UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

DESARROLLO DE UNA HERRAMIENTA PARA EL ANÁLISIS DE TRÁFICO DE INTERNET EN TIEMPO REAL

TRABAJO FIN DE GRADO

Julio -2025

AUTOR: ISLAME ALKAHFI

DIRECTOR/ES: OSCAR MARTINEZ BONASTRE

AGRADECIMIENTO

Quisiera hacer llegar mi más profundo agradecimiento a mi tutor, Oscar Martínez Bonastre, por el acompañamiento y la orientación que me brindó durante todo el proceso de este trabajo. Su experiencia y dedicación fueron realmente cruciales para llevar a cabo y finalizar este trabajo.

A mi familia, gracias por acompañarme siempre, por su ánimo constante, Y, sobre todo, a mis padres: no existen palabras que alcancen para mostrar mi gratitud por el enorme apoyo que me brindaron. Por sus sacrificios, su amor incondicional, dedico este título con todo mi corazón a ustedes.

RESUMEN

Hoy en día, analizar el tráfico de internet es un tema importante en el mundo de telecomunicaciones, es un método empleado por los gestores de redes para observar el comportamiento de la red, controlar su disponibilidad y detectar actividades extrañas. Por lo tanto, desarrollar una herramienta que haga más sencillo examinar el análisis del tráfico en internet de manera eficaz y al instante se transforma en una alternativa indispensable para administrar la información transmitida.

El objetivo de este trabajo de fin de grado es desarrollar una plataforma destinada al análisis del flujo de internet, centrando en la evaluación del tráfico relacionado con el nombre de dominios DNS utilizando tecnologías como PHP, JavaScript, MySQL y Python con Scapy. La herramienta permite visualizar informaciones como el volumen de tráfico, tipos de protocolos, tipos de registros DNS y dominios consultados a través de gráficos interactivos y filtros personalizados.

Los resultados demuestran que la herramienta procesa, analiza y representa datos de tráfico de internet de manera efectiva, facilitando al usuario la visualización clara y en tiempo real de información que previamente era invisible.

Este proyecto describe la configuración y la implementación técnica de esta herramienta que está propuesta para capturar, analizar y visualizar del tráfico de internet generado por un dispositivo de computación. Se presentan las características técnicas del entorno de recolección, el método utilizado para procesar los paquetes, poniendo un enfoque particular en el tráfico DNS. También se discuten los resultados obtenidos en los elementos visuales. Además, se examinan las limitaciones técnicas vinculadas a la hora de implementación de esta plataforma y posibles mejoras futuras que se podrían incorporar a la herramienta.

La plataforma se llama “TrafAnalyzer”, y este nombre es en inglés y surge de la combinación de “Traffic” y “Analyzer” que dos términos importantes relacionados con el tema. Así que es un nombre relevante y adecuado a lo que estamos examinando en el trabajo.

Palabras claves: Análisis de tráfico de internet, Captura de paquetes, Monitorización en tiempo real, DNS (Domain Name System), Visualización de datos.

ABSTRACT

Nowadays, analyzing internet traffic is an important topic in the telecommunications world; it is a method used by network managers to observe network behavior, control its availability, and detect unusual activities. Therefore, developing a tool that makes it easier to examine internet traffic analysis effectively and instantly becomes an indispensable alternative for managing transmitted information.

The objective of this final degree project is to develop a platform aimed at analyzing internet traffic, focusing on the evaluation of traffic related to DNS domain names using technologies such as PHP, JavaScript, MySQL, and Python with Scapy. The tool allows users to visualize information such as traffic volume, types of protocols, types of DNS records, and queried domains through interactive graphs and custom filters.

The results demonstrate that the tool effectively processes, analyzes, and represents internet traffic data, enabling the user to clearly and in real-time visualize information that was previously invisible.

This project describes the configuration and technical implementation of this tool, which is proposed to capture, analyze, and visualize internet traffic generated by a computing device. The technical characteristics of the collection environment, the method used to process the packets, with a particular focus on DNS traffic, are presented. The results obtained in the visual elements are also discussed. Additionally, the technical limitations associated with the implementation of this platform and possible future improvements that could be incorporated into the tool are examined.

The platform is called "TrafAnalyzer," and this name is in English, arising from the combination of "Traffic" and "Analyzer," which are two important terms related to the topic. So, it is a relevant and appropriate name for what we are examining in the work.

Keywords: Internet traffic analysis, Packet capture, Real-time monitoring, DNS (Domain Name System), Data visualization.

ÍNDICE

INDICE DE FIGURAS

LISTA DE ACRÓNIMOS

DNS	Domain Name System
IP	Internet Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
SMTP	Simple Mail Transfer Protocol
Ntopng	Network Top (Next Generation)
PRTG	Paessler Router Traffic Grapher
MYSQL	My Structured Query Language
XAMPP	Cross-platform Apache MariaDB PHP Perl
PHP	Hypertext Preprocessor
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
Npcap	Network Packet Capture
Nmap	Network Mapper
GeoIP	Geolocation Internet Protocol

1. INTRODUCCIÓN

1.1. Organización

Este trabajo final de grado se divide en varias secciones que presentan todo el proceso de desarrollo de una herramienta para capturar y examinar el tráfico de internet en un dispositivo local.

Primero, la introducción contiene una parte que explica la ordenación del trabajo dando una breve idea sobre cada capítulo, otro apartado que indica los motivos por lo que he elegido este tema, también, una sección donde se cita las herramientas que comparte casi el mismo enfoque que esta plataforma demostrando la necesidad de crear una herramienta fácil de usar para analizar el tráfico de internet, la última parte que detalla los objetivos principales de este trabajo.

A continuación, en la segunda sección de materiales y métodos se definen las herramientas y tecnologías utilizadas en la plataforma como Scapy, MySQL Workbench y XAMPP, también se describen los lenguajes de programación y de estilo usadas en la implementación. En cada descripción se presenta la justificación por lo que hemos elegido trabajar con estos materiales.

En el tercer capítulo sobre el diseño y la arquitectura de la plataforma, se ofrece una visión general de su diseño y arquitectura, subrayando los componentes esenciales y sus interconexiones. También se adjunta un diagrama de flujo que presenta los elementos principales del sistema y sus enlaces.

En la cuarta sección, el núcleo técnico de la plataforma, se detalla la configuración inicial, la evolución del script de captura y el script de consultas, así como la base de datos y la plataforma misma. Aparte, se expresa la comunicación entre el servidor y el cliente para materializar la herramienta. Además, en este capítulo se ocupa del procesamiento y explicación de datos, profundiza en el método de análisis de datos utilizado por la herramienta y su traducción visual para facilitar su incorporación. Se emplean gráficos dinámicos para presentar informaciones relevantes como el volumen de tráfico o las consultas y respuestas DNS, entre otras, ayudando así al usuario a detectar cómo se comporta su equipo.

La quinta parte de los resultados y discusión presenta los resultados obtenidos a partir del análisis de tráfico de internet y evalúa su efectividad y compara los diferentes resultados obtenidos en cada usuario.

La sexta parte, las conclusiones y trabajo futuro, resume lo que logramos y si alcanzamos los objetivos que nos habíamos propuesto. También se exponen las principales limitaciones encontradas y se proponen posibles líneas de mejora, como la incorporación de nuevas funcionalidades.

Por último, añadimos la bibliografía que consultamos mientras realizamos el trabajo.

1.2. Motivación

A lo largo de mi formación en Ingeniería de Tecnologías de Telecomunicación, especialmente en el área de Telemática, encontré varios temas que me motivaron a decidirme por este trabajo. Me interesó bastante el estudio de datos de red, especialmente aquellos propios de Internet, considerándolo esencial para el control y monitoreo de los dispositivos de comunicación. Además, consolidé mis conocimientos creando una plataforma durante las prácticas de una asignatura llamada "Aplicaciones Telemáticas", donde aprende los fundamentos esenciales para crear una aplicación web. Asimismo, descubrí que tengo un interés de crear una herramienta que puede ayudar a los estudiantes y los profesores de mi carrera especialmente la mención telemática, en otras palabras, busco dejar una huella significativa en mi trayectoria universitaria.

A lo largo de mi educación universitaria, cursé materias sobre redes, protocolos de comunicación y programación, sobre todo en el campo de las aplicaciones web. Esto me proporcionó una base teórica sólida para construir una plataforma como la que propongo.

Mi habilidad en la creación de sitios web se consolidó gracias al aprendizaje práctico durante la carrera. He logrado desarrollar una herramienta que examina la gestión hotelera asumiendo la organización de manera eficiente los datos esenciales, tales como información detallada de clientes, reservas y detalles laborales.

Aparte de mi interés personal, creo que este trabajo es muy importante hoy en día. Vivimos en un mundo altamente interconectado, donde incontables dispositivos usan internet sin parar: móviles, ordenadores, sensores, cámaras, etc. Esto genera un gran flujo de datos, y es fundamental entender qué pasa en la red, si hay problemas, si todo va bien o si hay algo raro. Acceder a esta información de forma clara y en tiempo real es valioso para los que gestionan redes, temas de seguridad o para los que están aprendiendo.

Un soporte fundamental del internet es el sistema DNS, es clave para navegar en la web, usar el correo, los servicios en la nube y muchas más cosas. Su función de traducir los nombres de dominio en direcciones IP no solo facilita el uso de Internet, sino que lo hace más seguro y fiable. El estudio del comportamiento del tráfico DNS y cómo evoluciona con el tiempo es útil para comprender mejor la actividad de una red. Por eso, crear una plataforma que ofrezca un análisis de tráfico de internet en tiempo real es útil porque ayuda a los que administran redes o están estudiando estos temas.

Entonces, este proyecto me permite aplicar mis conocimientos, seguir aprendiendo y desarrollar una solución rentable que podría usarse en diferentes ámbitos.

1.3. Estado del arte

En el ámbito del análisis de tráfico de internet, se observa la existencia de numerosas herramientas avanzadas disponibles tanto para entornos profesionales como académicos. Por ejemplo:

Wireshark [1] es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para análisis de datos y protocolos, y como una herramienta didáctica. Cuenta con todas las características estándar de un analizador de protocolos de forma únicamente hueca.

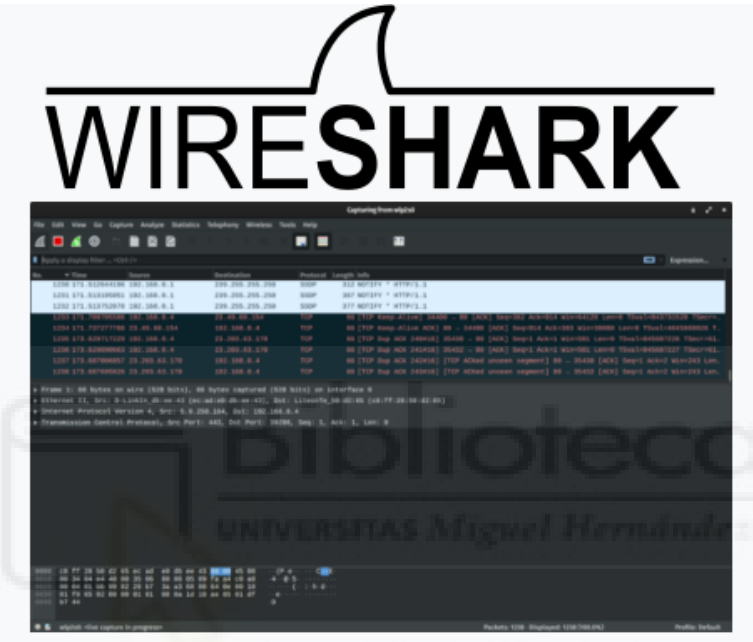

Figura 1: Interfaz de Wireshark durante la captura de paquetes.

Ntopng [2] es un excelente monitor de tráfico de red de nueva generación, es decir, es la versión actualizada de próxima generación del programa original conocido como Ntop, creado por la organización inglesa del mismo nombre. en realidad, lo que proporciona es una interfaz de usuario web intuitiva y encriptada para la exploración de información del tráfico de red en tiempo real y de forma histórica.

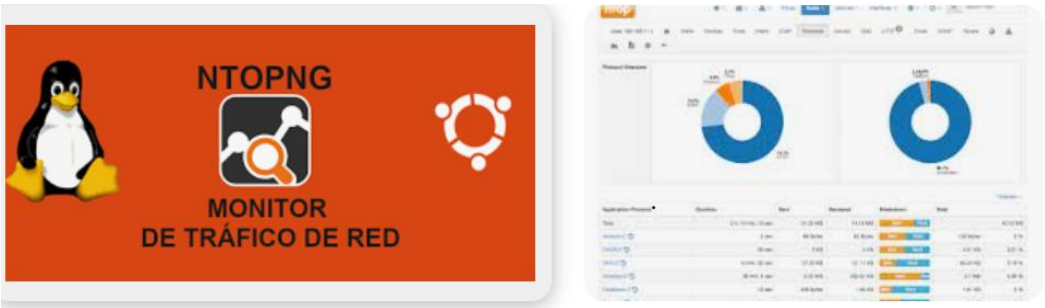

Figura 2:La herramienta Ntopng.

PRTG [3] es un software de supervisión de redes desarrollado por Paessler AG para supervisar y gestionar en tiempo real los dispositivos y la infraestructura de red. PRTG (Paessler Router Traffic Grapher) supervisa los servidores, el tráfico de red y diversos dispositivos de red para realizar un seguimiento del rendimiento de la red, incluido el uso del ancho de banda, el estado de los servidores y el rendimiento de las aplicaciones.

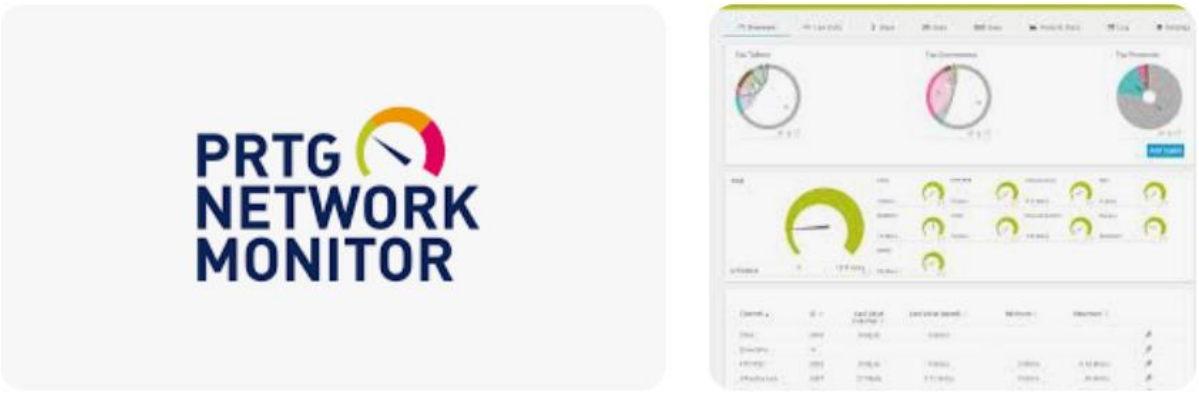

Figura 3:La herramienta PRTG.

Opciones como NTOPNG o el Monitor de red PRTG están dedicadas a mostrar el flujo de datos y proporcionan gráficos y estadísticas. Estas herramientas son esenciales, sin embargo, necesitan configuraciones técnicas complicadas o licencias de pago, lo que limita su uso en situaciones prácticas o en la educación.

En este trabajo, el objetivo principal fue crear una plataforma en línea que sea fácil de usar y accesible, y que mejore el seguimiento de datos en internet sin necesidad de programas complicados. A diferencia de otras opciones existentes, esta herramienta permite a los usuarios iniciar o parar la captura de datos directamente desde unos botones existentes en la herramienta, brindando mayor comodidad y control en su uso. Igualmente, la información recolectada se visualiza a través de diagramas dinámicos que se refrescan al instante lo que facilita la comprensión del rendimiento de internet de manera directa y fácil de entender. Esta plataforma no solo mejora la experiencia del usuario, sino que además concede a la herramienta un impacto significativo en entornos formativos y de entrenamiento, donde se pretende un método eficaz de instrucción sobre redes y tráfico DNS.

1.4. Objetivos

El objetivo principal de este trabajo final de grado es crear una herramienta para monitorizar el tráfico de internet en tiempo real, poniendo el foco en el protocolo de nombres de dominio (en adelante, DNS), mostrando los resultados a través de gráficos interactivos que faciliten su comprensión.

Las metas específicas de este trabajo incluyen:

- Diseñar una estructura modular que haga que el sistema sea fácil de mantener a lo largo del tiempo.

- Establecer el lado servidor que capture el tráfico de internet en tiempo real y lo guarde en una base de datos.

- Crear una interfaz web dinámica que posibilite comenzar, finalizar la captura y vaciar la base de datos para poder reiniciar una nueva directamente desde el navegador.

- Desarrollar un conjunto de gráficos interactivos que ayuden a visualizar diversos aspectos del tráfico de internet, con énfasis en el tráfico DNS.

- Posibilitar el análisis de intervalos de captura con el fin de comparar la cantidad de información solicitada en cada uno de ellos, permitiendo así observar variaciones en el tráfico a lo largo del tiempo.

- Facilitar el análisis de datos en algunas visualizaciones a través de filtros de direcciones IP de origen o de dominios DNS.

- Utilizar tecnologías como AJAX y lenguajes como HTML, CSS y JavaScript para asegurar que la experiencia del usuario ha sido fluida.

- Examinar el rendimiento del sistema con el objetivo de consideraciones para futuras mejoras.

2. MATERIAL Y MÉTODOS

2.1. Herramientas y tecnologías utilizadas

2.1.1. Herramienta Scapy

Scapy [4] es una herramienta de manipulación de paquetes para redes informáticas, escrita originalmente en Python por "Philippe Biondi". Puede falsificar o decodificar paquetes, enviarlos por cable, capturarlos y comparar solicitudes y respuestas. También puede gestionar tareas como escaneo, tracerouting, sondeo, pruebas unitarias, ataques y descubrimiento de redes.

Scapy proporciona una interfaz Python para libpcap o sockets nativos sin procesar, de forma similar a como Wireshark proporciona una interfaz gráfica de usuario (GUI) de visualización y captura. Se diferencia por su compatibilidad con inyección de paquetes, formatos de paquetes personalizados y scripts. Si bien es una herramienta exclusivamente de línea de comandos, puede interactuar con otros programas para proporcionar visualización, como Wireshark GnuPlot para generar gráficos, graphviz o VPython para visualización interactiva, etc.

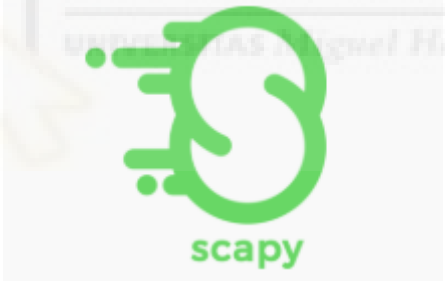

Figura 4: La herramienta SCAPY.

El uso de la herramienta Scapy en lugar de otras opciones se debe a que la plataforma necesita capturar tráfico de internet en tiempo real, lo que permite que los datos se inserten inmediatamente en la base de datos. Esto no se encuentra en la herramienta Wireshark, que puede capturar datos en cualquier momento, pero no ofrece una forma de insertar esos datos en la base de datos.

En este trabajo, se empleó Scapy para crear un script en Python que facilita la captura de paquetes del tráfico en tiempo real, permitiendo recoger información importante y guardarla de manera directa en una base de datos MYSQL.

2.1.2. Herramienta MySQL Workbench

MySQL [5] es un sistema de gestión de bases de datos (DBMS, por sus siglas en inglés) de código abierto desarrollado por Oracle. Se ha ganado su lugar en el mundo digital como una base de datos relacional que permite almacenar, organizar y recuperar datos de manera eficiente. MySQL es utilizado por una amplia variedad de organizaciones y aplicaciones en todo el mundo.

MySQL se utiliza en una gran variedad de ámbitos, desde sitios web y aplicaciones móviles hasta sistemas empresariales. Su **versatilidad** lo hace ideal para:

- **Almacenamiento de Datos**: MySQL almacena datos de manera eficiente, desde información de usuarios y productos hasta registros de transacciones.
- **Aplicaciones Web**: Es ampliamente utilizado para la creación de sitios web y aplicaciones online, ya que puede manejar grandes volúmenes de datos y solicitudes simultáneas.
- **Sistemas Empresariales**: Se utiliza en sistemas de gestión de recursos empresariales (ERP) y sistemas de gestión de relaciones con el cliente (CRM).
- **Análisis de Datos**: Puede ser parte de soluciones de análisis de datos, permitiendo consultas complejas y análisis en tiempo real.

MySQL utiliza un modelo de base de datos relacional, donde los datos se organizan en tablas con relaciones definidas. Utiliza el lenguaje SQL para realizar consultas y manipular datos. Sus componentes principales incluyen un servidor de base de datos, motores de almacenamiento y clientes que permiten la interacción con la base de datos.

El funcionamiento de MySQL se basa en almacenar los datos en el sistema de archivos del servidor. Cuando un programa del ordenador necesita acceder a los datos, envía una consulta SQL al servidor MySQL. El servidor MySQL procesa esta consulta y devuelve los resultados al programa.

Al igual que PostgreSQL, MySQL funciona sobre servidores Linux. Sin embargo, al tratarse de servidores de datos accesibles mediante TCP/IP que residen en máquinas dedicadas, pueden utilizarse indistintamente desde clientes Linux o Windows, por lo que podrá hacer uso de dichas bases de datos desde productos de alojamiento bajo cualquiera de los dos sistemas operativos.

- Desde aplicaciones en Perl o PHP se puede acceder a las bases de datos utilizando los controladores adecuados que están incluidos en nuestras instalaciones de dichos sistemas de scripting.
- Desde aplicaciones ASP puede crearse un DSN que apunte a la base de datos.

Características y ventajas de MySQL

MySQL tiene una serie de características que lo hacen un DBMS popular, entre las que se incluyen:

- **Es gratuito y de código abierto**: MySQL es un software de código abierto, lo que significa que su código fuente está disponible para que cualquiera lo vea y lo modifique. Esto hace que MySQL sea más accesible y flexible que los sistemas de bases de datos comerciales.
- **Escalabilidad**: MySQL se puede escalar para adaptarse a las necesidades de diferentes aplicaciones. MySQL puede manejar desde pequeñas aplicaciones hasta grandes sistemas empresariales.
- **Soporte multiplataforma**: Funciona en diversas plataformas, incluidos sistemas Windows, Linux y macOS.
- **Seguridad**: Proporciona opciones de autenticación y cifrado de datos para garantizar la seguridad.
- **Compatibilidad**: MySQL es compatible con numerosos lenguajes de programación y herramientas de desarrollo.
- **Rendimiento**: Ofrece una recuperación de datos rápida y eficiente, ideal para aplicaciones de alto rendimiento.
- **Flexibilidad**: Puede ser personalizado y configurado para adaptarse a las necesidades específicas de una aplicación.

Figura 5:La herramienta MYSQL.

La elección de trabajar con esta herramienta en lugar de otras opciones como PostgreSQL se debe a que tengo experiencia en MySQL y lo he utilizado en diversas materias, lo que representa una ventaja para mi trabajo.

En este trabajo, se utilizó MySQL Workbench para almacenar la información importante de los paquetes que se capturaron con el script de Python de Scapy en tablas, lo que facilita la relación entre ellos y la realización de consultas para obtener datos significativos de los paquetes.

2.1.3 Herramienta Microsoft Visual Studio 2022

Microsoft Visual Studio [6] es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Fortran, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades en línea bajo Windows Azure en forma del editor Mónaco.

Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno compatible con la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y videoconsolas, entre otros.

Figura 6:La herramienta Microsoft Visual Studio.

La elección de la herramienta Microsoft Visual Studio en lugar de otras alternativas porque es un entorno que facilita el trabajo con múltiples lenguajes y tecnologías en un solo sitio.

La utilización de esta herramienta en este trabajo es necesaria para implementar los códigos PHP, HTML, CSS y Javascript para formar una interfaz web que presenta las informaciones al usuario.

2.1.4. Herramienta XAMPP

XAMPP [7] es un servidor web local multiplataforma que permite la creación y prueba de páginas web u otros elementos de programación. Es un entorno de desarrollo popular diseñado para ser fácil de instalar y usar. Se utiliza principalmente como herramienta de desarrollo local para diseñadores y programadores.

Figura 7: La herramienta XAMPP.

Es una distribución de Apache que incluye softwares libres. El nombre es un acrónimo compuesto por las iniciales de los programas que lo constituyen:

1. Linux: Sistema operativo donde se instalará la aplicación. A diferencia de otros sistemas operativos como Windows, Linux es una distribución de software libre que ofrece seguridad, no requiere el pago de licencias y exhibe un rendimiento destacado.
2. Apache: El servidor web de código abierto es una aplicación ampliamente utilizada para la entrega de contenidos web. Las aplicaciones del servidor son proporcionadas como software libre por Apache Software Fundación.
3. PHP: Es un lenguaje de programación de servidor que permite crear páginas web o aplicaciones dinámicas. Es independiente de plataforma y es compatible con varios sistemas de bases de datos.
4. MySQL/MariaDB: Dicha aplicación cuenta con uno de los sistemas de gestión de bases de datos relacionales más destacados del mundo. Si se combina con el servidor web Apache y el lenguaje de programación PHP, se utiliza para el almacenamiento de datos en servicios web.
5. Perl: Dicho lenguaje de programación se utiliza en desarrollos web, administración de sistemas y programación de red, del mismo modo que posibilita la creación de aplicaciones web dinámicas.

El panel de control de XAMPP es una herramienta fundamental para la gestión de los diferentes módulos que componen este entorno de desarrollo. Permite controlar el servidor web, sistemas de gestión de bases de datos, así como otros componentes esenciales para el desarrollo de sitios web y aplicaciones.

Además, otras de sus funciones se basan en monitorizar el estado de los servicios activos y los puertos utilizados.

Para acceder al panel de control, simplemente abre el navegador web e introduce la dirección local correspondiente. En este, podrás visualizar y gestionar todos los módulos y herramientas disponibles en XAMPP:

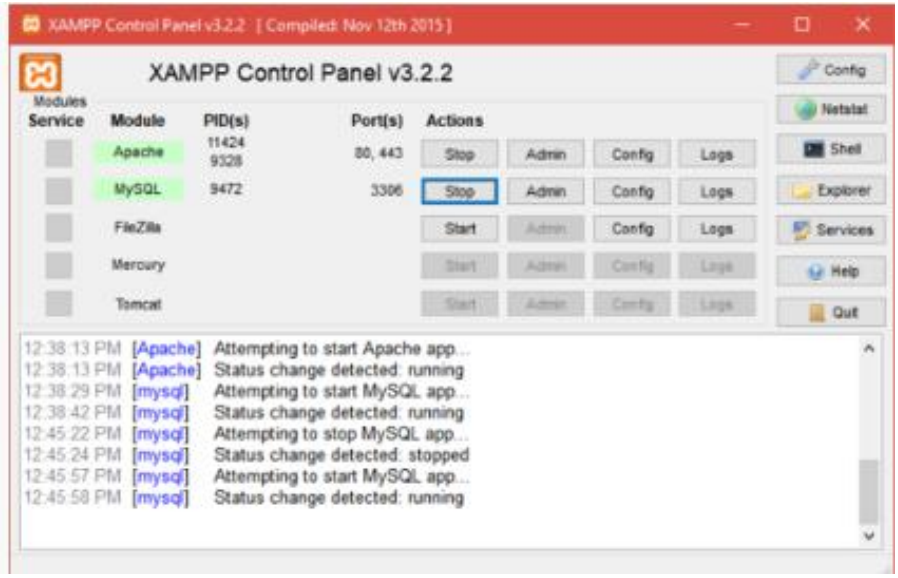

Figura 8:Panel De Control De XAMPP.

Seleccionar XAMPP como herramienta de desarrollo es principalmente porque es muy simple de instalar y configurar, lo que lo hace perfecto para trabajar en proyectos. XAMPP proporciona todo lo necesario para que las aplicaciones web con PHP y MySQL funcionen sin problemas.

La ventaja de XAMPP frente a otras alternativas como WAMP se puede funcionar en cualquier sistema operativo, además el panel de control hace que gestionar Apache y MySQL sea más sencillo, lo que acelera considerablemente el proceso de desarrollo y prueba.

2.2. Lenguajes de programación

2.2.1. Lenguaje Python

Python [8] es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

Figura 9: Lenguaje Python.

Los beneficios de Python incluyen los siguientes:

- Los desarrolladores pueden leer y comprender fácilmente los programas de Python debido a su sintaxis básica similar a la del inglés.
- Python permite que los desarrolladores sean más productivos, ya que pueden escribir un programa de Python con menos líneas de código en comparación con muchos otros lenguajes.
- Python cuenta con una gran biblioteca estándar que contiene códigos reutilizables para casi cualquier tarea. De esta manera, los desarrolladores no tienen que escribir el código desde cero.
- Los desarrolladores pueden utilizar Python fácilmente con otros lenguajes de programación conocidos, como Java, C y C++.
- La comunidad activa de Python incluye millones de desarrolladores alrededor del mundo que prestan su apoyo. Si se presenta un problema, puede obtener soporte rápido de la comunidad.
- Hay muchos recursos útiles disponibles en Internet si desea aprender Python. Por ejemplo, puede encontrar con facilidad videos, tutoriales, documentación y guías para desarrolladores.
- Python se puede trasladar a través de diferentes sistemas operativos de computadora, como Windows, macOS, Linux y Unix.

En este trabajo, el uso del lenguaje Python es debido a la necesidad de la librería Scapy para capturar y analizar los paquetes de internet en tiempo real.

2.2.2. Lenguaje PHP

PHP [9] es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario.

Entre los factores que hicieron que PHP se volviera tan popular, se destaca el hecho de que es de código abierto.

Esto significa que cualquiera puede hacer cambios en su estructura. En la práctica, esto representa dos cosas importantes:

- es de código abierto, no hay restricciones de uso vinculadas a los derechos. El usuario puede usar PHP para programar en cualquier proyecto y comercializarlo sin problemas.
- está en constante perfeccionamiento, gracias a una comunidad de desarrolladores proactiva y comprometida.

El PHP generalmente es definido como un lenguaje del lado del servidor. Esto significa que se aplica en la programación que tiene lugar en el servidor web responsable de ejecutar la aplicación o, más a menudo, en un sitio web.

Este trabajo previo permite cargar los elementos de una página antes de mostrarlos al usuario que accede a un sitio web, por ejemplo.

El código PHP se ejecuta en el servidor que, al leer los comandos, puede activar todos los elementos funcionales y la interfaz visual del sitio web.

Quizás, la aplicación principal del lenguaje PHP, cuando hablamos de la web, es estructurar sitios web en WordPress.

La simplicidad para aprender a usarlo y el desarrollo del código abierto le facilita el trabajo a los profesionales que eligen estructurar sitios web utilizando la plataforma, pues a medida que avanzan las configuraciones y ediciones se simplifican aún más.

Figura 10: Lenguaje PHP.

En este trabajo, se elige el lenguaje de programación PHP en lugar de otras opciones en el servidor ya que es adecuado para servidores web como Apache. Igualmente, es fácil establecer conexiones con bases de datos MYSQL. Además, se revisan las bases de datos con los paquetes reunidos y se manejan los datos que más tarde se enviarán al lado cliente para ser presentados en gráficos.

2.2.3. Lenguaje HTML

HTML [10] es el lenguaje con el que se define el contenido de las páginas web. Corresponde a las siglas en inglés de *Lenguaje de Marcado de Hipertexto*, básicamente son un conjunto de etiquetas que el navegador interpreta y se emplean para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, tablas, vídeos, etc.

El lenguaje HTML sirve para describir la estructura básica de una página y organizar la forma en que se mostrará su contenido, además de que HTML permite incluir enlaces hacia otras páginas o documentos.

Hay que mencionar que el HTML no es un lenguaje de programación, ya que no cuenta con funciones aritméticas, variables o estructuras de control propias de estos lenguajes, por lo que el HTML únicamente sirve para crear páginas web estáticas. Sin embargo, este lenguaje es muy útil ya que combinado con otros lenguajes de programación obtenemos páginas web dinámicas como las que conocemos hoy en día.

Por ejemplo, con HTML podemos:

- Crear párrafos.
- Insertar imágenes.
- Crear listas y tablas.
- Añadir enlaces a otras páginas.

Figura 11: Lenguaje HTML.

Para diseñar la interfaz de la plataforma, se eligió HTML (Lenguaje de Marcado de Hipertexto) como el lenguaje base, ya que es el estándar principal para estructurar contenido en internet. HTML permite crear la organización de las páginas de manera clara y garantiza que funcionen en cualquier navegador.

Aunque existen otras alternativas y framework para desarrollar interfaces en la web, como React, Angular o aplicaciones de escritorio, HTML continúa siendo la base sobre la que se construyen todas las interfaces, lo que lo convierte en algo esencial. En este trabajo, HTML fue utilizado junto a CSS y JavaScript para construir las páginas que presentan los datos obtenidos, incluyendo tablas y gráficos interactivos.

2.2.4 Lenguaje JavaScript

JavaScript [11] es un lenguaje de programación que los desarrolladores utilizan para hacer páginas web interactivas. Desde actualizar fuentes de redes sociales a mostrar animaciones y mapas interactivos, las funciones de JavaScript pueden mejorar la experiencia del usuario de un sitio web. Como lenguaje de scripting del lado del servidor, se trata de una de las principales tecnologías de la World Wide Web. Por ejemplo, al navegar por Internet, en cualquier momento en el que vea un carrusel de imágenes, un menú desplegable "click-to-show" (clic para mostrar), o cambien de manera dinámica los elementos de color en una página web, estará viendo los efectos de JavaScript.

Anteriormente, las páginas web eran estáticas, similares a las páginas de un libro. Una página estática mostraba principalmente información en un diseño fijo y no todo aquello que esperamos de un sitio web moderno. JavaScript surgió como una tecnología del lado del navegador para hacer que las aplicaciones web fueran más dinámicas. Por medio de JavaScript, los navegadores eran capaces de responder a la interacción de los usuarios y cambiar la distribución del contenido en la página web.

A medida que el lenguaje evolucionó, los desarrolladores de JavaScript establecieron bibliotecas, marcos y prácticas de programación y comenzaron a utilizarlo fuera de los navegadores web. En la actualidad, puede utilizar JavaScript para el desarrollo tanto del lado del cliente como del lado del servidor.

Figura 12:Lenguaje JavaScript.

El empleo de JavaScript en la creación del sitio web se debe a que puede funcionar en el navegador sin requerir instalaciones extras, ya que todos los navegadores actuales lo admiten sin problemas. Además, cuenta con una gran comunidad, numerosas bibliotecas y mucha documentación disponible. Por otro lado, este lenguaje se combina a la perfección con HTML, lo que posibilita el desarrollo de interfaces completas.

2.3 Lenguaje de estilo

2.3.1 Lenguaje CSS

CSS [12] significan «Hojas de estilo en cascada» y parten de un concepto simple pero muy potente: aplicar estilos (colores, formas, márgenes, etc...) a uno o varios documentos (generalmente documentos HTML, páginas webs) de forma automática y masiva.

Se le denomina estilos en cascada porque se lee, procesa y aplica el código desde arriba hacia abajo (siguiendo patrones como herencia o cascada que trataremos más adelante) y en el caso de existir ambigüedad (código que se contradice), se siguen una serie de normas para resolver dicha ambigüedad.

La idea de CSS es la de utilizar el concepto de separación de presentación y contenido. Este concepto se basa en que, como programadores, lo ideal es separar claramente el código que escribimos.

La idea es la siguiente:

- Los documentos HTML (contenido) incluirán sólo información y datos, todo lo relativo a la información a transmitir.
- Los documentos CSS (presentación) inclurán sólo los aspectos relacionados con el estilo (diseño, colores, formas, etc...).

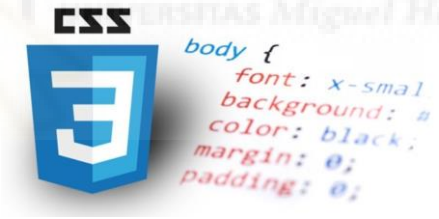

Figura 13:Leguaje CSS.

De esta forma, se puede unificar todo lo relativo al diseño, a lo visual en un solo documento CSS, y con ello, varias ventajas:

- Si necesitamos hacer modificaciones visuales, lo haremos en un sólo lugar y se aplica a todo el sitio.
- Se reduce la duplicación de estilos en diferentes lugares. Es más fácil de organizar y hacer cambios.
- La información a transmitir es considerablemente menor (las páginas se descargan más rápido).
- Es más fácil crear versiones diferentes para otros dispositivos: tablets, smartphones, etc...

Entones, la elección de lenguaje CSS porque es el lenguaje oficial, eficiente y universal para diseñar la interfaz visual de una web. No hay alternativa que lo sustituya completamente en el entorno web.

3. DISEÑO Y ARQUITECTURA DE LA HERRAMIENTA

3.1 Descripción general de la herramienta

El propósito de este trabajo es crear una plataforma que haga posible obtener, procesar y mostrar el flujo de internet al instante. La idea principal es elaborar un sistema que permita observar y analizar el tráfico de internet recibido y enviado por una computadora, abarcando la opción de capturar y guardar los paquetes de datos que se mandan en una base de datos.

Para realizar esto, se implementará un método de recolección de paquetes que usará herramientas y tecnologías de punta para asegurar que se junten datos que sean importantes y sirvan. Una vez que se recogen los paquetes, se dará paso a su manejo, lo que supone estudiar y ordenar la información en una base de datos estructurada. Así, hacer más fácil su búsqueda y estudio después.

Lo más interesante de este trabajo es la creación de una herramienta interactiva que dejará que los usuarios observen los datos del tráfico de internet de forma fácil y activa. Con esta plataforma, los usuarios podrán evaluar dibujos, números y formas del tráfico al momento lo que proporcionará una visión más clara del comportamiento del tráfico de internet en el dispositivo analizado. Aparte, se podrán añadir filtros para que los usuarios puedan capturar datos concretos de forma rápida y sin problemas.

El sistema viene con tres partes funcionales que se comunican entre sí de forma ordenada:

Primero, está el módulo que captura el tráfico de internet, un programa hecho en Python, usando la librería Scapy. Esto permite capturar paquetes directo de la interfaz Wi-Fi del sistema. El script está hecho para obtener paquetes, sacar información como las direcciones IPs de origen y destino, el país GeoIP, el tipo de protocolo y, si es un paquete DNS, el tipo de registro DNS, el tipo de mensaje y el tiempo DNS. Luego, guarda toda esta información automáticamente en una base de datos MYSQL.

Después, viene el módulo para guardar y tratar los datos, donde se usa una base de datos relacional MYSQL. Esto ayuda a poner los datos en una tabla bien organizada. Cada paquete capturado, crea un registro nuevo en la tabla. Este módulo asegura que los datos estén siempre ahí, para que se puedan hacer análisis y consultas sin problemas.

Finalmente, está el módulo que muestra y analiza todo en la interfaz web. La interfaz de usuario es una página web hecha con PHP, JavaScript, HTML y CSS. Primero, el usuario inicia la sesión luego entra directamente al Index (la página principal) de la plataforma, donde se observe todos los gráficos y las tablas en forma de bloques. PHP es como el traductor entre la base de datos y la interfaz, haciendo las consultas SQL necesarias y dando los resultados listos para mostrar en la aplicación web. JavaScript es la interfaz del usuario que regula la interacción dinámica en el explorador.HTML es la estructura básica del sitio web que establece la organización y la información. CSS es el estilista que da formato y estilo visual.

Entonces, esta solución se basa en tecnologías conocidas y compatibles, lo que nos permite preparar un diseño modular con diferentes responsabilidades.

3.2 Arquitectura del sistema

La plataforma se diseñó siguiendo una arquitectura modular, casi como si fueran niveles, donde cada parte tiene un papel bien definido en todo el proceso. Gracias a esta forma de organizarlo, es fácil distinguir entre la recogida de datos, dónde se guardan, cómo se procesan y cómo se muestran; esto hace que sea más sencillo mantenerlo.

El sistema está organizado así, por partes que cumplen diferentes funciones:

- Captura de tráfico

En esta parte tenemos un script hecho en Python, que usa la biblioteca Scapy, y se encarga de capturar el tráfico de internet en tiempo real, desde una interfaz Wi-Fi. Este script deja pasar todos los tipos de paquetes, saca la información que le interesa y la convierte a un formato que la base de datos pueda entender.

- Almacenamiento de datos

Toda la información que se recoge se guarda en una base de datos MySQL, organizada en la tabla que tiene la información importante de cada paquete (como direcciones IPs de origen y destino, tipo de protocolo, etc.). Esta base de datos es como el almacén principal de la información.

- Servidor web

En esta parte, usamos PHP como lenguaje de servidor, que se encarga de realizar consultas a la base de datos y preparar los datos para enviarlos a la parte que se observe. PHP también nos permite poner filtros de tiempo, separar los resultados y crear respuestas que cambian según lo que haga el usuario.

- Interfaz de presentación de resultados

Esta parte está hecha con HTML, CSS y JavaScript, y es lo que el usuario observe: los resultados de la captura en forma de gráficos y tablas que se pueden usar. JavaScript facilita la creación de interfaces interactivas, tales como filtros y gráficos que se actualizan de manera automática de acuerdo con la información suministrada por el servidor. Esta interfaz permite que los usuarios puedan navegar y entender la información que se ha recogido de forma sencilla.

3.3 Diagrama de arquitectura

Para explicar de manera sencilla la organización y operación de la plataforma creada, se incluye un diagrama de flujo que muestra los componentes clave del sistema y su relación entre ellos.

Este diagrama ilustra el movimiento de la información desde su recolección en tiempo real hasta su presentación en la interfaz en línea.

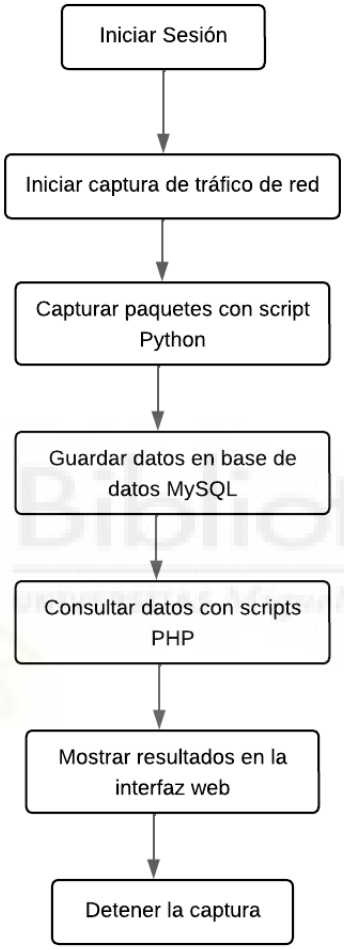

Figura 14:Diagrama De Flujo Del Sistema.

Descripción del diagrama:

- **Script Python con Scapy:** Se ejecuta a partir de una tarea programada para analizar el tráfico de internet en tiempo real, capturando los paquetes y extraer información importante para la aplicación web.
- **Base de datos MYSQL:** Obtiene de manera directa la información recolectada por el script de Python y la guarda en una tabla estructurada para consultas futuras.
- **Interfaz Web:** Formada por HTML, CSS y JavaScript, presenta la información al usuario final a través de gráficos y tablas interactivas. JavaScript facilita la aplicación de filtros y actualizaciones en tiempo real.

4. IMPLEMENTACIÓN DE LA HERRAMIENTA

4.1. Configuración inicial

Para realizar la puesta en marcha del sistema, se inició con la instalación y ajuste del entorno de desarrollo en un equipo con Windows.

Necesitamos instalar todas las independencias necesarias, la primera es Python desde este URL: https://www.python.org/downloads/

y lo añadimos al PATH del sistema:

Figura 15:Selección de la opción "Add Python to PATH" en el instalador".

Además, es importante instalar Npcap[13] desde este URL: https://npcap.com/#download que es la biblioteca para capturar y enviar paquetes del proyecto Nmap en Microsoft Windows. Es similar a libpcap, pero para Windows. También necesitamos hacerlo antes de instalar la biblioteca Scapy, ya que Scapy no puede captar tráfico sin la presencia de Npcap.

A continuación, para instalar la librería Scapy debemos abrir el CMD del sistema y ejecutar el comando:

```
pip install --pre scapy[complete]
```

Y para traer todos los módulos y funciones principales de Scapy en un script o en una consola interactiva de Python, ejecutamos el comando:

```
from scapy.all import *
```

De seguida, creamos una carpeta en el disco local del equipo llamada "WiresharkScripts" donde guardamos todos los scripts Python.

Luego, debemos instalar la herramienta Microsoft Visual Studio para crea los scripts PHP para consultar datos de la base de datos:

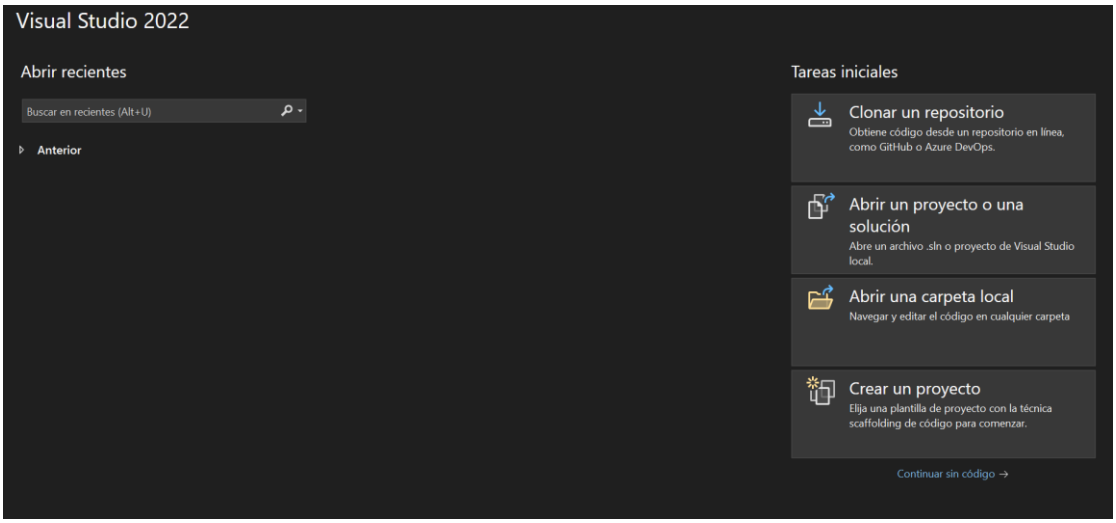

Figura 16:Entorno de desarrollo en Visual Studio 2022.

Seguidamente, procedemos a instalar MYSQL Workbench para establecer una base de datos en la que crearemos las tablas necesarias para añadir la información importante de los paquetes que capturamos en tiempo real.

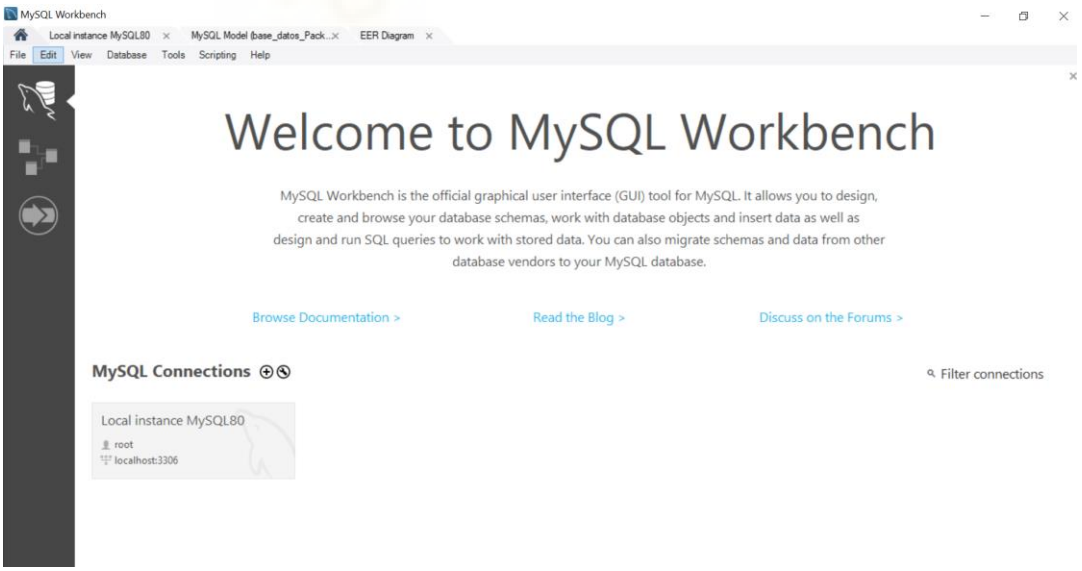

Figura 17:Interfaz de MySQL Workbench.

A continuación, tenemos que haber el servidor XAMPP configurado para nos permite la creación de nuestra aplicación Web, luego, pulsamos sobre “Start” del módulo Apache.

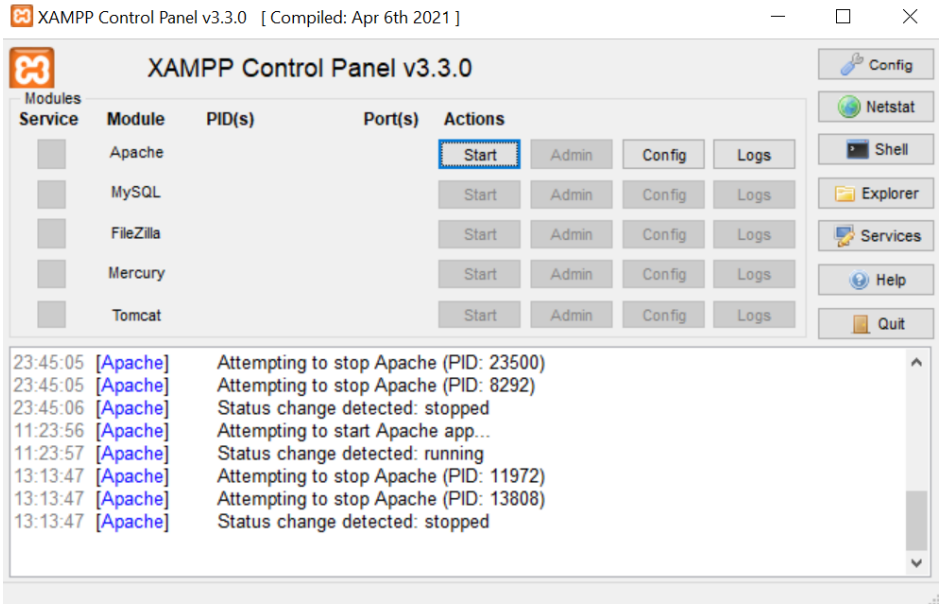

Figura 18:Servidor local XAMPP.

4.2 Desarrollo del script de captura

El objetivo del script Python es capturar tráfico de internet en tiempo real usando la librería Scapy, analizando de forma especial el tráfico DNS y procesar la información esencial y almacenarla en una base de datos MYSQL.

A continuación, explicamos los partes calves del script Python para entender el funcionamiento de este código:

Estructura general

La implementación del archivo Python se basa de una clase principal **PacketCapture** que actúa como el núcleo de toda la lógica de captura, primero se toma un identificador de usuario **user_id** para poder identificar la sesión de manera única, después se ajusta la configuración del registro, la conexión a MySQL [14], y los contadores necesarios; para terminar, se pone en marcha un diccionario llamado **dns_queries** para seguir de cerca las consultas DNS que están pendientes.

Al inicio, el script configura los directorios necesarios para logs, estado y caché, además de establecer variables de entorno clave para Scapy [15] que optimizan su funcionamiento. Estas configuraciones previas son esenciales para evitar errores comunes y garantizar que la captura de paquetes se realice sin problemas.

Configuración y recursos

En la etapa de inicio, se conecta a la base de datos, se carga el lector de la ubicación geográfica GeoIP [16] para obtener datos sobre el país de los paquetes y se organiza el sistema de registro, tanto para la consola como para el archivo. Asimismo, se crea un formato para anotar las solicitudes DNS y medir el tiempo entre una solicitud y su respuesta correspondiente, utilizando identificadores que combinan el nombre de dominio y el identificador de la transacción DNS.

Captura de paquetes

Para capturar el tráfico de red, es necesario establecer la función **sniff ()** de Scapy que sirve para capturar paquetes de internet en tiempo real desde una interfaz (Wi-Fi), también trabajar cada paquete con el método **process_packet** que captura los datos como:

- Número de paquete.
- Tiempo de captura.
- IP de origen y destino.
- Puerto de origen y destino.
- Protocolo detectado (DNS, HTTP, HTTPS, TCP, UDP, ICMP).
- Información específica de DNS: tipo de mensaje, tipo de registro, tiempo de respuesta, nombre consultado o respondido.
- País de destino (utilizando GeoIP).
- Identificador de usuario.

Almacenamiento en MYSQL

Para almacenar los datos en la base de datos, se utiliza el método **insert_packet** que se ocupa de almacenar de manera eficaz los datos de los paquetes dentro de una base de datos MySQL. Primero, establece una conexión utilizando la configuración predeterminada y lleva a cabo una consulta **INSERT** parametrizada para guardar toda la información del paquete (como direcciones IP, puertos, protocolos, datos de DNS, etc.) en la tabla **'datos_paquetes'**.

Mecanismo de parada

Para lograr que el script se detenga de manera controlada desde una interfaz externa, se establece un sistema que utiliza archivos [17]: al encontrar un archivo llamado stop_. txt, el script se detiene de manera segura. Asimismo, se guarda un registro del evento de finalización en la base de datos y se eliminan los archivos temporales asociados.

Consideraciones adicionales

La memoria caché DNS del sistema se borra al comenzar la captura, para garantizar que se registren nuevas solicitudes. Además, se establece un proceso en segundo plano que comprueba con regularidad si hay una señal para detenerse. También, se guarda información importante en archivos de registro dedicados a cada usuario, lo que permite seguimiento y corrección de errores.

Ejemplo de su ejecución:

Para visualizar la ejecución del este script de captura, iniciamos la captura pulsando sobre "Iniciar captura", se abre una ventana de consola que muestra mensajes de registro informativos sobre el estado del sistema. Los mensajes de consola confirman que se ha cargado la base de datos GeoIP, se ha iniciado la captura para el usuario activo, se ha limpiado la caché DNS del sistema, y que la captura comienza correctamente en la interfaz Wi-Fi.

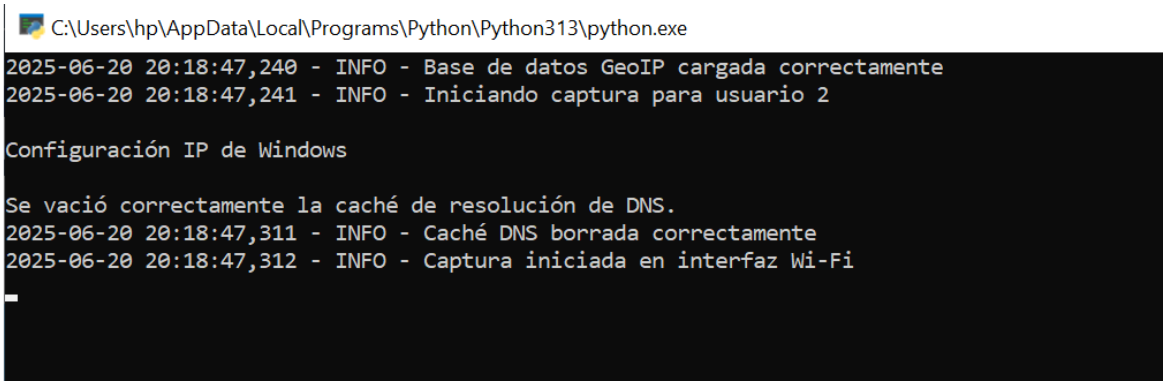

Figura 19:Ejecución de script de captura.

4.3. Desarrollo del script de peticiones

Este script Python permite realizar consultas repetidas a diferentes dominios, usando distintos tipos de registros DNS. Además, nos ayuda a minimizar las intervenciones manuales permitiendo hacer las consultas sin tener que hacerlas muchas veces a través de un navegador.

Seguidamente, explicamos los puntos clave de este script para entender su funcionamiento.

Primero, Instalamos **dnspython** [18] que es una biblioteca de Python diseñada para facilitar el trabajo con DNS en programas y aplicaciones escritos en Python. Este módulo permite a los desarrolladores realizar consultas DNS, analizar registros DNS, y gestionar operaciones relacionadas con DNS de manera sencilla y eficiente, sin tener que lidiar con los detalles técnicos de la implementación del protocolo DNS.

La instalación se realiza a través de comando siguiente:

```
C:\Windows\system32>pip install dnspython
Collecting dnspython
Downloading dnspython-2.7.0-py3-none-any.whl.metadata (5.8 kB)
Downloading dnspython-2.7.0-py3-none-any.whl (313 kB)
Installing collected packages: dnspython
Successfully installed dnspython-2.7.0
[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

A continuación, implementamos el código Python que tiene como objetivo generar tráfico DNS real que pudiera ser capturado y analizado por el script principal de monitorización. La generación de tráfico DNS se realiza a partir de ejecutar consultas DNS repetidas a una lista de dominios.

Descripción del funcionamiento

El script se fundamenta en una lista definida previamente de dominios (como google.com, vodafone.es, wikipedia.org, entre otros) y diferentes tipos de consultas DNS (A, AAAA, MX, SOA, PTR. . .). Para cada par de dominio y tipo, se realiza una consulta DNS utilizando la biblioteca dnspython, que es muy común en redes. Estas consultas se llevan a cabo en múltiples ciclos (por ejemplo, 200 veces) con un intervalo de pausa entre cada una. También se ha establecido un sistema para manejar errores que ayuda a identificar si un dominio no está disponible, si no hay respuesta para un tipo específico de registro o si se supera el tiempo de espera, entre otras situaciones.

Resultado

Con la ayuda de este script, conseguimos insertar tráfico DNS que pudimos seguir y reproducir en el sistema. De esta manera, verificamos, por ejemplo, que el analizador registraba adecuadamente las consultas, que estas se almacenaban en la base de datos, y que la interfaz mostraba y organizaba correctamente los diferentes tipos de registros DNS.

Ejemplo de su ejecución:

Proporcionamos un ejemplo de ejecución del script de consultas en el símbolo del sistema para observar cómo se realizan estas consultas, para ejecutar este script usamos el comando: Python dns_queries.py (dns_queries es el nombre del script).

Notamos que el script realiza correctamente diversas consultas DNS (A, AAAA, MX, PTR, SOA) a múltiples dominios, obteniendo en su mayoría respuestas válidas. No obstante, hay ciertas consultas que no obtienen respuestas, lo que podría ser resultado de limitaciones impuestas por el servidor DNS, ausencia de registros del tipo solicitado o dificultades en la conectividad.

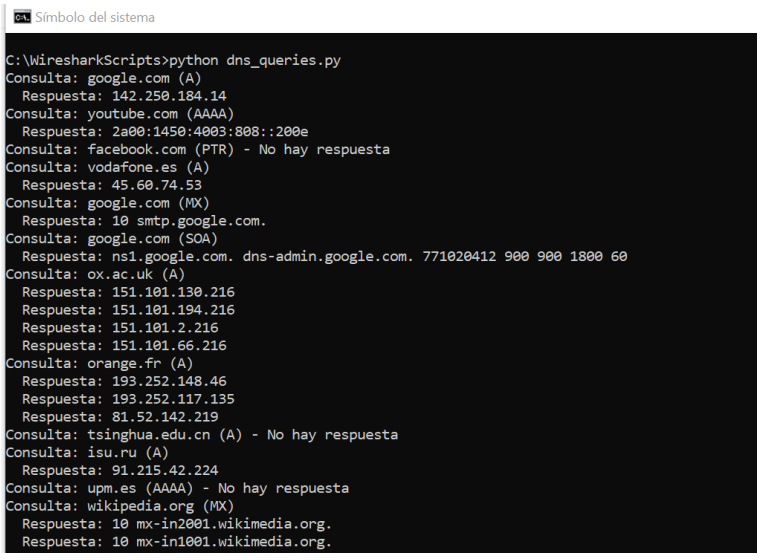

Figura 20:Ejecución de script de consultas.

4.4. Diseño de la base de datos

Para lograr un almacenamiento organizado de la información importante de los paquetes que el script de Python, utilizando la biblioteca Scapy, recoge durante el análisis del tráfico de internet, es necesario crear una base de datos relacional. Esta base de datos es fundamental para el proyecto y debe adaptarse a las necesidades del sistema, lo que permitirá luego realizar consultas, análisis y presentación a través de la plataforma web.

En esta sección, se explican los componentes de la base de datos llamada **db_paquetes**, proporcionando información sobre las tablas que se utilizan y cómo están relacionadas entre sí, así como los tipos de datos que se guardan, incluyendo direcciones IP, puertos tanto de origen como de destino, tipos de registros DNS y clases de mensajes DNS, entre otros.

1.Diseño de las tablas:

Luego de establecer la base de datos **"db_paquetes",** se generan las tablas. Dentro de estas tablas, existe la tabla **usuario** se encargará del proceso de inicio de sesión, ya que contiene la información del usuario. La tabla **datos_paquetes** será utilizada para almacenar los datos importantes de los paquetes capturados en tiempo real.

La conexión entre ambas tablas es de tipo uno a muchos, lo que significa que cada usuario cuenta con múltiples capturas y cada captura pertenece solo a un usuario.

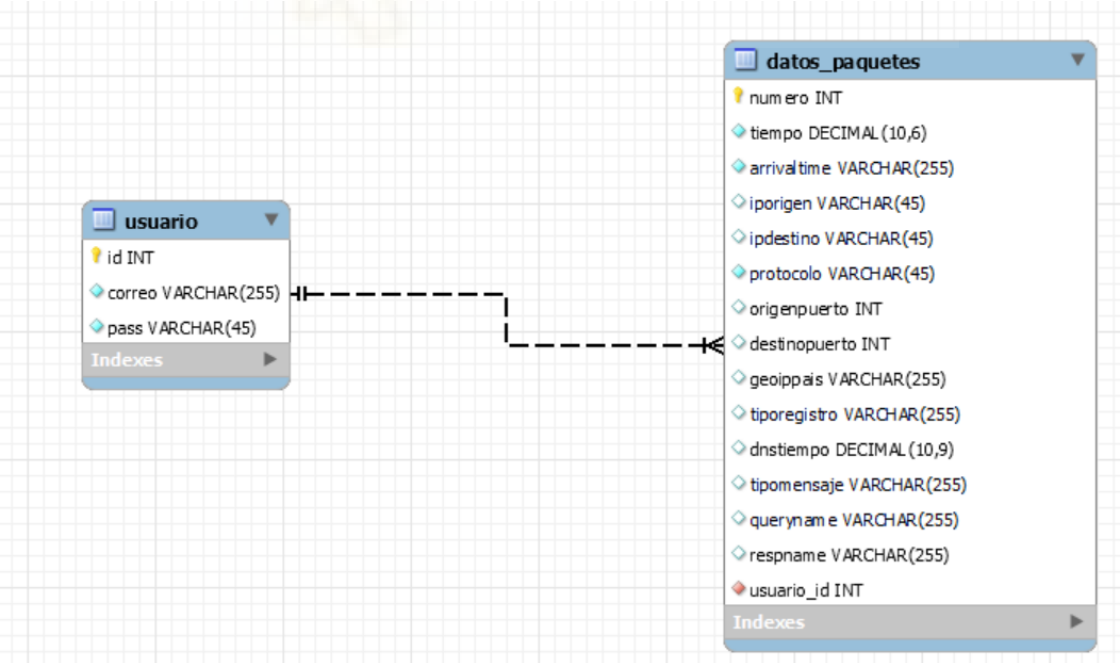

Figura 21: Vista parcial de la base de datos.

Descripción de las tablas:

Tabla: usuario

- **id:** Clave principal de la tabla.
- **correo:** Dirección de correo electrónico del usuario.
- **pass:** Contraseña del usuario.

Tabla: datos_paquetes

- **numero:** Clave principal de la tabla y es un valor entero. Es el identificador único de cada paquete.
- **tiempo:** Tiempo en segundos desde el inicio de la captura.
- **iporigen:** Dirección IP de origen.
- **ipdestino:** Dirección IP de destino.
- **protocolo:** Nombre del protocolo de red utilizado.
- **origenpuerto:** Número de puerto de origen. Es un valor entero.
- **destinopuerto:** Número de puerto de destino. Es un valor entero.
- **geoippais:** País de destino (según la base de datos GeoIP).
- **tiporegistro:** Tipo de registro DNS.
- **dnstiempo:** Tiempo de resolución DNS.
- **tipomensaje:** Tipo de mensaje DNS (consulta o respuesta).
- **queryname:** Dominio consultado.
- **respname:** Dominio respuesta de una consulta DNS.
- **usuario_id:** Clave foránea que apunta al campo id de la tabla usuario.

4.5 Funcionamiento de la herramienta

4.5.1 Gestión de usuarios y autenticación

La primera implementación que existe en la herramienta es un sistema de gestión de usuarios y autenticación para controlar el acceso a la plataforma. Este sistema pide dos datos importantes que son el correo electrónico y la contraseña para identificar si es un administrador o un usuario normal porque cada uno tiene permisos de visualización y operación dentro de la aplicación.

Figura 22: Inicio De Sesión.

El manejo del proceso de autenticación de usuarios se realiza mediante un código PHP que verifica primero las credenciales enviadas por POST contra una base de datos MYSQL, luego, comprueba si el usuario ya tiene una sesión activa, en ese caso, se redirige el usuario, seguidamente, valida que los campos de correo electrónico y contraseña no están vacíos, después consulta la base de datos para verificar las credenciales. Si son correctas, crea variables de sesión con el identificador y correo del usuario y lo redirige a la página principal; de lo contrario, muestra un mensaje de error.

Para cerrar la sesión, implementamos un apartado en el script de la página principal que se ocupa de diseñar el botón cerrar sesión con HTML y CSS que cuando pulsamos sobre él se redirige al script php que contiene la función **"session_destroy();"**se encarga de eliminar completamente la sesión actual en el servidor, concluyendo la sesión del usuario. Es concluyente para manejar apropiadamente el cierre de sesión y asegurar la protección del sistema.

4.5.2 Estructura de la página principal

Después de validar los datos del inicio de sesión, se redirige el usuario automáticamente a la página principal que se maneja mediante un código HTML que contiene un fragmento Javascript que gestiona la autenticación del usuario mediante una petición asíncrona a un código php que verifica el estado de autenticación del usuario y devuelve una respuesta en formato JSON. Si la respuesta es no autenticada, se devuelve directamente a la página de inicio de sesión, si la repuesta es válida, crea un campo donde se muestra el nombre del usuario.

Además, incluí el diseño HTML de la barra de navegación donde está colocado el nombre de la aplicación, botones de iniciación y detención de captura, botón de limpieza la base de datos, campo que muestra el estado de la captura, el nombre de usuario y el botón de cerrar la sesión.

Asimismo, contiene un fragmento que permite diseñar un panel a la izquierda de la página donde parecen tres asuntos: el tiempo de captura que permite visualizar el tiempo de inicio y de fin de captura, la duración total de la captura en minutos, el rango disponible es decir el tiempo en segundos del primer paquete y el último paquete capturado, Filtro que permite observar solamente un periodo de tiempo de captura, en otras palabras unos determinados paquetes capturados, esto se realiza mediante un formulario que permite navegar entre paquetes .Y un menú desplegable que admite notar todos los direcciones IPs de origen que aparecen en la captura.

A continuación, el script HTML contiene el diseño de las tarjetas donde se van a colocar los gráficos y las tablas que representaran aspectos importantes sobre el tráfico de internet.

Finalmente, el código contiene un apartado para representar el pie de la página donde se informa el año de realización de la plataforma y que sus derechos son reservados.

Figura 23: Interfaz de la herramienta.

4.5.3 Control de captura de tráfico

- **Iniciar Captura**

Para facilitar al usuario la función de capturar el tráfico de internet, hemos diseñado un botón desarrollado con HTML y estilizado con CSS en la barra de navegación para al pulsar en el empieza directamente a capturar el tráfico. Es decir, que este botón está conectado con funciones JavaScript que envía peticiones AJAX a scripts PHP.
Entonces, para llevar a cabo esta funcionalidad, primero hemos configurado tareas programadas según el parámetro identificador de usuario.
La figura representa el nombre de la tarea marcando que se ejecuta solamente cuando el usuario haya iniciado la sesión también que se ejecuta con los privilegios más altos.

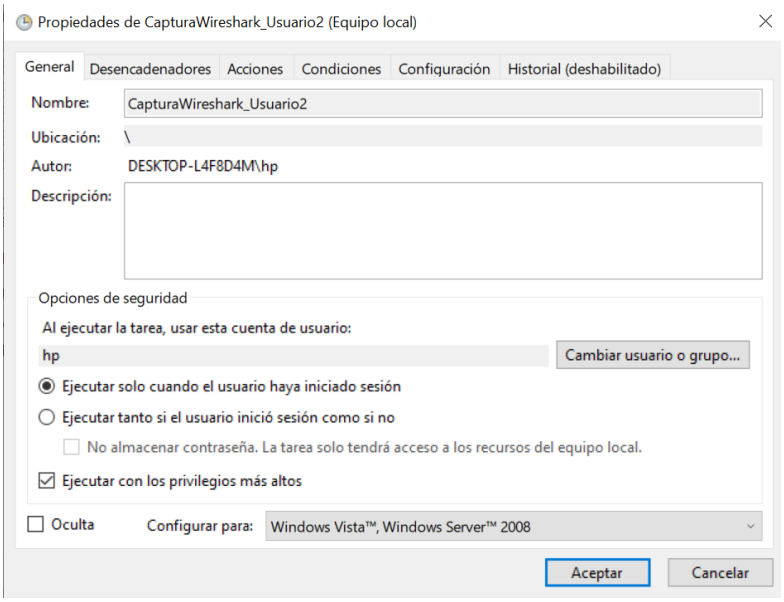

Figura 24: Propiedades de la tarea programada.

Luego, iremos a la sección Acciones donde establecimos una nueva tarea donde metemos la ubicación completa del sitio donde está instalado Python, además agregamos el argumento que sea el nombre del script Python junto a identificador de usuario en este caso el identificador de usuario es igual a 2 y lo iniciamos en la carpeta donde tenemos este script guardado.

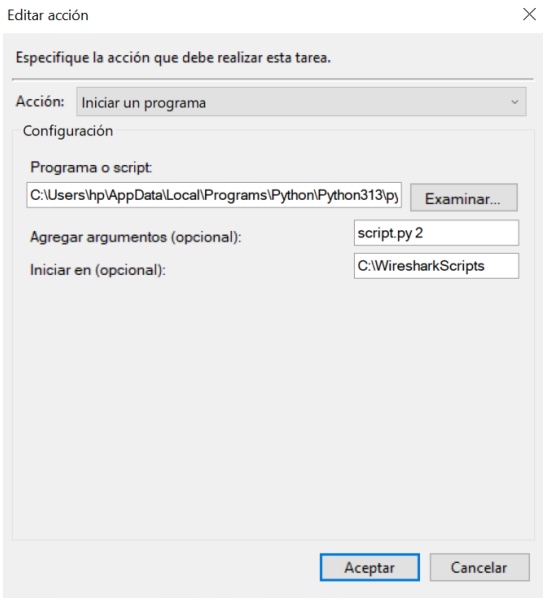

Figura 25: Acciones de la tarea programada.

Una vez listas las tareas programadas, nos dedicamos a establecer la comunicación entre la parte del usuario y el código Python usando PHP/AJAX.

El proceso arranca cuando el usuario accede a la plataforma y entra con sus credenciales. Luego de confirmar quién es, su identificador de usuario (user_id) queda unida a la sesión actual. Este aspecto es fundamental para monitorear las acciones realizadas por cada usuario.

En caso de que el usuario desee iniciar la captura de tráfico, se envía una solicitud al servidor utilizando una función AJAX. Esta función activa un archivo PHP que sirve de conexión entre la plataforma y el sistema operativo. Dicho archivo PHP se encarga de revisar el identificador de usuario user_id que recibe, para confirmar que la petición venga de alguien que sí está dentro del sistema. Después de esta revisión, el script PHP usa la orden del sistema **schtasks** [19] para activar la tarea programada, que pone en marcha el script Python que captura el tráfico de internet con la ayuda de Scapy.

Para evitar que las peticiones AJAX se guarden en la memoria y asegurar que cada llamada sea nueva, se usa un método que incluye indicaciones de tiempo. Así, el servidor siempre responde al momento, sin riesgo de usar datos antiguos.

La forma en que se organiza este intercambio asegura que cada momento de captura quede vinculado a un usuario en particular, lo cual facilita el manejo y el estudio del tráfico que genera cada sesión iniciada desde la plataforma.

- **Detener Captura**

  Para terminar el ciclo de control de captura de tráfico desde la interfaz web, usamos un método de detención remota que admite al usuario finalizar la sesión de forma controlada.

  La detención de captura se realiza mediante un botón implementado en HTML y estilizado con CSS, accesible desde la barra de navegación de la plataforma. Al pulsar este botón por parte de usuario, se ejecuta una función AJAX que envía solicitud al servidor. La respuesta de esta solicitud lo haga un archivo PHP que contiene la función responsable de generar un archivo de parada dentro de la carpeta de archivos de servidor que se trata de una señal de control para el script Python que está elaborando la captura de tráfico.

  El script Python contiene un fragmento que esta responsable para escuchar la presencia de este archivo de parada. A la hora de detectar este archivo, finaliza automáticamente la captura de paquetes y envía una solicitud AJAX que confirma que el archivo de parada se ha creado perfectamente actualizando dinámicamente la interfaz de web.

  Esta técnica presenta un dominio sencillo pero muy ventajoso que, para comprender todo el proceso de recopilación de datos, eliminando la necesidad de manejo manual de script.

- **Gestión de estado de captura**

  Para verificar si la grabación está en marcha o se ha pausado, se utiliza un fichero provisional. De esta manera, la interfaz puede señalar con precisión si una grabación está activa, impidiendo también que se lancen varias grabaciones a la vez.

  Esta gestión se realiza creando un archivo de estado que notifica que el proceso arrancó al empezar una captura. Al detenerse, se crea el archivo de detención que finaliza el script en Python, y se suprime el archivo de estado para comunicar que ninguna grabación está activa.

  Aparte, este trámite proporciona otras ventajas, como limitar nuevos intentos para dejar que la grabación actual termine, así como para sortear problemas en el sistema que puedan derivarse de la ejecución simultánea del script de grabación.

  En consecuencia, el empleo de ficheros temporales como indicadores de estado resulta ser una práctica eficiente en lugares donde se pretende mantener la sencillez y la separación de labores entre la lógica del servidor (PHP) y los procedimientos de grabación (Python).

4.5.4. Vaciado de la base de datos

La plataforma proporciona una función que es el vaciado de la base de datos para que se puede reiniciar el análisis de datos permitiendo a los usuarios autenticados eliminar sus propios registros de tráfico de internet.

Para controlar el vaciado de la base de datos, se ha implementado un control de acceso que permite comprobar que si existe una sesión activa y que el identificador del usuario que realiza la solicitud coincida con el que esta autenticado porque solamente los usuarios normales tienen el derecho de vaciar sus propios registros, en el caso de que el administrador intenta a vaciar los datos de algún usuario, el sistema bloquea directamente esta operación.

La eliminación de datos se realiza mediante una consulta que no elimina toda la base de datos sino únicamente las filas asociadas al usuario que ha iniciado la sesión. Tras el vaciado de estas filas, se actualizan automáticamente los gráficos mostrados en la interfaz web dando una experiencia coherente con el nuevo estado de la base de datos.

Con esta herramienta, el sistema garantiza que cada persona controle su información de forma privada y autónoma, sin interferir con la información de los demás ni comprometer la estructura del sistema. Esta habilidad refuerza los principios de privacidad, seguridad y gestión personal que se buscan ofrecer en el entorno de estudio del flujo de la red.

4.5.5 Menú de control de captura y filtrado

En la herramienta implementamos un panel de navegación llamado menú hamburguesa colocado en la parte superior izquierda, al ser activado muestra un panel lateral que incluye estas secciones: tiempo de captura, filtros y menú desplegables de direcciones IPs de origen capturadas. Este elemento está oculto por defecto para aprovechar al máximo el espacio de trabajo, apareciendo temporalmente sobre el contenido cuando el usuario necesita usar sus funciones.

- **Tiempo de captura**

  Para visualizar el tiempo de inicio y de fin de captura de paquetes de red asociada a la sesión del usuario de forma automática en el menú hamburguesa, debemos implementar en el servidor una consulta SQL para obtener los valores mínimo y máximo del campo **arrivaltime** dentro de la tabla **datos_paquetes** garantizando que los usuarios estándar solo observan sus propios tiempos de captura y el administrador obtenga el intervalo más amplio entre todos los usuarios disponibles.

Una vez realizada la consulta, los datos se devuelven al lado cliente en formato JSON y se emplea el código Javascript para cargar estos datos al iniciar la aplicación.

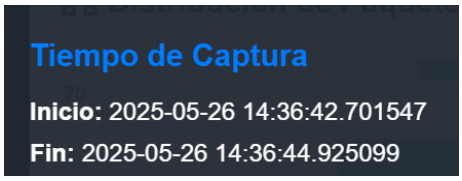

Figura 26:Visualización del tiempo de captura.

- **Filtrar la captura por periodos**

Para efectuar el filtro que permite analizar la captura por rangos disponibles en la captura, primero implementamos el formulario del filtro desarrollados en HTML y estilizado con CSS, luego crear un script PHP que contiene una consulta para obtener todos los valores únicos de tiempo registrados para el usuario en sesión. además, se utiliza **"session_start()"** para obtener el identificador de usuario user_id, y dependiendo de su valor, se restringe la búsqueda a un usuario_id o se extiende a varios usuarios (como en el caso del administrador). Esto ofrece mostrar al usuario solo los segundos en los que hay datos disponibles, mejorando la eficiencia del filtrado en el lado cliente.

A continuación, se creó un segundo script que calcula el rango total del tiempo de las capturas disponibles, utilizando métodos como **MIN(tiempo)** y **MAX(tiempo).** Esto permite visualizar al usuario el inicio, fin y duración total de captura en minutos.

Seguidamente, se implementa el código Javascript que solicita los tiempos disponibles y se encarga de precargarlos en los campos del filtro. Además, se solicita el rango general para mostrarlo al usuario. Esto admite al usuario entender los límites temporales disponibles y ajustar el análisis con precisión.

Figura 27: Visualización del filtro temporal.

- **Menú desplegable para los IPs de origen capturadas**

  La parte final del panel presenta un menú que se despliega y muestra todas las direcciones IP de origen que se identificaron durante la sesión. Para lograr esto: Se ejecuta una consulta en la base de datos para recoger las direcciones IP diferentes que han estado involucradas en el tráfico. Después, estos datos se envían al cliente, que crea de manera dinámica los elementos visuales del menú, lo que permite al usuario revisar de forma rápida las direcciones IPs encontradas.

Figura 28: Menú desplegable de IPs de origen.

4.6 Implementación técnica de los resultados visuales

- **Desarrollo técnico del gráfico: Distribución de Paquetes por Protocolo**

  Este gráfico muestra los protocolos que se han encontrado en el tráfico de internet que se ha registrado para cada usuario general, indicando cuántos paquetes hay para cada protocolo como TCP, UDP, DNS y otros, presentados en un formato de gráfico de barras.

  Los datos que se presentan en este gráfico se obtuvieron mediante un script PHP. Este script se encarga de atender las peticiones de los usuarios, volviendo la cantidad global de paquetes organizados según su protocolo. La información se puede mostrar en su totalidad o acotada a un periodo de tiempo especificado por el propio usuario. además, esta numeración de paquetes permite determinar si la captura sigue activa porque si el número de paquetes queda constante en varias consultas repetidas, se interpreta como que no hay actividad en la red.

  En la interfaz web donde se representa la información obtenida del servidor, encontramos el gráfico de barras que admite al usuario entender el tipo de tráfico predominante. Además, cuenta con un mecanismo de actualización automático; este se conecta al servidor cada cierto tiempo para conseguir la información más nueva. La herramienta también ofrece la opción de usar filtros de tiempo. Así, el usuario puede escoger un periodo en concreto y observar solo la información de ese momento. Esto facilita la atención en puntos concretos o en revisar áreas específicas del tráfico recopilado. Además, conseguimos que las diferentes secciones de la pantalla se comuniquen entre sí. Cuando aparecen nuevos datos, se envían a los otros componentes que los requieran, asegurando que todo se muestre de manera uniforme y simultáneamente.

  Al iniciar, notamos avisos en la pantalla que nos dicen cómo está la conexión y los datos. Si hay algún fallo al comunicar el cliente con el servidor o al manejar la información, saldrá un mensaje explicando el problema y dando ideas sobre cómo seguir, haciendo todo más simple y ayudando a evitar problemas.

- **Desarrollo técnico del gráfico: Volumen de Tráfico Acumulativo**

  Este gráfico representa la evolución de paquetes a lo largo de tiempo enviando una alerta en el correo cuando alcanza un umbral predefinido de paquetes en el correo electrónico. En lado servidor, se desarrolla un script PHP que está conectada a la base de datos para ejecutar la consulta que permite extraer la cantidad de paquetes recolectados del tráfico de red, además, para tener la opción del envió de alertas debemos implementar otro script PHP y que está configurado mediante la biblioteca PHPMailer y con la autenticación SMTP segura usando una contraseña de aplicación de Gmail.

  La librería PHPMailer [20] es una biblioteca de PHP que regula el envío de correos electrónicos a través de PHP. Para tener sus funcionalidades en el script PHP que, tenemos que respectar estas instrucciones:

1- Instalar PHPMailer a partir GitHub.
2- Descomprimir archivos: incluir los archivos PHP “PHPMailer.php”, “SMTP.php”, y “Exception.php” en el script de alertas.
3- Para acceder al PHPMailer, aseguramos de que los espacios de nombres están incluidos correctamente. Para ello, se utilizan las sentencias use.
4- Para evitar enviar datos sensibles desde tu servidor de correo a los usuarios en forma de mensaje de error, es recomendable envolver el envío de correos electrónicos con la sentencia Try-Catch.
5- Para utilizar PHPMailer, se debe autenticar a través de SMTP. Para ello, introduce la dirección del servidor de correo, el protocolo correspondiente (ya sea TLS/SSL o SMTP), el puerto, el nombre de usuario y la contraseña.
6- Introducir el destinatario del correo electrónico.
7- Introducir el contenido del correo electrónico. Este contenido suele consistir en un asunto y un texto, que se ofrece tanto en versión HTML como no HTML.
8- Enviar el correo electrónico.

Después la configuración de script de envío de alertas, se debe mencionarlo en el script que ejecuta la consulta, distingue entre usuarios normales y administrador, filtra los datos por rangos de tiempo y filtra los datos de una forma que incluye tomar muestras para que el cliente no se sature. Además, se ponen marcadores especiales en la gráfica justo donde saltó la alerta, siempre y cuando haya pasado dentro del tiempo que seleccionamos.

El lado cliente este encargado de implementar el diseño del gráfico interactivo en tiempo real usando la biblioteca Chart.js y el lenguaje Javascript que realiza peticiones periódicas al servidor mediante fetch que es una función integrada que se utiliza para hacer peticiones HTTP (como GET, POST, etc.) a servidores web obteniendo los datos extraídos de la base de datos a través de la consulta SQL que está en el script PHP. También, este script Javascript implementa un punto rojo que se sitúa sobre la línea del gráfico en el momento de alcanzar el umbral predefinido. Y los tooltips que ofrecen datos específicos cuando se coloca el cursor sobre un punto, indicando el momento exacto y la cantidad de paquetes contabilizados en ese segundo, lo que ayuda a realizar un análisis más preciso y en contexto de la información presentada.

Si surge algún error, el sistema produce notificaciones JSON bien estructurados que proporcionan notificaciones claras e información de tiempo precisa, lo que los hace más fáciles de usar para el sitio web.

- **Desarrollo técnico de la Mapa De Distribución de Paquetes**

  Este gráfico permite visualizar la distribución geográfica del tráfico de internet capturado mostrando en un mapa la cantidad de paquetes por país.

  Para lograrlo, se debe preparar la parte servidor que gestiona primero la conexión con la base de datos y que controla también los permisos de acceso: según el rol del usuario autenticado, si los usuarios son generales, acceden solamente a sus propios datos mientras que un administrador puede visualizar la distribución geográfica total de tráfico de internet. Además, se implementa una lógica para detectar si la captura es activa. Si el recuento de los datos durante muchas solicitudes consecutivas queda constante, se considera que la captura ha finalizado.

  En cuanto a la parte del cliente, se muestra los paquetes de forma dinámica a través de una interfaz interactuada. Además, se ofrece la posibilidad de observar los paquetes representados en un mapa global donde cada país se pinta de un color según la cantidad de paquetes recibidos. Adicionalmente, el mapa se actualiza automáticamente a intervalos establecidos para mostrar cualquier otro tráfico registrado.

  En cuanto al filtrado de direcciones IP de origen, el sistema ofrece a los usuarios la opción de elegir una dirección IP concreta para restringir la visualización del tráfico solamente a esa dirección. Este filtro se implementa desde el cliente, pero lo gestiona el servidor en el instante de la consulta, de manera que se devuelven únicamente los paquetes que pertenecen a la dirección IP elegida. Esto es beneficioso para analizar el comportamiento de una sola fuente en el tráfico total y ayuda a identificar patrones, irregularidades o posibles amenazas relacionadas con esa dirección IP específica. Al activar el filtro, la interfaz se actualiza automáticamente para mostrar solo la información vinculada a esa dirección IP, sin que sea necesario recargar la página de forma manual.

  El sistema también ofrece la opción de usar filtros por tiempo. Así, el usuario puede elegir un intervalo particular y revisar únicamente la información de ese periodo. Esto resulta útil para enfocarse en aspectos concretos o para analizar secciones específicas del tráfico recogido.

  En el caso de un error de conexión o en la obtención de datos, el cliente ofrece alertas informativas al usuario para garantizar una experiencia fácil de entender. Además, se registra información útil en la consola para el diagnóstico, lo que hace más sencillo resolver problemas mientras se utiliza.

- **Desarrollo técnico del gráfico: Tráfico DNS: Evolución de totales vs. dominio filtrado**

  El gráfico presenta un diseño lineal que muestra la comparación entre el desarrollo temporal de las solicitudes y respuestas DNS a nivel global, junto con las de un dominio específico que el usuario ha seleccionado.

  El servidor maneja la consulta a la base de datos MYSQL, que proporciona tanto las solicitudes y respuestas completas como aquellas filtradas por el dominio elegido. Con el filtro de dominio, uno puede enfocarse en las peticiones y las respuestas del dominio que se especificó. Aparte, el servidor se ocupa de revisar y darles paso a los usuarios, fijando limites según su categoría: los usuarios estándar apenas pueden observar la información limitada, pero los administradores tienen permiso para observar todos los datos.

  En la parte cliente, los datos se visualizan a partir de un gráfico lineal implementado a partir de una librería Chart.js que admite la visualización del flujo de tráfico DNS es decir la comparación entre las solicitudes y las respuestas total frente a aquellas específicamente relacionadas con el dominio filtrado en función de tiempo de captura. Una vez que inicia la captura, el gráfico de líneas se actualiza automáticamente gracias a las solicitudes AJAX que se envían a intervalos regulares. Esto facilita mostrar cambios en el instante, sin necesidad de recargar la página.

  El filtro de dominio se realiza a través de un formulario donde el usuario puede ingresar un dominio para aplicar el filtro. Esto permite al usuario observar las solicitudes y respuestas DNS de un dominio específico. Al introducir el dominio, el cliente envía esa información como un parámetro en las solicitudes AJAX, que se utiliza para modificar la consulta SQL y entregar solo los registros que pertenecen a ese dominio. Esta herramienta de filtrado resulta esencial para examinar a fondo cómo se comporta y qué hace el nombre de dominio DNS en relación con determinados dominios, permitiendo así descubrir tendencias en el flujo de información. Su funcionamiento pasa seguro para quien lo utiliza, pues el diagrama se actualiza solo y no requiere refrescar la página, lo que hace más dinámico el trabajo y simplifica el análisis interactivo de la información.

  Para una vista más cómoda a la hora de tener un tráfico de internet, se emplea un modal que aparece si pulsamos sobre cualquier línea del gráfico permitiendo observar las solicitudes y respuestas a lo largo de tiempo de forma más clara.

  Si algo no funciona como se espera, recibirás mensajes que te mostrarán si hay problemas con la conexión o al intentar obtener información. También se registran todos los eventos, incluidos los fallos, en la consola del navegador, lo cual es útil para entender lo sucedido y mantener las cosas organizadas.

- **Desarrollo técnico del gráfico: Distribución Totales de Tipos de Registros DNS**

  El gráfico circular presenta los tipos de registros DNS capturados en el tráfico de internet, a continuación, se describe su desarrollo que se llevó cabo a partir de un enfoque cliente-servidor.

  En el servidor, se establece un archivo PHP que está conectado a la base de datos encargado de ejecutar una consulta que extrae la cantidad de paquetes DNS hay en cada tipo de registro DNS filtrando los resultados según el identificador de usuario y por el rango de tiempo seleccionado. El resultado se organiza en formato JSON y se envía al usuario, lo que permite la actualización en tiempo real del gráfico que muestra estos tipos de registros como diferentes secciones, cada una con un color y etiqueta diferentes.

  En el lado cliente, el gráfico esta creado en HTML, CSS y Javascript usando la biblioteca Chart.js que facilita la creación de gráficos visuales. A la hora de iniciar la captura, se ejecuta en el Javascript una función que permite pedir los datos en forma de una petición al servidor (script PHP). Esta petición se repite cada intervalo de tiempo, una vez el servidor recibe la solicitud, responde con los datos en formato JSON para que se añaden los nuevos valores al gráfico sin recargar la página.

- **Desarrollo técnico de gráfico: Flujo DNS: Consultas y Respuestas entre IPs y Dominios**

  Este resultado visual representa un gráfico de dispersión dinámico que permite visualizar las solicitudes y respuestas entre las direcciones IPs de origen y los dominios consultados en la hora de capturar tráfico de internet. La implementación de este gráfico se establece a partir de una relación cliente-servidor.

  En la parte servidor, se implementa un script PHP que está conectado a la base de datos que almacena toda la información relevante del flujo de internet, además, ejecuta la consulta que obtiene y organiza las conexiones DNS almacenadas en la base de datos, haciendo una distinción entre solicitudes y respuestas. Utiliza una cláusula UNION ALL para unir dos grupos de información: el primero organiza por la dirección IP de origen y señala el dominio solicitado (queryname), mientras que el segundo organiza las respuestas usando el nombre del dominio que se respondió (respname). Los resultados están filtrados por dirección IP de origen, usuario y rango temporal y enviadas en formato JSON al usuario, lo que admite actualizaciones automáticas del gráfico.

  Dentro de los filtros mencionados existe el filtro de usuario y el filtro temporal que están aplicados en todos los gráficos y en todas las tablas de la plataforma acordando que el primero es un filtro que permite mostrar solamente los datos del usuario autenticado, el segundo es un filtro que admite seleccionar un intervalo de tiempo personalizado para visualizar la evolución del tráfico DNS, Adicionalmente, encontramos el filtro de direcciones IPs de origen que se trata de un formulario donde se filtran los resultados según la dirección IP introducida, así permitiendo visualizar únicamente los resultados de una dirección IP específica.

En el lado cliente, el gráfico de dispersión esta implementado a través el lenguaje Javascript junto con la librería Chart.js. En este gráfico, cada punto muestra una relación DNS entre una dirección IP de origen y un dominio que se busca o responde. En este tipo de representación, los nodos son entidades (direcciones IP y dominios), y las líneas que los unen representan las consultas o respuestas DNS, diferenciadas por colores o estilos según el tipo de mensaje. Esto ayuda a entender de forma visual las conexiones y la cantidad de tráfico DNS que produce cada fuente. Además, para la visualización del gráfico en la interfaz web en tiempo real, el cliente realiza peticiones AJAX al servidor cada vez que se actualiza el gráfico de esta manera obtenemos nuevos datos cada intervalo de tiempo.

Para mejorar la experiencia de usuario, se ha añadido una opción la cual es cuando se pulsa un nodo sale las relaciones dirección IP-dominio que tiene este nodo en un modal, ofreciendo una vista más cómoda para el análisis.

- **Desarrollo técnico del gráfico: Consulta de Dominios**

Este gráfico se trata de una tabla donde se consultan los dominios más o menos visitados.

En el servidor, se establece un script PHP que está conectado a la base de datos y encargado de procesar las solicitudes del cliente, también, ejecuta una consulta SQL que filtra los registros DNS por el campo queryname distinguiendo los dominios más solicitados o los menos solicitados según el parámetro tipo_consulta.

En esta tabla, se aplica el filtro temporal que restringe los resultados a un período particular dentro de la recolección de datos de tráfico DNS. También, se establece el filtro de usuario para que cada usuario únicamente sus propios datos.

En la parte de cliente, Javascript se encarga de gestionar el formulario, implementar los filtros de tiempo y de usuario y realizar las solicitudes AJAX al servidor. La información recopilada se organiza en una tabla adaptable que cambia de acuerdo con el tipo de búsqueda del usuario. Asimismo, en la tarjeta del grafico existe un campo donde se avisa mientras se trabaja con los datos, muestra avisos si hay algún problema y ofrece protecciones como la codificación de caracteres HTML.

- **Desarrollo técnico del gráfico: Respuesta de Dominios**

  Este gráfico se trata de una tabla donde se consultan los dominios con mayor y menor número de respuestas en el tráfico DNS capturado.

  En la parte servidor, se emplea un script PHP que está conectado a la base de datos, recibe los parámetros requeridos, también, realiza una consulta adaptada al usuario autenticado y a los filtros de tiempos seleccionados que identifica los dominios más frecuentes en las respuestas DNS capturadas devolviendo los resultados en formato JSON que son ordenados de forma descendente o ascendente según el tipo de respuesta seleccionado por el usuario.

  En lado cliente, JavaScript se utiliza para actualizar la información en el formulario y actualizarla utilizando el botón "consultar" dentro del navegador del usuario. También se incluye un filtro de tiempo que puede manejar eventos personalizados mostrando el indicador de carga cuando se traen datos y presentan los resultados inmediatamente en una tabla HTML.

  Para manejar los errores, se observa mensajes informativos para evitar errores o la ejecución de un código malicioso.

- **Desarrollo técnico del gráfico: Comparación de Dominios DNS**

  Este gráfico representa la evolución de los tipos de registro DNS (A, AAAA, NS, PTR y otros) de tres dominios presentes en el tráfico de internet capturado a lo largo de tiempo. Como los demás gráficos, la implementación sigue una arquitectura cliente-servidor.

  En el lado servidor, un script PHP accede a la base de datos para ejecutar la consulta que extrae los tipos de registros de cada dominio ingresado dentro de un intervalo de tiempo específico y por usuario, esta consulta permite contar cuántas veces ha aparecido cada clase de registro DNS durante el tiempo elegido, garantizando que la información sea relevante para el usuario que está visualizando el gráfico. Además, existe un formulario donde se puede elegir la visualización de la comparación de todos los tipos registros o de un tipo de registro DNS concreto. El resultado de esta consulta se convierte a formato JSON y se envía al navegador del usuario.

  En la parte cliente, se utiliza Javascript junto con la biblioteca Chart.js para generar este gráfico lineal donde se comparan los tipos de registros DNS de tres dominios, sea preferible observar los resultados cuando se termina la captura es decir cuando se acaba la captura de dominios DNS así se puede comparar los tipos de registros DNS de cualesquiera dominios recolectados de tráfico de internet.

5. RESULTADOS Y DISCUSSIÓN

5.1 Resultados obtenidos

Tras de la explicación de la implementación técnica de los gráficos, pasamos a mostrar la salida de los elementos visuales mostrando como se procesan y visualizan los datos capturados con un centro sobre las métricas, filtros y métodos gráficos utilizados en ellos.

- **Resultados: Distribución de paquetes por protocolo**

En la figura siguiente, podemos observamos que es un gráfico de barras situado en la primera tarjeta de la interfaz web, tiene como título Distribución de Paquetes por Protocolo, representa la cantidad de paquetes tenida en cada tipo de protocolo que se refiere a la cantidad total de paquetes de internet que se han recogido y que corresponden a cada protocolo particular, como HTTPS, DNS, TCP, UDP, y otros y representarlas en forma de barras coloradas en azul. Al pasar el ratón sobre la barra, se muestra el número de paquetes en un tooltip.

La métrica es un conteo de paquetes correspondientes a cada protocolo obteniendo una visión general del tipo de tráfico de internet. Sirve para diagnosticar, analizar el rendimiento y la detección de anomalías.

Antes de la visualización de los datos en el gráfico, se aplica el filtro de usuario que comprueba que usuario está iniciando la sesión. A continuación, se observa que si iniciamos la sesión con el primer usuario general fijamos que durante esta captura se han recopilado una cantidad grande de paquetes de tipo de protocolo HTTPS.

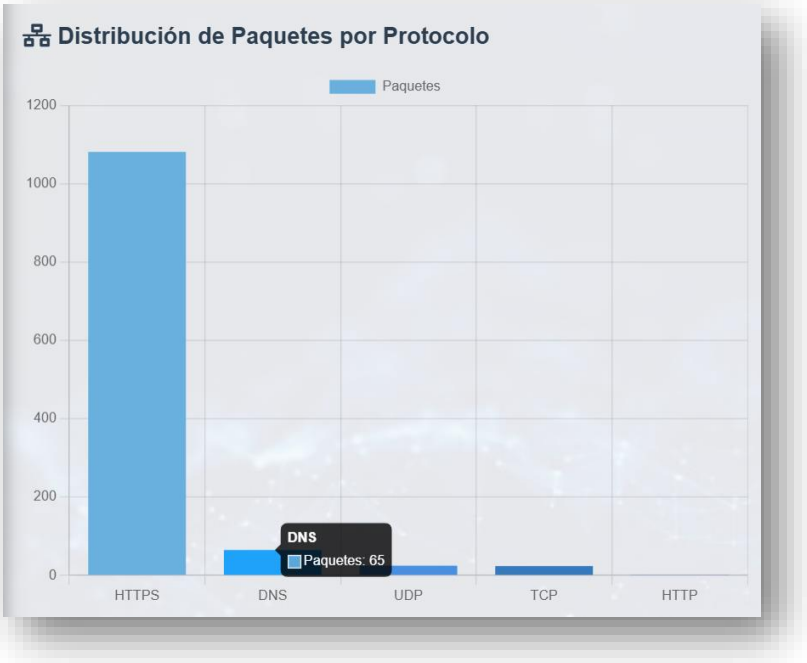

Figura 29: Distribución de paquetes por protocolos – Usuario 1.

Además, al analizar la captura de otro usuario, notamos que también se han registrado una gran cantidad de paquetes del tipo HTTPS.

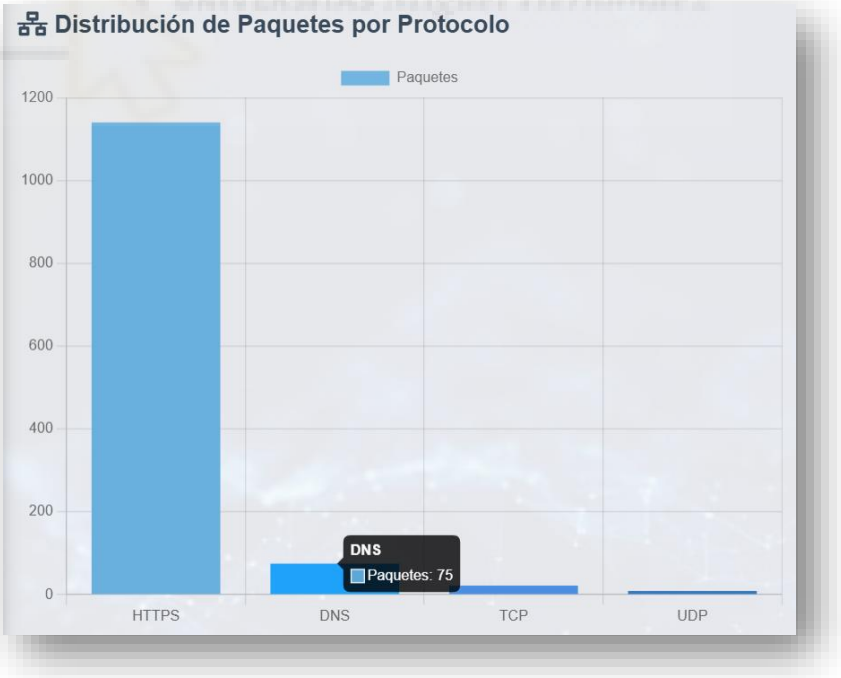

Figura 30: Distribución de paquetes por protocolos – Usuario 2.

A continuación, observamos cómo se añade el filtro de tiempo en este gráfico. Tomamos la captura del primer usuario normal como un ejemplo y notamos que la duración completa de la captura es de 5,33 minutos. Por ejemplo, seleccionamos los dos primeros minutos como intervalo y aplicamos el filtro.
Observamos que el gráfico esta actualizado dando nuevo número de paquetes para cada protocolo, como hemos fijado anteriormente en la captura completa que para este usuario hemos tenido un numero de paquetes de protocolo nombre de dominio DNS es 65 y en este intervalo de tiempo es 25 paquetes DNS.

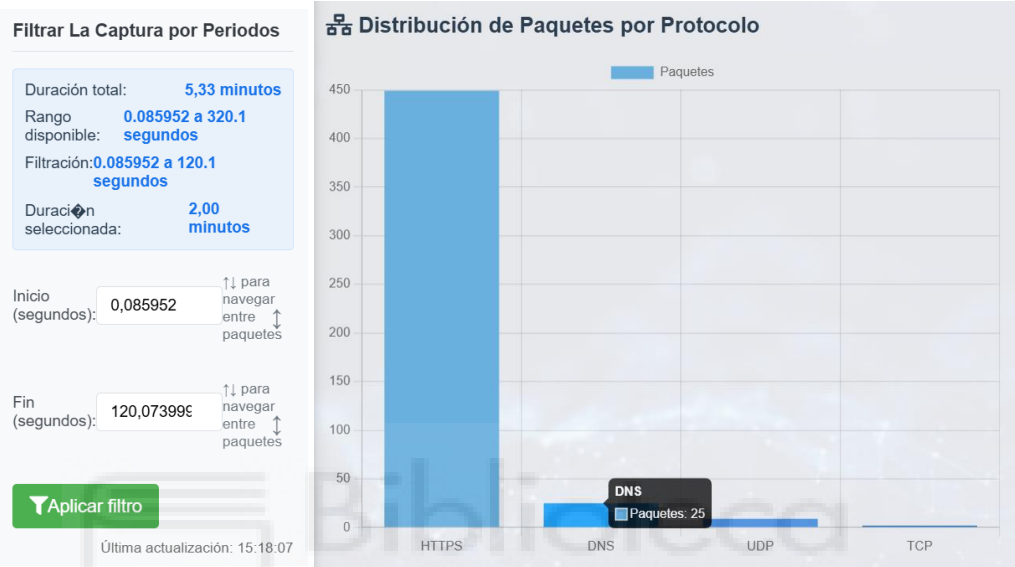

Figura 31: Filtro temporal aplicado a distribución de paquetes por protocolos.

- **Resultados: Volumen de Tráfico Acumulativo**

Este gráfico representa la evolución de tráfico de internet de manera acumulativa en función de tiempo de captura, es decir, figura el total de paquetes de la captura, también, implementa una opción de envío de una alerta al correo electrónico una vez alcanzamos un umbral de 1000 paquetes, esta opción ayuda a evitar la saturación de red o en lugar de quedar observando el gráfico constantemente, la alerta avisa que el análisis de datos ha alcanzado una cantidad predefinida de paquetes.

La figura siguiente nos permite visualizar que el gráfico es de tipo lineal, la línea que representa la evolución de paquetes es de color verde. La alerta está diseñada tal como un cuadro rojo que este situado exactamente sobre el momento en que la alerta fue enviada al correo electrónico. Además, notamos al pasar el ratón sobre el origen del gráfico sale un tooltip que proporciona datos sobre el primer y el milésimo paquete (alerta), incluyendo su hora de llegada y el tiempo de captura.

Adicionalmente, no olvidamos que el gráfico tiene como filtro, el filtro de usuario lo cual que la figura representa un ejemplo de captura del primer usuario normal.

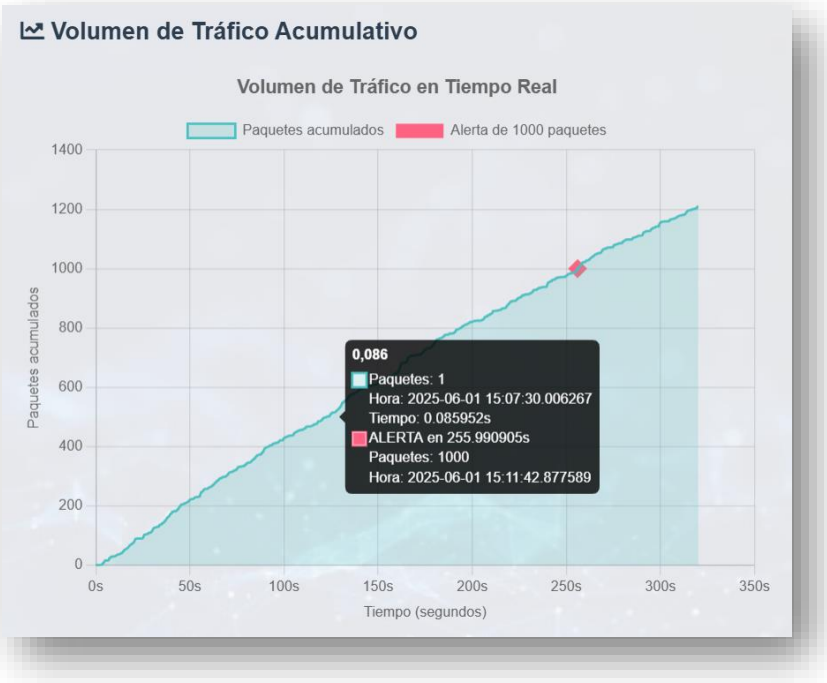

Figura 32: Volumen de tráfico acumulado – Usuario 1.

La alerta recibida en el correo electrónico cuando se alcanza un umbral de 1000 paquetes para el primer usuario normal.

Figura 33: Alerta de tráfico – Usuario 1.

Para el segundo usuario normal tiene como gráfico:

Figura 34: Volumen de tráfico acumulado – Usuario 2.

La alerta recibida en el correo electrónico cuando se alcanza un umbral de 1000 paquetes para el segundo usuario.

Figura 35: Alerta de tráfico – Usuario 2.

Para visualizar la aplicación del filtro de tiempo en este tipo de gráfico, empleamos como ejemplo la captura del segundo usuario normal, elegimos los dos últimos minutos y aplicamos el filtro. Observamos que en este periodo se han capturado 498 paquetes.

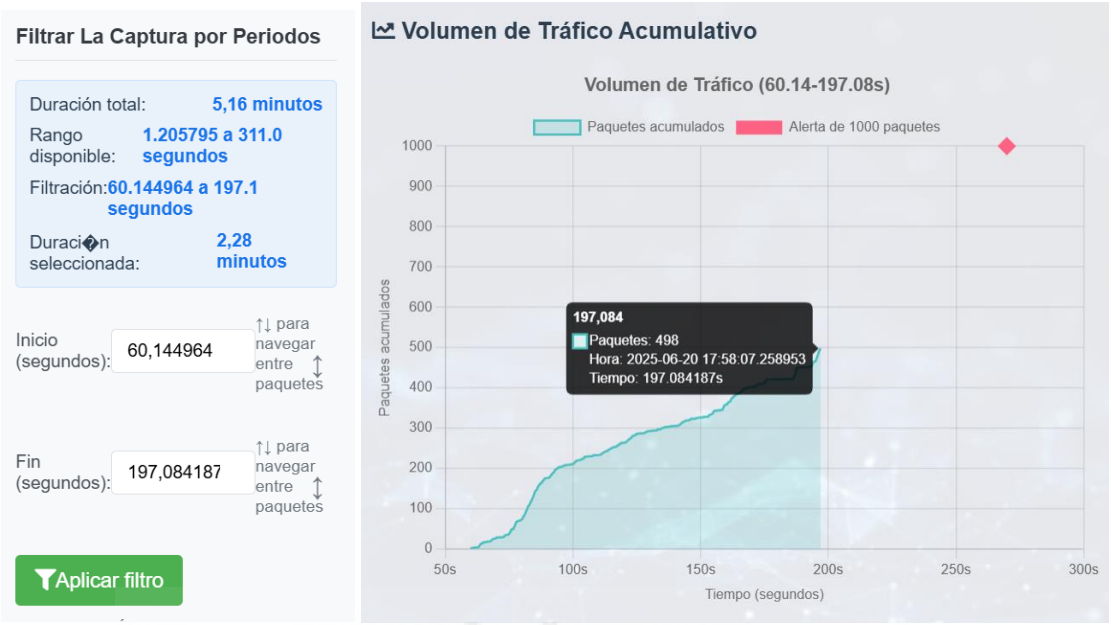

Figura 36: Filtro temporal aplicado al volumen de tráfico.

- **Resultados: Mapa de Distribución de Paquetes**

Este gráfico representa un mapa que esta visualizada gracias a la API de Google charts que permite mostrar datos en forma de un mapa geográfico. Este mapa admite la distribución total de los paquetes por país de destino que es una métrica que muestra el número total de paquetes de datos que se mandan a servidores en diferentes lugares del mundo, un cálculo que se hace ubicando geográficamente las direcciones IP a las que se dirigen los paquetes. Para visualizar el procesamiento de este gráfico, introducimos un ejemplo de los resultados de la captura de los usuarios de esta herramienta:

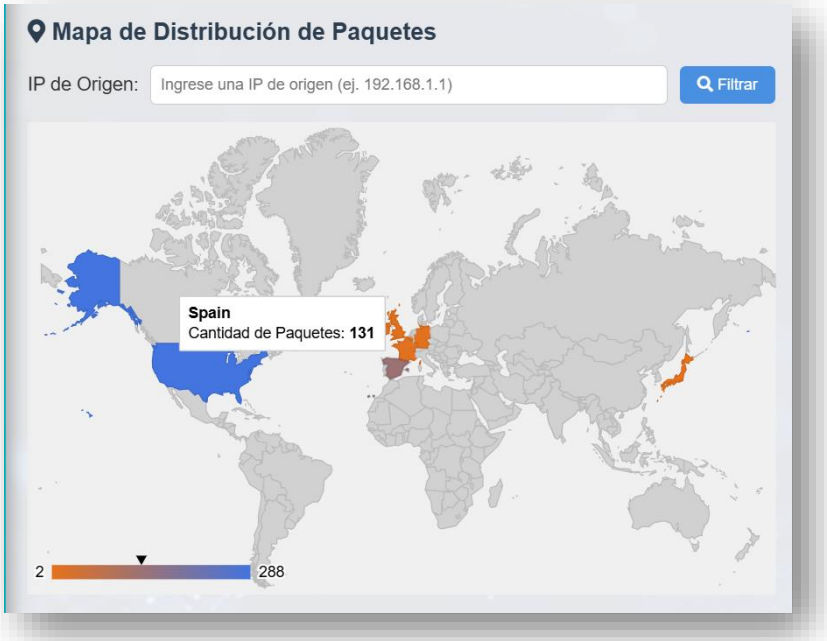

Figura 37: Mapa de distribución de paquetes – Usuario 1.

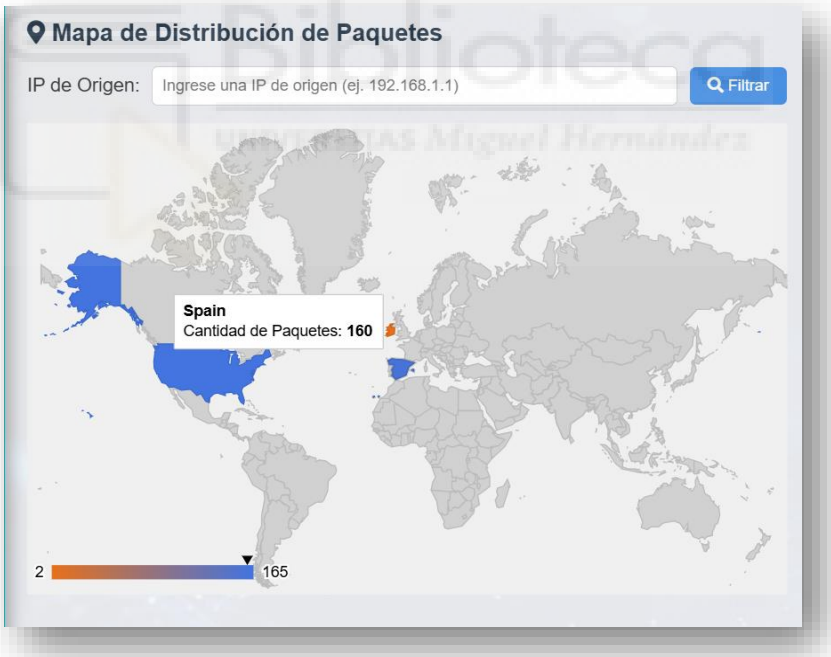

Figura 38: Mapa de distribución de paquetes – Usuario 2.

Entre los filtros que se utilizan en este tipo de gráfico, hay uno que se basa en la dirección IP de origen. Este filtro selecciona los paquetes de acuerdo con la dirección IP de origen, lo que significa que se piden los paquetes que corresponden a una dirección IP que se introduce en el formulario que aparece encima del mapa.

Para ver los resultados que aparecen al usar este filtro, tomamos como muestra los resultados de una captura del segundo usuario junto con el menú desplegable de las direcciones IP de origen existentes en la captura. Al observar la figura, notamos que no existen paquetes para otra dirección IP de origen diferente a 192.168.18.37 esto significa que solo los paquetes con dirección IP de origen del equipo están geolocalizados.

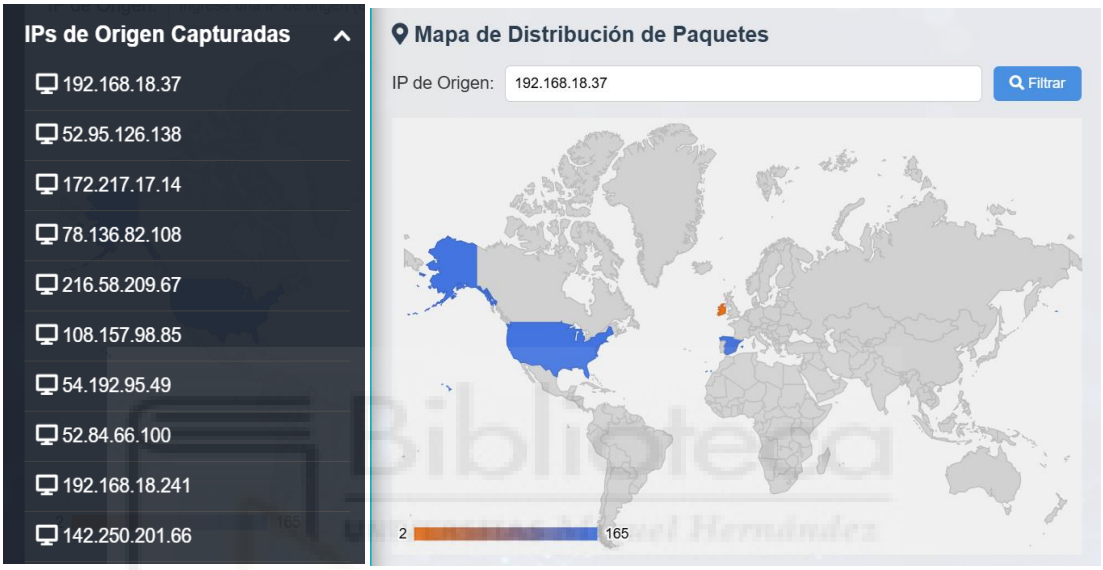

Figura 39: Filtro por IP 192.168.18.37.

Figura 40: Filtro por IP 192.168.18.241.

A continuación, fijamos los resultados de la utilización del filtro temporal sobre el mapa de la distribución de paquetes. Tomamos como ejemplo la captura del segundo usuario, configurando el filtro que tiene como duración total 3.29 minutos y seleccionamos el primer minuto como periodo de prueba.
Hemos observado que la captura completa que se mencionó anteriormente ha registrado un total de 160 paquetes, localizados en España. Ahora, notamos que durante el primer minuto se han registrado 47 paquetes con el mismo destino.

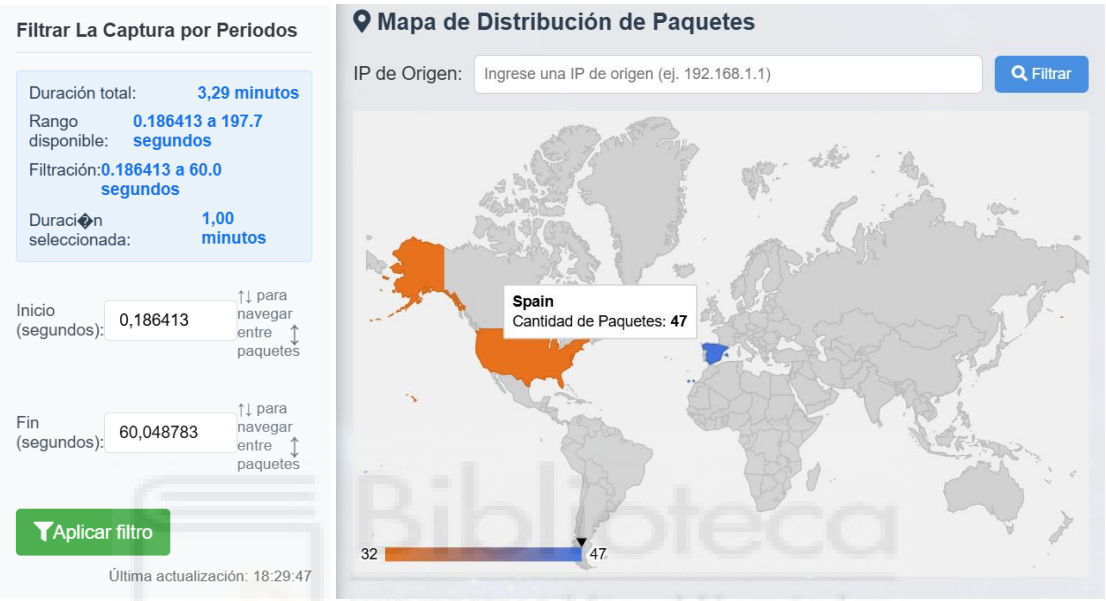

Figura 41: Filtro temporal aplicado al mapa de distribución de paquetes.

- **Resultados: Tráfico DNS: Evolución de Totales vs. Dominio Filtrado**

El gráfico representa un gráfico lineal que permite visualizar las solicitudes y respuestas totales DNS en función de tiempo captura frente a las solicitudes y respuestas de un dominio DNS. Esta métrica tiene como utilidad facilitar la comparación entre el tráfico DNS general y el de un dominio en particular, lo que es fundamental para identificar irregularidades o comportamientos extraños en la red. Por ejemplo, si un único dominio comienza a recibir muchas más solicitudes de lo habitual mientras que el tráfico total se mantiene constante, esto podría indicar la presencia de un programa malicioso, una configuración inadecuada o incluso un intento de ataque.

La figura siguiente nos permite visualizar que las solicitudes y repuestas están diseñadas en formato de unas líneas coloradas con puntos que significan el momento cuando se reciban los solicitudes y respuestas. Las solicitudes totales son de color rosa, las respuestas totales son de color azul. Además, existe un tooltip que indica en cada punto el número de solicitud o respuesta y la hora de llegada.

La figura es un ejemplo de una captura del primer usuario normal:

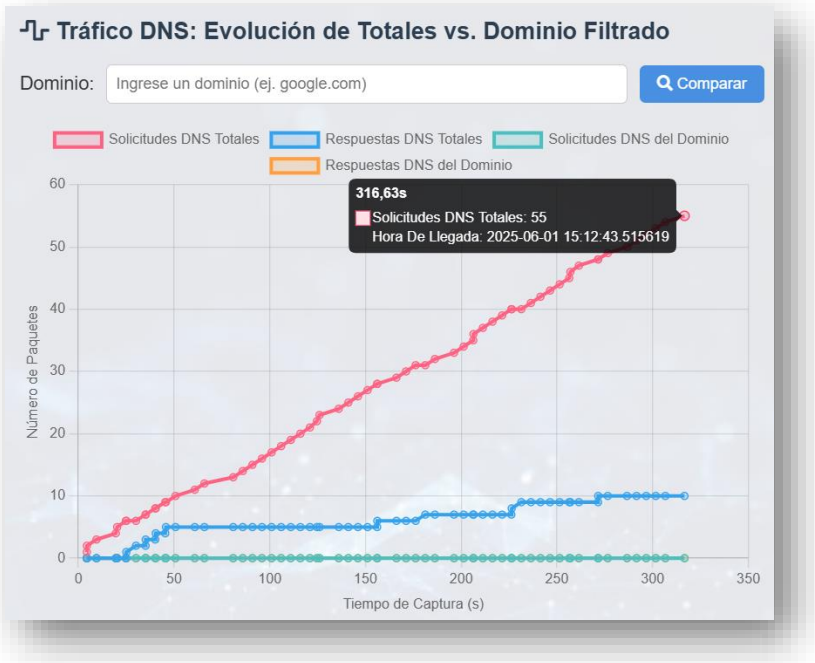

Figura 42: Evolución de tráfico DNS frente a dominio – Usuario 1.

Para el segundo usuario, el ejemplo de captura:

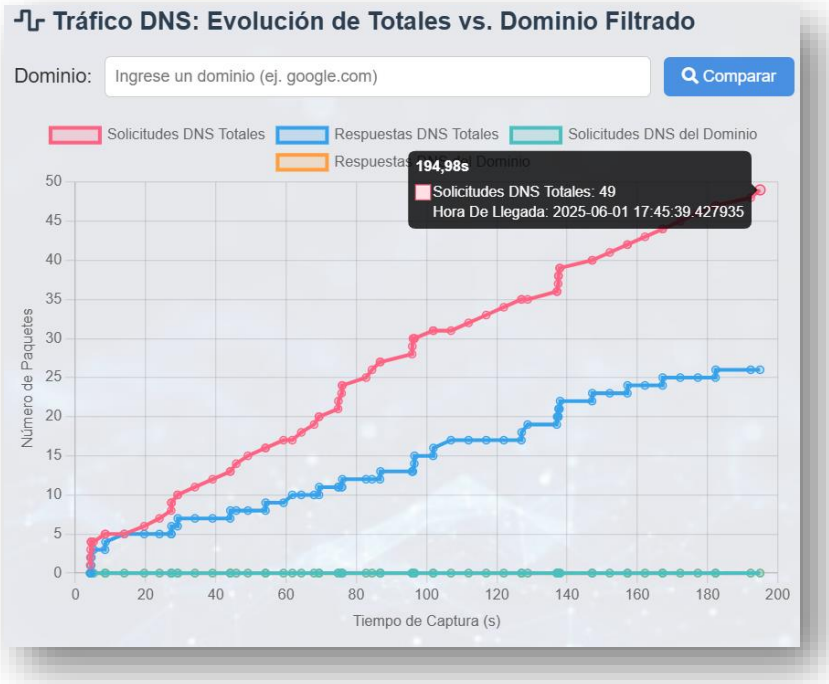

Figura 43: Evolución de tráfico DNS frente a dominio – Usuario 2.

A continuación, visualizamos la aplicación del filtro de dominio en la captura de primer usuario normal, observamos que existe un formulario por encima del gráfico donde podemos ingresemos un dominio y comparar sus resultados con el tráfico DNS total.

En este caso, observamos que el dominio DNS “Google.com” recibe la mayoría de las solicitudes, con aproximadamente 9 peticiones de un total de 10, y notamos que ninguna de estas solicitudes ha obtenido una respuesta DNS.

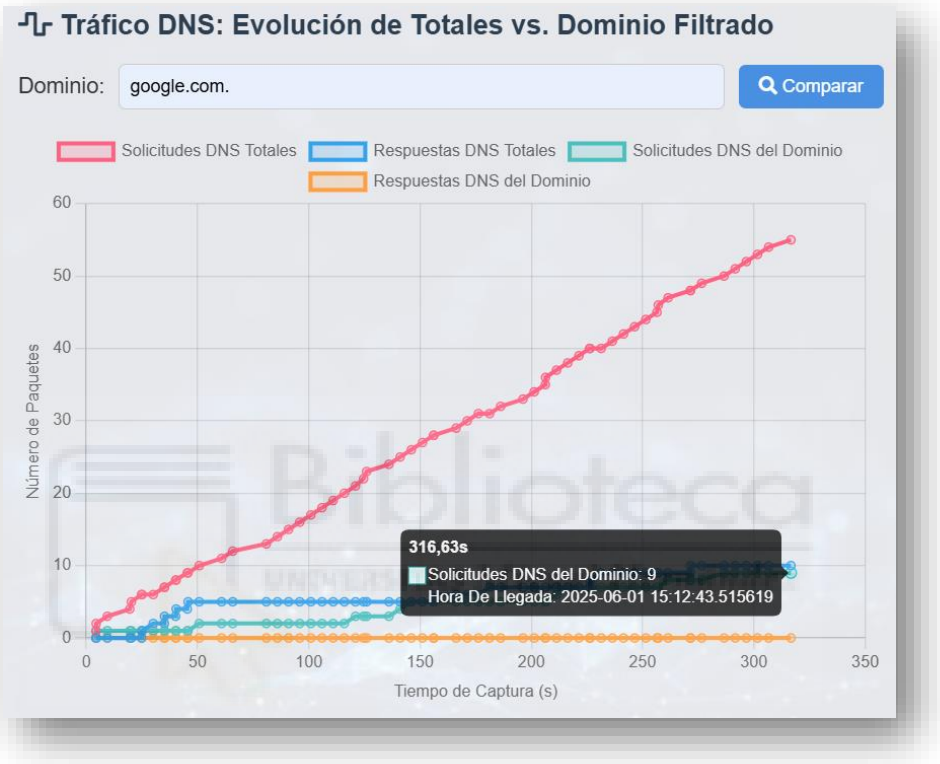

Figura 44: Filtro por dominio en tráfico DNS.

Seguidamente, observamos la aplicación el filtro temporal, tomamos como ejemplo los dos primeros minutos de la captura del primer usuario y fijamos que hay un cambio en el gráfico, lo cual se han registrado en este rango unas 20 solicitudes DNS totales y 5 respuestas DNS totales frente a dos consultas DNS de dominio ingresado.

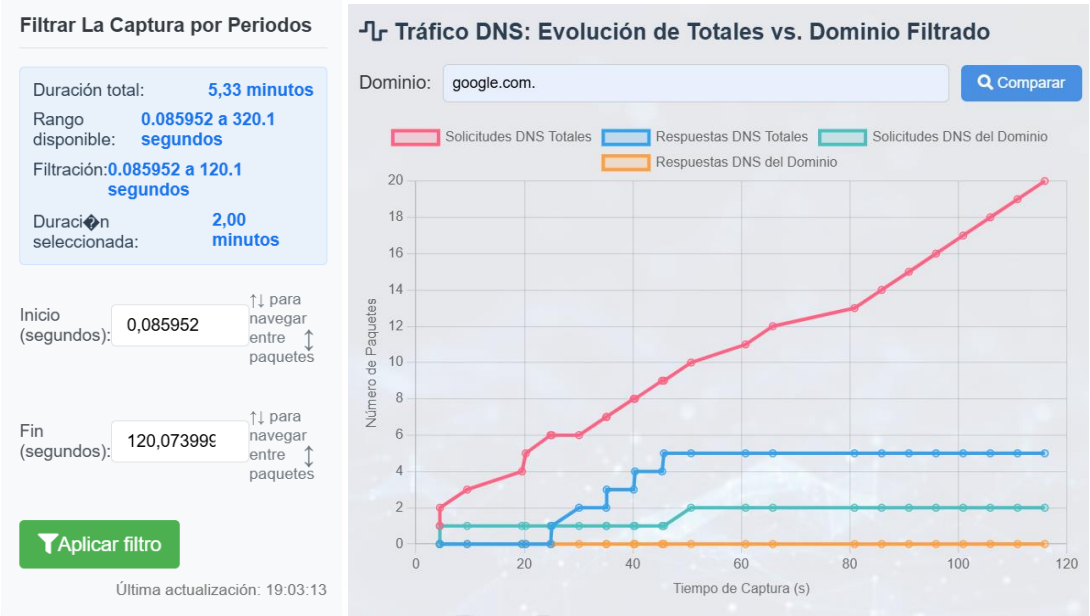

Figura 45: Filtro temporal aplicado al tráfico DNS.

- **Resultados: Distribución Totales de Tipos de Registros DNS**

Este gráfico representa la distribución total de tipos registros DNS en formato de un gráfico circular. Esta métrica es beneficiosa ya que ofrece una manera clara y rápida de observar qué variedades de registros DNS (como A, AAAA, CNAME, SOA, etc.) son más comunes en una red. Al mostrarla en un gráfico circular, se hace más fácil entender y comparar los distintos tipos de registros, lo que contribuye a detectar tendencias de uso, configuraciones atípicas o comportamientos sospechosos.

Las figuras son ejemplos de capturas del primer y segundo usuario que permiten visualizar la distribución de paquetes de tipos de registros DNS donde cada tipo esta colorado por un color predefinido.

Pare estos casos, observamos que se han registrado paquetes de nombre de dominio DNS de tipo A, SOA, MX, PTR y AAAA en la captura del primer usuario normal. En la captura del segundo usuario normal, se han anotado los mismos, pero también registros de tipo HTTPS, CNAME y TXT. Al colocar el cursor sobre el gráfico circular, observamos que para cada tipo aparece un cuadro de información que nos detalla la cantidad de paquetes registrados para cada tipo de registro DNS.

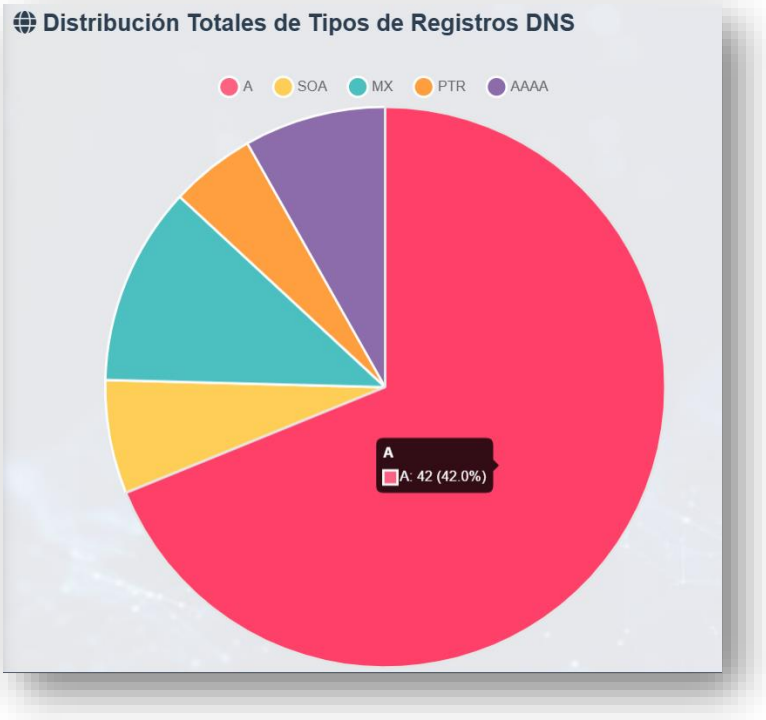

Figura 46: Distribución de tipos de registros DNS – Usuario 1.

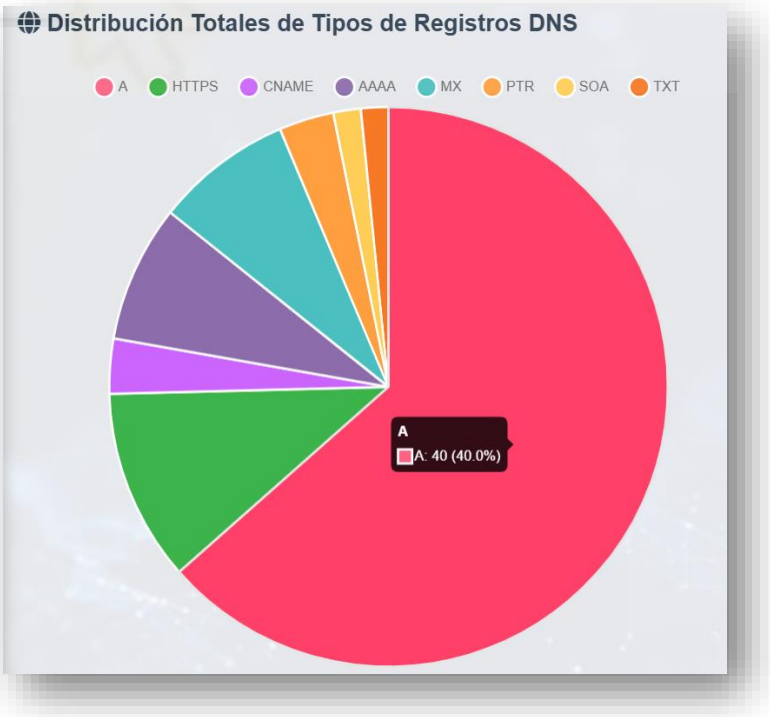

Figura 47: Distribución de tipos de registros DNS – Usuario 2.

A continuación, observamos la aplicación el filtro temporal, tomamos como un ejemplo el segundo minuto y notamos que hay un cambio en el gráfico de la captura del segundo usuario.
Fijamos que, al comparar la captura total anterior con el tráfico del periodo elegido, notamos que en el tiempo total se han detectado 42 paquetes de tipo A, mientras que en el periodo elegido se han registrado 14 paquetes de tipo A.

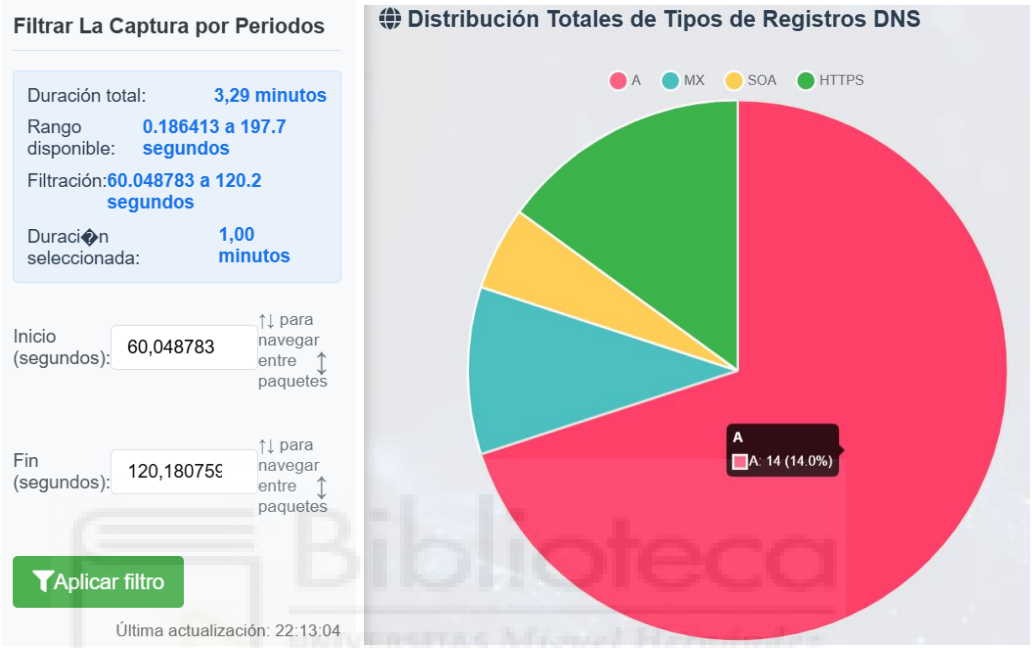

Figura 48: Filtro temporal en tipos de registros DNS.

- **Resultados Flujo DNS: Consultas y Respuestas entre IPs y Dominios**

Este gráfico permite visualizar las consultas y respuestas entre las direcciones IP de origen y Dominios DNS, esta métrica tiene como utilidad observar las conexiones y patrones de comunicación en la red, Facilitar la identificación rápida de qué direcciones IP están produciendo más tráfico DNS y saber qué dominios son más visitados y si hay comportamientos inusuales, como una dirección IP que consulta varios dominios diferentes.

Las figuras son ejemplos de captura del primer y segundo usuario lo que significa que en este gráfico también se aplica el filtro de usuario. A partir de estas figuras podemos visualizar que los nodos circulares colorados en azul son las direcciones IP de origen que tienen un tráfico DNS, los nodos cuadrados colorados en rosa son los dominios DNS capturadas en la red, las flechas son las conexiones, las que están en color azul son las solicitudes y las amarillas son las respuestas, además, cuanto más gruesa sea la flecha más conexiones existen en la dirección IP de origen y el dominio.

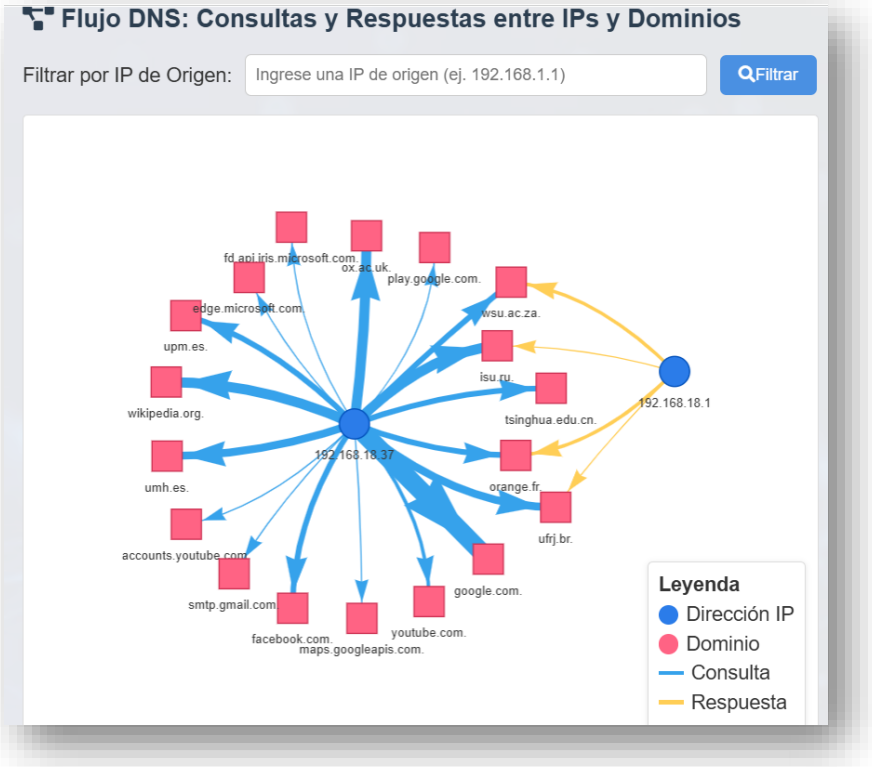

Figura 49: Dispersión DNS por IPs y dominios – Usuario 1.

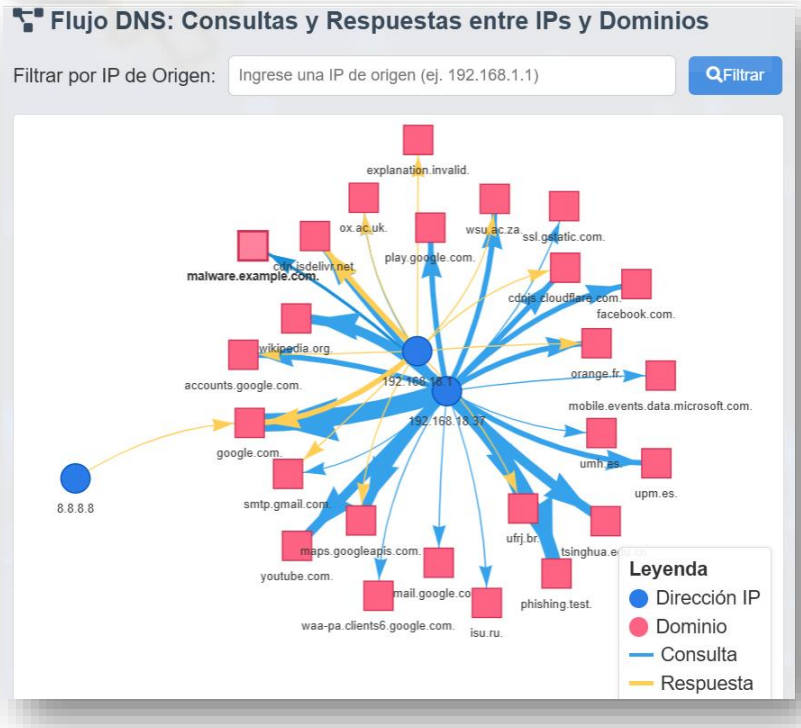

Figura 50:Dispersión DNS por IPs y dominios – Usuario 2.

A continuación, visualizamos la realización del Filtro por dirección IP de origen en la captura del primer usuario, primero, ingresamos por ejemplo la dirección IP 192.168.18.1 y notamos que el tipo de conexión que presenta es una respuesta relacionada con los dominios.

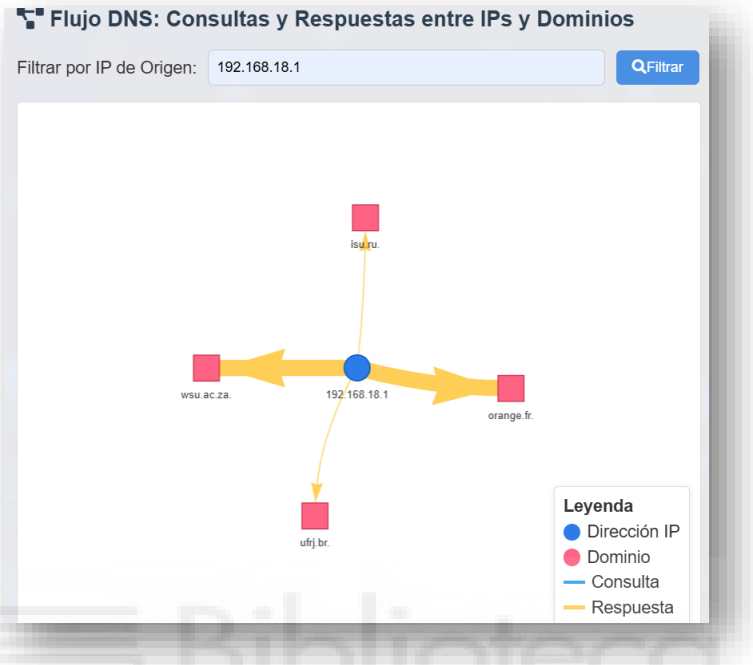

Figura 51: Filtro por IP en gráfico de flujo DNS.

Seguidamente, observamos la aplicación el filtro temporal, tomamos como un ejemplo el primer minuto y notamos que hay un cambio en el gráfico de la captura del segundo usuario.

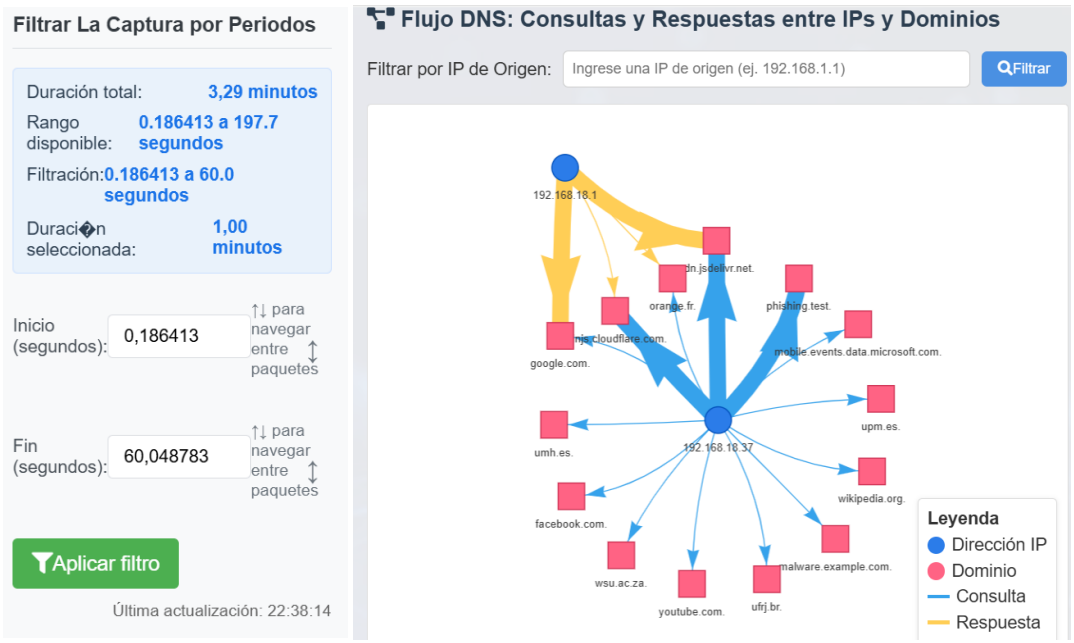

Figura 52:Filtro temporal en gráfico de flujo DNS.

- **Resultados: Consulta de Dominios**

Esta tabla nos permite visualizar los dominios más y menos consultados junto con un formulario que nos admite consultar un número predefinido de consultas. Esta métrica tiene como utilidad saber qué dominios han sido más o menos consultados en la red. Esto ayuda a detectar actividad sospechosa o maliciosa y observar qué páginas o servicios se usan más.

Para visualizar el procesamiento de esta tabla, tomamos como ejemplo la tabla de la captura de primer usuario poniendo como muestra los diez dominios más consultados.

Figura 53: Dominios más consultados-Usuario 1.

Para visualizar las solicitudes menos consultadas, cogemos como ejemplo la tabla del segundo usuario consultando las diez consultas menos visitadas.

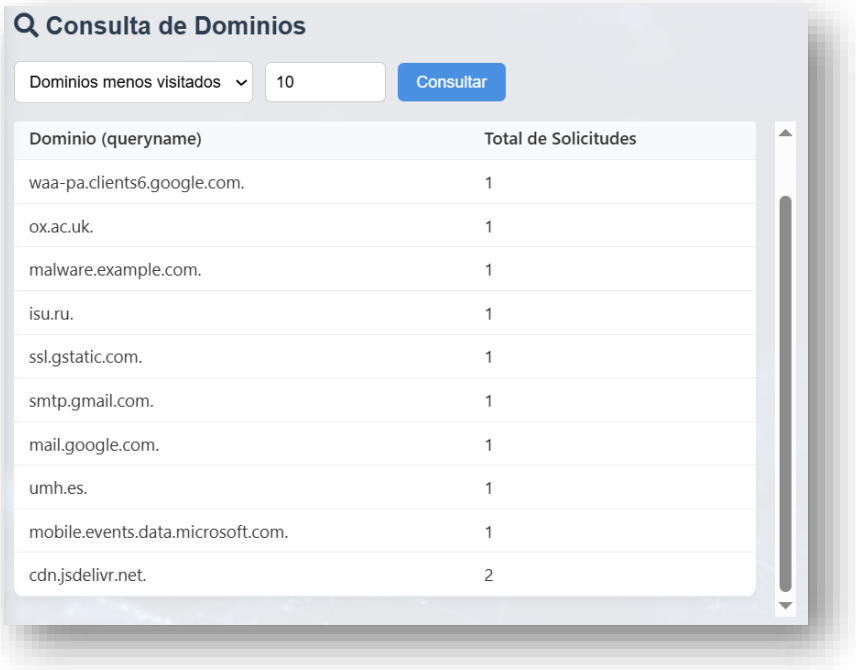

Figura 54: Dominios menos consultados – Usuario 2.

A continuación, observamos la realización del filtro temporal sobre la tabla del primero usuario, tomamos como rango el primer minuto de la captura luego analizamos.

Notamos, por ejemplo, que en este periodo se han registrado dos solicitudes con el nombre Google.com, en comparación con nueve en la captura total.

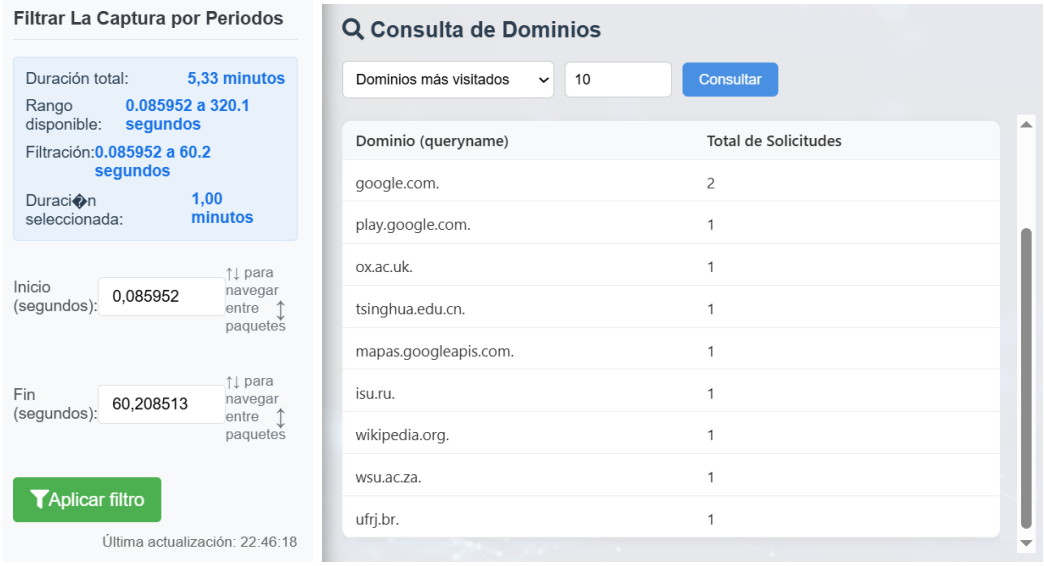

Figura 55: Filtro temporal en consultas DNS.

- **Resultados: Respuesta de Dominios**

La tabla facilita notar qué dominios tienen más y menos respuestas. Esta métrica es útil para identificar cuáles dominios obtienen más o menos respuestas. Asimismo, ayuda a notar si hay algo inusual en la red. Conocer los servicios que se utilizan más. Igualmente, contribuye a optimizar el control y la eficiencia de la red.

La figura siguiente permite visualizar el diseño de la tabla, tomamos como ejemplo la tabla del primer usuario consultando los diez dominios con más números de respuestas.

Observamos que en la captura entera hay únicamente cuatro dominios que tiene respuestas DNS.

Figura 56: Dominios con más respuestas – Usuario 1.

Para observar los dominios con menor número de respuestas, tomamos como ejemplo la tabla de segundo usuario consultando los diez dominios con menos número de respuestas.

En este caso, notamos que la mayoría de los dominios tienen una respuesta DNS.

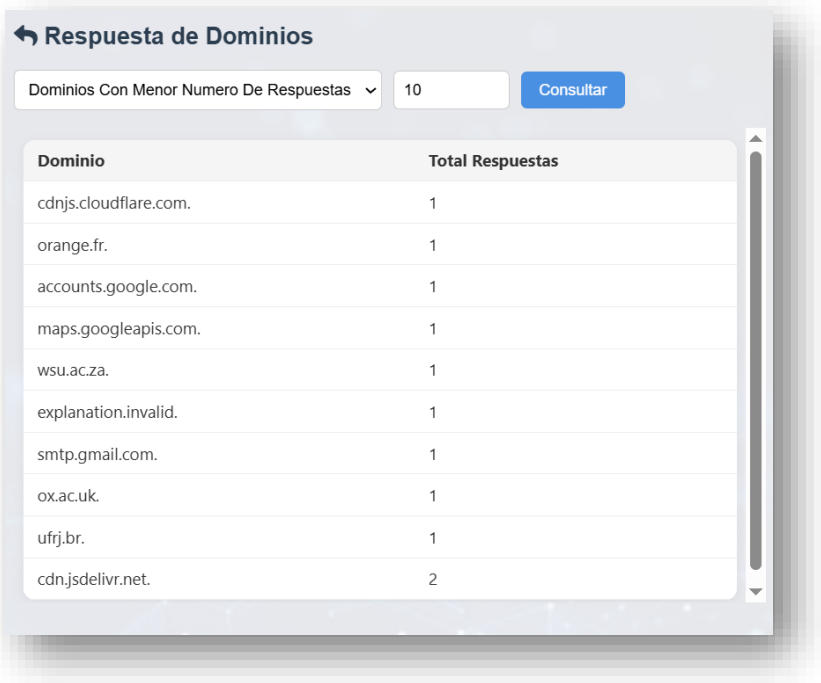

Figura 57: Dominios con menos respuestas – Usuario 2.

En esta tabla, se aplica el filtro temporal para visualizar su aplicación en la tabla de segundo usuario, tomamos como ejemplo los dos primeros minutos de la captura y analizamos. Observamos que este periodo se han registrado solamente siete dominios.

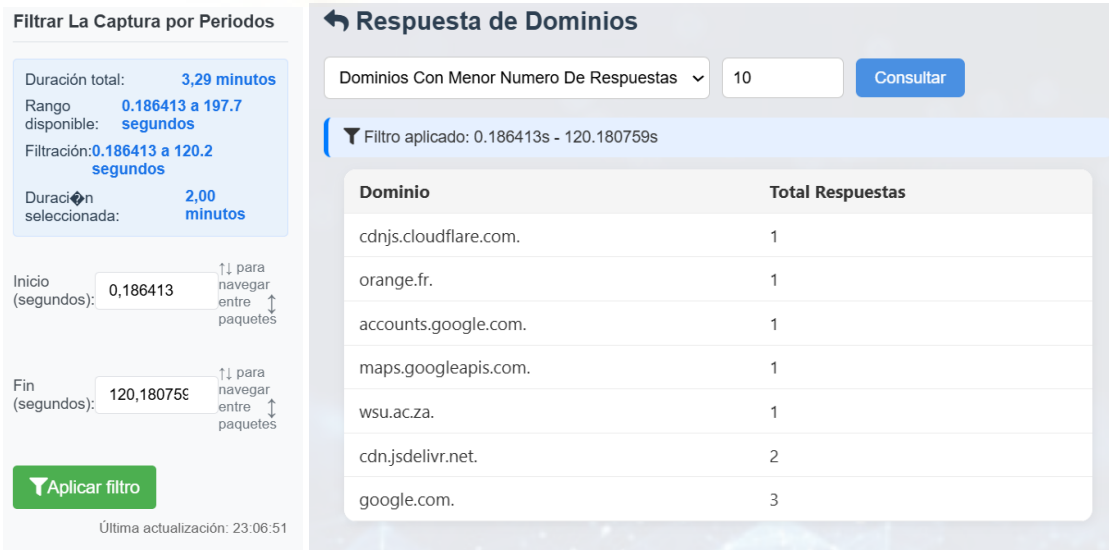

Figura 58:Filtro temporal en respuestas DNS.

- **Resultados: Comparación de Dominios DNS**

Este gráfico permite comparar las solicitudes y las respuestas de dominios según los tipos de registros DNS. Esta métrica tiene varias utilidades importantes para el análisis de tráfico de red, muestra qué tipos de registros DNS son más frecuentes en la red (A, MX, CNAME, etc.), también, ayuda a identificar patrones de uso de nombre de dominios DNS, asimismo, identifica qué tipos de registros consumen más recursos.

La figura siguiente es una muestra de una captura del primer usuario que admite la visualización del procesamiento de datos tomando como ejemplo los tres dominios más visitados en la red para compararlos. Asimismo, primero comparamos todos los tipos de registros DNS.

Observamos que en este caso existen nueve solicitudes del primer dominio, cinco solitudes de cada dominio 2 e 3 y una única respuesta de ultimo dominio.

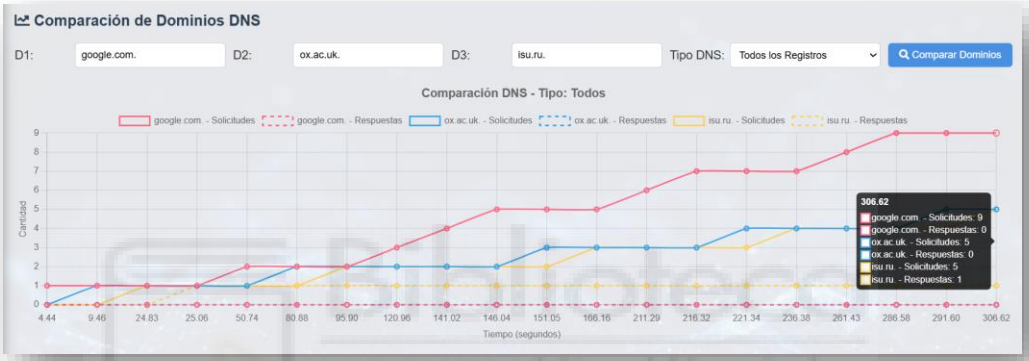

Figura 59: Comparativa de dominios DNS – Usuario 1.

Para comprar un tipo de registro DNS, elegimos en el formulario el tipo de registro requerido y comparamos las consultas y las respuestas DNS de esos tres dominios. Observamos que, en este caso, hay únicamente tres solicitudes del primer dominio que son de tipo registro A, lo demás dominios, notamos que todas las solicitudes son de tipo A.

Figura 60: Comparativa de tipo A en tres dominios – Usuario 1.

A continuación, visualizamos la aplicación del filtro temporal en la captura del segundo usuario, tomamos como ejemplo el primer minuto de la captura. Observamos que se ha capturado una única solicitud para cada dominio seleccionado y dos respuestas para el primer dominio seleccionado.

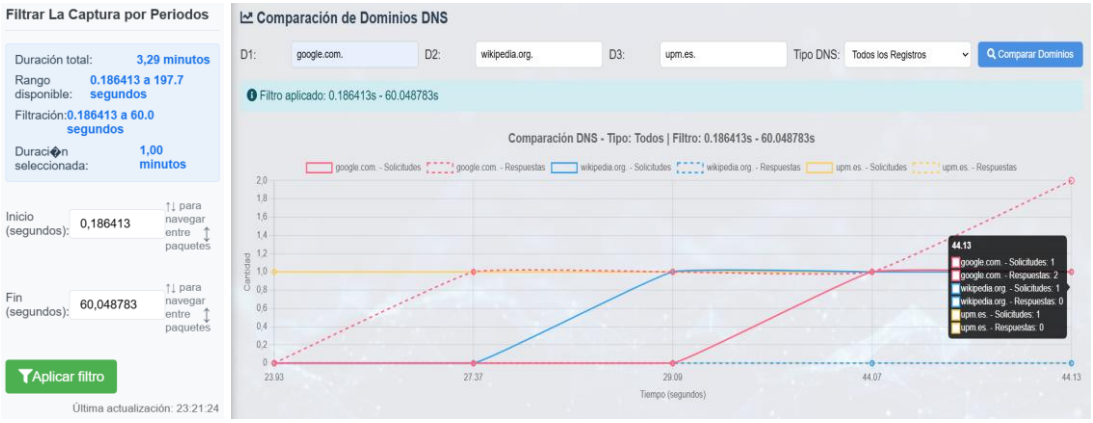

Figura 61: Filtro temporal en comparación de dominios DNS.

En este mismo caso, aplicando el filtro temporal y eligiendo el tipo de registro A en el filtro de registros DNS, notamos que en este periodo únicamente el primer dominio que tiene solicitudes y respuestas.

Figura 62: Filtro temporal y por tipo en comparación de registros DNS.

5.2 Discusión de los resultados

La herramienta que creamos hace más fácil observar al instante el comportamiento del tráfico de internet mientras se está capturando información. Como la base de datos se vacía antes de cada nuevo análisis, los resultados reflejan únicamente lo que está ocurriendo en ese momento, ayudando a concentrar el estudio en el estado actual de la red. Después, se explica los resultados obtenidos en estos gráficos.

Distribución de paquetes por protocolo

Este gráfico ofrece una visión clara de la distribución del tráfico entre los distintos protocolos identificados. A pesar de que el sistema no conserva información anterior, este gráfico nos permite observar de manera ágil cuál protocolo es el más utilizado en ese momento, lo que facilita identificar si el tráfico proviene principalmente de sitios web o nombres de dominios DNS o otros servicios.

En los casos implementados anteriormente, notamos un alto volumen de tráfico HTTPS en la captura de los dos usuarios generales, esto indica una navegación activa por sitios web cifrados.

Volumen de Tráfico Acumulativo

En este gráfico, se observa cómo cambia el volumen de tráfico con el tiempo nos deja notar la cantidad de paquetes que se generan mientras capturamos datos. La línea es siempre acumulativa porque en cada intervalo de tiempo se añaden nuevos paquetes. Cuando alcanza un umbral predefinido se envía una alerta al correo electrónico haciendo más sencillo el manejo del tráfico sin necesidad de estar observando todo el tiempo.

En los resultados obtenidos de los casos mencionados antes, observamos que hay una variación en el tiempo necesario para alcanzar el umbral. En la primera captura de usuario, se logra el umbral establecido en 2,26 minutos, mientras que en la segunda captura se alcanza en 4,25 minutos. Esta diferencia se debe a que la cantidad de tráfico o la velocidad con la que se recolectan los paquetes cambian según la variabilidad en la actividad de la red.

Tráfico DNS: Evolución de Totales vs. Dominio Filtrado

Este gráfico compara el tráfico DNS general con el de un dominio específico que elija el usuario. Así, podemos fijarnos en un dominio concreto dentro de todo el tráfico, lo que es útil para estudiar cómo se comporta en un momento dado, como la frecuencia con la que se consulta o su relación con ciertas actividades en la red.

Según el caso explorado en los resultados obtenidas, al examinar la conexión entre las solicitudes y las respuestas del DNS, se nota que el usuario 1 emitió 55 solicitudes, obteniendo solamente 10 respuestas. Esto indica una tasa de resolución baja, posiblemente atribuible a que algunos dominios están inactivos, hay bloqueos o se han excedido los tiempos de espera. Por otro lado, el Usuario 2, quien realizó 49 solicitudes y recibió 22 respuestas, muestra una eficacia superior en la resolución. Esto podría relacionarse con un tráfico más constante o dominios que se encuentran activos. Esta discrepancia refleja las variaciones en el comportamiento del tráfico de internet a lo largo de las sesiones.

Distribución Total de Tipos de Registros DNS

En este gráfico circular, notamos la proporción de los diferentes tipos de registros DNS que se capturan (por ejemplo, A, AAAA, MX, TXT, etc.). Esta información es clave para entender el tipo de actividades en curso, como navegar (A), enviar correos (MX) o hacer validaciones (TXT), etc.

En ambos casos, el dominio del registro A muestra un tráfico que se enfoca en convertir nombres en direcciones IPv4, algo típico durante la navegación por internet. La baja cantidad de registros TXT y CNAME indica que no se están realizando consultas que impliquen configuraciones adicionales, como la verificación de dominios o la creación de alias, lo que sugiere un uso elemental de los servicios de DNS. Sin embargo, la falta de estos registros en la captura del primer usuario normal y su presencia en la captura del segundo usuario normal muestra una variación en los tipos de servicios utilizados: el usuario 2 parece haber tenido acceso a dominios que requieren configuraciones más complejas, en contraste con el usuario 1, quien mostró una actividad más restringida.

Flujo DNS: Consultas y Respuestas entre IPs y Dominios

Este gráfico de dispersión nos permite observar cómo se relacionan las direcciones IP de origen con los dominios que consultan, incluyendo las respuestas. Este tipo de vista es muy útil para encontrar nodos activos, descubrir dominios que generan mucho tráfico DNS.

En los casos explorados en el apartado de resultados obtenidos, el gráfico de dispersión proporciona una manera clara de observar las conexiones entre las direcciones IP y los dominios solicitados durante la captura. En la sesión del primer usuario, el tráfico DNS muestra un patrón de navegación habitual, donde las solicitudes surgen de la dirección IP del dispositivo y las respuestas llegan directamente de los sitios web consultados. Sin embargo, en la sesión del segundo usuario, se detecta un nodo extra que corresponde al servidor DNS público 8. 8. 8. 8, lo que señala que se utiliza un resolutor externo para hacer las consultas. También se han registrado solicitudes hacia dominios clasificados como maliciosos, lo que indica un tráfico más variado y posiblemente más peligroso, ofreciendo métodos ventajosos para actividades de análisis de seguridad.

Consulta de Dominios

La tabla de dominios más y menos buscados ayuda a reconocer rápidamente cuáles fueron los sitios más frecuentemente consultados durante la captura. Los dominios que se consultan con mayor frecuencia suelen estar ligados a servicios populares o plataformas que funcionan en segundo plano, como actualizaciones del sistema, servicios de sincronización o la navegación web cotidiana. Por otro lado, los dominios que reciben menos consultas se relacionan con solicitudes específicas, que pueden haberse generado por acciones del usuario en un momento determinado o por servicios que se utilizan de forma menos habitual. Esta diferencia entre alta y baja frecuencia ofrece una visión clara del comportamiento de la red, facilita la identificación de patrones comunes y permite detectar, si es necesario, dominios que podrían ser sospechosos o que no se ajustan al comportamiento esperado.

Respuesta de Dominios

La tabla que muestra los dominios con mayor y menor número de respuestas refleja la efectividad y el funcionamiento del sistema DNS durante el tiempo de monitoreo. Es común observar que la cantidad de respuestas es menor que la de solicitudes, ya que no todas las consultas generan una respuesta rápida; en algunas ocasiones, no se pueden resolver o están bloqueadas. Los dominios que reciben más respuestas indican servicios o sitios activos y accesibles que suelen responder a las consultas, mientras que los dominios con menos respuestas podrían estar relacionados con objetivos inactivos, problemas para resolver o solicitudes a dominios menos comunes. Este análisis ayuda a comprender el comportamiento del tráfico DNS y a identificar posibles irregularidades o inconvenientes en la resolución de nombres.

Comparación de Dominios DNS

Este gráfico permite estudiar el rendimiento de hasta tres dominios al mismo tiempo, lo que facilita notar cómo cambian sus registros DNS y cuán a menudo aparecen en el tráfico que se captura. Además, da la oportunidad de notar variaciones en el comportamiento de los dominios dentro de una única captura, lo cual es útil para auditorías, análisis de comportamientos o investigaciones sobre tráfico específico.

Al analizar los datos presentados, se puede observar que, en la mayoría de las situaciones, los dominios tienen más solicitudes que respuestas. Por ejemplo, hay dominios que cuentan con hasta cinco solicitudes sin recibir ninguna respuesta, lo que podría indicar inconvenientes en la resolución, bloqueos temporales o simplemente que el servidor no contestó durante el período de captura. Además, se destaca un caso particular en la captura del segundo usuario, donde un dominio muestra una sola solicitud, pero recibe dos respuestas. Este comportamiento, puede ser el resultado de reintentos automáticos del sistema, redirecciones o una configuración especial en el servidor DNS que envía varias respuestas a una única consulta.

Las diferencias entre solicitudes y respuestas permiten identificar irregularidades en la resolución de nombres y pueden ser indicios de problemas de conectividad, políticas de filtrado o incluso actividad sospechosa. De este modo, el gráfico no solo ilustra el volumen de actividad por dominio, sino que también brinda indicios sobre la calidad y la continuidad del servicio DNS relacionado con cada uno.

6. CONCLUSIONES Y TRABAJO FUTURO

6.1 Conclusiones

A lo largo de este proyecto, hemos conseguido poner en marcha una solución muy útil para registrar, manejar y seguir de cerca el tráfico de datos en internet al instante, sobre todo el de nombre de dominios DNS. Durante las pruebas, hemos verificado que se pueden controlar bien las funciones para empezar a capturar, parar y limpiar la base de datos, lo que hace más fácil usar la herramienta desde el navegador.

Además, todos los gráficos integrados en la herramienta funcionan sin problemas y presentan la información de manera clara y fácil de entender. El gráfico de protocolos detectados ayuda a notar cuál es el protocolo más usado en el registro; el gráfico de volumen de tráfico en tiempo real indica cuántos paquetes se registran a cada momento y avisa automáticamente si se supera un límite fijado; el gráfico de distribución de paquetes enseña la localización geográfica el tráfico registrado; La herramienta también tiene un gráfico de peticiones contra respuestas DNS, que no solo ayuda a encontrar posibles fallos de resolución, sino que además permite comparar los totales con los de un dominio concreto. Aparte, el gráfico de tipos de registros DNS más habituales muestra qué tipo de registros DNS son los más comunes; el gráfico de dispersión que no solo enseña la relación petición-respuesta entre las direcciones IP de origen y los dominios DNS, sino que también destaca la aparición de nodos importantes como servidores DNS externos (por ejemplo, 8. 8. 8. 8) o dominios que podrían ser un riesgo. También, las tablas que enseñan los dominios más y menos visitados y los dominios con más o menos respuestas. Por último, el gráfico de comparación de dominios DNS, que hace más fácil analizar hasta tres dominios a la vez. Todos estos gráficos permiten dar una visión completa y en tiempo real de cómo se comporta el tráfico de internet.

Además, hemos comprobado que los filtros de tiempo permiten observar correctamente los datos en todos los gráficos, lo que facilita el análisis de momentos concretos. Del mismo modo, el filtro que se basa en la dirección IP de origen, que está en algunos gráficos, funciona como debe y permite centrarse en el tráfico de una dirección IP en particular.

En general, la herramienta ha sido un éxito, ya que cumple con sus objetivos desde el comienzo del proyecto. Observar y analizar el comportamiento de red ha demostrado ser ventajoso en entornos educativos, o puede servir como base para diseñar sistemas avanzados que controlen el tráfico de internet.

6.2 Limitaciones

Durante el desarrollo de este trabajo, enfrentamos ciertas limitaciones que afectan la eficacia y la experiencia de usuario. Estas limitaciones se dividen en dos categorías esenciales: técnicas y de usabilidad.

- **Limitación técnica**

  La herramienta se probó en un entorno controlado, que implicaba capturar el tráfico de internet de un usuario. En estas circunstancias, el sistema funcionó correctamente y sin problemas, a pesar de que no se ha llevado a los extremos con grandes cantidades de información.

  La ausencia de datos para determinar si el sistema mantiene la velocidad o no se debe a la falta de pruebas exhaustivas para analizar millones de paquetes. Con eso y todo, la herramienta cumple con el objetivo principal, que era observar su desempeño y analizar visualmente el tráfico en tiempo real, más que procesar datos masivos.

- **Limitación de usabilidad**

  El diseño de la plataforma está diseñado específicamente para su uso en computadoras. Por el momento, no es viable acceder a la plataforma utilizando teléfonos celulares o tabletas como una aplicación adecuada.

  Este inconveniente dificulta su uso, ya que no permite acceder a la herramienta desde dispositivos móviles, limitando así la herramienta a su uso exclusivo en computadoras. Sin embargo, esta decisión no afecta el objetivo inicial del proyecto, que es capturar, almacenar y presentar el tráfico internet en un sitio web diseñado específicamente para ordenadores.

6.3 Mejoras Futuras

La herramienta desarrollada ha alcanzado sus objetivos, pero hay partes donde puede mejorar sus características e interactuar mejor con los usuarios.

Actualmente, la herramienta se ha probado con una cantidad limitada de tráfico, que aunque supera el millar de paquetes es suficiente para confirmar su funcionamiento fundamental. A pesar de esto, para un mejor rendimiento en las redes con mayor actividad, es beneficioso ajustar la plataforma para manejar muchos más datos, y también para aumentar su capacidad de crecer. Esto podría implicar mejoras en la base de datos, un mejor manejo de paquetes y tareas, y métodos para manejar grandes datos de manera efectiva.

Además, el diseño visual de la plataforma se adapta para computadoras, y no funciona con teléfonos inteligentes o tabletas. Con más personas que usan estos dispositivos para usar aplicaciones web, se recomienda crear una versión flexible del sitio que proporcione una buena experiencia en pantallas más pequeñas. Esta mejora permitiría un uso más cómodo de la herramienta en diferentes dispositivos, incrementando su funcionalidad y área de influencia.

Además, la opción de agregar otros gráficos y elementos visuales que muestren datos importantes sobre el comportamiento el tráfico en internet. Por ejemplo, se podrían ofrecer comparaciones visuales entre diferentes usuarios, presentar otros datos más relevantes, o usar elementos interactivos para mejorar el análisis.

Del mismo modo, añadir la opción de guardar datos, gráficos o resúmenes en formatos estándar como CSV, PDF facilitaría la revisión, registrar y presentar los resultados sin acceso a internet.

Entonces, estas mejoras aumentarían la usabilidad y la facilidad de acceso de la herramienta, convirtiéndola en una alternativa más segura y adaptable para diferentes contextos de análisis del tráfico de internet.

7. BIBLIOGRAFÍA

[1] Wireshark:

Wikipedia. (2025b, abril 27). Wireshark. Wikipedia. Consultado el 15 de abril de 2025, de https://es.wikipedia.org/wiki/Wireshark

[2] Ntopng

Albert, J. (2020, Junio 29). Ntopng: Un excelente monitor de trafico de red de nueva generación. Desde Linux. Consultado el 15 de abril de 2025, de https://blog.desdelinux.net/ntopng-excelente-monitor-trafico-red-nueva-generacion/

[3] PRTG

Powell, Z. (2024, Mayo 14). ¿Qué es PRTG? Consultado el 15 de abril de 2025, de https://geekflare.com/es/software/networking/what-is-prtg/

[4] Scapy

Wikipedia. (2024, Noviembre 5). Scapy. Wikipedia. Consultado el 15 de abril de 2025, de https://en.wikipedia.org/wiki/Scapy

[5] MYSQL

Arsys. ¿Qué es MySQL? Explicación y características. Consultado 16 de abril de 2025, de https://www.arsys.es/blog/mysql

[6] Microsoft Visual Studio

Wikipedia. (2025, Mayo 15). Microsoft Visual Studio. Wikipedia. Consultado el 16 de abril de 2025, de https://es.wikipedia.org/wiki/Microsoft_Visual_Studio

[7] XAMPP

Moreno, O. (2024, Junio 12). XAMPP: Guía para las empresas. Consultado el 18 de abril de 2025, de https://aodatacloud.es/blog/xampp-guia-para-las-empresas/

[8] Python

AWS. (n.d.). Amazon Web Services. ¿Qué es Python? Consultado el 18 de abril de 2025, de https://aws.amazon.com/es/what-is/python/

[9] PHP

De Souza, I. (2020, Marzo 9). Descubre qué es el lenguaje de programación PHP y en qué situaciones se hace útil. Consultado el 20 de abril de 2025, de https://rockcontent.com/es/blog/php/

[10] HTML

Vadavo. (2024, Noviembre 18). ▷ HTML: Qué es y para qué sirve. Consultado el 20 de abril de 2025, de https://www.vadavo.com/blog/html-que-es-y-para-que-sirve/

[11] JavaScript

AWS. (n.d.). Amazon Web Services. ¿Qué es JavaScript? Consultado el 20 de abril de 2025, de https://aws.amazon.com/es/what-is/javascript/

[12] CSS

Lenguaje CSS. (s.f.). ¿Qué es CSS? Consultado el 22 de abril de 2025, de https://lenguajecss.com/css/introduccion/que-es-css/

[13] Npcap

NPCap. (2024, 6 febrero). Dr. Edu. Consultado el 23 de abril 2025, de https://dredu.mx/principal/intereses/obsesiones/informatica-y-computacion/seguridad-informatica/npcap/

[14] MySQL Connector/Python

MySQL :: MySQL Connector/Python Developer Guide. (n.d.). Consultado el 10 de Marzo de 2025, de https://dev.mysql.com/doc/connector-python/en/

[15] Documentación de Scapy

Welcome to Scapy's documentation! — Scapy 2.6.1 documentation. (n.d.). Consultado el 5 de Marzo de 2025, de https://scapy.readthedocs.io/en/latest/

[16] GeoIP

MaxMind GeoIP2 Python API — geoip2 5.1.0 documentation. (n.d.). Consultado el 12 de Marzo de 2025, https://geoip2.readthedocs.io/en/latest/

[17] Manejo de archivos

os — Miscellaneous operating system interfaces. (n.d.). Python Documentation. Consultado el 14 de Marzo de 2025 https://docs.python.org/3/library/os.html

[18] Dnspython

Vay3t. (2023, 4 agosto). Hacking con Python 3: Capitulo 13 — DNS y direcciones IP. Medium. Consultado el 20 de Marzo de 2025, de https://vay3t.medium.com/hacking-con-python-3-capitulo-13-dns-y-direcciones-ip-32907d4319d6

[19] Commando schtasks

Window Server (2025, 25 Marzo). schtasks run. Microsoft Learn. Consultado el 20 de abril de 2025, de https://learn.microsoft.com/es-es/windows-server/administration/windows-commands/schtasks-run

[20] Guia PHPMailer

Equipo editorial de IONOS. (2024, 12 marzo). Cómo enviar correos electrónicos con PHPMailer. IONOS Digital Guide. Consultado 23 de abril de 2025, de https://www.ionos.es/digitalguide/correo-electronico/cuestiones-tecnicas/phpmailer

[21] Manual PHP

PHP Group. *Manual de PHP* [en línea]. [s.l.]: PHP.net, s.f. Consultado: 18 de febrero de 2025, de https://www.php.net/manual/es/index.php

[22] Manual JavaScript

JavaScript | MDN. (s. f.). MDN Web Docs. Consultado 18 de febrero de 2025, de https://developer.mozilla.org/es/docs/Web/JavaScript

[23] Manual HTML

Manual web. (2023, 17 diciembre). Introducción HTML. Manual Web. Consultado el 20 de febrero de 2025, de https://www.manualweb.net/html/introduccion-html/

[24] Manual CSS

CSS | MDN. (s. f.). MDN Web Docs. Consultado el 20 de febrero de 2025, de https://developer.mozilla.org/es/docs/Web/CSS