



Realizado por
Jesús Manuel Contreras Herrera

Tutorizado por
María Cristina Alcaraz Tello
Juan Enrique Rubio Cortés

Departamento
Lenguajes y ciencias de la computación

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE DE 2020

UNIVERSIDAD DE MÁLAGA

E.T.S. INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERIA DE SOFTWARE

# Aplicación web para la gestión y monitorización de eventos

# Web application for event monitoring

Realizado por
**Jesús Manuel Contreras Herrera**

Tutorizado por
**María Cristina Alcaraz Tello**
**Juan Enrique Rubio Cortés**

Departamento
**Lenguajes y ciencias de la computación**

UNIVERSIDAD DE MÁLAGA
MÁLAGA, SEPTIEMBRE DE 2020

Fecha defensa:

# Resumen

En los últimos años, el avance en las técnicas de comunicación y el abaratamiento de los costes de fabricación de dispositivos, así como la publicación de estándares abiertos para la interconexión de equipos hardware y software han propiciado la integración de tecnologías tan novedosas como el Big Data, el Internet de las Cosas o la virtualización en todos los ámbitos de las sociedad. Ante esta tendencia creciente cabe plantearse la necesidad de hacer frente a los numerosos problemas de seguridad asociados a estos entornos que pueden poner en riesgo infraestructuras tan críticas como la red eléctrica, los sistemas de transporte o telecomunicaciones [1].

Por ello, en este proyecto se ha creado una plataforma web que permite la recogida, análisis y presentación de la información relacionada con el tráfico de una red, que puede complementar a un sistema de detección de intrusos (en inglés: Intrusion Detection System (IDS)) para la detección de anomalías. Además esta plataforma ha sido probada en un entorno industrial de pruebas proporcionado por la universidad de Málaga.

Para la construcción de la plataforma web se ha hecho uso del framework Django [2], un framework construido en Python [3]. Microsoft SQL Server [4] y Power BI [5] han sido usados para el almacenamiento de datos y posterior análisis y presentación. Además se han utilizado algoritmos de Machine Learning (ML) para realizar predicciones sobre los datos obtenidos.

**Palabras clave:** Sistemas de detección de intrusos, Machine Learning, seguridad informática, Scapy

# Abstract

In recent years, advances in communication techniques and lower device manufacturing costs, as well as the publication of standards for the interconnection of hardware and software equipment have led to the integration of innovative technologies such as Big Data, the Internet of Things or virtualization in all areas of society. Faced with this growing trend, it is necessary to consider the need to face the numerous security problems associated with these environments that, when integrated into the industry, can put at risk infrastructures as critical as the electricity grid, transport systems or telecommunications [1].

For this reason, a web platform has been created during this project allowing the collection, analysis and presentation of information related to network traffic, which can complement an Intrusion Detection System (IDS) for anomaly detection. Furthermore, this platform has been tested in an industrial environment provided by the University of Malaga.

Django framework [2] has been used for the construction of the web platform, a framework built in Python [3]. Microsoft SQL Server [4] and Power BI [5] have been used for the data storage and subsequent analysis and exposition of the data. In addition, Machine Learning algorithms have been used to make predictions on the data collected.

**Keywords:** Intrusion Detection System, Machine Learning, Scapy , cybersecurity

# Índice

# Acrónimos

**IDS:** Sistema de detección de intrusos

**ML:** Machine Learning

**SO:** Sistema operativo

**XP:** Extreme Programming

**SCADA:** Supervisory Control And Data Acquisition

**IP:** Internet Protocol

**IPv4:** Internet Protocol version 4

**TCP:** Transmission Control Protocol

**UDP:** User Datagram Protocol

**RAE:** Real Academia Española

**BPF:** Berkeley Packet Filter

**RAM**: Random Access Memory

# Tablas y figuras

## Tablas



## Figuras

# 1

# Introducción

## 1.1 Motivación, metodología y objetivos del TFG

La motivación principal por la cual se desarrolló este proyecto fue la curiosidad de profundizar en la seguridad informática desde el punto de vista de las redes de comunicación.

En los últimos años se han integrado tanto en el ámbito empresarial, como industrial y personal, el uso de dispositivos conectados entre sí para la recolección de datos que nos ayuden en nuestro día a día. Por lo tanto, es necesario la creación de sistemas de seguridad que nos ayuden a detectar y protegernos de un uso ilegítimo de la información que fluye a través de estas redes.

El objetivo principal será la creación de una herramienta web, que funcione tanto de manera aislada como complementaria a un sistema de detección de intrusos (en inglés: Intrusion Detection System (IDS)). La herramienta será capaz de leer el tráfico de red y almacenarlo para su posterior análisis.

Con la finalidad de detectar anomalías en el entorno se dotará a la herramienta de paneles visuales. Estos paneles visuales ayudarán a la toma de decisiones de seguridad en base a los datos extraídos. Dichos paneles contarán con distintos tipos de filtros para conocer el estado actual del sistema y/o realizar posibles predicciones sobre el comportamiento del sistema.

El sistema operativo (SO) sobre el que se ha desarrollado la aplicación web ha sido un Windows 10 [6]. Tras el desarrollo inicial en un ambiente doméstico se realizó el despliegue de la aplicación en una máquina virtual, situada en el entorno de pruebas industrial proporcionado por la Universidad de Málaga, con el mismo sistema operativo. El SO usado no es un limitante para la instalación de la aplicación web. La aplicación puede desplegarse en cualquier sistema que pueda ejecutar el software y las librerías que mencionaremos más adelante.

Para el desarrollo del software usaremos las directrices las *metodologías ágiles*. Estas directrices están recogidos en un documento que se le conoce como los Valores del Manifiesto Ágil [7]:

1. Valorar más a los individuos y sus interacciones más que a los procesos y las herramientas.
2. Valorar más el software funcionando que la documentación exhaustiva.
3. Valorar más la colaboración con el cliente que la negociación contractual.
4. Valorar más la respuesta ante el cambio que seguir un plan.

Estos valores son los pilares sobre los que se fundamentan los 12 principios a seguir en las metodologías ágiles [8]:

1. La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua del software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.

7. El software funcionando es la medida principal de progreso.
8. Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

Siguiendo estos principios hemos utilizado la metodología de Extreme Programming (XP). XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico [7].

Las historias de usuario son la herramienta utilizada en XP para recoger la información sobre las especificaciones del software. Estas historias son un conjunto de tarjetas con una breve descripción de las características que el sistema debe poseer, esta información debe ser lo suficientemente detallada para que el programador pueda implementarla en un corto periodo de tiempo, entendiendo como corto periodo de tiempo una duración máxima de un sprint.

Con el fin de adaptar está metodología a nuestro Trabajo Fin de Grado, se han modificado varias de las características de la metodología ágil XP, aunque se han seguido los principios generales de estas.

## 1.2 Estado del arte

En los últimos años, el avance en las técnicas de comunicación y el abaratamiento en los costes de fabricación de dispositivos, así como la publicación de estándares abiertos para la interconexión de equipos hardware y software, ha propiciado la integración de tecnologías tan novedosas como el Internet de las Cosas, el Big Data o la virtualización en todos los ámbitos de la sociedad. En este contexto, las infraestructuras más críticas no están exentas de dicha evolución, siendo objeto de integración para paradigmas futuribles tan prometedores como el 5G o el Fog/Edge Computing [10]. Ante esta tendencia creciente en cuanto a la complejidad de tecnologías de red y heterogeneidad de dispositivos interconectados, cabe plantearse la necesidad de hacer frente a los numerosos problemas de seguridad asociados a esos entornos que, cuando se integran en la industria, pueden poner en riesgo infraestructuras tan críticas como la red eléctrica, los sistemas de transporte o telecomunicaciones, entre otras [11]. Estamos hablando de sofisticadas amenazas muy diversas que permiten el robo de credenciales, la modificación de protocolos de comunicación o la intercepción de información sensible, que en última instancia pueden impedir por completo la disponibilidad de los servicios y resultar en pérdidas millonarias para las empresas y entidades involucradas [12].

Dado este escenario, la primera medida de protección en redes como las anteriormente mencionadas consiste en la implantación de sistemas de gestión de eventos que permitan visualizar el estado de la infraestructura al completo y en tiempo real, con objeto de detectar posibles comportamientos anómalos, acotar las secciones de la red más afectadas y reducir en lo posible el impacto de potenciales ataques. Como resultado, se obtiene una información especialmente valiosa para la puesta en marcha de sistemas de respuesta rápida de forma autónoma, mejorando al mismo tiempo la toma de decisiones por parte de administradores de seguridad [13].

En la última década, se ha investigado ampliamente el uso de Machine Learning (ML) para el procesamiento automático de una gran cantidad de datos de forma simultánea, permitiendo detectar los comportamientos anómalos de manera eficaz. Cuando

combinamos ML con elementos visuales lograremos mejores resultados relacionados con la toma de decisiones en la seguridad de las comunicaciones [14].

Uno de los enfoques más usado a la hora de proteger los sistemas de red es el basado en la detección de anomalías. Este enfoque requiere la recopilación de grandes cantidades de información, incluyendo información confidencial. La evaluación de estos datos permite crear modelos de comportamiento que ayuden a detectar cambios en el comportamiento normal del sistema [15]. Este tipo de métodos tienen la desventaja de que sacan a las luz posibles falsos positivos o incluso posibles falsos negativos.

Debido al continuo cambio de los métodos de ataque no solo necesitamos sistemas de protección que reconozcan ataques conocidos, sino que además sean lo suficientemente flexibles para detectar y clasificar ataques no conocidos. Para ello, se ha investigado el uso de Deep Learning [16] para facilitar la identificación los mejores algoritmos para la detección de futuros ciberataques [17].

Aunque el uso de IDS para la protección de las redes ha sido ampliamente investigado, la continua evolución de los técnicas de ataque y la creación de nuevos dispositivos que se añaden a nuestras redes hace que no podamos parar de investigar esta capa de defensa tan importante. Para resolver este problema nos hemos propuesto la creación de un IDS para sistemas SCADA (Supervisory Control And Data Acquisition) [18] basado en una plataforma web que utilice distintas técnicas para la detección de anomalías en la red.

## 1.3 Estructura de la memoria

En esta sección comentaremos brevemente la estructura de la memoria y realizaremos una breve introducción de cada capítulo.

- **Capítulo 2. Análisis de requisitos**

  Este capítulo contiene de una breve introducción de la importancia de la realización de análisis de requisitos para el correcto desarrollo de los proyecto y dos secciones donde

comentaremos brevemente los requisitos funcionales y no funcionales implementados. Además se añaden diagramas a alto nivel para modelar el sistema.

- **Capítulo 3. Fundamentos teóricos y herramientas**

  Este capítulo se subdivide en dos secciones. En la primera sección abordamos los conceptos teóricos básicos para entender la memoria, en el que hablaremos sobre tipos de análisis de red, protocolos de red y ML. En la segunda sección realizamos una introducción a las herramientas y tecnologías empleadas para el correcto desarrollo del proyecto.

- **Capítulo 4. Desarrollo del proyecto**

  En este capítulo hacemos un recorrido sobre cómo se ha desarrollado el proyecto, los requisitos recogidos durante las reuniones, su diseño e implementación y las pruebas y validaciones realizadas para comprobar su correcto funcionamiento.

- **Capítulo 5. Conclusiones y trabajo futuro**

  En este capítulo se comentarán las conclusiones, incidentes debidos a la pandemia y posibles líneas de trabajo futuro para la continuación del proyecto.

# 2

# Análisis de requisitos

La realización de un análisis de requisitos al comienzo del proyecto, y en la finalización de un sprint es indispensable para organizar, gestionar y priorizar el trabajo que se desarrolla a lo largo del proyecto.

Al comienzo del proyecto se realizó un estudio del entorno de trabajo para elegir las herramientas que mejor se adaptasen al entorno y a los conocimientos del equipo.

Durante el desarrollo del proyecto se realizaron varios sprint siguiendo las directrices y principios de las metodologías ágiles, las cuales hemos explicado anteriormente. Cómo resultado final de los requisitos implementados se ha obtenido las siguientes tablas que recogen el trabajo realizado.

Las historias de usuario son una breve descripción de las tareas a realizar en cada sprint. Las tablas presentadas a continuación es un resumen de las historias de usuario, las cuales, hemos divido en dos partes, funcionales y no funcionales.

## 2.1 Requisitos funcionales

Llamamos requisitos funcionales a las especificaciones que se deben desarrollar a lo largo del proyecto para llegar al objetivo deseado tabla 1.

| Nombre de requisito | Identificador del requisito |
|---|---|
| Leer el tráfico en tiempo real | **RF001** |
| Clasificar los paquetes de red | **RF002** |
| Agregar un nuevo host al sistema | **RF003** |
| Editar un host existente en el sistema | **RF004** |
| Eliminar un host existente en el sistema | **RF005** |
| Agregar interfaz de red desde donde se analiza el trafico | **RF006** |
| Editar interfaz de red desde donde se analiza el tráfico | **RF007** |
| Visualizar los nodos que pertenecen a la red | **RF008** |
| Visualizar la cantidad de paquetes enviados por nodo | **RF009** |
| Visualizar los diferentes tipos de paquetes enviados por nodo | **RF010** |
| Visualizar la cantidad de paquetes recibidos por nodo | **RF011** |
| Visualizar los diferentes tipos de paquetes recibidos por nodo | **RF012** |
| Visualizar la comunicación entre nodos | **RF013** |
| Visualizar la cantidad de paquetes que se intercambian entre nodos | **RF014** |
| Filtrar la visualización de la información por fechas | **RF015** |
| Leer archivos pcap | **RF016** |
| Mostrar de manera gráfica las sincronizaciones que se producen | **RF017** |
| Ver las distintas sincronizaciones que se producen en la red | **RF018** |
| Visualización de la media de paquetes enviados por un nodo | **RF019** |
| Visualización de la media de paquetes recibidos por un nodo | **RF020** |
| Crear alerta cada vez que aparece un nuevo Host | **RF021** |
| Crear alerta si la IP del host y la MAC no coinciden | **RF022** |

| | |
|---|---|
| Analizar las propiedades de un host existente en la red | **RF023** |
| Estudio del modelo de datos | **RF024** |
| Implementación del algoritmo de ML | **RF025** |
| Crear usuario en la plataforma web | **RF026** |
| Restringir el uso de la plataforma solo a usuarios registrados | **RF027** |

**Tabla 1.** Requisitos funcionales

## 2.2 Requisitos no funcionales

Los requisitos no funcionales describen las limitaciones del sistema, pueden derivar de los requisitos funcionales, del entorno donde se encuentra el sistema o de estándares que se deben cumplir tabla 2.

| Nombre de requisito | Identificador |
|---|---|
| Debe poder instalarse y ejecutarse en distintos SO | **RNF001** |
| No debe de perder más del 97% de paquetes que viajan en la red | **RNF002** |
| Debe ser accesible desde el navegador | **RNF003** |
| Las contraseñas deben ser almacenadas con Hash y Salt | **RNF004** |

**Tabla 2.** Requisitos no funcionales

## 2.3 Diseño

Tras realizar un estudio inicial podemos proceder a la creación de la base de datos, que contará con las siguientes entidades:

- Paquete: clase para almacenar información sobre los paquetes de red recogidos.
- Host: clase para almacenar información sobre los host existentes en el sistema.
- Alerta: recopilación de anomalías detectadas en la red.

- Interfaz: configuración del programa donde almacenar las interfaz de red sobre la que escuchamos.
- Pcapfile: almacenamiento de archivos de capturas de red para su posterior análisis.

Durante el proyecto se han implementado funciones para la realización de distintas acciones, como resultado final de proyecto dentro de la plataforma web se pueden realizar las acciones expuestas en el diagrama de casos de uso que se muestra en la figura 1.

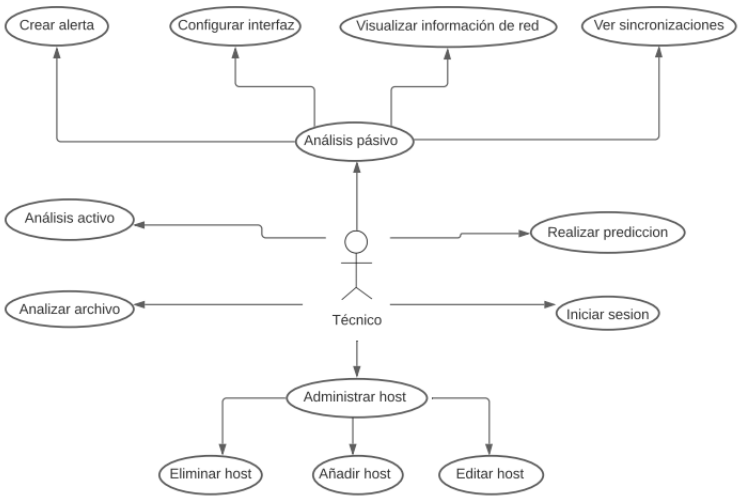


**Figura 1.** Diagrama de casos de uso

En el proyecto nos centraremos principalmente en el análisis los paquetes IPv4/TCP (en inglés: Internet Protocol Version 4, Transmission Control Protocol), de los host del sistema y las alertas que se creen en caso necesario. La información de estos objetos y están relacionados como se muestra en la figura 2.

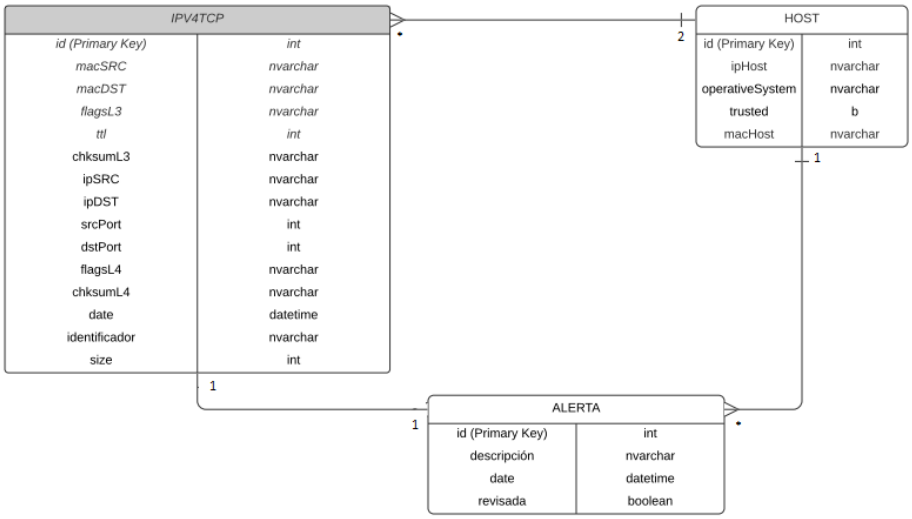


**Figura 2.** Diagrama E/R de las principales entidades del sistema

# 3

# Fundamentos teóricos y herramientas

## 3.1 Fundamentos teóricos

Durante esta sección procedemos a realizar una introducción de los conceptos teóricos necesarios para poder entender el desarrollo del proyecto.

### 3.1.1 Análisis del tráfico de red

Para la realización de un ataque remoto a una red informática, un atacante no sólo necesita conocer la dirección en la red de la víctima, sino también obtener la máxima información sobre ella. No basta con saber su dirección IP (en inglés: Internet Protocol), si un atacante logra conocer el sistema operativo, la versión del sistema operativo, los servicios y la versión de los servicios, esté podrá realizar un ataque más preciso sobre la red y los nodos que se encuentran en ella [19].

Para obtener está información es necesario realizar un análisis del tráfico de red. Llamamos análisis del tráfico al proceso de inferir información a partir de las características del tráfico de comunicación sin analizar la información que se intercambias los comunicantes [20]. En el desarrollo del proyecto nos hemos centrado principalmente en analizar las comunicaciones entre nodos origen y destino, la frecuencia y el tamaño.

Los análisis empleados en el proyecto los podemos clasificar en dos tipos, análisis activo; mediante la librería Nmap [21] ; y análisis pasivo; mediante el uso de Scapy [22].

#### 3.1.1.1 Análisis activo

Llamamos análisis activo de una red cuando se realiza una monitorización introduciendo paquetes en la red a analizar. Esta monitorización puede realizarse con el fin de analizar el rendimiento de una aplicación, descubrir nodos no conocidos en la red o analizar los nodos existentes.

Podemos realizar una análisis activo de puertos enviando paquetes TCP y UDP (en inglés: User Datagram Protocol) con el fin de detectar si están abiertos o cerrados, con esta información podemos saber qué servicios están disponibles.

Para realizar una análisis activo del sistema operativo existen numerosas técnicas, las más utilizadas en la actualidad se basan en que cada sistema operativo implementa de una forma distinta su pila TCP/IP, por lo que si se envían distintos paquetes y analizamos la respuesta recibida se puede descifrar el sistema operativo que utiliza el nodo en cuestión.
La ventaja de realizar una análisis activo es que se obtiene información precisa sobre la red y los nodos que estamos investigando en un corto periodo de tiempo.

La desventaja de este tipo de análisis es que es fácilmente detectable por un IDS. Además el uso de este tipo de análisis puede sobrecargar la red y causar problemas en los dispositivos donde se realizan, pudiendo causar una parada en el entorno industrial.
Para reducir el problema de la fácil detección se pueden utilizar varias técnicas:

- Análisis aleatorio de puertos y direcciones IP en lugar de realizar un análisis secuencial.
- Análisis lento, realizando una espera prolongada con el fin de no ser detectada como análisis.
- Análisis distribuido se utilizan varios computadores de manera simultánea y coordinada.

- Fragmentación de paquetes utilizados en el análisis, de manera que la información que contienen quede dividida y no sea detectada por los IDS [19].

#### 3.1.1.2 Análisis pasivo

El análisis pasivo consiste en la captura de los paquetes de información que viajan por la red durante un periodo de tiempo. El análisis pasivo permite un estudio en profundidad de una red de manera offline, especialmente cuando buscamos la causa raíz de una anomalía. A diferencia del análisis activo, el análisis pasivo no inyecta paquetes de prueba, simplemente recolecta la información que viaja a través de la red.

En la actualidad muchos de los IDS utilizan este tipo de análisis en tiempo real para la detección de intrusiones o anomalías en la red. Existen multitud de métodos para la detección de intrusiones, por ejemplo:

- Sistemas basados en firmas (en inglés: signature-based IDS) se centran en buscar patrones de ataques conocidos en los paquetes analizados, la principal desventaja radica en que son incapaces de detectar nuevos tipos de ataques cuyos patrones no estén contemplados [24].
- Sistemas de detección basados en anomalías (en inglés: anomaly-based IDS), este tipo de IDS comparan el estado actual del sistema con el comportamiento habitual del mismo, generando información sobre posibles desviaciones en el comportamiento. Este tipo de IDS requiere un análisis inicial para la creación de un modelo que sirva como base de comportamiento normal. Los problemas de este tipo de sistemas radican en la heterogeneidad de los datos que se analizan, así como en la diferenciación entre ataque y avería. Por este motivo, se han propuesto numerosas técnicas basadas en la minería de datos, análisis estadístico, basadas en el conocimiento o técnicas de aprendizaje automático mediante el uso de ML [24].

Las principales ventajas de realizar un análisis pasivo son que no se pueden detectar, puesto que no envía información. Además, permite la detección de nodos que solo están activos durante cortos periodos de tiempo, así como de servicios activos que no utilizan los puertos más conocidos.

Las desventajas o limitaciones de la realización de análisis pasivos son que si se quiere analizar un nodo concreto se debe esperar a que el nodo envíe información a la red. Para la correcta recolección del tráfico se debe instalar el software en un switch, desde donde el tráfico de red pueda ser monitoreado. En caso de que la información recogida sea demasiado extensa podría causar una sobrecarga en la CPU y con ello una bajada de rendimiento en la red.

#### 3.1.1.2.1 Protocolo IP

Según la Real Academia Española (RAE) [25] un protocolo es "*una secuencia detallada de un proceso de actuación científica, técnica, médica, etc.*". Cuando hablamos de un protocolo de red, nos referimos al conjunto de reglas que rigen cómo los paquetes de comunicación deben transmitirse a través de la red.

El protocolo IP fue desarrollado durante la década de 1970, y es el que se usa a través de Internet, redes empresariales y redes industriales. La mayoría de las redes actuales utilizan el protocolo estándar de IPv4, el cual está compuesto por direcciones IP de 4 bytes (32 bits) [26].

Los datos en el protocolo IP están organizados por mensajes, estos mensajes son conocidos como paquetes o datagramas. Cada paquete está compuesto por la siguiente información figura 3:

- Versión del protocolo, 4 bits.
- Longitud de la cabecera (IHL), 4 bits.
- Differentiated services code point (DSCP), 8 bits.
- Longitud total, 16 bits.
- Identificador del datagrama, 16 bits.
- Flags de fragmentación, 3 bits.
- Posición del fragmento, 13 bits.
- Tiempo de vida (TTL), 8 bits.
- Protocolo de la capa superior, 8 bits.
- Checksum, 16 bits.
- IP origen, 32 bits.

- IP destino, 32 bits.
- Opciones.

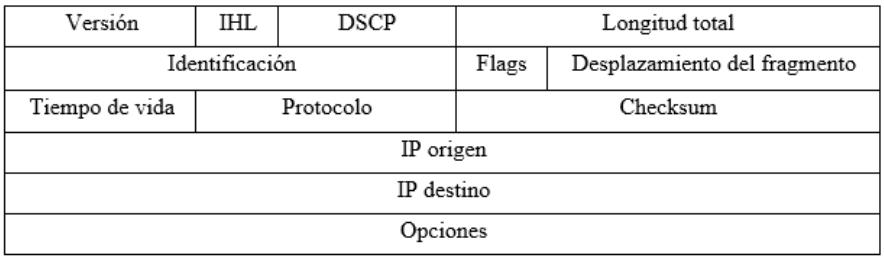


**Figura 3.** Cabeceras de un datagrama IP

A comiendo de la década de los 90 Internet se expandió a la industria y al comercio [9]. En la actualidad, con el uso extendido de teléfonos móviles y la aparición del Internet de las Cosas, vemos que el protocolo IPv4 tiene ciertas limitaciones. La más destacada es la cantidad de direcciones IP. $2*10^{32}$ direcciones IP posibles no es suficiente para la cantidad de nodos actualmente existentes. Está limitación fue el factor principal para la introducción de las redes clasificadas y la creación del "enrutamiento entre dominios sin clases", en inglés Classless Inter-Domain Routing (CIDR) [27]. Por otra parte, IPv4 fue creado para direccionar la información, sin tener en cuenta mecanismos de seguridad que permitan verificar la veracidad de la información que se está recibiendo, dejando esta tarea en manos de las capas superiores e inferiores.

IPv6 es un protocolo desarrollado para resolver los problemas y limitaciones de IPv4 [29]. IPv6 cuenta con un direccionamiento de 128 bits de longitud, $3.4*10^{38}$ direcciones posibles, resolviendo el problema principal de IPv4. La cabecera del protocolo IPv6 tiene un tamaño fijo de 40 bytes, a diferencia de las cabeceras de IPv4 que tienen un tamaño variable. En lo referente a seguridad IPv6 implementa, de manera obligatoria, el uso de IPsec. IPsec es un conjunto de protocolos del Internet Engineering Task Force (IETF) [28] que proporcionan autenticación y cifrado a nivel de red [29].

### 3.1.1.2.2 Protocolo TCP

El protocolo TCP es un protocolo de transporte, el cual garantiza que los datos enviados llegan a su destino sin errores mediante el uso de distintas técnicas.

En nuestra herramienta analizaremos principalmente el protocolo TCP. Los datos más relevantes, en el ámbito de nuestra aplicación, que nos aporta este protocolo son figura 4:

- Puerto origen.
- Puerto destino.
- Flags activos.
- Checksum.

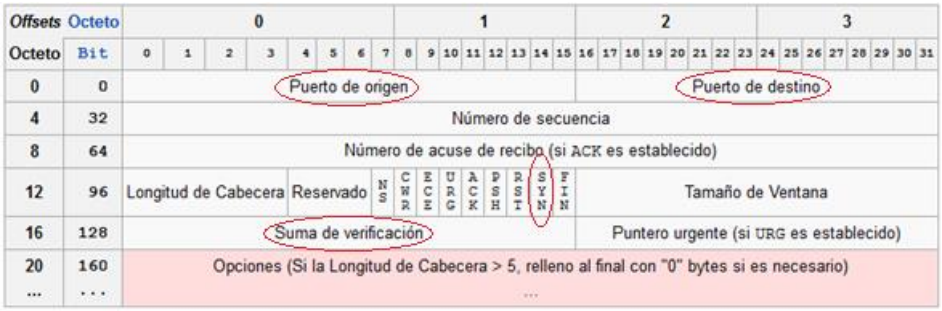


**Figura 4.** Caberas paquete TCP

Analizando está información podemos observar los servicios que están activos en los diferentes nodos de la red, los intentos de sincronización a los diferentes puertos con el fin de detectar anomalías en la red, abuso de puertos intentos de sincronizaciones a puertos específicos, etc.

### 3.1.1.2.3 Otros protocolos

Otros protocolos que cabe mencionar y de los que se recolecta información en la plataforma web son:

- Protocolo UDP es un protocolo de la capa de transporte que cumple la misma función que TCP. La diferencia entre TCP y UDP es que el protocolo UDP no asegura que los paquetes hayan sido entregados a su destino, esto hace que el protocolo sea menos fiable pero más rápido que el protocolo TCP [29].
- Protocolo ICMP (Internet Control Message Protocol) [41] se utiliza para enviar mensajes de error e información sobre la disponibilidad de un nodo o servicio.

### 3.1.2 Machine Learning

El ML es una rama de la inteligencia artificial que nos permite que las máquinas aprendan sin ser expresamente programadas para ello. Esta es una habilidad indispensable para hacer sistemas que funcionen de forma autónoma. El ML funciona a través del reconocimiento de patrones, al reconocer estos patrones el algoritmo es capaz de realizar predicciones sobre los datos obtenidos [30].

Para lograr estos resultados se utilizan diferentes tipos de algoritmos, estos algoritmos podemos clasificarlos en tres grupos [31]:

- **Algoritmos supervisados**: este tipo de algoritmos utilizan varios conjuntos de datos. Estos datos son etiquetados previamente para ser procesados por el algoritmo que realiza predicciones. Este proceso se repite hasta que el modelo alcanza la precisión objetivo.
- **Algoritmos semi-supervisados**: estos algoritmos utilizan datos etiquetados y datos no etiquetados para generar una función que se adapte al modelo de datos.
- **Algoritmos no supervisados**: el modelo de datos inicial no se encuentra etiquetado. Debido a esto el algoritmo debe ser capaz de analizar las estructuras presentes y eliminar la redundancia.

Además dentro de esta clasificación podemos encontrar ciertos algoritmos que tienen características más específicas. Entre los más importantes encontramos:

- **Deep Learning**: este tipo de algoritmo consiste en la creación de redes neuronales para la creación de representaciones abstractas de la información, facilitando de esta manera el aprendizaje [42].
- **Active Learning**: este tipo de algoritmos son capaces de interactuar con la persona de manera que puedan "reeducarse" para obtener los resultados esperados [31].

Para el desarrollo del análisis predictivo haremos uso de un algoritmo supervisado, en concreto utilizaremos un **algoritmo de regresión**. La regresión es un modelo estadístico que describe la relación entre variables. Por ello nosotros lo utilizaremos para en función

de la cantidad de paquetes que recorren la red predecir la cantidad de información que viaja por ella.

## 3.2 Herramientas

En esta sección procedemos a comentar las principales herramientas y tecnologías que usamos durante el desarrollo del proyecto.

### 3.2.1 Microsoft SQL Server

Inicialmente se hizo un estudio sobre los datos que iban a ser recogidos. Tras un análisis inicial, teniendo en cuenta las ventajas y desventajas, se optó por una base de datos que siguiese el modelo entidad-relación.

Microsoft SQL Server [4] es un sistema de gestión de bases de datos relacional que nos permite trabajar en modo cliente-servidor. Utilizaremos esta herramienta de base de datos relacional para mantener un cuadro de uso de elementos Microsoft, facilitando de esta manera, una posible línea futura para la migración de la aplicación web a la nube pública Azure [32].

### 3.2.2 Power BI

Power BI [5] es una colección de servicios de software, aplicaciones y conectores que funcionan conjuntamente para convertir orígenes de datos sin relación entre sí en información coherente, interactiva y atractiva visualmente.

Sus datos pueden ser una hoja de cálculo de Excel o una colección de almacenes de datos híbridos locales y basados en la nube. Power BI permite conectarse con facilidad a los orígenes de datos, visualizar y descubrir qué es importante y compartirlo con cualquiera o con todos los usuarios que desee [33].

Power BI consta de 3 partes:

- Aplicación de escritorio: es una aplicación gratuita incluida dentro de office 365. Permite conectarse a los datos desde varios orígenes, transformarlos, combinarlos y

crear objetos visuales de manera efectiva para la realización de informes. Dentro de la aplicación podemos encontrar 3 vistas diferentes:

  - Informe: para la creación de objetos visuales.
  - Datos: donde podemos ver las tablas, medidas, y datos que se usan en el modelo asociado al informe.
  - Modelo: desde el que podemos administrar las relaciones entre tablas.

- Servicio SaaS: es el servicio en línea de Power BI. Consta principalmente de:
  - Dashboard: paneles con información gráfica seleccionada de los informes.
  - Informes: tras las transformación de los datos se pueden crear una o más páginas de objetos visuales. Se pueden crear informes desde 0 o importar los informes que otros compañeros compartan.
  - Conjuntos de datos: es una colección de datos que importa o a lo que se conecta. Representa un índice de los distintos orígenes de datos utilizados en la creación de informes. Cada conjunto de datos puede estar asociado a diferentes áreas de trabajo.
  - Libros: es un tipo especial de conjunto de datos, principalmente provienen de archivos Excel.
- Aplicaciones móviles: ofrece la posibilidad de acceder a la información, tanto local como en la nube, de los informes desde su móvil. Además permite interactuar y explorar los datos de los paneles.

Power BI permite adaptar la forma de visualización de los datos dependiendo del rol que se realice dentro de un equipo o empresa, de manera que un desarrollador pueda operar sobre los datos con rol de editor, mientras que un empleado de marketing solo pueda ver los informes.

Power Bi permite la creación de tantos roles como sean necesarios, además se pueden establecer filtros de visualización de los datos mediante el lenguaje DAX(en inglés: Data Analytics Expression) [43]. DAX es un conjunto de funciones, operadores y constantes, los cuáles, se pueden usar y combinar en diferentes expresiones para calcular o devolver uno o más resultados.

### 3.2.3 Django

Django [2] es un framework de desarrollo web de código abierto que utiliza el patrón de diseño Modelo-Vista-Controlador (MVC). Elegimos Django como framework a la hora de desarrollar por varias razones:

- Estructura: este framework organiza la información de manera que nos ayuda a seguir fácilmente el patrón de diseño MVC, permitiendo acoplar y desacoplar las diferentes partes del programa sin tener que realizar grandes cambios.
- Funcionalidades: Django permite el uso de cualquier módulo existente en Python, además de un conjunto de paquetes propios de Django.
- Seguridad: Django implementa por defecto medidas de seguridad para evitar SQL Injection, Cross Site Request Forgery o Clickjacking [2].

### 3.2.4 Scapy

Scapy [22] es una herramienta, escrita en Python, que permite al usuario la realización de diferentes acciones sobre los paquetes de una red. Está herramienta es capaz de enviar, leer, diseccionar, crear y falsificar paquetes en la red. Esta capacidad permite la creación de programas que pueden sondear, escanear o atacar redes y/o crear pruebas unitarias [22].

Scapy posee una gran flexibilidad a la hora de construir los paquetes, ya que no impone ninguna restricción a la hora de apilar las diferentes capas de los paquetes, lo que permite construir protocolos que no se utilicen comúnmente. De esta forma aliviamos el requisito de escribir una herramienta distinta cada vez que el escenario cambia.

Otra gran ventaja de Scapy es que entrega los paquetes completos, sin interpretar, de manera que deja al lector la acción de interpretar los datos de manera correcta.

La función principal que utilizaremos es esnifar (en inglés, sniff) para realizar las lecturas de los diferentes paquetes en la red. Concretamente, analizamos los parámetros:

- prn(opcional): indica si se requiere algún procesamiento de los datos. Está función se realiza sobre cada paquete leído. En nuestro caso, la utilizamos para guardar la información “útil” de cada paquete.

- iface(obligatorio): indica la interfaz de red desde la que se realiza la lectura de paquetes. En caso de no especificarlo se utiliza el dato almacenado en la variable por defecto 'conf.iface'. Si no se especifica correctamente no obtendremos ningún dato.
- filter(opcional): utilizamos lenguaje Berkeley Packet Filter (BPF) para filtrar los paquetes que queremos recolectar.

Uno de los fallos conocidos de Scapy es la pérdida de paquetes cuando la carga de la red es muy alta. Para comprobarlo se realizaron pruebas sobre una red doméstica y sobre el entorno de pruebas proporcionado por la Universidad de Málaga, como herramienta para comparar los datos obtenidos se utilizó Wireshark [34].

Wireshark es el analizador de protocolos más utilizado en el mundo, permite realizar análisis y resolver problemas en sistemas de comunicación.

Durante las pruebas, se pudo observar que Scapy pierde una gran cantidad de paquetes a la hora de realizar un leer el tráfico. Sin embargo, al establecer los filtros de los paquetes que queremos analizar (ICMP, TCP y UDP) vemos que la perdida de paquetes no es significativa.

### 3.2.5 Nmap

Network Mapper [21], más conocida como Nmap, es una herramienta de código abierto utilizada para la exploración de redes y la auditoría de seguridad. Nmap utiliza paquetes IP para determinar información variada sobre la red, como puede ser: nodos disponibles, servicios ofrecido, sistemas operativos y versión utilizada, tipo de paquetes de filtros / firewalls en uso, etc. [21]

Nmap es multiplataforma, es decir, puede ejecutarse en la mayoría de los SO disponibles. Además Nmap cuenta con una suite de programas complementarios:

- Zenmap: Interfaz gráfica para analizar y representar los datos.
- Ncat: herramienta de transferencia de datos, redirección y depuración.
- Ndiff: herramienta para comparar resultados de escaneo.

- Nping: herramienta de análisis de respuesta y generación de paquetes.

Nmap es un programa muy flexible y potente, el cual ha sido probado tanto en pequeñas redes de comunicaciones como en grandes. Su extendido uso en todo el mundo ha creado una gran comunidad creadora de documentación de calidad donde acudir en caso de necesitar información.

# 4

# Desarrollo del proyecto

En las metodologías ágiles se conoce como sprint al periodo entre la reunión con el cliente y la entrega del trabajo realizado por los programadores. La duración recomendada es de dos a cuatro semanas. Durante este periodo de tiempo se deben implementar los requisitos necesarios para satisfacer los objetivos propuestos en el anteproyecto, recordemos:

- La adquisición de tráfico de red para el análisis del comportamiento de un conjunto de dispositivos interconectados, con la finalidad de detectar anomalías en el entorno y/o realizar predicciones sobre la evolución de las comunicaciones entre estos dispositivos.
- La implementación de un panel de visualización en forma de aplicación web, que sirva como cuadro de mando integral para la toma de decisiones de seguridad, en base a los datos extraídos. En dicha aplicación se puedan filtrar los distintos tipos de datos que queremos visualizar, conocer el estado actual del sistema y/o realizar posibles predicciones mediante el uso de algoritmos sobre futuros comportamientos del sistema.

En el desarrollo del software se han realizado cinco sprints, la duración y los requisitos implementados durante los sprints han sido recogidos en el diagrama de Gantt mostrado en la figura 5**.**

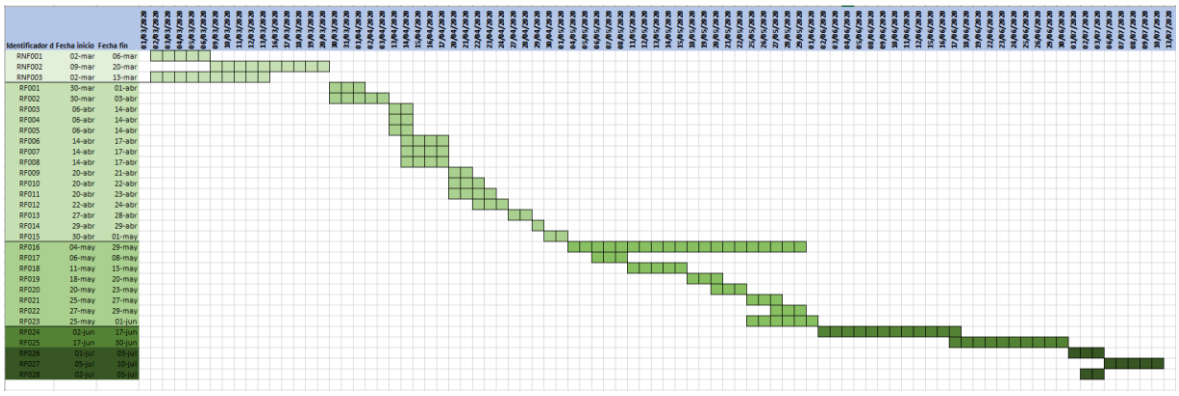


**Figura 5.** Diagrama de Gantt

## 4.1 Sprint 1 - estudio, elección y preparación del software

Se procede al estudio y elección del software a utilizar, explicado anteriormente en el capítulo 3.

Como Framework web elegimos Django, herramienta desarrollada en Python que nos ayudará a una mejor gestión y organización de la web. Como base de datos utilizaremos Microsoft SQL Server, la cual combinaremos con Power BI, herramienta de análisis y creación de paneles visuales. Se busca un cuadro de trabajo Microsoft para llegado el momento realizar una migración de la herramienta a la nube de Microsoft, conocida como Azure.

Para realizar análisis pasivo se utilizará Scapy, mientras que para el análisis activo se usará Nmap. Ambas herramientas tienen librerías adaptadas para Python.

Durante este periodo se inicia la formación del software a utilizar.

### 4.1.1 Requisitos implementados

Los requisitos implementados durante esta iteración son principalmente no funcionales, podemos observarlos en la tabla 3.

| Identificador | Descripción |
|---|---|
| **RNF001** | El software debe poder instalarse y ejecutarse en distintos SO |
| **RNF002** | No debe de perder más del 97% de paquetes que viajan en la red |
| **RNF003** | Debe ser accesible desde el navegador, siempre y cuando esté conectado a la misma Red privada Virtual |

**Tabla 3.** Historias de usuario primer sprint

### 4.1.2 Diseño e implementación

Durante el desarrollo tomamos un primer acercamiento al entorno de trabajo. Nos dedicamos a informarnos sobre distintos framework web, bases de datos compatibles con los framework web. Además realizamos tutoriales sobre Scapy y Power BI para adquirir los conocimientos necesarios para el desarrollo del proyecto.

Durante el desarrollo de este sprint procedemos a la creación de las entidades necesarias para lograr los objetivos del proyecto. Estás entidades y sus atributos son:

- IPV4TCP: id, macSrc, macDst, flagsL3, ttl, chksumL3, ipSrc, IpDst, srcPort, dstPort, flagsL4, chksumL4, date, identificador, size.
- Host: id, ipHost, operativeSystem, trusted, macHost.
- Alerta: id, ipSrc, description, date, revisada, paqueteAsociado.
- Interfaz: interfaz, estado.
- Pcapfile: id, nombre, archivo, ejecutado.

Otras entidades también propuestas para un análisis posterior son:

- IPV4ICMP: almacena información sobre los paquetes ICMP que son detectados en la red.
- IPV4UDP: almacena información sobre los paquetes UDP que son detectados en la red.
- OtherProtocol: almacena información sobre cualquier paquete que recorre la red para un análisis posterior.

### 4.1.3 Validación

Se validan los siguiente requisitos mostrados en la tabla 4.

| Identificador | Validación | Comentarios |
|---|---|---|
| **RNF001** | Validado | N/A |
| **RNF002** | Validado | Se aplica correctivo en el filtro parámetros para la óptima recolección de paquetes de red |
| **RNF003** | Validado | N/A |

**Tabla 4.** Requisitos validados en el primer Sprint

Para el correcto desarrollo del primer Sprint se comprobó que todo el software utilizado pudiese ser instalado tanto en sistemas operativos Windows como Linux.

Al ser una plataforma Web es accesible desde cualquier navegador web, siempre que el servicio esté activo y ambos nodos estén conectados a la misma red.

Por otro lado, se detectaron problemas de rendimiento con Scapy a la hora de realizar un "sniff" del tráfico de red, perdiendo una cantidad de paquetes significativa mostrado en la tabla 5. Tras la utilización de filtros en la ejecución del programa para que descartase los paquetes de red que no nos interesaban para el desarrollo del proyecto se logra alcanzar una pérdida de paquetes que cumple los requisitos propuestos tabla 6. La comparación de los resultado se hace con el software Wireshark. Wireshark es el analizador de protocolos más utilizado en el mundo, permite realizar análisis y resolver problemas en sistemas de comunicación.

| | 1 minuto | 3 minutos | 5 minutos |
|---|---|---|---|
| **Scapy** | 854 | 19755 | 36267 |
| **Wireshark** | 900 | 24685 | 41684 |

**Tabla 5.** Comparación paquete totales recogidos

| | 1 minuto | 3 minutos | 5 minutos |
|---|---|---|---|
| **Scapy** | 557 | 1003 | 25764 |
| **Wireshark** | 559 | 998 | 25784 |

**Tabla 6.** Comparación paquetes recogidos aplicando filtro.

Tras la realización de las pruebas podemos afirmar que la plataforma cumple los requisitos no funcionales propuestos.

## 4.2 Sprint 2 - captura y análisis pasivo del tráfico

Tras establecer las bases del proyecto nos volvemos a reunir para informar al cliente, en este caso los tutores de nuestro TFG, del software que vamos a utilizar. Tras esto se procede a la realización de los requisitos funcionales.

### 4.2.1 Requisitos implementados

Los requisitos implementados durante este Sprint están recogidos en la tabla 7.

| Identificador | Descripción |
|---|---|
| **RF001** | Leer el tráfico de red en tiempo real |
| **RF002** | Clasificar y almacenar los paquetes de red según sean TCP, UDP o ICMP |
| **RF003** | Agregar un nuevo host. El host debe ser almacenado automáticamente al detectarse en el paquete de red. También se debe poder añadir manualmente |
| **RF004** | Editar un host existente |
| **RF005** | Eliminar un host existente |
| **RF006** | Agregar interfaz de red desde donde se realizará la lectura del tráfico |
| **RF007** | Editar interfaz de red desde la que se realiza la lectura del tráfico |
| **RF008** | Visualizar gráfica de los nodos detectados que pertenecen a la red |

| | |
|---|---|
| **RF009** | Visualizar la cantidad de paquetes enviados por nodo |
| **RF010** | Visualizar los diferentes tipos de paquetes enviados por nodo |
| **RF011** | Visualizar la cantidad de paquetes recibidos por nodo |
| **RF012** | Visualizar los diferentes tipos de paquetes recibidos por nodo |
| **RF013** | Visualizar la comunicación entre nodos mediante grafos dirigidos |
| **RF014** | Visualizar la cantidad de paquetes que se intercambian entre nodos |
| **RF015** | Filtrar la visualización de la información por fechas |

**Tabla 7.** Historias de usuario segundo sprint

### 4.2.2 Diseño e implementación

Durante el desarrollo de este sprint realizamos un análisis pasivo del tráfico de red mediante la función 'sniff' de Scapy. La función permite la introducción de procedimientos para disección de los paquetes que recorren la red, de este modo podemos realizar la clasificación y almacenamiento de los paquetes de red. La instrucción final quedaría de la siguiente manera:

*sniff(prn=analizarPaquete,iface=interfaz.iFace,filter="ip and (icmp or tcp or udp)",store=0)*

Dentro del analizarPaquetes extraemos la siguiente información de la cabecera IP/TCP figura 6.

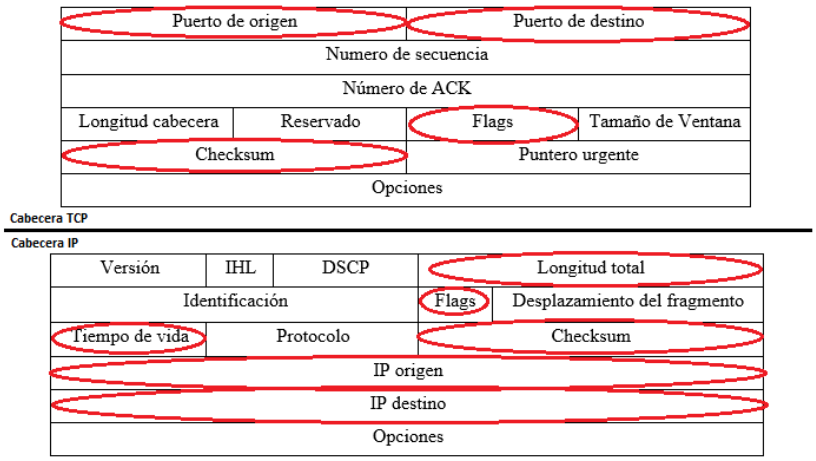

**Figura 6.** Información almacenada de las cabeceras IP y TCP

El procedimiento *analizarPaquete* nos permite almacenar información sobre los nodos que están en la red . Comprueba la IP destino y la IP fuente, si ambos se encuentran en la bases de datos no realiza ninguna acción, pero si se detecta un nuevo nodo queda registrado y marcado como desconocido.  Esta información es visualizada en la figura 7.

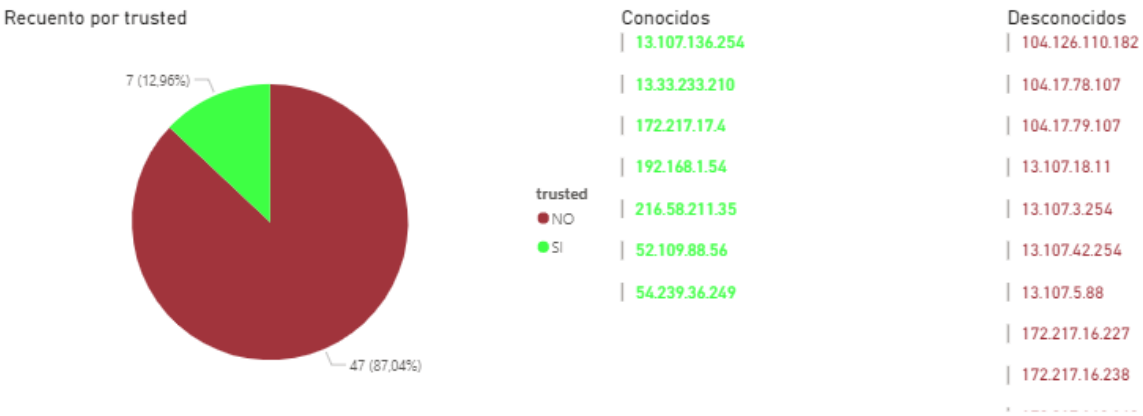


**Figura 7.** Visualización de nodos conocidos y no conocidos

Además, gracias a la herramienta Power BI realizamos análisis de los datos almacenados para crear diferentes paneles visuales permitiendo conocer la cantidad de información que recorre la red, esta información puede ser filtrada por fechas. Un ejemplo de estos paneles lo encontramos en la figura 8.

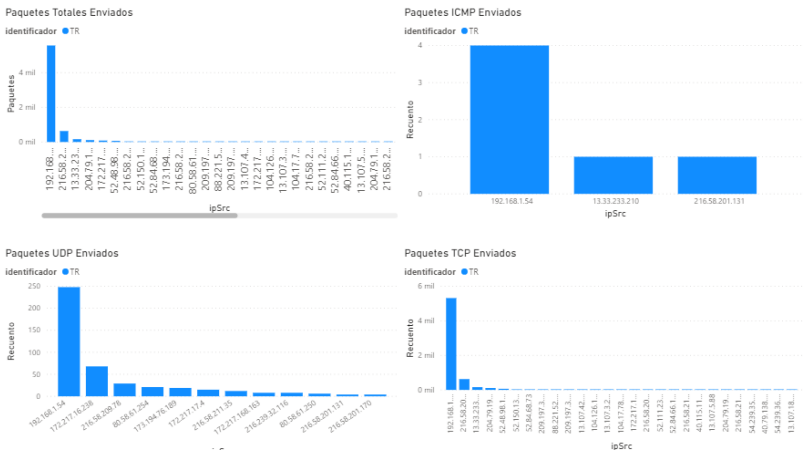

**Figura 8.** Visualización de los paquetes de red enviados por nodo

Por otro lado, para conocer qué nodos están conectados entre sí, procedemos a la creación de grafos dirigidos con Power BI. Para conocer mejor cuanta información se intercambian los nodos entre sí hacemos que el grosor de la arista varíe dependiendo de la cantidad de paquetes que se intercambian dichos nodos. Podemos ver un ejemplo en la figura 9.

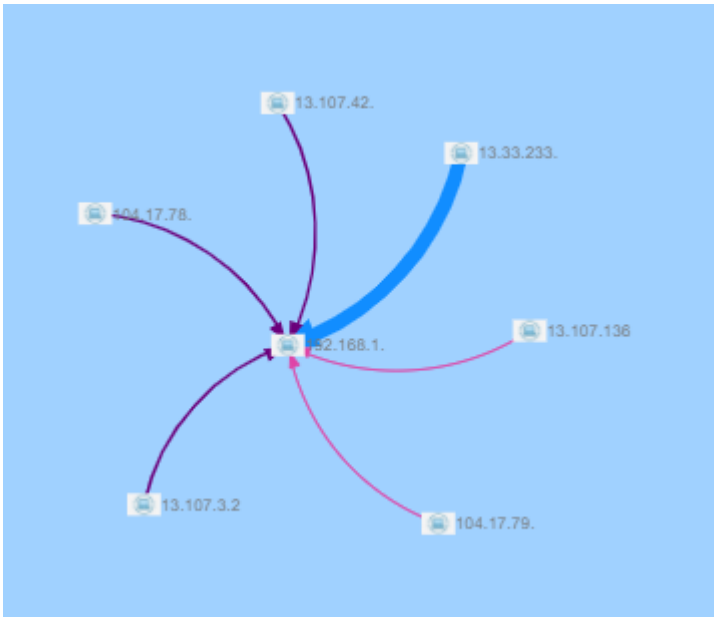


**Figura 9.** Visualización de grafo de los nodos conectados entre sí

### 4.2.3 Validación

Tras el desarrollo de este sprint se realizan las siguientes validaciones tabla 8.

| Identificador | Validación | Comentarios |
|---|---|---|
| **RF001** | Validado | Se añade el parametro 'store =0' para liberar la memoria RAM utilizada. |
| **RF002** | Validado | N/A |
| **RF003** | Validado | N/A |
| **RF004** | Validado | N/A |
| **RF005** | Validado | N/A |

| | | |
|---|---|---|
| **RF006** | Validado | N/A |
| **RF007** | Validado | En el momento de la creación se debe marcar como "no ejecutar". Una vez se ha creado ya podemos empezar a ejecutar en análisis pasivo. |
| **RF008** | Validado | N/A |
| **RF009** | Validado | N/A |
| **RF010** | Validado | N/A |
| **RF011** | Validado | N/A |
| **RF012** | Validado | N/A |
| **RF013** | Validado | N/A |
| **RF014** | Validado | N/A |
| **RF015** | Validado | Utilizamos la librería timezone para el correcto almacenamiento de las fechas. |

**Tabla 8.** Validaciones realizadas segundo Sprint

Durante el transcurso de este sprint se comprobó el correcto almacenamiento de la información en la red. Una vez trasladado al entorno industrial proporcionado por la universidad se comprobó que los nodos recogidos se corresponden con los existentes.

Se detectó una incoherencia en el almacenamiento de la fecha y hora en el análisis de los paquetes de red. El error era debido a que la fecha en la que se almacenaban los paquetes era igual a la hora en la que se había iniciado el servicio, en lugar de la hora a la que se almacenaba el paquete de red. Se corrige este error almacenando la fecha con la librería nativa de Django.

Se detectó un error de rendimiento en el uso de la función 'sniff ' durante un tiempo prolongado. La memoria RAM (en inglés: Random Access Memory) de la máquina virtual que utilizamos en el entorno industrial llegaba al 100% de uso, dejando el sistema operativo completamente parado. Este problema se corrigió añadiendo el parámetro 'store=0' en la función 'sniff'’.

## 4.3 Sprint 3 - análisis activo de la red

Tras la finalización del segundo sprint nos volvemos a reunir para comprobar los requisitos implementados. Tras la validación de estos procedemos al diseño e implementación de los requisitos necesarios para realizar un análisis activo y creación de alertas.

### 4.3.1 Requisitos implementados

Los requisitos implementados durante este periodo de tiempo son los expuestos en la tabla 9.

| Identificador | Descripción |
|---|---|
| **RF016** | Leer, almacenar y analizar archivos pcap de manera offline |
| **RF017** | Ver las distintas sincronizaciones que se producen en la red |
| **RF018** | Mostrar de manera gráfica las sincronizaciones que se producen |
| **RF019** | Visualización de la media de paquetes enviados por un nodo |
| **RF020** | Visualización de la media de paquetes recibidos por un nodo |
| **RF021** | Crear alerta cada vez que aparece un nuevo Host |
| **RF022** | Crear alerta si la IP del host y la MAC no coinciden |
| **RF023** | Analizar las propiedades de un host existente en la red |

**Tabla 9.** Historias de usuario tercer sprint

### 4.3.2 Diseño e implementación

Tras el desarrollo de los requisitos necesarios para realizar un análisis pasivo se procede al desarrollo de las actividades con el fin de realizar un análisis activo. Para la realización del análisis activo se hace uso de la librería Nmap adaptada para Python. Estos análisis activos se realizan sobre los host ya detectados en el análisis pasivo. Podemos listar todos los host almacenados en el sistema y pulsar en 'Analizar' para proceder con el análisis activo del host. Este análisis activo muestra información sobre la IP, la MAC, el sistema operativo, si

esta encendido y los puertos activos. Además para dar más información se muestra un panel donde se visualizan los paquetes enviados y recibidos durante las últimas 24 horas y a lo largo de los meses figura 10.

| Dato | Valor |
|---|---|
| IP | 192.168.0.53 |
| MAC | 28:63:36:F8:C5:66 |
| Sistema Operativo | Siemens AG |
| Estado | up |
| Puertos abiertos | tcp [443] |

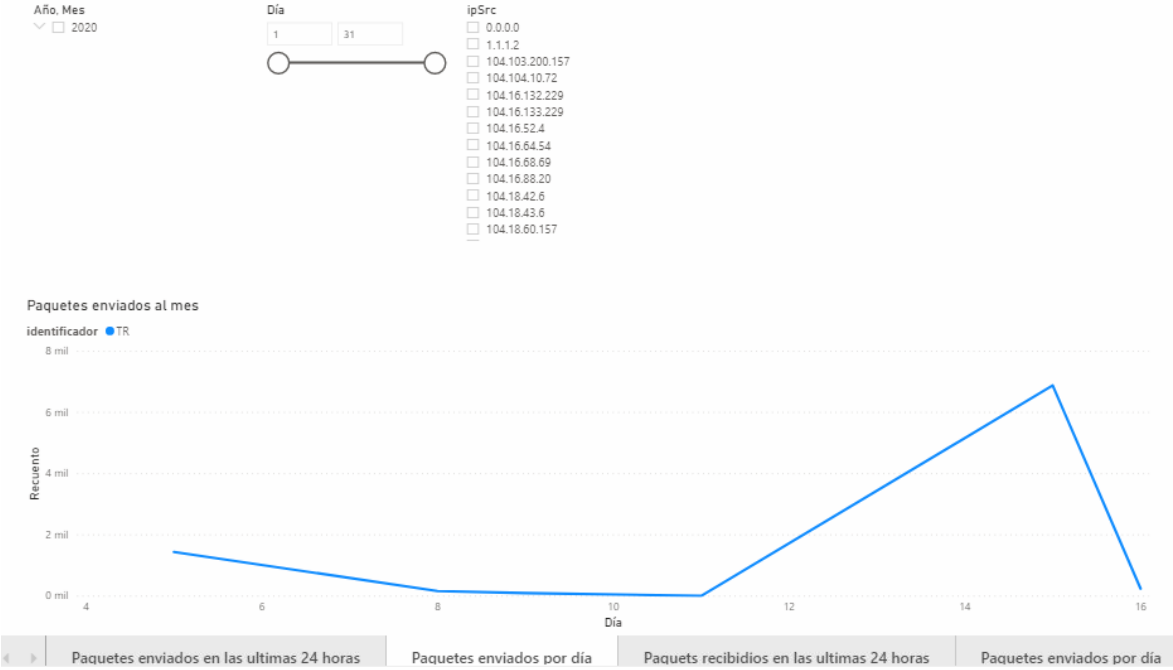


**Figura 10.** Análisis activo de un nodo

Se añaden funcionalidades para alertar sobre nuevos nodos descubiertos en el sistema o si hay cambios en los que ya se han detectados. Estas alertas están implementadas en el procedimiento ‘analizarPaquete’ dentro de la función ‘sniff’. Se mostrará en formato de tabla, una vez revisados los podemos eliminar de la base de información o marcar como revisados para tener constancia de su existencia figura 11.

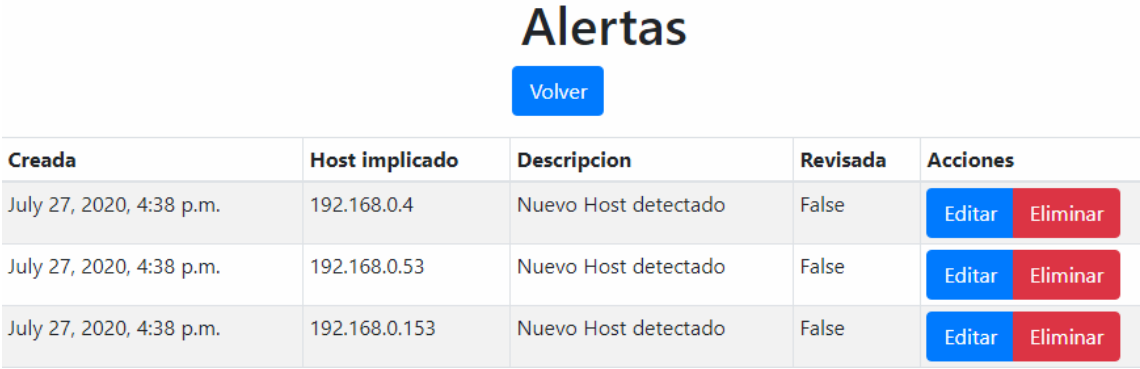


**Figura 11.** Vista de las alertas generadas al detectar un nuevo Host

Se añade funcionalidad para el análisis de archivos .pcap de manera que se puedan subir capturas de tráfico para ser analizadas de igual forma que el tráfico capturado en tiempo real. Estos archivo deben tener un identificador para diferenciar del tráfico recogido en tiempo real.

Se añade funcionalidad para ver los intentos de sincronización que se producen entre los nodos. De esta manera, podemos observar los puertos abiertos, y si se está produciendo un abuso en los intentos de sincronización de los nodos. El tamaño del nodo aumenta con el número de sincronizaciones, por lo que de un vistazo podemos ver que dispositivos son los más solicitados figura 12.

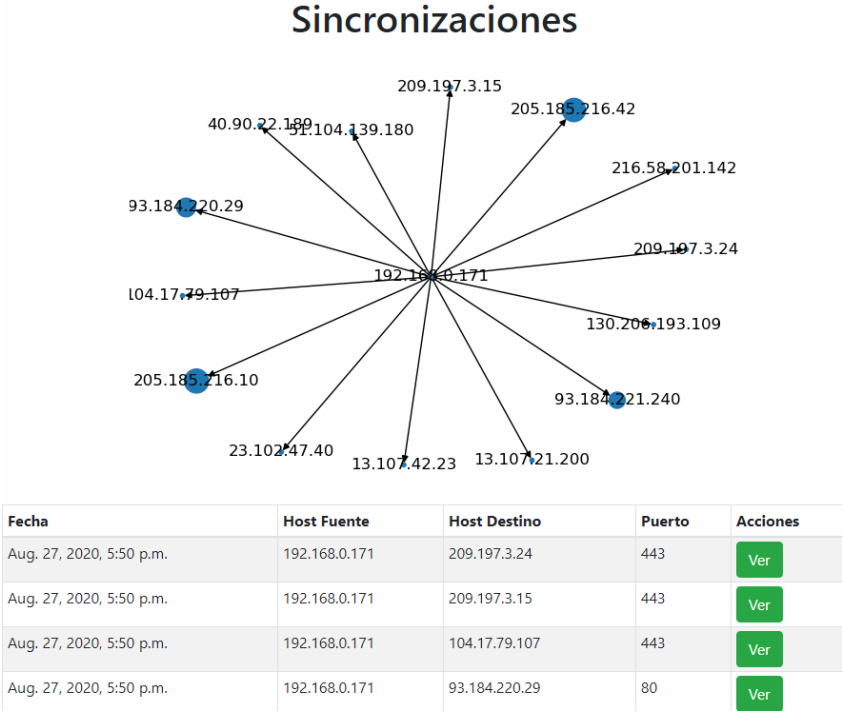


**Figura 12.** Grafo y tabla sobre intentos de sincronización

### 4.3.3 Validación

Tras el desarrollo de este sprint se realizan las siguientes validaciones tabla 10.

| Identificador | Validación | Comentarios |
|---|---|---|
| **RF016** | Validado con limitaciones | Podemos leer y analizar archivos pcap, pero debido al tratamiento que el programa hace de la memoria RAM estos archivos no deben de ser de un tamaño superior a 50MB |
| **RF017** | Validado | N/A |
| **RF018** | Validado | N/A |
| **RF019** | Validado | N/A |
| **RF020** | Validado | N/A |
| **RF021** | Validado | N/A |
| **RF022** | Validado | N/A |
| **RF023** | Validado | N/A |

**Tabla 10.** Validaciones tercer sprint

Volvemos a encontrarnos con el problema de la ocupación de memoria RAM al leer archivos .pcap. En este caso no conseguimos resolverlo mediante el workaround aplicado en el anterior sprint, por lo que tenemos una limitación para la lectura de archivos de manera offline. Esta limitación dependerá del tamaño de la memoria RAM donde se instale la plataforma web, actualmente, el tamaño máximo es de 50 MB. Tras la lectura del archivo se debe reiniciar la computadora para liberar la memoria RAM.

Se comprueba que aparecen todas las sincronizaciones en el grafo, y que además el tamaño del nodo que recibe los paquetes de red aumenta en relación a la cantidad de paquetes que recibe.

Se comprueba que al aparecer un nuevo host en la red aparece una alerta sobre el este host. Comprobamos que si la MAC (en inglés: Media Access Control) de uno de los host almacenados en el sistema cambia también aparece una alerta informando sobre el cambio.

Comprobamos que al realizar un análisis activo sobre los host permitidos aparece la información necesaria para identificar el nodo. Además añadimos información sobre la cantidad de información que envía o recibe de la red.

## 4.4 Sprint 4 - predicción del estado del sistema

Tras la implementación del análisis activo de la red procedemos a la implementación de ML que nos permitan realizar predicciones sobre el estado de la red. Para ello hacemos uso de un algoritmo regresión lineal.

### 4.4.1 Requisitos implementados

Los requisitos implementados durante este periodo de tiempo son los expuestos en la tabla 11.

| Identificador | Descripción |
|---|---|
| **RF024** | Estudio del modelo de datos |
| **RF025** | Implementación del algoritmo de ML |

**Tabla 11.** Requisitos implementados cuarto sprint

### 4.4.2 Diseño e implementación

Para estudiar el algoritmo a usar primero se recoge y analiza la cantidad de información que recorre la red. Tras varios días de observación podemos comprobar que la cantidad de paquetes y de información que recorre la red. Tras el análisis de los datos vemos una evidente relación entre la cantidad de paquetes que recorren la red y la cantidad de información que la recorre figura 13. Por lo que decidimos usar regresión lineal para predecir el comportamiento.

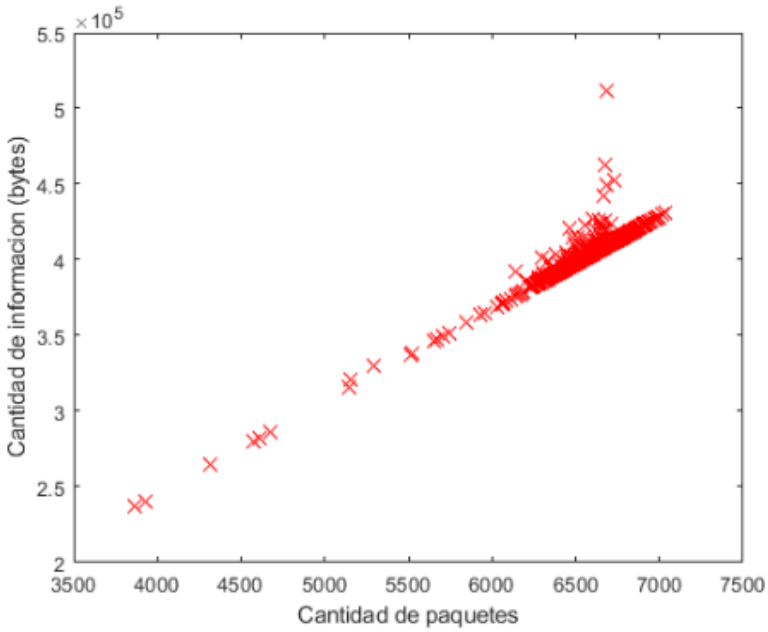


**Figura 13.** Modelo de datos recogido

Para encontrar la función que se adapte a nuestro modelo de datos necesitamos minimizar el valor de la función de coste [44]:

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Donde el valor de nuestra hipótesis $h_\theta(x)$ es dada por el modelo lineal [44]:

$$h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1$$

Los parámetros modelo son los valores de θ. Estos valores son los que se ajustaran para minimizar el coste de la función $J(\theta)$. Para realizar este ajuste utilizaremos un algoritmo de decrecimiento por lotes, donde cada iteración realiza una actualización de los valores de θ de manera simultánea [44]:

$$\theta_j := \theta_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

Por cada iteración del algoritmo los valores del parámetro 'J' estarán más cerca del mínimo coste. El parámetro 'α' de la ecuación nos indica la velocidad a la que queremos que la función de coste descienda. Este valor puede llegar a divergir si es demasiado grande, mientras que si es demasiado pequeño necesitaremos un número elevado de iteraciones para llegar al mínimo coste.

Para realizar el análisis del estado introducimos la cantidad de tiempo sobre el que queremos realizar el análisis. Se recogen la media cuadrática de los errores y se compara con el error recogido del modelo de datos. En caso de que el error sea superior se indicará que se ha detectado una anomalía en la red, mientras que si el error se mantiene dentro de los valores el funcionamiento de la red será correcto figura 14.

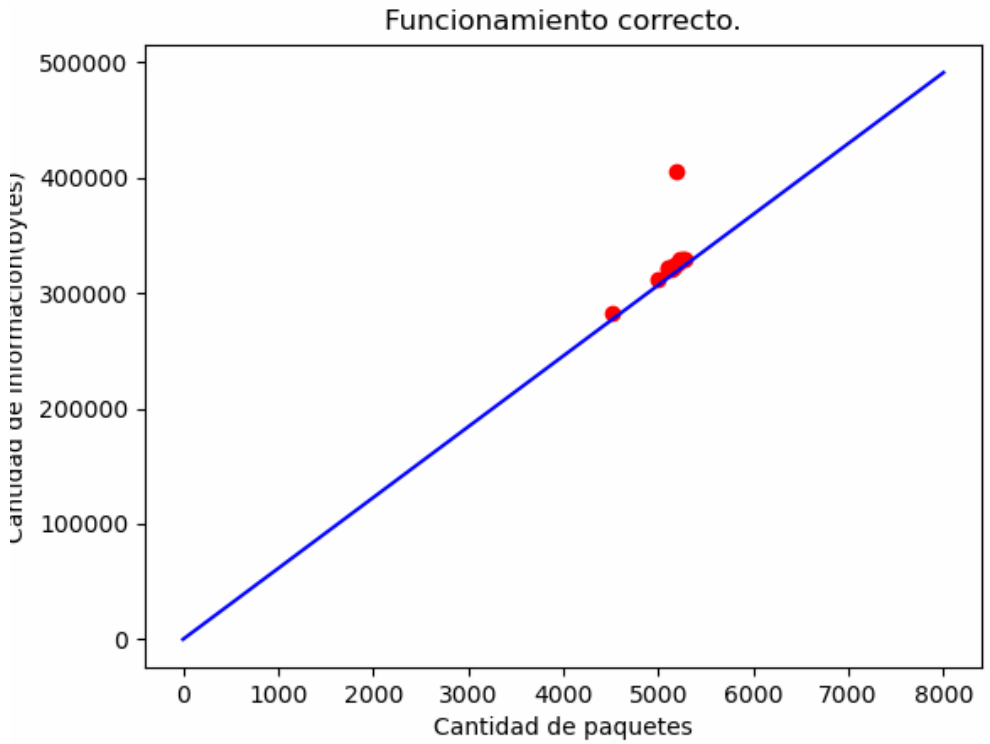


**Figura 14.** Feedback sobre el estado de la red

### 4.4.3 Validación

Tras el desarrollo de este sprint se realizan las siguientes validaciones tabla 12.

| Identificador | Validación | Comentarios |
|---|---|---|
| **RF024** | Validado | N/A |
| **RF025** | Validado | N/A |

**Tabla 12.** Validaciones cuarto sprint

Validamos que el algoritmo que utilizamos es útil para el análisis el comportamiento de la red, puesto que una disminución o aumento desmesurado en la cantidad de información que recorre la red es señal de posibles errores en la misma.

Validamos el correcto comportamiento cuando la red está funcionando correctamente. Para comprobar si detecta anomalías en la red realizamos dos pruebas sencillas. La primera aumentamos la cantidad de información de los paquetes guardado en la base de datos, y

comprobamos que al superar el error establecido se detecta la anomalía. La segunda prueba consiste en la reducción de la información que transportan los paquetes de datos causando de nuevo una reducción significativa, tras esto comprobamos que el error también es detectado.

## 4.5 Sprint 5 - identificación de usuarios

Tras la implementación del algoritmo predictivo nos disponemos a añadir una capa de seguridad más a nuestra plataforma web, restringiendo el acceso solo a usuarios registrados.

### 4.5.1 Requisitos implementados

Los requisitos implementados durante este periodo de tiempo son los expuestos en la tabla 13.

| Identificador | Descripción |
|---|---|
| **RF026** | Creación de usuarios en la plataforma web |
| **RF027** | Restringir el uso solo a usuarios registrados |
| **RNF004** | Las contraseñas de los usuarios deben ser guardadas con Hash y Salt |

**Tabla 13. Requisitos implementados cuarto sprint**

### 4.5.2 Diseño e implementación

Para la creación y gestión de usuarios utilizaremos los mecanismos que nos proporciona Django. Django nos permite gestionar usuarios y grupos, para ello es necesario utilizar dos módulos:

- Django.contrib.auth: contiene todos los elementos necesarios para gestionar la autenticación sus modelos por defecto.
- Django.contrib.contenttypes: permite que los permisos sean asociados.

Por defecto, django usa el algoritmo PBKDF2 ( Password-Based Key Derivation Function 2) [35] con un hash SHA256 [36] para la creación de contraseñas. Además también hemos habilitado las opciones de *salt* [37] a la hora de almacenar las contraseñas figura 15.


**Figura 15.** Uso de algoritmo, salt y hash para la creación de contraseñas

Se han añadidos dos tipos de usuarios, los usuarios administradores y los usuario de la plataforma web. Los usuarios administradores pueden crear, editar, borrar, habilitar o deshabilitar usuarios y resetear contraseñas figura 16, mientras que los usuarios de la plataforma solo pueden hacer uso de la herramienta.

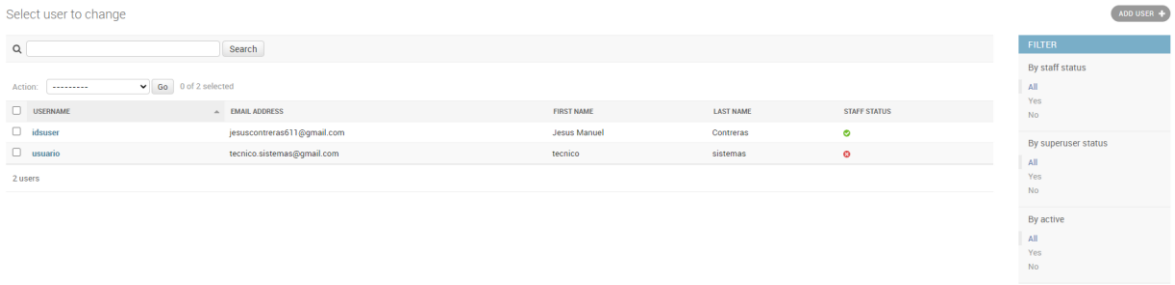


**Figura 16.** Panel de administración de usuarios

### 4.5.3 Validación

Tras el desarrollo de este sprint se realizan las siguientes validaciones tabla 14.

| Identificador | Validación | Comentarios |
|---|---|---|
| **RF026** | Validado | N/A |
| **RF027** | Validado | N/A |
| **RNF004** | Validado | N/A |

**Tabla 14.** Validación de requisitos quinto sprint

Comprobamos que se aplica el algoritmo criptográfico para el almacenamiento de las contraseñas en la base de datos figura 17.

| | id | password | last_login | is_superuser | username | first_name | last_name | email | is_staff | is_active |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | pbkdf2_sha256$120000$x7QimqgxUyUO$C3TYC8ngSFhV6MB... | 2020-09-19 09:55:39.1861500 | 1 | idsuser | | | | 1 | 1 |
| 2 | 2 | pbkdf2_sha256$120000$CMA0pMz8cJ65$hQvOWnBlFzW+Rf... | 2020-09-17 20:21:19.1576500 | 0 | usuario | | | | 0 | 1 |

**Figura 17.** Muestra contraseñas guardadas en la base de datos

Comprobamos que no se puede acceder a las páginas de la plataforma web sin autenticarnos, y comprobamos que nos redirige al enlace de inicio de sesión. Comprobamos que un usuario normal no puede acceder al panel de administración y mostramos mensaje de error figura 18.

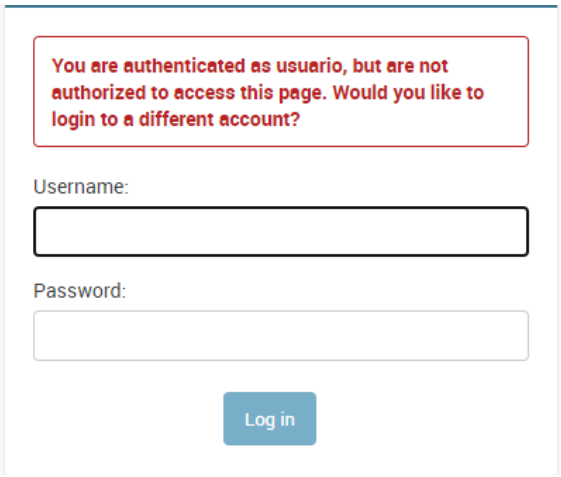


**Figura 18.** Mensaje de error al intentar acceder al panel de administración

# 5

# Conclusiones, incidencias y trabajo futuro

## 5.1 Conclusiones

Los objetivos principales propuestos al inicio del desarrollo del proyecto fueron la adquisición y análisis del tráfico de distintos dispositivos interconectados y la implementación de paneles visuales en forma de aplicación web. Estos objetivos han sido logrados y mejorados durante la consecución de cada sprint realizado.

La utilización de Scapy como escáner de red no ha sido satisfactoria, debido a la perdida de paquetes de información y al problema de uso de memoria RAM al realizar análisis de forma offline. Sin embargo, nos ha resultado muy practica a la hora de diseccionar los paquetes de red.

El uso de Django y Python como framework web y lenguaje de programación respectivamente ha sido todo un éxito. Inicialmente no se tenían conocimientos sobre ambos, pero la estructura y creación del proyectos en de Django nos ha resultado muy intuitiva y fácil de seguir.

La utilización de Power BI como herramienta para recolectar datos y crear paneles visuales vistosos permite la creación de gráficas que dan una gran cantidad de información sobre la red, el uso de esta herramienta ha permitido que la web sea más dinámica.

Se han adquirido una gran cantidad de conocimiento sobre algoritmos de ML, en concreto sobre redes neuronales. Inicialmente se intentó adaptar una red neuronal para la predicción del comportamiento en la red, dicha red neuronal no fue finalmente implementada debido a que no se disponía del tiempo suficiente para realizar una implementación tan compleja que produjese unos buenos resultados. Finalmente se decidió utilizar un algoritmo de regresión, el cual se ajusta con mucha precisión al modelo de datos que tenemos.

Por último destacar todo lo aprendido sobre la seguridad informática, especialmente en IDS. Este campo de la seguridad informática ha sido ampliamente trabajado, desarrollando multitud de algoritmos y métodos diferentes para proteger nuestros sistemas.

## 5.2 Incidencias

Power BI es una herramienta muy versátil pero nos hemos encontrado con una limitación a la hora de procesar los datos. Está limitación es que para la actualización de la información en tiempo real hace falta una suscripción de pago a Power BI PRO, teniendo que actualizar los datos recogidos de manera manual, es decir, abriendo la aplicación, actualizando los datos existentes en la base de datos y después subirlos a nuestro workspace para que se reflejen estos cambios en la plataforma web

Por otro lado comentar brevemente que debido a la COVID-19 se nos han presentado distintas dificultades en el desarrollo del TFG, como por ejemplo, la imposibilidad del acceso físico al entorno, el trabajo y la coordinación adicional necesaria por parte de los técnicos que trabajan en el entorno, la coordinación entre las actividades que desarrollaron errores en los IDS del entorno industrial, etc.

## 5.3 Trabajo futuro

Posibles líneas de investigación podrían ser:

- Utilización de Scapy para la disección y el análisis de distintos protocolos utilizados en entorno industrial como podrían ser Modbus [38], DNP3 [39], Profinet [40], etc.
- Actualización de licencia de Power BI para la creación de paneles visuales que se actualicen automáticamente y se pueda realizar el seguimiento en tiempo real de estado de la red.
- Creación de una red neuronal para la predicción añadiendo más parámetros que permitan la detección de distintas anomalías en el entorno.
- Migración de la web y la base de datos a la nube pública de Azure para hacer uso de las herramientas ofrecidas por la nube pública de Microsoft.

# 6

# Bibliografía

[1] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, Daesung Moon, and Jong Hyuk Park. A comprehensive study on apt attacks and countermeasures for future networks and communications: challenges and solutions. The Journal of Supercomputing, pages 1–32, 2016.

[2] Docs.djangoproject.com. 2020. *Django Documentation | Django Documentation | Django*. [en línea] Disponible en: <https://docs.djangoproject.com/en/2.2/> [ultimo acceso, 22 Septiembre 2020]. [3] Python.org. 2020. Welcome To Python.Org. [en línea] Disponible en: <https://www.python.org/> [último acceso, 22 Septiembre 2020].

[4] Docs.microsoft.com. 2020. Documentación Técnica De SQL Server - SQL Server. [en línea] Disponible en: <https://docs.microsoft.com/es-es/sql/sql-server/> [último acceso 22 Septiembre 2020].

[5] Docs.microsoft.com. 2020. Documentación De Power BI - Power BI. [en línea] Disponible en: <https://docs.microsoft.com/es-es/power-bi/> [último acceso, 22 Septiembre 2020].

[6] Windows. 2020. Explora El Sistema Operativo, Los Equipos, Las Apps Y Más Con Windows 10 | Microsoft. [en línea] Disponible en: <https://www.microsoft.com/es-es/windows> [último acceso, 22 Septiembre 2020].

[7] Letelier P, Penadés MC. Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP) [En línea]. CyTA. 2006. Disponible en: <https://www.cyta.com.ar/ta0502/v5n2a1.htm> [último acceso 2 junio 2020].

[8] Agilemanifesto.org. 2020. Manifiesto Por El Desarrollo Ágil De Software. [en línea] Disponible en: <https://agilemanifesto.org/iso/es/manifesto.html> [último acceso, 22 Septiembre 2020].

[9] CASTELLS, Manuel; CHEMLA, Paul. La galaxia internet. 2001. Disponible en: < https://irla.cat/wp-content/uploads/2017/06/La_Galaxia_Internet.pdf> [último acceso, 22 Septiembre 2020]

[10] A. Khan and K. Turowski. A survey of current challenges in manufacturing industry and preparation for industry 4.0. *In First International Scientific Conference "Intelligent Information Technologies for Industry" (IITI'16)*, pp. 15–26. Springer International Publishing, 2016.

[11] Saurabh Singh, Pradip Kumar Sharma, Seo Yeon Moon, Daesung Moon, and Jong Hyuk Park. A comprehensive study on apt attacks and countermeasures for future networks and communications: challenges and solutions. *The Journal of Supercomputing*, pp. 1–32, 2016.

[12] J . E. Rubio, C . Alcaraz, R. Roman, and J . Lopez, "Current Cyber-Defense Trends in Industrial Control Systems", Computers & Security Journal, Elsevier, Volume 87, 2019, ISSN 0167-4048

[13] Miloslavskaya. Analysis of SIEM systems and their usage in security operations and security intelligence centers. *In First International Early Research Career Enhancement School on Biologically Inspired Cognitive Architectures*, pp. 282-288. Springer, Cham, 2017.

[14] Xin Fan, Wenjie Luo, Xiaoju Dong, and Rui Su, "A Network Visualization System for Anomaly Detection and Attack Tracing" ICPCSEE 2018, CCIS 901, pp. 560–574, 2018.

[15] Shadi Aljawarneh, Monther Aldwairi, Muneer Bani Yassein, Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model, *Journal of Computational Science*, Volume 25, 2018, Pages 152-160

[16] Julián, G., 2020. Las Redes Neuronales: Qué Son Y Por Qué Están Volviendo. [en línea] Xataka.com. Disponible en: <https://www.xataka.com/robotica-e-ia/las-redes-neuronales-que-son-y-por-que-estan-volviendo> [último acceso, 22 Septiembre 2020].

[17] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," in *IEEE Access*, vol. 7, pp. 41525-41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[18] aula21 | Formación para la Industria. 2020. Qué Es Un Sistema SCADA, Para Qué Sirve Y Cómo Funciona | Aula21. [en línea] Disponible en: <https://www.cursosaula21.com/que-es-un-sistema-scada/> [último acceso, 22 Septiembre 2020].

[19] Profesores.fi-b.unam.mx. 2020. [en linea] Disponible en: <http://profesores.fi-b.unam.mx/cintia/analisispa.pdf> [último acceso 22 Septiembre 2020].

[20] Ríos, Rene, Fermin, José R., ANÁLISIS DE TRÁFICO DE UNA RED LOCAL UNIVERSITARIA. [en linea]. 2009;8(2):15-27. Recuperado de: https://www.redalyc.org/articulo.oa?id=78411787002

[21] Nmap.org. 2020. Nmap: The Network Mapper - Free Security Scanner. [en linea] Disponible en: <https://nmap.org/> [último acceso, 22 Septiembre 2020].

[22] Scapy.readthedocs.io. 2020. Welcome To Scapy'S Documentation! — Scapy 2.4.4. Documentation. [en linea] Disponible en: <https://scapy.readthedocs.io/en/latest/> [último acceso, 22 Septiembre 2020].

[24] C. Alcaraz, J. Rodriguez, R. Roman, and J. E. Rubio, "Estado y Evolución de la Detección de Intrusiones en los Sistemas Industriales", III Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), 2017.

[25] ASALE, R., 2020. Protocolo | Diccionario De La Lengua Española. [en linea] «Diccionario de la lengua española» - Edición del Tricentenario. Disponible en: <https://dle.rae.es/protocolo> [último acceso, 22 Septiembre 2020].

[26] Universidadviu.es. 2020. Qué Es Y Cómo Funciona El Protocolo Ip | VIU. [en linea] Disponible en: <https://www.universidadviu.es/funciona-protocolo-ip/> [último acceso, 22 Septiembre 2020].

[27] Amer Nizar Abu Ali, "Comprarison study between IPV4 & IPV6", *IJCSI International Journal of Computer Science Issues*, Vol. 9, Issue 3, No 1, May 2012.

[28] IETF. 2020. Home. [en línea] Disponible en: <https://www.ietf.org/> [último acceso, 22 Septiembre 2020].

[29] Emre Durdagi , Ali Buldu, "IPV4/IPV6 security and threat comparisons", *Procedia - Social and behavioral Sciences*, 2010.

[30] BBVA NOTICIAS. 2020. Te Contamos Qué Es El 'Machine Learning' Y Cómo Funciona. [en linea] Disponible en: <https://www.bbva.com/es/machine-learning-que-es-y-como-funciona/> [último acceso, 22 Septiembre 2020].

[31] Claudia Russo, Hugo Ramón, Nicolás Alonso, Benjamin Cicerchia, Leonardo Esnaola, Juan Pablo Tessore, ""Tratamiento Masivo de Datos Utilizando Técnicas de Machine Learning" Documento de conferencia. XVIII Workshop de Investigadores en Ciencias de la Computación (Entre Ríos, Argentina), WICC 2016

[32] Servicios de informática en la nube | Azure.microsoft.com. 2020.[en línea] Disponible en: <https://azure.microsoft.com/es-es/> [último acceso, 22 Septiembre 2020]

[33] ¿Qué es Power BI? - Power BI. Docs.microsoft.com. 2020 . [en línea] Disponible en: <https://docs.microsoft.com/es-es/power-bi/fundamentals/power-bi-overview> [último acceso, 22 Septiembre 2020]

[34] Wireshark · Go Deep. Wireshark.org. 2020. [en línea] Disponible en: <https://www.wireshark.org/> [último acceso, 22 Septiembre 2020]

[35] ERTAUL, Levent; KAUR, Manpreet; GUDISE, Venkata Arun Kumar R. Implementation and performance analysis of PBKDF2, Bcrypt, Scrypt algorithms. En *Proceedings of the International Conference on Wireless Networks (ICWN)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2016. p. 66.

[36] RACHMAWATI, D.; TARIGAN, J. T.; GINTING, A. B. C. A comparative study of Message Digest 5 (MD5) and SHA256 algorithm. En *Journal of Physics: Conference Series*. 2018. p. 012116.

[37] Arias D. Adding Salt to Hashing: A Better Way to Store Passwords . Auth0 - Blog. 2020 [en línea] Disponible en:<https://auth0.com/blog/adding-salt-to-hashing-a-better-way-to-store-passwords/> [último acceso, 22 Septiembre 2020][38] Modbus TCP/IP - Automation Networks. 2020 [en línea] Disponiblen en: <http://automation-networks.es/glossary/modbus-tcpip> [último acceso, 22 Septiembre 2020].

[39] Protocolo DNP3 - Logitek. 2020. [en línea] Disponible en: <https://logitek.es/productos-lk/dnp3> [último acceso, 22 Septiembre 2020].

[40] Protocolo PROFINET IO Logicbus. 2020. [en línea] Disponiblen en:<https://www.logicbus.com.mx/blog/que-es-el-protocolo-profinet-> [último acceso, 22 Septiembre 2020].

[41] Rosen R. (2014) Internet Control Message Protocol (ICMP). In: Linux Kernel Networking. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4302-6197-1_3

[42] F. Q. Lauzon, "An introduction to deep learning," 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), Montreal, QC, 2012, pp. 1438-1439, doi: 10.1109/ISSPA.2012.6310529.

[43] Power BI en Microsoft Learn. Docs.microsoft.com. 2020. [en línea] Disponible en: <https://docs.microsoft.com/es-es/power-bi/guided-learning/introductiontodax> [último acceso, 22 Septiembre 2020]

[44] Coursera | Aprendizaje Automático by Standord. Andrew Ng. 2020. [en línea] Disponible en: < https://www.coursera.org/learn/machine-learning> [último acceso, 22 Septiembre 2020]

# Anexo A

# Manual de instalación

**SLQ Server Express 14**

Instalación de los archivos "**SQLServer2017-SSEI-Expr.exe**" y "**SSMS-Setup-ENU.exe**". El primero es el motor la base de datos. El segundo es para administrar instancias de bases de datos de manera gráfica. En ambos seguimos las instalación por defecto. Creamos la instancia.

**Habilitar conexiones TCP/IP y configurar puerto de la conexión.**

1. Accedemos a la aplicación de SQL Server Configuration Manager.
2. Abrimos las opciones dentro de configuración de red de SQL Server
3. Pulsamos sobre protocolos.
4. Habilitamos protocolo TCP/IP.
5. Pulsamos sobre propiedad y hacemos scroll hasta el final, añadimos el puerto. Por defecto suele ser 1433 figura 19.

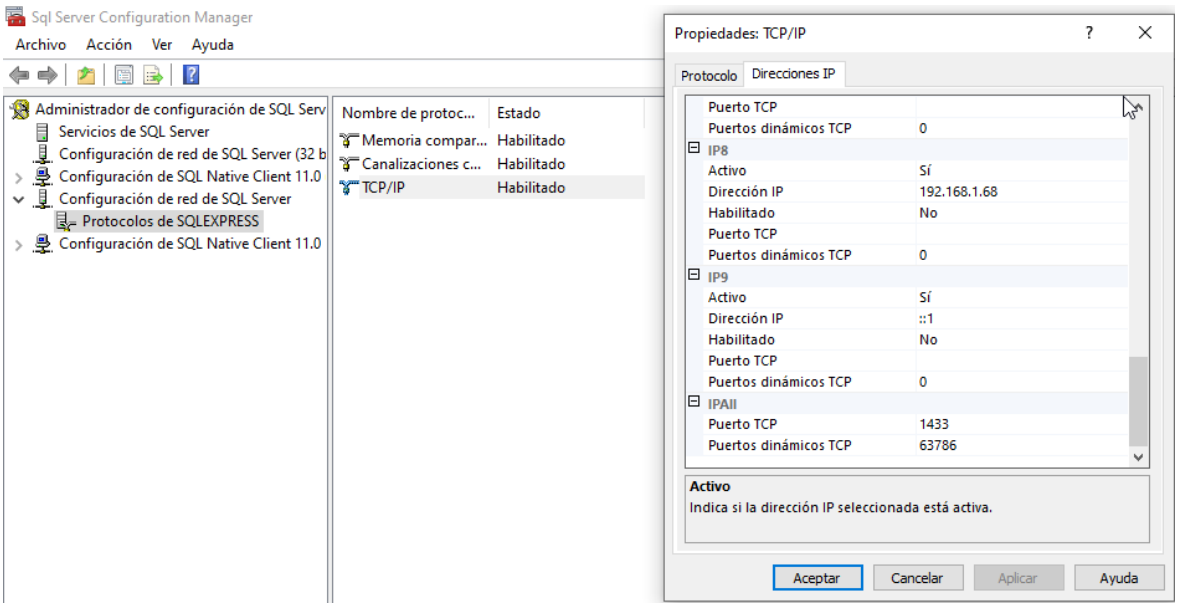


**Figura 19.** Configuración puertos SQL Server

**Habilitar la autenticación mixta**

1. Abrimos SQL Server Management Studio.
2. Pulsamos sobre la instancia con botón derecho y abrimos propiedades figura 20.

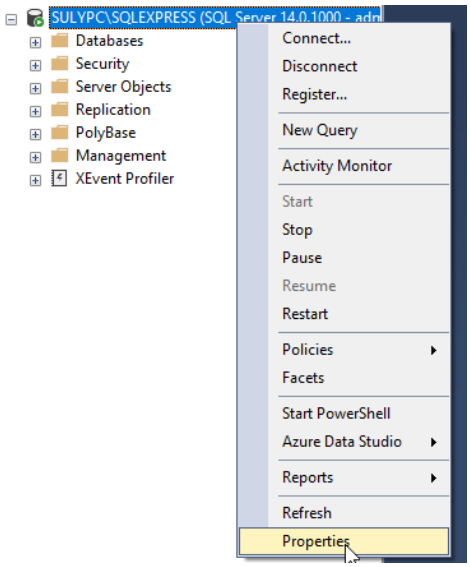


**Figura 20.** Propiedades de la base de datos

3. Dentro de seguridad habilitamos la autenticación de SQL Server y Windows figura 21.

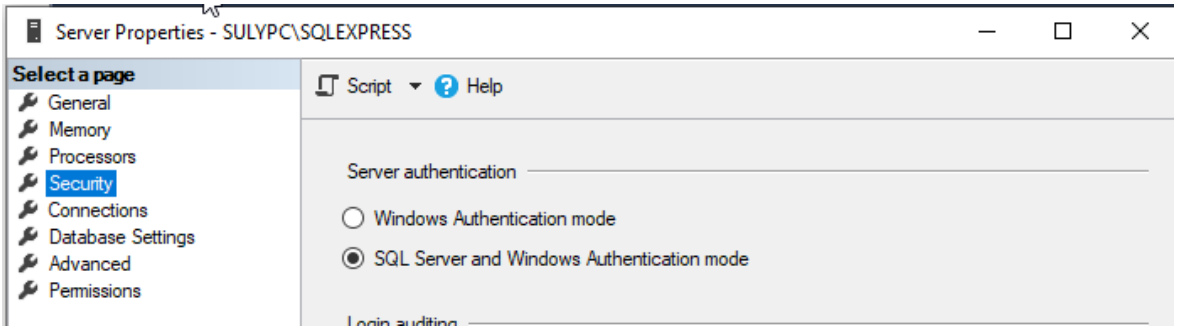


**Figura 21.** Cambiar modo de autenticación

4. Desplegamos la carpeta de security y pulsamos botón derecho sobre logins figura 22:

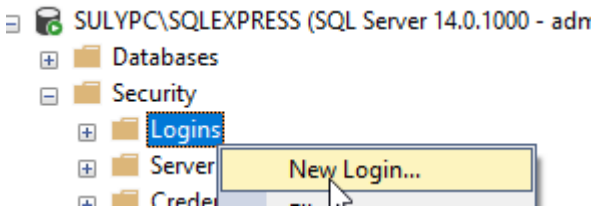

**Figura 22.** Nuevo login

5. En general pulsamos sobre SQL Server authentication, rellenamos Login name y nos aseguramos de que enforce password policy esté deshabilitado figura 23.

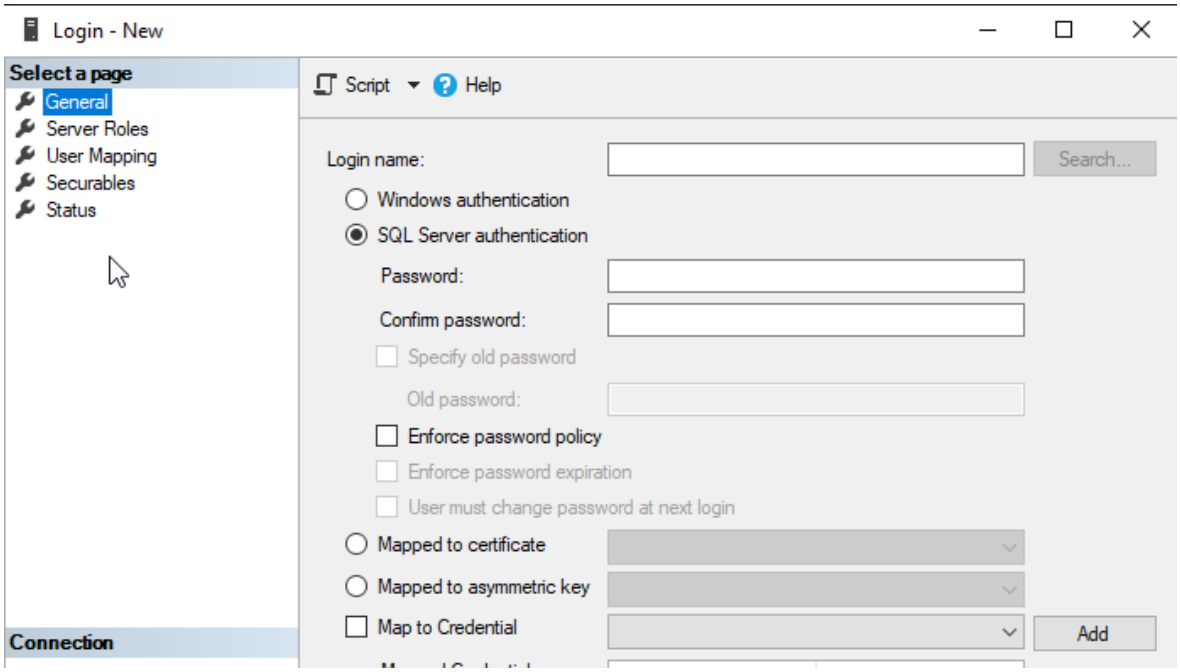


**Figura 23.** Datos a rellenar del usuario

6. Pulsamos sobre Server Roles y clicamos todos sobre todas las opciones. Pulsamos OK para guardar el usuario de SQL server figura 24.

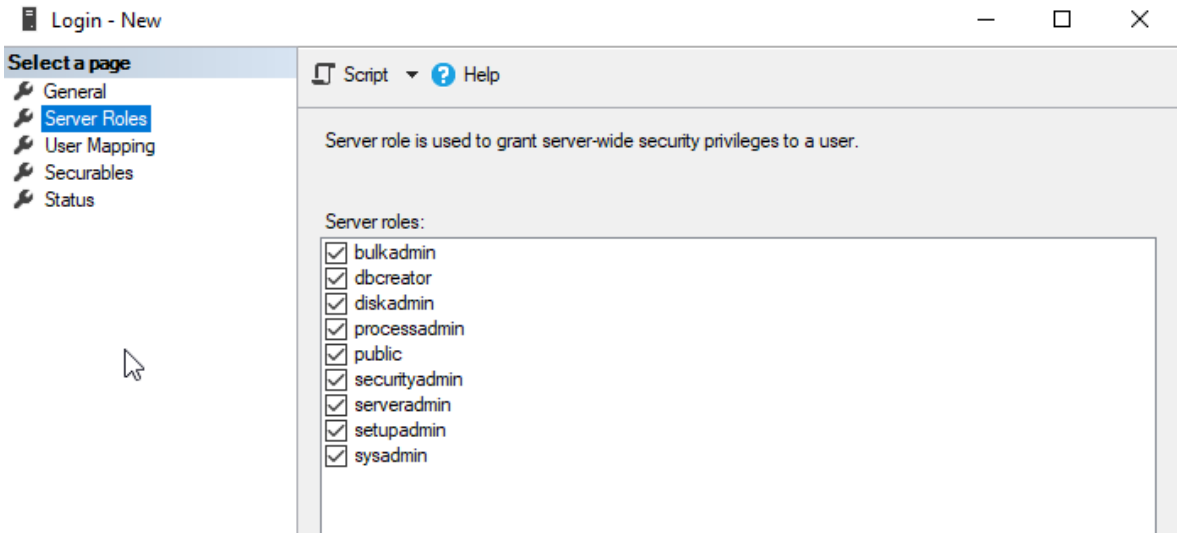

**Figura 24.** Asignación de permisos al usuario

7. Pulsamos sobre el botón derecho sobre el folder databases y creamos una nueva base de datos figura 25.

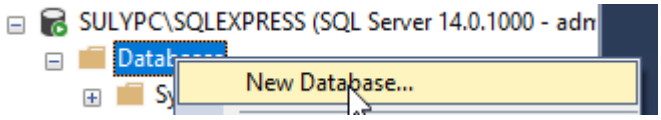


**Figura 25.** Creación de la base de datos

8. Nos salimos de SQL Server y vamos al programa IDS. En el archivo situado dentro de IDS\IDS\settings.py debemos añadir (sobre la línea 95) los datos correspondientes a lo que hemos creado figura 26:

```
DATABASES = {
    'default': {
        'ENGINE': 'sql_server.pyodbc',
        'NAME': 'IDS',
        'USER': 'adminIDS',
        'PASSWORD': 'adminIDS',
        'HOST': '127.0.0.1',
        'PORT': '1433',
        'OPTIONS': {
            'driver': 'ODBC Driver 17 for SQL Server',
            }
    },
}
```

**Figura 26.** Datos de la conexión a la base de datos

9. Instalamos el driver ODBC correspondiente. El archivo llamado "**msodbcsql.msi**"

**Python 3.7.7**

Seguir instalación por defecto del archivo "**python-3.7.7-amd64.exe**"

**Instalar Winpcap o Npcap**

Seguir instalación por defecto de "**Win10Pcap-v10.2-5002.msi"**

**Instalar Nmap**

Seguir instalación por defecto de "**nmap-7.80-setup.exe**". No instalar npcap en la instalación de nmap. Si instalamos la versión de nmap se produce un error en la ejecución de sniff de scapy.

En caso de instalación por error ejecutar el archivo "**npcap-0.9991.exe**", cuya versión es la correcta.

**Instalación de Django y dependencias del proyecto**

Ejecutar el script de instalación "**IDS_Instalation.ps1**".

**Ejecución del programa**

Para cargar la base de datos debemos utilizar el comando "Python manage.py migrate" en la misma ruta donde se encuentre el programa manage.py figura 27.

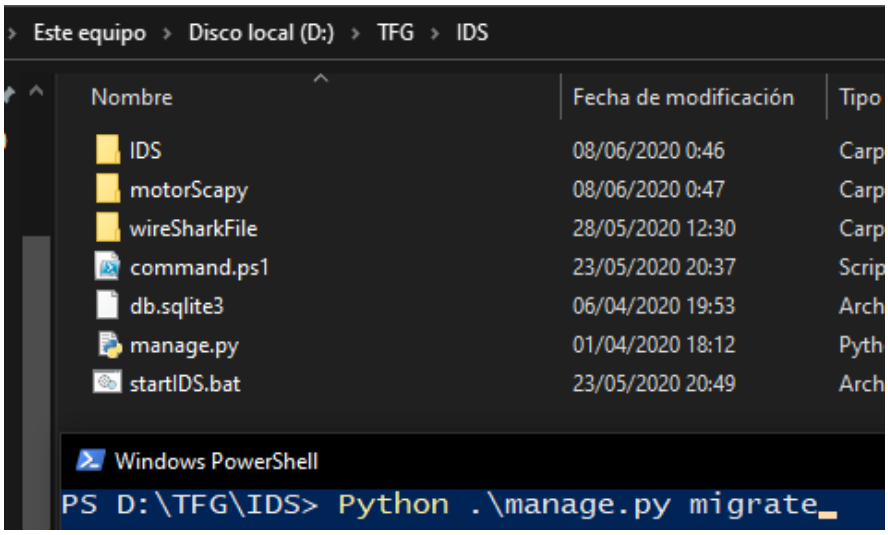


**Figura 27.** Comando para la creación de tablas

Una vez hecho esto se cargarán las tablas en la base de datos. Tras esto utilizamos el comando "Python manage.py runserver" para iniciar el programa, debe de salir lo siguiente figura 28:

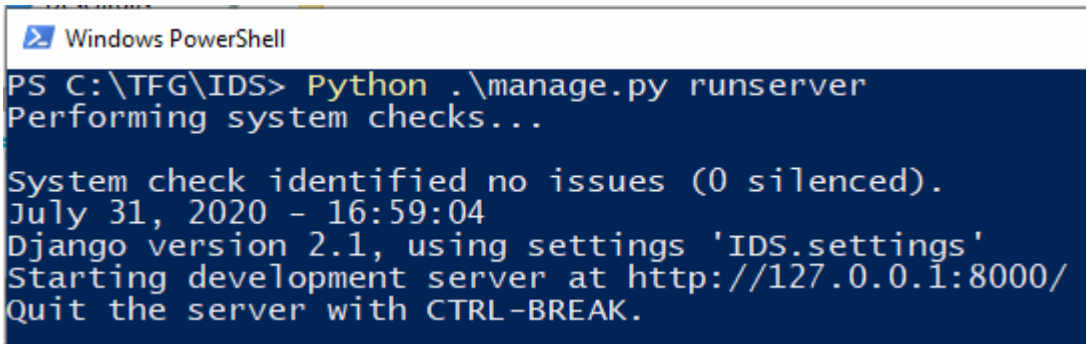


**Figura 28.** Correcto inicio de la aplicación

Para cargar la página web nos vamos al navegador (chrome por ejemplo) y copiamos la URL indicada, en mi caso "http://127.0.0.1:8000/"

No deberían de aparecer errores de dependencias, pero en caso de que aparezcan ejecutamos pip install nombreDependencia para realizar la instalación.

**Creación de usuario administrador**

Para poder acceder a las distintas funcionalidades del programa primero debemos crear un usuario administrador. Abrimos la consola de comando en la misma ruta en la que ejecutamos el servidor y escribimos el comando "Python .\manage.py createsuperuser". Esto iniciará el proceso de creación del usuario administrador.

Una vez dentro en la pestaña de configuración debemos de añadir la interfaz de red desde la que queremos analizar el tráfico. Inicialmente el apartado de En ejecución debe estar sin el check. Una vez creada le damos a Editar y la chequeamos para que inicie el análisis del tráfico figura 29.

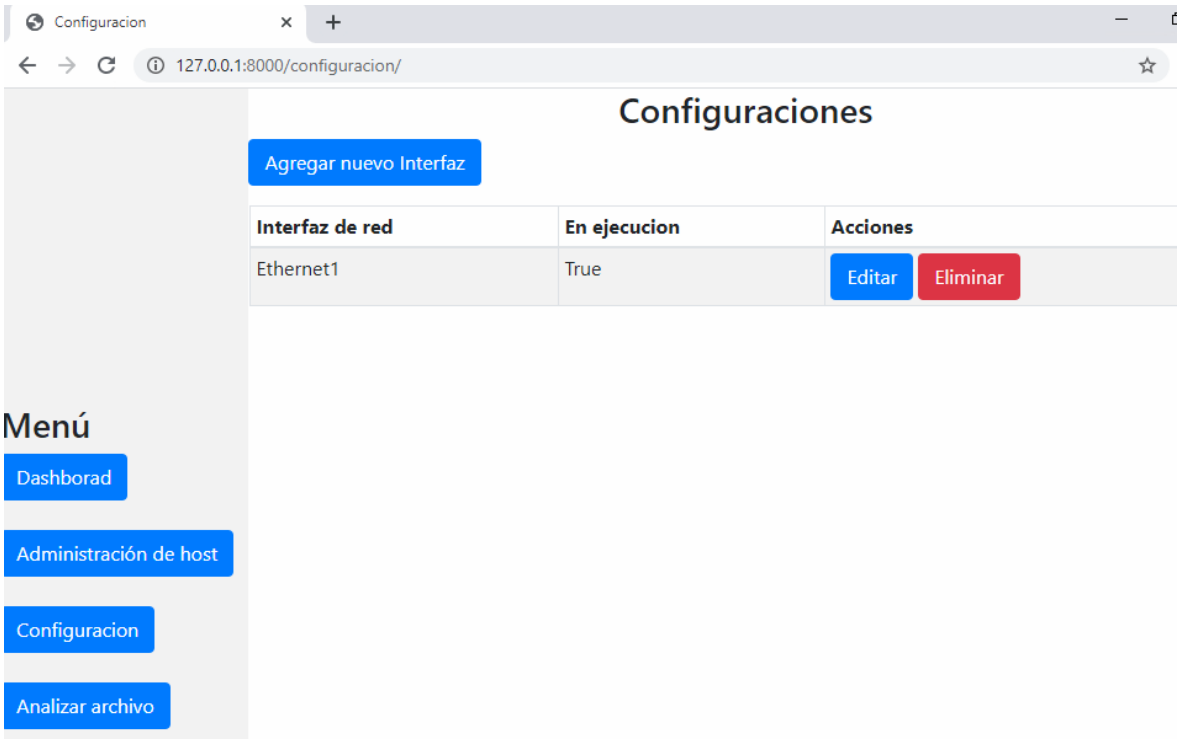


**Figura 29.** Configuración de la interfaz de la plataforma web

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