UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Estrategias de seguridad para dispositivos Internet de las cosas industriales (IioT)

Presentado por Víctor De Marco Velasco
en Universidad de Burgos — 7 de julio de 2025
Tutor: Carlos Cambra Baseca

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. Carlos Cambra Baseca, profesor del departamento de Digitalización, área de Ciencia de la Computación e Inteligencia Artificial.

Expone:

Que el alumno D. Víctor De Marco Velasco, con DNI 71706956V, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Estrategias de seguridad para dispositivos Internet de las cosas industriales (IioT).

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 7 de julio de 2025

Resumen

Este proyecto propone el desarrollo de diferentes aplicaciones web con el propósito de mejorar la seguridad de los dispositivos IoT y la interacción del usuario con los datos recopilados por dichos dispositivos. Una página web se encarga de recibir los datos, comprobar si dichos datos han sido realmente enviados por nuestro dispositivo o si es posible que hayan sido generados por un tercero con intenciones maliciosas y, una vez comprobado, proceda a almacenarlos. La segunda página web está más enfocada al usuario, permitiéndole interactuar con los datos recopilados e incluso almacenar/analizar los suyos propios.

Descriptores

internet de las cosas, LoRaWAN, Reglas de asociación, The things network, Detección de anomalías,Python, Flujos de red...

Abstract

This project proposes the development of different web apps with the aim of improving the security of IoT devices and the user's interaction with the data collected by these devices. One web page is responsible for receiving the data, verifying whether the data was genuinely sent by our device or if it could have been generated by a third party with malicious intent, and, once verified, storing it. The second web page is more user-focused, allowing the user to interact with the collected data and even store and analyze their own data..

Keywords

IoT, LoRaWAN, Association rules, The things network, Anomaly detection,Python, Network flows...

Índice general

Índice de figuras

Índice de tablas

1. Introducción

Durante el desarrollo de este proyecto, he realizado una serie de funciones con el propósito de mejorar la seguridad y gestión de datos procedentes de un dispositivo IoT. Estas funciones se pueden dividir en tres partes:

En primer lugar,la configuración del hardware involucrado: un gateway LoRaWAN Dragino y un sensor de detección de movimiento MerryIoT. Ambos dispositivos fueron integrados y conectados a la plataforma The Things Network (TTN), estableciendo un flujo de datos constante, que más tarde utilizaría para crear el dataset principal del proyecto.

La segunda función fue el diseño y desarrollo de una primera aplicación web con el propósito de recibir y clasificar los datos. Mediante la implementación de un webhook proporcionado por TTN y el uso de un túnel de Cloudflare, esta aplicación es capaz de recibir en tiempo real los paquetes enviados por el dispositivo IoT. Una vez recibidos, dichos paquetes eran evaluados mediante un conjunto de reglas heurísticas con el objetivo de determinar si correspondían a situaciones normales (reales) o a comportamientos sospechosos (infectados). Finalmente, todos los datos analizados eran almacenados en un archivo CSV.

Por último, se desarrolló una segunda aplicación web, esta vez centrada en la interacción con el usuario.Permitiéndole gestionar e interactuar con distintos archivos CSV. Entre sus funcionalidades destacar la posibilidad de subir nuevos datasets, analizarlos automáticamente, enviar paquetes personalizados a la primera aplicación web y visualizar gráficas interactivas generadas a partir de los datos del archivo que el usuario desee.

1.1. Estructura de la memoria

- **Introducción**: Descripción de las principales funciones realizadas durante el proyecto y la estructura de la memoria.
- **Objetivos del Proyecto**: Explicación de los objetivos generales, técnicos y personales que se intentan lograr durante el desarrollo del proyecto.
- **Conceptos Teóricos**: Descripción de los principales conceptos teóricos relacionados y aplicados durante el desarrollo del proyecto.
- **Técnicas y Herramientas**: Explicación de las técnicas y herramientas utilizaras para poder llevar a cabo el desarrollo del proyecto.
- **Aspectos Relevantes del Desarrollo del Proyecto**: Explicación detallada de las diferentes partes del proyecto, desde la generación del dataste hasta la creación de las distintas aplicaciones web relacionadas con el mismo.
- **Trabajos Relacionados**: Consideración de trabajos y proyectos anteriores relacionadas con el tema tratado en este proyecto.
- **Conclusiones y Líneas de Trabajo Futuras**: las conclusiones obtenidas al finalizar el proyecto y las posibles futuras lineas de trabajo que se podrían llevar a cabo en este proyecto.

1.2. Estructura de los anexos

- **Plan de Proyecto**: El plan de proyecto consta de analizar la planificación temporal y de estudiar la viabilidad económica y legal del proyecto.
- **Requisitos**: En este apartado se realiza una explicación de los requisitos funcionales y no funcionales ademas de nombrar un numero variado de casos de uso referentes al proyecto.
- **Diseño**: En este apartado se explica que datos utiliza el proyecto, su arquitectura y se muestran diferentes diagramas sobre el funcionamiento de sus aplicaciones.

- **Manual del programador**: En este apartado se explican la estructura de directorios y archivos que componen el proyecto. También se explica como crear el entorno necesario para el proyecto. Además se define cada paso necesario para lograr compilar, ejecutar e instalar el proyecto.
- **Manual del usuario**: En este apartado se detallan los distintos requisitos necesarios para que el usuario pueda usar el software desarrollado correctamente.
- **Sostenibilidad Curricular**: En este apartado se reflexiona sobre los aspectos de sostenibilidad que se han considerado durante el proyecto.

1.3. Materiales adicionales

Despliegue con Docker

El proyecto desarrollado está pensado para ser ejecutado a través de un contenedor de Docker, aun así puede ser ejecutado de forma local si así se desea.

2. Objetivos del proyecto

Este apartado explica de forma precisa y concisa cuáles son los objetivos que se persiguen con la realización del proyecto. Se puede distinguir entre los objetivos marcados por los requisitos del software a construir, los objetivos de carácter técnico que plantea a la hora de llevar a la práctica el proyecto y los objetivos personales del autor.

2.1. Objetivos Generales

- Diseñar e implementar una solución completa que permite recoger, analizar, almacenar y visualizar los datos generados por un dispositivo IoT conectado a una red LoRaWAN.

- Proporcionar a usuarios no técnicos una interfaz web intuitiva desde la que puedan interactuar con los datos, analizar su contenido y detectar posibles anomalías.

2.2. Objetivos Técnicos

- Configurar y poner en funcionamiento una infraestructura IoT basada en TTN (The Things Network), incluyendo un gateway LoRaWAN Dragino y un dispositivo MerryIoT.

- Desarrollar una aplicación web de recepción que utilice un webhook de TTN y un túnel de Cloudflare para recolectar en tiempo real los datos generados por el dispositivo.

- Implementar un sistema de evaluación heurística de los paquetes recibidos para clasificarlos automáticamente como reales o infectados según sus características.
- Almacenar los datos evaluados en un archivo CSV, creando así un dataset que será utilizado a futuro.
- Desarrollar una segunda aplicación web de gestión y visualización, donde el usuario pueda:
  - Subir y analizar sus propios archivos CSV.
  - Visualizar gráficamente información clave como temperatura o humedad.
  - Simular el envío de nuevos paquetes a la aplicación web de recepción.

2.3. Objetivos Personales

- Realizar el desarrollo completo de un proyecto con sus diferentes apartados, mejorando así mis conocimientos sobre la gestión de proyectos.
- Aumentar mis conocimientos sobre los dispositivos IoT y el ecosistema que los rodea (TTN , LoRaWAN, ...)
- Aprender a desarrollar webs en Python mediante frameworks como Flask.
- Familiarizarme con el uso de software de control de versiones como Git y con Github la plataforma utilizada para alojar repositorios.

3. Conceptos teóricos

Para comprender correctamente las herramientas y técnicas utilizas para cumplir los objetivos mencionados anteriormente es necesario comprender teóricamente estos conceptos, las redes de dispositivos IoT, la transmisión de datos mediante protocolos LoRaWAN y la recogida y análisis de datos. Dichos conceptos pasaremos a explicarlos detalladamente a continuación:

3.1. Internet de las Cosas (IoT)

Definición

El **internet de las cosas (IoT)** [1] hace referencia a una red de dispositivos físicos interconectados capaces de recopilar, intercambiar y procesar datos a través de Internet sin intervención humana directa. Esta tecnología permite que objetos cotidianos como sensores de movimiento, termostatos, electrodomésticos, vehículos o sistemas industriales puedan responder al usuario de forma inteligente.

El concepto de IoT surge de la evolución de tecnologías como los chips de bajo consumo, las etiquetas RFID y las telecomunicaciones de alta velocidad, que han facilitado la miniaturización y el abaratamiento de los componentes electrónicos. Gracias a estos avances, actualmente es viable integrar esos sensores en los objetos de uso cotidiano.

Cómo funciona

Un sistema común de IoT funciona mediante la recopilación y el intercambio de datos en tiempo real.

Un sistema del IoT tiene tres componentes:

- **Dispositivo inteligente**: Se trata de un dispositivo, en nuestro caso el **MerryIoT MS10 Motion Detection** [11] , al que se le dotó de capacidades de computación. Recopila datos de su entorno o de las entradas de los usuarios y comunica los datos a través de Internet hacia la aplicación de IoT.

- **Aplicación de IoT**: Una aplicación de IoT es un conjunto de servicios y software que integra los datos recibidos de varios dispositivos de IoT. Nuestra aplicación utiliza esos datos recibidos para analizarlos y tomar una decisión referente así el paquete recibido puede o no a ver sido infectado[1].

- **Una interfaz de usuario gráfica**: El dispositivo de IoT puede administrarse a través de una interfaz de usuario gráfica. En nuestro caso un sitio web que puede utilizarse para analizar y gestionar archivos de datos recopilados por estos dispositivos.

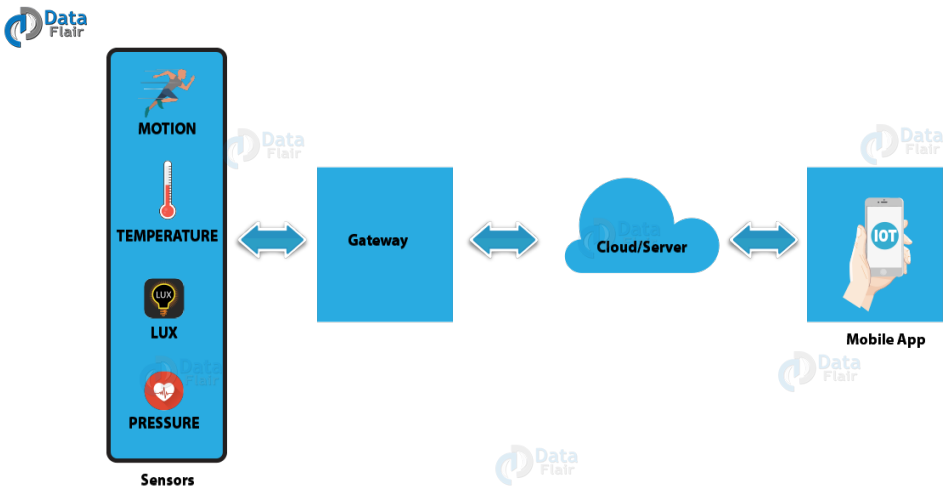

Figura 3.1: Diagrama de un sistema IoT (con soporte movil) [3]

[1]Se dice infectado de un paquete de datos recibido por la aplicación IoT, que tras ser analizado muestra características sospechosas que nos dan a entender que ese paquete puede ser fruto de un tercero con intenciones maliciosas

IioT o IoT industrial

El **Internet Industrial de las Cosas (IIoT)** [8] es el conjunto de sensores, instrumentos y dispositivos autónomos conectados a través de Internet a aplicaciones industriales. Esta red permite recopilar datos, realizar análisis y optimizar la producción, aumentando la eficiencia y reduciendo los costes del proceso de fabricación y prestación de servicios. Las aplicaciones industriales son ecosistemas tecnológicos completos que conectan dispositivos y a estos con las personas que gestionan los procesos en líneas de montaje, logística o distribución a gran escala.

La diferencia entre el Internet de las Cosas (IoT) y su versión industrial (IIoT) es que, mientras el IoT está enfocado a servicios para los consumidores, el IIoT se concentra en aumentar la seguridad y la eficiencia en los centros de producción. Aplicado a nuestro proyecto al medir factores como temperatura y humedad, podría aplicarse en un entorno referente a la industria de la agricultura, donde es relevante llevar un control de esos factores.

3.2. Redes LoRa y protocolo LoRaWAN

LoRa

Definición

LoRa (Long Range) [13] es una tecnología de modulación inalámbrica que destaca por su bajo consumo y gran alcance, haciéndola ideal para aplicaciones IoT. Sin embargo, LoRa por sí sola no especifica cómo gestionar la comunicación entre dispositivos.

¿Cómo funciona?

Una arquitectura de red de Lorawan consiste en nodos Lora, Puertas de enlace de Lora, el servidor de red y el servidor de aplicaciones.

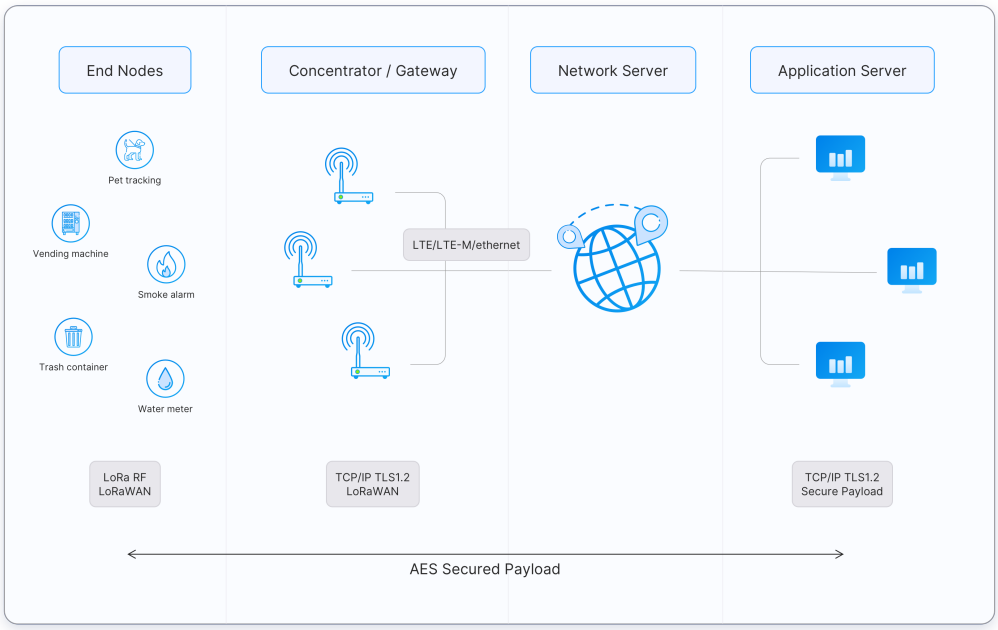

Figura 3.2: Arquitectura de una red LoRaWAN [17]

- **Nodos LoRa**: Los nodos finales son los elementos de la red Lora donde se realiza el control o la detección. Normalmente están a prueba de baterías y se encuentran remotamente. Los nodos finales envían datos a cada puerta de enlace en su vecindad y transmiten datos en periódico.

- **LoRa gateway**: La puerta de enlace recibe los datos de los nodos Lora End y luego los canaliza a un servidor de red. Una puerta de enlace de Lora generalmente consiste en un módulo de radio Lora, un microprocesador, y un medio de conectividad a Internet.

- **Servidor de red**: El servidor de red administra la red. Filma paquetes duplicados causados por múltiples puertas de enlace que reciben los mismos datos, realiza controles de seguridad, administra el tráfico y el enrutamiento de la puerta de enlace, controlar la tasa adaptativa, y reenvía mensajes al servidor de aplicaciones.

- **Servidor de aplicaciones**:El servidor de aplicaciones procesa datos del servidor de red, Analiza datos del sensor, admite funciones como visualización de estado y alertas en tiempo real, y opcionalmente puede enviar respuestas al nodo final.

LoRaWAN

Definición

LoRaWAN [14] es un protocolo de comunicación LPWAN diseñado para conectar dispositivos IoT con bajo consumo de energía y largo alcance. Como hemos mencionado antes, se basa en la tecnología LoRa, permitiendo la transmisión de datos a distancias de hasta 15 km en entornos rurales y 2-5 km en entornos urbanos.

Seguridad

La seguridad es un aspecto fundamental y al que le hemos dado mucha importancia durante el desarrollo de este proyecto, por ello voy a nombrar las principales medidas de seguridad de LoRaWAN:

1. **Cifrado de extremo a extremo**: Los datos se transmiten encriptados con AES-128, lo que garantiza su confidencialidad.
2. **Autenticación de dispositivos**: Cada sensor cuenta con una clave única para evitar accesos no autorizados [19].
3. **Gestión de claves de seguridad**: Se utilizan claves de sesión dinámicas para reducir el riesgo de ataques.

Sin embargo, como cualquier tecnología inalámbrica, LoRaWAN no es inmune a ataques. Es importante implementar buenas prácticas como el uso de firewalls, segmentación de redes y monitorización continua para detectar anomalías. En el caso de este proyecto, para combatir los posibles ataques se ha optado por la monitorización continua y detección de anomalías.

Diferencias entre LoRa y LoRaWAN

LoRa [13] describe la capa física inferior. LoRaWAN es un protocolo que describe las capas superiores de la red. LoRaWAN es un control de acceso a medios basado en la nube (MAC[2]) protocolo de capa, pero actúa principalmente como un protocolo de capa de red para gestionar la comunicación entre dispositivos de nodo final y puertas de enlace LPWAN, como protocolo de dirección, mantenido por la Alianza LoRa.

[2]Control de Acceso al Medio es una subcapa de la capa de enlace de datos en el modelo OSI,que regula cómo los dispositivos acceden y comparten un medio de transmisión común

LoRaWAN define la arquitectura del sistema y el protocolo de comunicación para la red, mientras que la capa física LoRa permite el enlace de comunicación de largo alcance.

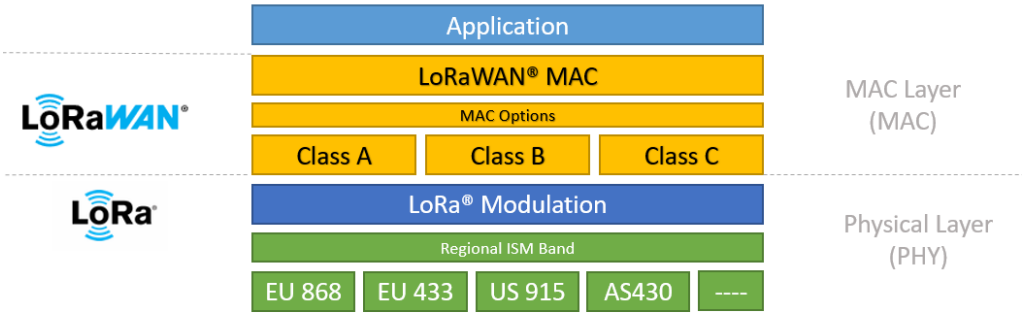

Figura 3.3: Diagrama de capas LoRaWAN [13]

Diferencias entre LoRa y otras tecnologías de transmisión inalámbricas

LoRaWAN es un protocolo de comunicación idóneo para paquetes útiles de pequeño tamaño (como datos de sensores) a largas distancias. La modulación LoRa proporciona un alcance de comunicación significativamente mayor con anchos de banda bajos en comparación con otras tecnologías inalámbricas de transmisión de datos. La siguiente imagen muestra algunas tecnologías de acceso que pueden utilizarse para la transmisión inalámbrica de datos, comparando su ancho de banda con su rango de transmisión.

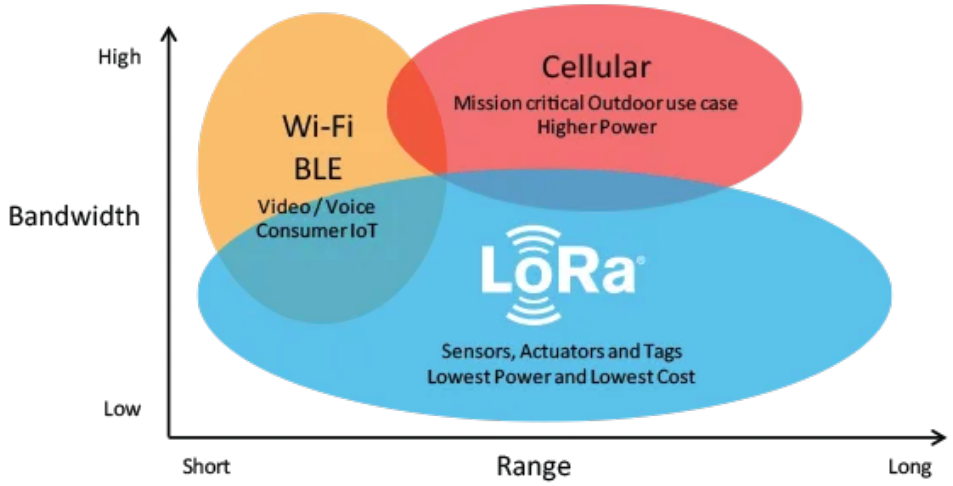

Figura 3.4: Gráfica comparativa de tecnologías de transmisión inalámbrica [18]

3.3. Estructura de los paquetes y análisis heurístico)

Estructura de los paquetes

Los paquetes enviados por el dispositivo MerryIoT, llegan codificados en un formato especificado en su manual de usuario [11], para poder utilizar esos datos es necesario decodificarlos con un payload[3] formatter.

[3]payload (o carga útil) es la parte del mensaje que contiene los datos de interés que se transmiten entre dos dispositivos, excluyendo los encabezados (headers), metadatos u otra información de control

Port			122		
Payload Length			9 bytes		
Bytes	0	1	2	34	5617
Field	Status	I Battery	Temp	RH Time Count	
Status	Sensors status				
	Bit [0]			1— occupied, 0 — free	
	Bit [1]			1— Button pressed, 0 – Button released	
	Bit [2] Bits (7:3)			1 —Tamper detected, 0 – No tamper detected RFU	
Battery	Battery level				
	Bits [3:0]			unsigned value v, range 0 — 15;	
				battery voltage in V = (21 + v) + 10.	
	Bits (7:4)			RFU	
Temp	Environment Temperature				
	Bits (7:0) sign integer temperature in °C				
RH	Relative humidity as measured by a digital sensor				
	Bits (6:0) Bit [7]			unsigned value in %, range 0-100. RFU	
Time	Time elapsed since the last event-triggered				
	Bits [15:01			unsigned value in minutes, range 0— 65,535.	
				`Note little-endian format.	
Count	Total count of event-triggered				
	Bits (23:0]			unsigned value, range 0— 16,777,215.	
				*Note little-endian format.	
	Note: This value is not stored persistently on the device, and may reset whenever the device i s power-cycled or rebooted.				

Figura 3.5: Formato del payload del dispositivo utilizado en el proyecto [11]

Una vez decodificados, se puede comenzar a analizar su contenido y extraer conclusiones del mismo.

Analisis del contenido

En este proyecto se ha determinado que la mejor forma para clasificar los datos recibidos es mediante el uso de reglas heurísticas generadas a partir de reglas de asociación y patrones observados durante el desarrollo del proyecto.

Reglas de asociación

Las Reglas de Asociación [16] son un método utilizado en minería de datos para descubrir relaciones significativas entre variables dentro de grandes conjuntos de datos. Estas reglas permiten identificar patrones frecuentes o comportamientos recurrentes, generalmente en forma de implicaciones del tipo:

Si A ocurre, entonces B tiende a ocurrir. (Ejemplo: Si un cliente compra pan, también compra jamón).

Para entender cómo se han obtenido estas reglas es importante aclarar antes unos conceptos:

- **Soporte**: Número de instancias que la regla predice correctamente.
  soporte(A –> C) = soporte(A U C)
- **Confianza**: Cociente entre el soporte y el número de instancias para las cuales la reglas es aplicable (el antecedente es cierto.).
  confianza(A –> C) = soporte(A U C)/ soporte(A)
- **Item**: un par de atributos/valores.
- **Item Set**: todos los items que aparecen en una regla.

Para obtener las reglas de asociación se busca superar un soporte y una confianza mínima.

La ventaja de los algoritmos de reglas de asociación sobre los algoritmos más estándar de árboles de decisión es que las asociaciones pueden existir entre cualquiera de los atributos. Un algoritmo de árbol de decisión generará reglas con una única conclusión, mientras que los algoritmos de asociación tratan de buscar muchas reglas, cada una de las cuales puede tener una conclusión diferente [9].

A diferencia de un árbol de decisión, el conjunto de reglas de asociación no se puede usar directamente para realizar predicciones del mismo modo que puede hacerse con un modelo estándar (como un árbol de decisión o una red neuronal). Esto se debe a las diversas conclusiones posibles que pueden derivarse de las reglas. Es necesario otro nivel de transformación para convertir las reglas de asociación en un conjunto de reglas de clasificación. Por tanto, las reglas de asociación producidas por algoritmos de asociación se conocen como modelos sin refinar.

Para refinar dicho modelo facilitado por las Reglas de asociación, utilizamos patrones identificados durante la recopilación de los datos.

Reglas heurísticas

Las reglas heurísticas son normas simples o criterios aproximados que se aplican para detectar comportamientos sospechosos o inusuales.Estas reglas se generan al analizar los patrones de comportamiento conocidos del dispositivo IoT.

El análisis heurístico es un proceso utilizado en campos como la seguridad informática (antivirus); por lo tanto, me parecía bastante adecuado para este proyecto [5].

Estas reglas heurísticas sumadas a las reglas de asociación conforman el conjunto encargado de analizar los datos recibidos y clasificarlos correctamente.

4. Técnicas y herramientas

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto.

4.1. Técnicas utilizadas

SCRUM

SCRUM es un marco de trabajo ágil utilizado principalmente en el desarrollo de software, divide el trabajo en ciclos cortos y regulares llamados "sprints", normalmente de 1 a 4 semanas, en los que se desarrolla un incremento funcional del producto.

Para obtener más información sobre la metodología Scrum utilizada durante el proyecto, consultar el apartado A.2.

4.2. Herramientas utilizadas

Hardware

Gateway Dragino LPS8N

El LPS8N [4] es una gateway LoRaWAN de código abierto. Permite enlazar una red inalámbrica LoRa con una red IP mediante WiFi o Ethernet. La tecnología inalámbrica LoRa permite a los usuarios enviar datos y alcanzar distancias extremadamente largas con tasas de transmisión de datos bajas.

Figura 4.1: Imagen del Gateway Dragino LPS8N [4]

Sensor MerryIoT MS10 Motion Detection

El MS10 es un sensor de movimiento que utiliza conectividad LoRaWAN para comunicar la presencia o ausencia de una persona, además de recopilar otros datos como temperatura y humedad. Su uso previsto es colocar el sensor con una buena vista de una habitación para detectar si hay movimiento o no en el área.

El sensor está compuesto por un detector infrarrojo pasivo (PIR) y una lente de Fresnel. El cuerpo principal contiene la electrónica activa necesaria para detectar movimiento y transmitir cualquier cambio a través de una red LoRaWAN.

También cuenta con detectores de vibración e inclinación en caso de manipulación. Una vez que se detecta un evento, el sensor enviará un mensaje ascendente (uplink).

Figura 4.2: Imagen del dispositivo MS10 [11]

Servicios en la nube y conectividad

The Things Network (TTN)/The Things Stack (Sandbox)

TTN es una infraestructura abierta para redes LoRaWAN. Permite registrar dispositivos, configurar gateways y gestionar los datos recibidos mediante webhooks, que envían automáticamente los paquetes recibidos hacia una URL determinada.

Figura 4.3: Logotipo de The Things Network

The Things Stack [20] es un servidor LoRaWAN de nivel empresarial (que incluye tanto las funciones de Servidor de red como de Servidor de aplicaciones mencionadas en la arquitectura de referencia de LoRaWAN).

Además, The Things Stack incluye servicios y herramientas para gestionar de forma segura millones de dispositivos LoRaWAN en entornos de producción.

En el proyecto se ha utilizado la variante The Things Stack Sandbox, una versión gratuita y pública del servidor The Things Stack, ofrecida por The Things Network (TTN), diseñada para que desarrolladores, estudiantes e investigadores puedan probar, experimentar y aprender a implementar soluciones LoRaWAN sin necesidad de infraestructura propia.

Figura 4.4: Logotipo de The Things Stack Sandbox

Cloudflare Tunnel

Cloudflare Tunnel [2] es una herramienta que te proporciona una forma segura de conectar tus recursos a Cloudflare sin necesidad de una dirección IP pública enrutable. Con Tunnel, no envías tráfico a una IP externa; en su lugar, un servicio ligero (daemon) llamado cloudflared, que se ejecuta en tu infraestructura, establece conexiones salientes únicamente hacia la red global de Cloudflare.

En este proyecto se ha utilizado para exponer de forma segura tu servidor Flask local al exterior, permitiendo que The Things Network (TTN) pueda enviar datos directamente a tu webhook sin necesidad de:

1. Configurar el router o abrir puertos manualmente.
2. Tener una IP pública fija
3. Contratar infraestructura externa (como un servidor en la nube)

Figura 4.5: Logotipo de Cloudflare

Entorno de desarrollo

IntelliJ IDEA

IntelliJ IDEA [10] es una herramienta que proporciona un entorno completo para el desarrollo de software, incluyendo herramientas para codificación, depuración, pruebas y despliegue.

Figura 4.6: Logotipo de IntelliJ IDEA

Docker/Docker Desktop

Docker [21] es un software de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de visualización de aplicaciones en múltiples sistemas operativos.

En este proyecto se ha utilizado tanto Docker como Docker Desktop para el despliegue de ambas aplicaciones web y de una base de datos. Docker Desktop es la aplicación nativa para ordenador diseñada por Docker para Windows y Mac, siendo la forma más sencilla de ejecutar, construir, depurar y probar aplicaciones dockerizadas.

Gracias al uso de Docker, se facilita el despliegue al usuario, evitando que deba preocupese por tener las dependencias necesarias o las versiones correctas de las herramientas utilizas para el correcto funcionamiento del proyecto.

Figura 4.7: Logotipo de Docker

Git/GitHub

Git [12] es un sistema de control de versiones distribuido. Estos repositorios locales plenamente funcionales permiten trabajar sin conexión o de forma remota con facilidad. Los desarrolladores confirman su trabajo localmente y, a continuación, sincronizan la copia del repositorio con la del servidor.

Figura 4.8: Logotipo de Git

GitHub [6] es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios para realizar proyectos. Los proyectos son almacenados en repositorios, permitiendo así:

- Presentar o compartir el trabajo.

- Seguir y administrar los cambios en el código a lo largo del tiempo.
- Dejar que otros usuarios revisen el código y realicen sugerencias para mejorarlo.

Figura 4.9: Logotipo de Github

Overleaf/LaTeX

Overleaf [24] es un editor colaborativo basado en la nube que se utiliza para escribir, editar y publicar documentos en LaTeX.

Figura 4.10: Logotipo de Overleaf

Draw.io

Draw.io es una herramienta gratuita y de código abierto para la creación de diagramas. Se utiliza ampliamente para diseñar diagramas de flujo, esquemas de red, diagramas UML, diagramas de arquitectura de software, organigramas y cualquier tipo de representación visual estructurada.

Figura 4.11: Logotipo de Draw.io

Backend

Python

Python [25] es un lenguaje de programación de alto nivel, conocido por su legibilidad y versatilidad. Se utiliza en una amplia gama de aplicaciones, desde desarrollo web hasta análisis de datos.

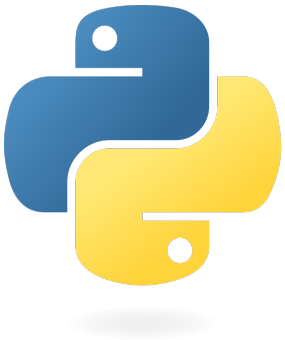

Figura 4.12: Logotipo de Python

Bibliotecas y frameworks de Python utilizados:

- Flask: Microframework web para Python que permite desarrollar aplicaciones web de forma sencilla y modular
- SQLAlchemy: facilita la interacción con bases de datos SQL mediante objetos Python.
- Pandas: permite usar estructuras eficientes como DataFrame, ideal para la lectura, manipulación y análisis de archivos CSV.

- Werkzeug: permite usar herramientas para construir aplicaciones compatibles con WSGI, un estándar para la comunicación entre servidores web y aplicaciones Python.
- Requests: permite realizar peticiones HTTP desde el backend.
- Pytz: permite gestionar zonas horarias de forma precisa
- Csv: permite la lectura y escritura de archivos CSV.
- Os: permite interactuar con el sistema operativo (crear carpetas, comprobar rutas, acceder a archivos...).
- Datetime: permite manejar fechas y horas, como la marca temporal de los paquetes.
- Traceback: permite imprimir mensajes de error con más detalles.
- Random: permite generar datos aleatorios.
- Functools: permite funciones de orden superior y operaciones sobre objetos invocables.
- mlxtend: permite complementar otras herramientas añadiendo funcionalidades útiles para campos como la minería de datos, en este caso con la generación de reglas de asociación.

Frontend

HTML5/JavaScript

HTML [22] es el lenguaje estándar para la creación de páginas web. Define la estructura básica de los contenidos de un sitio web mediante etiquetas, que permiten organizar elementos como textos, enlaces, imágenes, tablas, formularios y otros componentes visuales.

Figura 4.13: Logotipo de HTML 5

JavaScript [23] es un lenguaje de programación interpretado, orientado a objetos y basado en prototipos, diseñado originalmente para ejecutarse en navegadores web. Su propósito principal es añadir interactividad y dinamismo a las páginas web, permitiendo modificar el contenido, responder a eventos del usuario, validar formularios, actualizar datos en tiempo real sin recargar la página.

Librerías y frameworks de Html utilizados:

- Bootstrap 5: es un framework de CSS y JS que facilita la creación de diseños responsivos y modernos con componentes predefinidos (botones, formularios y tablas).
- Chart.js: es una biblioteca de JavaScript para generar gráficos interactivos.
- Pandas: permite usar estructuras eficientes como DataFrame, ideal para la lectura, manipulación y análisis de archivos CSV.
- Jinja2: es un motor de plantillas utilizado por Flask para insertar lógica (bucles, condiciones) y datos dinámicos en los archivos HTML.

5. Aspectos relevantes del desarrollo del proyecto

Este apartado recoge los aspectos más interesantes del desarrollo del proyecto y explica los detalles de mayor relevancia de las fases de análisis, diseño e implementación.

5.1. Despliegue de una red LoRaWAN

El primer paso en el desarrollo del proyecto consistió en la creación y configuración de una red LoRaWAN, con el objetivo principal de familiarizarme con el ecosistema de los sistemas IoT y, al mismo tiempo, establecer un flujo constante y fiable de datos enviados por un dispositivo IoT.

Estos datos serían posteriormente almacenados y estructurados en un dataset, que actuaría como pilar fundamental para el desarrollo y validación de las siguientes partes del proyecto. Por este motivo, era especialmente importante asegurar que esta primera parte se realizase de forma correcta.

A continuación voy a describir los diferentes pasos necesarios para el despliegue de una red LoRaWAN:

Configuración del gateway Dragino

La recepción y retransmisión de paquetes LoRaWAN en este proyecto se realizó a través del Gateway Dragino LPS8. Este gateway tiene como principal función recibir datos de dispositivos IoT LoRa–en el contexto de este proyecto, el dispositivo MerryIoT MS10– y después transmitirlos a través de Internet a un Network Server, en este caso, The Things Network.

Para configurar dicho dispositivo, lo primero sería asegurarse de que esté conectado a una fuente de alimentación y que tenga acceso a internet. En este caso, se le ha conectado vía Ethernet al router de mi entorno de trabajo.

Lo siguiente sería acceder a la interfaz web de configuración del dispositivo, dicha interfaz web puede encontrarse en la dirección http://IPADDRESS:8000 donde IPADDRESS es la dirección IP asignada al gateway por el router al que esté conectado. Antes de poder configurar la interfaz web te solicitará ingresar usuario y contraseña.

User Name: root

Password: dragino

Una vez dentro, será necesario configurar dos apartados para asegurar el correcto funcionamiento de la futura conexión con TTN.

El primero referente a permitir la conexión del gateway con el Network Server.

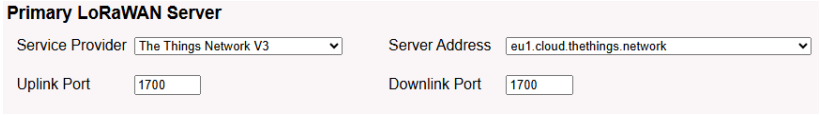

Figura 5.1: Configuración del gateway

El segundo referente a la configurar el correcto plan de frecuencia entre el sensor y el gateway.

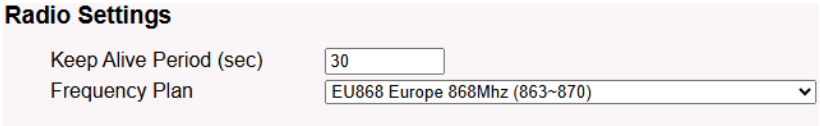

Figura 5.2: Frecuencia del gateway

Si has seguido correctamente estos pasos ya solo queda registrar ambos dispositivos IoT en TTN y observar el flujo de datos entre ellos.

Configuración y registro en The Things Network (TTN)

En este proyecto se ha optado por utilizar The Things Stack Sandbox(TTSS) para monitorizar los dispositivos IoT, ya que permite registrar dispositivos y observar el tráfico de red en tiempo real de forma gratuita.

A continuación se explicará el proceso seguido para lograr registrar correctamente tanto el gateway como el sensor en TTSS.

Una vez tienes creada tu cuenta en TTSS, primero debes elegir la opción register gateway. Esta opción, lo primero que te va a pedir es el Gateway EUI (se puede obtener en la interfaz web del gateway, en la pestaña referente a la configuración LoRaWAN), nombrar el gateway como el usuario desee y, por último, establecer un plan de frecuencia concorde con el configurado en la interfaz web.

Si has seguido los pasos correctamente en TTSS debería verse algo similar a esto:

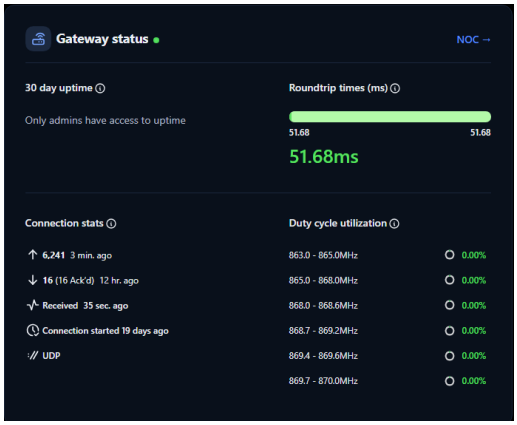

Figura 5.3: Estado del gateway

En TTSS es necesario crear una aplicación para poder gestionar los sensores. Solo se necesita elegir la opción Add Application y escoger un ID único y un nombre a elección del usuario.

Con la aplicación creada se selecciona la opción Add end device y se elige el método de añadir las especificaciones de forma manual. Lo siguiente es introducir el mismo plan de frecuencia que en el gateway y las versiones correspondientes de LoRaWAN y parámetros al sensor a utilizar, en el caso del sensor del proyecto dichas versiones son LoRaWAN Specification 1.0.4 y RP002 Regional Parameters 1.0.3. Por último te pide introducir DevEUI, AppEUI, y AppKey, parámetros que vienen incluidos en la etiqueta del sensor o en la documentación del fabricante.

Si todo ha ido bien, ya podríamos empezar a observar el flujo de datos entre nuestros dispositivos IoT.

Pruebas iniciales para observar el flujo de datos

Gracias TTSS podemos ver como interactúan los dispositivos IoT en la consola Live data, aunque todavía no podemos sacar información relevante debido a que nos falta por implementar un payload formatter, un pequeño fragmento de código capaz de decodificar el payload que envía el sensor IoT al gateway.

El payload formatter es diferente para cada sensor y debe ser creado a partir del payload de ejemplo mostrado en el manual de usuario del sensor [11].

Formatter code*

```
function decodeUplink(input) {
  var bytes = input.bytes;

  // Status byte
  var occupied = (bytes[0] & 0x01) ? true : false;
  var buttonPressed = (bytes[0] & 0x02) ? true : false;
  var tamperDetected = (bytes[0] & 0x04) ? true : false;

  // Battery
  var batteryLevel = bytes[1] & 0x0F;
  var batteryVoltage = (21 + batteryLevel) / 10.0;  // Voltios

  // Temperature
  var temperature = (bytes[2] > 127) ? bytes[2] - 256 : bytes[2];

  // Humidity
  var humidity = bytes[3] & 0x7F;  // solo 7 bits válidos (según manual

  // Time since last event (little endian)
  var timeMinutes = bytes[4] | (bytes[5] << 8);

  // Event count (little endian)
  var eventCount = bytes[6] | (bytes[7] << 8) | (bytes[8] << 16);

```

Figura 5.4: Codigo del payload formatter

Una vez aplicado el payload formatter la información útil que podemos sacar del mensaje enviado por el sensor se ve así:

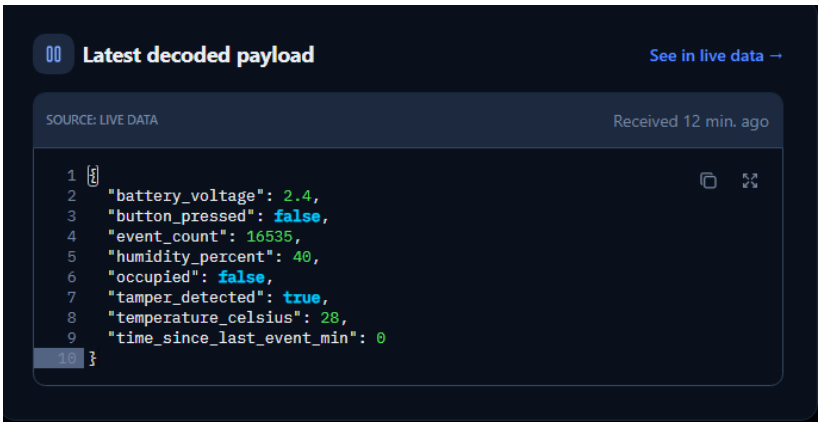

Figura 5.5: Payload decodificado

5.2. Desarrollo del analizador y receptor de paquetes

Una vez configurada la red LoRaWAN y comprobado el correcto flujo de datos entre los dispositivos, el siguiente paso era buscar cómo almacenar todos esos datos de forma segura para crear un dataset el cual analizar en el futuro.

Se estudiaron diversas formas de llevar a cabo este proceso, desde servicios web relacionados con bases de datos hasta almacenarlos de forma local, pero todas las opciones pasaban por configurar un webhook en TTSS.

Webhook

Primero voy a explicar qué es y comó se configura un webhook en TTSS:

Un webhook es un mecanismo de callback HTTP. Dicho mecanismo enviará automáticamente una petición HTTP POST a una URL específica cada vez que un dispositivo envíe datos (uplink).

En tu aplicación de TTSS, elege el apartado webhooks y selecciona Add webhook. Escribes un ID único y escoges JSON como formato. En la url base escribes la dirección que quieres que reciba los datos. En el caso del proyecto, como solo nos interesa recibir los datos (uplink), en el apartado Enabled event types marcamos Uplink message y añadimos /ttn. Así cada vez que el webhook quiera enviar un mensaje a la url base, lo hará al apartado /ttn.

Si has seguido los pasos correctamente, el webhook debería verse así:

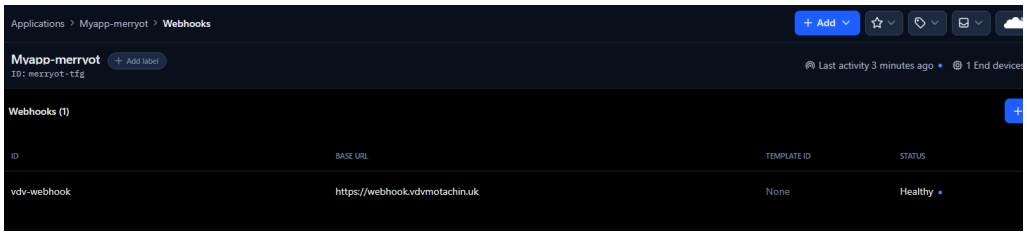

Figura 5.6: Imagen webhook TTSS

En este proyecto se optó por redirigir los mensajes a un servidor propio en vez de guardarlos en la nube, para facilitar el análisis que se llevaría a cabo posteriormente.

Para realizar este proceso se decidió usar un webhook de TTSS y Cloudflare Tunnel como método para exponer de forma segura el servidor local al exterior.

Gracias a realizarlo de esta forma, fue posible exponer el servidor Flask local sin necesidad de modificar la configuración del router, garantizar seguridad en la transmisión (ya que Cloudflare proporciona un túnel seguro y cifrado) y configurar de forma sencilla el flujo de datos.

Función de recibir y almacenar

Una vez terminado todo este proceso, era necesario crear una aplicación web en Flask capaz de recibir y almacenar los datos. Se optó por que fueran almacenados en un archivo csv. A futuro, cuando ya se habían almacenado suficientes datos y se disponía de un dataset de tamaño considerable, se procedió a añadir la función de analizar los paquetes de datos recibidos, pero primero nos centraremos en las funciones de recibir y almacenar.

La aplicación Flask se diseñó para exponer un endpoint HTTP (/ttn) accesible públicamente gracias al túnel de Cloudflare. Este endpoint es el receptor directo de los paquetes LoRaWAN enviados desde TTSS.

Cada paquete recibido contiene una estructura JSON con múltiples parámetros. Los datos son extraídos del decoded payload proporcionado por TTN y estructurados en un diccionario interno para luego ser almacenados en un archivo CSV local como una nueva fila. Dicho archivo es el que será utilizado como dataset durante todo el proyecto.

Elección de método para clasificar

En dicho dataset los paquetes de datos tienen un atributo denominado estado el cual al comienzo del proyecto se asumía que su valor era real". Cuando se disponían de suficientes paquetes como para tratar de automatizar un proceso capaz de analizar y determinar de forma correcta el valor del atributo estado de cada paquete, comencé a plantearme cuál sería el mejor método para realizar el análisis y la clasificación correspondiente.

Algunas de las opciones que fueron descartadas, pero que se plantearon son:

Modelos de aprendizaje automático supervisado: Se contempló el uso de algoritmos como árboles de decisión. Sin embargo, esta opción fue descartada ya que el dataset del que se disponía para entrenar el modelo no era el adecuado. Este dataset, creado a partir de los paquetes recibidos de un dispositivo IoT, tiene ciertas peculiaridades que imposibilitan el uso de esta técnica. Por poner un ejemplo, el modelo daba mucho peso al voltaje recibido a la hora de determinar si un paquete era real o falso. Como todos los paquetes del dataset habían sido enviados desde el mismo dispositivo, todos compartían el mismo voltaje. Por lo tanto, el modelo asumía incorrectamente que los paquetes reales debían tener un valor específico de voltaje, ignorando el resto de atributos del paquete, igual o más importantes. Dicho modelo podía llegar a clasificar como real un paquete que marcara 100 grados, siempre que su voltaje fuera 2.4, debido a las asociaciones incorrectas que estableció por contar con un dataset no lo suficientemente amplio ni variado, como requieren este tipo de modelos.

Detección de anomalías sin supervisión (clustering): Técnicas como k-means o algoritmos de aislamiento fueron consideradas como posibles herramientas para detectar desviaciones. No obstante, requerían una preconfiguración compleja y presentaban un alto riesgo de falsos positivos ante pequeñas variaciones legítimas.

Finalmente, se optó por un enfoque más sencillo, transparente y controlado: una combinación de reglas heurísticas definidas manualmente junto a un sistema complementario de reglas de asociación extraídas con ayuda de la librería mlxtend.

Función de clasificar/analizar

Las reglas heurísticas permitieron codificar conocimientos específicos sobre el comportamiento esperado del dispositivo, como los rangos normales de temperatura y humedad. En cambio, las reglas de asociación ofrecieron

una forma de complementar este sistema con correlaciones frecuentemente observadas entre atributos en paquetes etiquetados previamente, como las condiciones de detección de presencia.

Este método ofrece ciertas ventajas tales como:

- Fácil ajuste y mantenimiento: permite incorporar nuevas condiciones o relajar otras en función de nuevas observaciones.
- No requiere entrenamiento previo ni datasets amplios y variados.
- Robustez y eficacia suficiente para el entorno planteado por el proyecto.

Para obtener las reglas de asociación, se decidió tener en cuenta únicamente aquellas que superaran un 15 % de soporte, es decir, que la regla predijera correctamente al menos un 15 % de las instancias del dataset, y un 90 % de confianza, es decir, que de todas las instancias en las que la regla es aplicable, al menos sea capaz de predecir correctamente el 90 % de los casos.

```
# Generar itemsets frecuentes
frequent_itemsets = apriori(df_binarized, min_support=0.15, use_colnames=True)

# Generar reglas de asociación con confianza mínima de 0.9
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.9)
```

Figura 5.7: Fragmento de código encargado de determinar el soporte y confianza

Tras ejecutar ese código, las reglas obtenidas son:

```
 Reglas con confianza >= 0.9:
            antecedents           consequents  support  confidence
0  (tiempo_espera_largo)     (No hay presencia)    0.184    0.978723
1        (Hay presencia)  (tiempo_espera_corto)    0.760    0.994764
2  (tiempo_espera_corto)        (Hay presencia)    0.760    0.935961

 Traducción de reglas a condiciones Python:
if d["tiempo_espera_largo"] == True:  # soporte: 0.18, confianza: 0.98
    # entonces: d["No hay presencia"] == True

if d["Hay presencia"] == True:  # soporte: 0.76, confianza: 0.99
    # entonces: d["tiempo_espera_corto"] == True

if d["tiempo_espera_corto"] == True:  # soporte: 0.76, confianza: 0.94
    # entonces: d["Hay presencia"] == True
```

Figura 5.8: Reglas de asociación obtenidas

Si el tiempo de espera es largo, entonces No hay presencia. Traducido a nuestro dataset, es si time_since_last_event_min es mayor que 1, entonces tamper_detected es false. Esto se cumple en los paquetes reales.

Si el tiempo de espera es corto, entonces hay presencia. Traducido a nuestro dataset es si time_since_last_event_min es menor o igual que 1 y tamper_detected es true, el paquete es real.

Gracias a esas reglas podemos determinar que, si Hay presencia y tiempo de espera es largo, el paquete no va ser real.

Al principio, se podría interpretar que otra posible regla sería: si no hay presencia y el tiempo de espera es corto, entonces el paquete es falso. Sin embargo, esto no es aplicable, ya que si se observan las reglas de asociación obtenidas, la regla si no hay presencia, entonces el tiempo de espera es largo no aparece. Esto se debe a que no cumple con el soporte o la confianza mínimos, lo cual indica que existen más casos de los permitidos en los que no hay presencia y, aun así, el tiempo de espera es corto. Por lo tanto, no podemos asegurar que un paquete sin presencia vaya necesariamente a tener un tiempo de espera largo. Esto hace posible la existencia de paquetes reales con tiempo de espera corto y sin presencia, lo que desmiente la regla asumida inicialmente.

Las reglas heurísticas elegidas para acompañar a las reglas de asociación antes mencionadas son:

- El voltaje de la batería no puede superar 3V ni puede ser menor 2.4V como determina el manual de usuario del sensor.
- La temperatura medida no puede ser superior a 50 grados ni inferior a 0 grados.
- La humedad no puede ser superior a 100 % ni inferior a 0 %
- El atributo ocupado no puede ser True

Por último se añadió una última regla algo más compleja que las demás, ya que no depende solo del paquete recibido sino que depende también del último paquete real almacenado. Si ambos paquetes no detectan presencia, se comprueba que el tiempo que haya pasado entre ellos sea de aproximadamente una hora, que es el periodo de tiempo que tarda en enviar un nuevo paquete el sensor si no detecta ninguna presencia.

```
if (
        paquete_actual['tamper_detected'] == "False" and
        paquete_anterior['tamper_detected'] == "False"
):

    try:

        t_actual = datetime.fromisoformat(paquete_actual['timestamp'].replace("Z", "+00:00"))
        t_anterior = datetime.fromisoformat(paquete_anterior['timestamp'].replace("Z", "+00:00"))
        diferencia_min = abs((t_actual - t_anterior).total_seconds() / 60)
        tolerancia = 2

        if diferencia_min >= tolerancia:
            resto = diferencia_min % 60
            if not  resto <= tolerancia or resto >= (60 - tolerancia) :
                sospechoso = True
                razones.append("Diferencia horaria incorrecta")
        else:
            sospechoso = True
            razones.append("Diferencia horaria incorrecta")


    except Exception as e:
        razones.append(f"Error evaluando timestamps: {e}")
```

Figura 5.9: Fragmento de código referente a la ultima regla añadida

Por ultimo, una vez que el código evalúa el nuevo paquete con todas las reglas mencionadas, determina si es un paquete infectado o si es un paquete real y procede a almacenar en el archivo del dataset con el atributo estado determinado tras el análisis.

5.3. Desarrollo de la aplicación web enfocada al usuario

Una vez terminada la infraestructura necesaria para la recepción y análisis automático de los datos recibidos, el siguiente paso del proyecto consistió en diseñar y desarrollar una segunda aplicación web, esta vez centrada en la interacción directa con el usuario. Esta nueva interfaz tiene como objetivo proporcionar herramientas visuales e intuitivas que permitan gestionar, visualizar, analizar y simular datos provenientes del dispositivo IoT.

Esta aplicación web se desarrolló también con Flask y se desarrolló en torno a cumplir una serie de funcionalidades que permiten al usuario:

- Gestionar distintas cuentas diferenciadas entre si.
- Visualizar las gráficas de temperatura y humedad basadas en los datos almacenados.

- Gestionar distintos ficheros CSV (subir, eliminar, ver, descargar o analizar).
- Enviar paquetes personalizados o aleatorios al receptor para simular tráfico IoT.

Creación de usuarios

Lo primero que te encuentras al intentar acceder a esta aplicación web es una pestaña de login, esto se debe a que la aplicación cuenta con un login segurizado, que se caracteriza por no permitir al usuario acceder a ninguna funcionalidad de la aplicación, hasta que no haya iniciado sesión, permitiéndole solo iniciar sesión, registrarse o recuperar su contraseña. Evitando así posibles problemas generados al intentar usar las funcionalidades de la aplicación sin tener una cuenta asociada.

Al principio del proyecto se planteó la opción de desarrollar la aplicación sin la posibilidad de que hubiera diferenciación entre usuarios, pero esto provocaba más problemas que soluciones al complicar mucho más las funcionalidades referentes a la gestión de archivos.

Entonces se decidió implementar una pequeña base de datos encargada de almacenar a los usuarios con sus contraseñas y así separar las acciones de un usuario de las de otro.

Inicio de Sesión

Referente al inicio de sesión , la aplicación obliga al usuario a rellenar ambos campos (Usuario, Contraseña), para evitar fallas de seguridad y notifica al usuario si, al intentar iniciar sesión, el usuario introducido no existe o no concuerda con la contraseña introducida.

Si no se desea crear una cuenta, se puede iniciar sesión con Usuario: admin, contraseña:admin.

Registro

Referente al registro, aclarar que no se permite que existan dos usuarios con el mismo nombre. Se notifica al usuario si intenta registrarse con un usuario ya existente. Una vez creado el nuevo usuario, se le asigna una carpeta dentro del directorio csv del proyecto, donde se guardarán sus archivos futuros.

Recuperar Contraseña

Se planteó la posibilidad de que el usuario se olvidara de su contraseña. Para solucionar este caso, se optó por pedir al usuario que aportara el

nombre de uno de los ficheros csv que se encuentre en su carpeta personal. Si lo aporta, se le permite restablecer la contraseña.

Visualizar gráficas

Esta fue la primera funcionalidad que se planteó para la aplicación web ya que mi objetivo inicial era simplemente mostrar al usuario las gráficas referentes a los datos recopilados (Temperatura y Humedad) por el sensor IoT.

Con el transcurso del desarrollo del proyecto se optó por permitir al usuario interactuar mucho más con esta funcionalidad, desde que pueda elegir si solo mostrar en la gráfica los paquetes con el atributo estado igual a real, o permitir que visualice en la gráfica cualquier archivo csv que haya subido a la aplicación.

Todo este apartado se desarrolló gracias Chart.js que permite generar gráficos interactivos en HTML.

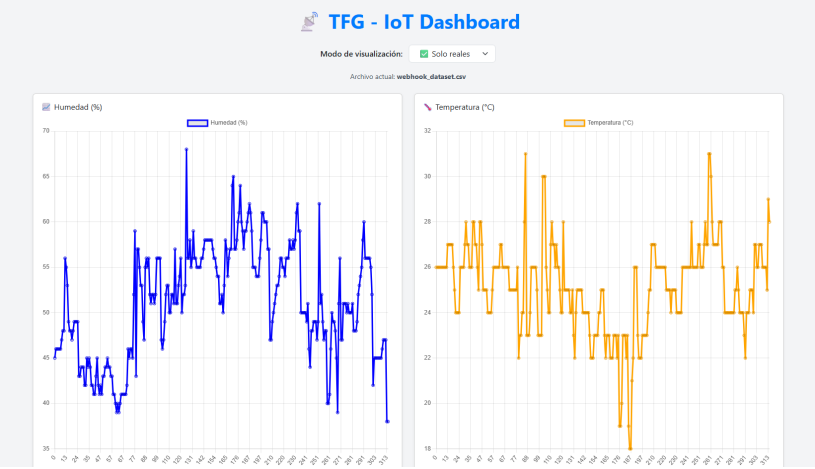

Figura 5.10: Interfaz gráfica referente a visualizar gráficas

En la siguiente imagen se observa la posibilidad de que el usuario visualice las gráficas correspondientes a un archivo csv propio.

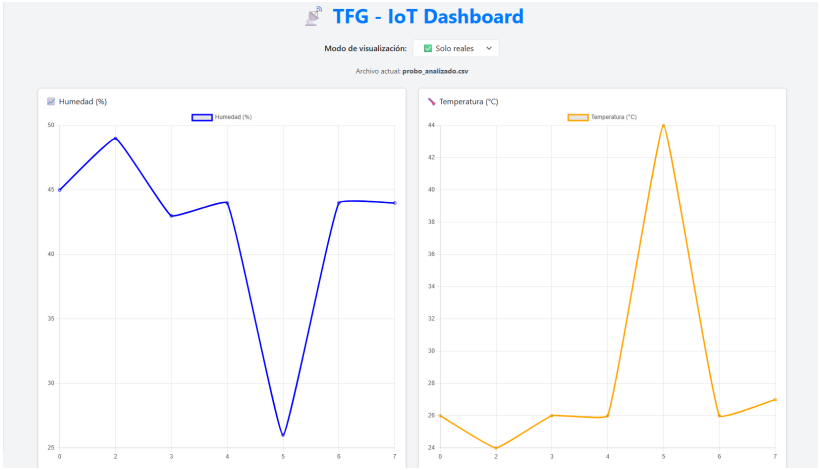

Figura 5.11: Visualizar gráficas de un archivo diferente al dataset

Gestionar ficheros CSV

Esta funcionalidad es la más extensa del proyecto debido a la variedad de opciones que permite al usuario realizar.

Explicaré brevemente las funcionalidades más simples:

- Visualizar contenido del archivo.
- Aplicar el archivo para que sea visible en el apartado de las gráficas.
- Eliminar el archivo.
- Descargar el archivo.

Las siguientes dos funcionalidades son analizar fichero y añadir fichero. Aunque puedan sonar similares, son funcionalidades completamente distintas.

Añadir fichero

Una vez implementada la posibilidad de diferenciar entre usuarios, era necesario implementar la posibilidad de que cada usuario pudiera interactuar con los archivos csv que deseara. Por ello, se desarrolló esta funcionalidad, la cual permite al usuario añadir un fichero que soporte la aplicación (Misma estructura que el fichero csv webhook_dataset) para interactuar con él de todas las formas mencionadas anteriormente.

Para facilitar al usuario completar esta tarea, se desarrolló un selector de archivos que solo muestra archivos csv. En caso de que el usuario intente

subir un fichero que no pueda ser soportado por la aplicación, se le notificará al instante cuál ha sido el error.

Aun así, esta funcionalidad tiene una exigencia adicional y es que el fichero a subir ya haya sido analizado previamente, es decir, que el archivo ya cuente con el atributo estado.

Lo cual puede ser un problema si el usuario no cuenta con una aplicación como la desarrollada en este proyecto, capaz de analizar ficheros csv y añadir el atributo estado.

Es por eso que el siguiente paso en el desarrollo del proyecto fue crear la función analizar fichero.

Analizar fichero

Como he mencionado antes, esta función se desarrolla para complementar a la función Añadir fichero, permitiendo así a los usuarios que desean subir sus ficheros csv a la aplicación web, pero no disponen de una herramienta capaz de analizarlos y clasificarlos en reales o infectados.

En términos de funcionamiento, es muy similar a añadir fichero, permitiendo solo analizar ficheros csv que tengan la misma estructura que el fichero webhook_dataset solo que en esta ocasión también permite que no tengan la última columna correspondiente al atributo estado ya que esta columna es la que va a añadir esta función.

Para realizar el análisis, importa la función de análisis correspondiente a la aplicación webhook y la aplica sobre todas las filas del fichero a analizar. Una vez termina, lo añade a la carpeta personal del usuario con el mismo nombre del fichero entregado pero añadiendo el sufijo _analizado para aclarar al usuario que ese fichero ha sido generado de esta forma.

Infectar

Para terminar, la última función implementada a la aplicación es infectar, surge de plantearse la posibilidad de que el usuario no tenga los medios para crear una red LoRaWAN pero aun así quiera disfrutar de la funcionalidad de la aplicación encargada de recibir y almacenar los paquetes de datos.

Para ello se crea la posibilidad de que el usuario desde la interfaz web pueda crear sus propios paquetes de datos ya sea de forma manual o de forma aleatoria y luego enviarlos a la aplicación webhook, la cual tratará esos paquetes como si hubieran sido enviados desde TTSS y los analizará y almacenará en el archivo correspondiente al dataset (webhook_dataset).

Permitiendo así al usuario probar el funcionamiento de la función de análisis de datos asociada al dataset.

A continuación se muestra la imagen de un paquete de datos generado aleatoriamente, listo para ser enviado.

Figura 5.12: Imagen visual de la interfaz web referente a la función infectar

6. Trabajos relacionados

Actualmente, el Internet de las Cosas (IoT) es un paradigma tecnológico en constante expansión que permite la conexión de dispositivos físicos a través de redes inalámbricas para la recogida, transmisión y análisis de datos en tiempo real.

Por eso mismo, garantizar que la comunicación establecida en esa red es segura y robusta es un trabajo altamente solicitada y con propuestas muy diversas para llevarla acabo.

En este apartado voy a mencionar tres trabajos relacionados con este campo, que considero importante comentar:

6.1. Desarrollo de un prototipo de modulo de análisis de trafico basado en wireshark para detección de ataques de denegación usando inspección de tramas lorawan que provea una capa de integración api rest

Este proyecto [26] se centra en la creación de un sniffer capaz de capturar y analizar paquetes LoRaWAN con el objetivo de detectar ataques de denegación de servicio (DoS/DDoS). A través del uso de herramientas como Wireshark, Pyshark y Tshark, así como del desarrollo de una API REST mediante Django, el autor consigue construir un sistema que no solo captura

y decodifica paquetes LoRaWAN, sino que también los almacena en una base de datos accesible para futuras evaluaciones de seguridad.

La propuesta de realizar una inspección de tramas en el entorno de red, es sin duda una técnica que podría aplicarse como capa adicional en mi sistema para mejorar las capacidades de diagnóstico y detección temprana de amenazas dentro del ecosistema LoRaWAN.

6.2. Trend Micro Finds LoRaWAN Security Lacking, Develops LoRaPWN Python Utility

Este proyecto [7] desarrolló LoRaPWN, una herramienta en Python diseñada para funcionar con radios definidas por software (SDR) compatibles con GNU Radio. LoRaPWN permite capturar, descifrar y generar paquetes LoRaWAN (versiones 1.0 y 1.1), romper (brute-force) el procedimiento OTAA, descifrar cargas útiles y los campos MIC, así como simular ataques DoS mediante replays y manipulaciones de paquetes.

Este proyecto está especialmente relacionado con el mío, ya que en mi proyecto se utiliza el procedimiento OTAA para unir el dispositivo IoT con TTSS. Sería interesante ver cómo respondería mi proyecto a un intento de ataque a través del método desarrollado por este artículo, y desarrollar posibles defensas ante este tipo de ataques.

6.3. Detecting IoT device compromise using Python

El objetivo principal de este proyecto [15] es desarrollar una herramienta en Python capaz de analizar tráfico de red IoT a través de archivos .pcap, con el fin de detectar y modelar flujos de comunicación entre dispositivos. Esta herramienta permite extraer información útil sobre el comportamiento de los dispositivos conectados a una red local, facilitando tareas de análisis de seguridad, identificación de patrones anómalos y caracterización de dispositivos IoT.

Esta solución permite identificar posibles vectores de ataque, caracterizar el comportamiento normal y anómalo, y servir como base para futuras estrategias de detección y mitigación de amenazas en entornos IoT que podrían aplicarse al proyecto que he desarrollado, mejorando así su seguridad.

7. Conclusiones y Líneas de trabajo futuras

7.1. Conclusiones

tras finalizar este proyecto y observar todo lo realizado en el mismo, puedo concluir que se han cumplido los dos objetivos generales que se tenían al comienzo del proyecto.

Se ha logrado configurar y crear una red LoRaWAN capaz de mantener un flujo real de datos y de conectar dicha red con una aplicación web capaz de recibir, analizar y almacenar esos datos para su posterior uso.

Además se ha logrado desarrollar una aplicación web completa, intuitiva y funcional que ayude a los usuarios no técnicos a interactuar con datos recopilados de dispositivos IoT e introducirles de forma sutil, como es el proceso de análisis y clasificación de datos con el objetivo de detectar ataques o brechas de seguridad en la red.

Aun así creo que esa parte es la que más margen de mejora tiene con respecto a lo desarrollado en el proyecto, ya que los métodos referentes a mejorar la seguridad de los entornos IoT es un campo de la informática que actualmente está en auge y queda mucho por mejorar. Desde la generación de conjuntos de datos de calidad y variados para ser usados en la búsqueda de patrones o técnicas de aprendizaje automático hasta mejorar la robustez de los medios de comunicación utilizados para transmitir los paquetes de datos.

Una vez terminado el proyecto, puedo afirmar que este me ha ayudado a darme cuenta de varias cosas:

La importancia de la base teórica y práctica adquirida durante todo el transcurso del grado de ingeniería informática, ya que muchas cosas realizadas durante el proyecto son temas que ya había tratado con anterioridad en el grado.

La ventaja que ofrece el tener claro desde el inicio del proyecto los objetivos que deseas cumplir y las tecnologías que se planean utilizar para cumplirlos.

Por último, agradecer al desarrollo de este proyecto el haber aumentado mis habilidades tanto en el ámbito técnico a la hora de desarrollar software y trabajar con hardware como son los dispositivos LoRaWAN como en el ámbito organizativo a la hora de gestionar el desarrollo de un proyecto de esta magnitud y documentar todo lo relacionado con su desarrollo.

7.2. Líneas de trabajo futuras

Como ya había mencionado anteriormente, y aun reconociendo los logros alcanzados con este proyecto, puedo asegurar que todavía hay muchos aspectos del mismo que se pueden mejorar o nuevas funcionalidades que no se habían planteado y que podría ser interesante tener en cuenta:

- **Mejora del dataset**: Ya he mencionado anteriormente los principales problemas referentes al dataset utilizado en el proyecto, por eso mismo un trabajo a futuro podría ser el mejorar la calidad del dataset generado haciéndolo mas amplio y completo. Facilitando así la tarea referente a analizar correctamente los datos recibidos.

- **Exploración de diferentes métodos de clasificación**: En este proyecto se ha optado por utilizar un método de clasificación que no tiene por que ser el mas adecuado para este trabajo, por ello estudiar diferentes opciones y probar si son mejores que la utilizada actualmente, mejorando así los resultados de clasificación del proyecto es una linea de trabajo a futuro completamente valida.

- **Generalización en el ámbito hardware**: Actualmente el proyecto desarrollado esta centrado y focalizado en dar soporte al sensor MerryIoT Motion Detection MS10, por lo tanto a futuro se podría plantear la idea de generalizar el proyecto para que sea capaz de soportar el flujo de datos de cualquier tipo de sensor IoT.

- **Análisis del flujo de datos recibido**: Este apartado se ha mencionado en trabajos relacionados y es la posibilidad de implementar al proyecto la capacidad de analizar el trafico de red IoT con el fin de detectar posibles vectores de ataque o comportamientos anómalos, mejorando así tanto la seguridad como el analizador del proyecto.
- **Creación de una interfaz referente a la red IoT**: Este apartado esta relacionado con el anterior y consiste en plantearse crear una interfaz que pueda usar el usuario, como hace en la segunda aplicación web pero referente a la primer aplicación encargada de analizar y recibir los datos, añadiendo funcionalidades adicionales como puede ser mostrar en tiempo real un grafo de nodos con el flujo de datos entre ellos.

Bibliografía

[1] Amazon Web Services. ¿Qué es el Internet de las cosas (IoT)? https://aws.amazon.com/es/what-is/iot/. [Internet; consultado el 30-junio-2025].

[2] Cloudflare. ¿Qué es Cloudflare? https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/. [Internet; consultado el 30-junio-2025].

[3] Data-flair. Imagen de un diagrama IoT . https://data-flair.training/blogs/how-iot-works/. [Internet; consultado el 30-junio-2025].

[4] Dragino. Manual del Gateway Dragino LPS8N. https://wiki.dragino.com/xwiki/bin/view/Main/User%20Manual%20for%20All%20Gateway%20models/LPS8N%20-%20LoRaWAN%20Gateway%20User%20Manual/. [Internet; consultado el 30-junio-2025].

[5] ESET. Uso de heurística en seguridad informatica. https://help.eset.com/ecsp/6/en-US/ud_antivir_threatsense_options.html#:~:text=Heuristics%20%E2%80%93%20Heuristics%20use%20an%20algorithm,that%20did%20not%20previously%20exist. [Internet; consultado el 30-junio-2025].

[6] Github. ¿Qué es Github? https://docs.github.com/es/get-started/start-your-journey/about-github-and-git. [Internet; consultado el 30-junio-2025].

[7] Gareth Halfacree. Trend micro finds lorawan security lacking, develops lorapwn python utility, 2021.

[8] Iberdrola. ¿Qué es el Internet industrial de las cosas (IioT)? https://www.iberdrola.com/innovacion/que-es-iiot#:~:text=QU%C3%89%20ES%20EL%20INTERNET%20INDUSTRIAL,de%20Internet%20a%20aplicaciones%20industriales. [Internet; consultado el 30-junio-2025].

[9] IBM. Ventajas del uso de Reglas de asociacion. https://www.ibm.com/docs/es/spss-modeler/saas?topic=nodes-association-rules. [Internet; consultado el 30-junio-2025].

[10] Jetbrains. ¿Qué es IntelliJ? https://www.jetbrains.com/es-es/idea/. [Internet; consultado el 30-junio-2025].

[11] Manuals. Manual del MerryIoT Motion Detection. https://manuals.plus/merryiot/ms10-915-motion-detection-manual. [Internet; consultado el 30-junio-2025].

[12] Microsoft. ¿Qué es Git? https://learn.microsoft.com/es-es/devops/develop/git/what-is-git. [Internet; consultado el 30-junio-2025].

[13] Moko Smart. ¿Qué LoRa? . https://www.mokosmart.com/es/lora-technology/#:~:text=LoRa%2C%20Corto%20para%20largo%20alcance,tradicionales%20como%20wifi%20o%20bluetooth.&text=Basado%20en%20la%20definici%C3%B3n%20anterior,consumo%20de%20energ%C3%ADa%20extremadamente%20bajo. [Internet; consultado el 30-junio-2025].

[14] Monolitic. ¿Qué es LoRaWAN? . https://www.monolitic.com/lorawan-que-es-como-funciona/#:~:text=%C2%BFQu%C3%A9%20es%20LoRaWAN%20y%20c%C3%B3mo,5%20km%20en%20entornos%20urbanos.. [Internet; consultado el 30-junio-2025].

[15] Amith Murthy. Detecting iot device compromise using python, Aug-2022.

[16] Ian H. Witten Eibe Frank Mark A. Hall Christopher J. Pal. *Data Mining Practical Machine Learning Tools and Techniques Fourth Edition.* Cambridge, MA : Morgan Kaufmanns, 2017.

[17] The Things Network. Imagen de una Arquitectura LoRaWAN . https://www.thethingsnetwork.org/docs/lorawan/architecture/. [Internet; consultado el 30-junio-2025].

[18] The Things Network. Imagen de una Grafica comparativa . https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/. [Internet; consultado el 30-junio-2025].

[19] The Things Network. Seguridad LoRawan . https://www.thethingsnetwork.org/docs/lorawan/security/. [Internet; consultado el 30-junio-2025].

[20] The Things Network. ¿Qué es TTS? . https://www.thethingsindustries.com/docs/getting-started/the-things-stack-basics/. [Internet; consultado el 30-junio-2025].

[21] Wikipedia. ¿Qué es Docker? https://es.wikipedia.org/wiki/Docker_(software). [Internet; consultado el 30-junio-2025].

[22] Wikipedia. ¿Qué es Html? https://es.wikipedia.org/wiki/HTML. [Internet; consultado el 30-junio-2025].

[23] Wikipedia. ¿Qué es JavaScript? https://es.wikipedia.org/wiki/JavaScript. [Internet; consultado el 30-junio-2025].

[24] Wikipedia. ¿Qué es Overleaf? https://es.wikipedia.org/wiki/Overleaf. [Internet; consultado el 30-junio-2025].

[25] Wikipedia. ¿Qué es Python? https://es.wikipedia.org/wiki/Python#:~:text=Python%20es%20un%20lenguaje%20de,en%20menor%20medida%2C%20programaci%C3%B3n%20funcional. [Internet; consultado el 30-junio-2025].

[26] Amaguaña Aguilar y Christian Javier. Herramientas de seguridad defensiva y ofensiva para redes lorawan : desarrollo de un prototipo de módulo de análisis de tráfico basado en wireshark para detección de ataques de denegación usando inspección de tramas lorawan que provea una capa de integración api rest, Oct-2022.