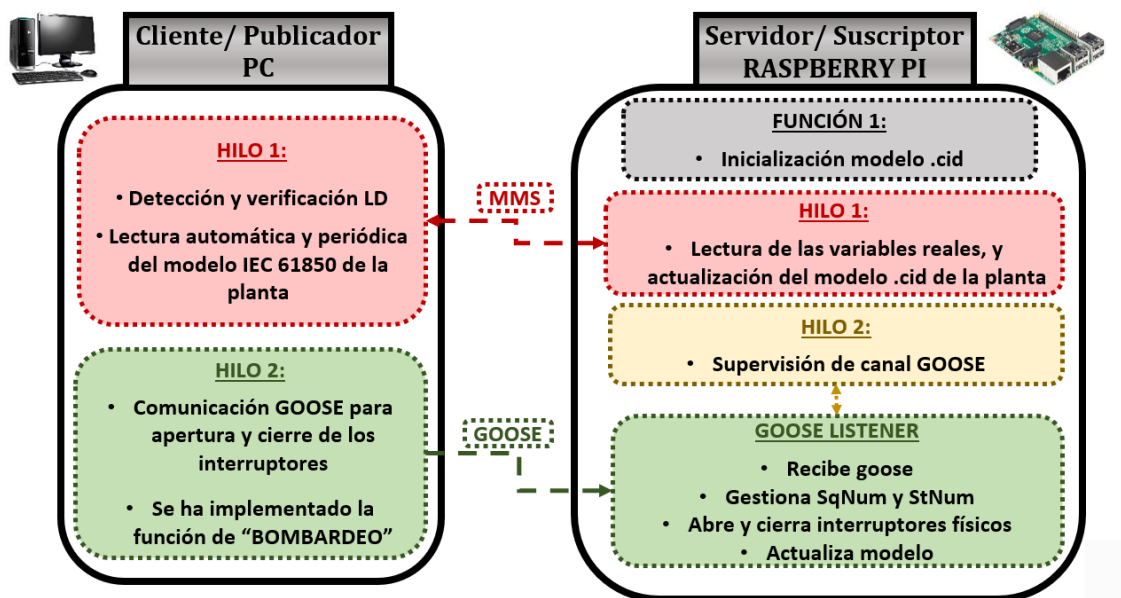
- Leer de forma automática y periódica el modelo de la planta del servidor/suscriptor (Raspberry Pi), de esta forma, se dispone en el PC de forma periódica de los valores de las variables actualizadas de la planta presentes en el modelo IEC 61850 de la planta a través de mensajería MMS.

- Realizar la comunicación GOOSE. Se define un mensaje GOOSE para la apertura y el cierre de los interruptores. De esta forma se garantiza que dicha acción se realiza en el menor tiempo posible.
- Por último, se implementa una función adicional que permite de forma automática detectar los LD presentes en el modelo de la planta. Esto permite incorporar dos funciones adicionales, el escaneo de equipos no reconocidos en la red, junto con la verificación del modelo de la planta esperado para el trabajo dentro de la planta o futuro sistema DER presente en el proyecto. De esta forma, si se conectase una nueva planta en el sistema y dicha planta no se encontrase modelada con los parámetros necesarios para su operabilidad, sería detectada automáticamente por el programa y permitiría realizar funciones como la desconexión automática de la misma.

Por otro lado, se tiene el servidor/suscriptor (Raspberry Pi), el cual se encarga de las siguientes funciones:

- Inicialización del modelo IEC 61850. Se define una función específica que permitirá inicializar el modelo IEC 61850 de forma que facilite la modificación de parámetros de la planta o de componentes que se verá representado en la modificación de DO en el modelo IEC 61850.
- Lectura de las variables reales de la planta. Por otro lado, se define una función que lea de forma periódica las variables eléctricas de la planta y a su vez actualice el modelo *.cid* IEC 61850 de la planta con el fin de que cualquier cliente pueda acceder y obtener sus valores.
- Supervisión de canal GOOSE. Como se ha comentado en el capítulo 2 (Figura 12) el servicio de mensajería GOOSE definido por la norma IEC 61850 presenta un bombardeo constante que tiene como objetivo principal el muestreo del canal para verificar la correcta transmisión. Para implementar esta funcionalidad, se desarrolla una función para la detección del fallo del canal de comunicación GOOSE. De esta forma en el momento que cese el “bombardeo” de mensajes GOOSE, salta una alarma avisando de que el canal de comunicaciones GOOSE está fallando.

- GOOSE Listener. Por otro lado, se implementa una función que se encarga de recibir el mensaje GOOSE enviado por el PC, gestionar los parámetros SqNum y StNum, abrir y cerrar los interruptores pertinentes, y, por último, una vez realizada la maniobra de apertura o cierre, se encarga de actualizar el estado del modelo IEC 61850.



De esta forma el esquema del sistema desarrollado en este trabajo queda según la Figura 38.

Figura 38: Flujo de comunicación de sistema de comunicaciones basadas en el estándar IEC 61850.

4.6.1 Programación Cliente/Publicador (PC)

Como se ha comentado al principio del trabajo, la implementación del estándar de comunicaciones IEC 61850 se ha realizado a través de la librería de código abierto libiec61850 [25].

Para ello, en la parte del cliente/publicador las librerías empleadas han sido:

- **Librería Hilos:** Con la finalidad de poder trabajar en paralelo y ofreciendo diferente prioridad, cada tarea se ha implementado en un hilo diferente como se expone en la Figura 38.
 - #include <pthread.h>

- **Librería Cliente:** Aquí se engloban las librerías necesarias para implementar el cliente IEC 61850.
 - #include "iec61850_client.h"
- **Librerías GOOSE:** Para poder hacer uso del servicio de mensajería GOOSE ha sido necesario implementar la librería:
 - #include "goose_publisher.h"
- **Librerías MMS:** Al igual que para el servicio GOOSE, para el servicio MMS ha sido necesario hacer uso de la librería:
 - #include "mms_value.h"
- **Librería sincronización:** Para implementar un sistema de medida de tiempo una vez que se tenía los dispositivos sincronizados por NTP la librería a usar ha sido:
 - #include <sys/time.h>
- **Otras librerías:** Otras librerías necesarias para el desarrollo del sistema han sido:
 - #include <stdint.h>
 - #include <string.h>
 - #include <stdbool.h>
 - #include <stdlib.h>
 - #include <stdio.h>
 - #include "hal_thread.h"
 - #include <math.h>

Una vez mencionadas las librerías necesarias para la implementación del sistema, se procede a detallar cómo se ha implementado el programa. En la Figura 39 se observa un flujograma del funcionamiento de la parte software implementada en el Cliente/Publicador, donde, de una forma esquemática se observa como existe una función principal (Main()), la cual inicializa los hilos que se ejecutan de forma paralela posteriormente. Los hilos inicializados por la función Main(), son el hilo 1 y el hilo 2, el hilo 1 se encarga de la gestión de la comunicación vía MMS, y el hilo 2 se encarga de la gestión de la comunicación vía GOOSE. Debido a que la comunicación vía MMS se utiliza para la lectura periódica del modelo IEC 61850 implementado en el Servidor/Suscriptor, el hilo 1 hace uso de diferentes funciones para realizar dicha

acción. En la función Leer_estado_planta() se implementan las sentencias necesarias para realizar los accesos vía MMS a la información ubicada en el modelo IEC 61850 en el Publicador/Suscriptor. Una vez que se recibe esa información, se hace uso de dos funciones Estado_mod() y Estado_q(), las cuales analizan los objetos y los atributos recibidos para discernir si la información leída es real o no.

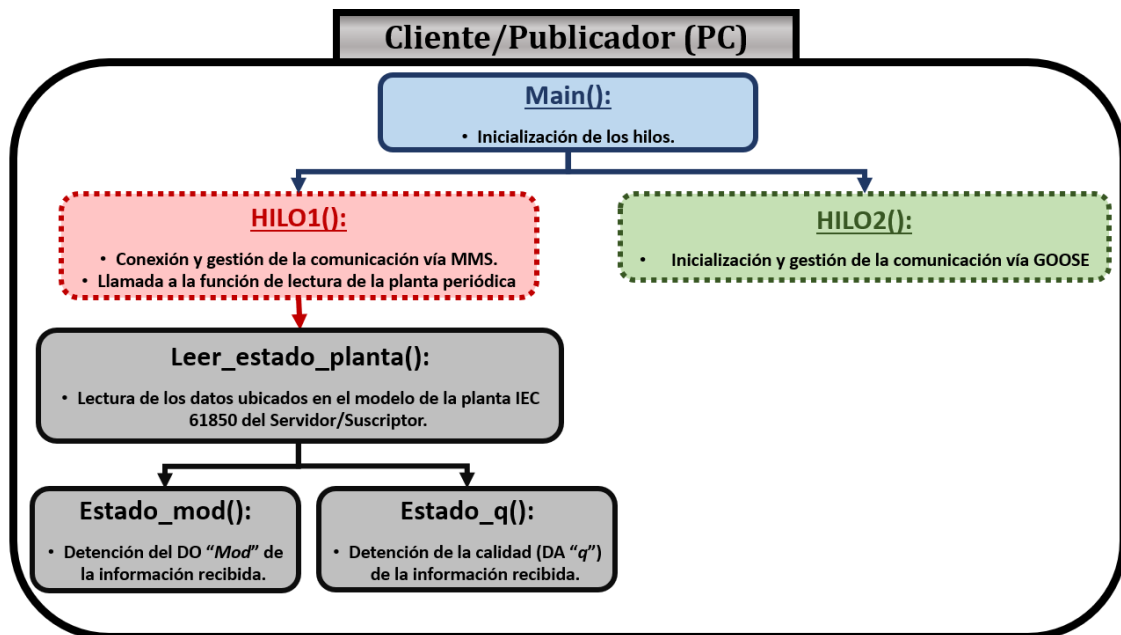


Figura 39: Flujograma programa del Cliente/ Publicador.

A continuación, se procede a realizar una explicación más detallada de cada función ubicada en el software implementado en el Cliente/Publicador:

- **Main:** Esta es la parte principal del programa y en ella únicamente se:
 - Imprime por pantalla un mensaje para dar por inicializada la comunicación de la Planta Helios bajo el estándar IEC 61850:

```
printf("\nCOMUNICACIÓN DE LA PLANTA HELIOS BAJO EL ESTANDAR IEC 61850\n\n");
```

- Se crean los hilos que se van a ejecutar, y se declara la sentencia para esperar hasta que terminen de ejecutarse ambos:

```
//Creamos los hilos
pthread_create (&h1, NULL, thread1, NULL);
pthread_create (&h2, NULL, thread2, NULL);

//Esperamos a terminar la función main a que terminen de ejecutarse los hilos
pthread_join(h1, NULL);
pthread_join(h2, NULL);
```

Como se ha comentado en apartados anteriores, el modo pruebas para poder probar funcionalidades o realizar mantenimientos en un sistema basado en IEC 61850 es

crucial. Para ello en este trabajo se ha implementado dos funciones, “*void estado_mod*” y “*void estado_q*”.

- **void estado_mod (int mod_est):** Se ha implementado una función para detectar el estado del DO “*Mod*” para poder determinar si la información que se recibe del nodo lógico es real o está en modo test y se están realizando pruebas. Para ello, la norma IEC 61850 indica que los valores para el DO “*Mod*” pueden ser:
 - **1:** ON
 - **2:** ON-BLOCKED
 - **3:** TEST
 - **4:** TEST/BLOCKED
 - **5:** OFF
 - **0:** SIN ESPECIFICAR.
- **void estado_q (Quality q_est):** Por otro lado, para detectar si el DO recibido es real o no, se ha implementado esta función. En este caso el DA “*q*” puede tomar más valores, no solo únicamente indica si el DO está en test o es real, sino que también ofrece información sobre la validez de la información que ofrece dicho DO:
 - **0:** “QUALITY_VALIDITY_GOOD”.
 - **2:** “QUALITY_VALIDITY_INVALID”.
 - **1:** “QUALITY_VALIDITY_RESERVED”.
 - **3:** “QUALITY_VALIDITY_QUESTIONABLE”.
 - **4:** “QUALITY_DETAIL_OVERFLOW”.
 - **8:** “QUALITY_DETAIL_OUT_OF_RANGE”.
 - **16:** “QUALITY_DETAIL_BAD_REFERENCE”.
 - **32:** “QUALITY_DETAIL_OSCILLATORY”.
 - **64:** “QUALITY_DETAIL_FAILURE”.
 - **128:** “QUALITY_DETAIL_OLD_DATA”.
 - **256:** “QUALITY_DETAIL_INCONSISTENT”.
 - **512:** “QUALITY_DETAIL_INACCURATE”.
 - **1024:** “QUALITY_SOURCE_SUBSTITUTED”.
 - **2048:** “QUALITY_TEST”.
 - **4096:** “QUALITY_OPERATOR_BLOCKED”.

- **Void *thread1():** El hilo 1 se encarga de realizar la conexión TCP/IP para gestionar los mensajes MMS y posteriormente realizar una llamada periódica a la función “*leer_estado_planta*”, la cual accede periódicamente a los atributos del modelo IEC 61850 para obtener información sobre el valor de las variables reales de la planta. El flujo del hilo 1 es:
 1. Crea un objeto *IedConnection* a través del cual realiza la conexión: *con = IedConnection_create();*
 2. Fija un tiempo de fin para realizar el intento de conexión con el servidor: *IedConnection_setConnectTimeout(con, 10000);*
 3. Ejecuta el intento de conexión con el servidor: *IedConnection_connect(con, &error, hostname, tcpPort);* Donde:
 - **con:** Es el objeto *IedConnection* creado anteriormente.
 - **&error:** Manejador de errores.
 - **hostname:** Dirección IP del servidor.
 - **tcpPort:** Puerto TCP a través del cual se realiza la conexión. Como se ha comentado en apartados anteriores, cuando se trabaja con la mensajería MMS se hace uso de la capa TCP/IP por lo que el puerto usado es el 102.
 4. Antes de iniciar la lectura del modelo de la planta, se ha implementado una función en dicho hilo donde se realiza una comprobación de los dispositivos lógicos que hay presente en la planta que se va a operar. De este modo se puede verificar si existen todos los dispositivos lógicos necesarios para operar dicha planta. Para esta función se hace uso de *LinkedList deviceList = IedConnection_getLogicalDeviceList(con, &error).*
 5. Una vez verificado el contenido de la planta a la que se accede, se procede a realizar lecturas periódicas del modelo de datos de dicha planta a través de la función “*leer_estado_planta*”.
- **Void leer_estado_planta():** Debido a que el acceso a los datos se realiza de forma periódica, se ha implementado una función que acceda a los atributos de cada objeto de cada nodo lógico en función del tipo de dato del mismo. Para realizar el acceso a la lectura de un objeto IEC 61850 haciendo uso de la librería *libiec61850*. Para exponer los pasos necesarios para el acceso a la lectura de un DO específico se va a exponer el ejemplo del acceso DO ECPTtype:

1. La lectura del DO se realiza a través de la función *IedConnection_readObject()*, esta función devuelve un puntero del tipo *MmsValue**, de tal forma que la llamada queda:

- *MmsValue* mms_mngt_dopr1_ECPTtype = IedConnection_readObject(con, &error, "RPBMNGT/DOPR1.ECPTtype.stVal", IEC61850_FC_ST);* Donde:

- **con**: Es el objeto *IedConnection* creado anteriormente.
- **&error**: Manejador de errores.
- **"RPBMNGT/DOPR1.ECPTtype.stVal"**: Es la referencia del atributo del DO que se desea leer. En este caso como se desea obtener el valor del dicho DO se accede al atributo *.stVal*. Esta referencia viene marcada por la estructura de datos IEC 61850, donde, ¡Error! No se encuentra el origen de la referencia..



Figura 40: Estructura referencia.

- **IEC61850_FC_ST**: Es la restricción funcional (*Fuctional Constrain*) del atributo al que se desea acceder. Este valor lo marca la norma en la parte IEC 61850-7-2. Según la norma, se observa que este es un atributo de *STATUS*, por lo que para el acceso se especifica IEC61850_FC_ST (STATUS INFORMATION).

