UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

GRADUADO EN INGENIERÍA DEL SOFTWARE

Protección ante amenazas y vulnerabilidades en Microgrids con conexión a infraestructuras basadas en puntos de carga

Protection against threads and vulnerabilities in Microgrids with connection to charging points-based infrastructures

Realizado por
Jesús Cumplido Almenara

Tutorizado por
María Cristina Alcaraz Tello
Francisco Javier López Muñoz

Departamento
LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO DE 2021

UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADUADO EN INGENIERÍA DEL SOFTWARE

Protección ante amenazas y vulnerabilidades en Microgrids con conexión a infraestructuras basadas en puntos de carga

Protection against threads and vulnerabilities in Microgrids with connection to charging points-based infrastructures

Realizado por
Jesús Cumplido Almenara

Tutorizado por
María Cristina Alcaraz Tello
Francisco Javier López Muñoz

Departamento
Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, JUNIO DE 2021

Fecha defensa: julio de 2021

Me gustaría transmitir mi más sincero agradecimiento a todos aquellos que me han ayudado a lo largo de esta etapa académica y formativa.

En primer lugar, agradecer a mis tutores, Cristina y Javier, por darme la oportunidad de trabajar con ellos y confiar en mí desde un principio.

También agradecer a mi familia, incluyendo mascotas, que han sido los principales promotores de mi formación académica y, sobre todo, educación. Gracias a todo vuestro apoyo y esfuerzo he logrado llegar hasta aquí.

Por último, quería hacer una especial mención y dedicar este trabajo a aquellas personas que me han impulsado en el mundo de las TIC desde que era pequeño. En especial, a mi antiguo vecino y amigo que me enseñó las primeras nociones de informática. Estés donde estés, Miguelito, gracias por todo.

Abstract

We are increasingly aware about the need and concern for the use of renewable energies. Energy and transportation sectors are two of the main contributors to gas emissions and pollution. In addition, electricity sector is also one of the most vulnerable industrial sectors in terms of safety, causing severe blackouts that affect a large part of society.

This project aims to cover these lines of needs. It studies the architecture and design of a charging points-based Microgrid, called *Secure Charging Points (SecCP)*. First, the most common vulnerabilities and threats are classified in this scenario in order to subsequently increase its cybersecurity using different support technologies.

The objective of this project is to create a control platform capable of securely monitoring simulated energy transactions provided by different charging stations and electric vehicle models. Moreover, for monitoring, the specific design of security services related to situational awareness, traceability, auditing and accountability is taken into account. For situational awareness, anomaly detection mechanisms based on learning techniques are used. On the other hand, to ensure proper traceability and auditability, a new technology and supporting infrastructure is added, such as a DLT (Distributed Ledger Technology), specifically a permissioned Blockchain network.

These technologies assist the system in adding a first line of defense, as recommended by the National Institute of Standards and Technology (NIST) in its cybersecurity framework for the critical infrastructure protection, in which is very critical to identify and detect anomalies and/or cyberattacks.

Keywords: Blockchain, Machine Learning, Anomalies Detection, Situational Awareness

Resumen

Cada día somos más conscientes de la necesidad y la preocupación por el uso de energías renovables. El sector energético y transporte son dos de los principales responsables de las emisiones de gases y contaminación. Además, el sector eléctrico también es uno de los sectores industriales más vulnerable en seguridad, provocando severos apagones que afectan a gran parte de la sociedad.

Este proyecto pretende cubrir estas líneas de necesidades. Se estudia la arquitectura y diseño de una Microgrid basada en puntos de cargas, denominada *Secure Charging Points (SecCP)*. En primer lugar, se clasifican las vulnerabilidades y amenazas más comunes en este escenario para, posteriormente, aumentar su seguridad usando diferentes tecnologías de soporte.

El objetivo de este proyecto es crear una plataforma de control capaz de monitorizar de forma segura las transacciones simuladas de energía proporcionadas por diferentes estaciones de cargas y modelos de vehículos eléctricos. Además, para la monitorización, se tiene en cuenta el diseño específico de servicios de seguridad relacionados con la conciencia situacional, trazabilidad, auditoría y responsabilidad. Para la conciencia situacional, se hace uso de mecanismos de detección de anomalías basadas en técnicas de aprendizaje. Por otro lado, para garantizar una correcta trazabilidad y auditoría, se añade una nueva tecnología e infraestructura de soporte, como es una DLT (Distributed Ledger Technology), específicamente una red Blockchain permisionada.

Estas tecnologías ayudan al sistema a añadir una primera línea de defensa, tal y como recomienda el National Institute of Standards and Technology (NIST) en su framework de ciberseguridad para la protección de infraestructuras críticas, en el que es muy importante identificar y detectar anomalías y/o ciberataques.

Palabras clave: Blockchain, Aprendizaje Automático, Detección de Anomalías, Consciencia Situacional

Índice

Índice de figuras

Índice de tablas

Lista de acrónimos

AEMA Agencia Europea del Medio Ambiente.

APT Advanced Persistent Threat.

ARP Address Resolution Protocol.

BLE Bluetooth Low Energy.

CA Consecuencia de Amenaza.

CEI Comisión Electrotécnica Internacional.

CPS Cyber-Physical System.

DDoS Denegación de Servicio Distribuido.

DER Distributed Energy Resource.

DF Flujo de datos.

DFD Diagrama de Flujo de Datos.

DLT Distributed Ledger Technology.

DNS Domain Name System.

DoS Denegación de Servicio.

DS Almacenamiento de datos.

EE Entidad Externa.

EMS Energy Management System.

ENISA European Union Agency for Cybersecurity.

EVCC Electric Vehicle Charger Controller.

EVSE Electric Vehicle Supply Equipment.

FDI False Data Injection.

GPIO General Purpose Input/Output.

IA Inteligencia Artificial.

ICS Industrial Control System.

IoT Internet of Things.

JSON JavaScript Object Notation.

MitM Man in the Middle.

MQTT Message Queing Telemetry Transport.

NIST National Institue of Standards and Technology.

OCPP Open Charge Point Protocol.

P Proceso.

P2P Peer-to-Peer.

PoA Proof of Authority.

PoW Proof of Work.

R Riesgo.

REST Transferencia de Estado Representacional.

SCADA Supervisory Control And Data Acquisition.

SecCP Secure Charging Points.

TFG Trabajo Fin de Grado.

TIC tecnologías de la información y la comunicación.

TLS Transport Layer Security.

UE Unión Europea.

UMA Universidad de Málaga.

1 Introducción

1.1. Motivación

La Unión Europea (UE) ha propuesto como objetivo, dentro del Pacto Verde Europeo [1], alcanzar la neutralidad climática en 2050 reduciendo las emisiones de gases entre un 80 % y 95 % con respecto las de 1990. Hasta ahora, se ha logrado el objetivo de reducir hasta un 20 % para el año 2020. Sin embargo, la Comisión Europea ha propuesto aumentar el objetivo de reducción de emisiones para 2030 a un mínimo del 55 % [2]. Como se muestra en la figura 1, para alcanzar el siguiente objetivo propuesto por la UE es necesario la inclusión de nuevas medidas que reduzcan drásticamente la proyección de emisiones de gases.

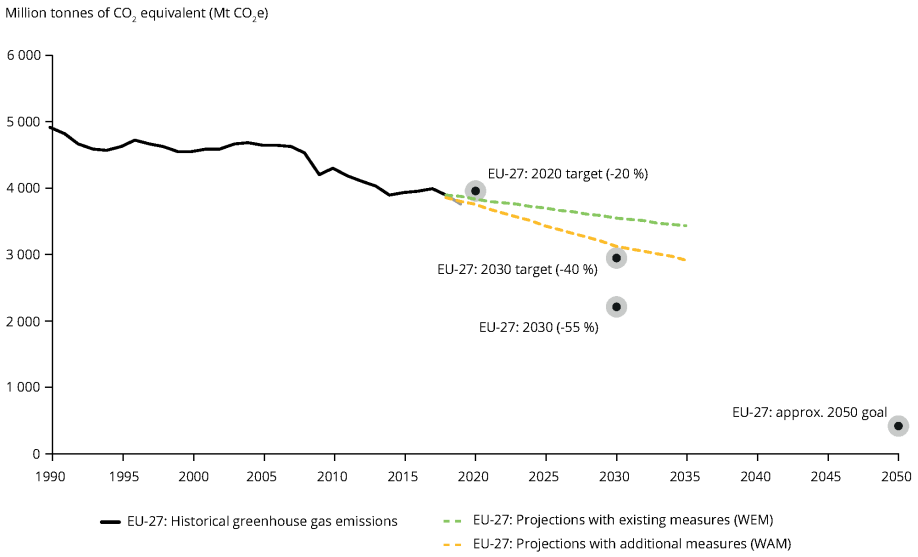


Figura 1: Proyección de los objetivos de emisiones de gases en el Pacto Verde Europeo [2]

Los estudios demuestran que la industria y el transporte son los agentes que más contribuyen en las emisiones de CO_2. Según la UE en [3], las emisiones por parte del transporte han aumentado estos últimos años, 2018 y 2019. Las proyecciones nacionales recopiladas por

la Agencia Europea del Medio Ambiente (AEMA) prevén que las emisiones del transporte en 2030 se mantendrán por encima de los niveles de 1990, incluso con las medidas actualmente previstas. Por ello, sugiere que es necesario tomar nuevas medidas, especialmente en el transporte por carretera, que es el que más contribuye a las emisiones del transporte, así como en la aviación y el transporte marítimo.

Uno de los grandes focos de nuevas medidas ecológicas se encuentra actualmente en la industria eléctrica. La generación y el consumo masivo de electicidad comenzó a finales del siglo XIX, cuando se extendió la iluminación eléctrica por las calles y casas. Desde entonces, el sector energético ha sido uno de los grandes pilares de la industria y ha crecido de manera exponencial. Sin embargo, esta red eléctrica tradicional se está quedando desfasada en cuanto a eficiencia, tecnologías y seguridad. Algunas de las grandes preocupaciones de las partes interesadas es (1) la reducción de emisiones CO_2, (2) la mezcla de generación de energía en centrales elécticas con un gran número de pequeños generadores distribuidos, (3) la disponibilidad intermitente de fuentes de energía renovable y (4) la eficiente transmisión y distribución de la electricidad [4].

El término *Smart Grid* (en español, red eléctrica inteligente) es usado para referirse al sistema eléctrico que engloba tango la generación, transmisión y distribución de energía, así como la recolección de datos y sensores de casas y edificios (smart meters) [5]. Una Smart Grid suministra electricidad de los proveedores a los consumidores utilizando tecnologías digitales para ahorrar energía, reducir costes y aumentar la fiabilidad y transparencia. Esta red inteligente se caracteriza por descentralizarse en subsistemas, llamados "*Migrogrids*". Una Microgrid es definida como una red local compuesta por un grupo de recursos de energía distribuidas (en inglés, Distributed Energy Resources (DERs)) y cargas, que operan normalmente conectados a la red inteligente principal (Smart Grid), o bien, de forma aislada (en modo isla) [6].

Tanto las Microgrids como Smart grids corresponden a sistemas ciberfísicos (en inglés, Cyber-Physical Systems (CPSs)). Estos heredan los mismos riesgos al encontrarse en un entorno crítico. Un fallo en el sistema eléctrico puede ocasionar daños humanos y/o elevados impactos económicos. Según la compañía de ciberseguridad Kaspersky en [7], el sector ener-

gético es el segundo sector industrial con mayor número de vulnerabilidades en sus sistemas de control, como se puede ver en la figura 2. Claros ejemplos de estas vulnerabilidades son los ataques producidos estos últimos años que han producido severos apagones en la comunidad. Es el caso del apagón en Ucrania en 2015, causado por intrusiones remotas tras la instalación de un malware en el sistema y afectando a aproximadamente 225.000 clientes [6]. O bien, el famoso malware *Stuxnet*, que espía y reprograma sistemas de control, afectando a infraestructuras críticas como centrales nucleares o redes eléctricas. Sin embargo, estos no son los únicos: Duqu (2011), BlackEnergy (2015 y 2016), Crashoverride (2016), NotPetya (2017) o Triton (2017) son ejemplos de la susceptibilidad de las redes eléctricas a diversos tipos de amenazas, donde muchos de ellos se encuentran registradas en el repositorio MITRE ATT&CK para sistemas de control industrial (en inglés, Industrial Control Systems (ICSs)) [8].

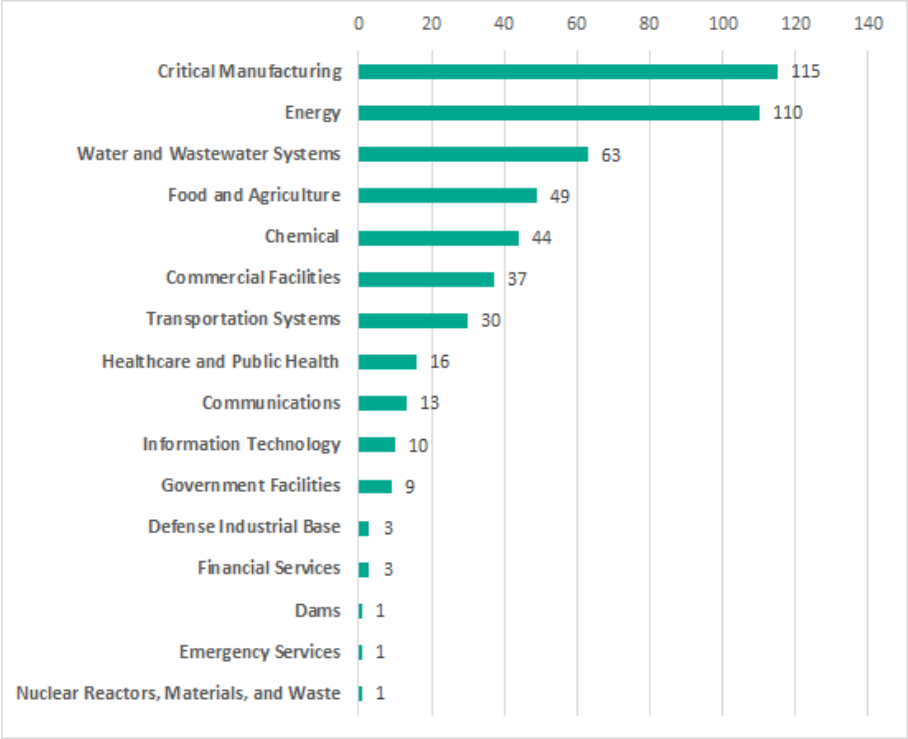


Figura 2: Número de productos vulnerables usados en diferentes industrias (de acuerdo a la clasificación US ICS-CERT). Vulnerabilidades publicadas en 2018 [7]

Por estas razones es necesario realizar estudios sobre la aplicación de seguridad en el sector energético, con el objetivo de lograr una correcta infraestructura distribuida y sostenible. Se deben seguir estándares, recomendaciones y guías de organizaciones estandarizadoras y regu-

ladoras, como el National Institue of Standards and Technology (NIST) y el European Union Agency for Cybersecurity (ENISA). NIST clasifica el área de conciencia situacional (en inglés, situational awareness) como una de las áreas prioritarias para proteger entornos de Smart Grids [9]. Una de las técnicas más usadas en este área es la detección basada en anomalías. Esta técnica es ampliamente usada para prevenir eventos imprevistos, incidentes o amenazas, a través de métodos estadísticos, machine learning o data-mining [10].

Los puntos de cargas son uno de los puntos más vulnerables en el entorno de una Microgrid. Son sistemas ciberfísicos abiertos al público y, por tanto, fácilmente manipulables físicamente y susceptibles a múltiples tipos de amenazas, lo que conlleva a diversos riesgos de seguridad. Como se demuestra en [11], atacantes pueden ocultar o mentir sobre la carga real para causar fraude (consumir más de lo que se paga), falsificar valores reales de consumo, suplantar la identidad de usuarios para robo de energía o alterar el sentido de la energía forzando los recursos de la Microgrid a sus límites máximos de tensión, corriente y/o potencia.

Por consiguiente, este Trabajo Fin de Grado (TFG) se enfoca en el estudio de una posible implementación y despliegue, de forma segura y limpia, de Microgrids basadas en puntos de carga de vehículos eléctricos, apoyando así el objetivo propuesto por la UE sobre la reducción de emisión de CO_2 (al incentivar el uso puntos de cargas provenientes de fuentes de energía renovable) y siguiendo, además, las recomendaciones de seguridad de organizaciones como NIST y ENISA y el repositorio MITRE ATT&CK para ICS.

1.2. Objetivos

El objetivo principal de este TFG es *implementar una primera línea de defensa en un sistema de control de una Migrogrid, centrando el estudio en medidas específicas de detección de anomalías y trazabilidad sobre el consumo real de energía en diversos puntos de carga.* Estas medidas permiten explicar de primera mano lo que está actualmente ocurriendo en el sistema (conciencia situacional), a través de un sistema de monitorización. Para ello, es necesario dividir el trabajo en los siguientes objetivos:

1. Realizar una taxonomía de vulnerabilidades y amenazas sobre una microgrid basada en puntos de cargas, usando el repositorio de ataques de MITRE ATT&CK para ICSs.

2. Diseño e implementación de una arquitectura basada en una red descentralizada de tipo Blockchain. Este objetivo requiere de un análisis previo sobre los diferentes tipos de redes Blockchain.

3. Identificar varias técnicas machine-learning que garanticen una rápida y fiable detección de anomalías. Para ello, se realizan comparativas entre ellas en términos de precisión, sensibilidad y fiabilidad.

4. Implementar un servicio de monitorización y trazabilidad, que visualice el estado del sistema y alerte de posibles amenazas en los puntos de cargas.

1.3. Estructura del documento

La estructura de este documento se encuentra organizada en varios grupos de secciones. Los primeros capítulos inician al lector sobre el problema a tratar en este proyecto. Posteriormente, los siguientes capítulos presentan el desarrollo del trabajo con sus correspondientes resultados y conclusiones. Por último, en la sección de Anexos, se recogen diferentes documentaciones sobre el proyecto que quedan fuera del cuerpo de la memoria.

El primer capítulo trata sobre el estado del arte del problema (sección 2), donde se recogen todos los conceptos básicos y estudios, hasta la fecha, sobre la ciberseguridad en Microgrids. Posteriormente, se indican las tecnologías que se han utilizado a lo largo del proyecto (sección 3) y la metodología de trabajo aplicada (sección 4).

El segundo grupo de capítulos comienza con los requisitos y diseño del sistema a implementar (sección 5), así como la definición y funcionalidades de sus componentes. Luego, se estudia una posible taxonomía de vulnerabilidades y amenazas para esta arquitectura (sección 6), usando para ello un modelo de amenazas. A continuación, en la sección 7, se muestra el despliegue y configuración de la red Blockchain, con sus principales características. El capítulo 8 explica la necesidad de un simulador de puntos de cargas, así como la recogida de datos y

ataques simulados en varios datasets. Estos datasets son usados posteriormente en el capítulo detección de anomalías (sección 9), donde se muestran los métodos de aprendizaje más eficientes y los resultados obtenidos para cada caso. Finalmente, el último capítulo de desarrollo (sección 10), explica las funcionalidades principales del sistema de monitorización, así como las opciones que tiene un usuario autorizado de observar el estado y las anomalías del sistema, finalizando así con la sección de pruebas y validaciones (sección 11) de los requisitos definidos en el sistema.

Por último, el apartado de conclusiones y líneas futuras, en la sección 12, analiza a nivel personal el trabajo realizado y los resultados obtenidos, mostrando también posibles ideas de nuevas líneas de trabajo que se podrían desarrollar a partir de éste.

2

Estado del arte

2.1. Introducción a las Smart Grids y Microgrids

La industria eléctrica ha sido testigo de muchos desarrollos recientes que no sólo han reavivado el interés por la investigación y el desarrollo, sino que también han dado lugar a importantes beneficios socioeconómicos. El incremento de la concienciación sobre el impacto medioambiental y la huella de carbono de la mayoría de fuentes de energía, incluida la producción de energía eléctrica, son motivos que han impulsado al crecimiento y la adopción de energías renovables y alternativas, apareciendo el término de "red inteligente" (comúnmente denominada en inglés, Smart Grid) [4].

Actualmente, el término Smart Grid presenta una amplia variedad de estudios en la literatura académica, con un resultado total de 45.033 búsquedas en IEE Xplore y de 949, en el buscador de la biblioteca de Málaga, Jabega. Según la Comisión Electrotécnica Internacional (CEI) [12], "*Una Smart Grid es una red que incorpora las tecnologías de la información y la comunicación (TIC) en todos los aspectos de la generación, el suministro y el consumo de energía con el fin de minimizar el impacto ambiental, mejorar los mercados, la fiabilidad, el servicio y la eficiencia, y reducir los costes*".

2.1.1. Características de una Smart Grid

Las Smart Grids proporcionan grandes ventajas tanto al sector energético como a todas las partes interesadas. Algunas de ellas son [12]:

- Reducir las emisiones de CO_2, incrementando la generación de electricidad de fuentes de energía renovable.

- Se introducen tecnologías de eficiencia energética para reducier el consumo global y gestionar las interacciones con el calor y el gas.
- Nuevas formas de almacenar energía, como el uso de las baterías de los vehículos eléctricos (VE) enchufados a la red.
- La red de transporte y distribución se moderniza para dar cabida a una mayor demanda, así como para ser más eficiente energéticamente.
- El uso de DERs, situadas cerca del lugar del consumo, complementan la red convencional y se integran en la red eléctrica, gracias al uso de los sistemas denominados *Microgrids*.

2.1.2. Concepto de Microgrid

CEI define una Microgrid como "*un grupo de cargas interconectadas y recursos energéticos distribuidos (DERs) con límites eléctricos definidos que forman un sistema local de energía eléctrica a niveles de tensión de distribución, que actúa como una única entidad controlable y es capaz de operar en modo conectado a la red (Smart Grid) o en modo isla (aislado)*".

Los recursos energéticos distribuidos (es decir, los DERs) son unidades de generación de energía a pequeña escala que operan localmente y se encuentran conectadas a una red eléctrica de mayor escala y distribuida. Este componente es fundamental en el sistema para que una Microgrid logre trabajar de modo aislada de la red principal. Estos recursos incluyen unidades de generación basadas en energía renovable, dispositivos de almacenamiento y generadores convencionales [13].

2.2. Arquitectura general de una Microgrid

La arquitectura de una Microgrid corresponde con la estructura general de un sistema ciberfísico (CPS). Una Microgrid está sujeta a un conjunto de componentes electrónicos (capa física) interconectados entre sí por diferentes tecnologías TIC [14].

Como se muestra en la figura 3, el modelo de una Microgrid se divide generalmente en 4 capas: (1) capa física, (2) capa de sensores y actuadores, (3) capa de comunicación y (4) capa de mantenimiento y control.

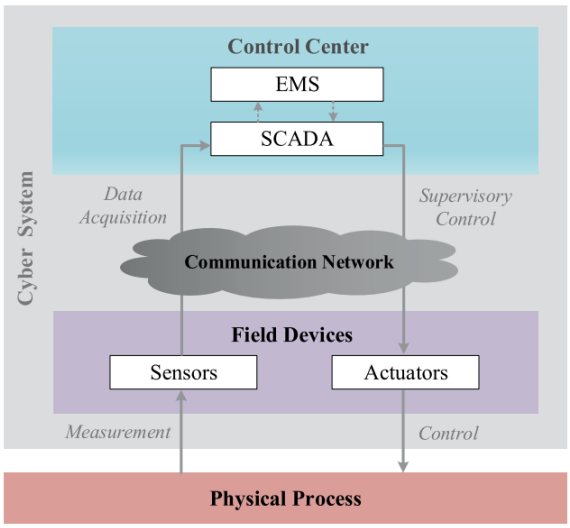


Figura 3: Arquitectura general de una Microgrid [14]

- La capa física contiene los componentes de energía como transformadores, generadores, convertidores, circuitos y cargas.

- La segunda capa consiste en dispositivos de medidas (sensores) y dispositivos que ejecutan las decisiones de control (controladores y actuadores) tomadas desde el sistema de control. Esta capa podría omitirse y considerarse como la interfaz de la capa física.

- La capa de comunicación se compone por dispositivos de red, como routres y switches, y el medio de comunicación (cableado o inalámbrico). Esta es la responsable del intercambio de información entre el resto de capas.

- Por último, la capa de mantenimiento corresponde con un sistema de control central que se encarga de decidir las operaciones de la Microgrid bajo diferentes condiciones. En este nivel se recibe información de medidas de los sensores (Data Acquisition) y produce señales de control (Supervisory Control) que optimicen las operaciones de la Microgrid. Estas señales son enviadas a los controladores y actuadores. El sistema de control central se divide en dos grandes componentes:

    - **Supervisión, Control y Adquisición de datos (en inglés, Supervisory Control And Data Acquisition (SCADA))**: sistema software que permite controlar y supervisar procesos industriales a distancia. Facilita la retroalimentación y el con-

trol de los diferentes dispositivos electrónicos (controladores, sensores y actuadores) de la Microgrid. Para ello, adquiere medidas en tiempo real de los sensores y envía las operaciones tomadas por el sistema de gestión energética a los controladores [13].

- **Sistema de gestión energética (en inglés, Energy Management System (EMS))**: sistema software que supervisa, controla y optimiza el rendimiento de las operaciones de la Microgrid. EMS corresponde con el procesador "back-end" con capacidades de toma de decisiones. Este ejecuta una colección de funciones para mantener la seguridad, confiabilidad, economía, resiliencia, sustentabilidad y eficiencia en la Microgrid. Se apoya de otras tecnologías, como la inteligencia artificial y la minería de datos [13].

2.3. Vulnerabilidades y requisitos de seguridad en CPS

Los sistemas ciberfísicos (CPS) incluyen sistemas de muchos tipos, como la robótica, automatización de máquinas, los ICSs, los sistemas de control de procesos, los sistemas SCADA, el internet industrial, el Internet of Things (IoT), las Smart Grids y las Microgrids.

Una Microgrid es propensa a las vulnerabilidades comunes que presenta un sistema ciberfísico. Estas vulnerabilidades pueden ser ocasionadas por amenazas desde un aspecto físico, como el cambio climático (altas temperaturas), ataques físicos causados por humanos o desastres naturales; o bien, desde un aspecto cibernético, como, por ejemplo, manipulación de datos, intrusión en el sistema de control, robo de información, denegación de servicio, etc. En la tabla 1, se puede observar ejemplos reales de incidencias en la industria eléctrica durante los últimos años.

Estas vulnerabilidades han sido objeto de estudio y preocupación durante muchos años en el campo de investigación. En el artículo de Yaacoub [15], se recoge una clasificación detallada sobre las diferentes vulnerabilidades, amenazas e impactos para cada uno de los casos: capa física y cibernética. Por otro lado, la organización sin ánimo de lucro, MITRE, ha recopilado una base de conocimiento útil para describir las acciones que un adversario puede realizar mien-

Tabla 1: Incidentes en el sector energético durante los últimos años [13]

Año	Lugar	Causa	Consecuencias
2003	Noreste	Fallo del sistema de alarma debido a un error (bug) de software	Más de 50 millones de clientes perdieron la electricidad
2003	Italia	Fallos en cascada entre las infraestructuras de energía y comunicación	Alrededor de 56 millones de personas fueron afectadas
2007	Arizona	Activación inesperada del programa de desconexión de la carga	Sobre 100.000 clientes perdieron 400 MW de carga
2008	Florida	Protección del relé desactivada durante un proceso de diagnóstico	Alrededor de un millón de clientes perdiern 3.650 MW de carga
2013	Austria	Mal configuración en el sistema de control	Partes de monitorización y control desactivadas
2015	Ucrania	Intrusiones cibernéticas remotas tras la instalación del malware	Aproximádamanete 225.000 clientes perdieron la electricidad
2017	Arabia Saudí	Inyección de malware, llamado Triton, con objetivo de causar daño físico	Robo de información y conocimiento completo de los protocolos

tras opera dentro de un sistema de control industrial, denominado MITRE ATT&CK for ICS [8].

Para cubrir estas vulnerabilidades, se intentaron implementar los pilares de seguridad más comunes: disponibilidad, integridad, confidencialidad, autenticación, autorización y no repudio. Sin embargo, organizaciones reguladoras, como NIST, propusieron nuevas soluciones. En 2015, NIST publicó la segunda versión de la guía de seguridad para ICS (NIST 800-82) [16]. Actualmente, NIST ha iniciado una tercera versión [17] en modo de borrador, que busca actualizar las recomendaciones de prácticas, debido a que muchas de las tecnologías y herramientas han cambiado desde la versión anterior. Además, en 2018, NIST publica el "*Framework for Improving Critical Infrastructure Cybersecurity*" [18], donde incluye el proceso completo para implementar ciberseguridad en un sistema ciberfísico. Este proceso consta de las siguientes cinco funciones (ver Figura 4):

1. **Identificar (ID)**: según NIST, la función de identificación ayuda a desarrollar una comprensión organizativa para gestionar los riesgos de ciberseguridad en sistemas, personas, activos, datos y capacidades. Comprender el contexto empresarial, los recursos que soportan las funciones críticas y los riesgos de ciberseguridad relacionados permiten a una organización centrar y priorizar sus esfuerzos, en consonancia con su estrategia de gestión de riesgos y sus necesidades empresariales.

2. **Proteger (PR)**: esta función implementa y desarrolla medidas de seguridad adecuadas para garantizar la entrega de servicios críticos. Esta función limita o contiene el impacto de un posible evento de ciberseguridad.

3. **Detectar (DE)**: la función de detección define las actividades apropiadas para identificar en tiempo real la ocurrencia de amenazas en el sistema.

4. **Responder (RS)**: incluye actividades para tomar acción sobre incidentes de ciberseguridad detectados. Esta función respalda la capacidad de contención del impacto de una amenaza en el sistema.

5. **Recuperar (RC)**: esta función identifica actividades para mantener los planes de resiliencia y para restaurar cualquier capacidad o servicio que se haya visto afectado por una amenaza.

Function Unique Identifier	Function	Category Unique Identifier	Category
ID	Identify	ID.AM	Asset Management
		ID.BE	Business Environment
		ID.GV	Governance
		ID.RA	Risk Assessment
		ID.RM	Risk Management Strategy
		ID.SC	Supply Chain Risk Management
PR	Protect	PR.AC	Identity Management and Access Control
		PR.AT	Awareness and Training
		PR.DS	Data Security
		PR.IP	Information Protection Processes and Procedures
		PR.MA	Maintenance
		PR.PT	Protective Technology
DE	Detect	DE.AE	Anomalies and Events
		DE.CM	Security Continuous Monitoring
		DE.DP	Detection Processes
RS	Respond	RS.RP	Response Planning
		RS.CO	Communications
		RS.AN	Analysis
		RS.MI	Mitigation
		RS.IM	Improvements
RC	Recover	RC.RP	Recovery Planning
		RC.IM	Improvements
		RC.CO	Communications

Figura 4: Funciones y categorías de "*Framework for Improving Critical Infrastructure Cybersecurity*" [18]

Aparte del framework propuesto por NIST (NIST 800-82), existen otros frameworks de ciberseguridad para ICS, como ISA/EIC 62443 y IC-IISF; e incluso, frameworks específicos para el sector eléctrico: IEE 402, ES-C2M2, NERC CIP, ENISA Smart Grid Threat Landscape and Good Practice Guide y ENISA Appropriate Security Measures for Smart Grids.

2.4. Detección de anomalías

Siguiendo las recomendaciones de seguridad mencionadas anteriormente, uno de las buenas prácticas es la detección basada en anomalías, para obtener una mayor conciencia situacional del sistema, y poder responder y recuperarse a tiempo.

A continuación, se detalla la taxonomía propuesta por [19] para ilustrar de forma general las técnicas de detección más empleadas en sistemas ciberfísicos.

1. **Técnicas basadas en reglas**: trabajan principalmente en base a un conocimiento previo del modelo. Estas técnicas son muy rápidas y baratas de implementar.

2. **Técnicas estadísticas**: son probablemente las más empleadas en este área. Se dividen en dos tipos: (1) paramétricas y (2) no paramétricas. Las técnicas paramétricas hacen uso de una supuesta distribución subyacente, mientras que las técnicas no paramétricas aprenden la distribución subyacente por sí mismas.

3. **Técnicas de minería de datos e inteligencia computacional**: aprenden a partir de datos de entrenamiento multivariados con un procedimiento de aprendizaje bastante complejo, que puede ser supervisado, semisupervisado o no supervisado.

2.4.1. Detección de anomalías basadas en Machine Learning

El aprendizaje automático (en inglés, machine learning) corresponde con un área de la Inteligencia Artificial (IA). Este es definido, según el glosario de Google [20], com *el programa o sistema que entrena un modelo predictivo a partir de datos de entrada (muestras).* El sistema usa el modelo aprendido para realizar predicciones útiles a partir de datos nuevos obtenidos de la misma distribución que la que se usó para entrenar el modelo.

Para justificar el comentario anterior, esta técnica es ampliamente usada para la detección de anomalías, que suelen corresponder a valores atípicos (en inglés, *outliers*) de la distribución. Esta técnica suele ser usada por modelos de clasificación donde el atributo a predecir (clase) corresponde con un valor binario: 0, si no es anomalía; 1, si corresponde a un valor atípico o anomalía. Estas técnicas de detección son las más adecuadas para entornos de Smart Grid

debido a su capacidad de detectar ataques potenciales desconocidos y no requiere de una base de datos de firmas con ataques preestablecidos. Algunas de las técnicas más empleadas para la predicción de anomalías en una Microgrid se encuentran recogidas en la tabla 2:

Tabla 2: Las técnicas de detección de anomalías basadas en Machine Learning más empleadas en una Microgrid [21]

Tipo de aprendizaje	Método
Aprendizaje Supervisado	Support Vector Machine (SVM) Decision Tree (DT) K-Nearest Neighbor (ENN) Artificial Neural Network (ANN)
Aprendizaje No Supervisado	K-Means Fuzzy C-Means (FCM) Gradient Boosting Classifier (GBC)
Aprendizaje Profundo	Multilayer Perceptron (MLP) Recurrent Neural Network (RNN) Convolutional Neural Network (CNN)

2.5. Distributed Ledger Technology

Finalmente, para proporcionar una correcta trazabilidad de cada punto de carga es necesario buscar una solución segura, que no comprometa la estabilidad de la red o el sistema central. Por ello, la tecnología Distributed Ledger Technology (DLT)) corresponde con una de las soluciones más prometedoras y extendidas últimamente. La tecnología DLT corresponde con una base de datos distribuida y gestionada por varios participantes. Concretamente, uno de los tipos de DLT más usados y con mayor avance en los últimos años es la famosa tecnología Cadena de Bloques (en inglés, Blockchain). Las características principales de una DLT o Blockchain son: inmutabilidad de los datos, los datos una vez escritos no se pueden modificar; distribuido, cada nodo tiene una copia completa o parcial de la base de datos; y compartido, el registro se comparte entre las partes a través de algoritmos de consenso. Estas características establecen una serie de garantías de redundancia de datos, que son fundamentales en entor-

nos críticos como una Smart Grid o una Microgrid para evitar pérdidas de datos (integridad) o retrasos a la hora de ser solicitados (disponibilidad).

Por otro lado, una DLT se suele dividir en dos tipos en base a los permisos de acceso y mantenimiento: (1) no permisionada (*permissionless*), si cualquiera puede unirse a la red y publicar un nuevo registro, y (2) permisionada (*permissioned*), si sólo usuarios autorizados pueden publicar nuevos registros.

Como en este TFG se busca crear un novedoso escenario basado en puntos de cargas distribuidos y conectados a un sistema central, donde se toman las operaciones. Para ello, se hace uso de una DLT permisionada y privada junto con técnicas de detección de anomalías para proporcionar así, una primera línea de consciencia situacional, a través de un sistema de monitorización.

Al ser una tecnología muy reciente, no existen apenas artículos donde se hayan probado o implementado este caso de uso, siendo este TFG pionero en este ámbito. En los artículos [22], [23], se estudian la posible solución de usar una DLT para el negocio energético, y precisamente en [24], se propone una arquitectura de IoT basada en una DLT permisionada y funciones de coste para recomendar el punto de carga más conveniente a los VE que necesiten cargarse.

3

Tecnologías usadas

A continuación, se listan las tecnologías implicadas y usadas a lo largo de este TFG.

3.1. Servidor

Máquina virtual instalada en la red eduroam de la Universidad de Málaga (UMA), con la capacidad suficiente para desplegar y soportar una red Blockchain permisionada con cuatro nodos, el sistema de monitorización, el sistema de detección de anomalías y las simulaciones de los puntos de carga. La configuración del servidor es la siguiente:

- **Sistema operativo**: Ubuntu 18.04.5 LTS
- **Arquitectura**: 64 bits
- **Procesador**: Intel(R) Xeon(R) Silver 4214 CPI @ 2.20GHz
- **Memoria RAM**: 16 GB
- **Disco de almacenamiento**: 156 GB SSD
- **Conexión a Red**: Ethernet

3.2. Red Blockchain

Se ha desplegado una red Blockchain permisionada y privada usando las propiedades y la documentación que nos facilita la red Ethereum y los clientes del proyecto Hyperledger, concretamente la tecnología Hyperledger Besu.

- **Ethereum**: plataforma global descentralizada de código abierto. Se construye sobre la innovación de Bitcoin, con la diferencia de que este es programable, así que permite a desarrolladores crear nuevas aplicaciones descentralizadas (DApp) [25].

- **Hyperledger Besu:** corresponde con el último cliente Ethereum diseñado en el proyecto Hyperledger de la organización Umbrella, aunque el principal contribuidor del código base ha sido la empresa neoyorquina ConsenSys. Este cliente, basado en java, es diseñado para ser amigable con las empresas para casos de uso de redes públicas o privadas (permisionadas). Este incluye varios algoritmos de consenso como *Proof of Work (PoW)* y *Proof of Authority (PoA)* [26].

3.3. MongoDB

MongoDB es un sistema de base de datos no relacional desarrollado por la empresa MongoDB Inc [27]. Este está basado en documentos, que se encuentran almacenados en formato JavaScript Object Notation (JSON). Es de código libre y permite su uso en local, o bien, desde su servicio alojado en la nube MongoDB Atlas [28]. Se ha usado esta base de datos para almacenar las anomalías de forma local en el servidor.

3.4. Lenguajes de programación

Durante el proyecto se ha usado varios lenguajes de programación para cada uno de los componentes del sistema. El principal lenguaje para el procesamiento de datos ha sido Python. Por otra parte, para el frontend de la página web (monitor) se ha usado principalmente React.js. Por último, para la definición de los contratos inteligentes se ha usado el reciente lenguaje Solidity.

- **Python**: lenguaje de programación interpretado, dinámico y multiplataforma desarrollado por Python Software Foundation. Se ha usado en gran parte del TFG para desarrollar tanto el back-end del sistema de monitorización, como los algoritmos de detección de anomalías y los simuladores de puntos de carga [29].

- **React.js**: corresponde con una biblioteca de JavaScript para construir interfaces de usuario, desarrollada por Facebook y la comunidad. Corresponde con un lenguaje declarativo y basado en componentes. Se ha usado para diseñar la página web del sistema de monitorización [30].

- **Solidity**: lenguaje de programación de alto nivel orientado a contratos inteligentes (en inglés, smart contracts). Su lenguaje es similar a JavaScript y es específico para la Máquina Virtual de Ethereum (EVM) [31].

3.5. Frameworks y Librerías

A continuación, se muestran algunas de las librerías y frameworks principales que se han usado para este escenario:

- **Django**: framework de código abierto sobre desarrollo web de alto nivel en Python [32]. Este fomenta el patrón Modelo Vista Controlador (MVC) y es usado, junto el framework Django Transferencia de Estado Representacional (REST), para el diseño del sistema de monitorización. Django REST corresponde con un conjunto de herramientas para construir una API REST web [33].

- **Nivo**: librería que proporciona un conjunto de componentes para mostrar gráficas, construidas a partir de las bibliotecas d3 (produce a partir de datos infogramas dinámicos e interactivos en navegadores web) y Reactjs [34].

- **Scikit-learn, pyod y anomatools**: bibliotecas para aprendizaje automático (Machine Learning) sobre el lenguaje Python. Tanto *pyod* [35], que contiene algoritmos de detección de valores atípicos (outliers), como *anomatools* [36], que incluye algoritmos sobre detección de anomalías, usan como base la librería *scikit-learn* [37].

- **Pymongo**: distribución de Python con las herramientas necesarias para trabajar con MongoDB [38].

- **Web3.py**: librería de Python para interactuar con Ethereum. Permite el envío de transacciones, interacción con los contratos inteligentes (los smart contracts) y lectura de bloques. Esta es derivada de la librería original Web3.js [39].

- **Truffle**: framework para JavaScript que trabaja sobre los activos de Ethereum. Facilita, entre otros, la compilación y despliegue de contratos inteligentes [40].

3.6. MetaMask

MetaMask es un monedero de criptomonedas y una puerta de entrada a las aplicaciones Blockchain [41]. Este puede ser instalada como extensión de navegador y permite interactuar con la plataforma Ethereum. Este monedero almacena y gestiona cuentas, con las cuales se pueden conectar a aplicaciones descentralizadas o enviar transacciones a otras cuentas. Se ha usado como base de autenticación para entrar en el sistema de monitorización, solo cuentas autorizadas tienen permiso de conexión a la web.

4

Metodología

Durante este TFG se ha seguido una metodología ágil, sobre todo usando las características comunes de la metodología *Scrum* [42]. Se ha llevado un desarrollo incremental de los requisitos basándose en bloques temporales cortos (sprints) y con un seguimiento continuo semanal por parte de los tutores. El proyecto se ha dividido en un total de tres sprints, donde al final de cada uno de ellos se ha realizado una reunión para verificar y corregir los entregables de dicho sprint. En la figura 5, se puede observar el tiempo de dedicación a cada una de las actividades y sprints. Cada uno de los sprints recogen los siguientes requisitos:

- **Sprint 1**: corresponde al período mayoritario de investigación sobre el estado de arte de cada una de las tecnologías implicadas y, sobre todo, en las amenazas más comunes en sistemas ciberfísico. Este sprint ha finalizado con el desarrollo del documento de taxonomía de amenazas de una Microgrid basada en puntos de cargas y dentro de una red Blockchain permisionada.

- **Sprint 2**: durante este sprint se ha desarrollado una red Blockchain permisionada de cuatro nodos, usando la tecnología Hyperledger Besu. Tras su despliegue, se ha implementado varios Smart Contracts para guardar datos sobre los puntos de cargas y transacciones. Para finalizar, se ha diseñado un sistema de trazabilidad que muestre a través de la web el estado de cada punto de carga en una determinada fecha.

- **Sprint 3**: en el último sprint, se ha investigado y probado diferentes técnicas de anomalías sobre los puntos de carga. Para ello, ha hecho falta también el diseño y desarrollo de simuladores de puntos de cargas y ataques simulados. Para finalizar, se ha mejorado el sistema de trazabilidad por un sistema de monitorización en tiempo real usando los detectores de anomalías, con los que se ha experimentado.

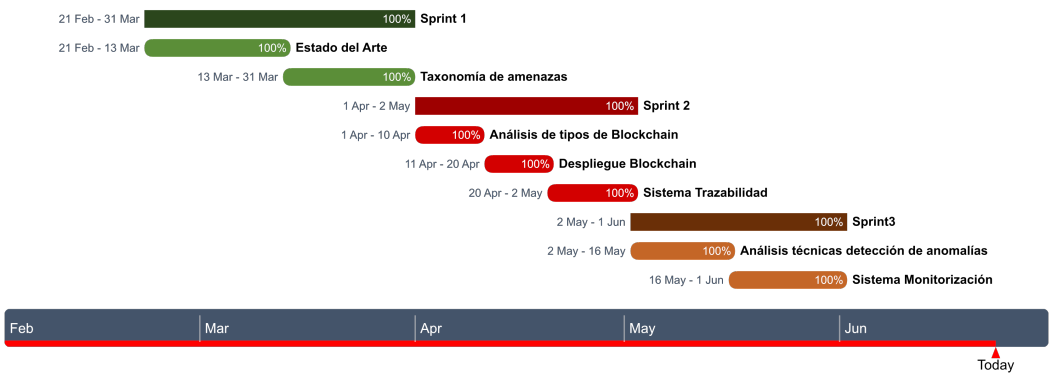


Figura 5: Diagrama de Gantt

5

Requisitos y diseño

En esta sección, se define los requisitos y el diseño del escenario implementado en el TFG, que posteriormente será utilizado, en la sección 6, para el análisis de amenazas y vulnerabilidades. Tras mostrar el diseño y la arquitectura, se explica detalladamente la funcionalidad de cada uno de los componentes del sistema, con el objetivo de facilitar la comprensión del mismo.

5.1. Requisitos

A continuación, se enumeran todos los requisitos funcionales (tabla 3) y no funcionales (tabla 4) que se exigen en este TFG para alcanzar los objetivos definidos previamente en la sección 1.2. Los requisitos se han priorizados de uno a cinco, donde cinco corresponde a la mayor prioridad.

Tabla 3: Requisitos funcionales de SecCP

Nº	Descripción	Prioridad
RF01	Desplegar una red Peer-to-Peer (P2P) permisionada y privada de 4 nodos	5
RF02	Almacenar datos de consumo sobre los estados y transacciones de los puntos de carga	5
RF03	Almacenar datos de consumo sobre las transacciones de carga en la red Blockchain	5
RF04	Trazar los estados y transacciones de los puntos de carga	5
RF05	Trazar las transacciones de carga de un usuario	2
RF06	Almacenar anomalías detectadas de manera local en base a uno o más métodos de aprendizaje automático.	5
RF07	Monitorizar los últimos estados, transacciones y anomalías en tiempo real	5
RF08	Localizar anomalías en tiempo real	4
RF09	Autenticar y autorizar el acceso al sistema de monitorización	3

Tabla 4: Requisitos no funcionales de SecCP

Nº	Descripción	Prioridad
RNF01	Interfaz gráfica intuitiva y fácil de usar. Se puede acceder con menos de cinco interacciones a cualquier funcionalidad.	3
RNF02	Curva de aprendizaje exponencial del sistema de monitorización. Menos de media hora para conocer todas las funcionalidades.	3
RNF03	Tiempo de respuesta en las consultas menor a 5 segundos (disponibilidad)	4
RNF04	Cifrado de todos los canales de comunicación (confidencialidad, integridad)	5
RNF05	Control de acceso y autorización en base a permisos en la red Blockchain	5
RNF06	Visualización de la localización de anomalías a través de mapas	3
RNF07	Visualización de datos estadísticos de consumo y anomalías a través de gráficas interactivas	3

5.2. Arquitectura de una Microgrid basada en puntos de cargas para vehículos eléctricos

A continuación, en la figura 6, se propone el diseño de una Microgrid basada en estaciones de cargas de vehículos eléctricos. Este escenario está compuesto por diferentes componentes independientes que interactúan entre sí, sin definir la red Blockchain que se discutirá en la sección 7. Para facilitar la redacción, denominamos al nuevo modelo diseñado con el nombre "Secure Charging Points (SecCP)".

5.3. Componentes de SecCP

En esta sección, se definen todos los componentes independientes involucrados en SecCP, los cuales se usarán posteriormente en la sección 6 para identificar las vulnerabilidades y amenazas en cada uno de ellos.

- **Sistema de Control Central**: corresponde con el servidor "back-end" del sistema. Este servidor se encarga de la interacción y administración de los elementos de la Microgrid, los puntos de carga y los usuarios. Se encuentra compuesta por dos capas: (1) una aplicación de validación, autorización y control encargada de recibir las peticiones de los usuarios; y (2) un centro de control de peticiones basadas en el protocolo Open Charge Point Protocol (OCPP) [43]. La funcionalidad principal de este servidor OCPP es recibir peticiones de transacción e interactuar y gestionar los puntos de carga. Además, este

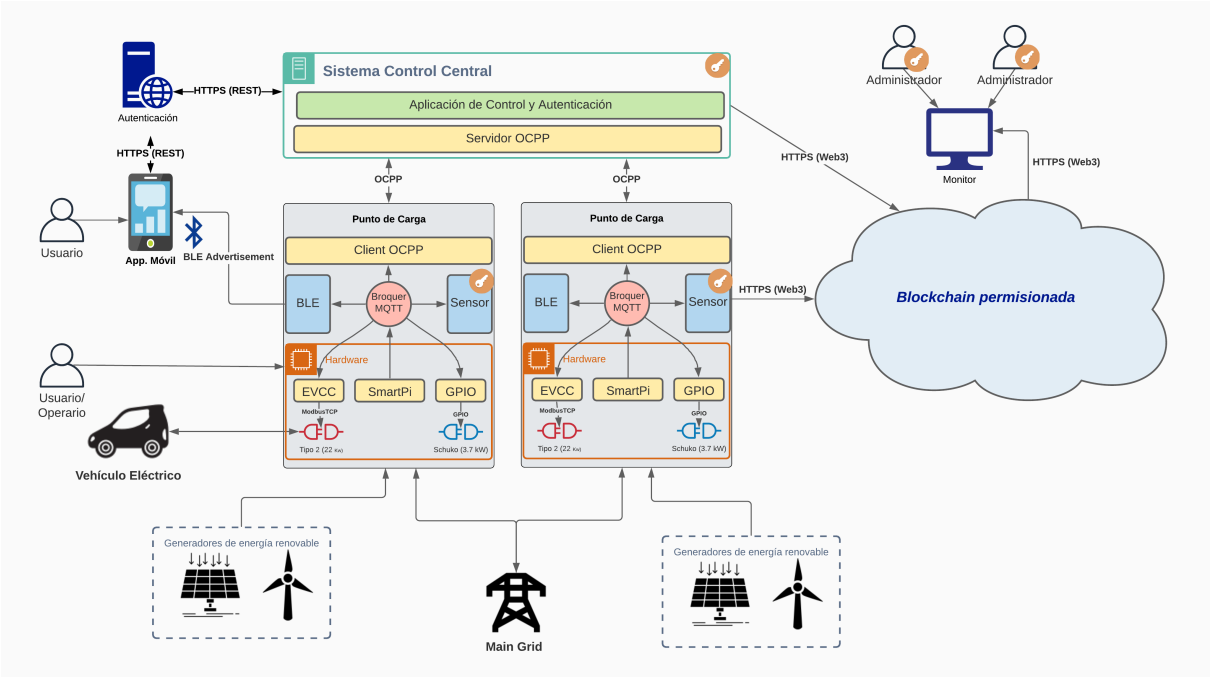


Figura 6: Arquitectura de SecCP

sistema central es el encargado de enviar en tiempo real los datos de consumo a la red Blockchain tras la finalización de cada transacción de un usuario.

- **Punto de carga**: representa el elemento más visible del sistema. Este pertenece a una plataforma hardware sobre la que se ejecutan una serie de componentes software. Los puntos de carga de SecCP presentan dos tomas de conexión: una toma monofásica de tipo F (Schuko) y otra trifásica de tipo 2 (Mennekes). Estas tomas se encuentran desactivadas inicialmente y son activadas a petición del usuario a través del Sistema de Control Central. Para que un usuario pueda realizar una petición, debe estar lo suficientemente cerca del punto de carga, usando la tecnología Bluetooth Low Energy (BLE).

  Se hace uso del protocolo Message Queing Telemetry Transport (MQTT) [44] para establecer la comunicación interna entre los diferentes componentes software del punto de carga. Este protocolo se basa en un modelo publicador/suscriptor y se encarga de difundir el estado actual del punto de carga. Además de la posibilidad de cargar en los dos posibles conectores, el punto de carga presenta las siguientes funcionalidades y módulos software:

  - **Cliente OCPP**: componente encargado de la interacción con el sistema central

a través del protocolo OCPP. El cliente recibe las peticiones del sitema central y publica los cambios (activar/desactivar) para los correspondientes conectores a través del protocolo MQTT. Además, registra y notifica los datos de consumo de cada transacción de carga.

- **Módulo BLE**: difunde de forma continua mensajes BLE de tipo "anuncio" (advertisment) que contienen tanto la presencia del punto de carga como el estado actual de los conectores. Su objetivo es que la aplicación móvil pueda detectar los puntos de carga cercanos y habilitar su interacción.
- **Módulo Sensor**: se caracteriza por enviar en tiempo real los datos recogidos por los distintos sensores integrados en el punto de carga a la red Blockchain de forma segura. Entre los datos recogidos y enviados se encuentran: el estado (activado/-desactivado), la potencia eléctrica demandada y el consumo acumulado de cada conector.
- **Módulo SmartPi**: este módulo representa un componente de interacción con el componente hardware SmartPi integrado en el punto de carga. Su función es analizar el consumo eléctrico de ambos conectores (Mennekes y Schuko). Para ello, monitoriza las líneas eléctricas del punto de carga y publica de forma periódica la potencia y el consumo actual de cada uno de los conectores al Broquer MQTT, que posteriormente difundirá la información a los suscriptores.
- **Módulo Electric Vehicle Charger Controller (EVCC)**: componente de interacción con el componente hardware EVCC de Phoenix Contact [45] integrado en el punto de carga. Este módulo controla el proceso de carga de un vehículo eléctrico a través del conector tipo 2 (Mennekes).
- **Módulo General Purpose Input/Output (GPIO)**: corresponde con el componente de interacción con el puerto GPIO de la Raspberry Pi integrada en el punto de carga. De forma similar que el módulo EVCC, este controla el proceso de carga del conector tipo F (Schuko).
- **Conector Mennekes**: conector estándar de la norma tipo 2 (IEC 62196) para dispositivos de conexión de carga para la movilidad eléctrica, fabricado por la empresa Mennekes [46]. Este permite la carga con una amplitud de potencia preestablecida

desde 1.4 kW a 22 kW, representando una curva de carga lenta. Normalmente el cable de conexión, de tipo Mennekes, de los vehículos eléctricos pueden establecer los siguientes niveles de potencia: 2.3kW (1x10A), 3.7kW (1x16A), 7.4kW (1x32A), 11kW (3x16A) y 22kW (3x32A). En la sección 9, se analiza mejor el comportamiento y la curva de carga de este conector, para las posteriores simulaciones y ataques.

- **Conector Schuko**: toma de corriente estándar de la norma tipo F (CEE 7) [47]. Corresponde con el enchufe genérico, de clavija y base, más utilizado en los aparatos eléctricos de la mayoría de hogares y oficinas europeas. Este conector no permite potencias superiores a 2.3kW (16A).

- **Aplicación Móvil**: se propone el uso de una aplicación móvil que reciba los puntos de cargas más cercanos junto con sus conectores a través de la tecnología BLE. Además, esta aplicación tiene la posibilidad de activar y desactivar cada uno de los conectores detectados, siempre y cuando se autentique correctamente ante la plataforma de autenticación.

- **Plataforma de autenticación**: Esta plataforma corresponde con un servidor de autenticación centralizado. Permite la autenticación y autorización a los usuarios para solicitar peticiones a través de la aplicación móvil hacia el servidor central. Esta plataforma podría ser sustituida por cualquier otro servicio de autorización, como por ejemplo, OAuth 2.0 [48].

- **Recursos distribuidos de energía renovable (DERs)**: conforman principalmente los generadores locales de energía renovable integrados y controlados en la Microgrid. Sirven como fuente de energía siempre y cuando logren abastecer toda la demanda. En cualquier otro caso, se activa como ayuda la energía procedente de la red principal (Smart Grid).

- **Vehículo Eléctrico**: corresponde con el componente destino y objetivo de este sistema. Los vehículos eléctricos sólo pueden cargar sus baterías a través de los conectores de los puntos de cargas. Sin embargo, puede existir el caso de uso donde se use la tecnología Vehicle-to-Grid (V2G), donde los vehículos eléctricos pueden cargar o descargar sus baterías en los puntos de carga. En este caso, se tendría que tener en cuenta este factor de

relación bidireccional en los posibles ataques, como, por ejemplo, un ataque de descarga masiva de energía en diferentes puntos de cargas saturando el almacenamiento local de la Microgrid.

- **Red Blockchain**: red privada y permisionada de nodos P2P que forman una base de datos distribuida. Esta red almacena, a través de contratos inteligentes, los estados y transacciones realizadas en cada punto de carga. El objetivo de la Blockchain es garantizar una auditoría, trazabilidad y autenticación en el sistema, usando firmas con criptografía pública, y evitar la centralización por parte de una autoridad (tercera parte confiable).
- **Sistema de Monitorización**: permite a los administradores autorizados observar el estado actual del sistema. Este sistema puede alertar de posibles amenazas o errores, como puntos de cargas inactivos, ataques de fraude o fallos en el sistema.

6

Taxonomía de vulnerabilidades y amenazas

En esta sección, se pretende realizar un análisis exhaustivo sobre todas las vulnerabilidades y amenazas que se deben tener en cuenta a la hora de desarrollar un sistema ciberfísico, como una Microgrid basada en puntos de cargas. Para ello, se ha apoyado sobre el modelo de amenazas STRIDE, que logra identificar correctamente la mayoría de vulnerabilidades en los componentes del sistema SecCP.

6.1. Modelo de Amenazas

Para identificar correctamente las amenazas y vulnerabilidades del sistema, se usará el modelo de amenazas STRIDE propuesto por Microsoft [49]. Este modelo clasifica los diferentes tipos de amenazas en seis clases, como se puede ver en la tabla 5. El modelo STRIDE es ampliamente usado debido a su metodología, la cual es simple y sencilla de apicar. STRIDE analiza las vulnerabilidades en cada uno de los componentes del sistema y que podrían ser explotadas por un atacante para comprometer a todo el sistema.

La elección de este modelo de amenazas para el escenario que se expone en este TFG se debe a las siguientes razones: (1) analiza las vulnerabilidades de cada componente basándose en su conocimiento técnico, (2) es comprensible y cubre las áreas de la tríada CIA (del inglés, Confidenciality, Integrity y Availability) y los aspectos de seguridad de autenticación, autorización y no repudio; y, por último, (3) proporciona una alta comprensión del impacto de un componente vulnerable en el sistema.

Tabla 5: Modelo de amenazas STRIDE [49]

Propiedad	Amenaza	Definición
Autenticación	Suplantación de identidad (**S**poofing)	Acceso y uso ilícito de la información de autenticación de otro usuario
Integridad	Manipulación de datos (**T**ampering)	Modificar datos o código
No repudio	Rechazo (**R**epudiation)	Afirmar no haber realizado ninguna acción
Confidencialidad	Divulgación de la información (**I**nformation disclosure)	Exponer información a alguien no autorizado
Disponibilidad	Denegación de servicio (**D**oS)	Denegar o degradar servicios a usuarios
Autorización	Elevación de privilegios (**E**levation of Privilege)	Obtener capacidades sin su previa autorización

6.2. Metodología STRIDE

Debido a la escasez de una metodología estándar, se usará la siguiente metodología propuesta por Khan en el artículo [50]. Se trata de un procedimiento en cinco pasos, como se muestra en la figura 7, aplicando el modelo de amenzas STRIDE. A continuación, se detallan los pasos de la metodología:

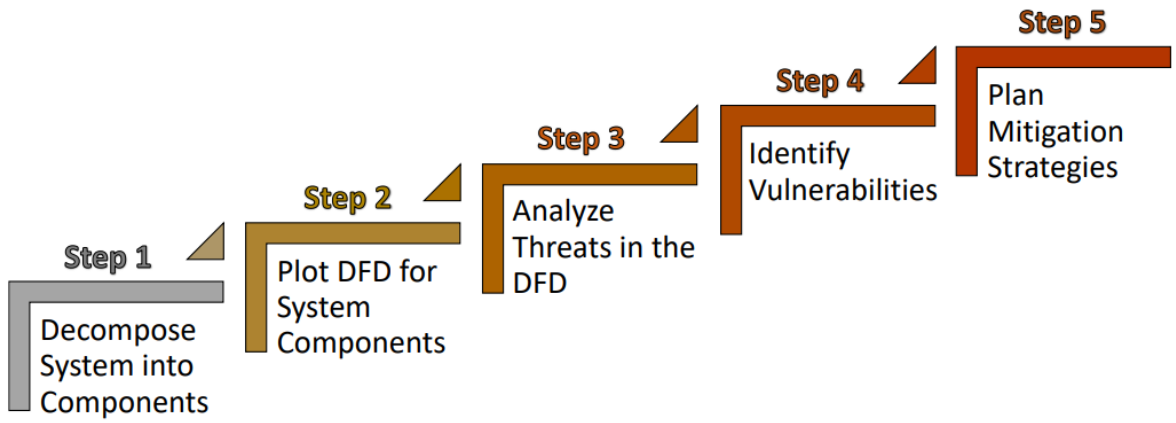


Figura 7: Metodología de un modelo de amenazas basado en STRIDE [50]

1. **Descomponer el sistema en componentes**: componentes lógicos o estructurales. Pueden ser procesos o elementos internos o externos que se comunican con el sistema. Este paso ya se ha realizado previamente en la sección 5.3

2. **Dibujar un Diagrama de Flujo de Datos (DFD)**: para cada uno de los componentes y visualizar sus funcionalidades dentro o fuera del sistema. El diagrama está compuesto principalmente por cuatro elementos estándares: (1) Entidad Externa (EE), (2) Proceso (P), (3) Flujo de datos (DF) y (4) Almacenamiento de datos (DS). Cada uno de estos elementos es suceptible a varias amenazas del modelo STRIDE (ver tabla 6). [1]

3. **Identificar amenazas STRIDE**: analizar todas las posibles amenazas más comunes y propensas para cada DFD.

4. **Identificar vulnerabilidades**: a partir de las amenazas detectadas, identificar las vulnerabilidades que se originan en cada una de ellas.

5. **Plan de estrategias de mitigación**: basadas en las vulnerabilidades descubiertas para aumentar la mayor eficacia de defensa posible.

El modelo de amenazas STRIDE puede ser adaptado desde dos puntos de vistas. Analizando el comportamiento y operaciones de cada elemento (**STRIDE-por-elemento**), o bien, observando las interacciones entre los componentes del sistema con tuplas (origen, destino, interacción) (**STRIDE-por-interacción**)

Tabla 6: Susceptibilidad de los elementos DFD a amenazas de STRIDE [50]

Elemento DFD	S	T	R	I	D	E
Entidad Externa (EE)	✓		✓			
Flujo de datos (DF)		✓		✓	✓	
Almacenamiento de datos (DS)		✓	✓	✓	✓	
Proceso (P)	✓	✓	✓	✓	✓	✓

[1] Debido a las recomendaciones de extensión de la memoria, los Diagramas de Flujo de Datos se han añadido como anexo.

6.3. Amenazas y vulnerabilidades

A continuación, se analizan los pasos 3 y 4 de la metodología STRIDE propuesta en la sección 6.2 (ver figura 7). Estos pasos consisten en identificar las amenazas y vulnerabilidades STRIDE en cada uno de los diagramas de flujo de los componentes del sistema SecCP, que se indican en la sección 5.3.

6.3.1. Consecuencias de amenazas

En primer lugar, para analizar las amenazas de cada elemento del sistema es necesario identificar las posibles intenciones de los atacantes. Para ello, se ha establecido en la tabla 7 las posibles consecuencias de las amenazas que pueden producirse en el sistema SecCP. Cada consecuencia de amenaza está identificada y referenciada con un código (Consecuencia de Amenaza (CA)).

Tabla 7: Posibles consecuencias de amenazas (CA) basadas en el conocimiento de SecCP

Código	Descripción	Riesgo
CA-1	Incapacidad de satisfacer la demanda de energía local de la Microgrid	R-2
CA-2	Incapacidad de controlar o configurar la Microgrid	R-1, R-2, R-3
CA-3	Denegación de comunicación con el servidor central	R-4
CA-4	Denegación de comunicación con los puntos de carga	R-4
CA-5	Denegación de comunicación con la red Blockchain	-
CA-6	Denegación de comunicación con la plataforma de autenticación	R-4
CA-7	Revelación de estados e información sensible del sistema	-
CA-8	Revelación de secretos de comunicación (claves de cifrado)	-
CA-9	Uso no autorizado de los puntos de carga	R-4
CA-10	Incapacidad de activar/desactivar los conectores	R-4
CA-11	Fraude sobre el consumo de energía	R-4
CA-12	Desincronización de los parámetros del sistema	R-1, R-2, R-3

Códigos de Riesgo (R): R-1 = lesiones humanas, R-2 = apagón (blackout), R-3 = daño de equipo, R-4 = daños económicos

6.3.2. Amenazas y vulnerabilidades STRIDE

A continuación, se describen las amenazas definidas por el modelo de amenazas STRIDE.

1. **Suplantación de identidad (S)**: si un atacante logra suplantar la identidad del servidor central a través de un acceso no autorizado podría deshabilitar, manipular y observar la comunicación con los puntos de carga (CA-3, CA-7, CA-9, y CA-10) y controlar las configuraciones de la Microgrid (CA-2, CA-11, CA-12). Por otro lado, los puntos de carga también pueden ser un objetivo de ataque debido a su alta exposición al público. Atacantes podrían acceder a los puntos de carga para denegar o controlar su servicio (CA-4, CA-9, CA-10, CA-12) o revelar información del estado del punto de carga (CA-7). Otra entidad, menos relevante de ataque, es la plataforma de autenticación o la aplicación móvil del usuario que permiten al atacante usar los servicios de activación/desactivación de puntos de carga de manera no autorizada (CA-9), ver datos sensibles sobre el usuario (CA-7) o denegar el servicio al usuario final (CA-6, CA-10). Al igual que las entidades externas, un atacante puede suplantar el comportamiento de un proceso del sistema para revelar información sensible (CA-7) o alterar las funcionalidades del sistema con propósito económico o dañino (CA-2, CA-10, CA-11, CA-12).
   Ejemplos de ataques que pueden llevarse a cabo en este ámbito son: ataques de interpretación o suplantación (*impersonation attacks*), *Address Resolution Protocol (ARP) spoofing*, ataques de acceso no autorizado, "*masquerading*" (atacante falsifica ser un usuario autorizado).

2. **Manipulación de datos (T)**: atacantes suelen interferir y violar la integridad de los mensajes de comunicación o durante un proceso. Un ataque muy común en sistemas críticos es la inyección de datos falsos (en inglés, False Data Injection (FDI)). El principal foco de ataque en este sistema es en el proceso de comunicación entre el sistema central y los puntos de carga (protocolo OCPP), o bien, los procesos de control y transacciones en la Microgrid. Eso se debe a su gran impacto como puede ser alterar los parámetros de la Microgrid (CA-2, CA-12), alterar los datos de consumo con fines económicos (CA-11) o activar/desactivar los conectores sin autorización (CA-9). Otro objetivo puede ser atacar a la autenticación y validación del usuario, con el objetivo de usar los servicios del sistema sin autorización (CA-9) o denegar el servicio al resto de usuarios (CA-10). Aun-

que la propia arquitectura de red Blockchain está protegida de manipulación de datos, gracias al conocido ataque del 51 %, no quita que atacantes manipulen los datos antes de ser introducidos en la Blockchain, es decir, alterar los datos de consumo y energía desde los puntos de cargas para que este cree una nueva transacción con estos datos manipulados. Esto permite que se puedan inyectar datos falsos en la base de datos distribuida de la red Blockchain, sin causar alertas.

Ejemplos de ataques que pueden llevarse a cabo en este ámbito son: ataques "Man in the Middle" (MitM), ataques de inyección de datos falsos (FDI), inyección de malware, manipulación de comandos, manipulación de código, ataques a la cadena de suministros, manipulación de base de datos.

3. **Rechazo (R)**: en este tipo de amenaza se intenta violar el requisito de seguridad no repudio, es decir, amenazas que eviten que entidades externas, flujo de datos o procesos garanticen realizar una determinada acción. En este sistema, la mayoría de elementos son susceptibles a este tipo de ataques. Los únicos elementos que registran acciones y datos son el proceso de autenticación del usuario, que recoge el identificador del usuario que solicita la acción, y el proceso de publicación de estado de los puntos de cargas, que recogen periódicamente los datos de consumo, potencia y configuraciones del sistema para guardarlos en la red Blockchain, firmando las transacciones.

4. **Divulgación de la información (I)**: atacantes pueden aprovechar brechas de información para obtener mayor conocimiento del sistema y preparar un ataque más elaborado y efectivo. Es importante garantizar la mayor confidencialidad posible para evitar amenazas persistente avanzadas (en inglés, Advanced Persistent Threat (APT)). Existen numerosos ataques que pueden proporcionar información sensible del sistema. El enlace de comunicación REST entre la aplicación móvil, la plataforma de autenticación y el servidor central y el protocolo OCPP entre el servidor central y los puntos de carga se encuentran cifrado a través del protocolo Transport Layer Security (TLS). Sin embargo, siguen siendo susceptible a ataques de robo de credenciales (CA-8) permitiendo a los atacantes observar todo el flujo de datos entre las distintas entidades (CA-7). Por último, un punto de ataque crítico y susceptibles a ataques de confidencialidad es el envío de estados y transacciones del sistema a la red Blockchain. Entre la información sensible, se

encuentra identificadores de usuario, datos de consumo de energía, potencia acumulada y localización de puntos de carga.

Ejemplos de ataques que pueden llevarse a cabo en este ámbito son: interceptación (*eavesdropping*), ataques Man in the Middle (MitM), análisis de tráfico, análisis de paquetes (*sniffing*) y descifrado de contraseñas (*password cracking*).

5. **Denegación de servicio (D)**: este tipo de amenazas inhabilita la disposición de servicio del sistema llegando a provocar severos impactos, como pueden ser impactos medioambientales o lesiones humanas tras la desconfiguración de parámetros, pérdida de control de la Microgrid y falta de datos de consumo, potencia, etc.; o incluso, impactos económicos al no satisfacer la demanda de los usuarios. Al tener el sistema segmentado en diferentes medios de comunicación y modos de operación, atacantes pueden denegar el servicio sobre uno de los componentes objetivo que más le interese considerando sus consecuencias. Si un atacante realiza una denegación de servicio sobre la comunicación con la plataforma de autenticación, sobre los anuncios BLE o los procesos de autenticación y activación/desactivación de conectores, consiguen que usuarios finales no puedan hacer uso de los puntos de carga (CA-6, CA-10). Por otro lado, si el punto de ataque es interferir en las transacciones OCPP entre servidor y puntos de carga y Microgrid, esto puede producir un mayor impacto provocando la pérdida de control en el sistema (CA-2, CA-3, CA-4, CA-10).

   Ejemplos de ataques que pueden llevarse a cabo en este ámbito son: "*jamming*" (ruido en el canal inalámbrico), ataques de repetición de paquetes (*replay attack*), *Denegación de Servicio (DoS)* o *Denegación de Servicio Distribuido (DDoS)*, inundación (*flooding*), privación del sueño, retraso de mensajes (*message relay*), "*black hole*" (eliminar todos los mensajes), "*grey hole*" (eliminar ciertos mensajes), daño físico y daño electromagnético.

6. **Elevación de privilegios (E)**: se puede interactuar a través de dos accesos: (1) como usuario a través del servidor de autenticación, donde sus únicas funcionalidades son activar/desactivar puntos de carga cercanos a su dispositivo móvil; y (2) como administrador a través del servidor central y el sistema de monitorización, con funcionalidades de control y configuración sobre la Microgrid y puntos de carga y el acceso a toda la información almacenada en la Blockchain. Si un atacante logra elevar sus privilegios

y accede al proceso de control y configuración de la Microgrid puede provocar graves consecuencias como hemos visto en las amenazas anteriores: incapacidad de configurar la Microgrid (CA-2), revelación de datos de configuración y estados del sistema (CA-7), alterar datos de consumo con objetivo de realizar fraude económico (CA-11), o incluso, desincronizar los parámetros del sistema (CA-12), poniendo en riesgo el equipo físico y vidas humanas.

Ejemplo de ataques que pueden llevarse a cabo en este ámbito son: descifrado de contraseñas, suplantación de identidad, manipulación de código, inyección de malware y APTs.

A partir de las amenazas identificadas en la sección anterior, se ha recogido, en la tabla 8, las vulnerabilidades más comunes y propensas de una Microgrid basada en puntos de cargas.

6.4. Estrategias de mitigación

En esta sección se propone dar una idea y esquema sobre las posibles mitigaciones a implementar en un sistema ciberfísico. A continuación, se enumeran la mayoría de técnicas de mitigación basadas en MITRE ATT&CK para ICS [8]. Se han clasificado las mitigaciones según su nivel de impacto en los requisitos de seguridad. Sin embargo, hay que tener en cuenta que la gran mayoría de ellas pueden influir en la mitigación de más de un requisito. [2]

1. **Disponibilidad**: aislamiento de aplicaciones y "Sandboxing" (M1048), prevención de ejecución (M1038), filtro de tráfico de red (M1037), listas de permisos de red (M0807), canales de comunicaciones fuera de banda (M0810), temporizadores de vigilancia (M0815) y anti-interferencias (Anti jamming).

2. **Integridad**: "Boot integrity" (M1046), firma de códigos (M1045), copia de seguridad de datos (M1053), prevención de pérdida de datos (M0803) y marcas de tiempo (timestamping).

3. **Confidencialidad**: cifrado del tráfico de red (M0808), cifrado de la información sensible (M1041), confidencialidad de la información operativa (M0809) e inspección SSL/TLS

[2] El código asociado entre paréntesis a una mitigación corresponde con el identificador asignado por "MITRE ATT&CK for ICS".

Tabla 8: Vulnerabilidades más comunes en Microgrids [13] y sus posibles amenazas STRIDE

Capa	Vulnerabilidad	S	T	R	I	D	E
Aplicación	Mala calidad del código	✓	✓	✓	✓		✓
	Gestión inadecuada de la configuración		✓				✓
	Gestión pobre de accesos y permisos	✓					✓
	Gestión inadecuada de rutas		✓		✓	✓	
	Control inadecuado de integridad de los datos		✓				
	Tratamiento inadecuado de errores	✓			✓	✓	✓
	Protección inadecuada de la base de datos	✓	✓	✓	✓	✓	✓
Red	Segmentación y segregación inadecuada	✓			✓	✓	✓
	Control de acceso inadecuado	✓	✓		✓		✓
	Prevención y detección de intrusiones débil	✓					✓
	Mecanismo de cifrado débil	✓	✓		✓	✓	✓
	Protección de datos sensibles inadecuado		✓		✓		
	Monitorización y auditoría de red inadecuada			✓			
	Trazabilidad de anomalías inadecuada			✓			
Física	Acceso físico no protegido	✓	✓		✓	✓	✓
	Configuración de dispositivos inapropiada	✓					✓
	Protección del firmware inadecuada	✓	✓		✓		✓
	Falta de hardware resistente a la manipulación		✓				
	Autenticación y autorización débil	✓					✓

(M1020).

4. **No repudio o trazabilidad**: auditoría (M1047), redundancia del servicio (M0811) e infraestructura de claves pública.

5. **Autenticación**: autenticidad de la comunicación (M0802), configuración Active Directory (M1035), autenticación de usuarios humanos (M0804), autenticación multifactor (M1032), política de contraseñas (M1027) y gestión de cuentas de usuario (M1018).

6. **Autorización**: gestión de acceso (M0801), política de uso de cuentas (M1036), limitación

de acceso a los recursos a través de la red (M1035), restricción de permisos de archivos y directorios (M1022) y principio de mínimos privilegios (PoLP).

7. **Otras mitigaciones a considerar**: antivirus/antimalware (M1049), segmentación de la red (M1030), protección contra exploits (M1050), prevención de intrusiones en la red (M1031), exploración de vulnerabilidades (M1016) y programa de inteligencia sobre amenazas (M109).

6.5. Conclusiones

La taxonomía de amenazas y vulnerabilidades, usando la metodología STRIDE, conlleva a que el componente crítico con mayor susceptibilidad a ataques es el propio punto de carga. Esto se debe a su alta exposición al público y a la gran cantidad de funcionalidades que tiene en el sistema. Un cibercriminal podría atacar al punto de carga de forma física o cibernética y vulnerar cualquiera de las 6 categorías mencionadas en STRIDE (tabla 5). Por ejemplo, el atacante podría robar la clave privada de un punto de carga y suplantar su identidad, obteniendo acceso así a la red Blockchain. O bien, manipular los datos de los estados antes de que se publiquen en la base de datos (inyección de datos falsos) o, incluso, realizar una denegación de servicio evitando su comunicación con el propio servidor central o la red Blockchain. También, podemos concluir que la mayoría de consecuencias conllevarían, sobre todo, a impactos económicos. Sin embargo, hay que priorizar de que existe un pequeño riesgo de lesiones humanas si se altera el control del sistema, provocando una inestabilidad energética en los recursos.

7

Red Blockchain permisionada

En esta sección, se estudia en profundidad el diseño de una red privada y permisionada usando las tecnologías Blockchain para proporcionar una solución de auditoría y almacenamiento distribuido en el sistema SecCP. En este TFG, se ha propuesto una solución de Blockchain permisionada desacoplada de los puntos de cargas, es decir, usando servidores externos como nodos de la red.

Otra posible solución, que se ha tenido en cuenta, es la posibilidad de que los propios puntos de cargas correspondan a nodos y formen entre ellas una red P2P (Blockchain permisionada acoplada en los puntos de cargas). Sin embargo, esta opción se descartó finalmente tras un análisis previo, debido a sus principales inconvenientes: los puntos de cargas son recursos con capacidad computacional muy limitada, que conllevaría a problemas de escalabilidad y rendimiento; y, además, su implementación es de una alta complejidad (se necesitaría al menos tres puntos de cargas totalmente funcionales).

7.1. Blockchain permisionada desacoplada de los puntos de carga

Finalmente, se ha decidido optar por la implementación de una red privada y permisionada formada por nodos desplegados en servidores. Estos nodos son los encargados de formar una red P2P, mantener la base de datos distribuida (cadenas de bloques) y crear nuevos bloques y transacciones. La información necesaria para crear nuevos bloques son enviados por los puntos de carga y el sistema de control central, como se muestra en la figura 8.

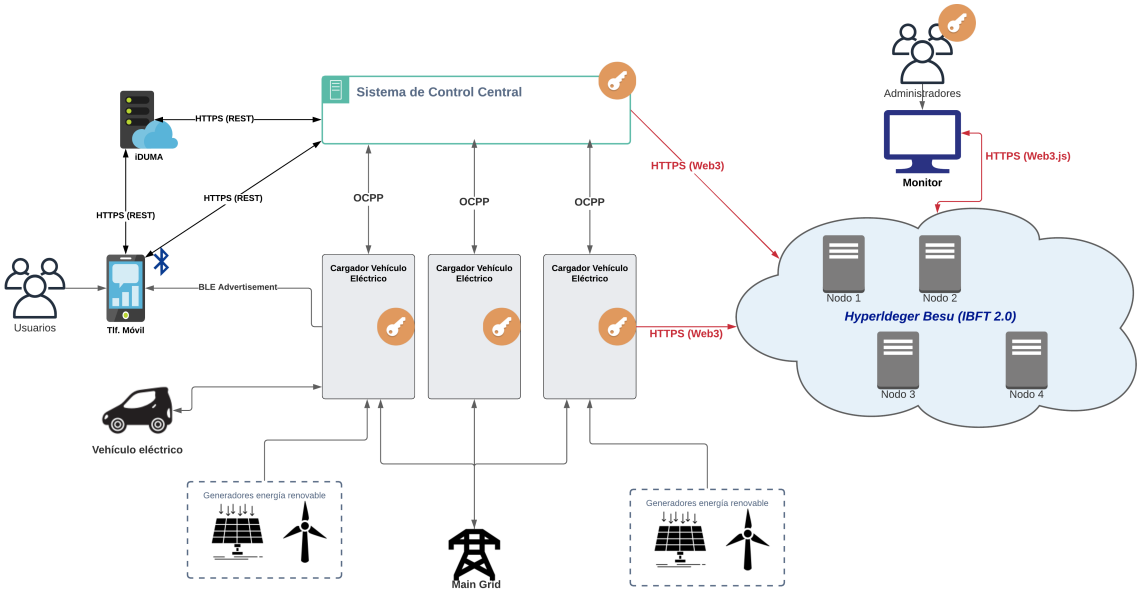


Figura 8: Diseño de red permisionada Blockchain en SecCP

Para controlar el acceso no autorizado y la autentiacción es necesario una gestión de cuentas de claves privadas y públicas. Cada punto de carga guarda, de forma segura, su propia clave privada que usa para firmar las transacciones en la red Blockchain, cada vez que sube los datos actualizados de sus conectores. De forma similar, el sistema de control también guarda una clave privada única para firmar las correspondientes transacciones de consumo que han realizado los usuarios. Por último, para garantizar un control de acceso, cada administrador también debe mantener en secreto una clave privada que lo identifique y permita el acceso al sistema de monitorización.

Una configuración ideal de esta red es establecer una vinculación de cada punto de carga con un único servidor o nodo. Esta vinculación puede ser dada gracias a un servicio descentralizado (como un Domain Name System (DNS)) que mantiene las asociaciones de pares (punto de carga, servidor). Por otro lado, para que esta red Blockchain tuviera mayor beneficio, sería necesario que los servidores sean totalmente independientes, por ejemplo, dividiendo los servidores y puntos de carga por zona, distrito, o instituciones. Esto permite a los administradores controlar y actuar rápidamente ante situaciones anómalas, evitando que un ataque hacia un punto de carga o servidor logre afectar al resto de dispositivos de la red.

7.2. Hyperledger Fabric vs Hyperldeger Besu

Para esta solución se pueden elegir varias tecnologías que son adecuadas para estas características. Una de ellas, con gran popularidad, es "Hyperledger Fabric". Según Hyperledger [51], "*Hyperledger Fabric pretende ser una base para desarrollar aplicaciones o soluciones con una arquitectura modular. Hyperledger Fabric permite que los componentes, como los servicios de consenso y membresía, sean plug-and-play. Su diseño modular y versátil satisface una amplia gama de casos de uso de la industria. Ofrece un enfoque único para el consenso que permite el rendimiento a escala mientras se preserva la privacidad.*"

Sin embargo, existen otros frameworks de Hyperledger que son ampliamente usados y con nuevas características. El último framework desarrollado corresponde con "Hyperledger Besu" [52]. Este framework se caracteriza por tener mayor flexibilidad, permitiendo seleccionar si se desea una red privada o pública, su algoritmo de consenso, o incluso, si se desea privacidad en la red usando Orion. En la tabla 9, se puede ver con más detalle las características de Hyperledger Besu con respecto a Hyperledger Fabric.

Tabla 9: Comparación entre Hyperledger Besu vs Hyperledger Fabric

Característica	**Hyperledger Besu**	**Hyperledger Fabric**
Descripción	Cliente Ethereum basado en Java que puede operar en una red pública o permisionada	Una red permisionada modular y versátil
Consenso	PoW, PoA e IBFT	CFT, BFT
Contratos Inteligentes	Solidity	Go, Java
Identidad digital	EhtSigner	CAs y MSP
Privacidad	Grupos disponibles usando el nodo Orion	Los canales incorporados se utilizan para mantener la confidencialidad
Moneda	Ether	No tiene

7.3. Despliegue y configuración

Se ha desplegado una red permisionada Blockchain con un total de cuatro nodos validadores en un único servidor dentro de la red eduroam. Para ello, se ha utilizado y seguido la guía de

instalación de la tecnología Hyperledger Besu [53]. Esta red permisionada se ha configurado de la siguiente manera:

- **Método de consenso**: IBFT 2.0 [54], método perteneciente a la categoría PoA y usado normalmente para redes privadas. En redes IBFT, las cuentas aprobadas, conocidas como validadores, validan las transacciones y los bloques. Estos se turnan para crear el siguiente bloque y, antes de insertarlo en la cadena, al menos el 66 % de los validadores deben firmar el bloque.

- **Número de nodos validadores**: cuatro, que corresponde con el mínimo requerido por el método de consenso IBFT para ser tolerante a fallos bizantinos. La tolerancia a fallos bizantinos es la capacidad de una red Blockchain de funcionar correctamente y alcanzar un consenso a pesar de que los nodos fallen o propaguen información incorrecta al resto de nodos.

- **Límite de gas**: gratuito (free gas). Esto permite que todos los bloques sean tratados por igual entre los nodos y no se consuma ningún coste al validarlos.

7.4. Smart Contracts

Según Javier Sebastián, responsable de Regulación Digital de DLT de BBVA Research, un contrato inteligente es definido como " *programa que puede definir reglas y consecuencias estrictas del mismo modo que lo haría un documento legal tradicional, pero a diferencia de los contratos tradicionales, también puede tomar información como 'input', procesarla según las reglas establecidas en el contrato y adoptar cualquier medida que se requiera como resultado de ello*".

Tras la configuración y el despliegue de los nodos en la red, se ha creado tres contratos inteligentes, donde se guardan los datos de consumo y energía y contiene las funciones necesarias para posibles búsquedas y consultas sobre ellas. Estos contratos inteligentes son programados en Solidity (>=0.7.0 <0.9.0) y compilados y desplegados en la red, gracias a la herramienta Truffle v5.3.1. A continuación, se definen los tres contratos inteligentes integrados en la red Blockchain:

- **ChargingStationState.sol**: contrato con una lista de estados para cada uno de los puntos de cargas. Este contrato, a través de un diccionario indexado por día, almacena los datos de energía y consumo enviados en tiempo real por cada punto de carga. Estos datos contiene el estado (activo/inactivo), potencia y energía consumida por cada uno de los conectores. Debido a que los puntos de cargas generan una nueva transacción sobre su estado durante cada minuto, este contrato presenta un elevado volumen de elementos con respecto a los otros contratos.

- **ChargingStationTransaction.sol**: almacena la colección de transacciones de carga realizadas en cada uno de los puntos de cargas. En cada transacción se indica los datos de energía total consumida, potencia media, tipo de conector, identificador de usuario, duración, hora de inicio y fin de la transacción. Estas transacciones son enviadas por el servidor central tras la finalización de una transacción de carga por parte de un usuario en algún punto de carga. Por tanto, este contrato mantiene un diccionario con los pares (dirección punto de carga, lista de transacciones de dicho punto de carga).

- **UserConsumptionManagement.sol**: de forma similar al contrato ChargingStationTransaction, este guarda las transacciones de carga realizadas por cada usuario. La diferencia principal recae en que este contiene un diccionario con los pares (identificador usuario, lista transacciones de dicho usuario). Este permite realizar fácilmente una trazabilidad y seguimiento de los consumos de cada usuario identificado en el sistema.

8

Simulación y datasets

Debido a la falta de tiempo y material (sobre todo, puntos de cargas y vehículos eléctricos), se ha desarrollado un pequeño programa en Python que simula el comportamiento real de varias estaciones de cargas distribuidas por Málaga. Este simulador, denominado como simulador Electric Vehicle Supply Equipment (EVSE), mantiene la interacción entre una lista de puntos de cargas, una lista de modelos de vehículos eléctricos y una lista de usuarios predefinidos para generar de forma periódica varios modelos de datos que son enviados a la red Blockchain. En la figura 9, se puede observar finalmente el diagrama de despliegue resultante del sistema SecCP.

8.1. Datasets y modelos de datos

Debido a la falta de datos reales sobre consumo de energía de vehículos eléctricos, se ha tenido que simular y generar los datos de entrenamiento teniendo en cuenta el comportamiento de los conectores y de sus curvas de carga.

Las curvas de carga lenta (hasta 3.7 kW) o semirápida (hasta 22 kW) presentan una curva lineal y constante, desde el comienzo de la transacción hasta que la batería se ha cargado por completo. Mientras tanto, para cargas rápidas (desde los 50 kW) las curvas de energía pueden variar según el modelo del vehículo. Estas suelen empezar con una potencia de carga constante hasta que la batería se ha cargado a un determinado porcentaje, donde comienza a disminuir. En la figura 10, se pueden ver las curvas de carga rápida para un Volkswagen ID.4 1st, y como varía entre un conector de 50 kW y un conector de más de 150 kW.

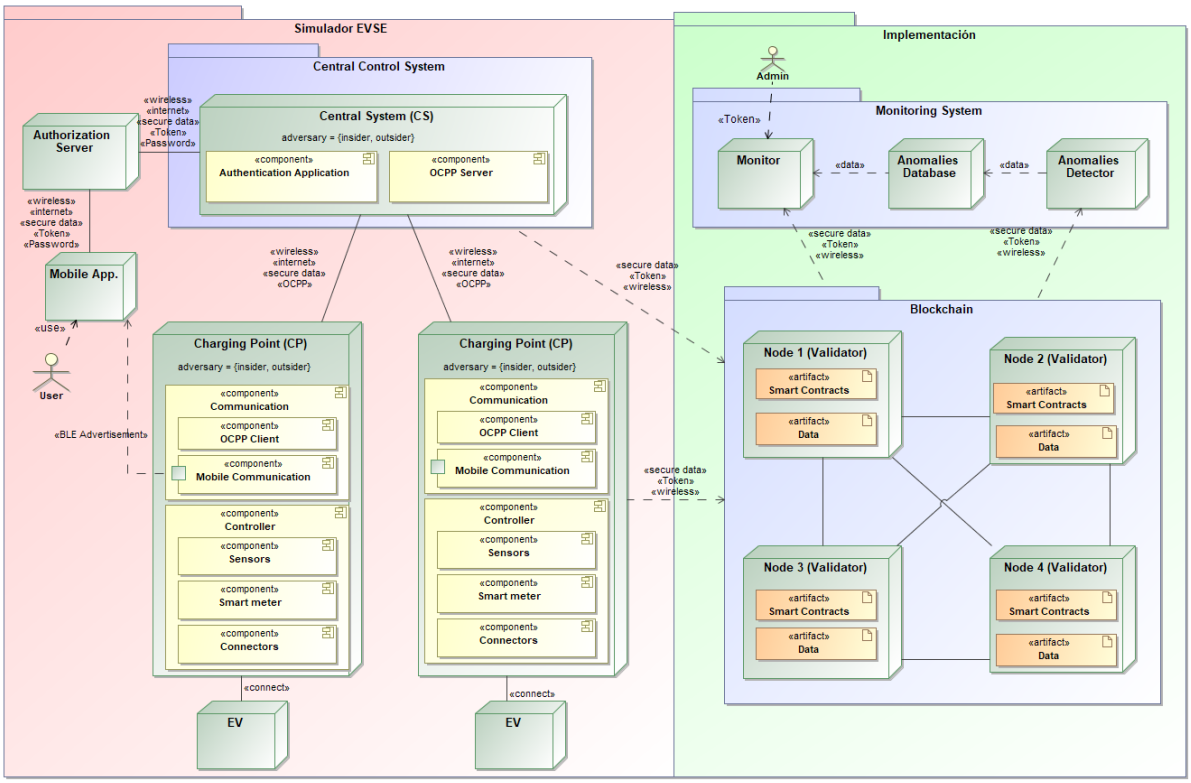


Figura 9: Diagrama de despliegue de SecCP

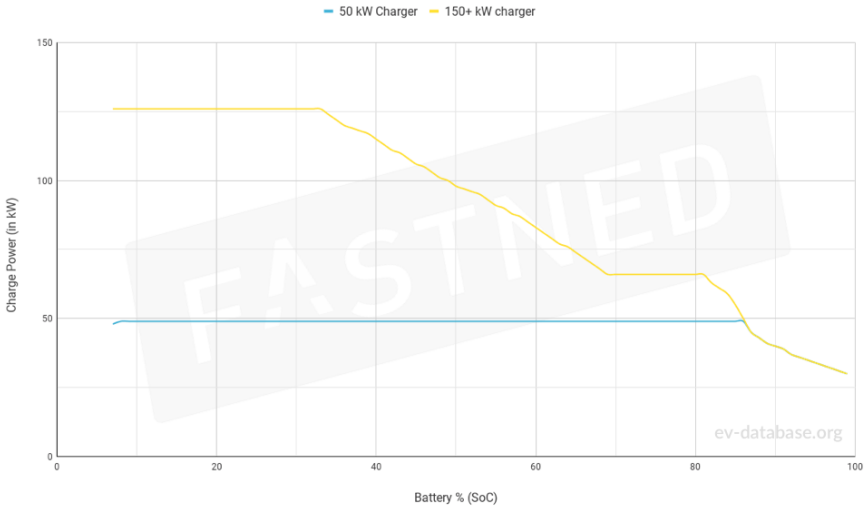


Figura 10: Curva de carga rápida en Volkswagen ID.4 1st [55]

Se ha utilizado la información ofrecida en [55] para generar datos simulados con las características de conectores de varios modelos de vehículos eléctricos. En este caso, para simplificar el proceso, no se ha considerado los conectores de carga rápida, sino solo los conectores de

carga lenta diseñados en el escenario SecCP: conector Schuko (tipo F) y conector Mennekes (tipo 2).

Por tanto, se ha implementado una pequeña aplicación de simulación en Python donde se recogen muestras de transacciones de diferentes modelos de vehículos eléctricos en diferentes puntos de carga y por diferentes usuarios (**EvseSimulator**). Estas muestras se dividen en dos modelos de datos que se usan como datasets para la detección de anomalías: "Estado de un punto de carga" (ChargingPointState) y "Transacción en un punto de carga" (ChargingPoint-Transaction) (ver figura 11). Además, dentro de la simulación, se ha implementado también diferentes modelos "anómalos" que corresponden a la simulación de posibles ataques o errores en el sistema.

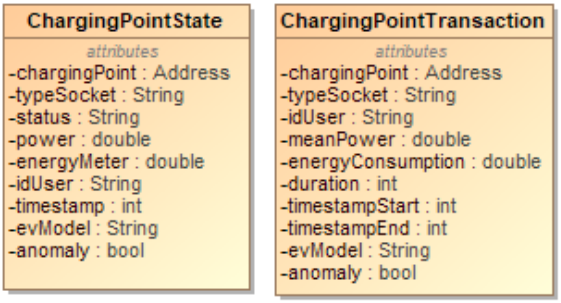


Figura 11: Modelos de datos generados por el simulador EVSE.

8.2. Modelos de ataques

Dentro de la simulación, se ha implementado también diferentes modelos "anómalos" que corresponden a la simulación de posibles ataques o errores en el sistema. El simulador puede elegir, con una pequeña probabilidad, un modelo anómalo al empezar una transacción de carga. Estos modelos anómalos simulan mayormente los ataques más comunes a los sistemas ciberfísicos y Microgrids: ataques de inyección de datos falsos (en inglés, False Data Injection Attacks (FDIAs)).

Dado el modelo [56]:

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{n}$$

donde $\mathbf{x}$ es un vector de potencias, $\mathbf{z}$ es un vector de medidas, $\mathbf{H}$ es una matriz Jacobiana

y **n** es un vector de ruido. El objetivo de la inyección de datos falsos es añadir un vector **a** sin que sea detectado. El resultado del modelo de ataque es [56]:

$$\mathbf{z_a} = \mathbf{Hx} + \mathbf{a} + \mathbf{n}$$

donde **a** es un vector no nulo.

En el proceso de simulación **EvseSimulator**, se ha implementado diferentes modelos de ataque, basados en el anterior, para simular posibles anomalías en el sistema.

1. **Anomalía en la duración de una transacción (DurationTransactionAnomaly):** se altera la duración de una transacción. El simulador inyecta un valor aleatorio, de tal manera que:

   $$a \in [-duracion/3, duracion/3]$$

   $$d_a = d + a$$

   donde *a* es el valor inyectado, *d* es la duración real en segundos y d_a es la duración anómala. Este ataque solo afecta al modelo de "Transacción en un punto de carga".

2. **Anomalía en la energía de una transacción (EnergyTransactionAnomaly):** se altera la energía total consumida en la transacción. Este ataque correspondería a un posible fraude por parte del usuario. El simulador inyecta un valor aleatorio, de tal manera que:

   $$a \in [energia/4, energia/2]$$

   $$n \in [-0.05, 0.05]$$

   $$e = p * t$$

   $$e_a = e - a + n$$

   donde *a* es el valor inyectado, *n* es el ruido, *e* es la energía consumida en kWh, *p* es la potencia en kW, *t* es el tiempo/duración en horas y e_a es la energía anómala. Este ataque solo afecta al modelo de "Transacción en un punto de carga".

3. **Anomalía en la potencia media de una transacción (MeanPowerTransactionAnomaly)**: se altera la potencia media obtenida durante la transacción. El simulador inyecta un valor aleatorio, de tal manera que:

$$a \in [-2, 5]$$

$$\bar{p_a} = \bar{p} + a$$

donde *a* es el valor inyectado, $\bar{p}$ es la potencia media y $\bar{p_a}$ es la potencia media anómala. Este ataque solo afecta al modelo de "Transacción en un punto de carga".

4. **Ruido en la potencia en un punto de carga (NoisedPowerStateAnomaly):** se altera la potencia actual calculada en los estados de un punto de carga. El simulador inyecta un valor aleatorio, de tal manera que:

$$n \in [-0.01, 0.01]$$

$$a \begin{cases} \in [-2, 2] & p < 5 \\ \in [-5, 5] & p < 10 \\ \in [-7, 7] & otherwise \end{cases}$$

$$p_a = p + a + n$$

donde *n* es el ruido, *a* es el valor inyectado, *p* es la potencia actual y p_a es la potencia anómala. Este ataque solo afecta al modelo de "Estado de un punto de carga" y solo al conector Mennekes.

5. **Anomalía en la potencia en un punto de carga (PowerStateAnomaly):** se altera la potencia continua de consumo en los estados de un punto de carga. En este caso, el modelo consume a una potencia no estándar de los que permite el conector Mennekes. El simulador altera la potencia, de tal manera que:

   - Si el conector es tipo Mennekes, entonces:

$$p_a \in \{3, 5.6, 8.8, 16, 18\}$$

     mientras que las potencias estándares en el conector Mennekes son:

$$p \in \{2.3, 3.7, 7.4, 11, 22\}$$

- Si el conector es tipo Schuko, entonces:

$$p_a \begin{cases} \in [1, 2.3] \\ \in [3.7, 10] \end{cases}$$

donde la potencia anómala puede ser de forma aleatoria menor o mayor que la potencia permitida en el conector $[2.3, 3.7]$.

6. **Consumo de potencia mientras se encuentra inactivo (ConsumptionInactiveStateAnomaly):** la potencia actual del estado de un punto es mayor que cero, es decir, un usuario está consumiendo energía, sin embargo, su estado actual es inactivo y no hay ningún usuario asignado al punto de carga. El simulador se comporta de la siguiente forma:

   - Si el conector Mennekes se encuentra inactivo, el simulador puede alterar la potencia actual (0 kW) con un valor aleatorio p_a, de tal forma que:

$$p_a \in [2.3, 22]$$

   - Si el conector Schuko se encuentra inactivo, el simulador puede alterar la potencia actual (0 kW) con un valor aleatorio p_a, de tal forma que:

$$p_a \in [2.3, 3.7]$$

9

Detección de anomalías

En esta sección, se muestran todos los aspectos considerados para el desarrollo del sistema de detección de anomalías. Primero, se indica las métricas e indicadores de evaluación que se han considerado. Luego, se muestra el procedimiento de aprendizaje así como las características extraídas para cada uno de los modelos de datos. Finalmente, se hace un análisis de los resultados usando gráficas y tablas, para posteriormente concluir con qué método de aprendizaje es el más adecuado.

9.1. Métricas de rendimiento

Existen diferentes metodologías de evaluación y, por tanto, de decisión sobre un algoritmo de aprendizaje según los intereses del individuo.

1. **Matriz de confusión**: herramienta que permite la visualización del rendimiento de un algoritmo de aprendizaje supervisado. Esta matriz representa gráficamente el número de predicciones correctas (verdaderos positivos (TP) y verdaderos negativos (TN)) y el número de incorrectas predicciones (falsos positivos (FP) y falsos negativos (FN)). A partir de esta matriz, se pueden obtener las siguientes métricas o indicadores:

    - Exactitud (accuracy): capacidad de un instrumento de acercarse al valor de la magnitud real. Está relacionada con el sesgo de una estimación. Tiene en cuenta todos los valores de la matriz de confusión. Esta métrica no es buena con datasets no balanceados, como es en este caso.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precisión (precision): Está relacionada con la dispersión del conjunto de valores. Representa la calidad de detección de elementos positivos, es decir, el número de verdaderos positivos con respecto al total de elementos identificados como positivos.

$$Precision = \frac{TP}{TP + FP}$$

- Sensibilidad (recall): es el ratio de positivos detectados en el dataset, es decir, de todos los positivos reales, cual es el ratio de positivos detectados correctamente. Nos da información sobre el rendimiento de un clasificador con respecto a falsos negativos.

$$Recall = \frac{TP}{TP + FN}$$

- Puntuación F1 (F1 score): media harmónica de la precisión y la sensibilidad. Es posible ajustar el valor F para dar más importancia a una de las dos métricas, como son la puntuación F0.5 y la puntuación F2.

$$F1_{Score} = \frac{2 * (Recall * Precision)}{Recall + Precision}$$

2. **Curva Característica Operativa del Receptor (en inglés, Receiver Operating Characteristic) (ROC)**: se trata de una representación gráfica de la razón de verdaderos positivos (VPR) frente a la razón de falsos positivos (FPR), según varía el umbral de discriminación. Esta herramienta ayuda a seleccionar los modelos posiblementemente óptimos independientemente del coste de la distribución. Los puntos por encima de la diagonal representan buenos resultados de predicción, siendo lo óptimo la esquina superior izquierda (punto (0,1)). El Área Bajo la Curva (en inglés, Area Under Curve (AUC)) suele ser el indicador más utilizado en ROC.

$$AUC_{ROC} = \int_0^1 ROC(t)\, dt$$

3. **Curva Precision-Recall (PR)**: representación gráfica entre las métricas de precisión y sensibilidad. Debido a que estas dos métricas se encuentran relacionadas inversamente (al aumentar la precisión, disminuye la sensibilidad), esta curva nos permite observar desde que punto de sensibilidad comienza a degradar la precisión [57]. En este caso, el método óptimo de predicción se encontraría en la esquina superior derecha (punto (1,1)).

Esta métrica es recomendable cuando el dataset no es balanceado (número de positivos ocurren con poca frecuencia). Uno de los indicadores es la precisión media (AP), que sirve para calcular el Área Bajo la Curva.

$$AUC_{PR} = \int_0^1 PR(t)\, dt$$

9.2. Características

A partir de los datasets, se extraen las características que más se adecuen en cada caso. Tras extraer las características, las medidas son estandarizadas (normalizadas) y, finalmente, se reducen las dimensiones a dos componentes usando el método Principal Component Analysis (PCA).

Para seleccionar correctamente las características se han tenido en cuenta la correlación entre ellas (observando el mapa de calor), seleccionando así características independientes, y los ataques más propensos.

Además, se ha divido ambos modelos, a su vez, en varios submodelos, con el objetivo de analizar si es más adecuado usar un modelo general independientemente del tipo de conector, o bien, un modelo para cada tipo de conector. Estos submodelos son: *Full* (modelo general), *Mennekes* (conector tipo 2) y *Schuko* (conector tipo F).

- **Estado de un punto de carga**: para este modelo de datos, se han seleccionado las características: (1) potencia actual del conector, (2) estado binarizado del conector (0, inactivo; 1, activo) y (3) tipo de conector, solo para el modelo genérico (Full),

- **Transacción en un punto de carga**: este modelo recoge las características: (1) duración total de la transacción, (2) potencia media, (3) energía total consumida y (4) tipo de conector, solo para el modelo genérico (Full).

9.3. Resultados

A continuación, en la tabla 10 se muestra la lista de los diferentes métodos y librerías que se han empleado durante el desarrollo. Sin embargo, solo cuatro de ellos (SVC, RFC, DTC y MLPC) presentan buenos resultados de predicción para los datasets definidos en la sección 8.1.

Tabla 10: Métodos de aprendizaje probados para la detección de anomalías

Categoría	Tipo	Método	Librería
Supervised Learning	Support Vector Machine	Support Vector Clasification (**SVC**)*	scikit-learn
Supervised Learning	Ensembles-based	Random Forest Classifier (**RFC**)*	scikit-learn
Supervised Learning	Ensembes based	Isolation Forest (**IForest**)	scikit-learn
Supervised Learning	Tree-based	Decision Tree Classifier (**DTC**)*	scikit-learn
Semi-Supervised Learning	Proximity based	Semi-Supervised k-Nearest Neighbor of Outliers (**SSKNNO**)	anomatools
Unsupervised Learning	Proximity based	Local Outlier Factor (**LOF**)	pyod
Unsupervised Learning	Proximity based	k-Nearest Neighbor (**KNN**)	pyod
Unsupervised Learning	Ensembles based	Lightweight On-line Detector of Anomalies (**LODA**)	pyod
Unsupervised Learning	Probabilistic	Fast Angle-based Outlier Detector (**FastABOD**)	pyod
Unsupervised Learning	Linear Model	One-Class Support Vector Machines (**OCSVM**)	pyod
Deep Learning	Neural Networks	**Auto Encoder**	pyod
Deep Learning	Neural Networks	Variational Auto Encoder (**VAE**)	pyod
Deep Learning	Neural Networks	Multi-Layer Perceptron Classifier (**MLPC**)*	scikit-learn

* Métodos de aprendizaje con mejores resultados

Para obtener los resultados en cada uno de los modelos, se ha usado el siguiente procedimiento de aprendizaje:

1. **Extracción de las características:** se ha filtrado del dataset las columnas correspondiente a las características del modelo.

2. **Split:** se ha dividido las muestras con un porcentaje de 0.7 para entrenamiento (x_train, y_train) y 0.3 para test (x_test, y_test). Para ello, se ha usado un conjunto con un total de 1 millón de muestras para cada método, excepto en SVC, que se ha usado un conjunto

de 100.000 muestras debido a su elevado coste temporal y computacional.

3. **Preprocesamiento:** se han normalizado las muestras (x_train, x_test), usando el modelo StandardScaler de la librería scikit-learn (valores entre 0 y 1).

4. **Descomposición:** reducción de las dimensiones de las características a 2 componentes, usando el modelo PCA de la librería scikit-learn.

5. **Entrenamiento:** se instancia el algoritmo machine learning correspondiente y se ajusta (entrena) con el anterior conjunto de muestras (x_train).

6. **Predicción:** tras el entrenamiento, se predice el subconjunto de test (x_test), obteniendo las etiquetas de predicción (y_test_pred).

7. **Resultados:** finalmente, se genera una matriz de confusión, según los resultados esperados (y_test) y obtenidos por el algoritmo (y_test_pred).

9.4. Análisis de resultados

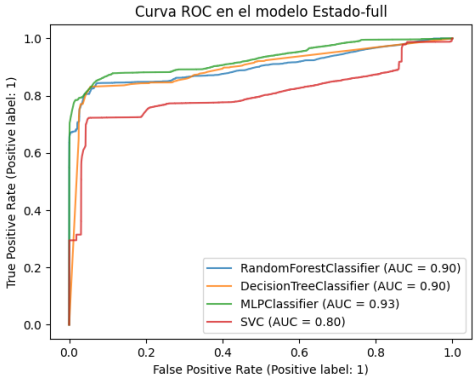
Tras entrenar cada modelo y recoger los parámetros de las matrices de confusiones, se evalúan cada uno de los modelos (SVC, RFC, DTC y MLPC) y así observar qué método de predicción es el óptimo en cada uno de los modelos de datos. Para ello, se usan las métricas de rendimiento explicadas en la sección 9.1.

9.4.1. Curva ROC

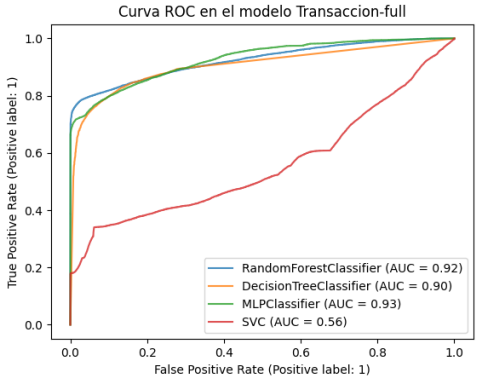
En este apartado, se visualizan las gráficas con la curva ROC para cada uno de los modelos. Estas son generadas a partir de la función *plot_roc_curve* de la librería Scikit-learn y muestran, en una misma imagen, la relación entre el ratio de falsos positivos y el ratio de verdaderos positivos de los diferentes métodos de aprendizaje, calculando así el Área Bajo la Curva (AUC) en cada uno de ellos. Estas gráficas se pueden ver en la figura 12.

9.4.2. Curva PR

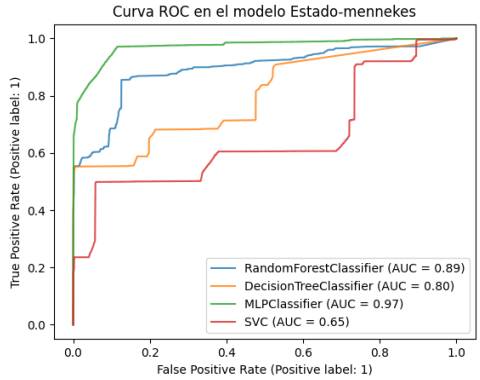
De forma similar a la sección 9.4.1, con la función *plot_precision_recall_curve* de la librería Scikit-learn se pueden generar gráficas con las curvas PR para cada uno de los modelos



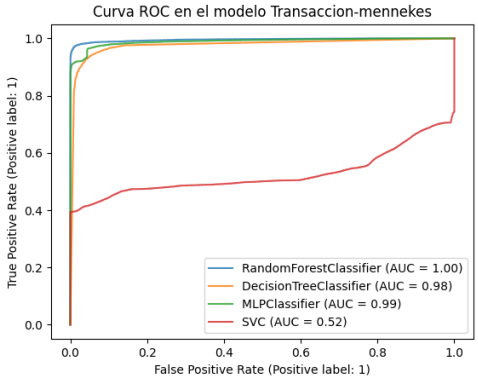
(a) Curva ROC en el modelo Estado-full



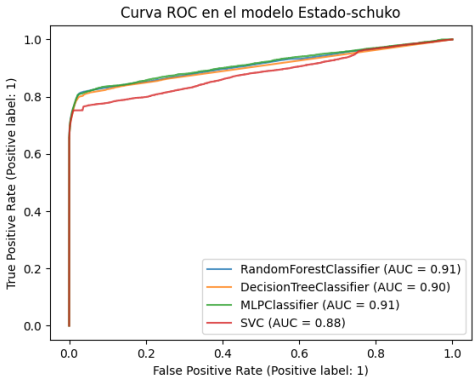
(b) Curva ROC en el modelo Transaction-full



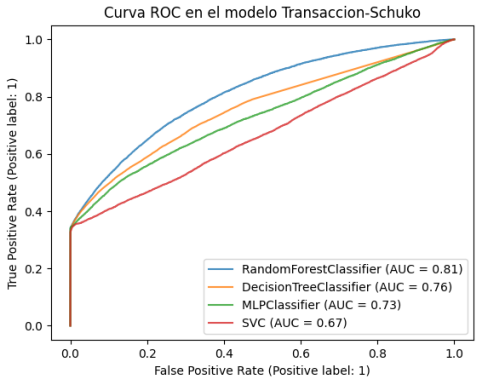
(c) Curva ROC en el modelo Estado-mennekes



(d) Curva ROC en el modelo Transaction-mennekes



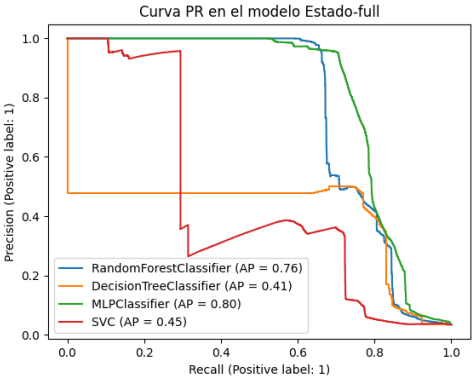
(e) Curva ROC en el modelo Estado-schuko



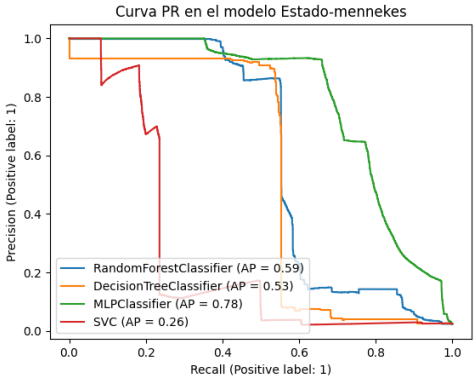
(f) Curva ROC en el modelo Transaction-schuko

Figura 12: Curvas ROC de los métodos de predicción

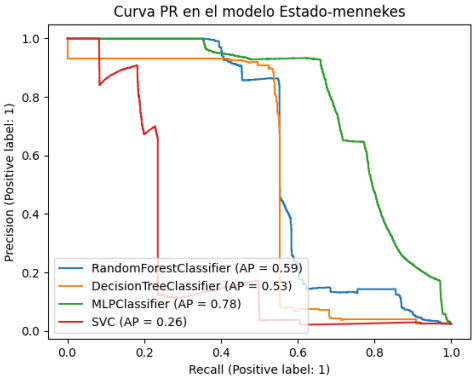
de datos. Esta curva permite comparar visualmente las métricas de sensibilidad y precisión para cada método de aprendizaje, calculando así sus precisiones media (AP). Estas gráficas se encuentran reflejadas en la figura 13.



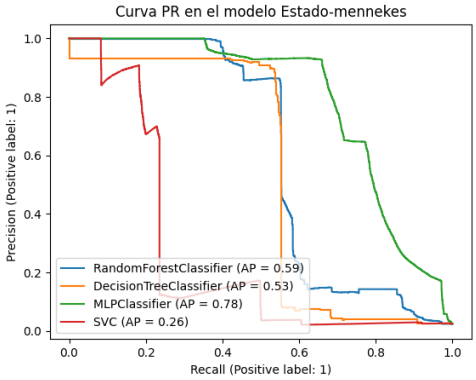
(a) Curva PR en el modelo Estado-Full



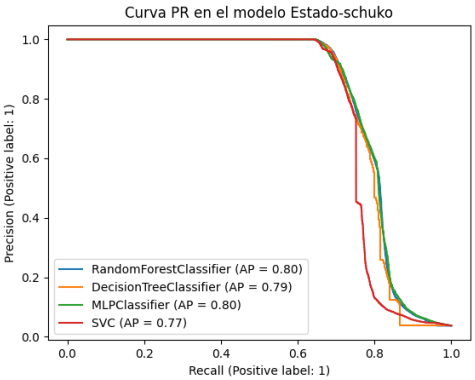
(b) Curva PR en el modelo Transaction-Full



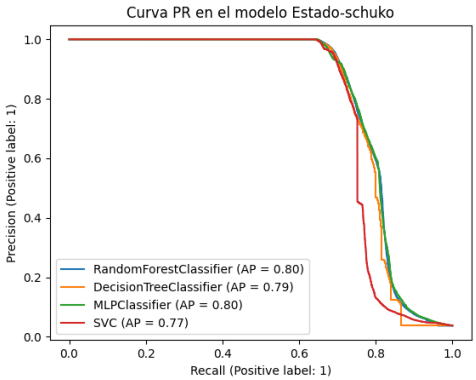
(c) Curva PR en el modelo Estado-Mennekes



(d) Curva PR en el modelo Transaction-Mennekes



(e) Curva PR en el modelo Estado-Schuko



(f) Curva PR en el modelo Transaction-Schuko

Figura 13: Curvas PR de los métodos de predicción

9.4.3. Resumen de métricas

A continuación, se reflejan en las tablas 11 y 12 todos los indicadores de evaluación para cada uno de los modelos y métodos. Además, se han coloreado los métodos con mejores resultados de predicción, independientemente del modelo. Estos métodos serán posteriormente discutidos en la sección de conclusiones, para elegir así cuál de ellos es el óptimo en cada caso.

Tabla 11: Resumen de los resultados obtenidos en el modelo "Estado"

Método	Modelo	Accuracy	Precision	Recall	$F1_{Score}$	AUC_{ROC}	AUC_{PR}
RFC*	Full	0.9900	0.97390	0.7273	0.8327	0.90	0.76
	Mennekes	0.9936	0.9672	0.7566	0.8490	0.89	0.59
	Schuko	0.9879	0.9684	0.6924	0.8074	0.91	0.80
DTC	Full	0.9900	0.9735	0.7276	0.8328	0.90	0.41
	Mennekes	0.9936	0.9672	0.7566	0.8490	0.80	0.53
	Schuko	0.9879	0.9646	0.6942	0.8074	0.90	0.79
MLPC*	Full	0.9862	0.9561	0.6231	0.7545	0.93	0.80
	Mennekes	0.9898	0.8676	0.6739	0.7586	0.97	0.78
	Schuko	0.9871	0.9441	0.6868	0.7952	0.91	0.80
SVC	Full	0.9672	0.9565	0.1686	0.2867	0.80	0.45
	Mennekes	0.9864	1	0.1108	0.1996	0.65	0.26
	Schuko	0.9857	1	0.4850	0.6532	0.88	0.77

* Métodos de predicción con mejores resultados

Tabla 12: Resumen de los resultados obtenidos en el modelo “Transacción”

Método	Modelo	Accuracy	Precision	Recall	$F1_{Score}$	AUC_{ROC}	AUC_{PR}
RFC*	Full	0.9837	0.9978	0.5860	0.7384	0.92	0.82
	Mennekes	0.9938	1	0.8474	0.9174	1.00	0.90
	Schuko	0.9748	0.9995	0.3324	0.4989	0.81	0.46
DTC	Full	0.9836	0.9436	0.6175	0.7465	.90	0.60
	Mennekes	0.9929	0.9734	0.8470	0.9058	0.98	0.40
	Schuko	0.9747	0.9912	0.3347	0.5005	0.76	0.43
MLPC*	Full	0.9833	1	0.5728	0.7284	0.93	0.77
	Mennekes	0.9934	1	0.8337	0.9093	0.99	0.94
	Schuko	0.9751	1	0.3398	0.5072	0.73	0.43
SVC	Full	0.9646	1	0.0741	0.1379	0.56	0.26
	Mennekes	0.9691	1	0.1660	0.2847	0.52	0.49
	Schuko	0.9730	1	0.3211	0.4862	0.67	0.40

*Métodos de predicción con mejores resultados

9.5. Conclusiones

Observando los resultados de las tablas 11 y 12 de la sección 9.4.3, podemos extraer varias conclusiones para ambos modelos.

La primera cuestión a resolver es ver si es más adecuado usar un modelo general (modelo *Full*) independientemente del tipo de conector, o bien, usar un modelo de predicción para cada uno de los conectores (usar el modelo *Mennekes*, para las transacciones y estados de los conectores tipo 2; y el modelo *Schuko*, para las transacciones y estados de los conectores tipo F).

En este caso, usar un modelo para cada conector parece ser la mejor opción. Esto se debe a que cada modelo se adaptaría mejor al comportamiento del conector, que si se usara un modelo general. Por ejemplo, el modelo Mennekes presenta un gran índice en todos los indicadores (tablas 11 y 12), debido a que sus curvas de carga son lineales y con una potencia preestable-

cida (2.3, 3.7, 7.4, 11 o 22 kW). Sin embargo, el conector Schuko presenta peores resultados a causa de que, aunque su curva de carga sigue siendo lineal, no tiene una potencia preestablecida. La potencia del conector Schuko puede variar entre 2.3 y 3.7 kW (el simulador genera una potencia aleatoria entre estos dos valores). Además, la opción de tener un detector por conector tiene una mayor escalabilidad que usar un modelo genérico (full). Si en un futuro se desea añadir, por ejemplo, un conector de carga rápida donde su curva no es lineal (como se ve en la figura 10), se tendría que crear un nuevo dataset específico para este modelo y entrenar un nuevo método de predicción (que no tiene por qué ser el mismo) con este dataset. Por lo contrario, si se usara un modelo genérico, sería necesario crear un nuevo dataset que incluya muestras de todos los tipos de conectores actuales más del nuevo modelo a añadir, y volver a entrenar el método de predicción.

La segunda cuestión a tener en cuenta es observar qué métrica de evaluación es la más adecuada para la elección de los métodos de predicción óptimos. Al tratarse de datasets no balanceados (pocas anomalías), el indicador de *exactitud* no proporciona información útil. Si se desea buscar un método de predicción lo mayor preciso posible, donde el sistema de monitorización no suela lanzar falsas alarmas, es necesario centrarse en el indicador de *precisión* y la *curva PR*. Por otro lado, si se desea encontrar el mayor ratio de anomalías posible, aunque conlleve a falsas alarmas, es más adecuado centrarse en el indicador de *sensibilidad* y la *curva ROC*. En el caso de que se desee un equilibrio entre precisión y sensibilidad, se debería usar los indicadores de *puntuación F1* (F1 Score) o la *curva PR*.

El modelo "Estado de un punto de carga" se encuentra más relacionado con las métricas de precisión. Este modelo genera un gran volumen de datos. Durante cada minuto, se genera un nuevo estado para cada conector en todos los puntos de carga. En este modelo no sería de mucha preocupación que en algún caso no se detectase una anomalía (falso negativo), ya que puede que en los siguientes estados, durante la misma transacción, se detecte otra anomalía provocada por el mismo ataque o error. Sin embargo, sería de mayor preocupación si el detector de anomalías es propenso a falsos positivos. Esto conllevaría a que el sistema de monitorización comience a mostrar continuamente falsas alarmas.

Por otro lado, en el modelo "transacción en un punto de carga" ocurre lo contrario. En esta ocasión es adecuado que el sistema detecte el mayor número de anomalías posible, aunque esto provoque la detección de falsos positivos. Es muy poco frecuente que se generen nuevas muestras de transacciones de carga, donde cada una de ellas son totalmente independientes. Por ello, es necesario que el ratio de anomalías detectadas sea lo mayor posible (sensibilidad).

Por último, queda por seleccionar el método de aprendizaje óptimo para la detección de anomalías en cada modelo (y para cada submodelo), teniendo en cuenta las consideraciones anteriores. Para ambos modelos, los métodos óptimos de predicción son los métodos *Random Forest Classifier* (RFC) y *Multi-Layer Perceptron Classifier* (MLPC). En la tabla 11, se puede observar que para el conector Mennekes, los indicadores de precisión y F1 Score de RFC son ampliamente superiores al resto de métodos (aunque el área bajo la curva PR (AUC_{PR}) es inferior que MLPC) y, por tanto, este es la mejor opción para el modelo "estado de un punto de carga". Por otro lado, en el conector Schuko, tanto RFC como MLPC obtienen resultados muy similares, siendo RFC ligeramente superior a MLPC.

De forma similar para el modelo "transacción en un punto de carga", en la tabla 12, vemos que los indicadores del conector Mennekes: área bajo la curva ROC (AUC_{ROC}) y las métricas de sensibilidad (Recall) y F1 Score de RFC; son ligeramente superior al método MLPC. En el caso del conector Schuko, el método de aprendizaje profundo MLPC es superficialmente superior y, por tanto, el método óptimo para este caso.

Podemos concluir, por tanto, que el conector Mennekes obtiene buenos resultados usando el método de aprendizaje supervisado **Random Forest Classifier**, debido a su comportamiento lógico con potencias preestablecidas y curvas de carga líneales. Mientras que el conector Schuko, muestra ligeramente mejores resultados aplicando el aprendizaje profundo **Multilayer Perceptron Classifier**. Esto se debe a que no presenta un comportamiento lógico, no hay una potencia predeterminada, sino que sigue un cierto patrón con potencias no superiores a los 3.7 kW (16 A).

Como se puede ver en la tabla 13, para cada modelo o conector, que se añada en un futuro en el sistema, es necesario realizar un estudio y análisis previo para seleccionar su algoritmo

Tabla 13: Clasificación de los métodos de detección óptimos según el tipo de conector

Modelo	Conector	Objetivo	Método óptimo
Estado	Mennekes	Precisión	Random Forest Classifier
Estado	Schuko	Precisión	Multi-Layer Perceptron Classifier
Transacción	Mennekes	Sensibilidad	Random Forest Classifier
Transacción	Schuko	Sensibilidad	Multi-Layer Perceptron Classifier

de detección más adecuado, que puede diferir con respecto a los analizados en este trabajo. Por tanto, el sistema de detección estaría formado por un conjunto de métodos de detección que dependen en todo caso del tipo de conector.

10

Sistema de monitorización

Por último, esta sección indica cómo se ha desarrollado el sistema de monitorización y qué funcionalidades puede encontrar un usuario autorizado. Primero, tras guardar los modelos de detección aprendidos y elegidos en la sección anterior (sección 9), se ha desarrollado un proceso que permite la consulta y análisis (predicción) de los últimos estados y transacciones en la red Blockchain. Si este proceso encuentra una anomalía, guarda la información necesaria para su identificación en una base de datos local. Finalmente, se ha implementado una página web de monitorización que consulta periódicamente (tras varios segundos) la información guardada tanto en la Blockchain como en la base de datos de anomalías, alertando en todo momento de posibles ataques o errores del sistema. A continuación, se explica detalladamente las funcionalidades de estos dos componentes.

10.1. Proceso de detección de anomalías

Dos programas software, implementados en Python, que ejecutan periódicamente, cada 5 y 10 minutos respectivamente, una consulta a la Blockchain de los últimos estados y transacciones respectivamente. Tras la consulta, se procede a usar el modelo de detección de anomalía, según el conector, y si el método de predicción devuelve un valor positivo (anomalía), estos estados o transacciones anómalas son almacenados en una base de datos no relacional local, MongoDB. En esta base de datos, solo se guarda la información necesaria para su correcta identificación desde la base de datos distribuida de la red Blockchain.

10.2. Interfaz gráfica del sistema de monitorización

Este corresponde con la parte visual y objetivo final del TFG: añadir una primera línea de conciencia situacional. Se ha implementado un servicio web usando principalmente las tecnologías React.js (frontend) y Django (backend). El servidor web consta de tres páginas web: una página principal, que corresponde con el monitor, y dos páginas de búsqueda sobre usuarios y puntos de cargas, que corresponden al sistema de trazabilidad en la red Blockchain. A continuación, se detalla las principales funcionalidades en cada una de ellas (para ver más información sobre su uso, consultar el manual de usuario en el apéndice A)

El monitor muestra de forma visual, a través de gráficas y mapas, el estado actual del sistema, así como alertas de todas las anomalías encontradas durante las últimas horas. Para ello, este se caracteriza por actualizar la información cada 30 segundos, mostrando en tiempo real el avance de salud de todos los puntos de carga. Además, esta página presenta funcionalidades interactivas y personalizables por el administrador, donde se permite entre otros: seleccionar un punto de carga a visualizar, indicar el rango de número de horas a consultar, mostrar información detallada sobre un estado anómalo o transacción anómala e interactuar con las gráficas.

Por otra parte, el sistema de trazabilidad se compone por las páginas de usuarios y puntos de cargas, donde se permite buscar y seleccionar un usuario o estación de carga específico. Al seleccionarlo, se puede ver inicialmente su información detallada, seguido del último estado o transacción registrada. Finalmente, en el pie de la página, se muestra un buscador y filtro, donde el usuario puede indicar un rango de fechas. Este buscador permite consultar, a través de gráficas, todos los estados y transacciones en las fechas indicadas.

11

Pruebas y validaciones

A continuación, se muestra de forma resumida las baterías de pruebas y validaciones sobre los requisitos definidos en la sección 5.1. Para ello se usa una de las herramientas recomendada por la guía "*Project Management Body of Knowledge*" (PMBOK), que es la matriz de trazabilidad de requisitos (en inglés, *Requirement Traceability Matrix (RTM)*). Se trata de un documento que relaciona y traza cada requisito con los casos de prueba (CP). Para ello, primero es necesario realizar una batería de pruebas, que es mostrada y definida en la tabla 14. Tras ejecutar las pruebas, se procede a trazar cada requisito con cada caso de prueba, según su relación. Esta matriz de trazabilidad es mostrada de forma gráfica en la tabla 15. Los requisitos coloreados en amarillo corresponden a requisitos no probados por ningún caso de prueba y pendientes de validar. Mientras que los requisitos coloreados en rojo son los requisitos que han fallado las pruebas y, por tanto, no se han cumplido en el proyecto.

Tabla 14: Batería de casos de prueba (CP)

Nº	Caso de prueba	Pasos	Resultado esperado
CP01	API REST-JSON Besu - Peers	1. Abrir Postman 2. Abrir collection Besu JSON-RPC API 3. Buscar método "net_peerCount" 4. Lanzar POST en "http://192.168.43.110:8545"	"result":"0x3"
CP02	Acceso Grafana	1. Abrir navegador web (red eduroam) 2. Acceder "http://192.168.43.110:3000" 3. Visualizar tablero Besu Overview	Gráficas con interacción (figura 24)
CP03	API REST SecCP - Estados	1. Abrir Postman 2. Abrir collection SecureChargingPoints 3. Buscar método "search states" 4. Lanzar GET con "http://192.168.43.110:8000/api/chargingStation/state/search ?account=0x04e66BB1Df9066C79f311A3c80a328eDCCE2A308 &timestampFirst=1624550586&timestampEnd=1624550886"	JSON con array de estados
CP04	API REST SecCP - Transacciones	1. Abrir Postman 2. Abrir collection SecureChargingPoints 3. Buscar método "search transactions" 4. Lanzar GET con "http://192.168.43.110:8000/api/chargingStation/transaction/search ?account=0x04e66BB1Df9066C79f311A3c80a328eDCCE2A308 &timestampFirst=1619061580&timestampEnd=1629061580"	Array de transacciones
CP05	API REST SecCP - Puntos de carga	1. Abrir Postman 2. Abrir collection SecureChargingPoints 3. Buscar método "ChargingStationList" 4. Lanzar GET con "http://192.168.43.110:8000/api/chargingStation/list"	Array de direcciones públicas (*address*)
CP06	API REST SecCP - Usuarios	1. Abrir Postman 2. Abrir collection SecureChargingPoints 3. Buscar método "Users" 4. Lanzar GET con "http://192.168.43.110:8000/api/userConsumption/users"	Array de identificadores de usuario (*address*)
CP07	API REST SecCP - Consumos	1. Abrir Postman 2. Abrir collection SecureChargingPoints 3. Buscar método "search consumptions" 4. Lanzar GET con "http://192.168.43.110:8000/api/userConsumption/data/1/4?id=0xA"	Array de consumos, no vacío
CP08	API REST SecCP - Anomalias - Estados	1. Abrir Postman 2. Abrir collection SecureChargingPoints 3. Buscar método "*search state anomalies*" 4. Lanzar GET con "http://192.168.43.110:8000/api/anomalies/state/search ?timestampIni=0&timestampEnd=1624550886 &chargingPoint=0x04e66BB1Df9066C79f311A3c80a328eDCCE2A308"	Array de anomalías, no vacío
CP09	API REST SecCP - Anomalias - Transacciones	1. Abrir Postman 2. Abrir collection SecureChargingPoints 3. Buscar método "*search state transactions*" 4. Lanzar GET con "http://192.168.43.110:8000/api/anomalies/transaction/search ?timestampIni=0&timestampEnd=1624550886 &chargingPoint=0x04e66BB1Df9066C79f311A3c80a328eDCCE2A308"	Array de anomalías, no vacío
CP10	Nuevo estado	1. Abrir proyecto 2. Abrir carpeta "*tests*" 3. Ejecutar "testsStatesBlockchain.py"	'OK'
CP11	Nueva transacción	1. Abrir proyecto 2. Abrir carpeta "*tests*" 3. Ejecutar "testsTransactionsBlockchain.py"	'OK'
CP12	Tests Tiempo Estados	1. Abrir proyecto 2. Abrir carpeta "*tests*" 3. Ejecutar "testsTimeSearch.py"	Consola no muestra *Exceptions*
CP13	Tests Anomalías	1. Abrir proyecto 2. Abrir carpeta "*tests*" 3. Ejecutar "testsAnomalies.py"	'OK'
CP14	Acceso Monitorización	1. Abrir navegador web (red eduroam) 2. Acceder a "*http://192.168.43.110:8000/*" 3. Introducir contraseña de MetaMask	Se muestra el monitor sin errores
CP15	Monitor: Mapa	1. Ir al monitor 2. Seleccionar un punto de carga en el mapa	Se abre el marcador
CP16	Monitor: Gráficas 1	1. Ir al monitor 2. Seleccionar duración: *Sin Definir*	Se muestran gráficas con anomalías
CP17	Monitor: Gráficas 2	1. Ir al monitor 2. Seleccionar un punto de carga	Se muestran gráficas estadísticas

Tabla 15: Matriz de trazabilidad de requisitos

	CP01	CP02	CP03	CP04	CP05	CP06	CP07	CP08	CP09	CP10	CP11	CP12	CP13	CP14	CP15	CP16	CP17
RF01	X	X															
RF02										X	X						
RF03											X						
RF04			X	X	X												
RF05					X	X	X										
RF06													X				
RF07			X	X				X	X			X		X	X	X	X
RF08														X	X	X	X
RF09															X		
RNF01																	
RNF02																	
RNF03														**Err**			
RNF04												**Err**					
RNF05														X			
RNF06															X		
RNF07																X	X

X: requisito validado por el CP, **Err**: requisito no pasa el CP

Un requisito se garantiza haberse implementado con éxito cuanto mayor es el número de CP que lo valide

Como se puede observar en la matriz de la tabla 15, se han logrado validar satisfactoriamente la mayoría de requisitos, excepto cuatro requisitos no funcionales. Los requisitos que corresponden al **diseño gráfico intuitivo (RNF01)** y a una **elevada curva de aprendizaje (RNF02)** no se han probado, debido a la necesidad de participación de partes interesadas externas y a la falta de tiempo. Por otro lado, el requisito no funcional de **tiempo de respuesta menor a 5 segundos en las búsquedas (RNF03)** no ha logrado pasar las pruebas. Esto se debe a que las consultas con un gran volumen de datos a devolver en la red de Blockchain son muy pesadas. Por ello, se ha tenido que limitar la búsqueda de estados a un máximo de 24 horas, mientras que para las transacciones, que no presenta volumen de datos elevados, si se permite cualquier rango de horas. Por último, el requisito no funcional de seguridad de **confidencialidad de datos en las comunicaciones (RNF04)** no se ha logrado debido a la falta de tiempo y recursos. Se ha desplegado el sistema de monitorización y API REST con el servidor web de desarrollo de Django, por tanto, el protocolo de transporte corresponde a *HyperText Transfer Protocol* (HTTP), que trae consigo el envío de mensajes en claro. Si se desea usar este proyecto en un futuro en producción, se debe contemplar esta vulnerabilidad y añadir seguridad en la red usando el protocolo *HyperText Transfer Protocol Secure* (HTTPS) y otros métodos de defensa, como cortafuegos (*firewall*) para bloquear el acceso no autorizado.

12

Conclusiones y Líneas Futuras

12.1. Conclusiones

En este apartado se expone las deducciones encontradas así como los problemas que han surgido a lo largo del proyecto.

Para comenzar, el Trabajo Fin de Grado surge de la continuación de un proyecto ya implementado y funcional, que pertenece al I Plan Propio de Smart Campus [58]. Sin embargo, por falta de material y tiempo, no se ha podido usar este proyecto para este escenario. Por suerte, se ha obtenido la documentación necesaria, por parte del grupo que trabajó en el proyecto, para conocer en detalle el comportamiento de este sistema y, sobre todo, el comportamiento de carga de los vehículos eléctricos en cada uno de los conectores (Mennekes y Schuko). Por esta razón, aunque no se ha podido implementar y probar con puntos de cargas reales, el simulador implementado se intenta asemejar lo máximo posible al sistema real.

Por otro lado, la escasez de conocimientos sobre Blockchain y detecciones de anomalías usando Machine Learning han sido retos que se han afrontado satisfactoriamente, gracias a la amplia documentación ofrecida en internet y al apoyo de los tutores que son expertos en la temática.

En cuanto a la tecnología Blockchain, la mayor complejidad ha sido y sigue siendo la correcta implementación de un sistema de trazabilidad, que permita buscar por fecha los datos energéticos almacenados. Se han implementado dos formas de búsqueda en base a un rango

de fechas: (1) con el uso de diccionarios en los contratos inteligentes, que almacenan listas indexadas por día los estados y las transacciones; (2) con el uso de Events y Logs, el cual permite añadir filtros de búsquedas y buscar por días. Sin embargo, estos no son métodos suficientemente eficientes para devolver un gran volumen de datos. Cuando el volumen es muy elevado, como es el caso de los estados de un punto de carga (que genera aproximadamente 1440 estados por día), el sistema tarda demasiado en devolver la sublista en rangos de fechas amplios (más de 48 horas), saltando en muchas ocasiones un error por "timeout". Por esta razón, el sistema de monitorización debe tener limitado las consultas sobre los estados hasta un máximo de horas, aunque para las transacciones si permita cualquier rango. Este problema surge por los escasos parámetros de búsquedas actuales que dispone la base de datos distribuida, al ser una tecnología reciente y en desarrollo. Este problema se discute como una de las posibles líneas futuras, que se definen en la siguiente sección.

Por último, las experimentaciones de los diferentes métodos de aprendizaje con los datos simulados nos llevan a las siguientes conclusiones. Los métodos de aprendizaje no supervisados no se ajustan correctamente con los datasets generados, debido a su elevada entropía (como medida de desorden) en la distribución de datos: en el caso de los estados, la potencia eléctrica cambia según la intensidad establecida por el usuario y el modelo del coche, y con un cierto ruido; y, en las transacciones, la duración depende en muchos casos de la disponibilidad del usuario o la batería del coche. Sin embargo, los árboles de decisión que se usan en los métodos de aprendizaje supervisado Decision Tree Classifier (DTC) y Random Forest Classifier (RFC) logran adecuarse correctamente en ambos conectores, identificando satisfactoriamente las anomalías (outliers). Finalmente, el método de aprendizaje profundo Multi-Layer Perceptron Classifier (MLPC) obtiene muy buenos resultados tanto para el conector Schuko como para Mennekes. Esto nos permite ver que los métodos Deep Learning pueden ser también una buena técnica para detectar patrones de comportamiento, en este caso posibles anomalías en los puntos de carga.

A pesar de los problemas que han surgido durante el proyecto, se considera que se ha logrado satisfactoriamente el objetivo final del TFG. A través del sistema de monitorización y la red Blockchain permisionada, se logra visualizar gráficamente el comportamiento del siste-

ma a un alto nivel (consciencia situacional), además de recopilar información sobre posibles amenazas. Esto presenta una primera línea de defensa en la Microgrid con conexión a infraestructuras basadas en puntos de cargas.

12.2. Líneas Futuras

Una de las ventajas de este Trabajo Fin de Grado son sus posibilidades de crecer en diferentes ámbitos. Esto se debe principalmente a las recientes tecnologías usadas, que se encuentran actualmente en expansión y estudio, como son las tecnologías Blockchain, Inteligencia Artificial (IA) y Microgrids. Además, mencionar que el avance de esta línea puede llegar a ser interesante por su relación con el avance de energías renovables y vehículos eléctricos en nuestra sociedad. Cada vez encontramos más puntos de cargas disponibles en las ciudades, que conllevan a un mayor riesgo a que sean usados de forma inapropiada. A continuación, se lista las principales futuras líneas de trabajo:

- **Sistema real**: el primer paso a realizar es la sustitución del simulador por un escenario real de estaciones de carga conectadas a un sistema central, donde usuarios puedan acceder y recargar sus vehículos eléctricos. Una vez implementado el entorno real, se tendría que volver a realizar las pruebas sobre los métodos de aprendizaje usando datos de consumo reales y ataques reales.

- **Sistema de trazabilidad mejorada**: como se mencionaba anteriormente, el principal problema de SecCP se encuentra en la eficiencia del sistema de trazabilidad: no se pueden buscar estados en un punto de carga con un rango elevado de horas, debido al enorme volumen de datos. Una de las alternativas propuesta es aprovechar una de las características de Ethereum: Events y Logs [59]. Estos logs quedan almacenados en la base de datos distribuidas y se pueden obtener a través de filtros creados por el usuario. Cada filtro permite buscar por topics (argumentos indexados) e incluye funciones que detectan cambios en la base de datos, como devolver solo nuevas entradas (*get_new_entries()*). Otra posibilidad es utilizar herramientas externas, como API TRACK de Telefónica [60], para el registro y consulta del estado actual y el histórico de sucesos de cada activo. Esta problemática puede ser solventada en un futuro con la incorporación de nuevas medi-

das de programación en los contratos inteligentes, como, por ejemplo, con la reciente incorporación del lenguaje de programación DAML creada por Digital Asset [61].

- **Puntos de carga rápida**: en esta línea se busca la ampliación y escalabilidad del sistema añadiendo nuevos tipos de conectores, como pueden ser conectores de carga rápidas (44kW-50kW), super rápidas (90kW-120kW) y ultra rápidas (130kW-150kW). Para ello, se necesitará de un análisis y estudio previo del comportamiento de sus curvas de carga, así como buscar el método de aprendizaje que mejor se ajuste a cada uno de los conectores, para una correcta detección de anomalías.

- **Aumentar la seguridad:** en este TFG se puede observar una primera capa de defensa enmarcada mayormente en las funciones de identificación, detección y protección (basadas en el framework propuesto por NIST [18]). Sin embargo, para garantizar un sistema totalmente resiliente frente ataques y amenazas es necesario la incorporación de nuevas técnicas sobre conciencia situacional, que se combinen con la actual detección de anomalías, para dar una mayor robustez al sistema. Además, es apropiado el diseño de estrategias de defensas para entornos críticos que ayuden al sistema a dar respuesta y a recuperarse cuando se analice o detecte un posible riesgo.

Referencias

[1] *Un Pacto Verde Europeo | Comisión Europea*. URL: https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_es (visitado 26-05-2021).

[2] *Total greenhouse gas emission trends and projections in Europe — European Environment Agency*. URL: https://www.eea.europa.eu/data-and-maps/indicators/greenhouse-gas-emission-trends-7/assessment (visitado 26-05-2021).

[3] *Greenhouse gas emissions from transport in Europe — European Environment Agency*. URL: https://www.eea.europa.eu/data-and-maps/indicators/transport-emissions-of-greenhouse-gases-7/assessment (visitado 26-05-2021).

[4] Mohamed E. El-Hawary. "The smart grid - State-of-the-art and future trends". En: *Electric Power Components and Systems* 42.3-4 (2014), págs. 239-250. ISSN: 15325008. DOI: 10.1080/15325008.2013.868558.

[5] Rabab Hassan y Ghadir Radman. "Survey on smart grid". En: *Conference Proceedings - IEEE SOUTHEASTCON* (2010), págs. 210-213. ISSN: 07347502. DOI: 10.1109/SECON.2010.5453886.

[6] Mostafa Farrokhabadi, Dimitris Lagos, Richard W. Wies y col. "Microgrid Stability Definitions, Analysis, and Examples". En: *IEEE Transactions on Power Systems* 35.1 (2020), págs. 13-29. ISSN: 15580679. DOI: 10.1109/TPWRS.2019.2925703.

[7] *Threat landscape for industrial automation systems. H2 2018 | Kaspersky ICS CERT*. URL: https://ics-cert.kaspersky.com/reports/2019/03/27/threat-landscape-for-industrial-automation-systems-h2-2018/ (visitado 26-05-2021).

[8] *attackics*. URL: https://collaborate.mitre.org/attackics/index.php/Main_Page (visitado 26-05-2021).

[9] Nist. "PRELIMINARY DISCUSSION DRAFT NIST Framework and Roadmap for Smart Grid Interoperability Standards , NIST Special Publication 1108R3 NIST Framework and Roadmap for Smart Grid Interoperability Standards ". En: *Nist Special Publication* 0 (2021), pág. 244. URL: http://dx.doi.org/10.6028/NIST.SP.1108r3.

[10] Weiming Tong, Lei Lu, Zhongwei Li y col. "A survey on intrusion detection system for advanced metering infrastructure". En: *Proceedings - 2016 6th International Conference on Instrumentation and Measurement, Computer, Communication and Control, IMCCC 2016* (2016), págs. 33-37. DOI: 10.1109/IMCCC.2016.193.

[11] Raju Gottumukkala, Rizwan Merchant, Adam Tauzin y col. "Cyber-physical System Security of Vehicle Charging Stations". En: *IEEE Green Technologies Conference*. Vol. 2019-April. IEEE Computer Society, abr. de 2019. ISBN: 9781728114576. DOI: 10.1109/GreenTech.2019.8767141.

[12] *smart energy | IEC*. URL: https://www.iec.ch/energies/smart-energy (visitado 27-05-2021).

[13] Zhiyi Li, Mohammad Shahidehpour y Farrokh Aminifar. "Cybersecurity in Distributed Power Systems". En: *Proceedings of the IEEE* 105.7 (2017), págs. 1367-1388. ISSN: 15582256. DOI: 10.1109/JPROC.2017.2687865.

[14] Farzam Nejabatkhah, Yun Wei Li, Hao Liang y col. "Cyber-Security of Smart Microgrids: A Survey". En: *Energies* 14.1 (2020), pág. 27. ISSN: 1996-1073. DOI: 10.3390/en14010027.

[15] Jean Paul A. Yaacoub, Ola Salman, Hassan N. Noura y col. "Cyber-physical systems security: Limitations, issues and future trends". En: *Microprocessors and Microsystems* 77 (2020). ISSN: 01419331. DOI: 10.1016/j.micpro.2020.103201.

[16] Keith Stouffer, Victoria Pillitteri, Suzanne Lightman y col. *Guide to Industrial Control Systems (ICS) Security*. Inf. téc. Gaithersburg, MD: National Institute of Standards y Technology, jun. de 2015. DOI: 10.6028/NIST.SP.800-82r2. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf.

[17] *SP 800-82 Rev. 3 (Draft), Guide to Industrial Control Systems (ICS) Security | CSRC*. URL: https://csrc.nist.gov/publications/detail/sp/800-82/rev-3/draft (visitado 27-05-2021).

[18] NIST Cybersecurity Framework Team. "Framework for improving critical infrastructure cybersecurity". En: *Proceedings of the Annual ISA Analysis Division Symposium* 535 (2018), págs. 9-25. ISSN: 10506527.

[19] Song Han, Miao Xie, Hsiao Hwa Chen y col. "Intrusion detection in cyber-physical systems: Techniques and challenges". En: *IEEE Systems Journal* 8.4 (2014), págs. 1052-1062. ISSN: 19379234. DOI: 10.1109/JSYST.2013.2257594.

[20] *Glosario sobre aprendizaje automático | Google Developers*. URL: https://developers.google.com/machine-learning/glossary?hl=es_419#a (visitado 28-05-2021).

[21] Ahmed S. Musleh, Guo Chen y Zhao Yang Dong. "A Survey on the Detection Algorithms for False Data Injection Attacks in Smart Grids". En: *IEEE Transactions on Smart Grid* 11.3 (2020), págs. 2218-2234. ISSN: 19493061. DOI: 10.1109/TSG.2019.2949998.

[22] Pietro Ferraro, C. King y Robert Shorten. "Distributed ledger technology for smart cities, the sharing economy, and social compliance". En: *IEEE Access* 6 (2018), págs. 62728-62746. ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2876766.

[23] Pierluigi Siano, Giuseppe De Marco, Alejandro Rolan y col. "A Survey and Evaluation of the Potentials of Distributed Ledger Technology for Peer-to-Peer Transactive Energy Exchanges in Local Energy Markets". En: *IEEE Systems Journal* 13.3 (2019), págs. 3454-3466. ISSN: 19379234. DOI: 10.1109/JSYST.2019.2903172.

[24] Michela Moschella, Pietro Ferraro, Emanuele Crisostomi y col. "Decentralized Assignment of Electric Vehicles at Charging Stations Based on Personalized Cost Functions and Distributed Ledger Technologies". En: *IEEE Internet of Things Journal* 4662.c (2021), págs. 1-12. ISSN: 23274662. DOI: 10.1109/JIOT.2021.3052045. arXiv: 1909.07073.

[25] *Inicio | ethereum.org*. URL: https://ethereum.org/es/ (visitado 31-05-2021).

[26] *Announcing Hyperledger Besu – Hyperledger*. URL: https://www.hyperledger.org/blog/2019/08/29/announcing-hyperledger-besu (visitado 31-05-2021).

[27] *La base de datos líder del mercado para aplicaciones modernas | MongoDB*. URL: https://www.mongodb.com/es (visitado 31-05-2021).

[28] *Managed MongoDB Hosting | Database-as-a-Service | MongoDB*. URL: https://www.mongodb.com/cloud/atlas (visitado 31-05-2021).

[29] *Welcome to Python.org*. URL: https://www.python.org/ (visitado 31-05-2021).

[30] *React – Una biblioteca de JavaScript para construir interfaces de usuario*. URL: https://es.reactjs.org/ (visitado 31-05-2021).

[31] *Solidity — documentación de Solidity - UNKNOWN*. URL: https://solidity-es.readthedocs.io/es/latest/ (visitado 31-05-2021).

[32] *The Web framework for perfectionists with deadlines | Django*. URL: https://www.djangoproject.com/ (visitado 31-05-2021).

[33] *Home - Django REST framework*. URL: https://www.django-rest-framework.org/ (visitado 31-05-2021).

[34] *Home | nivo*. URL: https://nivo.rocks/ (visitado 31-05-2021).

[35] *Welcome to PyOD documentation! — pyod 0.8.9 documentation*. URL: https://pyod.readthedocs.io/en/latest/ (visitado 31-05-2021).

[36] *anomatools · PyPI*. URL: https://pypi.org/project/anomatools/ (visitado 31-05-2021).

[37] *scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation*. URL: https://scikit-learn.org/stable/index.html (visitado 31-05-2021).

[38] *PyMongo — MongoDB Drivers*. URL: https://docs.mongodb.com/drivers/pymongo/ (visitado 31-05-2021).

[39] *Introduction — Web3.py 5.19.0 documentation*. URL: https://web3py.readthedocs.io/en/stable/ (visitado 31-05-2021).

[40] *GitHub - trufflesuite/truffle: A tool for developing smart contracts. Crafted with the finest cacaos*. URL: https://github.com/trufflesuite/truffle (visitado 31-05-2021).

[41] *MetaMask*. URL: https://metamask.io/ (visitado 31-05-2021).

[42] *Home | Scrum.org*. URL: https://www.scrum.org/ (visitado 31-05-2021).

[43] *Home - Open Charge Alliance*. URL: https://www.openchargealliance.org/ (visitado 01-06-2021).

[44] *MQTT - The Standard for IoT Messaging*. URL: https://mqtt.org/ (visitado 01-06-2021).

[45] *PHOENIX CONTACT | EV Charge Control Advanced*. URL: https://www.phoenixcontact.com/online/portal/es?1dmy&urile=wcm:path:/eses/web/main/products/subcategory_pages/AC_charging_controllers_for_charging_stations_and_wallboxes_P-29-04-02/18ac632a-4163-4a4a-8c7d-742b4688558a/18ac632a-4163-4a4a-8c7d-742b4688558a (visitado 01-06-2021).

[46] *MENNEKES Automotive | MENNEKES*. URL: https://www.mennekes.es/empresa/sobre-nosotros/mennekes-automotive/ (visitado 01-06-2021).

[47] *Power plug & outlet Type F (Schuko) - World Standards*. URL: https://www.worldstandards.eu/electricity/plugs-and-sockets/f/ (visitado 01-06-2021).

[48] *OAuth 2.0 — OAuth*. URL: https://oauth.net/2/ (visitado 01-06-2021).

[49] *STRIDE chart - Microsoft Security*. URL: https://www.microsoft.com/security/blog/2007/09/11/stride-chart/ (visitado 10-03-2021).

[50] R. Khan, K. McLaughlin, D. Laverty y col. "STRIDE-based threat modeling for cyber-physical systems". En: *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. 2017, págs. 1-6. DOI: 10.1109/ISGTEurope.2017.8260283.

[51] *Hyperledger Fabric – Hyperledger*. URL: https://www.hyperledger.org/use/fabric (visitado 19-03-2021).

[52] *Hyperledger Besu Enterprise Ethereum Client - Hyperledger Besu*. URL: https://besu.hyperledger.org/en/stable/ (visitado 22-03-2021).

[53] *Using IBFT 2.0 (PoA) - Hyperledger Besu*. URL: https://besu.hyperledger.org/en/stable/Tutorials/Private-Network/Create-IBFT-Network/ (visitado 02-06-2021).

[54] *IBFT 2.0 - Hyperledger Besu*. URL: https://besu.hyperledger.org/en/stable/HowTo/Configure/Consensus-Protocols/IBFT/ (visitado 02-06-2021).

[55] *Compare electric vehicles - EV Database*. URL: https://ev-database.org (visitado 29-04-2021).

[56] Mete Ozay, Iñaki Esnaola, Fatos Tunay Yarman Vural y col. "Machine Learning Methods for Attack Detection in the Smart Grid". En: *IEEE Transactions on Neural Networks and Learning Systems* 27.8 (2016), págs. 1773-1786. ISSN: 21622388. DOI: 10.1109/TNNLS.2015.2404803. arXiv: 1503.06468.

[57] *Curvas PR y ROC. Las curvas ROC y PR (precision-recall)... | by Jaime Ramírez | bluekiri | Medium*. URL: https://medium.com/bluekiri/curvas-pr-y-roc-1489fbd9a527 (visitado 10-05-2021).

[58] *Home | Scrum.org*. URL: https://www.scrum.org/ (visitado 31-05-2021).

[59] *Events and Logs - Hyperledger Besu.* URL: https://besu.hyperledger.org/en/stable/Concepts/Events-and-Logs/ (visitado 14-06-2021).

[60] *API Track: Trazabilidad completa con Blockchain | Telefónica.* URL: https://blockchain.telefonica.com/soluciones-para-tu-negocio/trustos/api-track/ (visitado 06-06-2021).

[61] *Hyperledger Besu now has DAML Smart Contracts.* URL: https://blog.digitalasset.com/press-release/ethereum-compatible-hyperledger-besu-now-has-enterprise-grade-daml-smart-contracts (visitado 14-06-2021).

Apéndice A
Manual de Usuario

A.1. Acceso

Para acceder al sistema de monitorización actualmente desplegada y funcional, se debe encontrar dentro de la red *eduroam* de la Universidad de Málaga. Desde aquí, se puede acceder a la servidor web, en desarrollo, con el siguiente enlace: http://192.168.43.110:8000/

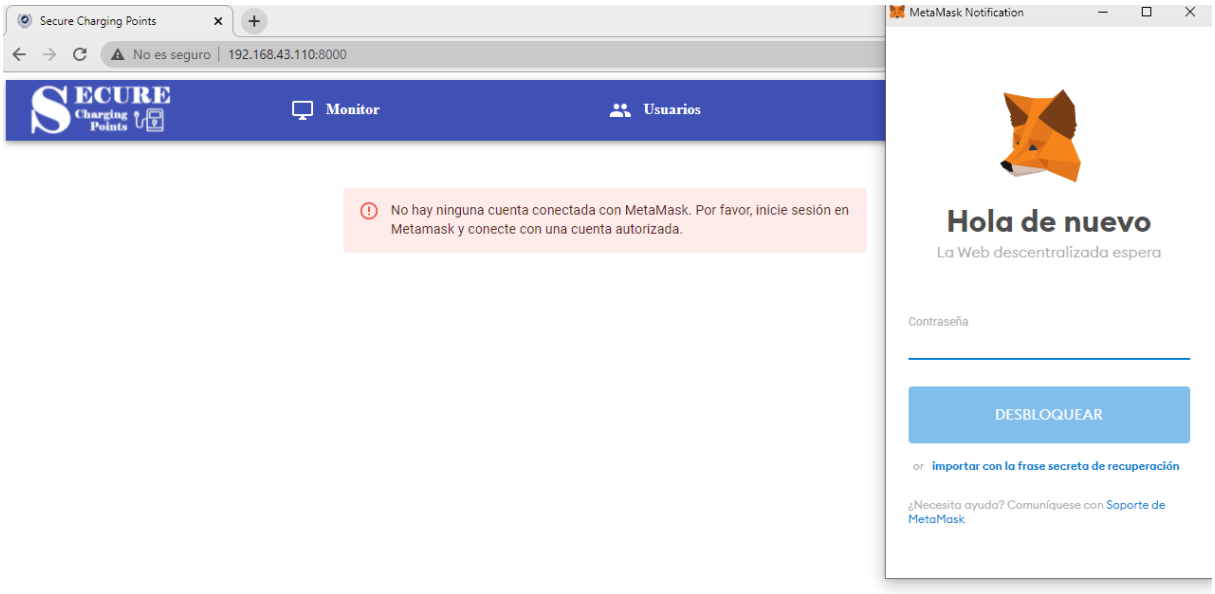


Figura 14: Vista de la página web sin acceso

Además, para autenticarse en la página web es necesario añadir la extensión de navegador MetaMask desde https://metamask.io/. Una vez descargada la extensión, se debe crear un nuevo monedero. Una vez creado el monedero se importará la siguiente clave privada, que corresponde con una cuenta autorizada en la red.

```
Clave privada a importar: 2dd956ebf2071d81b6b78298328a6f6e9d2f8341
5f387e8d978d72298f4ef8b6
```

Finalmente, se debe conectar la cuenta autorizada con la página web desde la extensión de Metamask (que aparece arriba a la derecha).

A.2. Monitor (Inicio)

La página principal o de inicio corresponde con el monitor y muestra de primera vista arriba el mapa con todos los puntos de cargas y debajo las gráficas con información sobre las anomalías (ver figuras 15 y 16).

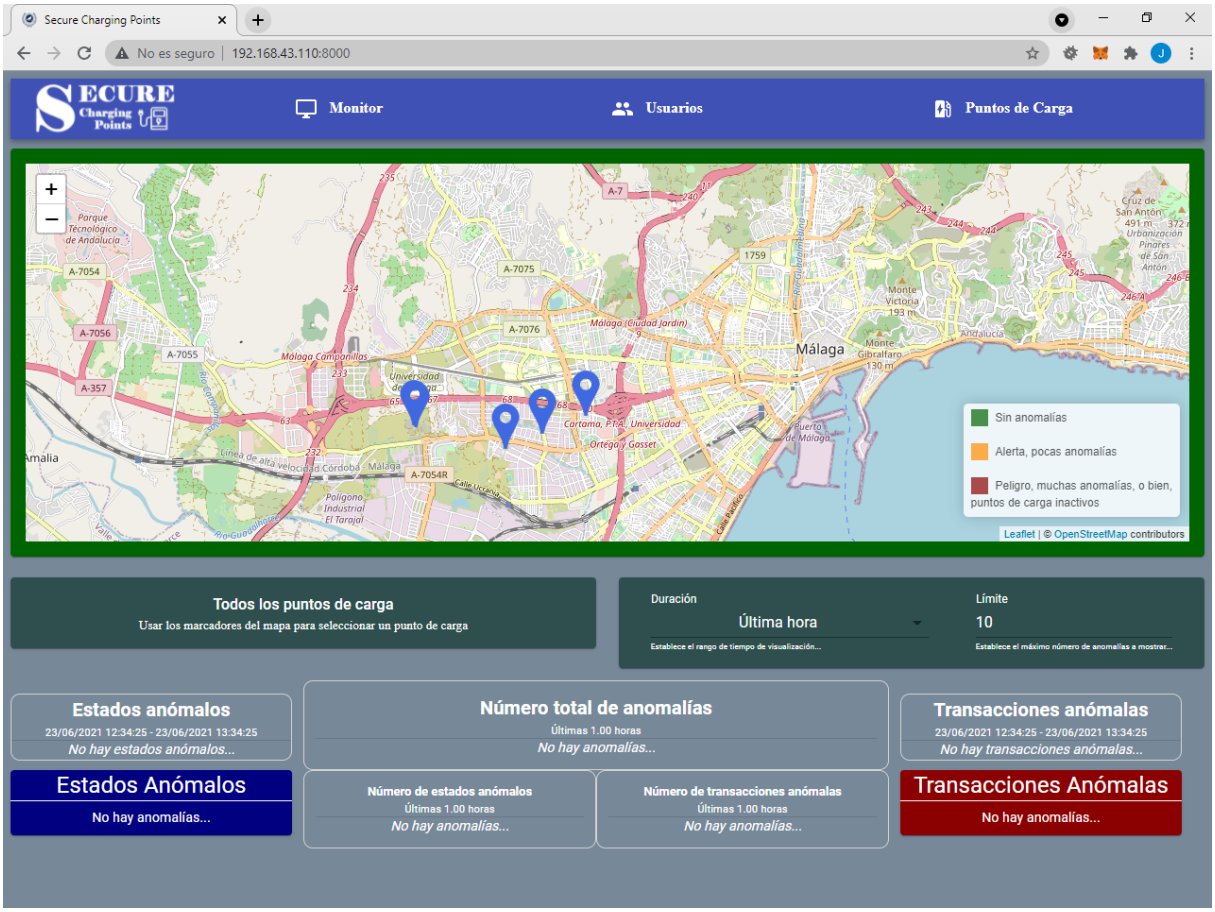


Figura 15: Página Web: Monitor sin anomalías

En el mapa se colorean con tres niveles cada punto de carga, según el nivel de peligro: verde, si no hay anomalías; amarillo, si hay pocas anomalías; y rojo, si hay muchas anomalías o se encuentra desactivado o sin comunicación. En cada marcador se puede ver el número de anomalías y la hora del último estado. Además, permite seleccionar el punto de carga para visualizar abajo gráficas detalladas de dicho punto de carga.

El panel de gráficas se divide en varios grupos. Arriba se encuentra el filtro de duración de rango de visualización (última hora, últimas 12 horas...) y el límite de anomalías a visualizar. Debajo se muestran las gráficas e información de todos los puntos de cargas, o bien, del punto

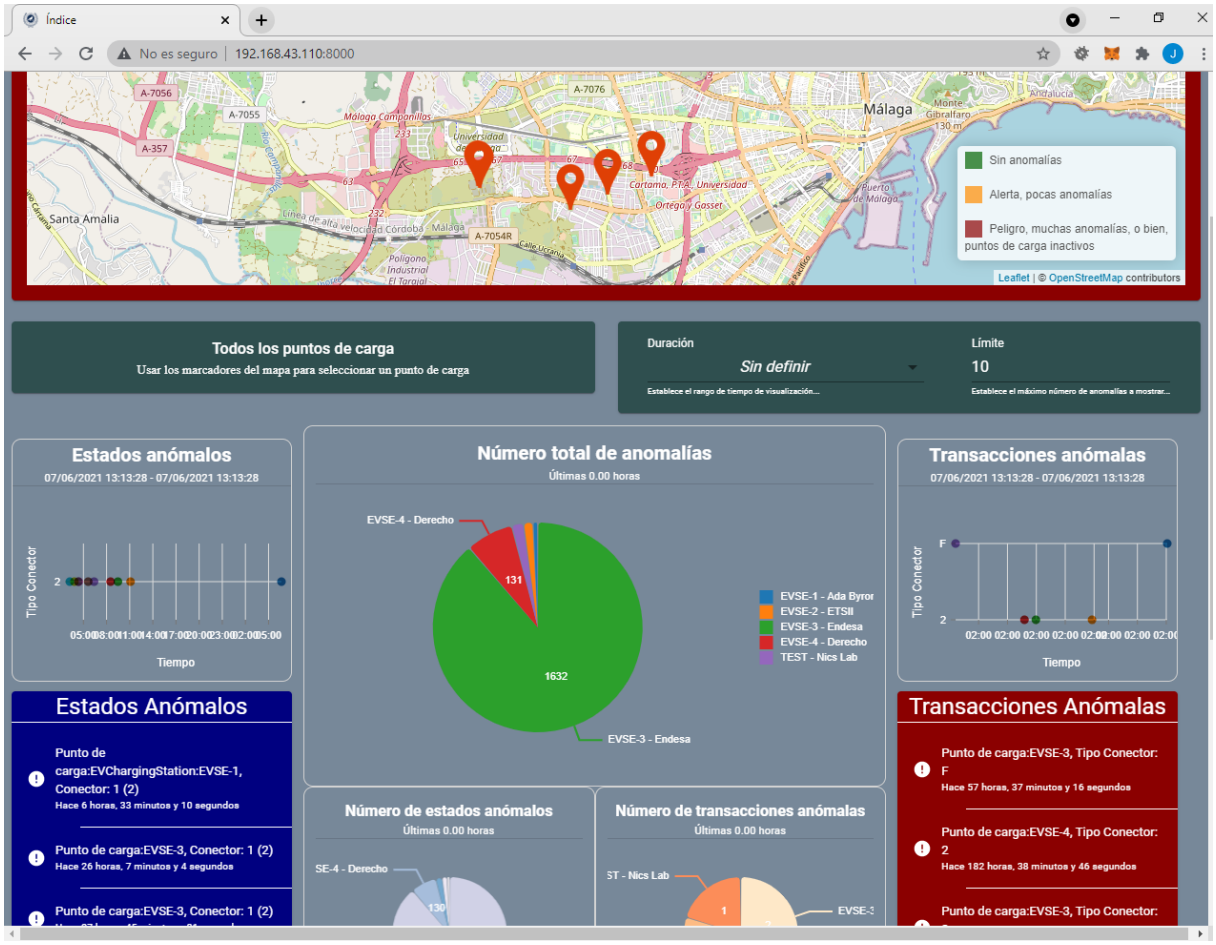


Figura 16: Página Web: Monitor con anomalías

de carga que se ha seleccionado. En la columna de la izquierda se muestran solo los estados anómalos y en la columna de la derecha, solo las transacciones anómalas, si hay.

Si el monitor está mostrando todos los puntos de cargas, entonces las gráficas centrales corresponden a gráficas circulares con los números de estados y transacciones anómalas correspondiente a cada punto de carga. Si se selecciona un punto de carga, entonces las gráficas centrales muestra lo siguiente: (1) información del último estado, (2) gráfica circular con el número de anomalías de estados y transacciones, (3) gráfica lineal de la potencia de cada conector con respecto al tiempo, (4) gráfica lineal del estado (activo/inactivo) de cada conector con respecto al tiempo, y (5) gráfica de puntos con las transacciones mostradas en relación potencia media y duración.

A.3. Lista de usuarios

En esta página web, se muestra todos los usuarios identificados en el sistema que han cargado sus vehículos eléctricos, al menos una vez, en algún punto de carga (ver figura 17). Además se permite una búsqueda por identificador del usuario. Para cada usuario listado se muestra información sobre su último consumo: energía consumida, potencia media, tipo de conector, duración de la carga y la fecha del consumo.

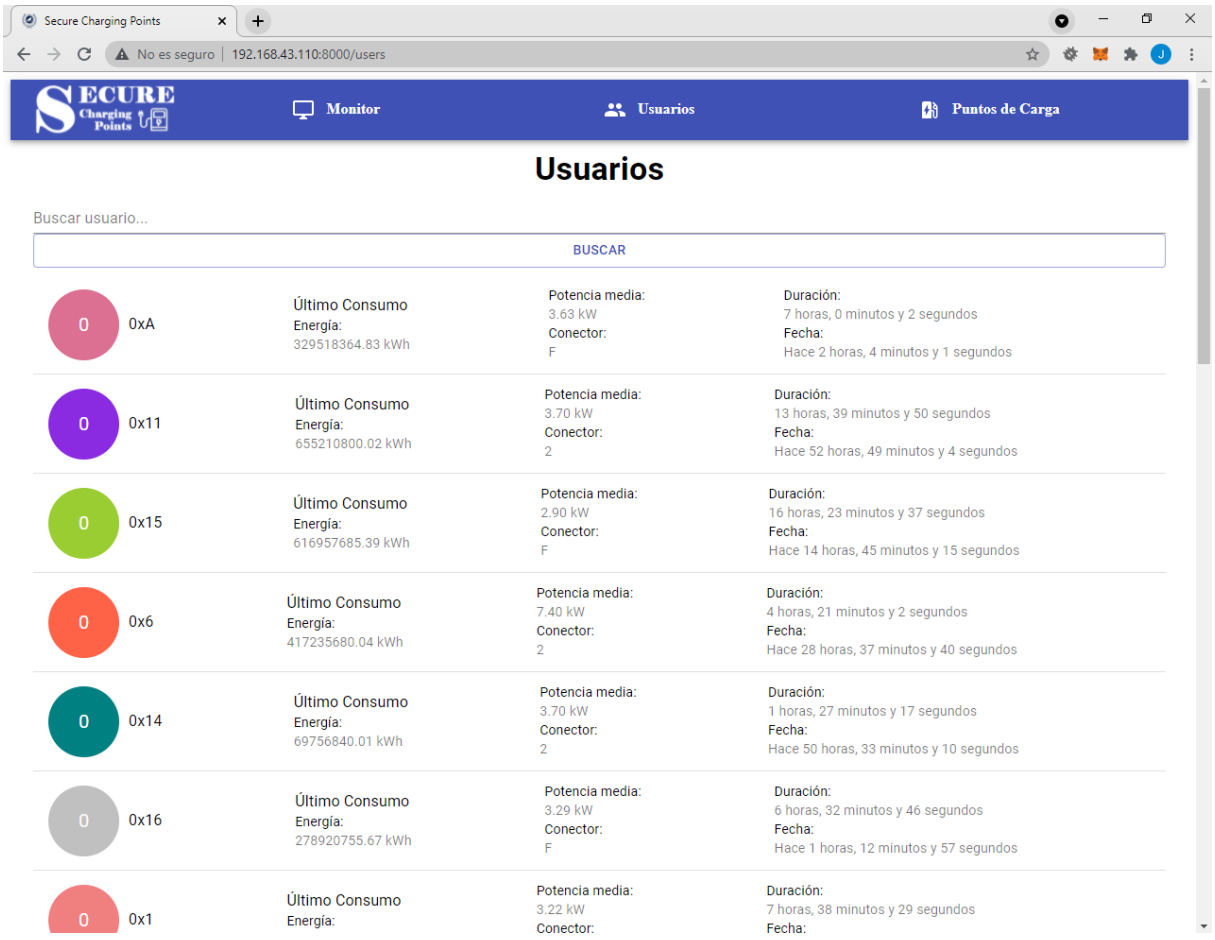


Figura 17: Página Web: Lista de Usuarios

Además, se puede seleccionar un usuario y abrir una página web con información detallada sobre las transacciones de dicho usuario. En esta página web, se muestra arriba el último consumo realizado y, a continuación, un filtro donde se permite buscar todas las transacciones realizadas por el usuario entre un rango de fechas (ver figuras 18 y 19).

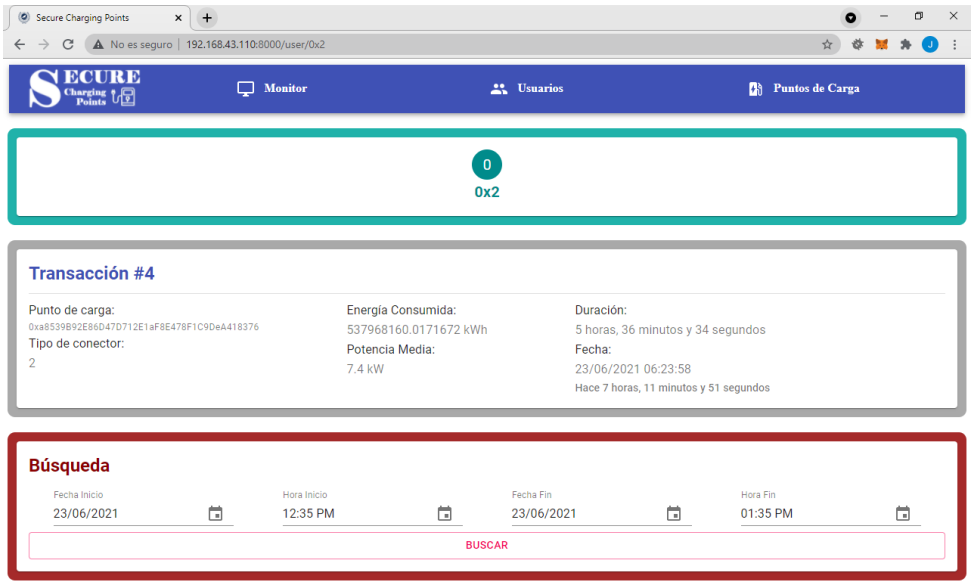


Figura 18: Página Web: Usuario

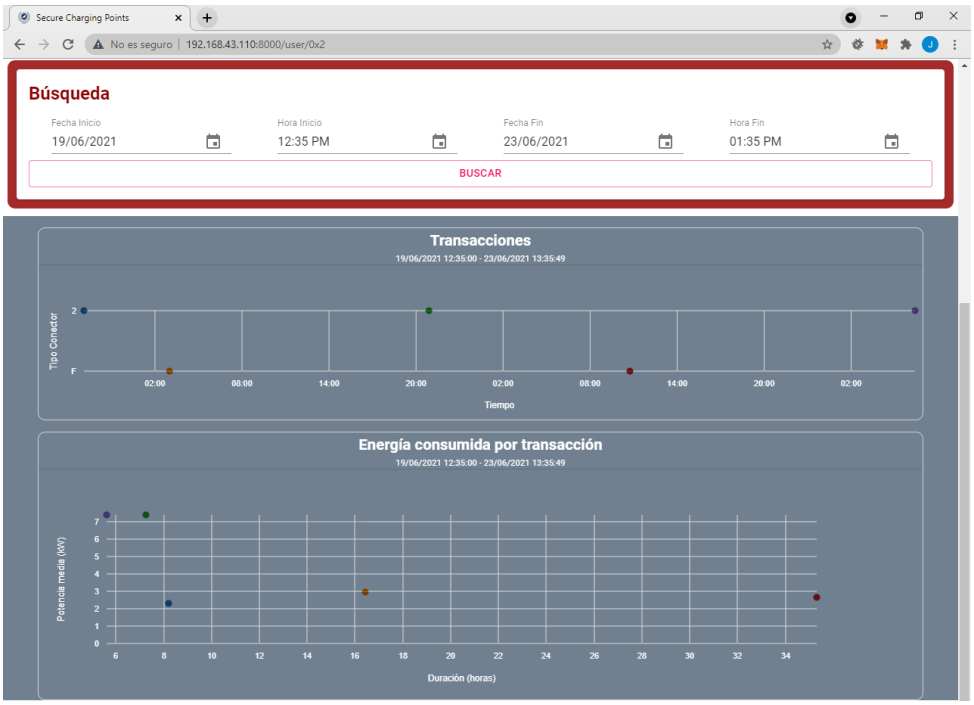


Figura 19: Página Web: Usuario (Búsqueda por Fechas)

A.4. Lista de puntos de carga

En esta página web, se listan todos los puntos de cargas. Además, permite la búsqueda de un punto de carga indicando su clave pública, que corresponde con el identificador único de cada uno de ellos (ver figura 20).

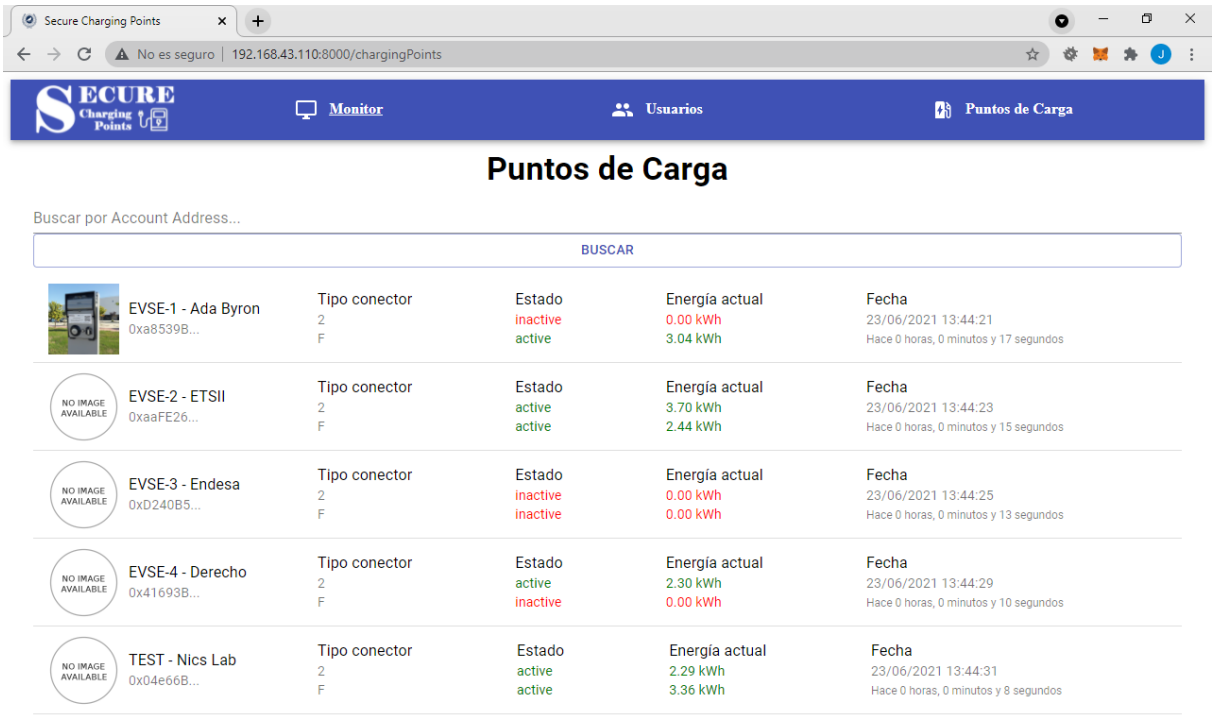


Figura 20: Página Web: Lista de Puntos de Cargas

Cada uno de los elementos de la lista muestra de primera mano información sobre: su nombre, los tipos de conectores y el último estado recogido (que incluye el estado y la potencia demandada en cada conector, y la hora que fue publicada). Además, se puede seleccionar un punto de carga y mostrar información detallada en cada uno de ellos.

Al seleccionar un punto de carga, se abre una página web que muestra lo siguiente (ver figuras 21 y 22): arriba se muestra la información complementaria del punto de carga, como id, nombre, dirección, vehículos permitidos...; a continuación, se muestra un bloque con información sobre el último estado recogido en ese punto de carga; le sigue, otro bloque con información sobre la última transacción de carga; y, al final, se encuentra un filtro que permite al usuario buscar los estados y transacciones entre un rango de fechas. Esta búsqueda muestra las siguientes gráficas: (1) gráfica circular con relación estados con estados anómalos, (2) gráfica de puntos que muestra los estados anómalos con relación tipo conector-tiempo, (3) gráfica lineal de potencia-tiempo por cada conector, (4) gráfica lineal con relación estado (activo/inactivo) y tiempo, (5) gráfica circular con relación número de transacciones correctas con anómalas, (6) gráfica de puntos con las transacciones anómalas en relación tipo conector-tiempo y (7) gráfica de puntos con todas las transacciones en relación potencia media-duración.

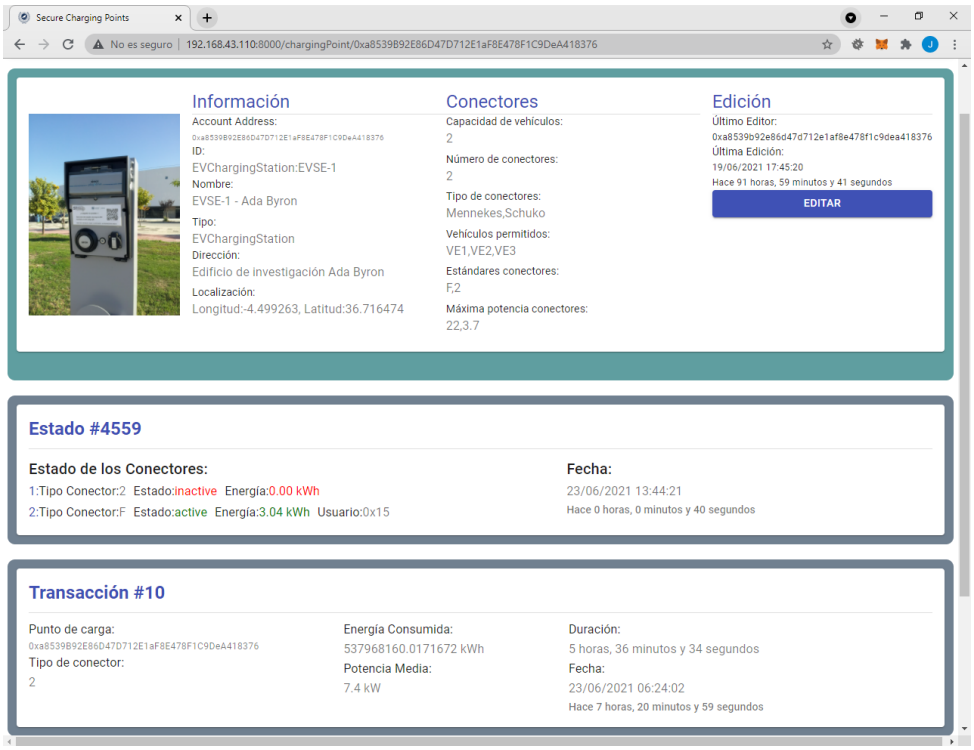


Figura 21: Página Web: Punto de Carga

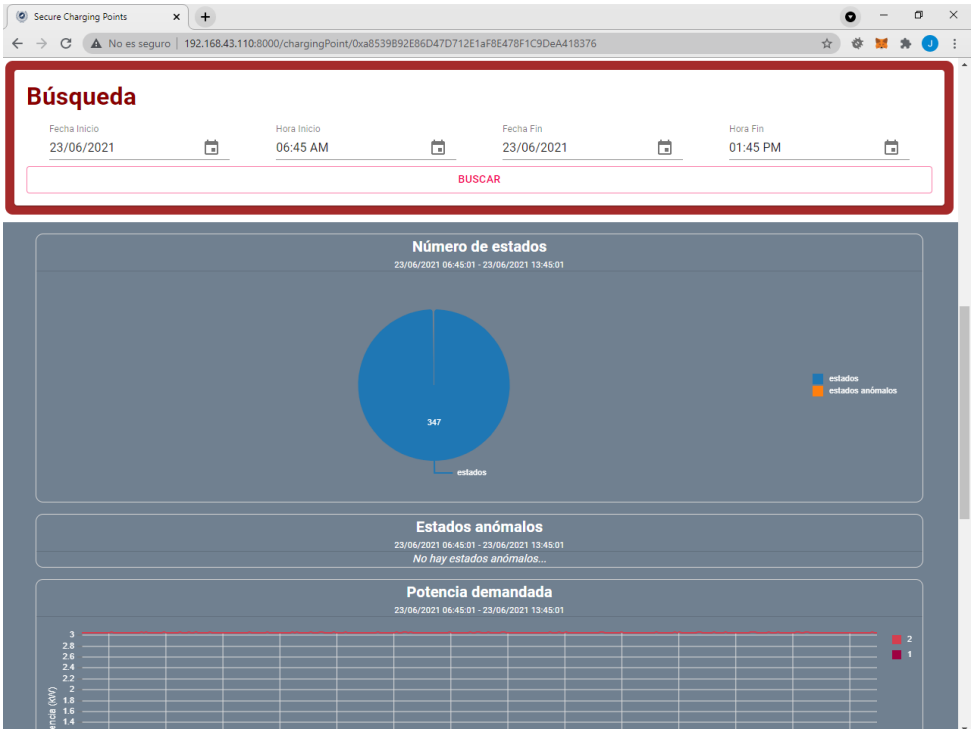


Figura 22: Página Web: Punto de Carga (Búsqueda por Fechas)

A.5. Explorador de métricas: Prometheus & Grafana

Prometheus es un servicio de monitorización y alerta que accede a las métricas de Hyperledger Besu. Para visualizar los datos recolectados de manera gráfica se ha usado el programa Grafana, que usa el nodo de Prometheus como *data source*. Estos exploradores se pueden ver en las figuras 23 y 24.

Para acceder a estas páginas webs, se debe dirigir a las siguientes direcciones (dentro de la red eduroam):

- **Prometheus:** *http://192.168.43.110:9090*
- **Grafana:** *http://192.168.43.110:3000*
  - Usuario: admin
  - Contraseña: nicsUrbanLab2021

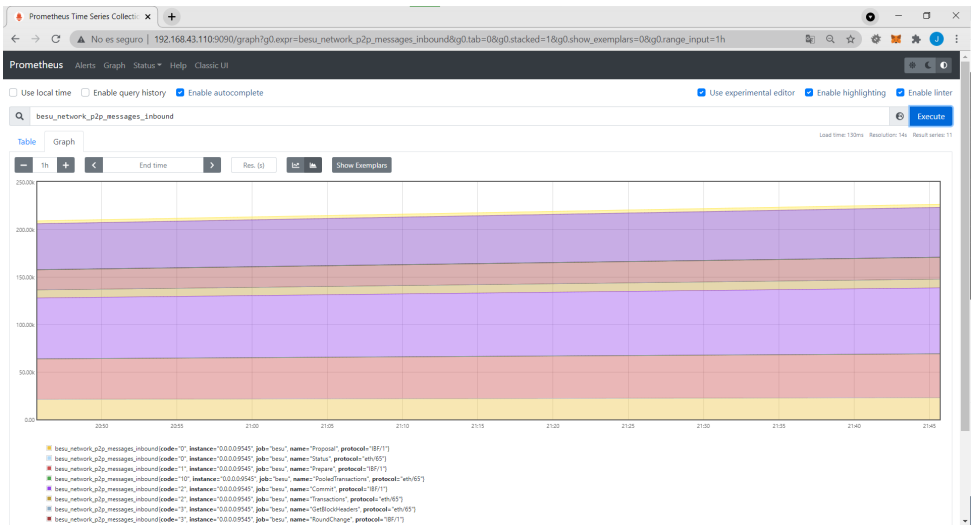


Figura 23: Monitorización de métricas de la red de Blockchain usando Prometheus

Figura 24: Monitorización de métricas de la red de Blockchain usando Grafana

Apéndice B
Manual de Instalación

B.1. Red de Blockchain

Se han seguido los pasos explicados en el tutorial: https://besu.hyperledger.org/en/stable/Tutorials/Private-Network/Create-IBFT-Network/

Tras desplegar los 4 nodos validadores, es necesario compilar y desplegar los contratos inteligentes usando Truffle:

```
sudo npm install -g truffle
sudo npm install --save @truffle/hdwallet-provider
sudo truffle migrate --reset --compile-all --network urbanLab
```

B.2. Sistema de monitorización

Para instalar el sistema de monitorización en local, se debe descargar el proyecto de github y ejecutar los siguientes comandos. Además, será necesario instalar todos los paquetes de python que sean requeridos.

```
sudo git clone https://github.com/jesuscumpli/UrbanLab_Monitorizacion.git
mv UrbanLab_Monitorizacion monitorizacion
cd monitorizacion
pip3 install django djangorestframework
[Instalar librerías frontend]
cd frontend
sudo npm install date-fns
sudo npm install
sudo npm run dev
[Ejecutar servidor web]
```

```
cd ..
python3 manage.py runserver 0.0.0.0:8000
```

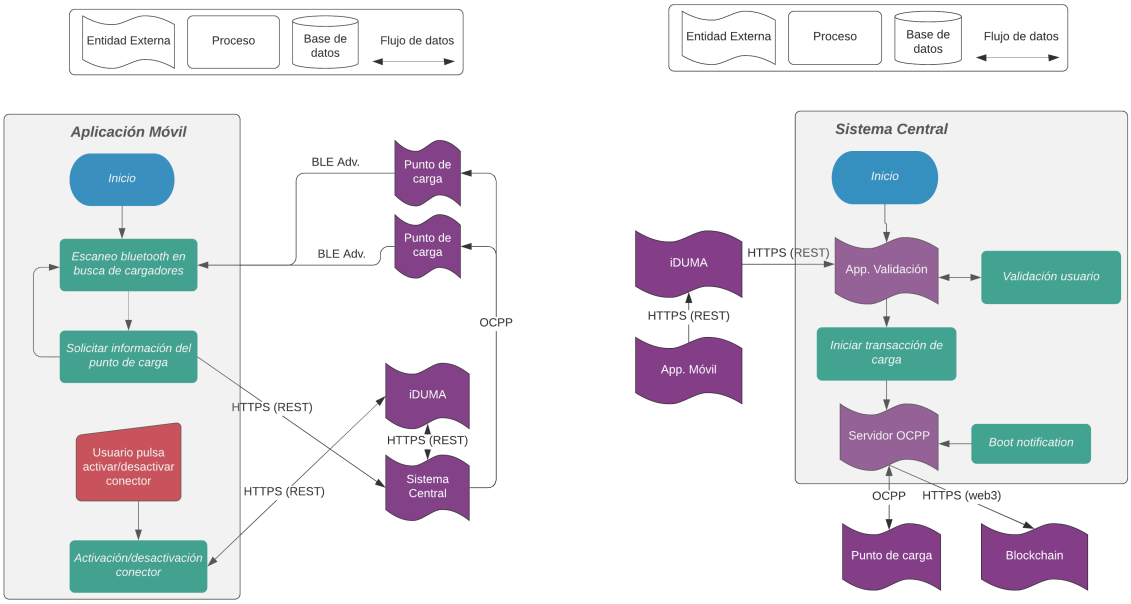
B.3. Simulador EVSE

Para ejecutar el simulador de puntos de cargas, tan sólo hay que ejecutar uno de los *mainSimulator.py* dentro de la carpeta *evseSimulator*. El script *mainSimulatorMongoDB.py* genera datos simulados de cargas y los guarda en la base de datos MongoDB. Este simulador es usado para generar los datasets de entrenamiento. Por otro lado, el script *mainSimulatorBlockchain* genera datos de energía en tiempo real y los almacena en la red de Blockchain. Este es usado como sustituto de los puntos de cargas reales.

B.4. Detección de anomalías

Para ejecutar el detector de anomalías, se debe dirigir a la carpeta *anomaliesDetection/detection*. En esta carpeta hay dos programas main de Python que deben estar ejecutándose siempre. Uno de ellos es *mainStatesDetection.py*, que obtiene cada 5 minutos los últimos estados y predice si algunos de ellos son anómalos, guardandolos en la base de datos local MongoDB. Por otro lado, el *mainTransactionsDetection.py* consulta cada 10 minutos las últimas transacciones y analiza si alguna de ellas es una anomalía, guardándola en la base de datos local.

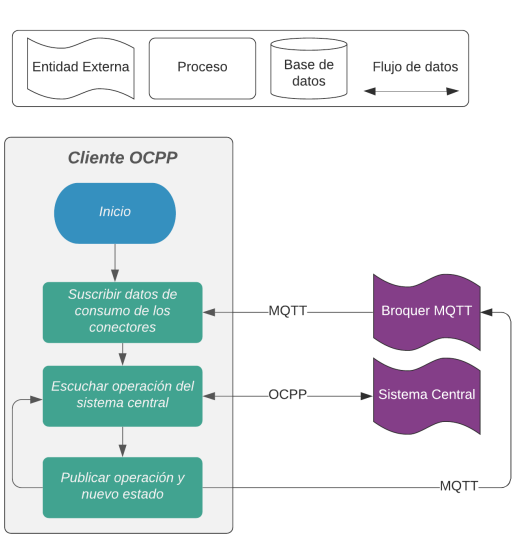
Apéndice C
Diagramas de Flujo de Datos



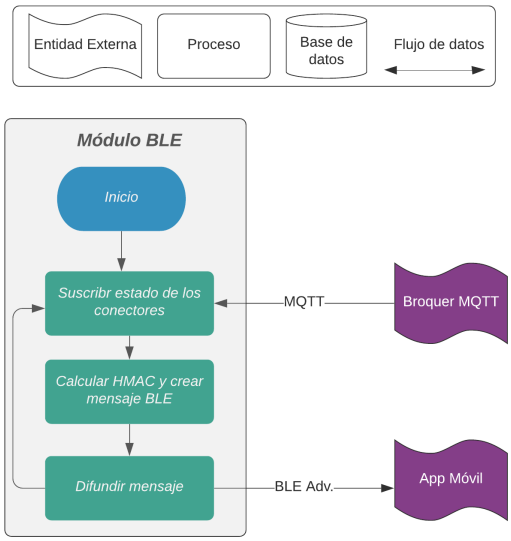
(a) DFD de la App. Móvil

(b) DFD del Sistema Central

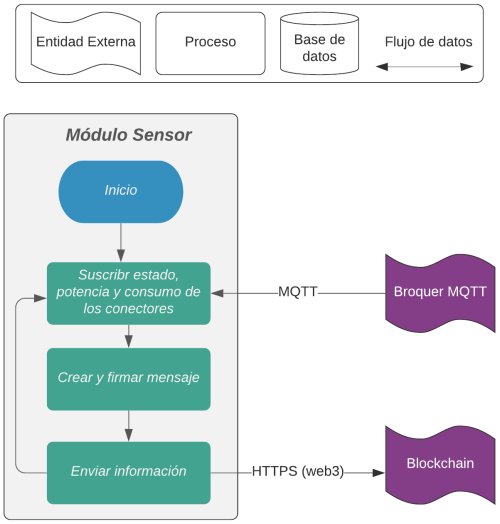
Figura 25: Diagramas de flujos de la App. Móvil y Sist. Central



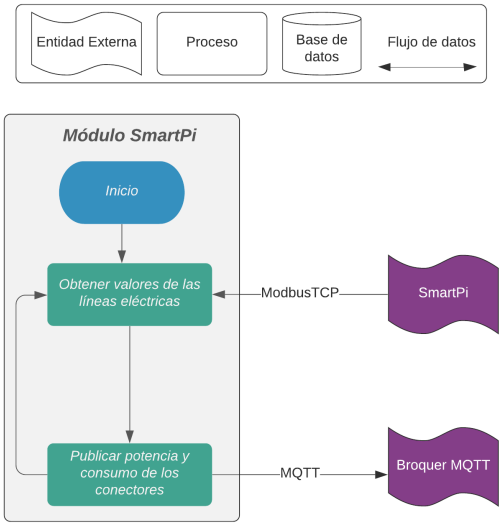
(a) DFD del Cliente OCPP



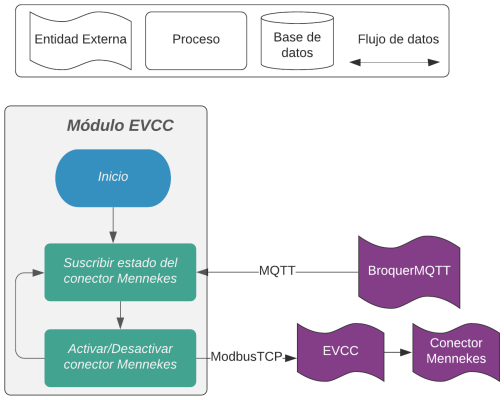
(b) DFD del Módulo BLE



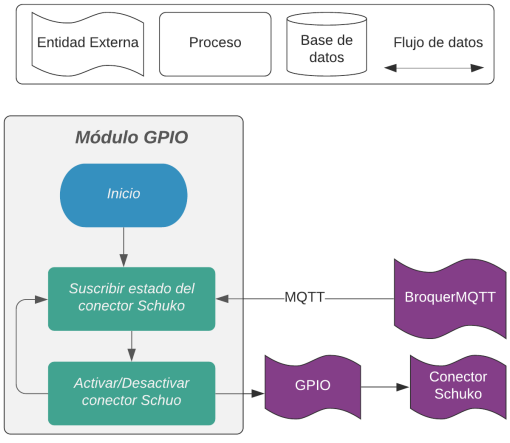
(c) DFD del Módulo Sensor



(d) DFD del Módulo SmartPi



(e) DFD del Módulo EVCC



(f) DFD del Módulo GPIO

Figura 26: Diagramas de flujos de los módulos de un punto de carga

UNIVERSIDAD DE MÁLAGA | uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA