



**C.P.R. Liceo “La Paz”
Proyecto fin de ciclo**

Análisis de Red con Wireshark y Qlik Cloud

Administración de Sistemas Informáticos en Red

Autor : Gabriel Enzo Barela Forselini

Tutor : Xabier Pérez Maestre

Resumen

El objetivo de este proyecto es, mediante el uso de la herramienta de análisis de redes Wireshark y la herramienta de procesamiento y visualización de datos Qlik Cloud, capturar, filtrar, organizar, corregir y analizar una serie de diferentes tipos de conexiones a la red (HTTP, HTTPS, descarga de archivos, streaming, correo electrónico y multimedia).

Este estudio se ha llevará a cabo a lo largo de un periodo de tiempo determinado, repitiendo en horario y forma el proceso de captura de cada tipo de conexión. Posteriormente se procederá a convertir y formatear los datos a un formato legible para Qlik Cloud y se desarrollará una estructura para la base de datos de la herramienta de análisis sobre la que se ejecutarán los distintos tipos de operaciones y visualizaciones relevantes de la información.

El objetivo principal es otorgar una visión detallada y comprensible de las características y patrones del tráfico de red asociados a cada tipo de conexión a través de una herramienta de análisis como Qlik. Desmostando así su versatilidad, su capacidad de llevar a cabo análisis profundos y detallados, así como su utilidad de cara a ofrecer al usuario un estudio interesante a la vez que intuitivo y de fácil manejo para el estudio de la red.

Una vez obtenidos los resultados, se proporciona una comprensión profunda de cómo diferentes actividades de red pueden afectar al rendimiento y la seguridad. Además de presentar un análisis comparativo que destaca las particularidades y demandas de cada tipo de conexión, protocolo, IPs y sus clases, puertos, etc. aportando perspectivas interesantes para la posible futura optimización y gestión de redes informáticas.

Abstract

The objective of this project is to utilize the network analysis tool Wireshark and the data processing and visualization tool Qlik Cloud to capture, filter, organize, correct, and analyze a series of different types of network connections (HTTP, HTTPS, file downloads, streaming, email, and multimedia). This study will be conducted over a period of time, repeating the capture process for each type of connection in a consistent manner. Subsequently, the data will be converted and formatted into a readable format for Qlik Cloud, and a database structure for the analysis tool will be developed to execute various relevant operations and visualizations of the information.

The primary aim is to provide a detailed and comprehensible insight into the characteristics and patterns of network traffic associated with each type of connection through an analytical tool like Qlik. This demonstrates its versatility, its ability to conduct deep and detailed analyses, as well as its usefulness in offering the user an interesting, intuitive, and user-friendly study.

Upon obtaining the results, a profound understanding is provided of how different network activities can affect performance and security. Additionally, a comparative analysis is presented that highlights the peculiarities and demands of each type of connection, protocol, IPs and their classes, ports, etc., providing valuable perspectives for the optimization and management of computer networks.

Palabras clave

Qlik Cloud ⁽¹⁾: Qlik Cloud es una plataforma en la nube de análisis e integración de datos creada para la Inteligencia activa. Ofrece servicios de análisis e integración de datos que se pueden usar juntos o de forma independiente.

Wireshark ⁽²⁾: Wireshark es un analizador de paquetes de red que presenta los datos de paquetes capturados con el mayor detalle posible.

Analizador de red: Un analizador de tráfico de red es una herramienta que está específicamente diseñada para capturar, analizar y evaluar el tráfico de datos que circula en una red informática.

Análisis de datos ⁽³⁾: El análisis de datos convierte datos sin procesar en información práctica. Incluye una serie de herramientas, tecnologías y procesos para encontrar tendencias y resolver problemas mediante datos.

Visualización de datos ⁽⁴⁾: La visualización de datos es el proceso de utilizar elementos visuales como gráficos o mapas para representar datos. De esta manera, se trasladan datos complejos, de alto volumen o numéricos a una representación visual más fácil de procesar. Las herramientas de visualización de datos mejoran y automatizan el proceso de comunicación visual para lograr precisión y detalle. Puede utilizar las representaciones visuales para extraer información práctica a partir de datos sin procesar.

Contenido

Introducción.....	3
Objetivos.....	7
Estado del arte.....	11
Caso de estudio.....	15
Desarrollo del proyecto.....	19
Conclusiones.....	51
Líneas abiertas de investigación.....	55
Referencias.....	58

Introducción

Introducción

La seguridad de la red constituye un pilar fundamental en el ámbito de las tecnologías de la información y las comunicaciones, garantizando la integridad, la confidencialidad y la disponibilidad de los datos. En un contexto de creciente interconectividad y sofisticación de las amenazas cibernéticas, resulta imprescindible implementar y mantener robustas estrategias de protección para salvaguardar la infraestructura digital de la empresa o la red pública.

En el mismo sentido, la alta disponibilidad y la eficiencia de la red son componentes esenciales para el funcionamiento óptimo de los sistemas informáticos. La alta disponibilidad asegura que los servicios y aplicaciones estén accesibles de manera ininterrumpida, minimizando tiempos de inactividad y mejorando la resiliencia ante fallos. Por otro lado, la eficiencia de la red optimiza el uso de recursos, garantizando un rendimiento adecuado y una rápida respuesta a las demandas del usuario.

Así pues, el conjunto de ambas, seguridad y alta disponibilidad, son un requisito primordial a la hora de conformar cualquier estructura de red. Siendo su implementación, supervisión y solución algo muy demandado en el mercado actual.

Una de las partes fundamentales de dicho proceso es el de la monitorización y análisis de la red. Para llevarlo a cabo existen multitud de herramientas interesantes de cara la captura de los paquetes que circulan por la red (Wireshark, tcpdump...), el escaneo de puertos (nmap) o el sistema de detección de intrusos en red (snort), etc.

Siguiendo esto, se podría decir que Wireshark es la herramienta más potente, versátil y usada en el mercado en cuanto al análisis del tráfico de la red. Por ello es que ha sido seleccionada para el desarrollo que se propone.

Wireshark, anteriormente conocido como *Ethereal*, es un analizador de paquetes de red que captura toda la información que transita a través de una conexión. Esta herramienta se utiliza para realizar análisis y solucionar problemas en redes de comunicaciones, llevar a cabo auditorías de seguridad, desarrollar software y protocolos, entre otras cosas. Es gratuito y de código abierto y además, permite examinar datos de una red activa o de un archivo de captura guardado en el disco. También cuenta con un amplio lenguaje de filtrado y análisis propio que permite la selección de los paquetes o de las conversaciones que puedan interesar en un determinado momento.

Por otro lado, también se encuentra una rama de la informática dedicada al análisis de la información o el análisis de datos. Este campo aborda los datos obtenidos y registrados (por ejemplo guardándolos en una base de datos) en un sistema para su posterior manejo y uso. A diferencia de una base de datos convencional (relacional), su

objetivo no es el de meramente acceder al dato concreto y extraerlo, sino que tiene como perspectiva la de analizar toda la información en su conjunto y de manera asociativa y/o comparativa. Esto no quiere decir que no pueda, y que de facto no lo haga, extraer datos concretos dentro del conjunto almacenado.

Existen diversos tipos de análisis de datos que se suelen dividir según el tamaño o cantidad de los datos, es decir, análisis de datos locales o localizables: son aquellos cuyos datos están disponibles en una ruta o serie de rutas concretas y tienen relación directa entre sí, pueden estar en la nube o en local y, a pesar de que habitualmente cuenten con una enorme cantidad de datos, dicha cantidad es mucho menor que el otro tipo de análisis que se menciona, este es, el Big Data; que se caracteriza por provenir de multitud de fuentes y localizaciones así como de no estar necesariamente relacionada la información, además, por supuesto, de conformar una cantidad ingente de información.

Relativo a esto, se puede encontrar diversas herramientas de análisis de datos en el mercado especializadas en cada tipo de análisis (Qlik, PowerBI, Tableau, Apache Spark, etc).

Una de ellas es la tecnología Qlik (Qlik Sense, Qlik Cloud y Qlik View). Su concepción fundamental está orientada al análisis de datos empresariales relacionables, pero también cuenta con toda una serie de implementaciones dedicadas al campo del análisis Big Data y la automatización de procesos de análisis. Cuenta con un lenguaje propio (lenguaje qlik) basado en diversos lenguaje en función del aspecto a trabajar: un derivado de SQL y JSON para la gestión y carga de los datos, lenguajes de hojas de cálculo y de análisis de datos como Python, R o MATLAB, así como CSS para el diseño del frontend de la aplicación. La diferencia principal de Qlik Cloud a respecto de las otras dos herramientas de Qlik es que la primera está pensada para un pleno desarrollo en la nube, en vez de en local.

Sus funcionalidades básicas son: depurar, filtrar, organizar, analizar (operar con los datos) y visualizar los datos de un modo intuitivo y práctico. Esto último no solo está pensado para otorgarle facilidad en el trabajo al desarrollador, sino también para que el usuario final pueda acceder a los datos de forma cómoda y accesible a todos los públicos, es decir, en unos pocos clics; de ahí su nombre.

En este caso concreto se usará exclusivamente en su aspecto de análisis de menor nivel, aunque ello indique precisamente la posibilidad de escalar el mismo proceso a un nivel mucho mayor de carga.

En definitiva, lo que se pretende es que mediante la combinación de ambas facetas: el análisis de red con Wireshark y el análisis de datos con Qlik, procesar y analizar los datos relevantes de una red para su interpretación y valoración final.

Objetivos

Objetivos

El objetivo principal de este proyecto es demostrar la utilidad práctica de la combinación de las herramientas Wireshark y Qlik Cloud en un entorno empresarial o cualquier estructura de red de cara un análisis de red. Esto implica mostrar cómo estas herramientas pueden ser utilizadas para mejorar la gestión y el rendimiento de la red en entornos reales, mediante la creación de informes y dashboards que proporcionen información valiosa y fácilmente interpretable. La versatilidad y escalabilidad de Qlik Cloud también serán destacadas, mostrando su capacidad para realizar análisis detallados y adaptarse a las necesidades de redes de diferentes tamaños y complejidades.

Otro aspecto importante del proyecto es resaltar la simplicidad y rapidez del proceso gracias a la accesibilidad de las herramientas utilizadas. Wireshark, siendo una herramienta gratuita, y Qlik Cloud, con su enfoque flexible y poderoso para el análisis de datos, hacen posible que cualquier organización, independientemente de su tamaño o presupuesto, pueda implementar estos análisis. La facilidad de uso de estas herramientas permitirá demostrar que el proceso de captura y análisis de datos de red puede ser implementado y automatizado de manera eficiente, sin necesidad de una inversión significativa en tiempo o recursos técnicos.

Además, se busca proporcionar una visión clara y detallada de las características y patrones del tráfico de red asociados a cada tipo de conexión. Este análisis permitirá no solo comprender mejor la cantidad de datos transferidos, la latencia y el uso del ancho de banda, los protocolos y puertos más usados, etc. en función del tipo de conexión a la red; sino también identificar posibles problemas y dificultades como cuellos de botella, pérdidas de paquetes y retrasos, lo cual es crucial para mejorar tanto el rendimiento como la seguridad de la red.

Así pues, el proyecto incluirá un análisis comparativo que resalte las peculiaridades y demandas de cada tipo de conexión, protocolo, IPs y puertos. Este análisis proporcionará una comprensión profunda de cómo diferentes actividades de red pueden afectar al rendimiento y la seguridad, identificando posibles vulnerabilidades y riesgos de seguridad.

En resumen, el proyecto tiene como objetivo demostrar la utilidad práctica de las herramientas empleadas, proporcionar un análisis detallado y comprensible del tráfico de red, identificar problemas y dificultades de su implementación, y subrayar la simplicidad y rapidez del proceso, todo con el fin de optimizar y gestionar de manera más efectiva cualquier red informática.

No se trata en ningún caso de aportar soluciones específicas a posibles problemas concretos o reales relativos a la seguridad de las conexiones o la alta disponibilidad.

Estado del arte

Estado del arte

Existe en la actualidad diversidad de herramientas con capacidades de análisis de red combinado con representación gráfica.

Entre ellas podemos encontrar a *Datadog Network Performance Monitoring*, que se podría decir que es una de las más semejantes a lo propuesto en esta investigación: representaciones gráficas interactivas sobre el tráfico de red. En ella es posible modificar algunos gráficos y seleccionar algunas dimensiones de interés para luego exportarlos.

También podemos encontrar a *Nagios XI*, un software libre dedicado al monitoreo y visualización de redes. Otras herramientas puede ser *ManageEngine OpManager*, *PRTG Network Monitor* o *SolarWinds Network Performance Monitor*, similares al anterior pero su caso de pago.

La diferencia de todas estas herramientas a respecto de la implementación propuesta en este proyecto es que están plenamente orientadas al monitoreo, análisis y representación en tiempo real; con la única funcionalidad relativa de exportar algunos datos y gráficos relevantes. Sin embargo, no permiten un análisis tan pormenorizado como puede ser el que ofrece la combinación de una herramienta como Wireshark, que trata paquete a paquete, y Qlik, que abarca unas dimensiones de análisis sin comparación.

Por tanto, no es sencillo de encontrar en la actualidad una herramienta que permita representar al nivel de detalle que se logra con esta implementación propuesta, alcanzando incluso la posibilidad de analizar, relacionar y representar apenas dos paquetes sueltos sin conexión temporal o de origen. Tampoco permiten las operaciones, automatizaciones y funciones con los datos, como puede ser la identificación de paquetes vía caracteres, el recuento de paquetes en función del tiempo relativo, del tipo de paquete, etc., el seguimiento de *conversaciones* en la red y su representación, y un sin fin de opcionalidades más.

Caso de estudio

Caso de estudio

Este proyecto contará, por su propia estructura, con dos fases principales: 1) la parte relativa al monitoreo y captura de los datos con Wireshark, 2) la parte relacionada con el análisis y presentación de los datos con Qlik Cloud.

Como se mencionaba anteriormente, el objetivo central del trabajo es demostrar la efectividad y utilidad de combinar la tecnología Qlik con la tecnología Wireshark de cara realizar un análisis de red. Para comprobar si el objetivo propuesto se cumple, se han planteado los siguientes objetivos específicos, cada uno de ellos desglosado en acciones concretas y detalladas:

1. Captura de datos de red:

- **Planificación de sesiones de captura:** Establecer un cronograma detallado para la captura de datos de red en distintos horarios y escenarios, asegurando la consistencia en el proceso de captura.
- **Configuración de Wireshark:** Configurar Wireshark con los filtros y ajustes necesarios para capturar datos específicos de cada tipo de conexión (HTTP, HTTPS, descargas, streaming, correo electrónico y multimedia).
- **Realización de múltiples sesiones de captura:** Ejecutar múltiples sesiones de captura en una máquina virtual (CyberOps Workstation) para cada tipo de conexión, registrando información sobre el contexto y las condiciones de cada sesión (por ejemplo, n.º de paquete, tipo de tráfico predominante, protocolo, IPs, Puertos, etc.).

2. Filtrado y organización de datos dentro de Wireshark:

- **Aplicación de filtros específicos:** Utilizar los filtros avanzados de Wireshark para aislar la información relevante de cada tipo de conexión, eliminando datos redundantes o irrelevantes. Esto, como se verá más adelante, supondrá una serie de problemas que requerirán de otras soluciones alternativas.

3. Conversión y corrección de datos:

- **Conversión de formatos de datos:** Convertir los datos capturados por Wireshark a un formato legible y procesable por Qlik Cloud, concretamente a formato .csv, corrigiendo formatos de fecha, delimitadores y diversos errores derivados del proceso de volcado.

4. Análisis de datos con Qlik Cloud:

- **Desarrollo de una estructura de base de datos:** Crear una estructura de base de datos en Qlik Cloud diseñada específicamente para alojar y manejar los datos capturados, optimizando el almacenamiento y el acceso a la información (scripting de Qlik) para la propia herramienta.
- **Implementación de operaciones analíticas:** Configurar y ejecutar diversas operaciones analíticas en Qlik Cloud, tales como agregaciones, cálculos, agrupaciones y correlaciones entre diferentes tipos de datos.

5. Visualización y presentación de resultados:

- **Desarrollo de dashboards interactivos:** Crear dashboards interactivos en Qlik Cloud que permitan explorar y visualizar los datos de manera intuitiva y dinámica, facilitando la interpretación de los resultados.
- **Generación de gráficos y visualizaciones:** Utilizar las capacidades de visualización de Qlik Cloud para generar gráficos y otros elementos visuales que representen claramente las características y patrones del tráfico de red.

6. Optimización y gestión de redes:

- **Demostración de la utilidad de Qlik Cloud:** Mostrar la versatilidad y capacidad de Qlik Cloud para realizar análisis detallados y ofrecer herramientas útiles para la toma de decisiones en la gestión y optimización de redes.

Desarrollo del proyecto

Desarrollo del proyecto

Para abordar la ejecución del trabajo en cuestión se ha propuesto una división de las fases de desarrollo en dos grandes secciones: 1) La fase de capturas de datos con Wireshark y 2) La fase de análisis y visualización de los datos con Qlik Cloud.

1 Captura de datos con Wireshark

1.1 Primeros pasos

Un primer problema que se encontró durante el inicio del desarrollo del estudio fue precisamente el de encontrar una herramienta no solo potente y útil a la hora de analizar el tráfico de red, sino también que fuera apropiada para su integración en un posterior análisis de datos con otra herramienta.

Se probaron herramientas como nmap y tcpdump, siendo ambas descartadas por diversos motivos. Por un lado, nmap no se ajustaba tanto a los intereses propuestos para el proyecto. Si bien es una herramienta con la capacidad de analizar el tráfico de red, su diseño y concepto están más enfocados al análisis de puertos de la red, lo cual no es la única información relevante que interesa. Por otro lado, tcpdump sí que es una tecnología más enfocada al análisis general de la red, su uso por comandos y los problemas relativos a los datos capturados que exportaba (mal formateados), hicieron que fuera rápidamente sustituida por Wireshark.

Esta última herramienta juega además con la ventaja de ser en la actualidad la herramienta más potente y usada en el mercado, por lo que la cantidad presente de información y ayuda relativa a la misma la hace altamente indicada para ser seleccionada.

Para implementar dicha herramienta se procedió a instalar directamente en una máquina real la versión de Wireshark 4.2.4.. El problema de optar por esta opción apareció en el momento de proceder a realizar la primera captura de red de prueba. El caso es que un equipo real cuenta con toda una serie de procesos en segundo plano que están necesariamente conectados a la red y de la cual no es posible, de primeras y sin afectar al rendimiento del equipo, desconectar.

Esto implica que las capturas realizadas cuenten, además de los paquetes relativos a las conexiones pretendidamente establecidas, con toda una serie de paquetes e información relativos a otras conexiones en segundo plano como puede ser del antivirus que está conectado a la red, de la herramienta de backup en la nube, etc. A estos paquetes se les denominará *ruido* a partir de este momento.

Como se ha mencionado ya en diversas ocasiones, el objetivo del estudio no es proporcionar una respuesta concreta a un problema real, es decir, a una conexión real; sino que es demostrar la capacidad analítica de Qlik Cloud para con datos de red obtenidos a través de otra herramienta de captura. Por ello lo interesante es acotar lo máximo posible las conexiones y así poder alcanzar un análisis más detallado y contrastable entre los resultados obtenidos y las conexiones propuestas.

La solución ofrecida a este problema fue en un primer momento optar por el uso de una máquina virtual personalizada de tal modo que contara con el mínimo imprescindible de programas conectados a la red. Para ello se probó con Arch Linux, una de las distros más personalizables y ligeras en este sentido. Sin embargo, presentó una serie de problemas a la hora de desconexión de ciertos programas a la red por lo que las capturas seguían conteniendo demasiado ruido. Esto último es fácilmente comprobable si se procede a capturar la red con Wireshark y no se tiene ninguna conexión pretendidamente establecida y a pesar de ello le llegan paquetes.

```
[root@archlinux gabriel]# neofetch
```

```


      .-
     .o+`
    .ooo/
   `+oooo:
  `+ooooooo:
 -+ooooooo+:
 `/:-:++oooo+:
  /++++/+++++++:
 /+++++++/+++++++:
  /++oooooooooooooo/`
 ./ooooSSSSso++oSSSSSSso+`
 .oSSSSSSso-`/oSSSSSSs+`
 -oSSSSSSso.      :SSSSSSso.
 :oSSSSSSs/      oSSSSso+++.
 /oSSSSSSSSs/      +SSSSoooo/-
 `oSSSSSSso+/-:-  -:/+oSSSSso+-
 +SSo+:-`          `.-/+oso:
 ++:..              -/+/:
 .                  -/+/:

```

```

root@archlinux
-----
OS: Arch Linux x86_64
Host: VirtualBox 1.2
Kernel: 6.9.1-arch1-2
Uptime: 5 mins
Packages: 681 (pacman)
Shell: bash 5.2.26
Resolution: 1280x800
DE: KDE
WM: KWin
Theme: Adwaita [GTK3]
Icons: Adwaita [GTK3]
Terminal: konsole
CPU: Intel i7-9750H (4) @ 2.592GHz
GPU: 00:02.0 VMware SVGA II Adapter
Memory: 1046MiB / 7943MiB

```



La siguiente opción a probar, y a la que finalmente se optó, fue la de usar la máquina virtual gratuita ofrecida por *Cisco: CyberOps Workstation*. Esta herramienta se caracteriza por ser una máquina virtual específicamente destinada a la investigación y prueba con redes. Incluye toda una serie de programas dirigidos al análisis y gestión de la red, entre ellos Wireshark. Lo más interesante de esta máquina es que está completamente aislada de la red a no ser que se le indique lo contrario.

Con esta herramienta se procedió entonces a realizar las primeras pruebas y las futuras capturas de datos.

1.2 Planificación y organización de las capturas

El siguiente paso a realizar fue el de organizar la recolección de los datos de tal modo que esto tenga sentido y no afecte a la coherencia interna de los datos.

Por ello se escogió mantener las capturas siempre en un mismo horario (20:00), todos los días durante 5 días.

También se escogieron los *tipos* de conexiones que se iban a realizar: a una web HTTP (http://www.hipertexto.info/documentos/internet_tegn.htm), a una web HTTPS (<https://www.cisco.com>), realizando una descarga de un archivo (<https://www.openoffice.org/es/descargar/>), a una web con contenido de streaming (<https://www.twitch.tv>), un envío de correo electrónico y a una web con contenido multimedia (<https://www.youtube.com>).

Esta clasificación de ningún modo se caracteriza por el protocolo en que se realiza la conexión (fundamentalmente TCP/IP) sino por el contenido que se transfiere durante la conexión y el modo y cantidad en que se transfieren dichos datos.

Así pues se realizaron 6 capturas por cada tipo de conexión, durante 5 días (entre el 27 de mayor y el 31 de mayo) a la misma hora cada día.

1.3 Configuración de Wireshark

Antes de proceder a analizar la red es importante saber qué datos pueden ser interesante a la hora de capturar. En este sentido Wireshark ofrece una enorme cantidad de opciones a escoger, tanto a nivel de conversación (los paquetes y sus relaciones) como a nivel de paquete. En este caso solo interesa el primer nivel, ya que el análisis pormenorizado lo llevará a cabo la otra herramienta de análisis de datos. Por ello es que tampoco se procedió a ningún filtrado dentro de la propia herramienta de Wireshark (ya que lo que interesa es ver la capacidad de filtrado de Qlik).

Los datos seleccionados fueron estos:

No.	Time
Absolute Time	Source address
Destination address	Source Port
Destination Port	Protocol
Length	Cumulative Bytes
Source MAC	Destination MAC
IP DSCP Value	Info

Básicamente en todo análisis de datos en los que los datos tienen **a)** un carácter repetitivo, es decir, que tratan de lo mismo (en este caso de un análisis de una conexión a la red), y **b)** una dimensión temporal, necesitarán tres columnas **(dimensiones) clave:**

- 1)** Una dimensión temporal (Date) que en este caso se la añadió a mano debido a que Wireshark presenta muchos problemas en sus campos de fecha.
- 2)** Una dimensión identificativa de qué representa cada conjunto de datos (Tipo), que también se agregó a mano ya que esta no es una opción que ofrezca la herramienta.
- 3)** Una dimensión identificativa de cada dato (No.), que asocia cada dato (celda) a su respectivo dato relacional (filas). En este caso sí usamos la propia dimensión No. que nos ofrece Wireshark y que nos indica el número de paquete.

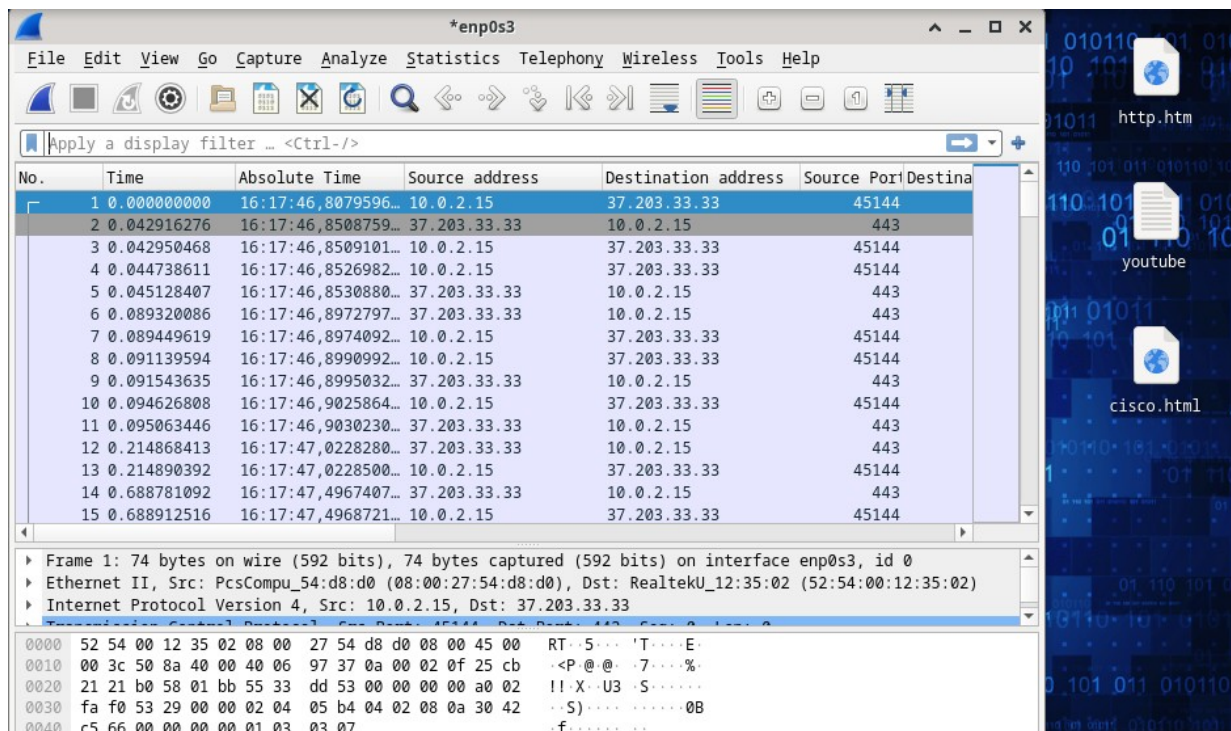
Más adelante veremos que al final algunos de los datos capturados no resultaron útiles debido a su información poco relevante.

1.4 Captura de Paquetes

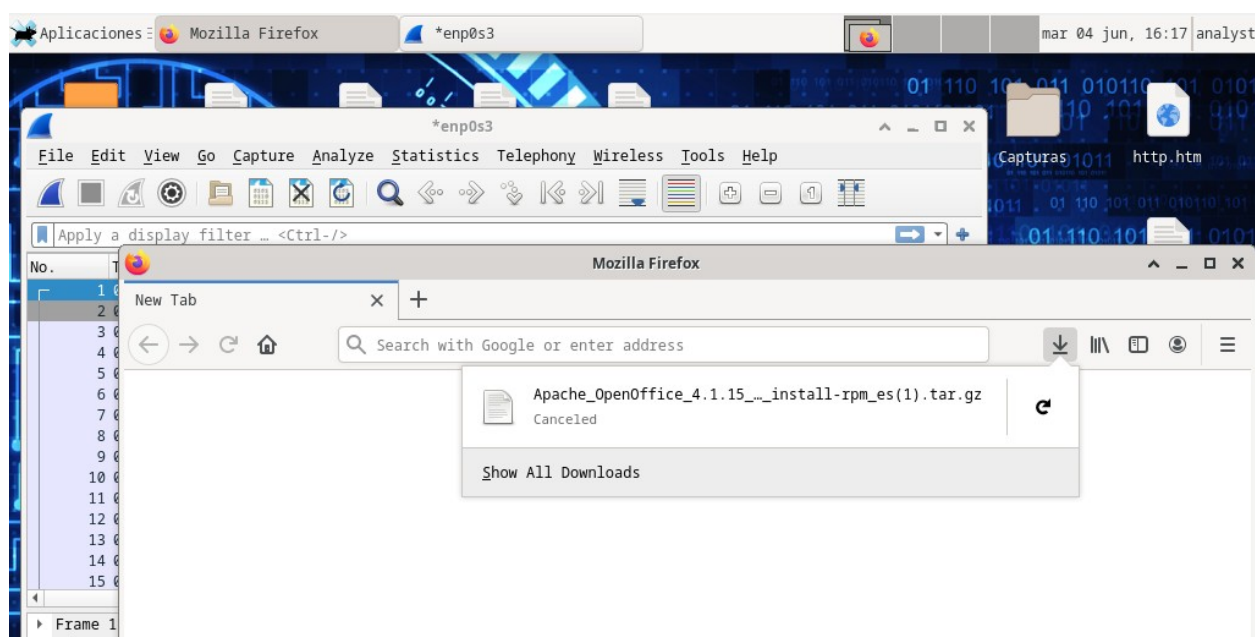
Con todo ello ya configurado se procedió a la captura de los datos:

Lo primero que se hizo fue hallar la forma de realizar la conexión directamente y en el menor tiempo posible. Esto se debe a que la propia apertura de una herramienta como un buscador ya activa la entrega y recepción de paquetes. Esto, por supuesto, será inevitable y además forma parte del proceso de conexión que se pretende analizar, pero reducir la cantidad de ruido que pueda haber en la conexión siempre favorece de cara un análisis fiel a los datos.

La solución que se encontró fue acceder directamente a través de archivos .html a la ruta especificada, evitando así la mayor cantidad posible de paquetes entre el tiempo de conexión a la red y la solicitud de conexión a una dirección URL determinada.



Así, por ejemplo, se hizo con el archivo de Descarga:



Posteriormente se procedió a repetir el proceso con cada tipo de conexión. Así, una vez obtenido los datos, se volcaron dichos datos a un archivo .csv con el que poder trabajar y cargar en Qlik Cloud. Siendo esta la estructura final del archivo de salida:

A1	No., "Time", "Absolute Time", "Source address", "Destination address", "Source Port", "Destination Port", "Protocol", "Length", "Cumulative Bytes", "Source MAC", "Destination MAC", "IP DSCP Value", "Info"															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	No.	"Time"	"Absolute Time"	"Source address"	"Destination address"	"Source Port"	"Destination Port"	"Protocol"	"Length"	"Cumulative Bytes"	"Source MAC"	"Destination MAC"	"IP DSCP Value"	"Info"		
2	1	"0.000000000"	"05:01:59,805065217"	"10.0.2.15"	"52.223.197.143"	"51248"	"443"	"TLSv1.2"	"1475"	"1475"	"PcsCompu_54:d8:d0"	"RealtekU_12:35:02"	"CS0"	"Application Data"		
3	2	"0.001797697"	"05:01:59,806862914"	"52.223.197.143"	"10.0.2.15"	"443"	"51248"	"TCP"	"60"	"1535"	"RealtekU_12:35:02"	"PcsCompu_54:d8:d0"	"CS0"	"443 > 51248 [ACK] Seq=1 Ack=1422 Win=65535 Len=0"		
4	3	"0.050195345"	"05:01:59,855260562"	"52.223.197.143"	"10.0.2.15"	"443"	"51248"	"TLSv1.2"	"294"	"1829"	"RealtekU_12:35:02"	"PcsCompu_54:d8:d0"	"CS0"	"Application Data"		
5	4	"0.050195955"	"05:01:59,855261172"	"52.223.197.143"	"10.0.2.15"	"443"	"51248"	"TCP"	"1514"	"3343"	"RealtekU_12:35:02"	"PcsCompu_54:d8:d0"	"CS0"	"443 > 51248 [PSH, ACK] Seq=241 Ack=1422 Win=65535 Len=1460 [TCP s		
6	5	"0.050196017"	"05:01:59,855261234"	"52.223.197.143"	"10.0.2.15"	"443"	"51248"	"TCP"	"1514"	"4857"	"RealtekU_12:35:02"	"PcsCompu_54:d8:d0"	"CS0"	"443 > 51248 [PSH, ACK] Seq=1701 Ack=1422 Win=65535 Len=1460 [TCP		
7	6	"0.050196065"	"05:01:59,855261282"	"52.223.197.143"	"10.0.2.15"	"443"	"51248"	"TCP"	"1514"	"6371"	"RealtekU_12:35:02"	"PcsCompu_54:d8:d0"	"CS0"	"443 > 51248 [PSH, ACK] Seq=3161 Ack=1422 Win=65535 Len=1460 [TCP		

Como se puede comprobar, carece de delimitaciones y muchos de los datos (sobre todo los de tiempo y fecha) han cambiado su formato de origen. Para ello fue necesario llevar a cabo todo un proceso de limpieza, formateo y corrección de los datos. Así como el de añadir las columnas identificativas de Tipo y Fecha mencionadas anteriormente. Siendo este el resultado final:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	No.	Time	Absolute Time	Source address	Destination address	Source Port	Destination Port	Protocol	Length	Cumulative Bytes	Source MAC	Destination MAC	IP DSCP Value	Info	Date	Tipo
2	1	0.000000000	38:08.9	10.0.2.15	172.217.168.174	56130	443	TLSv1.2	449	449	PcsCompu_54	RealtekU_12:35:02	CS0	Application Data	28/05/2024	MAIL
3	2	0.000135997	38:08.9	10.0.2.15	172.217.168.174	56130	443	TLSv1.2	6919	7368	PcsCompu_54	RealtekU_12:35:02	CS0	Application Data	28/05/2024	MAIL
4	3	0.000440054	38:08.9	172.217.168.174	10.0.2.15	443	56130	TCP	60	7428	RealtekU_12:35:02	PcsCompu_54	CS0	443 > 56130	28/05/2024	MAIL

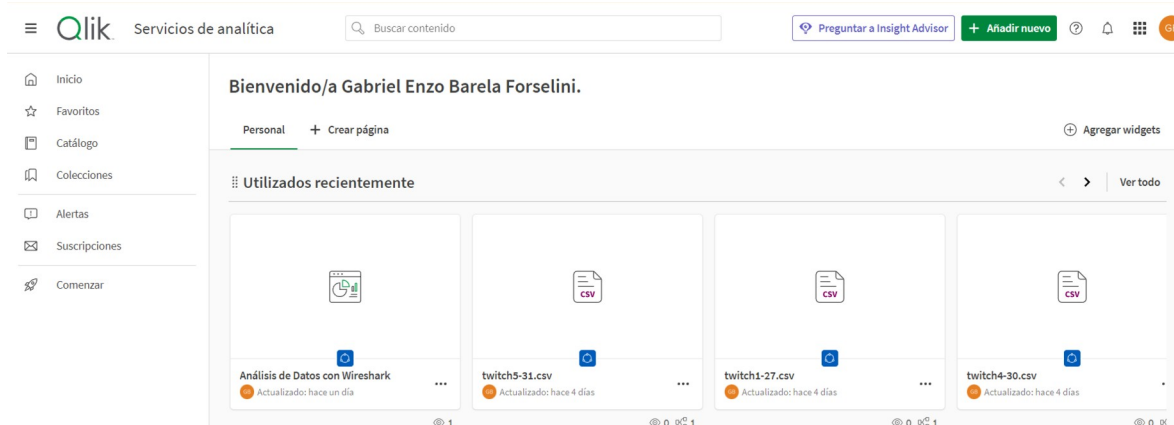
2 Análisis y visualización de los datos con Qlik Cloud

2.1 Qlik Cloud

En cuanto a Qlik Cloud, su selección frente a otras posibles herramientas de análisis como pueda ser PowerBI, la que es su gran competidora, se debe principalmente a dos motivos: 1) es la herramienta más potente en su nicho de análisis (análisis de datos relacionales o asociativos), 2) se cuenta con los conocimientos necesarios para su manejo.

Para ello se optó por usar la versión gratuita de prueba de un mes (con posible ampliación de hasta dos meses) que ofrece. Esta versión cuenta con todas las funcionalidades de Qlik Cloud. No requiere de instalación ya que se encuentra en la nube y tan solo se necesita crear una cuenta personal con la que acceder.

Este es el entorno principal de trabajo en Qlik Cloud:



2.2 Creación del tenant y de la APP

Una de las características principales del entorno Qlik es que se trabaja a nivel de *tenant* y APPs. Un *tenant* es básicamente un espacio común que puede ser compartido por diversos usuarios en los que se integra todo lo necesario para crear una o varias APPs. Es decir, contiene todos los datos, recursos, permisos de edición a cada usuario, etc. Por otro lado, las APPs son las aplicaciones finales (conjunto de Dashboards) que se entregarán al usuario final para que pueda interactuar con los datos.

En este caso se ha creado un único *tenant* con una única APP y para un único usuario (propietario).

2.3 Carga de datos y construcción de la base de datos

Una vez se tiene el espacio en el que trabajar se procede a la carga *bruta* de los datos. Se diferencia entre carga bruta y carga de datos en tanto que la primera se corresponde a la carga íntegra de los datos al programa de Qlik y la segunda a las sucesivas cargas posteriores sobre dichos datos una vez estén ya organizados, definidos, filtrados, operados, etc. Sobre esta última carga es sobre la que realmente se trabaja en la parte analítica y de visualización.

La carga bruta de los datos no requiere más que la subida de los archivos que se quiera al *tenant*. Por ello no será necesaria su explicación.

En cuanto a la carga en sí de los datos a la APP, se procederá a explicar paso a paso su constitución:

2.4 Main

El *Main* es la parte autogenerada por Qlik Cloud en la que se establecen varios parámetros relevantes a la hora de configurar la aplicación como puede ser el formato de la hora, el formato de los días de la semana, cuando empieza el primer día del año, los símbolos usados como delimitadores decimales, etc.

En este caso lo único que se modificó fueron los símbolos delimitadores para los miles y los decimales (pasando de “.” a “,” en el caso de estos últimos y viceversa). Puede parecer bastante superfluo, pero esta nimiedad puede suponer un auténtico problema a la hora de trabajar con funciones, algoritmos y operaciones con Qlik. Tanto es así, que este ha sido uno de los casos en los que ha sucedido.

La estructura final del Main es la siguiente:

```
SET ThousandSep='.';
SET DecimalSep=',';
SET MoneyThousandSep=',';
SET MoneyDecimalSep='.';
SET MoneyFormat='$ ###0.00;-$ ###0.00';
SET TimeFormat='h:mm:ss TT';
SET DateFormat='DD/MM/YYYY';
SET TimestampFormat='M/D/YYYY h:mm:ss[.fff] TT';
SET FirstWeekDay=6;
SET BrokenWeeks=1;
SET ReferenceDay=0;
SET FirstMonthOfYear=1;
SET CollationLocale='en-US';
SET CreateSearchIndexOnReload=1;
SET MonthNames='Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sep;Oct;Nov;Dec';
SET LongMonthNames=
'January;February;March;April;May;June;July;August;September;October;November;December';
SET DayNames='Mon;Tue;Wed;Thu;Fri;Sat;Sun';
SET LongDayNames='Monday;Tuesday;Wednesday;Thursday;Friday;Saturday;Sunday';
SET NumericalAbbreviation='3:k;6:M;9:G;12:T;15:P;18:E;21:Z;24:Y;-3:m;-6:μ;-9:n;-12:p;-15:f;-18:a;-21:z;-
24:y';
```

2.5 Carga de las tablas

El siguiente paso es cargar todas las tablas previamente subidas con los campos y requisitos que se precisen. El orden y la organización importa para el lenguaje qlik, que de forma similar a otros lenguajes de programación lee el código y lo ejecuta de arriba hacia abajo. Lo importante es identificar, como ya se hizo, el campo principal por el que se va a identificar cada tipo distinto de dato (Tipo) y cada dato (No.).

Una de las peculiaridades de Qlik en cuanto a su modelado de datos es que, a diferencia de otros lenguajes de carga y gestión de información como puede ser SQL, permite establecer relaciones entre campos que nada tienen que ver entre sí, o que cuyo formato no son el mismo, o que no se cuenten con el mismo número de filas, etc. Este campo identificador incluso puede ser más de uno y relacionar las tablas por más de un campo, dando como resultado lo que se denomina tablas sintéticas. Este tipo de tablas son preferiblemente evitables, aunque hay ocasiones en las que pueden ser necesarias y muy útiles. De todas formas los modelos óptimos son aquellos que o bien cuentan con una única tabla, o bien conforman un modelo en estrella con una tabla central. Las relaciones se establecen meramente por los nombres de los campos.

En este caso se escogieron todos los campos y se renombraron en función de varios aspectos:

- 1) “Tipo” funcionará como ID. Sin embargo, al crear en este caso una única tabla, este ID no estará relacionado con nada pero seguirá haciendo su función de identificador junto con la Fecha y el N.º de paquete. Habrá una relación posterior pero con un carácter muy peculiar que ya se abordará más adelante.
- 2) “Fecha”, como se ha dicho, será un campo particular que relacionará cada tabla de cada tipo con el resto de tablas del mismo tipo.
- 3) Qlik Cloud trabaja con los nombres de los campos y procede de una manera muy particular en este sentido: cuando todos los campos se denominan igual procede a fusionar las distintas tablas en una única tabla haciendo un *concatenate* implícito. Este comando se ha explicitado para un mejor entendimiento del código.
- 4) Lo que se hace en este trozo de código: `FROM [lib://Proyecto Qlik Cloud>DataFiles/cisco*.csv]` es buscar en la dirección en la que se encuentra el archivo todos los archivos que compartan dirección y nombre excepto aquellos caracteres que se encuentran entre el “*”. Por ello se nombraron todos los archivos csv con el mismo nombre numerado según el tipo y la fecha. Esto ahorra mucho código innecesario para la carga.

Este es el resultado (se repetiría exactamente el mismo proceso para cada tipo de conexión capturada, dando lugar a carga de 30 tablas distintas [6 tipos x 5 días]):

Temp1:

LOAD

```
Tipo,
No. AS Num.,
"Time" AS Tiempo,
"Absolute Time" AS Tiempo_Sin_Formato,
"Source address" AS Direccion_Origen,
"Destination address" AS Direccion_Destino,
"Source Port" AS Puerto_Origen,
"Destination Port" AS Puerto_Destino,
Protocol AS Protocolo,
"Length" AS Longitud,
"Cumulative Bytes" AS Bytes_Acumulados,
"Source MAC" AS MAC_Origen,
"Destination MAC" AS MAC_Destino,
"IP DSCP Value" AS Valor_DSCP_IP,
"Info" AS Informacion,
"Date" AS Fecha
```

```
// * coge todos los archivos "cisco...csv"
```

```
FROM [lib://Proyecto Qlik Cloud:DataFiles/cisco*.csv] (txt, codepage is 28591,
embedded labels, delimiter is ';', msq);
```

Concatenate(Temp1):

LOAD

```
Tipo,
No. AS Num.,
"Time" AS Tiempo,
```

```

"Absolute Time" AS Tiempo_Sin_Formato,

"Source address" AS Direccion_Origen,

"Destination address" AS Direccion_Destino,

"Source Port" AS Puerto_Origen,

"Destination Port" AS Puerto_Destino,

Protocol AS Protocolo,

"Length" AS Longitud,

"Cumulative Bytes" AS Bytes_Acumulados,

"Source MAC" AS MAC_Origen,

"Destination MAC" AS MAC_Destino,

"IP DSCP Value" AS Valor_DSCP_IP,

"Info" AS Informacion,

"Date" AS Fecha

FROM [lib://Proyecto Qlik Cloud:DataFiles//http*.csv] (txt, codepage is 28591,
embedded labels, delimiter is ';', msq);

```

Así con todo, este es el resultado final de la carga de las tablas:

Secciones	+	1 Concatenate(Templ)
		2 LOAD
		3 Tipo,
		4 No. AS Num.,
		5 "Time" AS Tiempo,
⋮ Main		6 "Absolute Time" AS Tiempo_Sin_Formato,
		7 "Source address" AS Direccion_Origen,
⋮ HTTPs		8 "Destination address" AS Direccion_Destino,
		9 "Source Port" AS Puerto_Origen,
		10 "Destination Port" AS Puerto_Destino,
⋮ HTTP		11 Protocol AS Protocolo,
		12 "Length" AS Longitud,
		13 "Cumulative Bytes" AS Bytes_Acumulados,
⋮ DESCARGA		14 "Source MAC" AS MAC_Origen,
		15 "Destination MAC" AS MAC_Destino,
		16 "IP DSCP Value" AS Valor_DSCP_IP,
⋮ MAIL		17 "Info" AS Informacion,
		18 "Date" AS Fecha
		19 FROM [lib://Proyecto Qlik Cloud:DataFiles/twitch*.csv]
⋮ MULTIMEDIA		20 (txt, codepage is 28591, embedded labels, delimiter is ';', msq);
		21
		22
⋮ STREAMING	🗑	
⋮ Operaciones		

2.6 Modificaciones y Operaciones con los datos

Una vez se han cargado las correspondientes tablas, llega el momento de trabajar ya sí con ellas. Este trabajo puede llevarse a cabo desde dos niveles:

- 1) Desde el frontend mediante la creación de *medidas* o *dimensiones maestras*, que son básicamente operaciones realizadas con las dimensiones (columnas) cargadas anteriormente.
- 2) Desde la propia carga mediante la creación de nuevas dimensiones (columnas). Este medio es el más recomendado para operaciones complejas que requieran de mucho procesamiento de datos y que además sirvan para luego operar nuevamente sobre ellas.

En este proyecto se han procedido a realizar diversas funciones y operaciones desde ambos lados de la APP. Comenzando por la carga de datos se pueden observar:

2.6.1 Creación de un rango con el n.º de paquetes

Usando la función `INLINE` que permite crear tablas con datos manualmente en el script en lugar de conectarse a archivos y bases de datos ⁽⁵⁾, se crea una serie de rangos que posteriormente serán establecidos como intervalos dentro de los propios datos del n.º de paquetes mediante la función `INTERVALMATCH`.

```
1 // Se crea una tabla manual usando INLINE
2 Grupo_Paquetes:
3 LOAD * INLINE [
4     Start, Stop, Grupo_Paquetes
5     0, 100, 0-100
6     101, 500, 101-500
7     501, 1000, 501-1000
8     1001, , 1001+
9 ];
10
11 // Se establecen los intervalos (rangos) usando INTERVALMATCH
12 INTERVALMATCH(Num.)
13 LEFT JOIN (Grupo_Paquetes)
14 LOAD
15     Start,
16     Stop
17 RESIDENT Grupo_Paquetes;
18
19 DROP FIELDS Start, Stop;
```

El resultado obtenido es el siguiente (asocia a cada paquete un rango definido):

Num.	Q	Grupo_Paquetes	Q
	96	0-100	
	97	0-100	
	98	0-100	
	99	0-100	
	100	0-100	
	101	101-500	
	102	101-500	

2.6.2 Creación de un rango con los bytes (Longitud)

Exactamente del mismo modo que con el n.º de paquetes:

```

40 Rango_Bytes:
41 LOAD * INLINE [
42     Start, Stop, Rango_bytes
43     0, 100, 0-100
44     101, 200, 101-200
45     201, 300, 201-300
46     301, 400, 301-400
47     401, 500, 401-500
48     501, 750, 501-750
49     751, 1000, 751-1000
50     1001, 1500, 1001-1500
51     1501, 2000, 1501-2000
52     2001, 2500, 2001-2500
53     2501, 3000, 2501-3000
54     3001, 3500, 3001-3500
55     3501, 4000, 3501-4000
56     4001, 5000, 4001-5000
57     5001, 6000, 5001-6000
58     6001, 7000, 6001-7000
59     7001, 8000, 7001-8000
60     8001, 9000, 8001-9000
61     9001, 10000, 9001-10000
62     10001, 25000, 10001-25000
63     25001, 50000, 25001-50000
64     50001, , 50001+
65 ];
66
67 INTERVALMATCH(Longitud)
68 LEFT JOIN (Rango_Bytes)
69 LOAD
70     Start,
71     Stop
72 RESIDENT Rango_Bytes;
73
74 DROP FIELDS Start, Stop;

```


2.6.3 La creación de un rango con el tiempo

Igual que antes:

```
Rango_Tiempo:
LOAD * INLINE [
    Start, Stop, Rango_Tiempo
    0.000000000, 0.100000000, 0.000000000-0.100000000
    0.100000001, 0.500000000, 0.100000001-0.500000000
    0.500000001, 1.000.000.000, 0.500000001-1.000.000.000
    1.000.000.001, 5.000.000.000, 1.000.000.001-5.000.000.000
    5.000.000.001, 10.000.000.000, 5.000.000.001-10.000.000.000
];

INTERVALMATCH(Tiempo)
LEFT JOIN (Rango_Tiempo)
LOAD
    Start,
    Stop
RESIDENT Rango_Tiempo;

DROP FIELDS Start, Stop;
```

Una vez esté constituida la tabla principal (a la que se van a añadir estas nuevas columnas creadas a partir de sus propios campos), se procedió a unirlas mediante el uso de un LEFT JOIN.

Un LEFT JOIN en Qlik, para que sea correcto, necesita de que haya al menos un campo que relacione ambas tablas a unir. En este caso ese campo lo otorga el INTERVALMATCH en el momento que relaciona los rangos con una dimensión de la tabla principal (Num., Longitud y Tiempo).

```
// Se unifican las tablas con los tres rangos
LEFT JOIN (Wireshark)
LOAD *
RESIDENT Grupo_Paquetes;

LEFT JOIN (Wireshark)
LOAD *
RESIDENT Rango_Tiempo;

LEFT JOIN (Wireshark)
LOAD *
RESIDENT Rango_Bytes;
```

* Aquí todavía no se ha constituido definitivamente la tabla principal (por ello esto va después).

2.6.4 Carga de la tabla principal y operaciones

Para la carga de la tabla principal (en la que se encontrarán todos los datos relevantes) se procedió repitiendo los campos que se han ido concatenando en las cargas anteriores.

Debido que el campo “Tiempo” presenta una serie de errores a la hora de volcarlo desde Wireshark hacia el archivo .csv, se le aplicará la función NUM que le otorgará *“formato a un número, es decir, convierte el valor numérico de la entrada para mostrar texto en vez, utilizando el formato especificado en el segundo parámetro. Si se omite el segundo parámetro, utiliza los separadores de decimal y de miles definidos en el script de carga de datos”* ⁽⁶⁾.

```

76 // Tabla final
77 Wireshark:
78 Load
79     Tipo,
80     Num.,
81     Num(Tiempo) AS Tiempo,
82     Tiempo_Sin_Formato,
83     Direccion_Origen,
84     Direccion_Destino,
85     Puerto_Origen,
86     Puerto_Destino,
87     Protocolo,
88     Longitud,
89     Bytes_Acumulados,
90     MAC_Origen,
91     MAC_Destino,
92     Valor_DSCP_IP,
93     Informacion,|
94     Fecha,
```

A continuación se procedió a sacar del campo “Información” una serie de distintos tipos de contenido de cada paquete:

1) Paquetes de datos (Application Data): Son aquellos que dentro del tráfico cifrado TLS/SSL contienen la mayor parte de los datos propios de la información que se está a recibir o enviar (código html de la web, contenido multimedia, etc.).

2) Protocolo de enlace de 3 vías TCP (SYN, SYN-ACK, ACK): Es un proceso que se utiliza en una red TCP/IP para establecer una conexión entre el cliente y el servidor. Se trata de un proceso de tres pasos que requiere que tanto el cliente como el servidor intercambien paquetes SYN (sincronización) y ACK (reconocimiento) antes de que comience el proceso real de comunicación de datos.

3) PSH-ACK (Pasar) ⁽⁷⁾: *“Indica a las pilas de red que omitan el almacenamiento en búfer.”*

4) FIN-ACK (Finalizar) ⁽⁷⁾: *“Se termina armoniosamente la conexión TCP.”*

5) Client Hello y Server Hello: Es la primera fase del protocolo TLS/SSL, que se utiliza para iniciar una conexión segura entre un cliente y un servidor. Cuando un cliente (como puede ser un navegador web) desea establecer una conexión segura con un servidor web, inicia el proceso de *handshake* TLS/SSL enviando un mensaje "Client Hello". Una vez que el servidor recibe el mensaje "Client Hello", responde con un mensaje "Server Hello".

Para llevarlo se usó el lenguaje de *scripting* para Qlik con la función WildMatch que *“compara el primer parámetro con todos los siguientes y devuelve el número de la expresión”*.

```
96 // Paquetes Sincronización y Reconocimiento (Synchronize-Acknowledge)
97 // Paquetes de datos (Application Data)
98 // Negociación de parámetros de seguridad (Client Hello y Server Hello)
99
100 if(
101   WildMatch(Informacion, '*[SYN, ACK] Seq=*'), 'SYN-ACK',
102   if(
103     WildMatch(Informacion, '*[SYN]*') and not WildMatch(Informacion, '*[SYN, ACK] Seq=*'), 'SYN',
104     if(
105       WildMatch(Informacion, '*[ACK]*') and not WildMatch(Informacion, '*[SYN, ACK] Seq=*'), 'ACK',
106       if(
107         WildMatch(Informacion, '*[PSH, ACK]*'), 'PSH-ACK',
108         if(
109           WildMatch(Informacion, '*[FIN, ACK]*'), 'FIN-ACK',
110           if(
111             WildMatch(Informacion, '*Server Hello*'), 'Server Hello',
112             if(
113               WildMatch(Informacion, '*Client Hello*'), 'Client Hello',
114               if(
115                 WildMatch(Informacion, '*Application Data*'), 'Application Data',
116                 'Other'
117               )
118             )
119           )
120         )
121       )
122     )
123   )
124 ) AS Tipo_Contenido_Paquetes,
```

Algo similar se hizo para sacar las distintas clases de IP de origen e IP de destino, pero en este caso mediante la función SUBFIELD combinada con NUM:

```

128 // Sacar la clase de la IP de Origen
129 If(
130     Num(SubField("Direccion_Origen", '.', 1)) >= 1 and Num(SubField("Direccion_Origen", '.', 1)) <= 126, 'Clase A',
131     If(
132         Num(SubField("Direccion_Origen", '.', 1)) >= 128 and Num(SubField("Direccion_Origen", '.', 1)) <= 191, 'Clase B',
133         If(
134             Num(SubField("Direccion_Origen", '.', 1)) >= 192 and Num(SubField("Direccion_Origen", '.', 1)) <= 223, 'Clase C',
135             'No clasificada'
136         )
137     )
138 ) AS Clase_IP_Origen,
139
140 // Sacar la clase de la IP de Destino
141 If(
142     Num(SubField("Direccion_Destino", '.', 1)) >= 1 and Num(SubField("Direccion_Destino", '.', 1)) <= 126, 'Clase A',
143     If(
144         Num(SubField("Direccion_Destino", '.', 1)) >= 128 and Num(SubField("Direccion_Destino", '.', 1)) <= 191, 'Clase B',
145         If(
146             Num(SubField("Direccion_Destino", '.', 1)) >= 192 and Num(SubField("Direccion_Destino", '.', 1)) <= 223, 'Clase C',
147             'No clasificada'
148         )
149     )
150 ) AS Clase_IP_Destino,

```

SUBFIELD “se utiliza para extraer componentes de subcadenas de un campo de cadena principal, donde los campos de registro originales constan de dos o más partes separadas por un delimitador.”⁽⁸⁾

Por último, se procedió a elaborar un cálculo con la dimensión “Tiempo_Sin_Formato” (Absolute Time) para evitar su uso de caracteres como “.” y transformar su contenido a segundos con milésimas de segundo. Esta nueva dimensión será usada posteriormente para calcular la latencia entre paquetes.

```

152 //Sacar el tiempo real en segundos
153
154 (Num(SubField(Tiempo_Sin_Formato, ':', 1)) * 60 + Num(SubField(Tiempo_Sin_Formato, ':', 2))
155 + Num(SubField(Tiempo_Sin_Formato, ',', 1)) / 10) as Tiempo_Convertido
156
157
158 RESIDENT Temp1;
159

```

Una vez establecida la dimensión y cargada la tabla, se creará una tabla relacionada a través de la nueva dimensión “Tiempo_Convertido” que calculará el mismo tiempo pero con su valor siguiente, esto es: al primer valor de la columna “Tiempo_Convertido” le corresponderá el segundo valor de la misma columna ahora en una nueva columna. Para ello se usarán las funciones RecNo y PEEK.

```

160 Latencia:
161 LOAD
162     Tiempo_Convertido,
163     If(RecNo() = 1, 0, Peek('Tiempo_Convertido')) as Tiempo_Convertido_Siguiente
164 RESIDENT Wireshark;

```

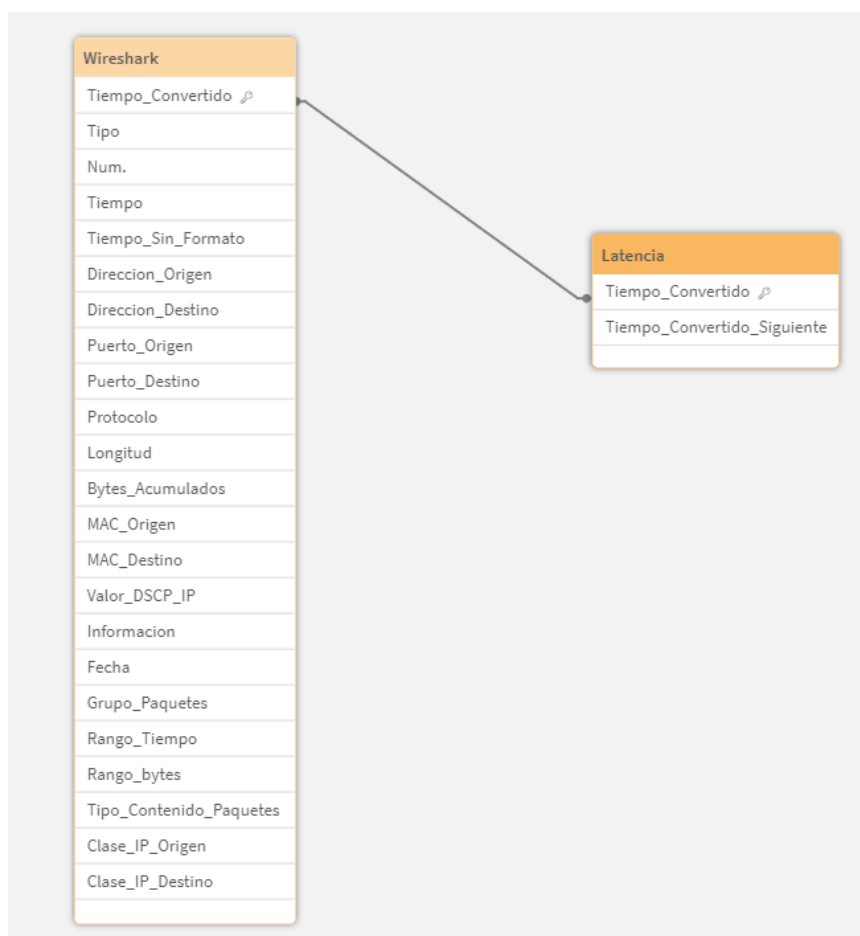
La función Peek() “devuelve el valor de un campo en una tabla para una fila que ya se ha cargado o que existe en la memoria interna. El número de fila se puede especificar, así como la tabla. Si no se especifica un número de fila, se utilizará el último registro cargado anteriormente.” ⁽⁹⁾

RecNo por su parte, “devuelve un entero con el número de la fila actual de un tabla interna. El primer registro es el número 1.” ⁽¹⁰⁾

Por último, se procede (previa relación con los LEFT JOINS ya explicados) a borrar las tablas de datos temporales que se usaron para trabajar con los datos; y se cargan los datos.

```
180 // Se borran las tablas temporales
181 DROP TABLE Grupo_Paquetes;
182 DROP TABLE Rango_Tiempo;
183 DROP TABLE Rango_Bytes;
184 DROP TABLE Temp1;
```

Este es el resultado final:

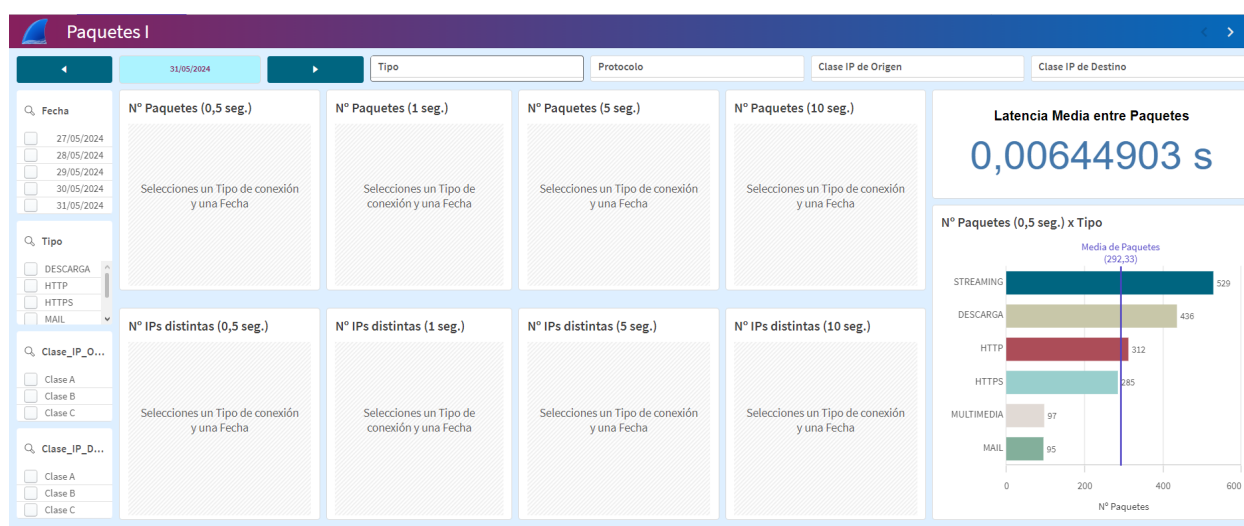


2.7 Creación de los Dashboards y las visualizaciones

El primer Dashboard de la APP realizado costa de diversos elementos interesantes a la hora de mostrar las funcionalidades de Qlik Cloud. Esta hoja resume en un único vistazo todo lo relativo a los paquetes y las IPs, el n.º de paquetes x tiempo, el n.º de paquetes x tipo de conexión y la latencia media entre paquetes.

Se ha escogido una serie de medidas de tiempo relevantes (0,5; 1; 5 y 10 segundos) para observar numéricamente la evolución de la cantidad de paquetes y las Ips.

Para ello se ha creado un *estado alterno*, es decir, un campo alternativo al principal por el que se van a agrupar las medidas. Esto implica que si se selecciona en el menú principal una fecha, esta fecha no afectará a las medidas marcadas como alternas, pero sí lo hará si se modifica la fecha alterna (dicho estado alterno se encuentra a la izquierda en vertical de la hoja).



También se han añadido unos botones y un menú de texto que moverán intuitivamente el estado general de la Fecha. Se usa para ello la función ONLY, la función Date, la función MAX y la función TOTAL. Básicamente lo que hace esta combinación es seleccionar una fecha sobre el total del campo "Fecha" (sin tener en cuenta si hay alguna fecha ya seleccionada) y añadirle un valor (ir hacia adelante en la fecha) o restarle otro (ir hacia atrás).

```
1 =Only({<Fecha={"$(=Date(Max(TOTAL Fecha) -1))"}>} Fecha)
```

```
1 =Only({<Fecha={"$(=Date(Max(TOTAL Fecha) + 1))"}>} Fecha)
```

Para contar el n.º de Paquetes simplemente se ha optado por el uso de una función COUNT combinada con un set análisis que delimita las selecciones a los campos indicados. En el siguiente ejemplo, cuenta todo los paquetes (Num.) donde el campo “Tiempo” haya sido inferior a 0,5 segundos:

```
1 Count({<Tiempo={"<=0.500000000"}>} Num.)
```

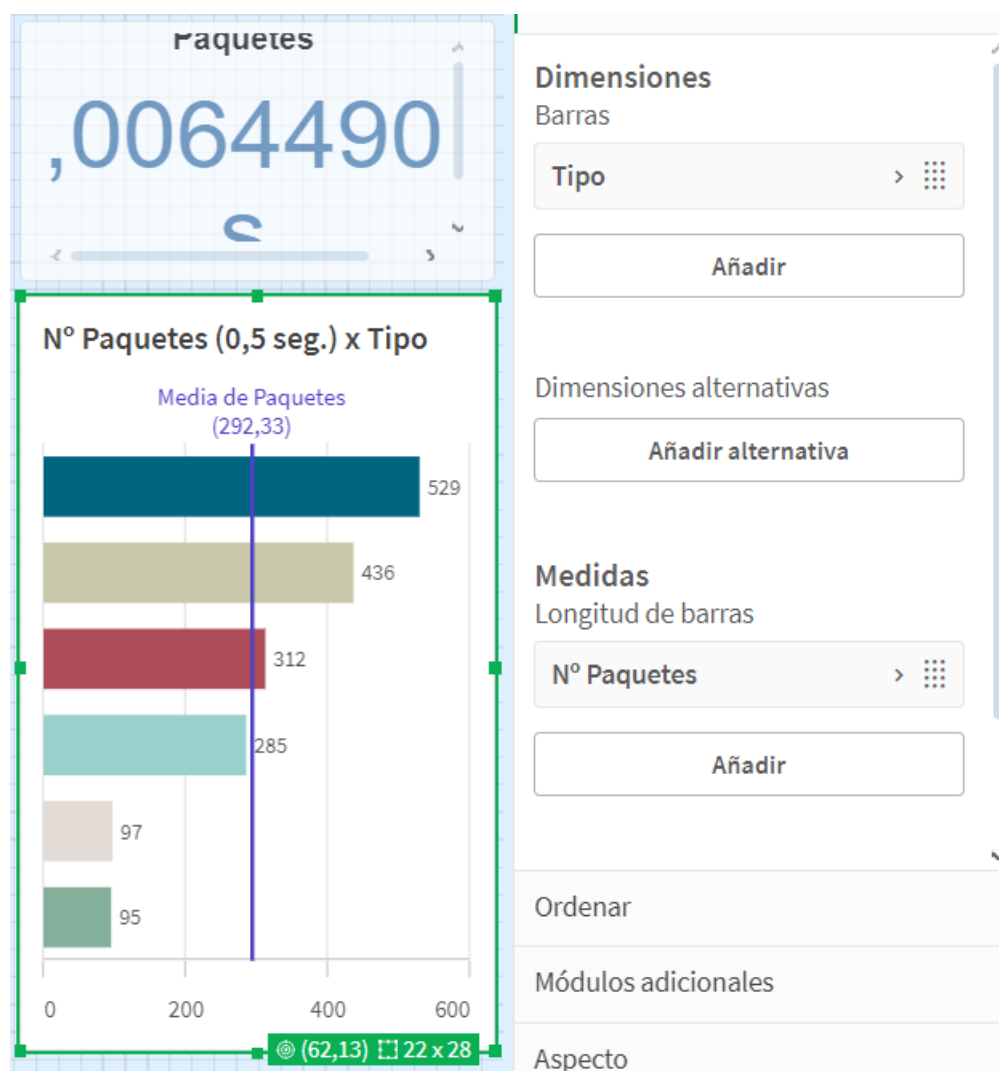
Por su parte, la latencia, uno de los cálculos más complejos a pesar de su simplicidad aparente, se ha logrado operando con las dos dimensiones creadas en la carga de datos. Sin embargo, no resultaría eficiente hacer dicha operación en la misma carga, por lo que se optó por realizarla en el *frontend* como una medida:

```
1 AVG(if(Tiempo_Convertido >= Tiempo_Convertido_Siguiente,  
2     if(Tiempo_Convertido - Tiempo_Convertido_Siguiente <= 2, Tiempo_Convertido - Tiempo_Convertido_Siguiente, 0),  
3     if(Tiempo_Convertido_Siguiente - Tiempo_Convertido <= 2, Tiempo_Convertido_Siguiente - Tiempo_Convertido, 0)  
4 ))
```

Como se puede observar, lo que se opera es básicamente la diferencia existente entre la llegada de un paquete y el siguiente (Tiempo 1) – (Tiempo 2). Además, se crea una condición según la cual si un valor de tiempo supera los 2 segundos de diferencia, este no se tenga en cuenta. Esto se debe a que a la hora de analizar algunas de las conexiones a un streaming, surgieron problemas en cierto momento debido a las limitaciones de la red. Estos valores se consideran residuales en cuanto a su número de apariciones (apenas en 3 paquetes) pero afectarían significativamente al cálculo de la media.

También se ha creado un gráfico de barras que mide por la dimensión de “Tipo” (tipo de conexión) entre el n.º de paquetes (COUNT) y se le ha realizado la media como línea de referencia:

```
1 =Count({<Tiempo={"<=0.500000000"}>} Num.) / Count(distinct Tipo)
```



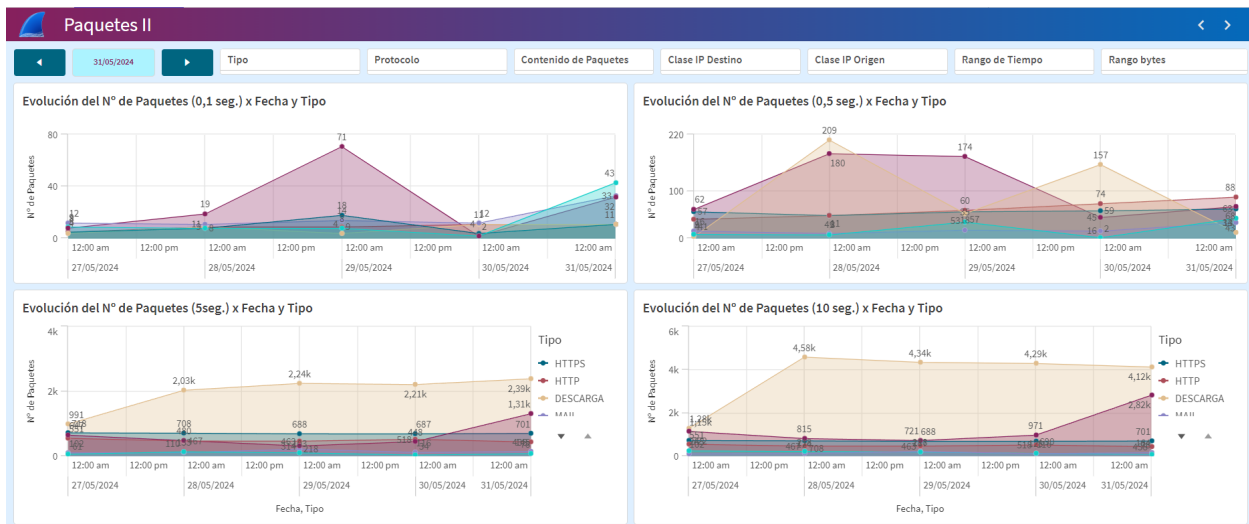
Otro de los cálculos interesantes realizados es el de IPs distintas (con la función DISTINCT) tanto de origen como de destino. La medida es la siguiente:

```
1 Count({<Tiempo={"<=0.500000000"}>} distinct Direccion_Origen)
```

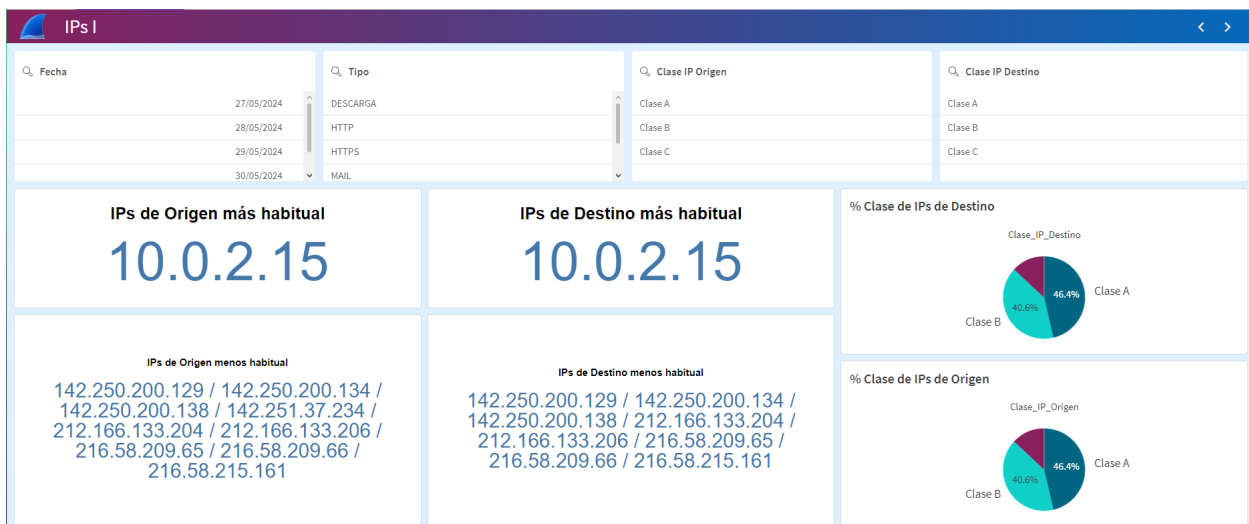
Y por último, además de otorgarle algo de estilo a la hoja, se ha creado un panel de filtrado como estado general que, con variaciones, afectará al conjunto de todas las hojas.

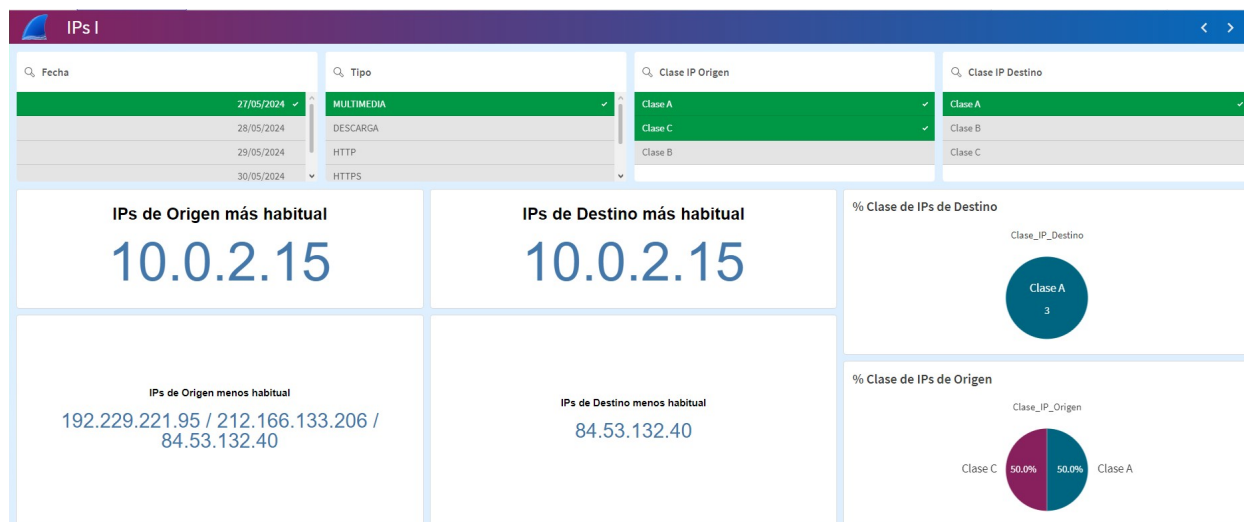
Desarrollo del proyecto

La siguiente hoja no es más que una continuación de la anterior con una serie de gráficos de líneas combinados en los que se entrelazan los distintos tipos de conexión por fecha y tiempo.



El siguiente cuadro corresponde al análisis de las IPs. En él se muestran las IPs más habituales y menos habituales, así como el porcentaje respectivo de cada clase de IP. Esto es relativo por supuesto a cada selección:





La expresión usada tanto para encontrar la IP que más (MAX) se repite como la que menos (MIN) es:

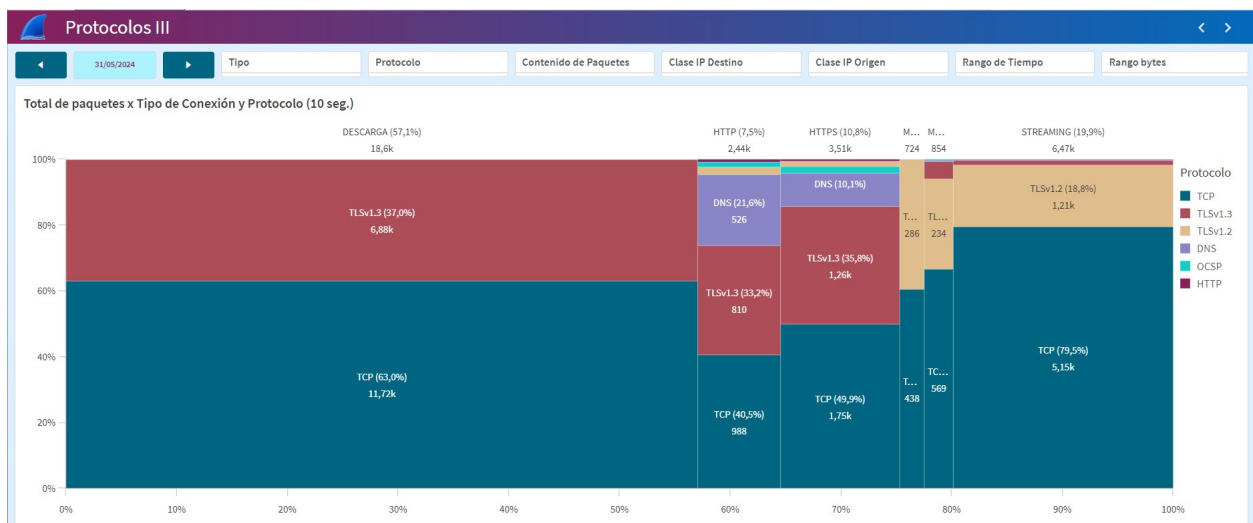
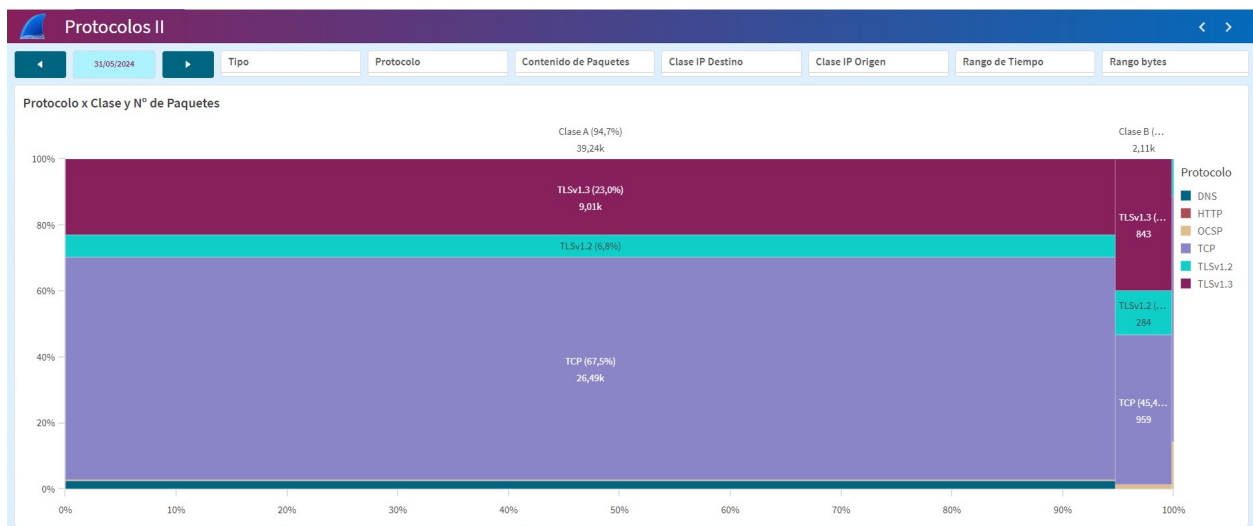
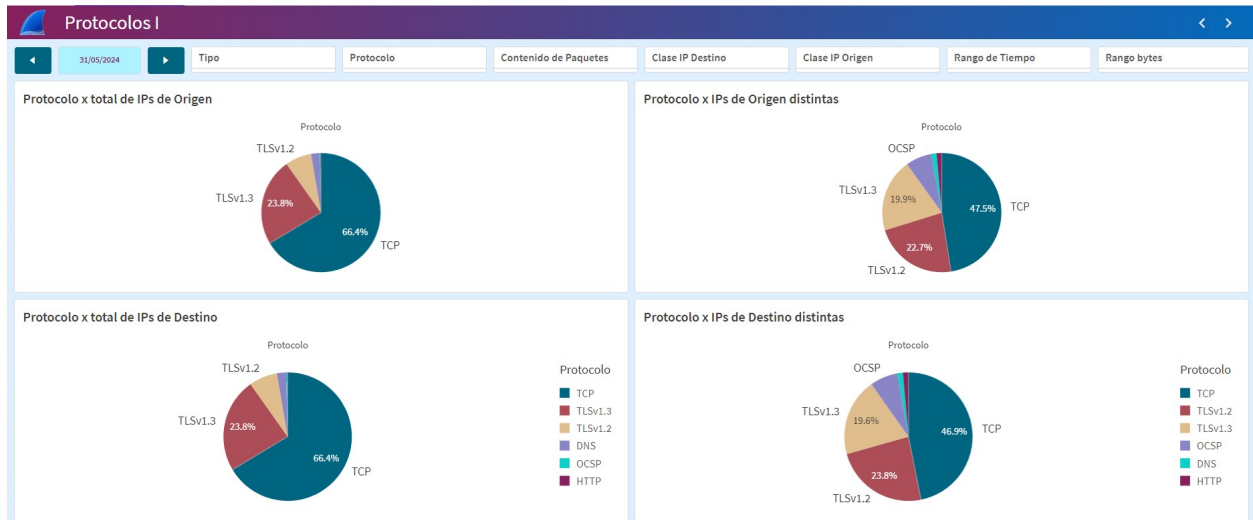
```
1 Concat(DISTINCT If(Aggr(Count(Direccion_Destino), Direccion_Destino)
2 = Max(TOTAL Aggr(Count(Direccion_Destino), Direccion_Destino)), Direccion_Destino), ' / ')
```

```
1 Concat(DISTINCT If(Aggr(Count(Direccion_Destino), Direccion_Destino)
2 = Min(TOTAL Aggr(Count(Direccion_Destino), Direccion_Destino)), Direccion_Destino), ' / ')
```

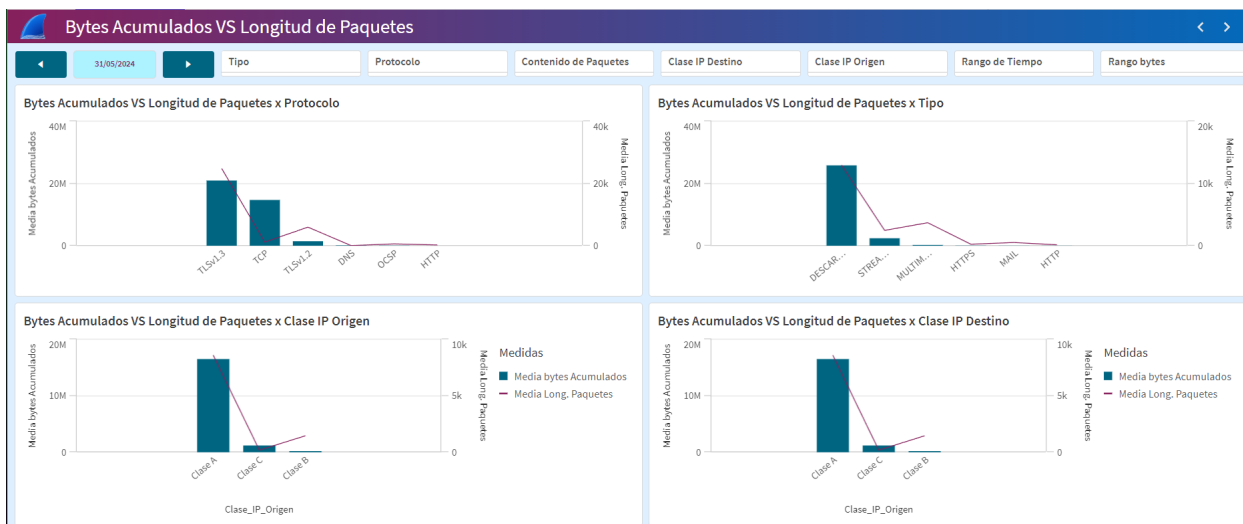
En ella se usa una estructura IF en la que se seleccionan los valores distintos, agregados (función AGGR: devuelve un conjunto de valores para la expresión calculada sobre la dimensión o dimensiones indicadas. ⁽¹¹⁾) por la IP de destino o de origen y luego se concatenan en caso de que correspondan a más de un valor usando como delimitador “/”.

Para el gráfico de tarta no es necesario más que añadir como dimensión la dimensión creada “Clase_IP_Destino” o “Clase_IP_Origen” y una medida que cuente el número de direcciones de destino u origen, respectivamente.

Las siguientes hojas se han ido realizando con distintos tipos de gráfico como gráficos de bloques que calculan medidas a partir de dos dimensiones, gráficos de dispersión que también trabajan en dos dimensiones, gráficos de líneas, tablas pivotantes que son útiles a la hora de trabajar con valores los cuales interesa cambiar de relación entre sí, y demás opciones visuales que puedan resultar atractivas e interesantes. En cuanto a las expresiones usadas no hay ninguna especialmente relevante o diferente a las ya vistas.



Desarrollo del proyecto



MAC I

31/05/2024 Tipo Protocolo Contenido de Paquetes Clase IP Destino Clase IP Origen Rango de Tiempo Rango bytes

N° MACs de Origen: 2

N° MACs de Destino: 2

MAC de Origen x Contenido de Paquete

Contenido del Paquete	MAC Origen
ACK	PcsCompu_54:d8:d0
ACK	RealtekU_12:35:02
Application Data	PcsCompu_54:d8:d0
Application Data	RealtekU_12:35:02
Client Hello	PcsCompu_54:d8:d0
FIN-ACK	PcsCompu_54:d8:d0
FIN-ACK	RealtekU_12:35:02
Other	PcsCompu_54:d8:d0
Other	RealtekU_12:35:02
PSH-ACK	PcsCompu_54:d8:d0
PSH-ACK	RealtekU_12:35:02
Server Hello	RealtekU_12:35:02
SYN	PcsCompu_54:d8:d0
SYN-ACK	RealtekU_12:35:02

MAC de Destino x Contenido de Paquete

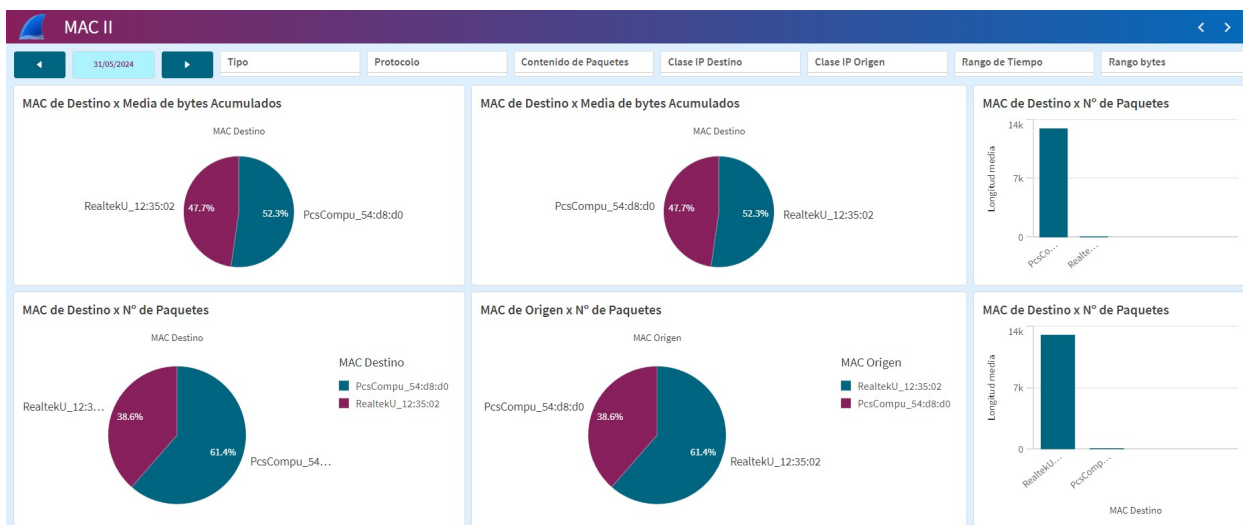
Contenido del Paquete	MAC Destino
ACK	PcsCompu_54:d8:d0
ACK	RealtekU_12:35:02
Application Data	PcsCompu_54:d8:d0
Application Data	RealtekU_12:35:02
Client Hello	RealtekU_12:35:02
FIN-ACK	PcsCompu_54:d8:d0
FIN-ACK	RealtekU_12:35:02
Other	PcsCompu_54:d8:d0
Other	RealtekU_12:35:02
PSH-ACK	PcsCompu_54:d8:d0
PSH-ACK	RealtekU_12:35:02
Server Hello	PcsCompu_54:d8:d0
SYN	RealtekU_12:35:02
SYN-ACK	PcsCompu_54:d8:d0

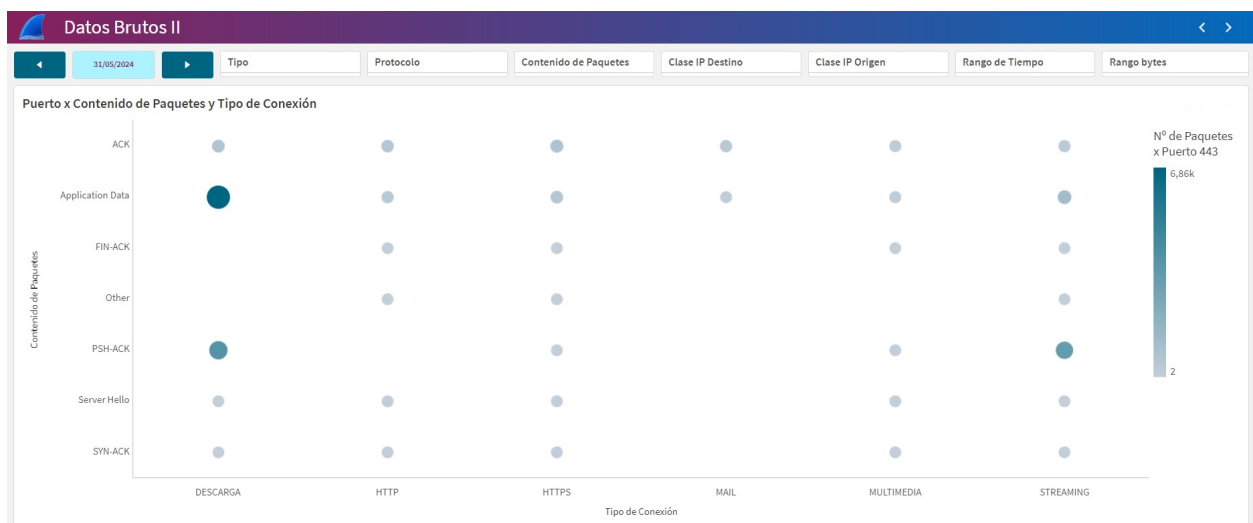
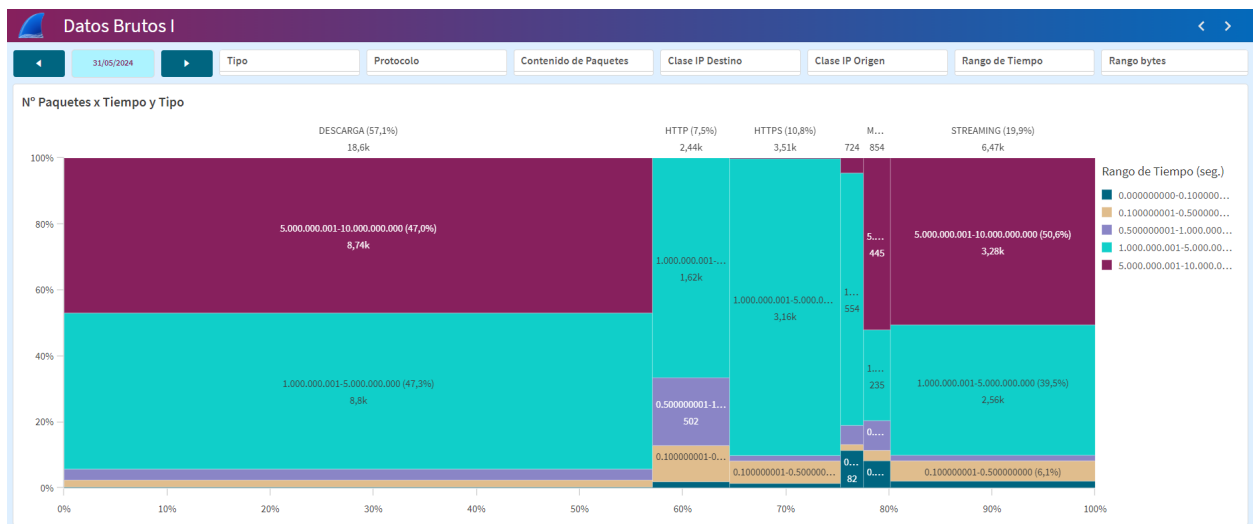
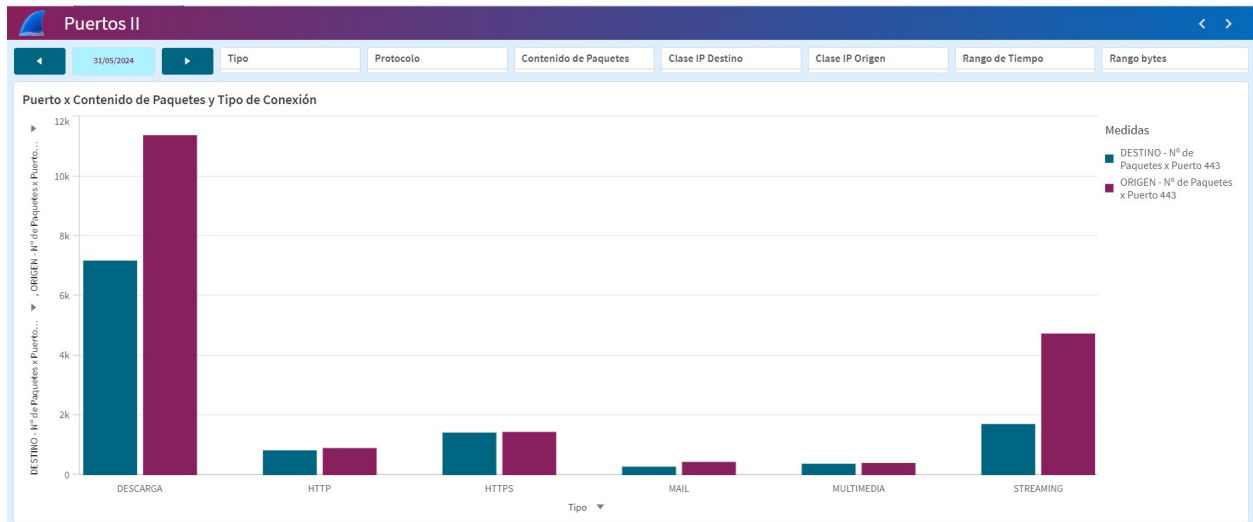
MAC x Puerto

Puerto_Origen: Valores:

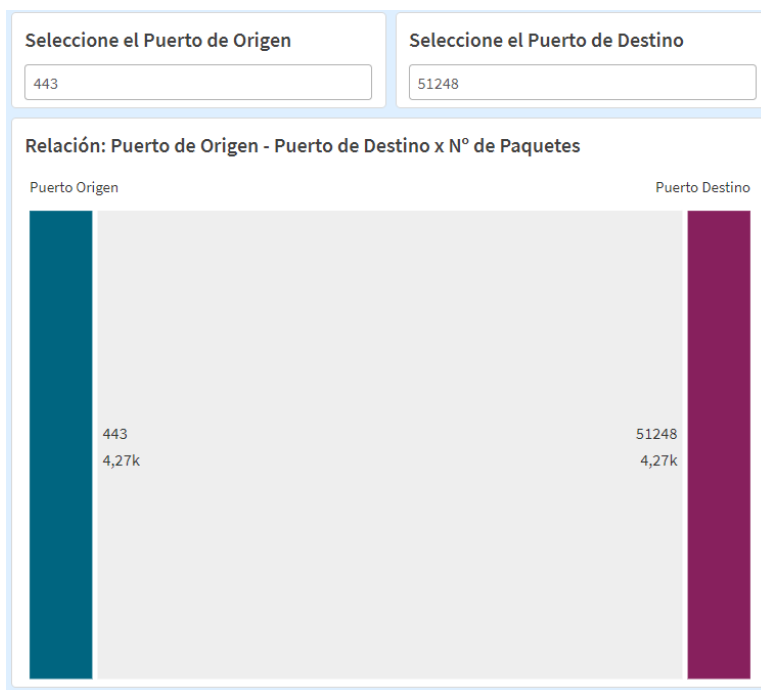
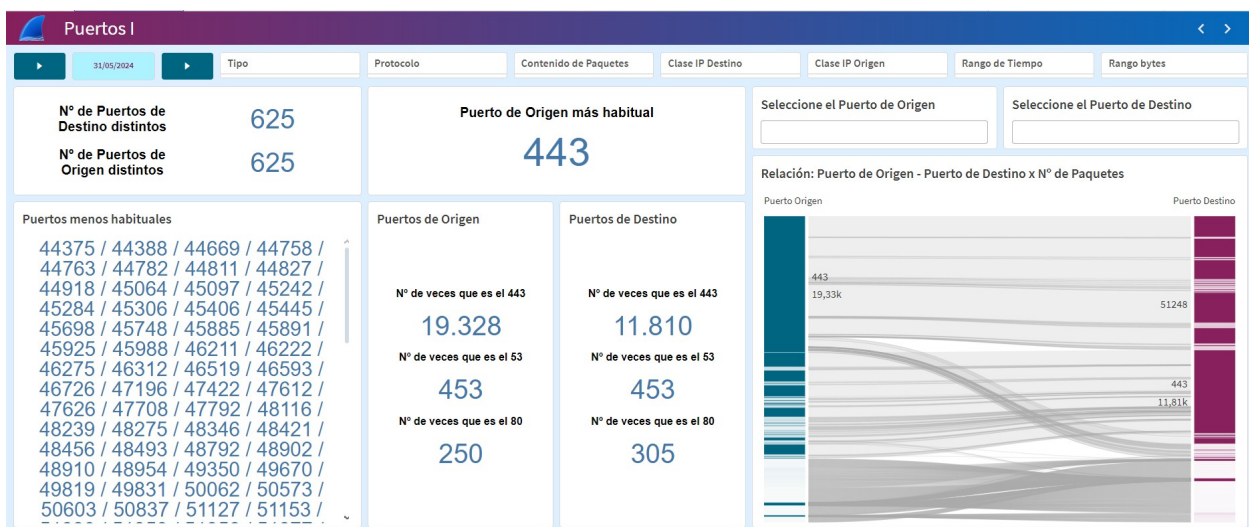
Puerto_Destino:

	MAC_Destino	MAC_Origen
53	PcsCompu_54:d8:d0	RealtekU_12:35:02
80	PcsCompu_54:d8:d0	RealtekU_12:35:02
443	PcsCompu_54:d8:d0	RealtekU_12:35:02
32872	RealtekU_12:35:02	PcsCompu_54:d8:d0
32913	RealtekU_12:35:02	PcsCompu_54:d8:d0
33062	RealtekU_12:35:02	PcsCompu_54:d8:d0
33177	RealtekU_12:35:02	PcsCompu_54:d8:d0
33179	RealtekU_12:35:02	PcsCompu_54:d8:d0
33218	RealtekU_12:35:02	PcsCompu_54:d8:d0
33362	RealtekU_12:35:02	PcsCompu_54:d8:d0
33397	RealtekU_12:35:02	PcsCompu_54:d8:d0
33410	RealtekU_12:35:02	PcsCompu_54:d8:d0
33564	RealtekU_12:35:02	PcsCompu_54:d8:d0
33695	RealtekU_12:35:02	PcsCompu_54:d8:d0
33702	RealtekU_12:35:02	PcsCompu_54:d8:d0
33768	RealtekU_12:35:02	PcsCompu_54:d8:d0
33790	RealtekU_12:35:02	PcsCompu_54:d8:d0





Otra de las funcionalidades interesantes del entorno Qlik es sin duda el de las variables. Las variables son aquellas expresiones que dentro de una función permiten otorgar distintos valores a una operación de medida. Estos valores pueden ser relativos, es decir, si acontece una selección X a la variable se le otorga un valor Y, o, como sucede en este caso, el valor es funcional, es decir, se le otorga directamente en función de lo que se quiera (sujeto por supuesto a los valores posibles de una dimensión). En este caso la variable está pensada para ser manejada por el usuario final, otorgándole la posibilidad de seleccionar vía escrita los puertos de origen y destino interesados:



La expresión utilizada para lograr esto fue la siguiente, donde “vPuertos” y “vPuertos2” se corresponden a las variables creadas para los puertos de origen y de destino, respectivamente:

```

1 Count(
2   {
3     <
4     Tiempo = {"<=10.000.000.000"},
5     $(=if(len(trim('$(vPuertos)')) > 0, 'Puerto_Origen = {' & vPuertos & '}', '')),
6     $(=if(len(trim('$(vPuertos2)')) > 0, 'Puerto_Destino = {' & vPuertos2 & '}', ''))
7   >}
8   [Num.]
9 )

```

Como se puede apreciar, también es posible combinar indefinidamente (mientras las dimensiones lo permitan) distintos *set análisis*:

```

1 Count({<Tiempo={"<=10.000.000.000"}, Puerto_Destino={"443"}>}Puerto_Destino)

```

En este caso se obtiene el número total de veces que el puerto de destino es el 443 en los primeros diez segundos de captura.

Quedarían aún toda una serie de expresiones, funciones y funcionalidades interesantes por explicar pero que debido a la extensión de las mismas y las dificultades que presentaría exposición en este formato, serán obviadas.

Conclusiones

Conclusiones

Una vez obtenidos los resultados propuestos al inicio del proyecto se puede decir que estos están acorde a lo esperado. Por un lado, se ha logrado capturar y analizar distintos tipos de conexiones, demostrando la versatilidad y capacidad de estas herramientas para ofrecer análisis gráficos interactivos e intuitivos sobre redes informáticas. Por el otro, se ha demostrado la utilidad de dichos análisis gráficos, por ejemplo a la hora de encontrar puertos con menor interacción en la red que puedan suponer alguna entrada menos conocida, o por el contrario, aquellos puertos que soportan un mayor volumen de carga.

Así pues, los resultados obtenidos ofrecen una comprensión profunda de cómo diferentes actividades de red afectan al rendimiento y la seguridad. La identificación de patrones de tráfico y la detección de posibles problemas como cuellos de botella o pérdidas de paquetes permiten implementar mejoras que optimizan la eficiencia y la seguridad de la red, algo muy demandado en la actualidad.

Otro de los aspectos más destacados del proyecto es la simplicidad y rapidez del proceso de captura y análisis de datos. La accesibilidad de Wireshark y la flexibilidad de Qlik Cloud han permitido implementar un análisis de red eficiente sin necesidad de una inversión significativa en recursos técnicos o financieros. Esto demuestra que cualquier organización, independientemente de su tamaño o presupuesto, puede beneficiarse de estas tecnologías para mejorar la gestión de su infraestructura de red.

El trabajo también ha puesto de manifiesto la escalabilidad de las herramientas utilizadas. Si bien el análisis se ha centrado en un entorno controlado, las mismas técnicas y procesos pueden aplicarse a redes de mayor tamaño y complejidad. Qlik Cloud, en particular, ofrece una solución flexible que puede adaptarse a las necesidades cambiantes de cualquier organización, permitiendo ampliar el análisis a un nivel mucho mayor de carga y datos, así como implementar funcionalidades de Big Data o automatizaciones inteligentes con su modelo de IA integrado.

En resumen, el proyecto ha cumplido con creces los objetivos planteados, demostrando la utilidad práctica y la eficacia de combinar Wireshark y Qlik Cloud para el análisis y visualización de datos de red. La implementación de estas herramientas proporciona conocimientos sobre el entorno muy valiosos que contribuyen a la optimización y gestión efectiva de redes informáticas, destacando la simplicidad, versatilidad y escalabilidad del proceso.

Lineas abiertas de investigación

Líneas abiertas de investigación

Algunos de los problemas que se han encontrado durante el proceso de investigación, que si bien por falta de conocimientos y tiempo no han sido resueltos de manera directa, pueden resultar muy atractivos e interesantes de cara posibles investigaciones futuras.

Entre estas dificultades se encuentra, dentro del ámbito de la captura de tráfico de red, aquella relacionada con el aislamiento completo de una conexión concreta a una dirección URL. Si bien es factible delimitar las capturas de la conexión ofrecidas por Wireshark mediante su lenguaje de filtros, esto no resulta nada sencillo. La solución más inmediata fue conseguir un entorno cerrado creado para ello, pero es probable que existan soluciones alternativas más interesantes a la hora de realizar un caso práctico real que no permita optar por la solución planteada.

Otro punto de investigación que podría resultar interesante sería abordar la problemática de la falta de soluciones de análisis y presentación de datos de forma gráfica y a gran escala dedicados exclusivamente al ámbito de la red más allá de los sencillos gráficos ofertados por el propio Wireshark y similares. La implementación de una combinación entre una herramienta de análisis de red como puede ser Wireshark y un programa de análisis de datos como puede ser Qlik Cloud, como es este caso, constituye una buena solución. Sin embargo, esto no resuelve la escasez de herramientas concretas para este objetivo, siendo el siguiente paso, el de la implementación de una herramienta conjunta o al menos dedicada, una propuesta interesante de cara a ser trabajada.

Referencias

Referencias

⁽¹⁾ Qlik Help, "Bienvenido/a a Qlik Cloud," Qlik.com.

https://help.qlik.com/es-ES/cloud-services/Content/Sense_Helpsites/Home.htm (fecha de acceso: 10 de junio de 2024)

⁽²⁾ Wireshark.org, "Chapter 1. Introduction," Guía de usuario Wireshark.

https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroWhatIs (fecha de acceso: 10 de junio de 2024)

⁽³⁾ AWS, "¿Qué es el análisis de datos?," <https://aws.amazon.com/es/what-is/data-analytics/> (fecha de acceso: 10 de junio de 2024)

⁽⁴⁾ AWS, "¿Qué es la visualización de datos?," <https://aws.amazon.com/es/what-is/data-visualization> (fecha de acceso: 10 de junio de 2024)

⁽⁵⁾ Qlik Help, "Usar cargas inline para cargar datos," Qlik.com.

https://help.qlik.com/es-ES/cloud-services/Content/Sense_Helpsites/Home.htm (fecha de acceso: 10 de junio de 2024).

⁽⁶⁾ Qlik Help, "Num - función de script y de gráfico," Qlik.com.

https://help.qlik.com/es-ES/cloud-services/Content/Sense_Helpsites/Home.htm (fecha de acceso: 10 de junio de 2024).

⁽⁷⁾ "Protocolo de enlace de 3 vías TCP (SYN, SYN-ACK, ACK)," <https://www.guru99.com/es/tcp-3-way-handshake.html> (fecha de acceso: 11 de junio de 2024).

⁽⁸⁾ Qlik Help, "SubField - función de script y de gráfico," Qlik.com. https://help.qlik.com/es-ES/cloud-services/Subsystems/Hub/Content/Sense_Hub/Scripting/StringFunctions/SubField.htm (fecha de acceso: 11 de junio de 2024).

⁽⁹⁾ Qlik Help, "Peek - función de script," Qlik.com.

https://help.qlik.com/es-ES/cloud-services/Subsystems/Hub/Content/Sense_Hub/Scripting/InterRecordFunctions/Peek.htm (fecha de acceso: 11 de junio de 2024).

⁽¹⁰⁾ Qlik Help, "RecNo - función de script," Qlik.com.

https://help.qlik.com/en-US/cloud-services/Subsystems/Hub/Content/Sense_Hub/Scripting/CounterFunctions/RecNo.htm (fecha de acceso: 11 de junio de 2024).

⁽¹¹⁾ Qlik Help, "Aggr - función de gráfico," Qlik.com.

https://help.qlik.com/es-ES/cloud-services/Subsystems/Hub/Content/Sense_Hub/ChartFunctions/aggr.htm (fecha de acceso: 11 de junio de 2024).