2. Una vez leído el valor, se observa en la norma (Figura 41) que dicho atributo ofrece diferentes valores numéricos y en función de este da un tipo de información. Por ello lo primero que se hace es convertir el valor *MmsValue* a un valor tipo INT para poder trabajar con él en el programa. Para ello se hace uso de *int int_mngt_dopr1_ECPTtype = MmsValue_toInt32(mms_mngt_dopr1_ECPTtype);*. Una vez que se tiene el valor convertido a una variable INT, se observa la información que proporciona, donde:

- **0**: No aplicable/Desconocido.
- **1**: Conexión de una única DER a una carga local.

- **2:** Conexión del grupo de DERs al EPS local, sirviendo una carga local.
- **3:** Conexión del EPS local con carga local al EPS de la zona (PCC).
- **4:** Conexión del EPS local sin carga local al EPS de la zona (PCC).
- **99:** Otro.

Status			
ECPID	ING	Identity of ECP	
ECPTYPE	ENS	Type of ECP	
		Value	Explanation
		0	Not applicable / Unknown
		1	Connection of one DER to local load
		2	Connection of group of DERs to local EPS serving local load
		3	Connection of local EPS with local load to area EPS (PCC)
		4	Connection of local EPS without local load to area EPS (PCC)
99	Other		
			M

Figura 41: DO ECPTYPE [3].

3. Por último, antes de validar la información se observa el estado del atributo q de dicho DO. Para ello se hace uso de la función $estado_q()$, junto con la función de acceso a dicho atributo $estado_q (ledConnection_readQualityValue(con, \&error, "RPBMNGT/DOPR1.ECPTYPE.q", IEC61850_FC_ST))$; En función del valor recibido, se hace uso de dicha información o se expone con la calidad recibida.

De esta forma, se programa el acceso a todos los DO presentes en la planta y se exponen por el terminal de forma periódica para validar el estado de la planta y ofrecer de la forma más clara posible al usuario información de la misma abstrayéndose de la norma IEC 61850. Se debe tener en cuenta que la lectura de cada atributo dentro de cada objeto se debe realizar teniendo presente la norma, ya que cada atributo ofrece un dato tipo MmsValue que posteriormente debe ser convertido en función de lo que ofrezca siguiendo la norma, pudiendo ser una variable tipo INT, tipo FLOAT, tipo STRING, o incluso tipo DBPOS como es el caso del dato obtenido del posicionamiento de los interruptores a través del DO "XCBRx.Pos". A continuación, se expone la Figura 42 con la lectura del modelo de datos desde el cliente al servidor:

```

INFORME DEL ESTADO DE LA PLANTA HELIOS:

Modo LD_MNGT (1): ON
|----> Modo LD_MNGT_LN_DCRP (1): ON
|      |----> DO PlntObSelF: Propósito/Obligación en el ECP: Funcionamiento de forma pasiva (siempre que sea posible FV
|      |----> DO PlntObBlck: Propósito/Obligación en el ECP: No funciona como Backup
|----> Modo LD_MNGT_LN_DOPR (1): ON
|      |----> DO ECPTyp: Conexión del EPS local con carga local al EPS de la zona (PCC),      q: QUALITY_VALIDITY_GOOD
|----> Modo LD_MNGT_LN_DOPM (1): ON
|----> Modo LD_MNGT_LN_DPST (1): ON
|      |----> DO ECPConn: Conectado eléctricamente al ECP,      1      q: QUALITY_VALIDITY_GOOD
|----> Modo LD_MNGT_LN_DSCH (1): ON
|----> Modo LD_MNGT_LN_DSCH (1): ON
|----> Modo LD_MNGT_LN_XCBR1 (1): ON
|      |----> DO Pos: Posición del interruptor XCBR1 (BATERIAS): ON
|----> Modo LD_MNGT_LN_XCBR2 (1): ON
|      |----> DO Pos: Posición del interruptor XCBR2 (RECTIFICADOR): ON
|----> Modo LD_MNGT_LN_XCBR3 (1): ON
|      |----> DO Pos: Posición del interruptor XCBR3 (RED): ON
|----> Modo LD_MNGT_LN_XCBR4 (1): ON
|      |----> DO Pos: Posición del interruptor XCBR4 (CARGAS_NC1): ON
|----> Modo LD_MNGT_LN_XCBR5 (1): ON
|      |----> DO Pos: Posición del interruptor XCBR5 (CARGAS_NC2): ON
|----> Modo LD_MNGT_LN_XCBR6 (1): ON
|      |----> DO Pos: Posición del interruptor XCBR6 (CARGAS_NC3): ON

Modo LD_PV (1): ON
|----> Modo LD_MNGT_LN_DPVM (1): ON
|      |----> DO MduLcFgTyp: Tipo de configuración del módulo PV: Concentrado
|      |----> DO MduLWRtg: potencia nominal en vatios pico STC: 330.00 (Pmax/W)
|      |----> DO MaxMduLV: Voltaje del módulo a máxima potencia en STC: 37.70 (Vmpp/V)
|      |----> DO MaxMduLA: Corriente del módulo a máxima potencia en STC: 8.76 (Impp/A)
|      |----> DO MduLOpnCctV: Voltaje del módulo en circuito abierto: 45.90 (Voc/V)
|      |----> DO MduLSrtCctA: Corriente del módulo en cortocircuito: 9.27 (Isc/A)
|      |----> DO MduLWmpDrt: % de reducción de la potencia en función de la Tª: -0.40 %
|      |----> DO MduLATmpDrt: % de reducción de la corriente en función de la Tª: 0.06 %
|      |----> DO MduLVmpDrt: % de reducción de la tensión en función de la Tª: -0.31 %
|----> Modo LD_MNGT_LN_DPVA (1): ON
|      |----> DO Typ: Tipo de configuración del módulo PV: Planta
|      |----> DO MduLCnt: Números de módulos por serie: 11
|      |----> DO StrgCnt: Números de series en paralelo por sub-matriz: 1
|      |----> DO SubArrCnt: Número de sub-matrices en paralelo por matriz: 3
|      |----> DO GrndConn: Tipo de conexión a tierra: Puesta a tierra positiva
|      |----> DO ArrWRtg: Potencia nominal del array (Wp): 3633.00 Wp
|----> Modo LD_MNGT_LN_DPVC (1): ON
|      |----> DO ArrModCtr: Modo seleccionado para el control de la potencia de salida de la matriz: Seguimiento del punto de máxima potencia (MPPT)
|----> Modo LD_MNGT_LN_MMDC (1): ON
|      |----> DO Watt: Potencia medida en PV: 3005.00 V
|      |----> DO Amp: Corriente medida en PV: 0.00 V
|      |----> DO Vol: Voltaje medido en PV: 48.00 V

Modo LD_BESS (1): ON
|----> Modo LD_BESS_LN_ZBAT(1): ON
|      |----> DO BatHl: Alarma Bateria alta: OFF,      q: QUALITY_VALIDITY_GOOD
|      |----> DO BatLo: Alarma Bateria alta: OFF,      q: QUALITY_VALIDITY_GOOD
|      |----> DO LoBatVal: Valor de alarma de bateria baja: 10.00 V
|      |----> DO HiBatVal: Valor de alarma de bateria alta: 15.00 V
|----> Modo LD_BESS_LN_ZBTC(1): ON
|      |----> DO ChaV: Voltaje de carga: 12.00 V
|      |----> DO ChaA: Corriente de carga: 5.00 V
|      |----> DO BatChaTyp: Tipo de carga de bateria: Voltaje constante

Modo LD_ACGRID (1): ON
|----> Modo LD_ACGRID_LN_ZRCT(1): ON
|      |----> DO IsoTyp: Tipo de aislamiento: Sin aislamiento, aislamiento de la fuente DC
|      |----> DO VRegTyp: Tipo de regulación de voltaje: Regulación de salida: voltaje fijo

Modo LD_PLOAD (1): ON
|----> Modo LD_PLOAD_LN_MMDC (1): ON
|----> Modo LD_PLOAD_LN_MMDC (1): ON
|      |----> DO Watt: Potencia medida en PLOAD: 1005.00 V
|      |----> DO Amp: Corriente medida en PLOAD: 48.00 V
|      |----> DO Vol: Voltaje medido en PLOAD: 48.00 V

Modo LD_NPLOAD (1): ON
|----> Modo LD_NPLOAD_LN_MMDC (1): ON
|----> Modo LD_NPLOAD_LN_MMDC (1): ON
|      |----> DO Watt: Potencia medida en NPLOAD: 1005.00 V
|      |----> DO Amp: Corriente medida en NPLOAD: 48.00 V
|      |----> DO Vol: Voltaje medido en PLOAD: 48.00 V

Modo LD_WEATHERSTATION (1): ON
|----> Modo LD_WEATHERSTATION_LN_MMET (1): ON
|      |----> DO EnvTmp: Temperatura ambiente: 22.00 °C
|      |----> DO EnvHum: Humedad: 5.00 %
|      |----> DO DctInsol: Irradiancia normal directa: 1265.00 W/m²
|      |----> DO HorWdSpd: Velocidad del viento horizontal: 10.00 km/h
|      |----> DO EnvPres: Presión barométrica: 955.00 hPa

```

Figura 42: Valores del modelo de la planta.

- **Void *thread2()**: Debido a la importancia y las exigencias de los mensajes GOOSE presentes en esta planta, se programa otro hilo para que trabaje en paralelo con mayor prioridad, y sea el encargado de enviar las órdenes de apertura y cierre de los interruptores de la planta a través de mensajería GOOSE. Los pasos que se siguen en este hilo son:

1. Primero, se definen los parámetros (*CommParameters* *gooseCommParameters;*) con los que se definirá el GOOSE, entre ellos los más destacables son:
 - **appId:** Este parámetro es el identificador del GOOSE y debe estar igual definido en el publicador del GOOSE y en el SUSCRIPTOR del GOOSE.
 - **dstAddress:** Este parámetro define la dirección multicast y viene definida en la norma IEC 61850. La dirección multicast seleccionada ha sido 01: 0c: cd: 01: 00: 01.
2. Una vez definidos los parámetros GOOSE, se procede a crear la instancia del GOOSE publicador a través de: *GoosePublisher publisher = GoosePublisher_create(&gooseCommParameters, interface)*, donde:
 - **&gooseCommParameters:** Son los parámetros definidos en el apartado anterior.
 - **Interface:** Es la interfaz física a través de la cual se van a realizar los intercambios de información GOOSE. En el caso del PC Linux se hace uso de “enp4s0”.
3. Una vez generada la instancia del GOOSE, se procede a generar el *dataset* compuesto por la información que se desea transmitir a través del mensaje GOOSE. Debido a que la mensajería GOOSE en este trabajo se implementa para la apertura y los cierres de los interruptores, y además que estos están relacionados unos con otros debido a que en función de la configuración que adopte la planta se abren unos interruptores y se cierran otros al mismo tiempo, se decide aprovechar el mismo GOOSE para enviar las órdenes de apertura y/o cierre de los 6 interruptores presentes en la planta al mismo tiempo. De esta forma, en vez de enviar un único GOOSE por cada orden, lo que llevaría a enviar 6 mensajes GOOSE, se decide crear un conjunto de datos (*dataset*) que informe del estado de los 6 interruptores al mismo tiempo. Para ello:
 - Se crea un *dataset* a través del objeto *LinkedList*: *LinkedList datasetValues = LinkedList_create();*
 - Posteriormente se crean los 6 valores correspondientes a los 6 estados de los 6 interruptores presentes en la planta. Según la norma IEC 61850, el estado de un interruptor puede ser:

- **2:** On.
- **1:** Off.
- **0:** Indeterminado.
- **3:** Mal estado.

Por ello, se crean 6 valores *MmsValue* creados a partir de variables de enteros de 8 bits (*int8*), y se añaden a la lista *dataset* creada anteriormente:

```
datasetValues = LinkedList_create();
LinkedList_add(datasetValues, MmsValue_newIntegerFromInt8(2)); //XCBR1: ON=2, OFF=1, INTERMEDIATE=0, BAD_STATE=3
LinkedList_add(datasetValues, MmsValue_newIntegerFromInt8(2)); //XCBR2: ON=2, OFF=1, INTERMEDIATE=0, BAD_STATE=3
LinkedList_add(datasetValues, MmsValue_newIntegerFromInt8(2)); //XCBR3: ON=2, OFF=1, INTERMEDIATE=0, BAD_STATE=3
LinkedList_add(datasetValues, MmsValue_newIntegerFromInt8(2)); //XCBR4: ON=2, OFF=1, INTERMEDIATE=0, BAD_STATE=3
LinkedList_add(datasetValues, MmsValue_newIntegerFromInt8(2)); //XCBR5: ON=2, OFF=1, INTERMEDIATE=0, BAD_STATE=3
LinkedList_add(datasetValues, MmsValue_newIntegerFromInt8(2)); //XCBR6: ON=2, OFF=1, INTERMEDIATE=0, BAD_STATE=3
```

Se debe tener en cuenta que por cada modificación del *dataset* enviado por el GOOSE, la norma IEC 61850 marca que se debe incrementar en una unidad el parámetro *StNum*: *GoosePublisher_increaseStNum(publisher)*. El parámetro *SqNum* se actualiza de forma automática a través de la instancia *Goosepublisher*.

- Una vez generado el *dataset* con sus 6 valores, y actualizado los valores de configuración del servicio de mensajería GOOSE, se procede a realizar la publicación de dicho *dataset*. Se debe tener en cuenta que la librería *libiec61850* te ofrece una solución para implementar un sistema de comunicaciones basado en la norma IEC 61850, pero es el usuario el encargado de hacer cumplir la norma a través de dicha librería. Esto es importante debido a que la librería te ofrece el manejo del servicio de mensajería GOOSE abstrayéndote de las capas más bajas de comunicación, pero luego es necesario que se implemente el sistema de bombardeo GOOSE explicado en 2.1.8. Por ello, en función de si se ha modificado el *dataset*, el sistema realiza el envío periódicamente en *T0* o se realiza el bombardeo GOOSE desde el inicio incrementando los valores según la Figura 12. Los tiempos seleccionados para este trabajo han sido:
 - **T0:** 1000 ms.
 - **T1:** 2ms.
 - **T2:** 4ms.
 - **T3:** 6ms.

4.6.2 Programación Servidor/Suscriptor (Raspberry Pi)

La parte del Servidor/Suscriptor se implementa en la Raspberry Pi. Las librerías utilizadas son:

- **Librería Hilos:** Al igual que en la parte del Cliente/Publicador y con la finalidad de poder trabajar en paralelo y de forma óptima, las tareas como se exponía en la Figura 38 se han implementado en diferentes hilos.
 - `#include <pthread.h>`
- **Librería Servidor:** Aquí se engloban las librerías necesarias para implementar el servidor IEC 61850.
 - `#include "iec61850_server.h"`
- **Librerías GOOSE:** Para poder hacer uso del servicio de mensajería GOOSE ha sido necesario implementar las librerías:
 - `#include "goose_receiver.h"`
 - `#include "goose_subscriber.h"`
- **Librería tiempos y sincronización:** Para implementar un *timer* para supervisar el canal de comunicación GOOSE y para medir tiempos una vez que se tenía los dispositivos sincronizados por NTP la librería a usar ha sido:
 - `#include <sys/time.h>`
- **Otras librerías:** Otras librerías necesarias para el desarrollo del sistema han sido:
 - `#include <signal.h>`
 - `#include <stdlib.h>`
 - `#include <stdio.h>`
 - `#include "hal_thread.h"`
 - `#include <math.h>`
 - `#include <unistd.h>`

Una vez expuestas las librerías necesarias para implementar el programa, se procede a explicar las funciones presentes en el mismo. En la Figura 43, se expone un flujograma del funcionamiento del software implementado en el Servidor/Suscriptor. Como se observa, el programa parte de la función principal `Main()`, la cual se encarga de inicializar los servicios de comunicación, tanto el MMS

como el GOOSE, de realizar la llamada a la función que se encarga de gestionar el inicio del modelo, y por último, inicializa los hilos.

Lo primero que realiza la función Main() es llamar a la función inicio_modelo(), esta función se encarga de inicializar el modelo IEC 61850 ubicado en el Servidor/Suscriptor. Es una función muy útil ya que se puede modificar a mano o enlazar con un SCADA para modificar los valores base del sistema, sin tener la necesidad de inicializar el modelo .cid a través de un software comercial.

Una vez que el modelo se ha inicializado se arrancan tantos los hilos como el manejador de los mensajes GOOSE. El hilo 1 (Thread1()), se encarga de la lectura de las variables reales de la planta, y posteriormente de actualizar con esos valores el modelo IEC 61850. El hilo 2 (Thread2()), se encarga de supervisar el canal de comunicaciones GOOSE, y, por último, el manejador GOOSE (GOOSE LISTENER), se encarga de la recepción y lectura de los mensajes GOOSE enviados por el Cliente/Publicador. Además, una vez que obtiene la información recibida a través del mensaje GOOSE, la función GOOSE LISTENES se encarga de la apertura y el cierre reales de los interruptores de la planta, y posteriormente de la actualización de los mismos en el modelo IEC 61850.

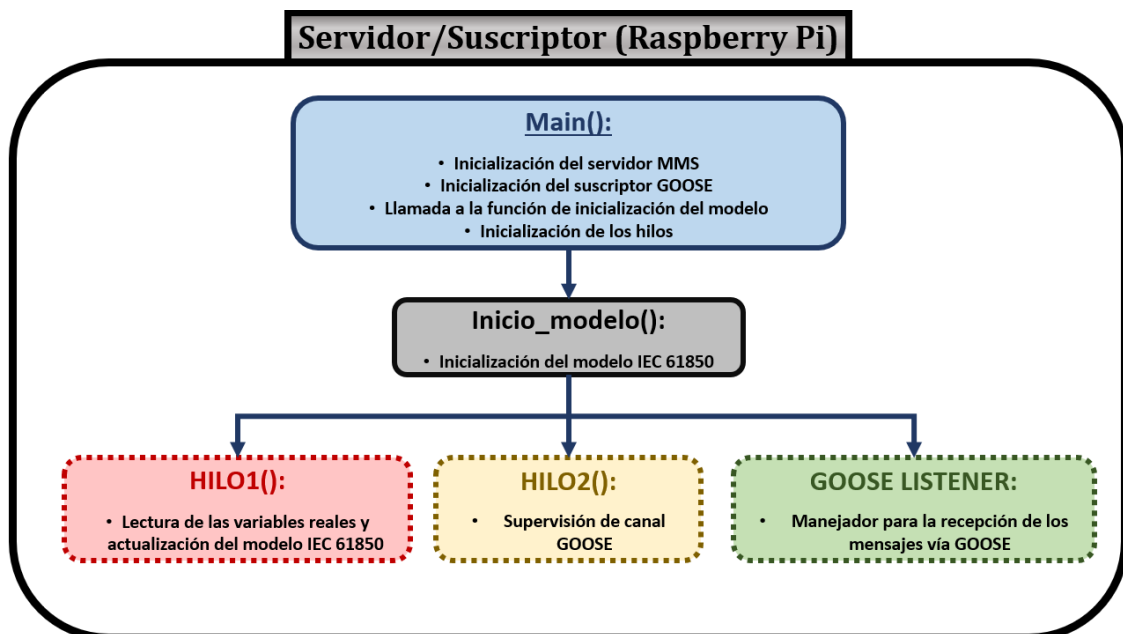


Figura 43: Flujograma programa del Servidor/Suscriptor.

A Continuación, se procede a realizar una explicación más detallada de la programación de cada función:

- **Main:** En la función principal del programa se configuran las partes del servidor MMS y la parte del suscriptor GOOSE.
 - **Servidor MMS:** En primer lugar se crea el objeto de configuración del servidor: `IedServerConfig config = IedServerConfig_create();` y posteriormente se configuran los parámetros de conexión:

```

/* Set buffer size for buffered report control blocks to 200000 bytes */
IedServerConfig_setReportBufferSize(config, 200000);

/* Set stack compliance to a specific edition of the standard (WARNING: data model has also to be checked for compliance) */
IedServerConfig_setEdition(config, IEC_61850_EDITION_2);

/* Set the base path for the MMS file services */
IedServerConfig_setFileServiceBasePath(config, "./vmd-filestore/");

/* disable MMS file service */
IedServerConfig_enableFileService(config, false);

/* enable dynamic data set service */
IedServerConfig_enableDynamicDataSetService(config, true);

/* disable log service */
IedServerConfig_enableLogService(config, false);

/* set maximum number of clients */
IedServerConfig_setMaxMmsConnections(config, 2);

/* Create a new IEC 61850 server instance */
iedServer = IedServer_createWithConfig(&iedModel, NULL, config);

/* configuration object is no longer required */
IedServerConfig_destroy(config);

/* Install handler for operate command */
IedServer_setConnectionIndicationHandler(iedServer, (IedConnectionIndicationHandler) connectionHandler, NULL);

/* MMS server will be instructed to start listening for client connections. */
IedServer_start(iedServer, 102);

```

Adicionalmente se deben añadir los permisos de escritura en el modelo desde el cliente:

```

//Comando necesario para permitir la escritura desde el cliente al modelo de datos IEC61850_FC
IedServer_setWriteAccessPolicy(iedServer, IEC61850_FC_MX, ACCESS_POLICY_ALLOW);
IedServer_setWriteAccessPolicy(iedServer, IEC61850_FC_DC, ACCESS_POLICY_ALLOW);
IedServer_setWriteAccessPolicy(iedServer, IEC61850_FC_ST, ACCESS_POLICY_ALLOW);

```

- **Suscriptor GOOSE:** En esta parte se configura la parte de la suscripción GOOSE.
 - Lo primero, se crea el objeto de recepción del GOOSE `GooseReceiver receiver = GooseReceiver_create();` posteriormente se fija la interfaz a través de la cual se recibe la mensajería GOOSE, en este caso "eth0": `GooseReceiver_setInterfaceId(receiver, "eth0");`
 - A continuación, se crea el objeto de suscripción al GOOSE `GooseSubscriber subscriber = GooseSubscriber_create("RPBMNGT/XCBRGOgcbDbpos", NULL),` donde

“*RPBMNGT/XCBR\$GO\$gcbDbpos*” es el nombre del dataset recibido a través del mensaje GOOSE.

- Se debe fijar el *AppId*, este parámetro debe coincidir con el *AppId*, fijado en el publicador GOOSE.
 - A continuación, se genera el manejador de la suscripción GOOSE: *GooseSubscriber_setListener(subscriber, gooseListener, NULL)*;
 - Se adjunta la suscripción configurada a la recepción del GOOSE creada: *GooseReceiver_addSubscriber(receiver, subscriber)*.
 - Por último, se inicia la recepción del GOOSE: *GooseReceiver_start(receiver)*.
- **Llamada inicialización modelo:** A continuación, en la función principal se llama a la función que inicializa el modelo de la planta Helios presente en este trabajo. Esta función “*inico_modelo()*” se expondrá a continuación.
 - **Inicialización del hilo:** Por último, se inicializa el hilo que posteriormente maneja el *timer* para supervisar el canal de comunicación GOOSE *pthread_create (&h1, NULL, thread1, NULL)* y *pthread_join(h1, NULL)*.
- **Inicio_modelo():** Con el objetivo de ofrecer una mayor versatilidad al sistema desarrollado para el proyecto Helios, y con la idea de independizar el sistema de otras herramientas comerciales, este trabajo desarrolla la función “*inicio_modelo()*”, la cual ofrece la posibilidad de inicializar todas las variables del sistema. Esto es muy útil ya que, si la planta sufre modificaciones tanto del sistema completo como de alguno de sus componentes, a través de esta función puede actualizarse dichos parámetros si necesitar de una herramienta externa para inicializar el modelo IEC 61850. Además, ofrece la posibilidad de *vincular* dichas variables a un SCADA externo para ofrecer una aplicación más familiar a cualquier usuario a la hora de inicializar el modelo de la planta Helios. Para la inicialización del modelo se debe realizar la escritura con la información deseada de cada uno de los parámetros presentes en la planta y digitalizarlos a través del modelo *.cid* IEC 61850. Para la escritura de cada uno de los objetos se debe:

- Obtener la marca de tiempo para actualizar el modelo con el *timeStamp* adecuado:

```
//Obtenemos Timestamp para actuaizar en modelo
uint64_t timestamp = Hal_getTimeInMs();
Timestamp iecTimestamp;
Timestamp_clearFlags(&iecTimestamp);
Timestamp_setTimeInMilliseconds(&iecTimestamp, timestamp);
Timestamp_setLeapSecondKnown(&iecTimestamp, true);
Timestamp_setClockNotSynchronized(&iecTimestamp, true);
```

- Antes de realizar la escritura en el modelo, se debe bloquear previamente para evitar problemas de accesos simultáneos a través de diferentes clientes. Previamente se bloquea el modelo y posteriormente se debe desbloquear:

```
IedServer_lockDataModel(iedServer);

//Una vez actualizadas todas las variables se desbloquea
IedServer_unlockDataModel(iedServer);
```

- Posteriormente, se procede a actualizar el atributo deseado. Para ello, se debe actualizar la calidad y el atributo *Mod* conforme a la norma y posteriormente hacer uso de una de las funciones presentes en la librería *libiec61850* para actualizar cada atributo conforme a la norma en función de si dicho atributo parte de una variable INT, BOOL o DBPOS. Para la actualización de los atributos, se hace uso de una de las funciones *IedServer_updateXXAttributeValue(iedServer, DataAttribute, Value)*, donde:

- **XX:** Hace referencia al tipo de variable a actualizar.
- **iedServer:** Es el objeto manejador del servidor creado previamente.
- **DataAttribute:** Es la referencia necesaria que dirección al atributo que se desea modificar. Por ejemplo, si se desea modificar el atributo *stVal* (valor), del objeto *pos* (posición) del *LN* (nodo lógico) *XCBR*, contenido en el *LD* (dispositivo lógico) *MNGT*, el *DataAttribute* será: *IEDMODEL_MNGT_XCBR6_Pos_stVal*.

A continuación, se presenta el ejemplo de la actualización del posicionamiento de un interruptor:

```

/*LN XCBR6*/
//Mod LN XCBR6 (test,test/blocked,on...)
IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MNGT_XCBR6_Beh_t, &iecTimestamp);
IedServer_updateQuality(iedServer, IEDMODEL_MNGT_XCBR6_Beh_q, QUALITY_VALIDITY_GOOD);
IedServer_updateInt32AttributeValue(iedServer, IEDMODEL_MNGT_XCBR6_Beh_stVal, 1);
IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MNGT_XCBR6_Mod_t, &iecTimestamp);
IedServer_updateQuality(iedServer, IEDMODEL_MNGT_XCBR6_Mod_q, QUALITY_VALIDITY_GOOD);
IedServer_updateInt32AttributeValue(iedServer, IEDMODEL_MNGT_XCBR6_Mod_stVal, 1);
//DO Pos: Conexión de la planta DER en el ECP
IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_MNGT_XCBR6_Pos_t, &iecTimestamp);
IedServer_updateQuality(iedServer, IEDMODEL_MNGT_XCBR6_Pos_q, QUALITY_VALIDITY_GOOD);
IedServer_updateDbposValue(iedServer, IEDMODEL_MNGT_XCBR6_Pos_stVal,DBPOS_ON);

```

En este otro extracto de código se refleja la actualización de los objetos y los atributos del LN DPVM del LD PV:

```

/*LD PV*/
//Modo del LD MNGT Mod (test,test/blocked,on...)
IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_PV_LLNO_Beh_t, &iecTimestamp);
IedServer_updateQuality(iedServer, IEDMODEL_PV_LLNO_Beh_q, QUALITY_VALIDITY_GOOD);
IedServer_updateInt32AttributeValue(iedServer, IEDMODEL_PV_LLNO_Beh_stVal, 1);
IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_PV_LLNO_Mod_t, &iecTimestamp);
IedServer_updateQuality(iedServer, IEDMODEL_PV_LLNO_Mod_q, QUALITY_VALIDITY_GOOD);
IedServer_updateInt32AttributeValue(iedServer, IEDMODEL_PV_LLNO_Mod_stVal, 1);
/*LN DPVM*/
//Mod LN DPVM (test,test/blocked,on...)
IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_PV_DPVM1_Beh_t, &iecTimestamp);
IedServer_updateInt32AttributeValue(iedServer, IEDMODEL_PV_DPVM1_Beh_stVal, 1);
IedServer_updateQuality(iedServer, IEDMODEL_PV_DPVM1_Beh_q, QUALITY_VALIDITY_GOOD);
IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_PV_DPVM1_Mod_t, &iecTimestamp);
IedServer_updateQuality(iedServer, IEDMODEL_PV_DPVM1_Mod_q, QUALITY_VALIDITY_GOOD);
IedServer_updateInt32AttributeValue(iedServer, IEDMODEL_PV_DPVM1_Mod_stVal, 1);
//DO MdulCfgTyp: Tipo de configuración del módulo PV
IedServer_updateInt32AttributeValue(iedServer, IEDMODEL_PV_DPVM1_MdulCfgTyp_setVal, 2);
//DO MdulWrtg: Módulo de potencia nominal en vatios pico STC
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_PV_DPVM1_MdulWrtg_setMag_f, 330);
//DO MdulW200Rtg: Módulo de potencia nominal en vatios pico a 200 W/m²
//IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_PV_DPVM1_MdulW200Rtg_setMag, 320);
//DO MaxMdulV: Voltaje del módulo a máxima potencia en STC
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_PV_DPVM1_MaxMdulV_setMag_f, 37.70);
//DO MaxMdulA: Corriente del módulo a máxima potencia en STC
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_PV_DPVM1_MaxMdulA_setMag_f, 8.76);
//DO MdulOpnCctV: Voltaje del módulo en circuito abierto
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_PV_DPVM1_MdulOpnCctV_setMag_f, 45.9);
//DO MdulSrtCctA: Corriente del módulo en cortocircuito
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_PV_DPVM1_MdulSrtCctA_setMag_f, 9.27);
//DO MdulWtmpDrt: % de reducción de la potencia en función de la temperatura (potencia/temperatura) por encima de 25°
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_PV_DPVM1_MdulWtmpDrt_setMag_f, -0.40);
//DO MdulATmpDrt: % de reducción de la corriente en función de la temperatura (potencia/temperatura) por encima de 25°
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_PV_DPVM1_MdulATmpDrt_setMag_f, 0.06);
//DO MdulVTmpDrt: % de reducción de la tensión en función de la temperatura (potencia/temperatura) por encima de 25°
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_PV_DPVM1_MdulVTmpDrt_setMag_f, -0.31);

```

- **Thread10):** Como se ha comentado en la Figura 38, el hilo 1 de la parte del servidor/suscriptor, se encarga de la lectura de las variables eléctricas del sistema y posteriormente de la actualización del modelo de la planta Helios IEC 61850 con dichas variables. Este trabajo se centra en la implementación de un sistema de comunicaciones basado en la norma IEC 61850 por lo que simula la lectura de las variables reales y se centra en la actualización del modelo de la planta IEC 61850.

Una vez obtenida la variable eléctrica del sistema real, los pasos para actualizar el modelo serán:

- o Obtener la marca de tiempo para actualizar el *TimeStamp* de la variable a actualizar:

```
//Una vez leída la variable se modifica el valor de la misma en el modelo
uint64_t timestamp = Hal_getTimeInMs();
Timestamp iecTimestamp;
Timestamp_clearFlags(&iecTimestamp);
Timestamp_setTimeInMilliseconds(&iecTimestamp, timestamp);
Timestamp_setLeapSecondKnown(&iecTimestamp, true);
/* toggle clock-not-synchronized flag in timestamp */
if (((int) t % 2) == 0)
    Timestamp_setClockNotSynchronized(&iecTimestamp, true);
```

- o Posteriormente, la actualización del modelo se realiza a través de las mismas funciones que la inicialización, pero en este caso el valor a actualizar será el valor leído de la planta:

```
/*LD PV*/
/*LN MMDC*/
//Mod LN MMDC (test,test/blocked,on...)
IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_PV_MMDC1_Beh_t, &iecTimestamp);
IedServer_updateInt32AttributeValue(iedServer, IEDMODEL_PV_MMDC1_Beh_stVal, 1);
IedServer_updateQuality(iedServer, IEDMODEL_PV_MMDC1_Beh_q, QUALITY_VALIDITY_GOOD);
IedServer_updateTimestampAttributeValue(iedServer, IEDMODEL_PV_MMDC1_Mod_t, &iecTimestamp);
IedServer_updateQuality(iedServer, IEDMODEL_PV_MMDC1_Mod_q, QUALITY_VALIDITY_GOOD);
IedServer_updateInt32AttributeValue(iedServer, IEDMODEL_PV_MMDC1_Mod_stVal, 1);
//DO Watt: Potencia PV
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_PV_MMDC1_Watt_mag_f, pv_watt);
//DO Amp: Corriente PV
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_PV_MMDC1_Amp_mag_f, pv_amp);
//DO Vol: Tensión PV
IedServer_updateFloatAttributeValue(iedServer, IEDMODEL_PV_MMDC1_Vol_mag_f, pv_vol);
```

- **Thread2()**: Este hilo se encarga de gestionar el *timer* encargado de supervisar el canal GOOSE, de modo que, si ante la recepción de un mensajes GOOSE donde el stNum no se ha modificado, se sabe por la norma que deberá volverse a recibir pasado un periodo de tiempo T_0 (Figura 12), por lo que si se detecta que pasados $T_0 + 0,2 * T_0$ no se ha recibido el GOOSE que se debería haber recibido, se activa una alarma y se da un aviso al usuario de “posible fallo de comunicación GOOSE”. Este canal de supervisión es muy importante debido a la importancia de los mensajes GOOSE en la planta ya que se encargan de la apertura y el cierre de los interruptores. Ante una necesidad de desconexión de la misma se perdería la posibilidad de realizar la apertura de forma remoto si el canal de comunicación GOOSE no se encuentra activo. Por ello, esta alarma debe ser gestionada con prioridad y mandar a personal a revisar la instalación para dejar la planta en una combinación de interruptores segura de forma manual en el caso de que no se recuperen los canales de comunicación.

- **GOOSE LISTENER:** Esta función se encarga de manejar la recepción de los GOOSE. Sus principales funciones son:
 - Una vez recibido el GOOSE arranca el *timer* para la supervisión del canal GOOSE.
 - Posteriormente obtiene el valor del *StNum* del GOOSE recibido y lo compara con el del anterior. En el caso de tener el mismo valor, como no se ha producido ninguna modificación en el *dataset*, descarta el GOOSE y no realiza ningún proceso adicional.
 - En el caso de que el *StNum* sea diferente al último almacenado, actualiza dicho valor con el nuevo *StNum* y se procede a descomponer el *dataset* para obtener la información sobre el estado y la actualización que se debe realizar de los interruptores de la planta.

Posteriormente, se observa el nuevo valor del estado de cada interruptor, en el caso de que el valor sea igual al estado actual no se realiza ninguna acción. Por el contrario, si el valor se ha modificado, se procede a realizar la apertura o el cierre de dicho interruptor y, posteriormente, se actualiza el valor del mismo en el modelo IEC 61850.

En el caso del interruptor 3 *XCBR3*, además de actualizar el *DO pos* de los LN *XCBR*, se actualizan también los *DO ECPTtype* y *ECPCConn* debido a que en este caso la planta se desconecta o conecta de la Red Eléctrica por lo que cambia el estado de la misma.

CAPÍTULO 5: PRUEBAS EXPERIMENTALES

A la hora de realizar pruebas con el sistema se deben valorar diferentes aspectos. El primero es comprobar que el sistema cumple las necesidades impuestas por el proyecto Helios donde se va a implementar. Estos requerimientos son principalmente probar que la comunicación bajo el estándar IEC 61850 es correcta, y la información viaja en ambos sentidos tanto desde las Raspberry Pi hacia el ordenador Linux como viceversa. También es importante probar las funcionalidades extras explicadas en el trabajo como la supervisión del canal GOOSE.

Una vez comprobado el correcto funcionamiento del sistema diseñado, al haber basado el mismo en un estándar internacional cuyo objetivo principal es la interoperabilidad entre sistemas, se debe verificar dichos requerimientos. De este modo, se debe conectar el sistema a clientes externos de diferentes fabricantes de tal modo que se pueda comprobar que son capaces de captar la información presente en el sistema incluso modificar los parámetros necesarios.

Por último, para verificar el correcto funcionamiento del sistema, una vez que se han indicado los requerimientos de interoperabilidad se deben comprobar los requerimientos de tiempos de forma que todas las transmisiones de información presentes en el sistema cumplan los requerimientos fijados tanto por la norma IEC 61850 como por la ingeniería de diseño del sistema.

5.1.1 PRUEBAS ENTRE LAS RASPBERRY PI Y EL PC

Para realizar estas pruebas se conectan la Raspberry Pi y el PC Linux a través de un cable Ethernet. Para comprobar el correcto funcionamiento se hace uso de la consola de comandos, donde se imprime la información IEC 61850.

En la Raspberry Pi lo primero que se imprime es una pequeña cabecera mencionando la versión de la librería con la que se está trabajando. Esto es de utilidad ya que puede ser necesario la actualización de las librerías para obtener nuevas funcionalidades IEC 61850. También se imprime la interfaz a través de la cual se van a recibir los GOOSE, y posteriormente, se imprimen los GOOSE recibidos con las órdenes de apertura (1) o cierre (2) de los interruptores de la planta. Solo se imprimen los GOOSE que se reciben con nuevo stNum, es decir, con una modificación en el *dataset*. Además de estos valores, se imprime por pantalla las

acciones que conllevan el *dataset* recibido (*CERRAR XCBRx* y/o *ABRIR XCBRx*), junto con el conjunto de 6 valores reales que contiene el *dataset* (Figura 44).

```
Using libIEC61850 version 1.4.0
Using interface eth0

Arrancamos Hilo 1: Lectura de variables y actualización del modelo de la planta

GOOSE event:
  stNum: 1 sqNum: 0
  timeToLive: 1000
  timestamp: 1611168665.468
  CERRAR XCBR1
  CERRAR XCBR2
  CERRAR XCBR3
  CERRAR XCBR4
  CERRAR XCBR5
  CERRAR XCBR6
  {2,2,2,2,2,2}

GOOSE event:
  stNum: 2 sqNum: 0
  timeToLive: 1000
  timestamp: 1611168676.477
  ABRIR XCBR1
  ABRIR XCBR2
  ABRIR XCBR3
  ABRIR XCBR4
  ABRIR XCBR5
  ABRIR XCBR6
  {1,1,1,1,1,1}

GOOSE event:
  stNum: 3 sqNum: 0
  timeToLive: 1000
  timestamp: 1611168688.583
  CERRAR XCBR1
  CERRAR XCBR2
  CERRAR XCBR4
  CERRAR XCBR5
  CERRAR XCBR6
  {2,2,1,2,2,2}

GOOSE event:
  stNum: 4 sqNum: 0
  timeToLive: 1000
  timestamp: 1611168700.688
  ABRIR XCBR1
  CERRAR XCBR3
  {1,2,2,2,2,2}
```

Figura 44: GOOSE en el suscriptor.

Si se observan los GOOSE recibidos con stNum 3 y 4, las acciones que se realizan reales son únicamente la apertura y cierre de los interruptores que han visto modificados su estado en el *dataset* recibido.

Por otro lado, se aprovecha a simular un fallo en el canal de comunicaciones GOOSE. Para realizar esta simulación se genera un tiempo superior de espera entre el envío de uno de los mensajes GOOSE de bombardeo con stNum 1 y se observa como

aparece una alarma en la consola de comandos avisando al usuario de que se ha perdido el canal de comunicaciones (Figura 45).

```
Using libIEC61850 version 1.4.0
Using interface eth0

Arrancamos Hilo 1: Lectura de variables y actualización del modelo de la planta

GOOSE event:
stNum: 1 sqNum: 0
timeToLive: 1000
timestamp: 1611168960.194
  CERRAR XCBR1
  CERRAR XCBR2
  CERRAR XCBR3
  CERRAR XCBR4
  CERRAR XCBR5
  CERRAR XCBR6
  {2,2,2,2,2,2}

¡ALARMA: PERDIDA DEL CANAL GOOSE!

GOOSE event:
stNum: 2 sqNum: 0
timeToLive: 1000
timestamp: 1611168971.202
  ABRIR XCBR1
  ABRIR XCBR2
  ABRIR XCBR3
  ABRIR XCBR4
  ABRIR XCBR5
  ABRIR XCBR6
  {1,1,1,1,1,1}

GOOSE event:
stNum: 3 sqNum: 0
timeToLive: 1000
timestamp: 1611168983.313
  CERRAR XCBR1
  CERRAR XCBR2
  CERRAR XCBR4
  CERRAR XCBR5
  CERRAR XCBR6
  {2,2,1,2,2,2}

GOOSE event:
stNum: 4 sqNum: 0
timeToLive: 1000
timestamp: 1611168995.422
  ABRIR XCBR1
  CERRAR XCBR3
  {1,2,2,2,2,2}
```

Figura 45: Pérdida del canal de comunicaciones.

Por la parte del PC Linux, se hace de igual manera uso de la consola para imprimir la información captada del modelo IEC 61850.

Lo primero que se imprime es un pequeño título e información de la inicialización de los hilos que se van a ejecutar. Seguidamente, se expone un pequeño listado con el nombre del IED detectado, junto con los LD que contiene dicho IED (Figura 46).

```

COMUNICACIÓN DE LA PLANTA HELIOS BAJO EL ESTANDAR IEC 61850

- Arranca Hilo 1 para lectura del estado de la planta
- Arranca Hilo 2 para enviar GOOSE en caso de que sea necesario

DETECCIÓN DINÁMICA DE LOS IED Y LOS LD PRESENTES EN EL SERVIDOR:
IED: RPB
  |-->LD: RPBACGRID
  |-->LD: RPBESS
  |-->LD: RPBNGT
  |-->LD: RPBNPLOAD
  |-->LD: RPBLOAD
  |-->LD: RBPV
  |-->LD: RPBWEATHERSTATION
    
```

Figura 46: Extracción de los LD del servidor.

Posteriormente y de forma periódica se van actualizando los datos adquiridos del modelo de la planta IEC 61850 actualizado en la Raspberry Pi. Para ello se imprime por pantalla en el PC Linux los valores leídos de la misma (Figura 47).

```

INFORME DEL ESTADO DE LA PLANTA HELIOS:

Modo LD_MNGT (1): ON
|---->Modo LD_MNGT_LN_DCRP (1): ON
|   |--> DO PlntObsSelf: Próposito/Obligación en el ECP: Funcionamiento de forma pasiva (siempre que sea posible FV)
|   |--> DO PlntObsBck: Próposito/Obligación en el ECP: No funciona como BackUp
|---->Modo LD_MNGT_LN_DOPR (1): ON
|   |--> DO ECPType: conexión del EPS local con carga local al EPS de la zona (PCC),      q: QUALITY_VALIDITY_GOOD
|---->Modo LD_MNGT_LN_DOPM (1): ON
|---->Modo LD_MNGT_LN_DPST (1): ON
|   |--> DO ECPConn: Conectado eléctricamente al ECP,      1      q: QUALITY_VALIDITY_GOOD
|---->Modo LD_MNGT_LN_DSCC (1): ON
|---->Modo LD_MNGT_LN_DSCH (1): ON
|---->Modo LD_MNGT_LN_XCBR1 (1): ON
|   |--> DO Pos: Posición del interruptor XCBR1 (BATERIAS): ON
|---->Modo LD_MNGT_LN_XCBR2 (1): ON
|   |--> DO Pos: Posición del interruptor XCBR2 (RECTIFICADOR): ON
|---->Modo LD_MNGT_LN_XCBR3 (1): ON
|   |--> DO Pos: Posición del interruptor XCBR3 (RED): ON
|---->Modo LD_MNGT_LN_XCBR4 (1): ON
|   |--> DO Pos: Posición del interruptor XCBR4 (CARGAS_NC1): ON
|---->Modo LD_MNGT_LN_XCBR5 (1): ON
|   |--> DO Pos: Posición del interruptor XCBR5 (CARGAS_NC2): ON
|---->Modo LD_MNGT_LN_XCBR6 (1): ON
|   |--> DO Pos: Posición del interruptor XCBR6 (CARGAS_NC3): ON

Modo LD_PV (1): ON
|---->Modo LD_MNGT_LN_DPVW (1): ON
|   |--> DO MdulCfgTyp: Tipo de configuración del módulo PV: Concentrado
|   |--> DO MdulWrTg: potencia nominal en vatios pico STC: 330.00 (Pmax/W)
|   |--> DO MaxMdulV: Voltaje del módulo a máxima potencia en STC: 37.70 (Vmpp/V)
|   |--> DO MaxMdulA: corriente del módulo a máxima potencia en STC: 8.76 (Impp/A)
|   |--> DO MdulOpnCctV: Voltaje del módulo en circuito abierto: 45.90 (Vcc/V)
|   |--> DO MdulSrtCctA: Corriente del módulo en cortocircuito: 9.27 (Isc/A)
|   |--> DO MdulVmpDrt: % de reducción de la potencia en función de la T': -0.40 %
|   |--> DO MdulVmpDrt: % de reducción de la corriente en función de la T': 0.06 %
|   |--> DO MdulVmpDrt: % de reducción de la tensión en función de la T': -0.31 %
|---->Modo LD_MNGT_LN_DPVVA (1): ON
|   |--> DO Typ: Tipo de configuración del módulo PV: Planta
|   |--> DO MdulCnt: Números de módulos por serie: 11
|   |--> DO StrgCnt: Números de series en paralelo por sub-matriz: 1
|   |--> DO SubArrCnt: Número de sub-matrices en paralelo por matriz: 3
|   |--> DO GrndConn: Tipo de conexión a tierra: Puesta a tierra positiva
|   |--> DO ArrWrTg: Potencia nominal del array (Wp): 3633.00 Wp
|---->Modo LD_MNGT_LN_DPVV (1): ON
|   |--> DO ArrModCtr: Modo seleccionado para el control de la potencia de salida de la matriz: Seguimiento del punto de máxima potencia (MPPT)
|---->Modo LD_MNGT_LN_MMDC (1): ON
|   |--> DO Watt: Potencia medida en PV: 3005.00 V
|   |--> DO Amp: Corriente medida en PV: 0.00 V
|   |--> DO Vol: Voltaje medido en PV: 48.00 V
|   |--> DO BatHl: Alarma Batería alta: OFF,      q: QUALITY_VALIDITY_GOOD
|   |--> DO BatLo: Alarma Batería alta: OFF,      q: QUALITY_VALIDITY_GOOD
|   |--> DO LoBatVal: Valor de alarma de batería baja: 10.00 V
|   |--> DO HlBatVal: Valor de alarma de batería alta: 15.00 V
|---->Modo LD_BESS_LN_ZBTC(1): ON
|   |--> DO ChAv: Voltaje de carga: 12.00 V
|   |--> DO ChA: Corriente de carga: 5.00 V
|   |--> DO BatChaTyp: Tipo de carga de batería: Voltaje constante

Modo LD_ACGRID (1): ON
|---->Modo LD_ACGRID_LN_ZRCT(1): ON
|   |--> DO IsoTyp: Tipo de aislamiento: Sin aislamiento, aislamiento de la fuente DC
|   |--> DO VRegTyp: Tipo de regulación de voltaje: Regulación de salida: voltaje fijo

Modo LD_PLOAD (1): ON
|---->Modo LD_PLOAD_LN_MMDC (1): ON
|---->Modo LD_PLOAD_LN_MMDC (1): ON
|   |--> DO Watt: Potencia medida en PLOAD: 1005.00 V
|   |--> DO Amp: Corriente medida en PLOAD: 48.00 V
|   |--> DO Vol: Voltaje medido en PLOAD: 48.00 V

Modo LD_NPLOAD (1): ON
|---->Modo LD_NPLOAD_LN_MMDC (1): ON
|---->Modo LD_NPLOAD_LN_MMDC (1): ON
|   |--> DO Watt: Potencia medida en NPLOAD: 1005.00 V
|   |--> DO Amp: Corriente medida en NPLOAD: 48.00 V
|   |--> DO Vol: Voltaje medido en PLOAD: 48.00 V

Modo LD_WEATHERSTATION (1): ON
|---->Modo LD_WEATHERSTATION_LN_MMET (1): ON
|   |--> DO EnvTmp: Temperatura ambiente: 22.00 °C
|   |--> DO EnvHum: Humedad: 5.00 %
|   |--> DO DctInsol: Irradiancia normal directa: 1265.00 W/m²
|   |--> DO HorWdSpd: Velocidad del viento horizontal: 10.00 km/h
|   |--> DO EnvPres: Presión barométrica: 955.00 hPa
    
```



```

INFORME DEL ESTADO DE LA PLANTA HELIOS:

Modo LD_MNGT (1): ON
----> Modo LD_MNGT_LN_DCRP (1): ON
|-----> DO PlntObsSelf: Proposito/Obligación en el ECP: Funcionamiento de forma pasiva (siempre que sea posible FV
|-----> DO PlntObsBack: Proposito/Obligación en el ECP: No funciona como BackUp
---->Modo LD_MNGT_LN_DOPR (1): ON
|-----> DO ECPType: Conexión de una única DER a una carga local, q: QUALITY_VALIDITY_GOOD
---->Modo LD_MNGT_LN_DOPM (1): ON
---->Modo LD_MNGT_LN_DPST (1): ON
|-----> DO ECPConn: No conectado eléctricamente al ECP, 0 q: QUALITY_VALIDITY_GOOD
---->Modo LD_MNGT_LN_DSCC (1): ON
---->Modo LD_MNGT_LN_DSCH (1): ON
---->Modo LD_MNGT_LN_XCBR1 (1): ON
|-----> DO Pos: Posición del interruptor XCBR1 (BATERIAS): OFF
---->Modo LD_MNGT_LN_XCBR2 (1): ON
|-----> DO Pos: Posición del interruptor XCBR2 (RECTIFICADOR): OFF
---->Modo LD_MNGT_LN_XCBR3 (1): ON
|-----> DO Pos: Posición del interruptor XCBR3 (RED): OFF
---->Modo LD_MNGT_LN_XCBR4 (1): ON
|-----> DO Pos: Posición del interruptor XCBR4 (CARGAS_NC1): OFF
---->Modo LD_MNGT_LN_XCBR5 (1): ON
|-----> DO Pos: Posición del interruptor XCBR5 (CARGAS_NC2): OFF
---->Modo LD_MNGT_LN_XCBR6 (1): ON
|-----> DO Pos: Posición del interruptor XCBR6 (CARGAS_NC3): OFF

Modo LD_PV (1): ON
---->Modo LD_MNGT_LN_DPVM (1): ON
|-----> DO MduLCfgTyp: Tipo de configuración del módulo PV: Concentrado
|-----> DO MduLWRtg: potencia nominal en vatios pico STC: 330.00 (Pmax/W)
|-----> DO MaxMduLV: Voltaje del módulo a máxima potencia en STC: 37.70 (Vmpp/V)
|-----> DO MaxMduLA: Corriente del módulo a máxima potencia en STC: 8.76 (Impp/A)
|-----> DO MduOpnCctV: Voltaje del módulo en circuito abierto: 45.90 (Voc/V)
|-----> DO MduLWmpDrt: Corriente del módulo en cortocircuitos: 9.27 (Isc/A)
|-----> DO MduLWmpDrt: % de reducción de la potencia en función de la T°: -0.40 %
|-----> DO MduLWmpDrt: % de reducción de la corriente en función de la T°: 0.06 %
|-----> DO MduLVmpDrt: % de reducción de la tensión en función de la T°: -0.31 %
---->Modo LD_MNGT_LN_DPVA (1): ON
|-----> DO Typ: Tipo de configuración del módulo PV: Planta
|-----> DO MduLCnt: Números de módulos por serie: 11
|-----> DO StrgCnt: Números de series en paralelo por sub-matriz: 1
|-----> DO SubArrCnt: Número de sub-matrices en paralelo por matriz: 3
|-----> DO GrndConn: Tipo de conexión a tierra: Puesta a tierra positiva
|-----> DO ArrWRtg: Potencia nominal del array (Wp): 3633.00 Wp
---->Modo LD_MNGT_LN_DPVC (1): ON
|-----> DO ArrModCtr: Modo seleccionado para el control de la potencia de salida de la matriz: Seguimiento del punto de máxima potencia (MPPT)
---->Modo LD_MNGT_LN_MMDC (1): ON
|-----> DO Watt: Potencia medida en PV: 3086.00 V
|-----> DO Amp: Corriente medida en PV: 0.00 V
|-----> DO Vol: Voltaje medido en PV: 48.00 V

Modo LD_BESS (1): ON
---->Modo LD_BESS_LN_ZBAT(1): ON
|-----> DO BathI: Alarma Batería alta: OFF, q: QUALITY_VALIDITY_GOOD
|-----> DO BathO: Alarma Batería alta: OFF, q: QUALITY_VALIDITY_GOOD
|-----> DO LoBatVal: Valor de alarma de batería baja: 10.00 V
|-----> DO HlBatVal: Valor de alarma de batería alta: 15.00 V
---->Modo LD_BESS_LN_ZBTC(1): ON
|-----> DO ChaV: Voltaje de carga: 12.00 V
|-----> DO ChaA: Corriente de carga: 5.00 V
|-----> DO BatchTyp: Tipo de carga de batería: Voltaje constante

Modo LD_ACGRID (1): ON
---->Modo LD_ACGRID_LN_ZRCT(1): ON
|-----> DO IsoTyp: Tipo de aislamiento: Sin aislamiento, aislamiento de la fuente DC
|-----> DO VRegTyp: Tipo de regulación de voltaje: Regulación de salida: voltaje fijo

Modo LD_PLOAD (1): ON
---->Modo LD_PLOAD_LN_MMDC (1): ON
---->Modo LD_PLOAD_LN_MMDC (1): ON
|-----> DO Watt: Potencia medida en PLOAD: 1006.00 V
|-----> DO Amp: Corriente medida en PLOAD: 48.00 V
|-----> DO Vol: Voltaje medido en PLOAD: 48.00 V

Modo LD_NPLOAD (1): ON
---->Modo LD_NPLOAD_LN_MMDC (1): ON
---->Modo LD_NPLOAD_LN_MMDC (1): ON
|-----> DO Watt: Potencia medida en NPLOAD: 1006.00 V
|-----> DO Amp: Corriente medida en NPLOAD: 48.00 V
|-----> DO Vol: Voltaje medido en PLOAD: 48.00 V

Modo LD_WEATHERSTATION (1): ON
---->Modo LD_WEATHERSTATION_LN_MMET (1): ON
|-----> DO EnvTmp: Temperatura ambiente: 22.00 °C
|-----> DO EnvHum: Humedad: 6.00 %
|-----> DO DctInsol: Irradiancia normal directa: 1266.00 W/m²
|-----> DO HorWdSpd: Velocidad del viento horizontal: 10.00 km/h
|-----> DO EnvPres: Presión barométrica: 956.00 hPa

```

Figura 47: Modelo de datos IEC 61850.

Con estas pruebas, además de ofrecer una herramienta al usuario para ver el estado de la planta, se verifica el correcto funcionamiento de la aplicación y se observa, por ejemplo, como se va actualizando el valor del estado de los interruptores en función de si están abiertos o cerrados.

Se puede observar cómo en la primera iteración de lectura del estado de la planta, se observan todos los interruptores de la planta cerrados Figura 48.

```

---->Modo LD_MNGT_LN_XCBR1 (1): ON
|----> DO Pos: Posición del interruptor XCBR1 (BATERIAS): ON
---->Modo LD_MNGT_LN_XCBR2 (1): ON
|----> DO Pos: Posición del interruptor XCBR2 (RECTIFICADOR): ON
---->Modo LD_MNGT_LN_XCBR3 (1): ON
|----> DO Pos: Posición del interruptor XCBR3 (RED): ON
---->Modo LD_MNGT_LN_XCBR4 (1): ON
|----> DO Pos: Posición del interruptor XCBR4 (CARGAS_NC1): ON
---->Modo LD_MNGT_LN_XCBR5 (1): ON
|----> DO Pos: Posición del interruptor XCBR5 (CARGAS_NC2): ON
---->Modo LD_MNGT_LN_XCBR6 (1): ON
|----> DO Pos: Posición del interruptor XCBR6 (CARGAS_NC3): ON

```

Figura 48: Estado de los interruptores de la planta a ON.

Mientras que, en la siguiente interacción, se puede observar cómo se han cerrado todos los interruptores, y actualizado el modelo de la planta IEC 6180 Figura 49.

```

|----> DO Pos: Posición del interruptor XCBR1 (BATERIAS): OFF
---->Modo LD_MNGT_LN_XCBR2 (1): ON
|----> DO Pos: Posición del interruptor XCBR2 (RECTIFICADOR): OFF
---->Modo LD_MNGT_LN_XCBR3 (1): ON
|----> DO Pos: Posición del interruptor XCBR3 (RED): OFF
---->Modo LD_MNGT_LN_XCBR4 (1): ON
|----> DO Pos: Posición del interruptor XCBR4 (CARGAS_NC1): OFF
---->Modo LD_MNGT_LN_XCBR5 (1): ON
|----> DO Pos: Posición del interruptor XCBR5 (CARGAS_NC2): OFF
---->Modo LD_MNGT_LN_XCBR6 (1): ON
|----> DO Pos: Posición del interruptor XCBR6 (CARGAS_NC3): OFF

```

Figura 49: Estado de los interruptores de la planta a OFF.

Si se observa, las dos primeras iteraciones de lectura coinciden con el estado de los interruptores ordenado por los mensajes GOOSE, el cierre Figura 50 y la apertura Figura 51.

```

GOOSE event:
stNum: 1 sqNum: 0
timeToLive: 1000
timestamp: 1611168960.194
  CERRAR XCBR1
  CERRAR XCBR2
  CERRAR XCBR3
  CERRAR XCBR4
  CERRAR XCBR5
  CERRAR XCBR6
  {2,2,2,2,2,2}

```

Figura 50: Orden GOOSE de cierre de todos los interruptores de la planta.

```

GOOSE event:
stNum: 2 sqNum: 0
timeToLive: 1000
timestamp: 1611168971.202
  ABRIR XCBR1
  ABRIR XCBR2
  ABRIR XCBR3
  ABRIR XCBR4
  ABRIR XCBR5
  ABRIR XCBR6
  {1,1,1,1,1,1}

```

Figura 51: Orden GOOSE de apertura de todos los interruptores de la planta.

5.1.2 PRUEBAS CON CLIENTE IEC 61850 COMERCIAL

Para comprobar la compatibilidad y la interoperabilidad con otros dispositivos a través del estándar IEC 61850, se decide hacer uso de una herramienta comercial ajena al proyecto para ver si puede conectarse con el servidor/suscriptor de la Raspberry Pi y extraer y manipular el modelo de datos IEC 61850 de la misma. Con esta prueba se garantiza el cumplimiento del estándar IEC 61850 y la capacidad de interoperabilidad del sistema.

Como herramienta para verificar el estándar IEC 61850 se hace uso del IEDSCOUT, Figura 52.



Figura 52: Logo OMICROM.

Aunque existe diferentes herramientas para analizar sistemas IEC 61850 [34], el IEDSCOUT es una de las herramientas más potentes actualmente en el mercado, es usada por las empresas más grandes involucradas en el desarrollo e implementación del estándar de comunicaciones IEC 61850 en entornos industriales.

El IEDSCOUT es una herramienta desarrollada por OMICRON y permite a los usuarios trabajar con dispositivos IEC 61850. Además, permite analizar internamente IEDs IEC 61850 y sus comunicaciones. Todos los datos modelados e intercambiados quedan visibles y accesibles. Además, IEDScout realiza muchas tareas útiles, que de otro modo solo podrían ser realizadas por herramientas de ingeniería específicas o incluso una estación maestra funcional. IEDScout muestra una visión global que representa el flujo de trabajo típico de la puesta en servicio, pero también proporciona información detallada cuando se solicita [35].

Para poder comprobar la interoperabilidad del sistema IEC 61850 desarrollado en este trabajo, se conecta la Raspberry Pi vía Ethernet al ordenador y se pone en el rango (Figura 53) de la Raspberry Pi en el ordenador.

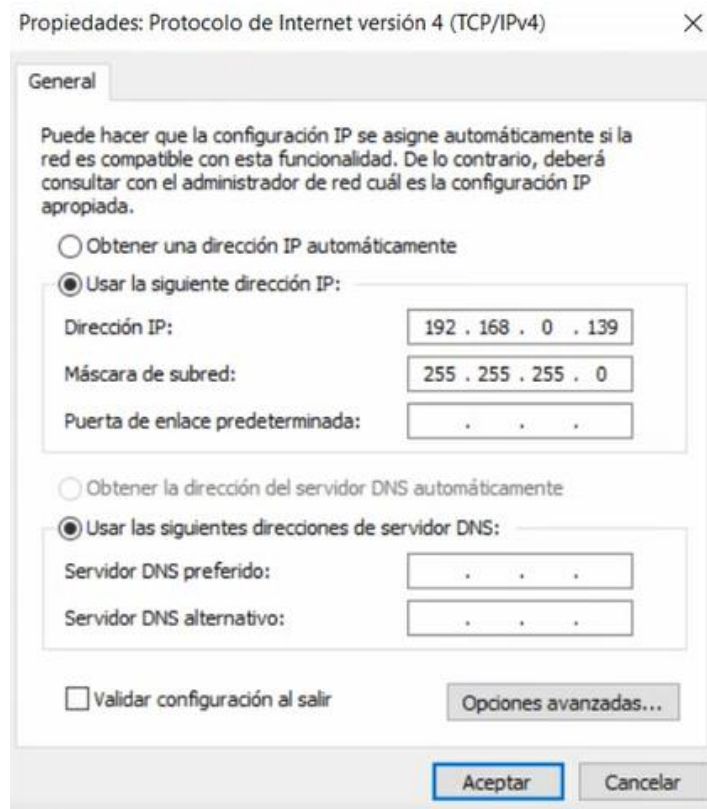


Figura 53: Configuración IP ordenador.

Posteriormente, a través de la aplicación se realiza una búsqueda del IED por IP (Figura 54). Este tipo de programas comerciales orientados como herramientas para trabajar con sistemas IEC 61850 permiten detectar equipos conectados al sistema IEC 61850 una vez conocida su dirección IP. De este modo al introducir la dirección IP del dispositivo IED que se desea detectar, el programa realiza una búsqueda del mismo y en el caso de detectarlo realiza un escaneo del modelado de datos IEC 61850 que este presenta. Esta misma funcionalidad se ha implementado en este trabajo y ha sido descrita en el apartado 4.1.5, donde el sistema desarrollado en este trabajo detecta los LD presentes en el modelo IEC 61850 implementado en el equipo a comunicar a través de la IP definida del mismo.

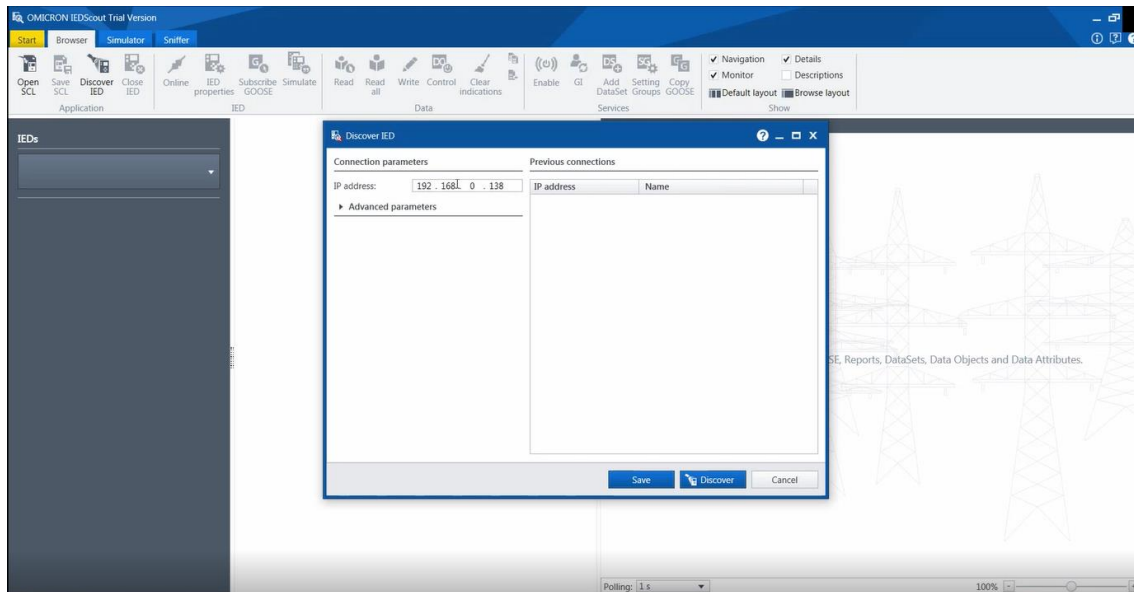


Figura 54: Detección automática de Raspberry Pi con IEDSCOUT.

Posteriormente, el IEDSCOUT detecta de forma automática el IED Raspberry Pi, y extrae su modelo de datos IEC 61850. En las Figura 55, Figura 56, Figura 57, Figura 58, se observa como el IEDSCOUT descompone el modelo IEC 61850 completo implementado en la Raspberry Pi. Es capaz de obtener desde los LD definidos en el equipo hasta los DA de cada DO implementado en el mismo. A través de este paso, no únicamente se consigue detectar el dispositivo IEC 61850 implementado en este trabajo por un sistema comercial, sino que se consigue validar el modelo IEC 61850 definido para el mismo debido a que un sistema externo desarrollado por otro fabricante es capaz de extraer toda la información IEC 61850 definida internamente en el dispositivo IEC 61850 desarrollado en este trabajo.

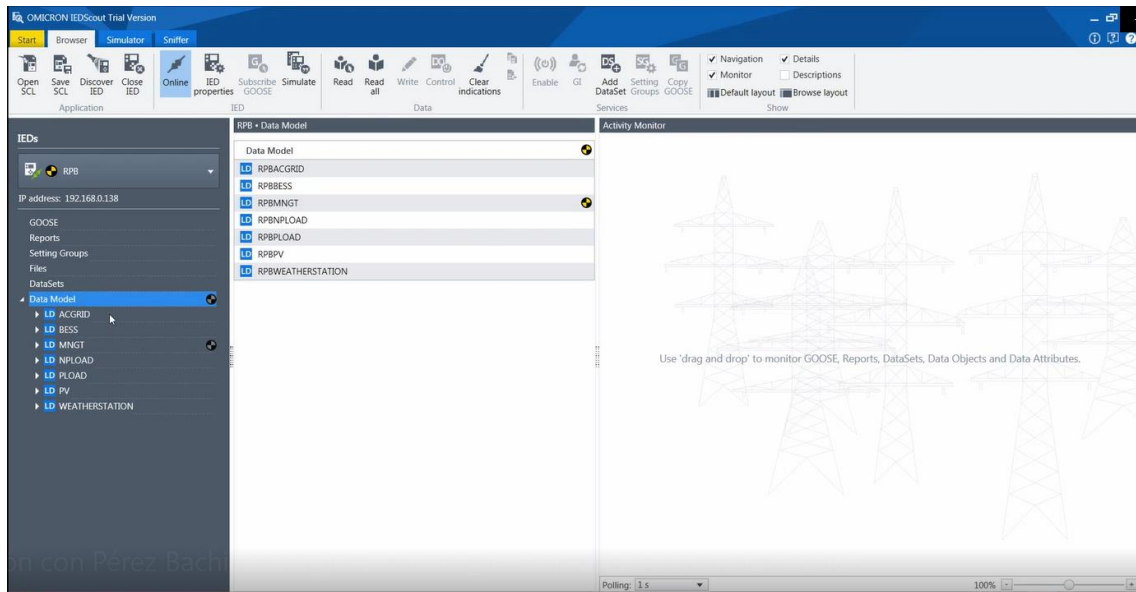


Figura 55: Detección de LD de la Raspberry Pi a través del IEDSCOUT.

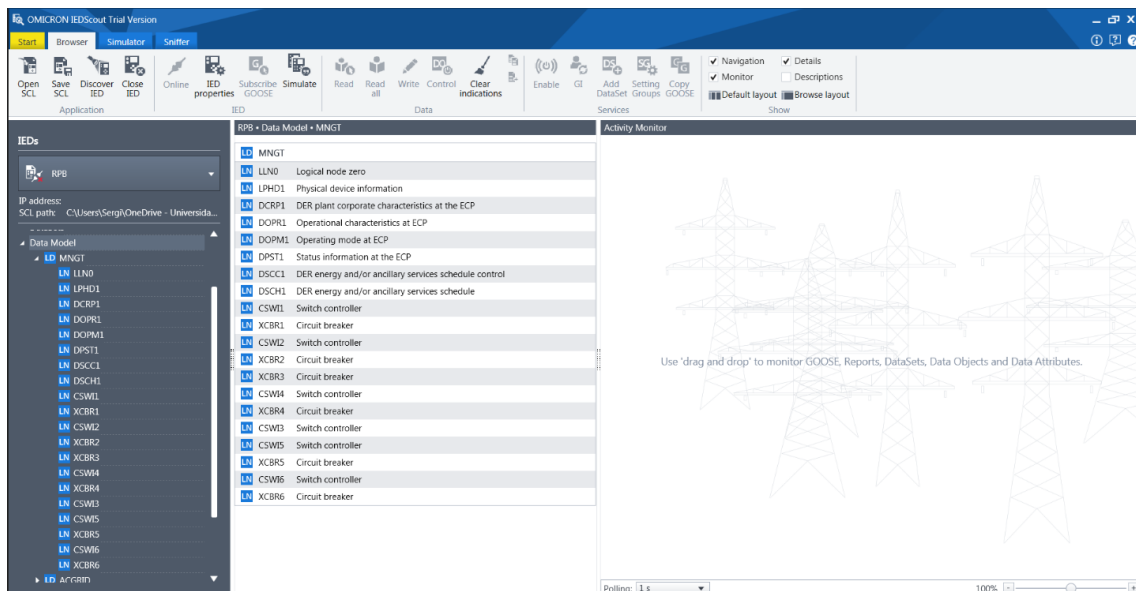


Figura 56: Detección de LN de la Raspberry Pi a través del IEDSCOUT.

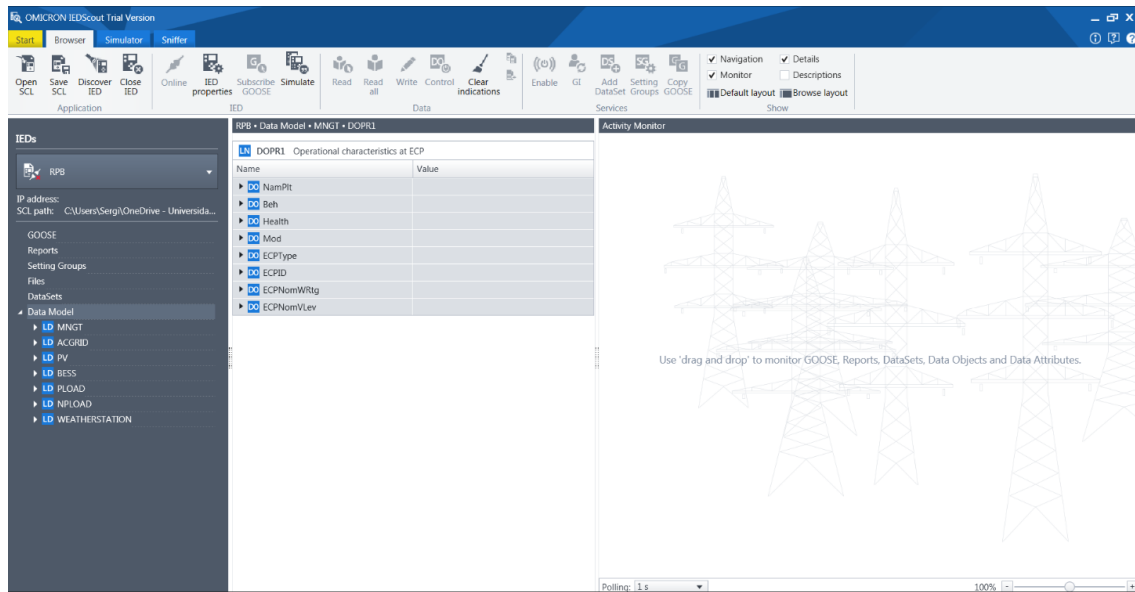


Figura 57: Detección de DO de la Raspberry Pi a través del IEDSCOUT.

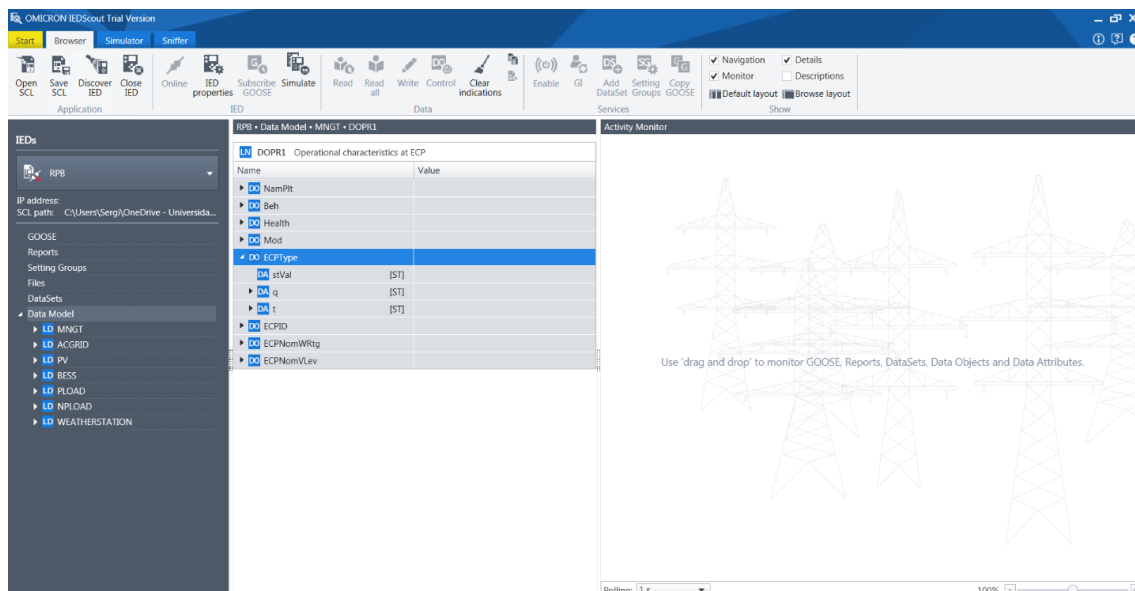


Figura 58: Detección de DA de la Raspberry Pi a través del IEDSCOUT.

Además de ofrecerte acceso al modelo de datos IEC 61850, si se está conectado online con el IED (Figura 59), es decir se tiene una conexión directa entre el dispositivo donde que se ha detectado, el programa IEDSCOUT permite que se ejecuta el programa IEDSCOUT permite habilitar la conexión “Online” y obtener el valor en tiempo real de cada DO presente en el modelo IEC 61850 del equipo detectado. Esta funcionalidad es muy útil ya que te permite ver tanto los valores de cada DO en tiempo real como su calidad, lo cual ante un incidente o un mal funcionamiento del sistema o de la recepción e un valore, te puede permitir analizar y detectar el origen del problema.

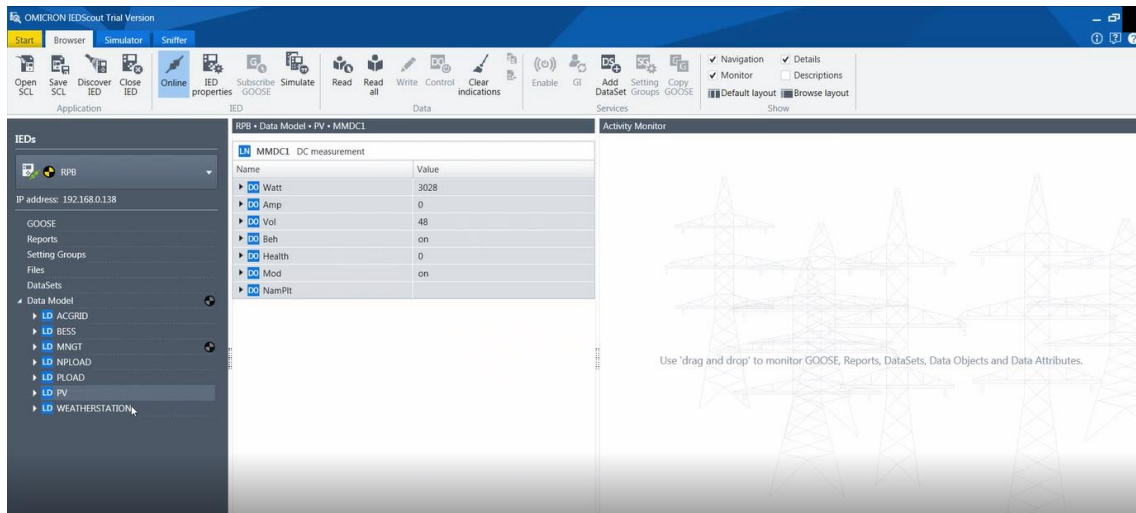


Figura 59: Conexión Online del IEDSCOUT con la Raspberry Pi.

Otra de las utilidades de esta aplicación es la capacidad de simular valores y modificar valores del modelo de datos real ubicado en la Raspberry Pi. En este caso se realiza una simulación modificando el valor de la tensión continua de 48V a 46V. Para ello se debe ir a la pestaña *Simulator* del IEDSCOUT (Figura 60).

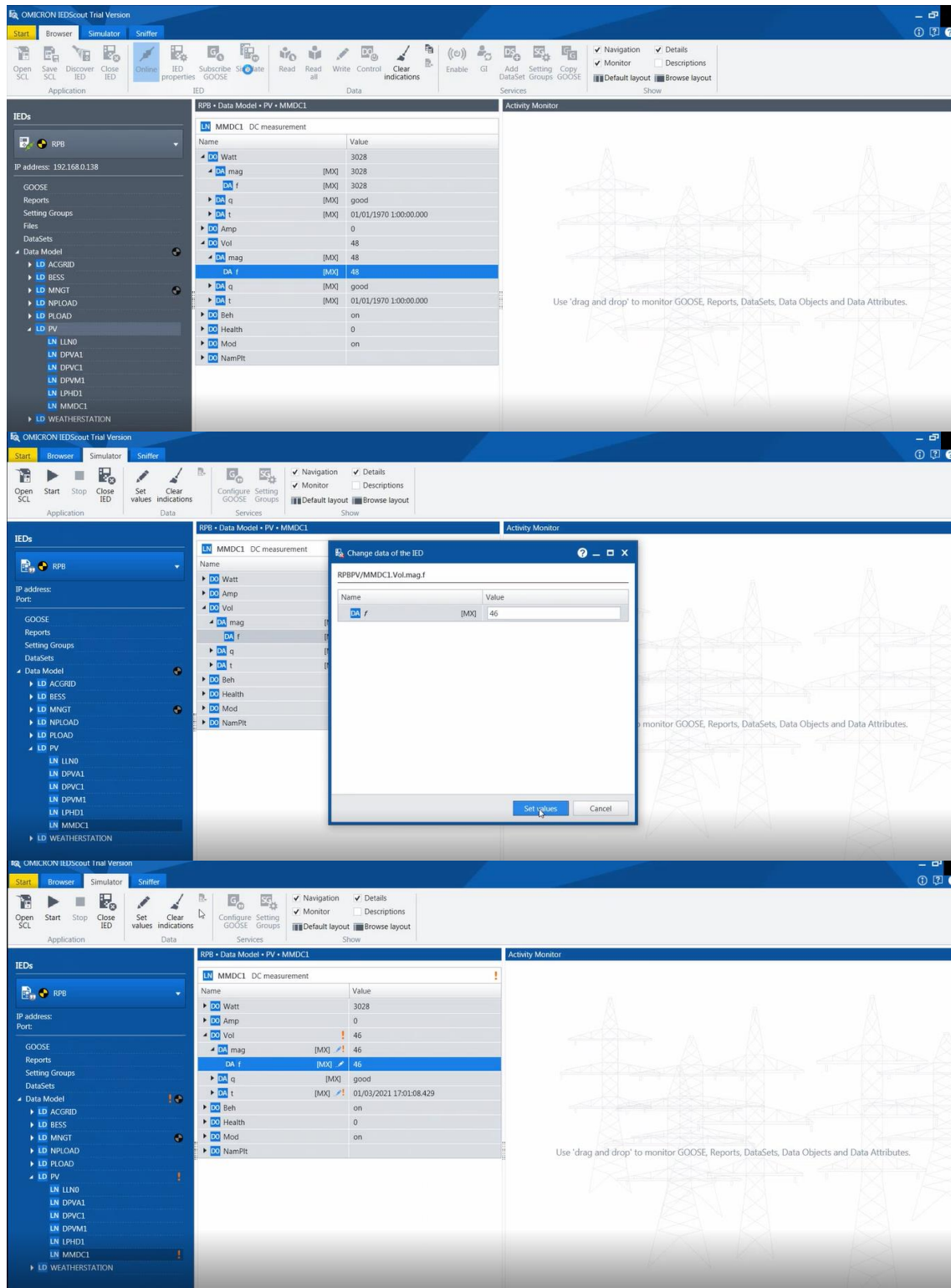


Figura 60: Valores en tiempo real de DA de la Raspberry Pi a través del IED SCOUT.

Como se observa en la Figura 60, la aplicación añade una exclamación para advertir que se ha forzado el valor de dicho objeto. De esta forma permite al usuario ver de una forma rápida los valores que han sido forzados y los valores reales que no han sido forzados.

Con estas pruebas se verifica la interoperabilidad del sistema gracias al desarrollo de comunicaciones basadas en el estándar de comunicaciones IEC 61850. Esto es así ya que se consigue acceder al sistema IEC 61850 a través de otro cliente y otra herramienta desarrollada por un fabricante ajeno al proyecto.

5.1.3 MEDIDAS DE TIEMPOS SOBRE REQUISITOS GOOSE CON NTP.

Una vez analizado el funcionamiento del sistema de comunicaciones desarrollado en este trabajo, junto con su compatibilidad, se procede a realizar un último análisis relacionado con los requisitos de tiempos heredados de la norma IEC 61850 y definidos por las necesidades de la planta.

Para ello, se realiza el análisis de los tiempos que emplea el sistema para la mensajería GOOSE ya que es la más exigente del proyecto. En el apartado 4.1.3 se define unos requisitos de comunicaciones donde los GOOSE del sistema deben ser transmitidos en menos de 20 ms, mensajes tipo 1B/P3.

Para ello se monitorizan los tiempos que emplea el sistema desarrollado en este trabajo para transmitir los mensajes GOOSE. Los GOOSE analizados son los propios del sistema desarrollado, es decir, se trata de un único GOOSE que envía un único *dataset*, el cual contiene 6 elementos tipo enteros de 8 bits, uno por cada estado de los interruptores presentes en la planta. Sin embargo, aunque el *dataset* real que se va a utilizar en la planta consta de 6 elementos, para las pruebas de medición de tiempos se va a incluir un elemento más (Figura 61), que será la marca de tiempo de cuándo ha sido enviado el GOOSE desde el publicador. Con el objetivo de tener mayor precisión se utiliza un elemento entero de 32 bits (int32). Se puede observar cómo las pruebas son incluso más restrictivas que el sistema real, dado que el *dataset* a transmitir es mayor.

```

dataSetValues = LinkedList_create();
LinkedList_add(dataSetValues, MmsValue_newIntegerFromInt8(1)); //XCBR1: ON=2, OFF=1, INTERMEDIATE=0, BAD_STATE=3
LinkedList_add(dataSetValues, MmsValue_newIntegerFromInt8(1)); //XCBR2: ON=2, OFF=1, INTERMEDIATE=0, BAD_STATE=3
LinkedList_add(dataSetValues, MmsValue_newIntegerFromInt8(1)); //XCBR3: ON=2, OFF=1, INTERMEDIATE=0, BAD_STATE=3
LinkedList_add(dataSetValues, MmsValue_newIntegerFromInt8(1)); //XCBR4: ON=2, OFF=1, INTERMEDIATE=0, BAD_STATE=3
LinkedList_add(dataSetValues, MmsValue_newIntegerFromInt8(1)); //XCBR5: ON=2, OFF=1, INTERMEDIATE=0, BAD_STATE=3
LinkedList_add(dataSetValues, MmsValue_newIntegerFromInt8(1)); //XCBR6: ON=2, OFF=1, INTERMEDIATE=0, BAD_STATE=3
gettimeofday(&tiempo_envio, NULL);
tiempo_envio_aux = tiempo_envio.tv_usec;
LinkedList_add(dataSetValues, MmsValue_newIntegerFromInt32(tiempo_envio_aux));

```

Figura 61: Introducción de la marca de tiempo en el dataset del GOOSE.

De esta forma, la medición consiste en, una vez recibido el GOOSE en el suscriptor, se obtiene la marca de tiempo de llegada, y se compara con la marca de tiempo recibida a través del dataset del GOOSE. De esta forma se puede obtener los tiempos que emplean los mensajes GOOSE en el sistema aquí presente.

Aunque la red de comunicaciones implicada para este sistema es una red reducida donde solo hay implicado 2 equipos, al ponerse en juego diferentes tipos de mensajes (MMS y GOOSE) que hacen uso de diferentes capas OSI para su transmisión, resulta una red donde los tiempos no son deterministas. Por ello, se considera que se deben medir dichos tiempos ejecutando el sistema en diferentes momentos.

Se realizan 5 ejecuciones independientes del sistema, y en cada ejecución se almacena el tiempo de transmisión de 100 GOOSE para ser analizados posteriormente.

A continuación, se exponen los resultados de los 100 GOOSE de cada una de las 5 ejecuciones realizadas del sistema. En cada gráfica se expone en el eje X el número de GOOSE analizado y en el eje Y el tiempo de transmisión en ms.

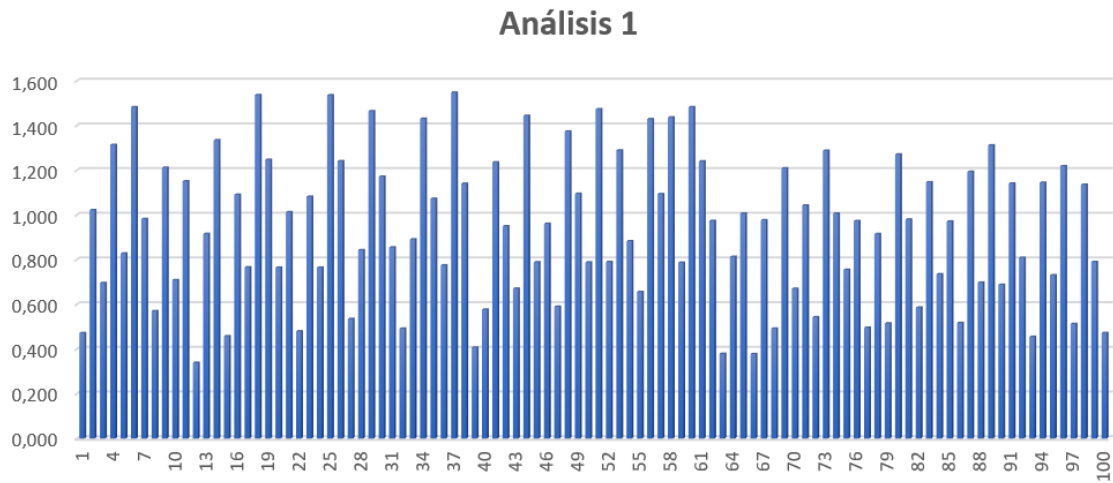


Figura 62: Análisis de tiempos 1.

Valores medio, máximo y mínimo obtenidos en el primer análisis (Figura 62):

- **Valor medio** = 0.936 ms.
- **Valor máximo** = 1.548 ms.
- **Valor mínimo** = 0.339 ms.

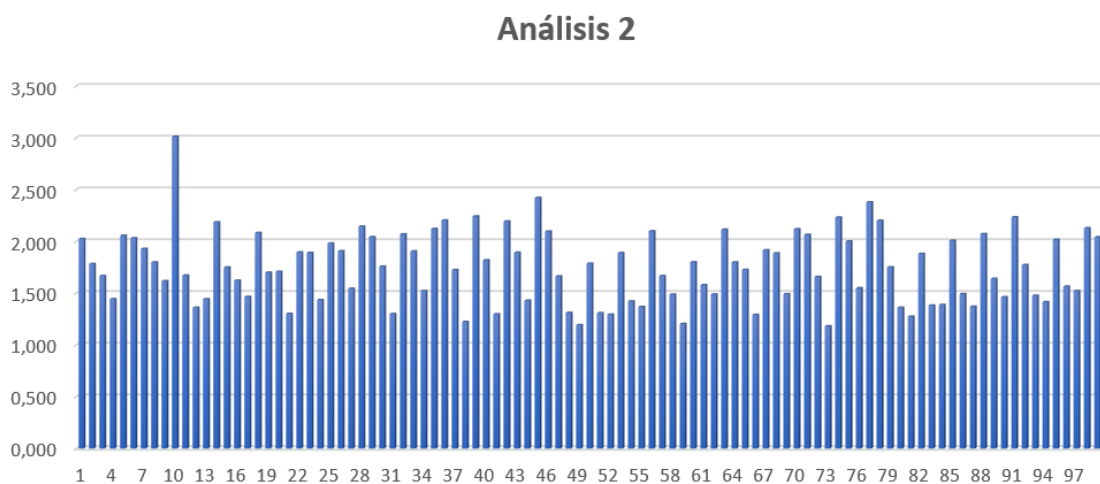


Figura 63: Análisis de tiempos 2.

Valores medio, máximo y mínimo obtenidos en el segundo análisis (Figura 63):

- **Valor medio** = 1.751 ms.
- **Valor máximo** = 3.012 ms.
- **Valor mínimo** = 1.182 ms.

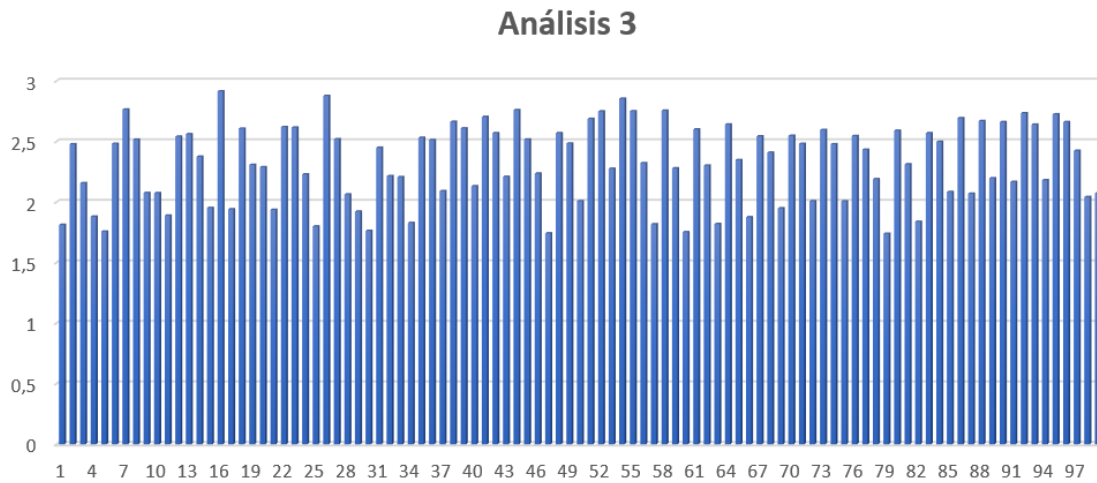


Figura 64: Análisis de tiempos 3.

Valores medio, máximo y mínimo obtenidos en el tercer análisis (Figura 64):

- **Valor medio** = 2.324 ms.
- **Valor máximo** = 2.915 ms.
- **Valor mínimo** = 1.739 ms.

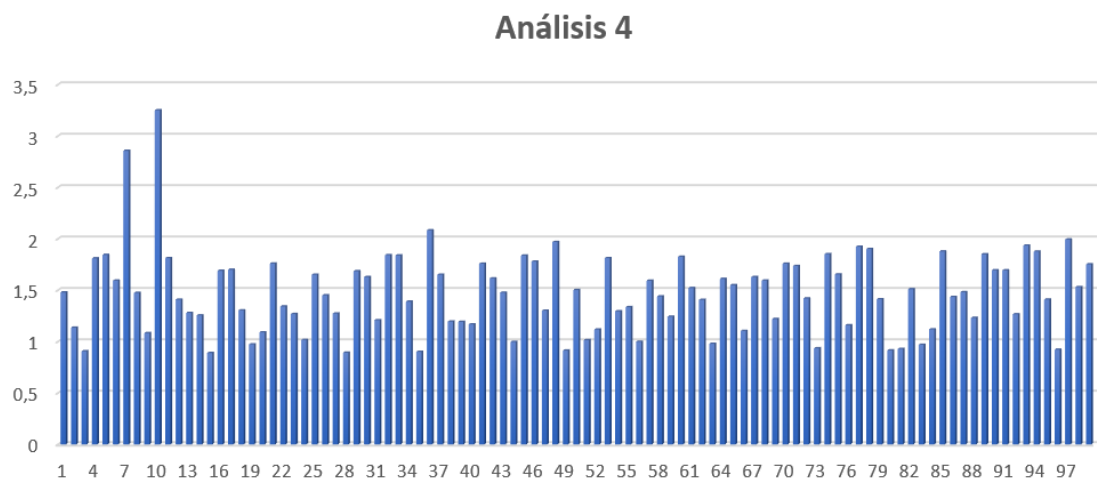


Figura 65: Análisis de tiempos 4.

Valores medio, máximo y mínimo obtenidos en el cuarto análisis (Figura 65):

- **Valor medio** = 1.476 ms.
- **Valor máximo** = 3.250 ms.
- **Valor mínimo** = 0.889 ms.

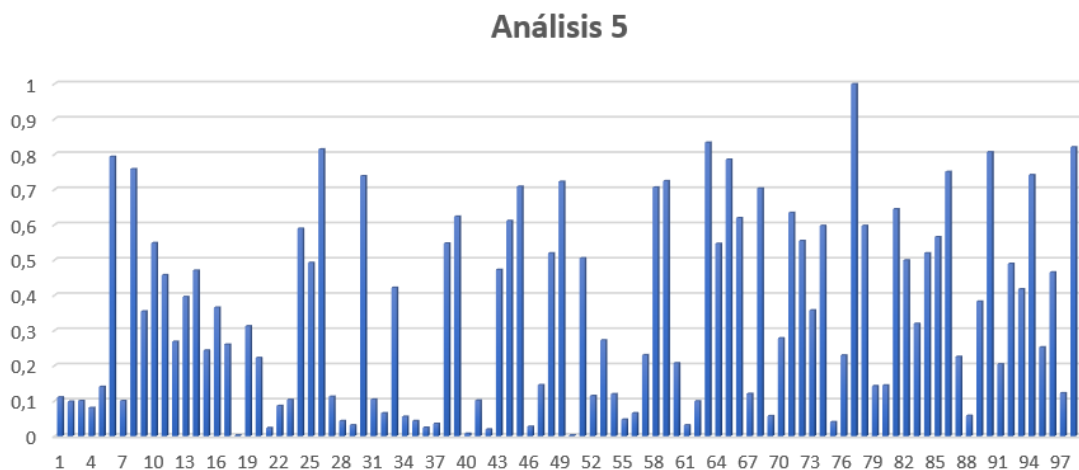


Figura 66: Análisis de tiempos 5.

Valores medio, máximo y mínimo obtenidos en el quinto análisis (Figura 66):

- **Valor medio** = 0.342 ms.
- **Valor máximo** = 0.999 ms.
- **Valor mínimo** = 0.002 ms.

Observando los resultados expuestos de los cinco casos, podemos obtener diferentes conclusiones. En primer lugar se observa que los valores de transmisión son muy bajos, y las medias se encuentran alrededor de 1.370 ms. En el peor de los casos en los análisis realizados, el tiempo de transmisión para la mensajería GOOSE máximo ha sido de 3.250 ms, muy por debajo de los 20 ms que se definen como requisitos de tiempos para esta aplicación (Tipo 1B/P3). Los tiempos tan bajos presentes en este sistema dependen de varios factores, el primero es que la transmisión se ha realizado a través de mensajes GOOSE diseñados para priorizar la velocidad de transmisión. Además, el sistema presente en este trabajo presenta una reducida carga en el canal de transmisión debido a que solo existen dos dispositivos. Por otro lado, en este sistema los equipos se encuentran muy cercanos (reducida distancia del cable de transmisión) y se transmite la información de punto a punto si introducir retardos como por ejemplo el de los equipos *switch* típicamente utilizados en estos sistemas cuando aparecen más IEDs.

Aunque los tiempos conseguidos en el sistema sean muy reducidos y pudiesen cumplir requisitos más exigentes que los tipo 1B/P3 (en el apartado 2.1.9 se explican los diferentes requerimientos de tiempos), como se comenta en el apartado

4.1.3, para este tipo de sistemas no es justificable elegir requerimientos de tiempos más exigentes dado que en esta implementación los GOOSE no se utilizan para el disparo de interruptores en el caso de necesitar proteger cualquier elemento de la planta ante la aparición de una incidencia en el sistema. En este sistema los mensajes GOOSE se utilizan para modificar el estado de los interruptores en función de cómo se desee explotar la planta en cada momento atendiendo a los algoritmos de control desarrollados en el proyecto Helios. Por otro lado, se debe tener en cuenta que es preferible cumplir de manera holgada con los requisitos de comunicaciones debido a que el sistema desarrollado en este trabajo presenta la posibilidad de ser extrapolado e implementado en la gestión de diferentes plantas DER dentro de una VPP.

CAPÍTULO 6: CONCLUSIONES Y TRABAJOS FUTUROS

En el TFM se han conseguido alcanzar los objetivos planteados, expuestos en el apartado 1.1.2. Como objetivo principal, se ha desarrollado un sistema de comunicaciones para permitir la interoperabilidad entre dispositivos en una MG. A su vez, también permite la interoperabilidad del conjunto de la MG frente a otras MG. Para ello, este trabajo ha implementado un sistema de comunicaciones entre dispositivos electrónicos inteligentes basado en el estándar de comunicaciones IEC 61850.

Para la implementación del estándar de comunicaciones IEC 61850 se han modelado y digitalizado los elementos presentes en la planta experimental del proyecto HELIOS. También se han analizado los requisitos de comunicación presentes en la norma IEC 61850 y, posteriormente, se han definido para el caso particular presente en este proyecto. Por último, se han analizado los diferentes métodos de mensajería presentes en la norma y se han seleccionado los más indicados, en función de los requerimientos de comunicación.

Una vez analizado el sistema y modelado bajo el estándar de comunicaciones IEC 61850, se ha desarrollado un cliente/publicador y un servidor/suscriptor empleando la librería “libiec61850” para desarrollar los intercambios de comunicación necesarios.

Una vez realizada la implementación, se han realizado las pruebas de conformidad donde se han verificado el cumplimiento de los requisitos necesarios para el correcto funcionamiento de la planta, los requisitos impuestos para el cumplimiento de la norma IEC 61850, y, por último, la correcta interoperabilidad entre el sistema desarrollado en este trabajo y sistemas comerciales basados en la norma IEC 61850.

Como trabajos futuros se destacan la posibilidad de escalar el diseño realizado, para modelar e interconectar diferentes plantas DER. Por otro lado, quedaría abierta la implementación de un SCADA local que facilitase al usuario la manipulación de la planta.

También quedaría abordar e implementar uno de los aspectos más preocupantes hoy en día en el desarrollo de sistemas digitales, la ciberseguridad. Para ello, se

debería analizar e implementar los requisitos y las recomendaciones presentes en la norma IEC 62351.

Por último, otro de los trabajos futuros sería el desarrollo y la propuesta para el uso de sistemas de intercambio de energía a través de tecnología *Blockchain* estandarizado bajo la norma IEC 61850 [36].

CAPÍTULO 7: PRESUPUESTO

En este apartado se van a exponer todas las partidas presupuestarias necesarias para la realización del proyecto. En él, se desglosan tanto los costes del material utilizado, como los costes directos del sistema, los cuales vienen dados por las horas empleadas por el desarrollador, en este caso el propio alumno encargado de realizar este TFM, además de los gastos indirectos del sistema y los honorarios facultativos.

7.1.1 Coste del material utilizado

En la Tabla 10 se adjunta el desglose de los costes asociados a cada uno de los elementos físicos necesarios para la realización del proyecto:

Material	Precio Unitario (€)	Unidades	Periodo de amortización (años)	Uso (meses)	Coste de amortización (€)
Raspberr y Pi 3 b+	44,89 €	1	2	9	16,84 €
ESP-32	9,49 €	1	2	9	3,56 €
S.O. "Ubuntu"	0	1	3	9	0
S.O. "Raspber ry Pi OS"	0	1	3	9	0
PC ASUS i7- 4710HQ	1150 €	1	3	9	287,5 €
Total	1204,38 €				307,90 €

Tabla 10: Costes del material utilizado.

7.1.2 Costes directos del sistema

Como coste directo debido al tiempo invertido tanto por el desarrollador, como por el escritor para la realización de la aplicación.

A continuación, se observa un desglose de los precios y los tiempos invertidos para la realización del sistema. Se considera que se ha trabajado 37,5 horas semanales y que un mes de media tiene 22 días laborables. De esta forma, para un total de seis meses de trabajo, el coste del personal se expone en la Tabla 11.

Actividad	Coste (€/h)	Tiempo (Horas)	Coste Total (€)
Graduado en ingeniería industrial	11,40€	90	11.286,00 €
Total			11.286,00 €

Tabla 11: Costes personal.

7.1.3 Costes indirectos del sistema

Los costes indirectos del sistema vienen dados por las licencias de los programas necesarios para el desarrollo del proyecto. En la Tabla 12 se observa un desglose de estos.

Licencia	Coste (€)	Periodo de amortización (años)	Uso (meses)	Coste de amortización (€)
Matlab r2019b	2000 €	2	7	583,30 €
S.O. "Ubuntu"	0	3	9	0
S.O. "Raspberry Pi OS"	0	3	9	0
IDE GEANY	0	3	9	0
Microsoft Office 365	450	2	7	131,25 €
Total	2450 €			714.55 €

Tabla 12: Costes indirectos del sistema.

7.1.4 Honorarios Facultativos

Según el Colegio Oficial de Ingenieros Industriales de Madrid, estos gastos se establecen como el 0,255 % sobre el total (costes de material, costes directos y costes indirectos del proyecto), debido a que el coste del proyecto no sobrepasa los 30.050 €.

Concepto	Coste íntegro del proyecto (€)	Coste de amortización (€)
Coste del material utilizado	1204,38 €	307,90 €
Costes directos del sistema	11.286,00 €	11.286,00 €
Costes indirectos del sistema	2450 €	714,55 €
Total	19568,88 €	12.308,45 €
Coste por honorarios facultativos		31,39 €

Tabla 13: Coste por honorarios facultativos.

7.1.5 Costes del proyecto

Por lo tanto, en la Tabla 14 se muestra un resumen de todos los gastos asociados al proyecto, y el coste total de la aplicación.

Concepto	Coste íntegro del proyecto (€)	Coste de amortización (€)
Coste del material utilizado	1204,38 €	307,90 €
Costes directos de la aplicación	11.286,00 €	11.286,00 €
Costes indirectos de la aplicación	2450 €	714,55 €
Coste por honorarios facultativos	31,39 €	31,39 €
Total (sin IVA)	14.971,77 €	12.339,84 €
Total (IVA incluido 21%)	18.115,84 €	14.931,21 €

Tabla 14: Coste del proyecto.

Por lo tanto, el importe total estimado del proyecto presente en este trabajo asciende a la cantidad de:

Catorce mil novecientos treinta y un euros con veintiún céntimos.

Guadalajara, a 14 de septiembre de 2021.

Firmado: Sergio Pérez Bachiller.

Ingeniero Industrial.

CAPÍTULO 8: BIBLIOGRAFÍA

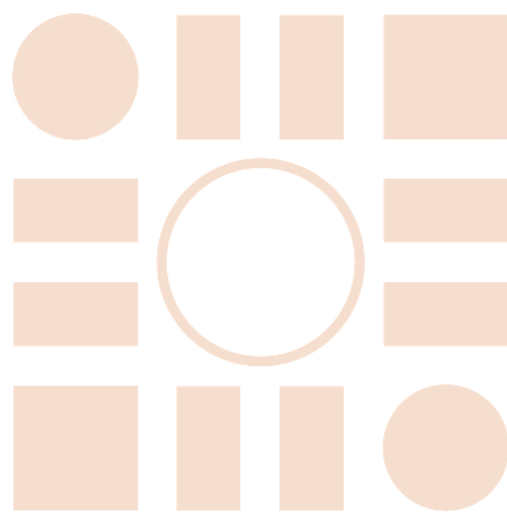
Bibliografía

- [1] ArturoAlarcon-Rodriguez, GrahamAult y StuartGalloway, «Multi-objective planning of distributed energy resources: A review of the state-of-the-art,» *ELSEVIER*, 2010.
- [2] «Solar Power Europe,» 2019. [En línea]. Available: <https://www.solarpowereurope.org/eu-market-outlook-for-solar-power-2019-2023>.
- [3] «IEC International Electrotechnical Commission, "Communication networks and systems for power utility automation - Part 7-420: Basic communication structure - Distributed energy resources logical nodes",» 2009.
- [4] Miteco, 2020. [En línea]. Available: <https://www.miteco.gob.es/es/cambio-climatico/temas/mitigacion-politicas-y-medidas/objetivos.aspx>.
- [5] S. Marzal, R. Salas, R. González-Medina, G. Garcerá y E. Figueres, «Current challenges and future trends in the field of communication architectures for microgrids,» *ELSEVIER*, 2018.
- [6] SuhailHussain y IkbaliS.M., «Control and management of distribution system with integrated DERs via IEC 61850 based communication,» *ELSEVIER*, 2017.
- [7] F. Knirsch, A. Unterweger y D. Engel, «Privacy-perserving blockchain-based electric vehicle charging with dynamic tariff decision,» de *Computer Science - Research and Development*, vol. 33, p. 71-79, 9, 2017.
- [8] Salazar-Zuluaga, «A. Salazar-Zuluaga, L.A. Aguilar-Londoño, G.D. Zapata-Madrigal, y r. García-Sierra, "implementación de nodos lógicos der IEC 61850-7-420 en una placa electrónica", *tecnológicas*, vol. 20, no 40, pp. 13-28, 2017.,» 2017.
- [9] L. Zhu, D. Shi y P. Wang, «"IEC 61850-based information model and configuration description of communication network in substation automation," *Power deliv. iee trans.*, vol. 29, no. 1, pp. 97-107, 2014.,» 2014.

-
- [10] J. Schmutzler, C. Wietfeld y C. A. Andersen, «“distributed energy resource management for electric vehicles using IEC 61850 and ISO/IEC 15118,” Veh. power propuls. conf. (vppc), 2012 ieee, pp. 1457–1462, 2012.,» 2012.
- [11] K. Dielmann y A. v. d. Velden, «Virtual power plants (VPP) - a new perspective for energy generation?,» *IEEE*, 2003.
- [12] Taha Selim Ustun, M. A. Aftab, I. Ali y S. M. S. Hussain, «A Novel Scheme for Performance Evaluation of an IEC 61850-Based Active Distribution System Substation,» *IEEE*, 2019.
- [13] A. F. Raab, M. Ferdowsi, E. Karfopoulos, I. G. Unda, S. Skarvelis-Kazakos, P. Papadopoulos, E. Abbasi y L. Cipcigan, «Virtual Power Plant Control concepts with Electric Vehicles,» *IEEE*, 2011.
- [14] M. R.E., «Overview of IEC 61850 and benefits,» 2006.
- [15] F. A. Acevedo-Cardozo, L. C. Calderón-Soto, J. C. González-Castillo y H. R. Vargas-Torres, «Comunicación de sistemas eléctricos basados en la Norma IEC 61850: Desarrollo de caso usando sampled values en servicios Cliente-Servidor,» *Revista UIS Ingenierías*, 2019.
- [16] «IEC International Electrotechnical Commission, "Communication networks and systems for power utility automation - Part 7-1: Basic communication structure - Principles and Models",» 2011.
- [17] «IEC International Electrotechnical Commission, "Communication networks and systems for power utility automation - Part 5: Communication requirements for functions and device models",» 2013.
- [18] J.-C. Gu, C.-H. Liu, J.-M. Wang y M.-T. Yang, «Using IEC 61850 GOOSE messages in microgrid protection,» de *ELECTRICAL ENERGY SYSTEMS*, 2019.
- [19] «IEC International Electrotechnical Commission, "Communication networks and systems for power utility automation - Part 7-2: Basic information and communication structure - Abstract communication service interface (ACSI)",» 2010.
- [20] «IEC International Electrotechnical Commission, "Communication networks and systems for power utility automation - Part 7-4: Basic communication

- structure -Compatible logical node classes and data object classes",» 2010. [En línea].
- [21] «TRIANGLE MICROWORKS,» 2020. [En línea]. Available: <https://www.trianglemicroworks.com/home>.
- [22] Y. Yang, K. McLaughlin, L. Gao, S. Sezer, Y. Yuan y Y. Gong, «Intrusion Detection System for IEC 61850 based Smart Substations,» IEEE, 2016.
- [23] Kalam, S. M. S. Hussain y Akhtar, «A Review of IEC 62351 Security Mechanisms for IEC 61850 Message Exchanges,» *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, vol. 16, nº 9, 2020.
- [24] A. Elgargouri y M. Elmusrati, «Analysis of Cyber-Attacks on IEC 61850 Networks,» IEEE, 2017.
- [25] «LIBIEC61859,» 2020. [En línea]. Available: <https://libiec61850.com/libiec61850/about/>.
- [26] G. Moreno, C. Santos, P. Martín, F. J. Rodríguez, R. Peña y B. Vuksanovic, «Intra-Day Solar Power Forecasting Strategy for Managing Virtual Power Plants,» *sensors*, vol. 21, 2021.
- [27] G. Moreno, P. Martin, C. Santos, F. J. Rodríguez y E. Santiso, «A Day-Ahead Irradiance Forecasting Strategy for the Integration of Photovoltaic Systems in Virtual Power Plants,» *IEEE*, vol. 8, 2021.
- [28] M. Tradacete, C. Santos, J. A. Jiménez, F. J. Rodríguez, P. Martín, E. Santiso y M. Gayo, «Turning Base Transceiver Stations into Scalable and Controllable DC Microgrids Based on a Smart Sensing Strategy,» *sensors*, 2021.
- [29] A. Huawei. [En línea]. Available: https://support.huawei.com/view/PdfRead/EDOC1000112660/SUPE_DOC/6001/document.pdf.
- [30] «PSU4850A,» [En línea]. Available: http://www.audiowerk.at/downloads/temp/hannes/PSU4850A-Rectifier-User-Manual-V1_0.pdf.
- [31] Zillgith y Michael, «libIEC61850,» 2021. [En línea]. Available: <https://libiec61850.com/libiec61850/>.

-
- [32] Zillgith y Michael, «OpenIEC61850,» 2021. [En línea]. Available: <https://www.openmuc.org/iec-61850/>.
- [33] MEINBERG. [En línea]. Available: <https://www.meinberg.es/soporte/informacion/network-time-protocol.htm>.
- [34] Sánchez y H. V. Alegría, PROPUESTA DE HERRAMIENTAS PARA LA VERIFICACIÓN DE LA OPERACIÓN, 2019.
- [35] IEDSCOUT. [En línea]. Available: <https://www.omiconenergy.com/es/productos/iedscout/>.
- [36] M. Gayo, C. Santos, F. J. R. Sánchez, P. Martín, J. A. Jiménez y M. Tradacete, «Addressing Challenges in Prosumer-Based Microgrids with Blockchain and an IEC 61850-Based Communication Scheme,» *IEEE Access*, vol. 8, 2020.
- [37] Acciona, 2019. [En línea]. Available: <https://www.imnovation-hub.com/es/energia/que-es-una-virtual-power-plant/>, último acceso 29 de noviembre 2019.
- [38] «IEC International Electrotechnical Commission, "Communication networks and systems for power utility automation - Part 8-1: Specific communication service mapping (SCSM) - Mapping to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3,» 2011.



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá