

UNIVERSIDAD POLITÉCNICA DE VICTORIA

DISPOSITIVO IOT PARA MONITOREO DE CARGA DE RED LOCAL CON ESP32

TESINA
QUE PARA OBTENER EL GRADO DE
INGENIERÍA EN TECNOLOGÍAS DE LA
INFORMACIÓN

PRESENTA:
GLORIA PATRICIA MARSEL ACOSTA HEREDIA

DIRECTOR
DR. JORGE ARTURO HERNÁNDEZ ALMAZÁN

CO-DIRECTOR
DR. SAID P. MALDONADO

ORGANISMO RECEPTOR
UNIVERSIDAD POLITÉCNICA DE VICTORIA
CIUDAD VICTORIA, TAMAULIPAS, ENERO 2024



Ciudad Victoria,
Tamaulipas, a
**19 de Enero de
2024**

UNIVERSIDAD POLITÉCNICA DE VICTORIA
DR. SAID POLANCO MARTAGON
PRESENTE



La Universidad Politécnica de Victoria tiene a bien presentar a **ACOSTA HEREDIA GLORIA PATRICIA MARSEL** estudiante del programa académico de **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**, con número de matrícula **1930432** y seguro facultativo IMSS número **68150001862**; quien deberá realizar su práctica profesional de **ESTADÍA**, a partir del **08 de Enero de 2024 al 12 de Abril de 2024**, con duración de **600 horas** desarrollando el proyecto "**Dispositivo IoT para Monitoreo de Carga de Red Local con ESP32**" que le fué asignado.

Al concluir se le extenderá la carta de liberación al evaluarlo satisfactoriamente y además le solicitaremos su valiosa opinión respondiendo el formulario que se le enviará por mail, referente al desempeño del practicante, la información que nos proporcione, es de vital importancia para la mejora de los programas académicos que ofrece la Universidad Politécnica de Victoria para la formación de profesionistas altamente especializados.

El practicante deberá cumplir con el Reglamento Interno aplicable al personal en su centro de trabajo.

Sin otro particular.

ATENTAMENTE

M. A. OTHÓN CANO GARZA
DIRECTOR DE VINCULACIÓN

C.C.P. JORGE ARTURO HERNÁNDEZ ALMAZÁN
ASESOR INSTITUCIONAL



UNIVERSIDAD POLITÉCNICA DE VICTORIA

Av. Nuevas Tecnologías 5902
Parque Científico y Tecnológico de Tamaulipas
Carretera Victoria Soto La Marina Km. 5.5
Cd. Victoria, Tamaulipas. C.P. 87138

Tel: (834) 1711100 al 10
www.upvictoria.edu.mx



UNIVERSIDAD POLITÉCNICA DE VICTORIA

Victoria Tamaulipas, a 11 de Abril del 2024
Asunto: Carta de Aceptación

M.A OTHÓN CANO GARZA
DIRECTOR DE VINCULACIÓN
UNIVERSIDAD POLITÉCNICA DE VICTORIA
PRESENTE

Hacemos de su conocimiento que hemos aceptado al estudiante **ACOSTA HEREDIA GLORIA PATRICIA MARSEL** del programa académico de **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**, con número de matrícula **1930432** de la Universidad Politécnica de Victoria, para realizar su **ESTADÍA** en nuestra empresa **UNIVERSIDAD POLITÉCNICA DE VICTORIA**, durante el periodo comprendido del día **08 de Enero de 2024** al **12 de Abril de 2024**, el estudiante estará colaborando en el proyecto "**Dispositivo IoT para Monitoreo de Carga de Red Local con ESP32**" con una carga horaria total de **600 horas**

Al concluir satisfactoriamente sus prácticas profesionales, se le entregará al estudiante su carta de liberación debidamente formalizada.

Con el objetivo de colaborar en la mejora de los programas académicos de la Universidad Politécnica de Victoria, estamos de acuerdo en responder a la brevedad posible el formulario de evaluación del desempeño del practicante previo a emitir la liberación de **ESTADÍA**.

Sin otro particular.


ATENTAMENTE
DR. SAID POLANCO MARTAGON
ASESOR EMPRESARIAL



UNIVERSIDAD POLITÉCNICA DE VICTORIA

Victoria Tamaulipas, a 12 de Abril del 2024
Asunto: Carta de Liberación

M.A OTHÓN CANO GARZA
DIRECTOR DE VINCULACIÓN
UNIVERSIDAD POLITÉCNICA DE VICTORIA
PRESENTE

Por medio de la presente me permito comunicarle que el estudiante **ACOSTA HEREDIA GLORIA PATRICIA MARSEL** del programa académico de **INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN**, con número de matrícula **1930432** de la Universidad Politécnica de Victoria, terminó satisfactoriamente su **ESTADÍA** desarrollando trabajos y actividades directamente relacionadas con el proyecto **Dispositivo IoT para Monitoreo de Carga de Red Local con ESP32**, con una duración de **600 horas**, en el periodo comprendido del **08 de Enero de 2024 al 12 de Abril de 2024**.

Sin otro particular por el momento, hago propicia la ocasión para enviarle un cordial saludo.

ATENTAMENTE
DR. SAID POLANCO MARTAGON
ASESOR EMPRESARIAL





CARTA DE ACEPTACIÓN DEL DOCUMENTO PARA SU IMPRESIÓN

Cd. Victoria, Tamaulipas a FEcha de carta

Gloria Patricia Marsel Acosta Heredia
PRESENTE

Le comunico que el Programa Académico de Ingeniería en Tecnologías de la Información le ha otorgado la autorización para la impresión de su Tesina de Estadía Práctica cuyo título es:

Dispositivo IoT para Monitoreo de Carga de Red Local con ESP32
ATENTAMENTE

Dr. Jorge Arturo Hernández Almazán
ASESOR INSTITUCIONAL

c.c.p Director de programa académico

UNIVERSIDAD POLÍTÉCNICA DE VICTORIA

Av. Nuevas Tecnologías 5902
Parque Científico y Tecnológico de Tamaulipas
Carretera Victoria Soto La Marina Km. 5.5
Cd. Victoria, Tamaulipas. C.P. 87138

Tel: (834) 1711100 al 10
www.upvictoria.edu.mx



EVALUACIÓN DE ESTADÍA

Rúbrica para evaluación de la presentación y el reporte de estadía

 Nombre del alumno: **GLORIA PATRICIA MARSEL ACOSTA HEREDIA**

Calificación final: _____

 Periodo: **ENERO-ABRIL 2024**

Ponderación	Aspecto a Evaluar	Competente 10	Independiente 9	Básico Avanzado 8	No Competente 5
40	Resultados y Actividades	Estrechamente relacionados al perfil de egreso de su programa académico	Parcialmente relacionados al perfil de egreso de su programa académico	Escasamente relacionados al perfil de egreso de su programa académico	Escasamente relacionados al perfil de egreso de su programa académico
30	Exposición de las actividades de la estadía	Detalladas y sustentadas con respecto a los resultados que se obtuvieron	Detalladas y sustentadas parcialmente con respecto a los resultados que se obtuvieron	Detalladas parcialmente con respecto a los resultados que se obtuvieron	Detalladas escasamente con respecto a los resultados que se obtuvieron
10	Material visual Lenguaje verbal	Uso el lenguaje y la terminología apropiadas; El material visual está organizado, adecuado y suficiente	Uso el lenguaje y la terminología apropiadas El material visual está parcialmente organizado y es suficiente	Uso el lenguaje y la terminología son parcialmente apropiadas; El material visual está parcialmente organizado y es suficiente	Uso el lenguaje y terminología es inapropiado; El material visual no está organizado y es insuficiente
10	Exposición en Idioma Inglés	Pronunciation is clear so language is easily understood (2.5) Uses fluent connected speech, occasionally disrupted by search for correct form of expression (2.5) Uses topic related vocabulary without problems (2.5) Responds to questions using varied and descriptive vocabulary and language structures (2.5)	Pronunciation is understandable, but there are slight errors (2.25) Speech is connected but frequently disrupted by search for correct form of expression (2.25) Uses some topic related vocabulary sufficient to communicate ideas (2.25) Responds to questions using simple but accurate vocabulary and language structures (2.25)	Pronunciation is understandable most of the time, marked native accent and many errors (2) Speaks with simple sentences, sometimes not connected, but is understood (2) Uses basic vocabulary to communicate ideas (2) Partly responds to simple questions, with limited vocabulary and language structures (2)	Pronunciation makes language very difficult to understand (1) Uses one-word/two-word utterances (1) Unable to communicate ideas due to lack of vocabulary (1) Uses isolated words or sentence fragments to respond to questions (1)
5	Respuesta a los cuestionamientos de los evaluadores	Clara y satisfactoria	Clara y parcialmente satisfactoria	Clara e insuficiente	Confusa e insuficiente
5	Autorización de tesina en tiempo y forma	Presenta en tiempo y forma	Presenta en tiempo y forma con la mayoría de requerimientos solicitados	Presenta en tiempo y con algunas limitantes de los requerimientos solicitados.	Presenta fuera de tiempo y con los mínimos requerimientos solicitados.

Dr. Jorge Arturo Hernández Almazán
 ASESOR INSTITUCIONAL

Dr. Héctor Hugo Avilés Arriaga
 EVALUADOR

Dr. Héctor Hugo Avilés Arriaga
 EVALUADOR DE INGLÉS



REGISTRO DE EVALUACIÓN DE EXPOSICIÓN DE ESTADÍA

Siendo las 09:00 horas del día 15 de Abril del 2024, el alumno **Gloria Patricia Marsel Acosta Heredia**, del programa académico **Ingeniería en Tecnologías de la Información**, con matrícula **1930432**, presentó la exposición de la estadía realizada durante el cuatrimestre **Enero-Abril 2024**, en la **Universidad Politécnica de Victoria**, con el proyecto titulado **Dispositivo IoT para Monitoreo de Carga de Red Local con ESP32**.

Una vez concluido el proceso de evaluación, y con base a la rúbrica establecida para éste propósito, se determina que la calificación de la estadía es _____.

Dr. Jorge Arturo Hernández Almazán
ASESOR INSTITUCIONAL

Dr. Héctor Hugo Avilés Arriaga
EVALUADOR

Dr. Héctor Hugo Avilés Arriaga
EVALUADOR DE INGLÉS

Agradecimientos

Quiero agradecer a mi familia por su amor sin límites y su apoyo firme. Agradezco a mis padres, Oscar y Alma por su sacrificio y dedicación, y a mis hermanos Sophia y Brunopor su comprensión y constante ánimo. Agredezco a mis mascotas, Theodora,Benjamin y Bruna por hacerme inmensurablemente feliz solo por existir Estoy realmente agradecido con mis amigos y compañeros de clase por su apoyo moral y aliento durante los momentos difíciles.

Gracias a todos ustedes.

Resumen

Este proyecto se enfoca en la creación de un dispositivo IoT utilizando el ESP32 para realizar un monitoreo eficiente de la carga en una red local. El dispositivo recopila datos en tiempo real sobre el tráfico de red, analiza la carga y proporciona retroalimentación visual y auditiva basada en los niveles detectados. Destacan características como el monitoreo constante en tiempo real, alertas visibles y auditivas, interfaz gráfica vista desde Grafana para representación visual detallada, configuración personalizada de umbrales de carga, conectividad IoT para acceso remoto a datos, y la optimización de recursos para mejorar el rendimiento de la red.

Palabras clave: Dispositivo IoT,ESP32,Monitoreo eficiente,Retroalimentación visual y auditiva,Conectividad IoT.

Summary

This project focuses on the creation of an IoT device using ESP32 to perform efficient load monitoring in a local network. The device collects real-time data about network traffic, analyzes the load, and provides visual and audible feedback based on detected levels. Features such as constant real-time monitoring, alerts real-time constant monitoring, visible and audible alerts, graphical interface on TFT displays for detailed visual representation, customizable threshold settings detailed visual representation, customized configuration of load thresholds, IoT connectivity for remote data access, and resource optimization to improve network performance.

Keywords: IoT device,ESP32,Efficient monitoring,Visual and auditory feedback,IoT connectivity.

Índice

Agradecimientos	VII
Resumen	VIII
Summary	IX
Índice	X
1. Introducción.	1
1.1. Antecedentes	1
1.2. Definición del problema y justificación del proyecto.	1
1.3. Objetivo General.	2
1.4. Objetivo Particular.	2
1.5. Alcances y limitaciones del Proyecto.	3
1.6. Organización del Documento de Tesina.	3
2. Marco Teórico.	4
2.1. Monitoreo de trafico de red.	4
2.1.1. Razones para usar el monitoreo de tráfico de red.	4
2.1.2. Tecnicas de analisis de traficos.	4
2.2. pfSense	7
2.2.1. Caracteristicas.	7
2.3. Ntopng.	8
2.3.1. Características.	8
2.3.2. NTop vs. NTopng.	9
2.4. ESP 32.	12
2.4.1. Características.	12
2.5. Raspberry Pi 3 Model B vi. 2	13
2.5.1. Características.	13
2.6. InfluxDB.	14
2.6.1. Características.	14
2.6.2. Diferencias entre InfluxDB v1 y InfluxDB v2.	14
2.7. Grafana.	16
2.7.1. Características.	16
2.8. Herramientas y tecnologías utilizadas.	16
2.8.1. Lenguaje C.	16
2.8.2. Raspberry Imager.	16
2.8.3. Extencion SSH.	17
2.8.4. Arduino IDE.	17
2.8.5. InfluxDbClient.	18
3. Trabajo relacionados.	19
3.1. Implementación de un servidor como gestión y monitoreo de servicios para la red de datos en la UGEL Huamanga, 2018	19

3.2. Monitoreo de datos mediante un administrador de flujo de datos, 2013	21
3.3. Inteligencia artificial para el control de tráfico en redes de datos : Una Revisión,2021	23
3.4. Analysis of centralized computer security systems through the alienvault ossim tool,2022	24
3.5. Evaluación de herramientas TIC para gestionar el monitoreo y análisis de la red de datos del Recinto de Golfito de la Universidad de Costa Rica,2020	25
3.6. Nodo de gestión y monitoreo de calidad de servicio de la empresa Ajnet en el cantón Latacunga,2023	27
4. Sistema Propuesto.	28
4.1. Instalación y configuración de los software de monitoreo de trafico.	29
4.1.1. Instalación y configuración de Pfsense.	29
4.1.2. Instalación y configuración de Ntopng.	33
4.2. Consulta de los reportes de tráfico.	36
4.3. Configuración de la Raspberry.	38
4.4. Instalación de Grafana y Influx.	39
4.5. Conectar Influx con NTopng.	39
4.5.1. Configuración Influx con Grafana.	41
4.6. Pruebas.	43
4.6.1. Comunicación entre InfluxDB y el ESP32.	43
5. Implementación del Sistema y Pruebas.	44
5.1. Designación de Hardware y Software utilizado.	44
5.2. Despliegue y ejecución del sistema.	44
5.3. Pruebas.	45
5.3.1. Pruebas de conexión a internet.	45
5.3.2. Pruebas de conexión a Influxdb: Tabla Host:traffic.	45
5.3.3. Pruebas de conexión a Influx: Host que utiliza más recursos.	46
5.4. Resultados.	48
5.4.1. Ntop con Influx.	48
5.4.2. Influx con Grafana.	48
5.4.3. Consulta Influx desde ESP32.	49
5.4.4. Prueba de alarma.	50
6. Conclusiones y Trabajo Futuro.	52
6.1. Conclusiones.	52
6.2. Trabajo Futuro	52
Índice de figuras	53
Índice de cuadros	55
Referencias	56

1. Introducción.

En un entorno cada vez más dependiente de la conectividad y la tecnología, la estabilidad y eficacia de los sistemas de red se vuelven fundamentales para el funcionamiento fluido de cualquier institución. En este contexto, surge la necesidad de abordar y mejorar la gestión de estos sistemas, optimizando su rendimiento y capacidad de respuesta ante posibles contratiempos.

1.1. Antecedentes

En el panorama tecnológico actual, la estabilidad y el rendimiento de los sistemas de conexión a redes son aspectos críticos para el funcionamiento eficiente de cualquier institución. La capacidad de mantener estos sistemas estables y seguros refleja directamente la eficacia del trabajo de automatización de alarmas y la gestión del sistema de red en su conjunto. Este proyecto fue abordado inicialmente por el departamento de sistemas, en colaboración con el Dr. Said Maldonado, con el objetivo de mejorar la infraestructura de red existente.

En una primera etapa, se implementó un sistema basado en Telegraf y pfSense. Sin embargo, se identificó que Telegraf proporcionaba datos de manera desorganizada, lo que dificultaba la detección y resolución eficiente de problemas. Esta limitación planteó la necesidad de explorar nuevas soluciones para mejorar la eficiencia y capacidad de respuesta del sistema.

Con el propósito de superar estos desafíos, se tomó la decisión de implementar ntop y desarrollar este proyecto. El objetivo principal de esta iniciativa es optimizar la gestión de la red, proporcionando una mayor capacidad de resolución de problemas y mejorando la eficiencia operativa en la institución.

1.2. Definición del problema y justificación del proyecto.

La estructura de la red de la Universidad Politécnica de Victoria ha experimentado una expansión a lo largo de los años, llegando a estar conformada actualmente por cinco edificios en todo el campus. Este crecimiento ha sido acompañado por un aumento en el consumo de los servicios de red debido a la población estudiantil, lo que ha generado una considerable congestión en el tráfico, saturando el ancho de banda. Esto ha resultado en la prestación de un servicio de baja calidad y, en casos extremos, ha provocado la saturación de la red durante períodos indefinidos, haciendo que sea imposible acceder a la misma, lo cual es insatisfactorio tanto para los trabajadores como para los estudiantes que requieren una conexión eficiente.

Previamente se identificó que la fuente principal de este problema son usuarios que emplean una gran cantidad de recursos de la red, ya sea consciente o inconscientemente. Dentro de los casos de inconsciencia, se ha detectado que un porcentaje de estos alumnos o personal de la universidad tienen malware en sus equipos de trabajo, lo que genera múltiples peticiones al servidor, saturándolo. Para abordar este problema, el departamento de sistemas resolvía la situación manualmente, aislando a estos usuarios de la red mediante la identificación de la dirección IP del equipo y bloqueando el acceso a los dispositivos que estuvieran consumiendo la red.

Se propone implementar un sistema de alarma que notifique la presencia de usuarios que estén saturando la red, utilizando la plataforma NTopng, un software de monitoreo de redes que proporciona esta información en intervalos de tiempo definidos en un archivo JSON, que almacena el estado de las peticiones y los bytes enviados y recibidos tanto por TCP como por UDP.

1.3. Objetivo General.

En primera instancia, se busca desarrollar y presentar el primer prototipo de la alarma para el departamento de sistemas. Para llevar a cabo este objetivo, se utilizará la herramienta de monitoreo de red NTopng, así como el firewall PFsense y una placa ESP32. Se llevará a cabo un estudio detallado de la implementación de NTopng para monitorear una red y enviar datos a InfluxDB para su procesamiento en una Raspberry Pi 3 Model B.

Este prototipo servirá como una prueba inicial para evaluar la viabilidad y efectividad de NTopng como herramienta de monitoreo de red en el contexto de la necesidad de identificar a los usuarios que están saturando la red. Posteriormente, se llevará a cabo una prueba de prototipo para validar que nuestro dispositivo pueda monitorear un número definido de usuarios y evaluar el desempeño de la red en uso mientras los usuarios realizan sus tareas cotidianas. Además, se buscará detectar cualquier potencial amenaza para la red, identificándola en un display la dirección IP potencialmente peligrosa junto con una descripción del equipo. En un futuro, se plantea también identificar el tipo de problema que tenga el equipo e imprimirla en el display.

1.4. Objetivo Particular.

- Monitoreo de Tráfico en Tiempo Real: Utilizar NTopng para monitorear el tráfico de red en tiempo real y detectar patrones de saturación o comportamientos anómalos.
- Acceso Remoto a la Información de NTopng: Utilizar la Raspberry Pi para acceder de forma remota a los archivos JSON generados por NTopng, permitiendo al ESP32 obtener información actualizada sobre el estado de la red.
- Identificación de Usuarios Problemáticos: Configurar NTopng para identificar a los usuarios que estén saturando la red mediante un análisis detallado de sus actividades y consumo de ancho de banda.
- Generación de Alertas Automáticas: Implementar un mecanismo en NTopng para generar alertas automáticas cuando se detecte una saturación de red o la actividad de usuarios problemáticos, y enviar estas alertas al sistema de alarma.
- Visualización de Alertas en el Dispositivo ESP32: Configurar el ESP32 para recibir y mostrar las alertas generadas por NTopng en un display, proporcionando una interfaz visual para el monitoreo de la red.

1.5. Alcances y limitaciones del Proyecto.

En esta fase inicial del proyecto, se desarrollará un prototipo de monitoreo de la carga de la red local de la Universidad Politécnica de Victoria. Este prototipo tiene como objetivo:

- Desarrollo del Prototipo: Incluye el diseño, desarrollo e implementación de un sistema de alarma funcional utilizando NTopng, PFsense y una placa ESP32.
- Monitoreo de Red: El sistema será capaz de monitorear el tráfico de red en tiempo real utilizando NTopng para detectar patrones de saturación y usuarios problemáticos.
- Generación de Alertas: Se implementará un mecanismo para generar alertas automáticas cuando se detecte una saturación de red o actividad anómala de usuarios. Estas alertas se visualizarán en un display y se activarán a través de leds conectados al ESP32 que conforman la alarma.
- Acceso Remoto a la Información: Se garantizará el acceso remoto a la información de monitoreo mediante una Raspberry Pi 3 Model B del 2015, la cual proporcionará los datos necesarios para el sistema de alarma.
- Pruebas y Evaluación: Se llevarán a cabo pruebas exhaustivas del prototipo en un entorno controlado para evaluar su efectividad en la detección y respuesta a problemas de saturación de red y usuarios problemáticos.

Este enfoque integra el desarrollo del proyecto con las especificaciones de funcionalidad de una primera versión del dispositivo de alarma. El proyecto se centra en el desarrollo de un dispositivo que implementa NTopng y PFsense para el monitoreo, junto con la placa ESP32 para el hardware de la alarma, incluyendo un display para mostrar la dirección IP del equipo problemático y leds para la alarma. Para garantizar el acceso remoto a la información, se emplea la Raspberry Pi modelo 3 como intermediario mediante Influx.

1.6. Organización del Documento de Tesina.

En los próximos capítulos del presente documento se desglosa el capítulo 2, el cual se refiere al marco teórico referente a los conceptos que son precisos tener en cuenta para la correcta comprensión del contenido consiguiente. El capítulo 3 consta de las técnicas y herramientas implementadas a lo largo del desarrollo del presente proyecto, detallando los procesos y explicando detenidamente lo utilizado para la conclusión del mismo. Para el capítulo 4 se muestra el resultado obtenido tras el desarrollo realizado, mediante la experimentación se muestran diferentes escenarios y respuestas obtenidas. Finalmente en el capítulo 5 se dan las conclusiones consecuentes de la experimentación y testeo de lo desarrollado, además se ofrece trabajo futuro que aporta mejoras al proyecto de primera instancia.

2. Marco Teórico.

En este capítulo se abordan los conceptos y definiciones más importantes para el mejor entendimiento de este proyecto, tendrá terminologías del procesamiento de investigación para llevar a cabo el dispositivo, a fin de que sea de la mejor comprensión para el lector.

2.1. Monitoreo de tráfico de red.

El monitoreo de tráfico de red es una práctica fundamental en la gestión de redes que consiste en observar y analizar el flujo de datos que circula a través de una red de computadoras[1]. Esta actividad se realiza utilizando herramientas especializadas que recopilan datos sobre el tráfico de red en tiempo real y proporcionan información detallada sobre su uso y rendimiento.

2.1.1. Razones para usar el monitoreo de tráfico de red.

El monitoreo de tráfico de red se realiza por varias razones, entre las que se incluyen:

- Diagnóstico de problemas de red: El monitoreo de tráfico de red permite identificar y diagnosticar problemas potenciales, como cuellos de botella, congestión de red, pérdida de paquetes y latencia. Esto es crucial para mantener un rendimiento óptimo de la red y garantizar la disponibilidad de los servicios[1].
- Optimización del rendimiento: Al analizar el tráfico de red, los administradores pueden identificar áreas de la red que están subutilizadas o que podrían beneficiarse de una mayor capacidad. Esto les permite tomar decisiones informadas sobre cómo mejorar el rendimiento de la red y asignar recursos de manera más eficiente.[1]
- Seguridad de red: El monitoreo de tráfico de red es una herramienta importante para detectar actividades maliciosas o no autorizadas, como intrusiones, ataques de denegación de servicio (DDoS) y tráfico de malware. Al identificar y responder rápidamente a estas amenazas, los administradores pueden proteger la integridad y la seguridad de la red y los datos que transitan por ella.[1]
- Cumplimiento normativo: Muchas organizaciones están sujetas a regulaciones y normativas que requieren la monitorización y el registro del tráfico de red, especialmente en sectores como la salud, las finanzas y el gobierno. El monitoreo de tráfico de red ayuda a cumplir con estos requisitos y proporciona registros detallados de la actividad de la red que pueden ser auditados en caso necesario.

2.1.2. Técnicas de análisis de tráficos.

El análisis y monitoreo del tráfico de red son aspectos fundamentales para comprender el funcionamiento y la seguridad de una red de computadoras. En este artículo, exploraremos algunas técnicas comunes utilizadas para este propósito.

- Captura de Paquetes: La captura de paquetes es una técnica que implica la observación y registro de paquetes de datos que se transmiten a través de la red.

Herramientas como Wireshark o tcpdump permiten capturar y analizar estos paquetes para obtener información detallada sobre el tráfico de red.

- Análisis de Flujo: El análisis de flujo se centra en el tráfico agregado entre puntos de la red.
Herramientas como NetFlow, sFlow o IPFIX recolectan datos de flujo que describen el tráfico de red de manera resumida, facilitando el monitoreo del tráfico y la detección de anomalías.
- Inspección Profunda de Paquetes (DPI): La inspección profunda de paquetes implica examinar el contenido de los paquetes de datos para obtener información más detallada sobre el tráfico de red.
DPI puede utilizarse para identificar aplicaciones específicas, así como para detectar amenazas como malware o actividades maliciosas.
- Análisis de Comportamiento: El análisis de comportamiento implica el monitoreo continuo del tráfico de red para identificar patrones y tendencias de comportamiento normales.
Cualquier desviación significativa de estos patrones puede indicar problemas de rendimiento o actividades sospechosas.
- Monitorización de Rendimiento: La monitorización de rendimiento implica el seguimiento de métricas como ancho de banda utilizado, latencia y pérdida de paquetes.
Herramientas como Nagios, Zabbix o Cacti permiten monitorear estos parámetros y generar alertas cuando se superan umbrales predefinidos.

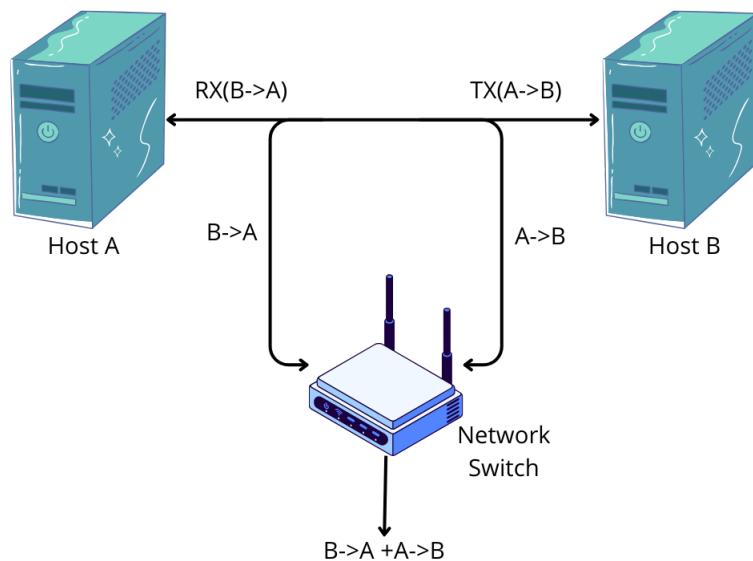
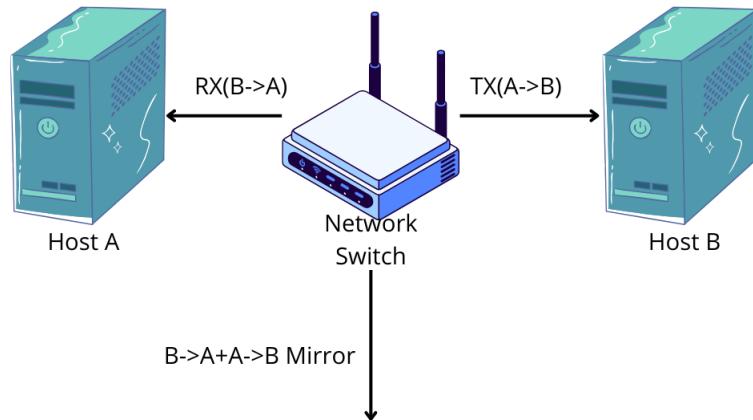


Figura 1: Conexión de dispositivos totalmente activos y pasivos correspondientes.

El monitoreo de tráfico de red es una práctica esencial para garantizar un rendimiento óptimo, una seguridad robusta y el cumplimiento normativo de las redes de computadoras. Permite a los administradores de red obtener información valiosa sobre el uso y el comportamiento de la red, lo que les permite tomar decisiones informadas y mantener la red en funcionamiento de manera eficiente.

2.2. pfSense

pfSense es una distribución de software de código abierto basada en FreeBSD que se utiliza como firewall y enrutador de red. Proporciona una plataforma robusta y personalizable para la gestión y protección de redes informáticas.

2.2.1. Características.

Las características principales de pfSense incluyen:

- Firewall: Ofrece capacidades avanzadas de filtrado de paquetes y reglas de firewall para proteger las redes contra amenazas externas e internas [2].
- Enrutamiento: Funciona como un enrutador de red, permitiendo dirigir el tráfico entre redes locales y externas, así como la configuración de rutas estáticas y dinámicas.
- VPN (Red Privada Virtual): Ofrece soporte para diferentes tipos de VPN, como IPSec, OpenVPN y L2TP/IPSec, permitiendo a los usuarios establecer conexiones seguras entre redes o dispositivos remotos[3].
- Proxy: Proporciona funcionalidades de proxy para mejorar el rendimiento y la seguridad al acceder a recursos en línea.
- Balanceo de Carga: Permite distribuir el tráfico de red entre múltiples enlaces de Internet para optimizar el rendimiento y la disponibilidad de la red. Servicios de Red:** Incluye una amplia gama de servicios de red, como DHCP, DNS, NTP y SNMP, para facilitar la administración y el funcionamiento de la red.[4]

PfSense es una solución de software de red versátil y potente que se utiliza ampliamente en entornos empresariales y domésticos para proteger, gestionar y optimizar el tráfico de redes informáticas.

2.3. Ntopng.

NTopng (Network Top Next Generation) es una herramienta de monitoreo de red de código abierto que proporciona una visión detallada y en tiempo real del tráfico de una red como se muestra en la imagen 4. Es una evolución de su predecesor, NTop, y ha sido diseñado para ofrecer características mejoradas y una interfaz de usuario más moderna[5].

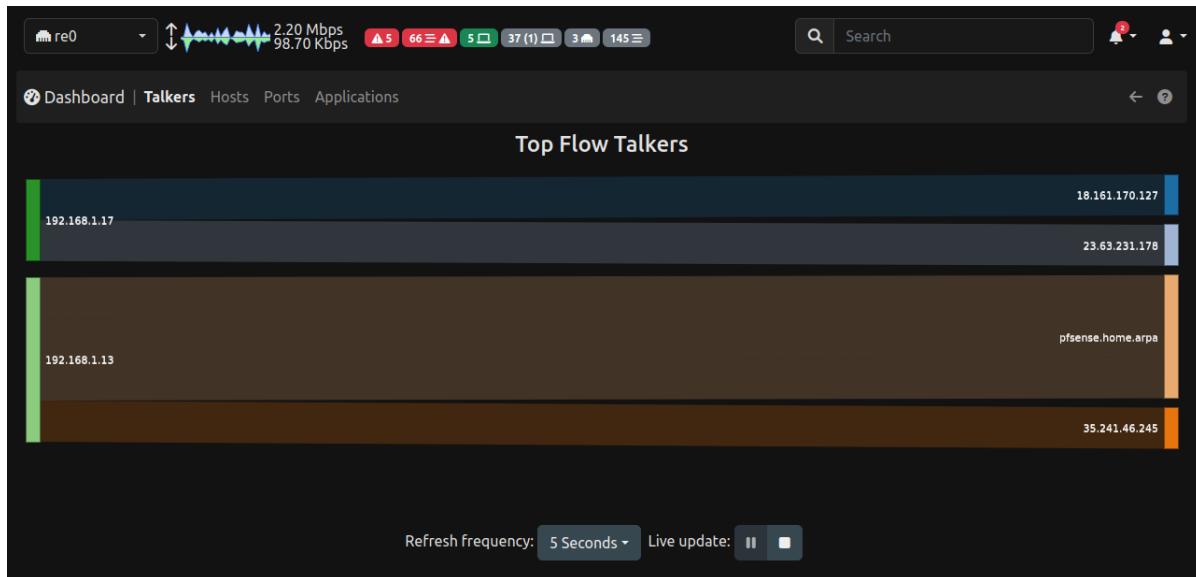


Figura 2: Versión de Ntopng utilizada en el proyecto.

2.3.1. Características.

Las principales características de NTopng incluyen:

- **Monitoreo en Tiempo Real:** Proporciona una visualización en tiempo real del tráfico de red, permitiendo a los administradores de red observar la actividad de la red de manera instantánea[5].
- **Análisis Profundo del Tráfico:** Ofrece análisis detallados del tráfico, incluyendo estadísticas sobre los protocolos utilizados, las direcciones IP, los servicios y las conexiones activas.
- **Interfaz Web Intuitiva:** NTopng presenta una interfaz de usuario web moderna y fácil de usar, que permite a los administradores acceder a la información de monitoreo desde cualquier lugar con conexión a la red[5].
- **Detección de Usuarios y Dispositivos:** Identifica y clasifica usuarios y dispositivos en la red, proporcionando información detallada sobre su actividad y su consumo de ancho de banda.
- **Generación de Informes:** Permite generar informes personalizados sobre el tráfico de

red y el rendimiento, facilitando el análisis y la presentación de datos para informes de auditoría o revisión[5].

- Integración con Otras Herramientas: Puede integrarse con otras herramientas y sistemas, facilitando la colaboración y la interoperabilidad en el entorno de red.

NTopng es útil para administradores de red, analistas de seguridad y profesionales de TI que buscan comprender y gestionar el tráfico de sus redes de manera eficiente. Al proporcionar información detallada y en tiempo real, NTOPNG contribuye a la identificación rápida de problemas, la optimización del rendimiento y la seguridad de la red.

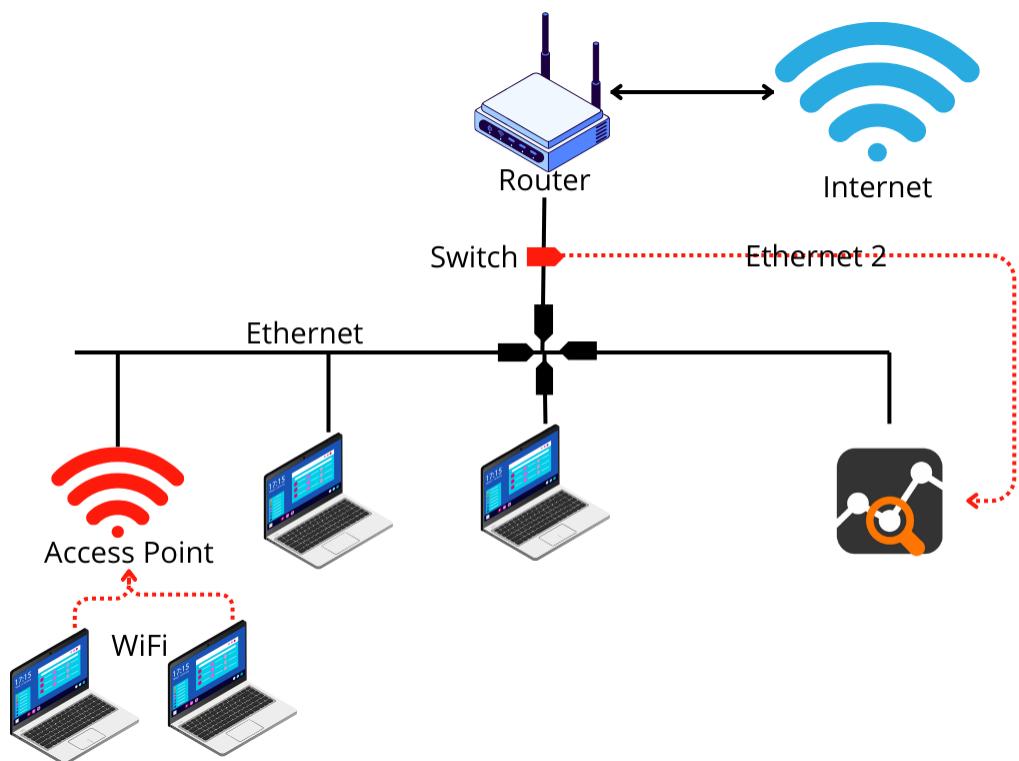


Figura 3: Demostración general de cómo funciona Ntop.

2.3.2. NTop vs. NTopng.

NTop (Network Top) es una herramienta de código abierto diseñada para el monitoreo de redes. Proporciona información detallada sobre el tráfico de red, incluyendo estadísticas sobre protocolos utilizados, direcciones IP, hosts y más. NTop se utiliza para analizar el tráfico de red en tiempo real y generar informes detallados sobre el rendimiento y el uso de la red, en la siguiente imagen podemos ver que entre un sistema Ntop funcionando [22](#).

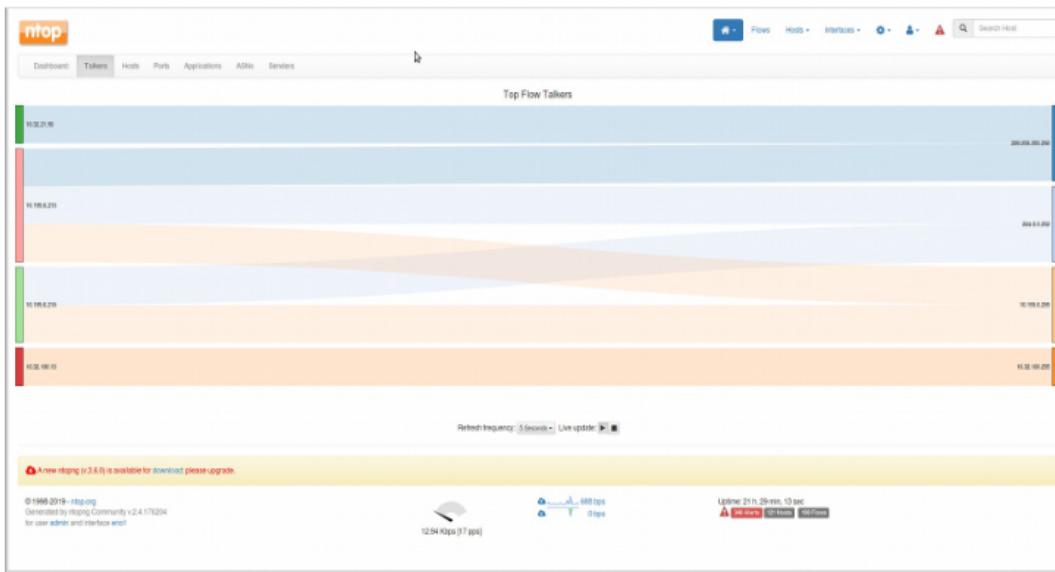


Figura 4: Versión de NTopng del proyecto de Evaluación de herramientas TIC para gestionar el monitoreo y análisis de la red de datos del Recinto de Golfito de la Universidad de Costa Rica.

Por otro lado, NTopng (Network Top Next Generation) es una versión mejorada y actualizada de NTop. NTopng es una aplicación de monitoreo de tráfico basada en web capaz de:

- Monitoreo pasivo del tráfico mediante la captura pasiva del tráfico de red
- Recopilar flujos de red (NetFlow, sFlow e IPFIX)
- Supervise activamente los dispositivos de red seleccionados
- Monitorear una infraestructura de red vía SNMP

La principal diferencia entre ntopng y un recopilador de tráfico es que ntopng no solo informa estadísticas de tráfico, además saca conclusiones sobre el tipo de tráfico observado e informa métricas de ciberseguridad. Ambas herramientas son ampliamente utilizadas en entornos de redes para monitorear y analizar el tráfico, lo que permite a los administradores de red tomar decisiones informadas y mantener la seguridad y el rendimiento de la red.

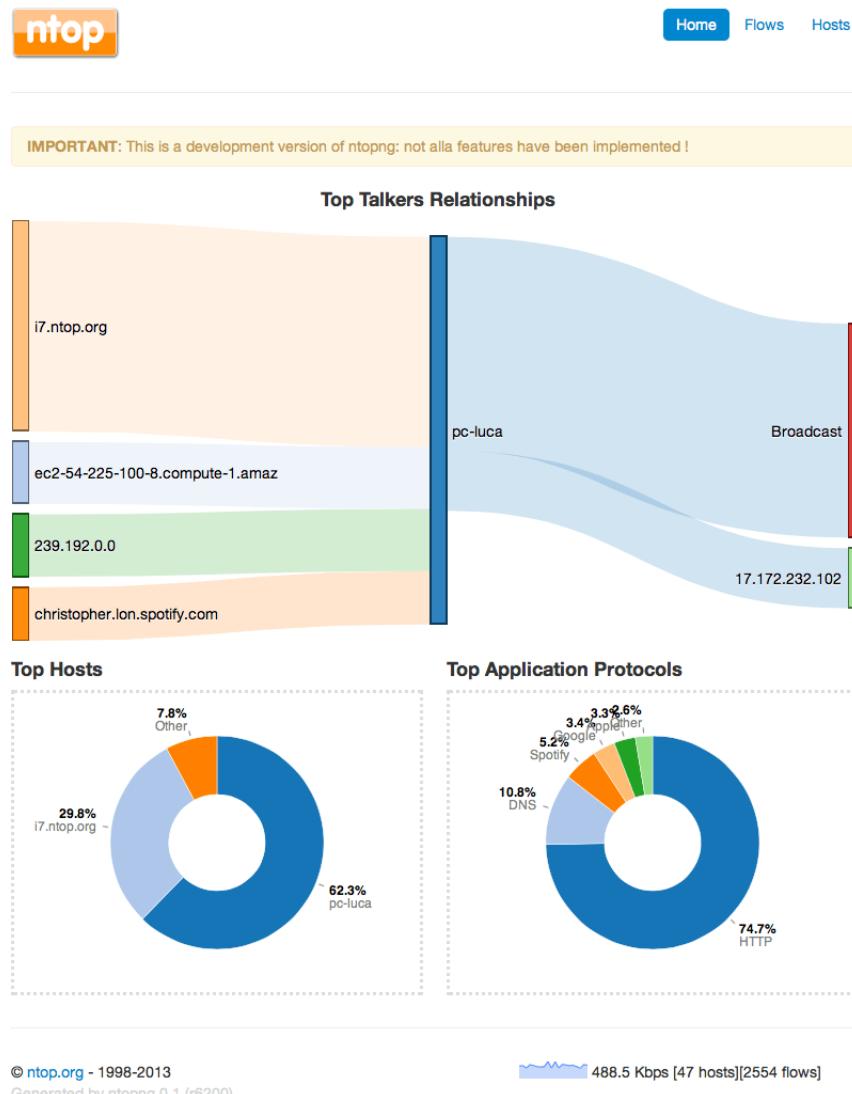


Figura 5: Versión de Ntop donde se muestra la forma de reporte del Top Talkers Relationship.

2.4. ESP 32.

La ESP32 es un microcontrolador de bajo costo y alto rendimiento diseñado por Espressif Systems. Es una evolución de la popular ESP8266 y ofrece una amplia gama de características y capacidades para proyectos de Internet de las cosas (IoT) y aplicaciones embebidas^[6].

2.4.1. Características.

Algunas características destacadas de la ESP32 incluyen:

- Doble núcleo: La ESP32 cuenta con dos núcleos de procesador Xtensa LX6, lo que permite ejecutar múltiples tareas de manera simultánea y mejorar el rendimiento general del dispositivo^[6].
- Conectividad Wi-Fi y Bluetooth: Integra módulos de Wi-Fi 802.11 b/g/n y Bluetooth v4.2 BR/EDR y BLE, lo que permite la conexión a redes inalámbricas y la comunicación con dispositivos compatibles.
- Bajo consumo de energía: La ESP32 está diseñada para ofrecer un bajo consumo de energía, lo que la hace ideal para aplicaciones alimentadas por batería o energía solar^[7].
- Amplia gama de periféricos: Cuenta con una variedad de interfaces periféricas, incluyendo UART, SPI, I2C, I2S, ADC y DAC, lo que facilita la conexión con una amplia variedad de sensores, actuadores y dispositivos externos.
- Seguridad integrada: Ofrece características de seguridad integradas, como soporte para criptografía AES, RSA, SHA y ECC, así como un generador de números aleatorios seguro (RNG), que ayudan a proteger las comunicaciones y los datos del dispositivo.
- Soporte para desarrollo: Es compatible con una amplia gama de entornos de desarrollo, incluyendo el popular entorno de desarrollo integrado (IDE) Arduino, así como el framework de desarrollo ESP-IDF de Espressif, que proporciona acceso a herramientas y bibliotecas avanzadas para el desarrollo de aplicaciones.

La ESP32 es una poderosa plataforma de desarrollo que ofrece una combinación única de características y capacidades para una amplia variedad de aplicaciones IoT y embebidas^[7]. Su versatilidad, rendimiento y bajo costo la convierten en una opción popular para proyectos de hardware de código abierto y comerciales.

2.5. Raspberry Pi 3 Model B vi. 2

La Raspberry Pi 3 Model B v2 es una versión mejorada del popular miniordenador de placa única Raspberry Pi 3 Model B. Tiene un conjunto de características similares, pero con algunas mejoras y actualizaciones.

2.5.1. Características.

Aquí hay algunas características de la Raspberry Pi 3 Model B v2:

- Procesador: Incorpora un procesador de cuatro núcleos ARM Cortex-A53 de 64 bits con una velocidad de reloj de 1.4 GHz, que ofrece un rendimiento mejorado en comparación con las versiones anteriores[8].
- Memoria RAM: Tiene 1 GB de memoria RAM LPDDR2 integrada, lo que permite una multitarea más fluida y un mejor rendimiento general del sistema[8].
- Conectividad inalámbrica: Ofrece conectividad Wi-Fi 802.11b/g/n y Bluetooth 4.2 BLE integradas, lo que permite una fácil conexión a redes inalámbricas y dispositivos compatibles[9].
- Puertos y conectores: Cuenta con puertos USB 2.0 (cuatro puertos), un conector Ethernet, un conector HDMI, un conector de audio de 3.5 mm, un conector para la cámara Raspberry Pi (CSI), un conector para la pantalla táctil Raspberry Pi (DSI), y un conector para la tarjeta microSD[8].
- GPIO: Tiene un encabezado GPIO de 40 pines que permite la conexión de una variedad de periféricos y componentes electrónicos adicionales, lo que lo hace ideal para proyectos de hardware y prototipado.
- Compatibilidad: Es compatible con una amplia gama de sistemas operativos, incluyendo Raspbian (basado en Debian), Ubuntu, Windows 10 IoT Core, y otros sistemas operativos Linux[9].



Figura 6: Raspberry utilizada en el proyecto.

2.6. InfluxDB.

InfluxDB es una base de datos de series temporales diseñada especialmente para almacenar y consultar datos que varían con el tiempo, como métricas, eventos y datos de sensores[10].

2.6.1. Características.

Algunas características clave de InfluxDB son:

- Base de datos de series temporales: Está optimizada para almacenar y consultar datos que varían con el tiempo, como métricas de sistemas, datos de sensores y registros de eventos[10].
- Alta disponibilidad y escalabilidad: Ofrece capacidades de alta disponibilidad y escalabilidad horizontal para manejar grandes volúmenes de datos de series temporales[10].
- Lenguaje de consulta InfluxQL: Proporciona un lenguaje de consulta potente y expresivo, llamado InfluxQL, que facilita la extracción y manipulación de datos de series temporales[10].
- Integración con otros sistemas: Ofrece integraciones con una variedad de herramientas y sistemas, incluyendo frameworks de monitorización, plataformas IoT y servicios en la nube[10].

2.6.2. Diferencias entre InfluxDB v1 y InfluxDB v2.

Dentro de las principales diferencias entre las dos versiones que ofrece InfluxDB podemos encontrar los siguientes puntos.

Modelo de datos y almacenamiento:

- **InfluxDB v1:** Utiliza un modelo de datos basado en series temporales con conceptos como series, puntos de datos y retención de políticas. Almacena datos en un formato basado en columnas llamado TSM (Time-Structured Merge Tree).
- **InfluxDB v2:** Introduce el concepto de "buckets" (cubetas) para organizar los datos, junto con "measurements" (mediciones) y "fields" (campos). Utiliza un sistema de almacenamiento basado en el motor de almacenamiento nativo de BoltDB.

API y lenguaje de consulta:

- **InfluxDB v1:** Tiene una API HTTP RESTful y utiliza un lenguaje de consulta llamado InfluxQL, que es específico de InfluxDB y optimizado para consultas de series temporales.
- **InfluxDB v2:** Continúa usando una API HTTP RESTful pero también ofrece una API gRPC. Introduce Flux, un nuevo lenguaje de consulta más poderoso y flexible que permite realizar operaciones más avanzadas y complejas en los datos.

Gestión de usuarios y autenticación:

- **InfluxDB v1:** Ofrece autenticación basada en usuarios y contraseñas, así como en roles.
- **InfluxDB v2:** Mejora la gestión de usuarios y la seguridad con la introducción de tokens y organizaciones. Además, proporciona una interfaz de usuario más completa para administrar usuarios, permisos y recursos.

Capacidades de escalabilidad y clúster:

- **InfluxDB v1:** Ofrece capacidades de escalabilidad horizontal limitadas a través de la replicación y la agrupación de servidores.
- **InfluxDB v2:** Mejora la escalabilidad con el soporte nativo para clústeres y una arquitectura más flexible y robusta para el escalado horizontal y vertical.

Interfaz de usuario y experiencia de usuario:

- **InfluxDB v1:** Tiene una interfaz de usuario básica a través de la interfaz web de Chronograf o a través de herramientas de terceros.
- **InfluxDB v2:** Introduce una interfaz de usuario más moderna y completa, llamada InfluxDB UI, que proporciona una experiencia de usuario mejorada para la administración, visualización y consulta de datos.

2.7. Grafana.

Grafana es una plataforma de visualización y análisis que permite crear paneles interactivos y gráficos a partir de datos almacenados en diversas fuentes, incluyendo InfluxDB[11].

2.7.1. Características.

- Visualización de datos flexible: Permite crear paneles de control interactivos y altamente personalizables para visualizar datos de diversas fuentes, incluyendo bases de datos de series temporales como InfluxDB[11].
- Amplia variedad de visualizaciones: Ofrece una amplia gama de opciones de visualización, incluyendo gráficos de líneas, barras, tortas, mapas, termómetros, entre otros, para representar datos de manera efectiva.
- Alertas y notificaciones: Permite configurar alertas y notificaciones basadas en umbrales y condiciones predefinidas para recibir avisos sobre eventos importantes.
- Exploración y análisis de datos: Facilita la exploración y el análisis de datos mediante funciones de zoom, filtro y agregación, así como la capacidad de comparar diferentes series de datos.
- Extensibilidad y comunidad activa: Es altamente extensible y cuenta con una comunidad activa que desarrolla complementos y paneles adicionales, así como una biblioteca de paneles predefinidos para diferentes casos de uso[11].

2.8. Herramientas y tecnologías utilizadas.

A continuación se muestran las herramientas y tecnologías utilizadas a lo largo del proyecto.

2.8.1. Lenguaje C.

El lenguaje C es un lenguaje de programación de propósito general que se utiliza ampliamente en el desarrollo de sistemas operativos, aplicaciones de bajo nivel y software de sistemas embebidos. Es un lenguaje de programación estructurado y de nivel medio, que combina la potencia y la flexibilidad del lenguaje ensamblador con la estructura y la portabilidad del lenguaje de alto nivel.

El lenguaje C es conocido por su eficiencia, velocidad y capacidad para acceder directamente a las características del hardware, lo que lo hace especialmente adecuado para el desarrollo de software de sistemas embebidos y de bajo nivel.

2.8.2. Raspberry Imager.

Raspberry Pi Imager 7 es una herramienta oficial proporcionada por la Fundación Raspberry Pi para facilitar la instalación de sistemas operativos en tarjetas SD o microSD para su uso en placas Raspberry Pi.

Esta herramienta permite a los usuarios descargar, seleccionar y grabar imágenes de sistemas operativos compatibles directamente en una tarjeta de memoria, simplificando así el proceso de preparación de una tarjeta para su uso con una Raspberry Pi. Raspberry Pi Imager es una aplicación multiplataforma que está disponible para Windows, macOS y Linux, lo que la hace accesible para una amplia variedad de usuarios.



Figura 7: Raspberry imager.

2.8.3. Extencion SSH.

SSH (Secure Shell) es un protocolo de red que proporciona a los usuarios una forma segura de acceder y administrar dispositivos remotos a través de una conexión cifrada. La extensión SSH se refiere al uso y la configuración de SSH en sistemas informáticos para permitir el acceso remoto seguro a través de la línea de comandos o la transferencia de archivos. SSH se utiliza comúnmente para administrar servidores y dispositivos de red de forma remota, así como para transferir archivos de manera segura entre sistemas.

La extensión SSH puede incluir la configuración de servidores SSH, la generación y gestión de claves SSH, así como el uso de clientes SSH para conectarse a sistemas remotos de manera segura.

2.8.4. Arduino IDE.

Arduino es una placa de desarrollo basada en un microcontrolador Atmel. Es importante precisar que los microcontroladores son circuitos integrados en los que es posible grabar instrucciones, las que se deben escribir con un lenguaje de programación y utilizando un entorno de desarrollo compatible.

El entorno de desarrollo integrado o IDE de Arduino es una aplicación multiplataforma que puedes utilizar para escribir y cargar programas en placas Arduino y también en aquellas que sean compatibles. Pero no solo eso, ya que gracias a núcleos generados por terceros, también se puede utilizar para cargar programas en placas de desarrollo de otros proveedores [12].

Características principales.

Arduino IDE se distribuye de forma gratuita, por lo que solo necesitas acceder al sitio web oficial de la aplicación para descargarla con una licencia libre, de modo que es posible acceder al código fuente del IDE y construir e instalar desde él o realizar las modificaciones que consideres necesarias. Para la mayoría de los usuarios, bastará con descargar el instalador adecuado para el sistema operativo y proceder con la instalación.

Otra de las ventajas de este IDE es que se trata de una aplicación multiplataforma, que puede ser instalada y utilizada en diferentes sistemas operativos, por ejemplo, Microsoft Windows, GNU/Linux o macOS. Para obtener la instalación adecuada, basta con visitar el sitio web oficial y elegir la opción que necesites.

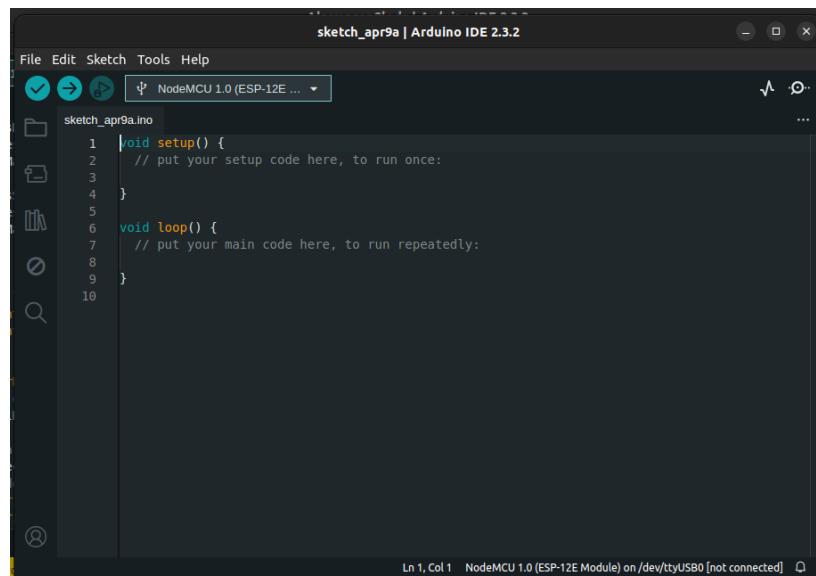


Figura 8: Interfaz de Arduino IDE

2.8.5. InfluxDbClient.

La librería InfluxDbClient permite la interacción con una base de datos InfluxDB. InfluxDB es una base de datos de series temporales diseñada para manejar grandes volúmenes de datos de series temporales de forma eficiente. La librería InfluxDbClient facilita el envío de consultas a la base de datos InfluxDB y la recepción de resultados de estas consultas. Esto es útil para aplicaciones que necesitan almacenar y analizar datos de series temporales, como datos de sensores, telemetría, monitoreo de infraestructura, etc.

3. Trabajo relacionados.

En la presente sección, se encuentran aquellos trabajos que tienen cierta semejanza con el proyecto desarrollado durante la estadía. Entre los trabajos relacionados encontramos varias manera en que los predecesores monitorean sus redes mediante el uso de plataformas como ntopng, ntop, sedia, entre otros.

3.1. Implementación de un servidor como gestión y monitoreo de servicios para la red de datos en la UGEL Huamanga, 2018

El trabajo realizado por el ingeniero Bach. Omar Jesús Fernández Huaytalla [13]. consiste en la implementación de un sistema de monitoreo de dispositivos y servicios mediante el protocolo SNMP, utilizando tecnologías de virtualización, el Sistema Operativo CentOS 7 y Nethserver 7, junto con herramientas de código abierto como Ntopng, Suricata y Zabbix. El propósito de este sistema es monitorear el consumo de ancho de banda, así como analizar el tráfico en busca de ataques y anomalías, y asegurar el correcto funcionamiento del hardware y software utilizados en su investigación.

La problemática que el ingeniero Jesús Fernández encontró dentro de su institución, la UGEL Huamanga, fue la saturación de la red de trabajo debido a la expansión de la universidad y al aumento de dispositivos que intentan conectarse a la red actualmente, además de la falta de un servidor que regule el acceso a los sitios web permitidos. Entre las tecnologías utilizadas por el ingeniero se encuentran los sistemas operativos (SO) de Microsoft Windows 7, 8 y 10, además del SO Linux CentOS 7.

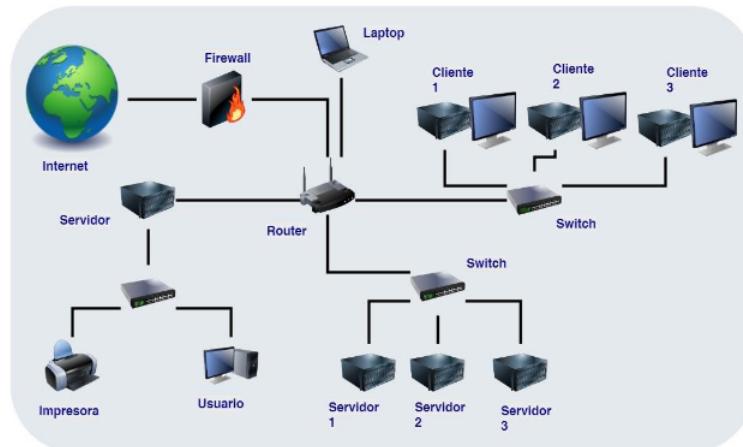


Figura 9: Arquitectura de la sistema de red local.

Dentro de los servicios de tratamiento de datos, el ingeniero se enfoca principalmente en sistemas desarrollados para Linux, como Nethserver, Ntopng, Suricata, Zabbix y VMware Workstation Player.

Entre los resultados obtenidos, se destaca el análisis del consumo de ancho de banda utilizado

por IPS o hosts individuales, por puertos, y la identificación de los protocolos de red más utilizados en tiempo real respecto al tráfico de la red. Se logró la detección y bloqueo de anomalías e infiltrados en la red; el sistema realizó la tarea de bloquear, alertar y guardar los eventos en un registro detallado referente al host por fecha y la descripción de las categorías uniformes. Asimismo, se registró el estado actual, como la capacidad, el rendimiento y la disponibilidad de los servicios, servidores y hardware.

En caso de problemas de conectividad, la plataforma envió notificaciones mediante el correo electrónico. Además, se implementó una estación de gestión gráfica (NMS) vía web en esta breve imagen se ilustra como funciona los NMS [10](#), que permitió mejorar el concepto del uso de las herramientas de código abierto proporcionando varias opciones para visualizar los datos recopilados, desde listas de problemas hasta gráficos simples.

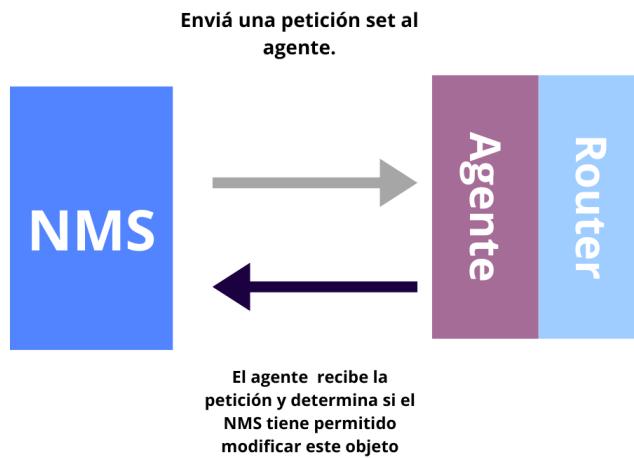


Figura 10: Peticiones NMS.

3.2. Monitoreo de datos mediante un administrador de flujo de datos, 2013

El proyecto, a cargo de los ingenieros Felipe Andrés Luco Pacheco [14] y Brian Valenzuela Ramos, tiene como objetivo principal la implementación de un sistema de consulta sobre un flujo de datos. Para ello, se realiza un análisis exhaustivo de los fundamentos del Sistema de Gestión de Flujo de Datos (DSMS por sus siglas en inglés) y de los principales DSMS dentro del flujo a consultar como se aprecia en la figura 11 consultado en [15]. Se busca estudiar la infraestructura de software necesaria para la implementación del sistema de consulta de flujo de datos, así como refutar los fundamentos teóricos de los DSMS, abarcando sus operadores, precondicionamientos de consultas y lenguajes de consulta.

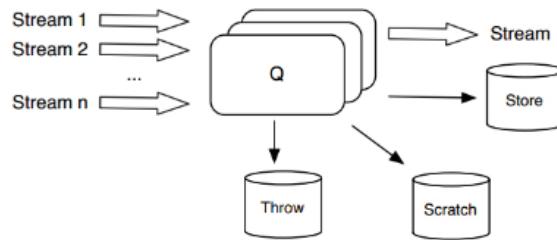


Figura 11: Arquitectura típica de un DSMS

Es importante destacar que Esper es un procesador de flujos de eventos y un motor de correlación de eventos dirigidos a las arquitecturas de eventos en tiempo real (EDA). Se trata de un kernel ligero escrito en Java, totalmente integrable con cualquier proceso que utilice este lenguaje de programación.

Asimismo, se resaltan las principales diferencias entre los sistemas de Gestión de Bases de Datos (DBMS) como los mostrados en la figura 13 de la misma fuente [15] y los DSMS, como el lenguaje de consulta CQL, que incluye IStream, DStream y RStream, y los operadores necesarios para estas consultas.

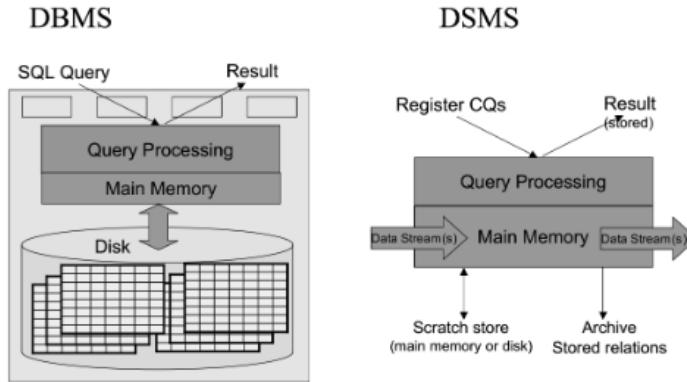


Figura 2: Diferencias entre la Arquitectura de un DBMS y DSMS.

Figura 12: Diferecias entra los DSMS y DBMS.

El trabajo proporciona un análisis detallado de la arquitectura y el funcionamiento de los sistemas administradores de flujos de datos, subrayando la importancia de los motores de consultas continuas en la toma de decisiones empresariales. Se enfatiza la capacidad de captura de eventos y procesamiento de datos en tiempo real de los DSMS, como Esper, así como su papel esencial en la recopilación, análisis y presentación de información para el monitoreo de datos. Se discute cómo los DSMS abordan el desafío de procesar flujos de datos continuos mediante el uso de ventanas deslizantes. Se destaca el cambio de paradigma que representan las consultas continuas en flujos de datos en tiempo real, en contraste con los enfoques tradicionales de trabajo con datos estáticos y finitos. Además, se menciona la recolección de información sobre distintos sistemas administradores de flujos de datos y lenguajes de consultas para poder implementar sistemas de consultas sobre flujos de información según los requisitos del proyecto.

3.3. Inteligencia artificial para el control de tráfico en redes de datos : Una Revisión,2021

El estudio desarrollando por los ingenieros D. A. León, J. G. Martínez, I. A. Ardila y D. J. Mosquera[16] se enfocó en mejorar las técnicas de control de tráfico de redes de datos mediante el uso de inteligencia artificial, específicamente algoritmos de aprendizaje automático y aprendizaje profundo. Se llevó a cabo un exhaustivo proceso de revisión bibliográfica sobre los aportes de la inteligencia artificial en el control de tráfico de redes, analizando estudios previos y tendencias actuales en el campo.

Durante el proceso, se emplearon algoritmos de aprendizaje automático y aprendizaje profundo, así como técnicas de representación de imágenes para la detección de malware y clasificación de tráfico. Se destacó el uso de redes neuronales convolucionales para el aprendizaje por representación y clasificación del tráfico de malware.

Los resultados obtenidos mostraron mejoras significativas en la eficiencia en la ejecución de algoritmos, la precisión de los resultados y la reducción de las tasas de error en el control de tráfico de redes. Se logró una mayor capacidad de detección de intrusiones y una mejor clasificación del tráfico, lo que contribuyó a una gestión más eficiente de las redes.

Dentro de las conclusiones del proyecto, se destaca que la inteligencia artificial, especialmente los algoritmos de aprendizaje automático y aprendizaje a profundidad, han sido fundamentales para mejorar las técnicas de control de tráfico de redes dentro de la visión de este artículo. Esto ha permitido a los sistemas de gestión de redes trabajar de manera más autónoma, proactiva y eficiente, mejorando la calidad del servicio ofrecido a los usuarios.

A través de este artículo se vislumbra un futuro prometedor en la aplicación de la inteligencia artificial en el ámbito del control de tráfico de redes. La evolución continua de los algoritmos de aprendizaje automático y aprendizaje profundo promete seguir impulsando mejoras significativas en la gestión de redes, lo que podría traducirse en una mayor resistencia a las amenazas cibernéticas y una adaptación más ágil a los cambios en el entorno de las redes. Además, la integración de técnicas de representación de imágenes para la detección de malware sugiere un enfoque multifacético y adaptable que podría ser clave en la prevención y mitigación de futuros ataques. En última instancia, este estudio resalta el papel crucial de la investigación interdisciplinaria y la innovación tecnológica en la mejora continua de la infraestructura de comunicaciones, sentando las bases para un futuro más seguro y eficiente en la gestión de redes de datos.

3.4. Analysis of centralized computer security systems through the alienVault ossim tool, 2022

Este artículo de investigación llevado a cabo por los ingenieros Ferruzola Gómez, Enrique Colón, Bermeo Almeida, Oscar Xavier, Arévalo Gamboa y Lissett Margarita[17] se centró en analizar los sistemas centralizados de seguridad informática mediante la herramienta AlienVault OSSIM. Para llevar a cabo este análisis, realizaron pruebas en distintos escenarios de la red con el objetivo de identificar la falta de procesos de monitorización en las empresas.

En cuanto a las tecnologías y herramientas utilizadas, integraron AlienVault OSSIM con diversas herramientas de seguridad de código abierto, como Snort, Ntop, OpenVAS, Arpwatch, OSSEC, Osiris, Nagios, OCS, Kismet, entre otras. Estas herramientas permitieron realizar un análisis exhaustivo de la seguridad de la red y la información de los sistemas centralizados emitida por la red.

Los resultados obtenidos mostraron que las empresas analizadas estaban expuestas a problemas y amenazas debido a la falta de procesos de monitorización de seguridad en la red. Por lo tanto, concluyeron que la implementación de una herramienta de gestión y monitoreo de seguridad informática, como AlienVault OSSIM, es crucial para detectar anomalías, solucionar problemas y optimizar la seguridad de la red.

En resumen, el estudio resalta la importancia de contar con procesos de monitorización de seguridad en las empresas y la utilidad de herramientas como AlienVault OSSIM para mejorar la gestión y detección de amenazas en la red.

A partir de los hallazgos de este estudio, se abre la puerta a reflexiones profundas sobre la necesidad imperante de fortalecer las estrategias de seguridad informática en las empresas. La creciente complejidad de las amenazas ciberneticas y la sofisticación de los ataques exigen una vigilancia constante y proactiva de los sistemas de información. La integración de herramientas de código abierto, como las mencionadas, proporciona una base sólida para la detección temprana de posibles brechas de seguridad y la respuesta eficaz ante incidentes. Sin embargo, es fundamental destacar que la implementación de estas herramientas no debe considerarse como una solución definitiva, sino como un primer paso hacia una cultura organizacional que valore y priorice la ciberseguridad en todas sus facetas.

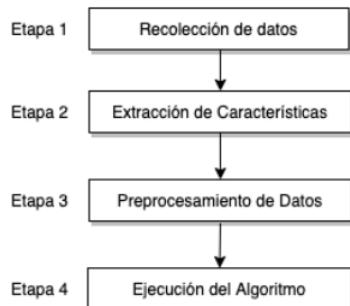


Figura 13: Etapas del proceso de clasificación haciendo uso de técnicas de Machine Learning

3.5. Evaluación de herramientas TIC para gestionar el monitoreo y análisis de la red de datos del Recinto de Golfito de la Universidad de Costa Rica, 2020

El proyecto a cargo del Ingeniero Alan Martín Corrales Rodríguez[18] es una investigación para obtener el título de maestría profesional en Tecnologías de la Información y Comunicación en la Gestión Organizacional. Según el ingeniero, el tráfico de red de datos del Recinto Golfito ha experimentado un aumento debido al crecimiento de la población estudiantil y docente, lo que compromete el ancho de banda con contenido no prioritario para las labores de enseñanza y administrativas. Por lo tanto, es crucial realizar un análisis inteligente de los datos, siguiendo las recomendaciones de Tenelanda y Vallejo (2012), para extraer información útil y compleja que contribuya a mejorar el fenómeno.

En el informe se mencionan diversas técnicas y herramientas de minería de datos y visualización de información para llevar a cabo este análisis. La transformación de los datos en información clara es esencial para la administración de redes y los directivos, empleando las estadísticas en el proceso de exploración.

La Oficina de Informática del Recinto Golfito realizó una investigación con el objetivo de mejorar la gestión de la red de datos con ejemplos de las interfaz de cada una de ellas como se muestra en la imagen de su interfaz ntop [22] que es una versión antigua del ntopng que se utiliza actualmente y los servicios ofrecidos mediante el desarrollo e implementación de herramientas de monitoreo y análisis. Se utilizaron herramientas como Wireshark como se muestra en la imagen como muestra de la interfaz [15], Nmap, Ntopng y EtherApe para la captura de datos de la red local, así como Telegraf e InfluxDB para el monitoreo de equipos y Networkminer, Grafana y Rapidminer para el análisis e interpretación de datos. Se detallaron sus ventajas, desventajas, comandos más utilizados y funcionalidades.

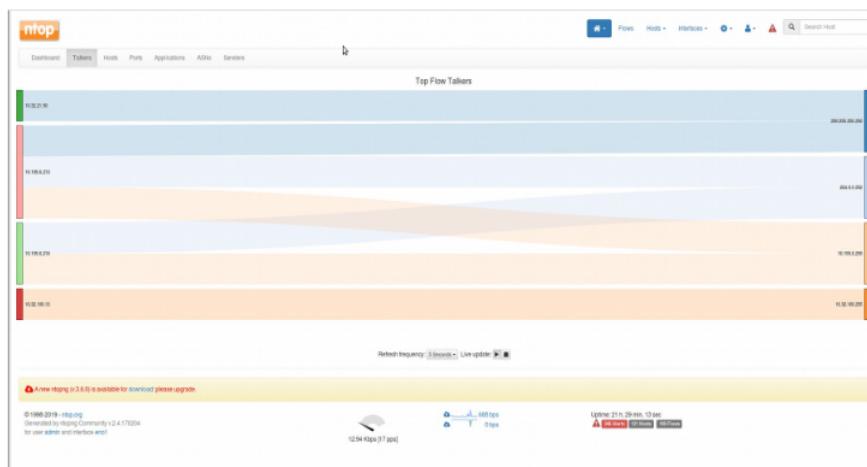


Figura 14: Interfaz de Ntop de la Evaluacion de herramientas TIC para gestionar el monitoreo y analisis de la red de datos

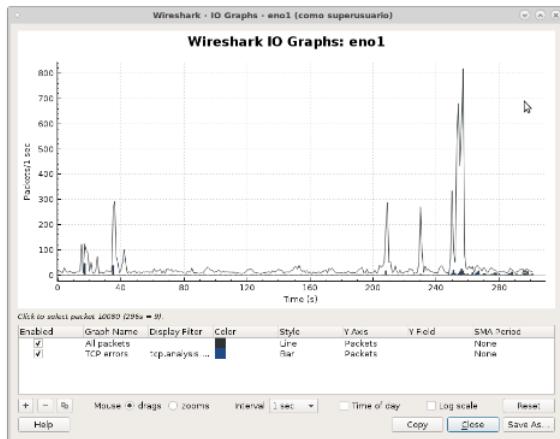


Figura 15: Interfaz de Wireshark de la Evaluacion de herramientas TIC para gestionar el monitoreo y analisis de la red de datos

Las recomendaciones resultantes de la evaluación de estas herramientas se centran en el uso de alternativas libres. Las opciones de software libre han mejorado significativamente en términos de instalación, soporte y desarrollo, haciéndolas más adecuadas para entornos organizacionales. Se enfatiza la importancia de distinguir entre herramientas completamente libres y aquellas con ciertas limitaciones en sus versiones gratuitas. Se recomienda el uso de herramientas libres o con versiones gratuitas, ya que satisfacen los requisitos de la oficina y proporcionan resultados efectivos para una mejor interpretación de la información.

3.6. Nodo de gestión y monitoreo de calidad de servicio de la empresa Ajnet en el cantón Latacunga,2023

El proyecto de la ingeniera Nancy Estefanía Zapata Tapia [19] aborda la implementación de un sistema de Gestión y Monitoreo de Calidad de Servicios (QoS) en AJnet, una empresa proveedora de servicios de internet ubicada en el cantón Latacunga. Este sistema se desarrolló con el objetivo de controlar el tráfico de internet y garantizar el rendimiento de las aplicaciones utilizadas por los usuarios.

Se analizó la situación actual de la empresa y se consideraron los equipos disponibles, implementando un servidor y realizando enrutamiento con routers Mikrotik. Además, se recopiló información sobre la calidad de servicio a través de encuestas a los clientes para identificar los parámetros que requieren mejoras y que sean medibles en el desarrollo de la aplicación.

La aplicación desarrollada es un sistema web alojado en un servidor Ubuntu Server que integra herramientas de monitoreo de redes como Zabbix, Ntop y Graphing de MikroTik. Cuenta con un sistema de alertas que envía mensajes a través de Telegram tanto al detectar un problema como al resolverlo, y dispone de una base de datos para el registro de la información y la comunicación con la plataforma principal, así como para la realización de pruebas de funcionamiento y corrección de errores en el sistema. Además, se proporciona un manual de usuario para facilitar su uso.

Este sistema desarrollado permite enviar alertas al personal de soporte técnico a través de mensajes por Telegram, mejorando así la calidad de los servicios y cumpliendo con los parámetros establecidos por la normativa de ARCOTEL para la provisión de acceso a internet, evitando problemas como la falta de servicio prolongada, intermitencia, lentitud y fallas generales de la red.

Se hacen algunas recomendaciones, entre ellas verificar las características de los equipos actuales de la empresa y su compatibilidad con las versiones de software de monitoreo de redes, implementar un servidor externo para evitar conflictos en la red, configurar el servidor en una red privada con acceso limitado, considerar análisis repetitivos por cliente para identificar correctamente las causas de las alertas generadas, y tener en cuenta los modelos de routers instalados en los clientes.

4. Sistema Propuesto.

En la sección de sistema propuesto, se abordan las tareas realizadas durante el periodo de estadía, así como también se explican los pasos, estructuras, funcionalidad y módulos desarrollados del proyecto.

En la siguiente figura 16 se muestra una gráfica de pastel con el porcentaje del tiempo utilizado para cada una de las actividades establecidas en el proyecto.

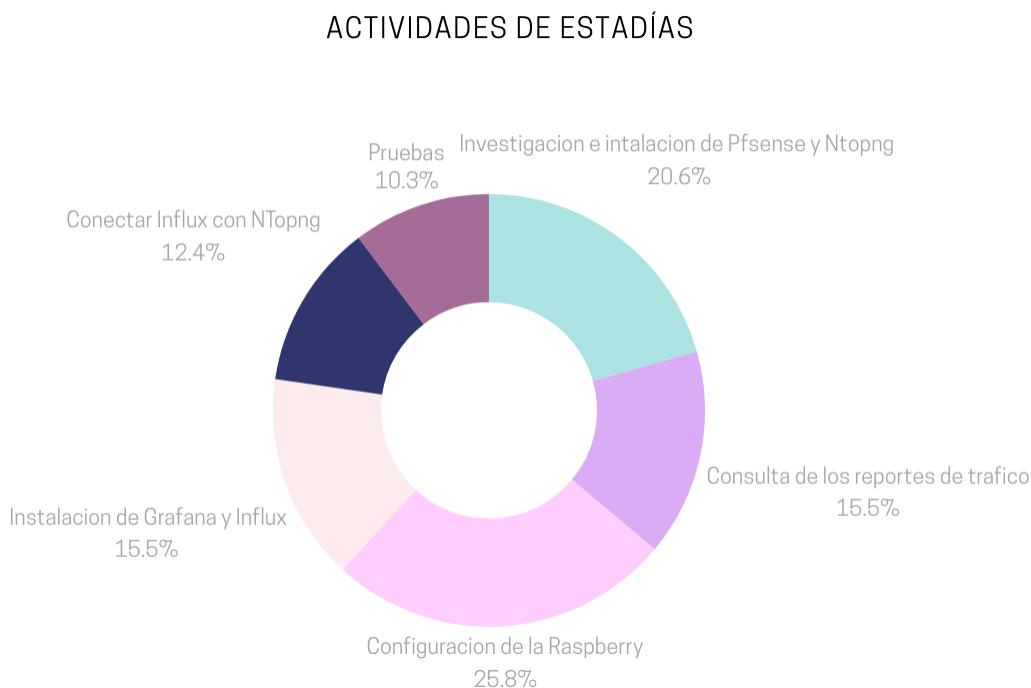
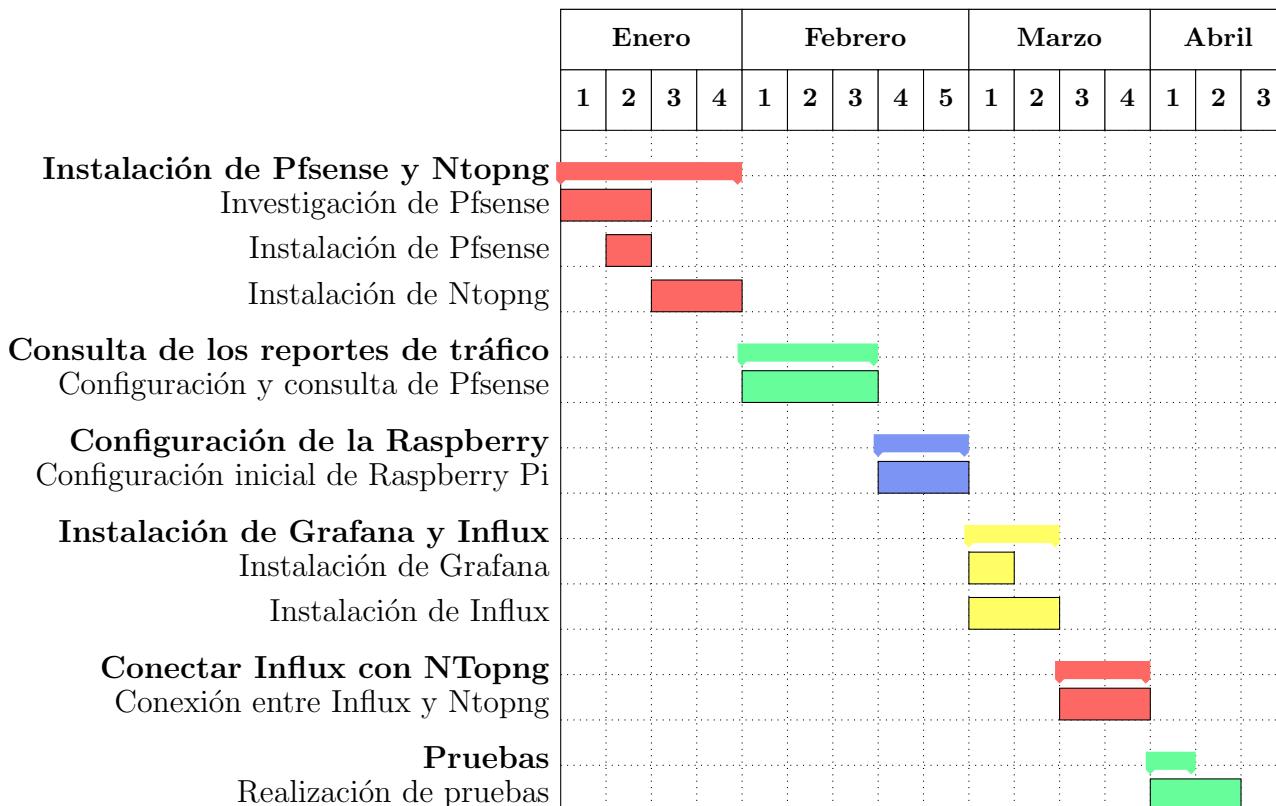


Figura 16: Gráfica de pastel con tiempos distribuidos.

Para cada una de las actividades marcadas en la gráfica de pastel (figura 16) se despliegan tareas específicas, estas son mostradas a continuación en el cuadro 1, siendo un diagrama de Gantt para ilustrar los tiempos invertidos en cada una, detallando también las actividades realizadas dentro del plazo de entrega del cuatrimestre.



Cuadro 1: Cronograma de actividades

4.1. Instalación y configuración de los software de monitoreo de tráfico.

En esta sección, se describe el proceso de instalación y configuración de los software de monitoreo de tráfico utilizados en el proyecto. Estos programas son fundamentales para comprender y gestionar eficientemente el flujo de datos en una red y como monitorear el tráfico de dicha red para el uso del dispositivo como se ve en la imagen 35.

4.1.1. Instalación y configuración de Pfsense.

En este apartado se describen las primeras tareas llevadas a cabo durante la estadía de proyecto. Durante el primer periodo de investigación del proyecto, se realizaron actividades en el departamento de sistemas. Se instaló el sistema Pfsense en el CPU asignado por el departamento, el cual contenía la versión 2.7.2 de Pfsense. Posteriormente, se procedió a instalar el equipo en el espacio designado por el departamento y a conectarlo a la red.

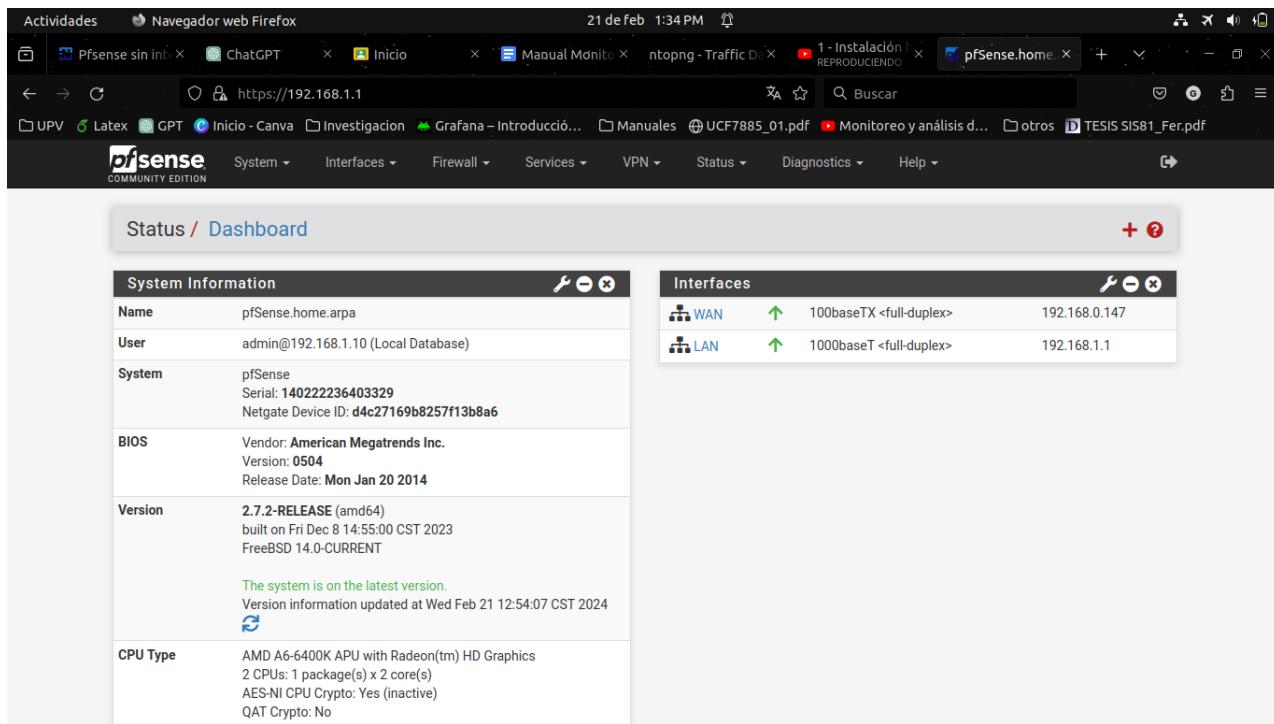


Figura 17: Pfsense: Inicio de Proyecto.

Se encontraron dificultades con la conectividad a la red desde el Pfsense, así como problemas para instalar los paquetes de Ntopng y Telegraf en el apartado de Gestor de Paquetes o Package Manager. Estos inconvenientes se resolvieron después de solucionar los problemas de conexión a la red del servidor Pfsense y conectar el equipo de trabajo externo mediante un cable de red a la LAN del servidor.

Para complementar este proceso inicial, se llevó a cabo un exhaustivo análisis de las configuraciones de red existentes y se identificaron posibles puntos de mejora en la infraestructura. Esto incluyó la revisión de la topología de red, la asignación de direcciones IP, así como la evaluación de posibles y vulnerabilidades en la seguridad.

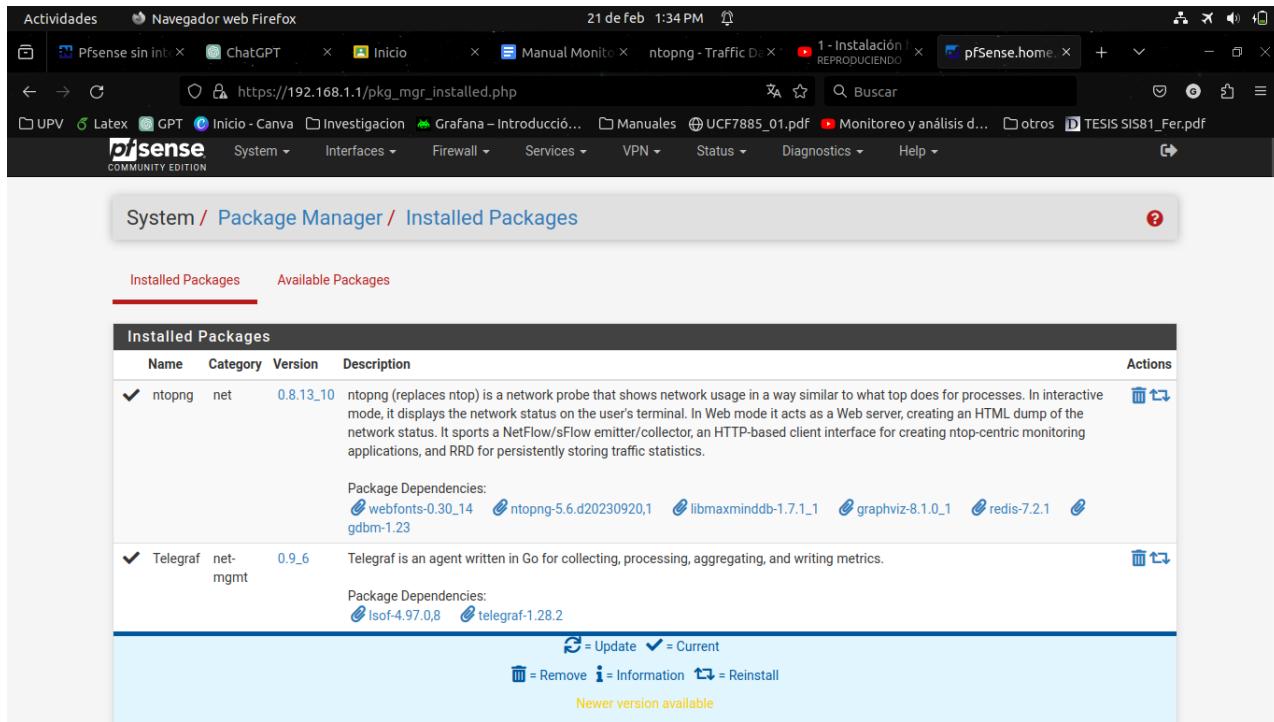


Figura 18: PfSense: Package Manager Error solucionado.

Una configuración general de PfSense implica definir la arquitectura de red, asignando interfaces como WAN, LAN y, si es necesario, OPT para segmentos adicionales. Luego, se establecen direcciones IP estáticas o se configura un servidor DHCP para asignar direcciones automáticamente. La configuración del firewall es esencial, definiendo reglas para controlar el tráfico entre las interfaces y hacia/desde Internet, junto con la configuración de NAT si se requiere. Se activan servicios de red como DNS, DHCP y NTP para el funcionamiento básico de la red, mientras que la configuración de VPN proporciona conectividad segura para dispositivos remotos.

The screenshot shows the pfSense web interface with the URL <https://192.168.1.1>. The main menu is visible at the top, and the 'Diagnostics' dropdown is open, showing various network monitoring tools like ARP Table, Authentication, and Ntopng. The 'Ntopng' option is selected. On the left, there's a 'System Information' panel with details about the pfSense system, including its name (pfSense.home.arp), user (admin@192.168.1.10), system serial (140222236403329), and BIOS information. The 'CPU Type' section indicates an AMD A6-6400K APU with Radeon(tm) HD Graphics. On the right, there's a 'Interfaces' panel showing two WAN ports (uplex) connected to 192.168.0.147 and 192.168.1.1 respectively. Below the interfaces is a 'Ntopng' section with options for settings, capture, and monitoring.

Figura 19: Pfsense: Configuración de la instalación de Ntopng en el sistema.

La supervisión de la actividad de red a través de registros y la programación de actualizaciones automáticas para mantener el sistema seguro y estable son pasos críticos en la configuración. Dependiendo de las necesidades específicas, se pueden implementar servicios adicionales y medidas de seguridad para fortalecer la red contra posibles amenazas estas configuraciones se hacen con el apoyo de el ingeniero Jaime del Departamento de sistemas y de la información sobre instalacion del sistema y configuracion[20].

This screenshot shows the same 'System Information' panel as Figure 19, providing detailed hardware and software specifications for the pfSense system. It includes the pfSense version (2.7.2-RELEASE), CPU type (AMD A6-6400K APU with Radeon(tm) HD Graphics), and memory (2 CPUs: 1 package(s) x 2 core(s)). The 'CPU Type' section also lists AES-NI and QAT support.

Figura 20: Pfsense: Especificaciones del Pfsense.

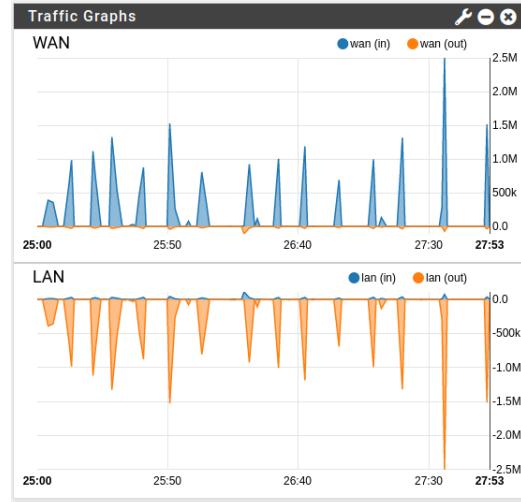


Figura 21: Pfsense: Gráfica de tráfico vista desde Pfsense.

4.1.2. Instalación y configuración de Ntopng.

Para configurar Ntopng en Pfsense, primero se instala el paquete desde el Gestor de Paquetes de Pfsense. Una vez instalado, se accede a la interfaz web de Ntopng para realizar la configuración inicial, que incluye la definición de las interfaces de red a monitorear y la configuración de las opciones de visualización y generación de informes. Es importante configurar correctamente las opciones de almacenamiento para evitar la saturación del disco duro. Además, se pueden configurar alertas para notificar sobre eventos importantes en la red. Después de la configuración inicial, se monitorea continuamente el tráfico de red utilizando Ntopng para obtener información detallada sobre el uso de la red, el tráfico de aplicaciones y las tendencias de uso. Ntopng por defecto está en el puerto 3000, lo cual se debe tomar en cuenta para las instalaciones y sistemas utilizados en el futuro.

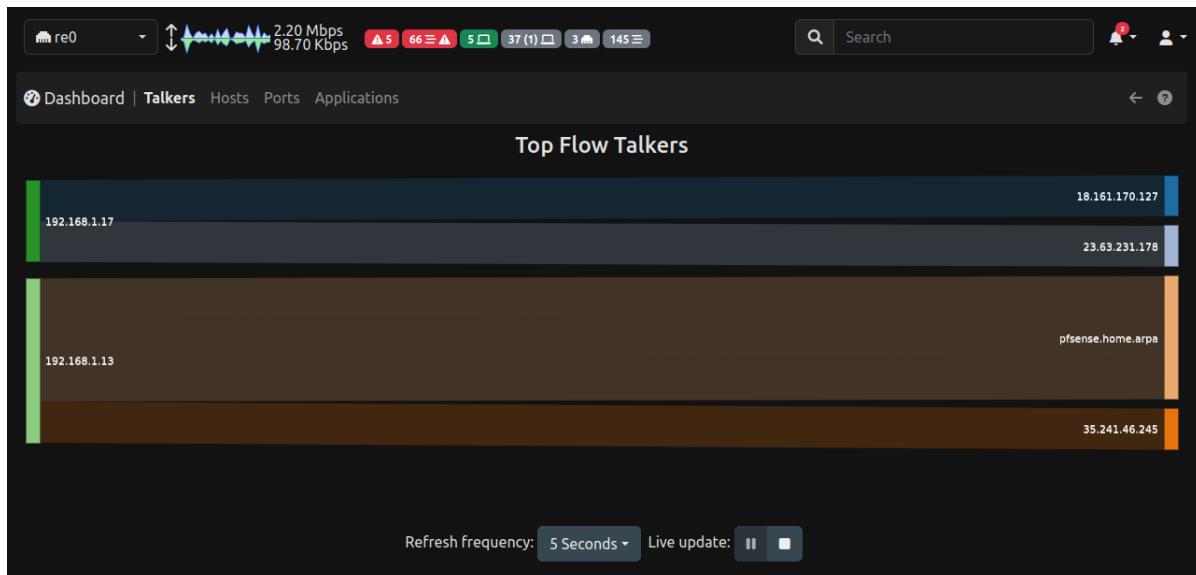


Figura 22: NTopng: Inicio de Proyecto.

Dentro de Ntopng, una herramienta avanzada de monitoreo de red, se pueden encontrar diversas funcionalidades y datos relacionados con el tráfico de red. Algunas de las características y datos que se pueden obtener incluyen:

- 1. Información de tráfico en tiempo real:** Ntopng proporciona una vista en tiempo real del tráfico de red, incluyendo el flujo de datos, protocolos utilizados, direcciones IP de origen y destino, puertos, y mucho más. Esto permite a los administradores de red tener una comprensión instantánea de cómo se está utilizando la red en ese momento.
- 2. Estadísticas de tráfico histórico:** Además del monitoreo en tiempo real, Ntopng también ofrece la capacidad de visualizar estadísticas históricas de tráfico. Esto incluye datos sobre el tráfico pasado, tendencias de uso de la red, patrones de tráfico, y análisis de la actividad de red a lo largo del tiempo.

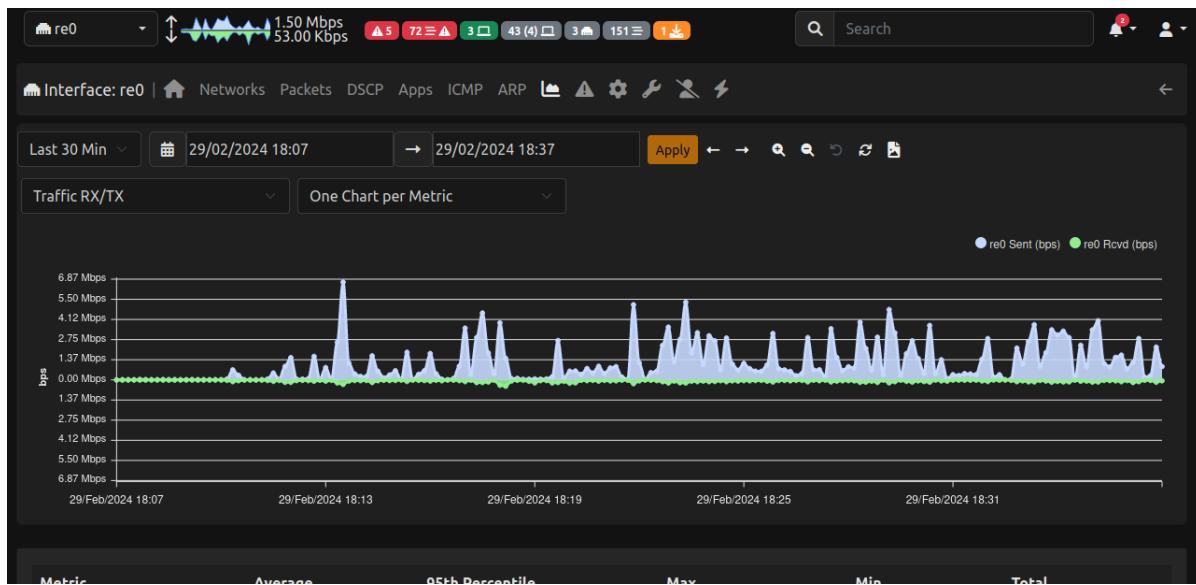


Figura 23: Ntopng: Vistazo al Networkchart dentro de Ntopng.

3. **Análisis detallado de protocolos:** Ntopng es capaz de analizar y categorizar el tráfico de red según los diferentes protocolos utilizados. Esto incluye protocolos como TCP, UDP, ICMP, así como aplicaciones específicas como HTTP, FTP, DNS, entre otros. Proporciona información detallada sobre la cantidad de tráfico generado por cada protocolo, las conexiones establecidas y otros detalles relevantes.
4. **Identificación de aplicaciones y servicios:** Ntopng puede identificar las aplicaciones y servicios específicos que generan tráfico en la red. Esto permite a los administradores detectar y supervisar el uso de aplicaciones específicas, identificar posibles problemas de rendimiento o seguridad, y tomar medidas apropiadas según sea necesario.

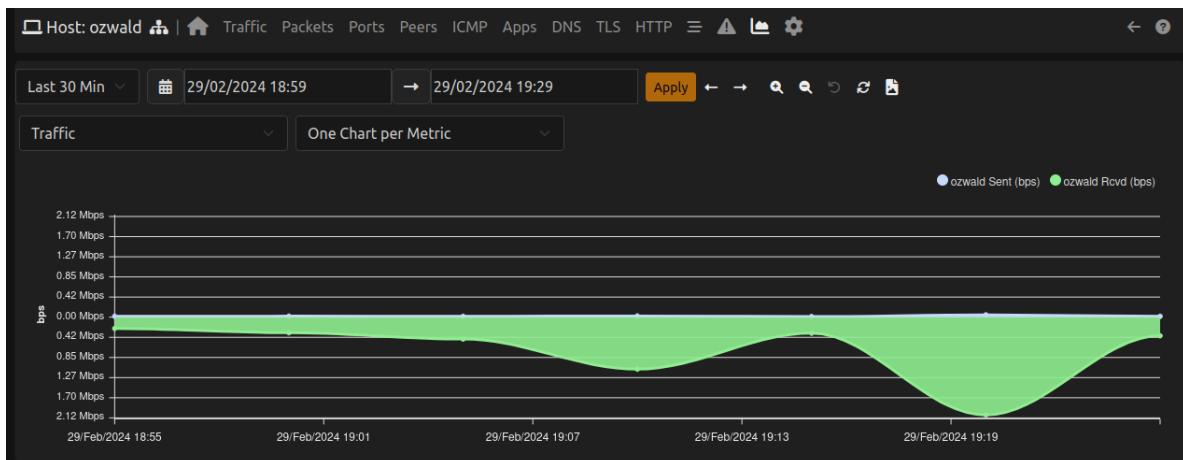


Figura 24: Ntopng: Vistazo a la infomacion de host chart dentro de Ntopng.

5. **Gestión de ancho de banda:** Ntopng ofrece herramientas para gestionar y controlar el ancho de banda disponible en la red. Permite establecer límites y prioridades para diferentes interfaces y aplicaciones, asegurando que se cumplan los objetivos de calidad de servicio (QoS).

lar el ancho de banda de la red. Esto incluye la capacidad de establecer políticas de QoS (Calidad de Servicio), priorizar ciertos tipos de tráfico sobre otros, limitar el ancho de banda para aplicaciones específicas, y realizar otras acciones para optimizar el rendimiento de la red.

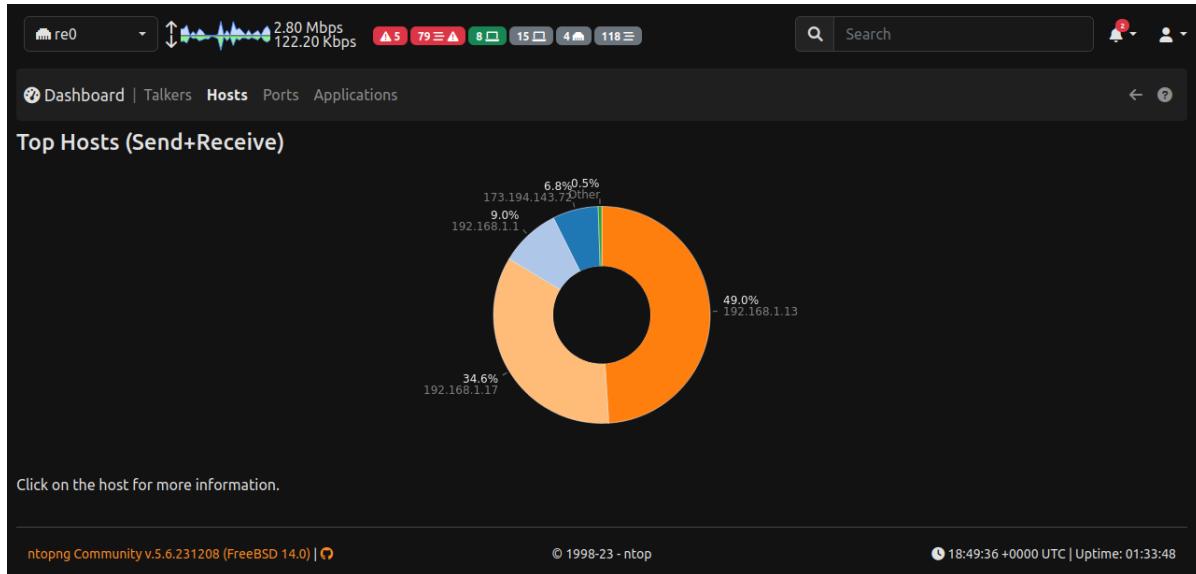


Figura 25: Ntopng: Vistazo de los Top Host dentro del Dashboard de Ntopng.

Para facilitar el acceso remoto al PfSense y Ntopng en el contexto del proyecto, se implementó una solución mediante la conexión de un punto de acceso (Access Point) a la LAN del servidor PfSense. Esta configuración permite a los usuarios acceder de forma remota al PfSense ingresando a través de la dirección IP 192.168.1.1. El punto de acceso está dedicado exclusivamente para este propósito, garantizando un acceso seguro y controlado al sistema desde ubicaciones externas a la red local.

La incorporación del punto de acceso proporciona una capa adicional de seguridad al permitir que el acceso remoto se realice de manera controlada y protegida. Los usuarios autorizados pueden conectarse de forma segura al PfSense y Ntopng desde cualquier ubicación externa, lo que mejora la flexibilidad y la accesibilidad del sistema sin comprometer la integridad de la red interna. Esta solución garantiza que el acceso remoto sea seguro y eficiente, cumpliendo con los requisitos del proyecto para habilitar la supervisión y gestión remota de la red a través de PfSense y Ntopng.

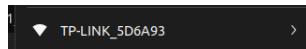


Figura 26: Red del pfSense dada por el access point conectado al pfSense.

4.2. Consulta de los reportes de tráfico.

El reporte de tráfico proporciona información detallada sobre los dispositivos conectados, incluyendo su dirección IP, dirección MAC, sistema operativo, ID de red local y tráfico de red, como UDP y TCP. Durante el proceso de implementación, se consideraron dos posibles soluciones: migrar el archivo .JSON de forma pública como se muestra en la figura 27, aunque no se contaba con información detallada sobre cómo realizar este proceso, o implementar una Raspberry Pi con Influx y Grafana. Después de probar ambas opciones, se decidió finalmente optar por la segunda alternativa. Para llevar a cabo esta decisión, se utilizó Raspberry Imager para instalar una versión específica de Linux Server (v23.10) en la Raspberry. Sin embargo, surgieron desafíos físicos, especialmente relacionados con el cargador de la Raspberry, que causaron interrupciones en dos ocasiones. Actualmente, se ha resuelto este problema mediante el uso de un cargador de 5V y 3.3A para garantizar un suministro de energía adecuado y evitar futuros problemas de configuración debidos a la falta de voltaje.

La versión empresarial de ntopng ofrece un reporte más exhaustivo; sin embargo, es posible que acceder a este informe desde la ESP32 requiera autorización de acceso al sistema sin mostrar directamente los resultados. Una alternativa que se planteó fue acceder a la información a través de Influx. Sin embargo, se identificó que ntopng no es compatible con la versión v2 de InfluxDb, por lo que se optó por utilizar una de las versiones v1, específicamente la 1.8.10, para garantizar la compatibilidad y el funcionamiento adecuado del sistema. Este enfoque permitirá una integración efectiva de los datos de tráfico en la plataforma de monitoreo, facilitando el análisis y la visualización de la información recopilada.

```

{
  "rc": 0,
  "rsp": {
    "udp.bytes.rcvd": 191293133,
    "pkts_ratio": -0.46737080812454,
    "udp.packets.rcvd": 159384,
    "bytes.sent.anomaly_index": 23,
    "bytes.rcvd.anomaly_index": 13,
    "tcp.packets.seq_problems": true,
    "other_ip.bytes.sent.anomaly_index": 25
  },
  "ndpi": {
    "Skype_Teams": {
      "packets.sent": 17,
      "breed": "Acceptable",
      "num.flows": 0,
      "bytes.sent": 2392,
      "bytes.rcvd": 17359,
      "packets.rcvd": 19,
      "duration": 5
    },
    "MDNS": {
      "packets.sent": 50,
      "breed": "Acceptable",
      "num.flows": 9,
      "bytes.sent": 6143,
      "bytes.rcvd": 704,
      "packets.rcvd": 8,
      "duration": 115
    }
  },
  "HTTP": {
    "packets.sent": 8,
    "breed": "Acceptable"
  }
}

```

Figura 27: NTopng: Informacion del reporte de flujo en el formato .json

Host: ozwald		Traffic	Packets	Ports	Peers	ICMP	Apps	DNS	TLS	HTTP	≡	⚠	⠇	⚙️
Router/AccessPoint MAC Address	Cisco-Li_C4:6D:9E													
Host MAC Address	0E:39:36:6C:70:F2													Computer
IP Address	192.168.1.17 [192.168.1.0/24]													Host Pool: Default
OS	Android [WAChat/1.2]													
Name	ozwald	🕒	⚙️	🌐	💻	🖨️	📅	🕒	⚙️	🌐	💻	🖨️	📅	
Active Monitoring	Add ICMP Monitor +													
Behavioural Counter Anomalies	4													
First / Last Seen	29/02/2024 11:54:33 [01:33:57 ago]													29/02/2024 13:28:10 [00:20 ago]

Figura 28: Ntopng: Informacion de un host designado dentro de la red.

Reset Host Stats	Reset Host Stats	Reset Blacklisted Hosts Stats
Additional Host Names	Source	Name
	DHCP	ozwald
	MDNS	android-5
Download	JSON	1 min Filter (BPF) pcap Download

Figura 29: Ntopng: Información de un host designado dentro de la red.

La implementación de soluciones de monitoreo de tráfico se ha enfrentado a diversos desafíos técnicos y físicos, pero se ha logrado avanzar hacia una configuración exitosa utilizando una Raspberry Pi 3 Model B con InfluxDb v1.8.10 [21] y Grafana. A pesar de las limitaciones de compatibilidad entre ntopng e InfluxDb, se ha encontrado una solución viable mediante la utilización de la versión 1.8.10 de InfluxDb, lo que asegura la integración efectiva de los datos de tráfico para su posterior análisis y visualización.

4.3. Configuración de la Raspberry.

Durante la instalación del sistema operativo en la Raspberry Pi mediante Raspberry Imager, se exploraron las opciones avanzadas de configuración. Inicialmente, se consideraron los sistemas CORE 22 y CORE 20, sin embargo, debido a dificultades de configuración, se optó por el sistema Linux Server 23.10, el cual ha sido utilizado desde entonces. A pesar de esta elección, surgieron varios problemas relacionados con la conectividad a la red y el acceso mediante SSH, que se atribuyeron a problemas de configuración de puertos y otros inconvenientes técnicos. Para la instalación del sistema operativo, se empleó una tarjeta Micro SD Adata v30 A2 [31](#).



Figura 30: Raspberry y Micro SD que se usaron en el proyecto.

Además de los problemas de conectividad mencionados, se identificó un inconveniente con el cargador utilizado inicialmente. Se empleaba un cargador de 5V y 2.5A, el cual no proporcionaba la potencia suficiente y ocasionaba que la Raspberry Pi se apagara de manera intermitente. Esta situación se resolvió posteriormente con la adquisición y uso de un cargador de mayor capacidad (5V y 3.3A), garantizando un suministro de energía adecuado y evitando futuras interrupciones en el funcionamiento del dispositivo.

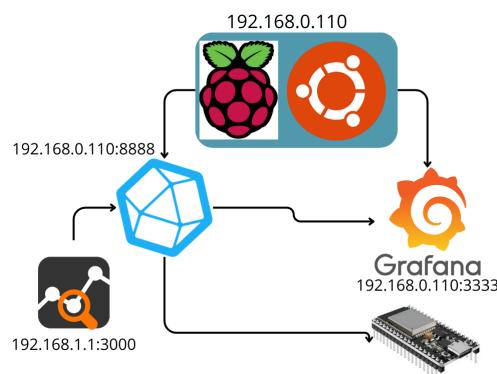


Figura 31: Configuración de la Raspberry Pi como sistema final.

4.4. Instalación de Grafana y Influx.

En el desarrollo de este proyecto, se utilizaron Grafana y InfluxDB en su versión v1 para facilitar la consulta externa de los datos recopilados por Ntopng a través de la Raspberry Pi. Con el fin de evitar conflictos de puertos, se modificaron las configuraciones predeterminadas: Grafana se configuró para operar en el puerto 3333, mientras que InfluxDB se configuró para funcionar en el puerto 8888. Esto se decidió porque Ntopng utiliza el puerto 3000 de forma predeterminada [22].

La conectividad entre InfluxDB y Grafana se estableció con éxito después de realizar ajustes en la configuración de los puertos y llevar a cabo una investigación exhaustiva sobre cómo realizar la conexión y vinculación entre ambos programas. Estos ajustes fueron fundamentales para garantizar una comunicación sin problemas entre Grafana, InfluxDB y Ntopng, permitiendo así el análisis y la visualización eficientes de los datos de tráfico de red recopilados por Ntopng a través de la Raspberry Pi.

4.5. Conectar Influx con NTopng.

La conexión entre InfluxDB y Ntopng en el contexto de este proyecto implica establecer una integración que permita almacenar los datos recopilados por Ntopng en la base de datos de series temporales proporcionada por InfluxDB. Para lograr esto, se siguen varios pasos:

1. **Configuración de InfluxDB:** En primer lugar, se configura InfluxDB para que esté listo para recibir y almacenar los datos de Ntopng. Esto incluye la instalación de InfluxDB en la raspberry, la configuración de la base de datos designada como ntop.

```

> SELECT * FROM "host:traffic" LIMIT 10
name: host:traffic
time          bytes_rcvd bytes_sent host      ifid
----          -----
1712001061000000000 1573360   25446072  192.168.0.110 0
1712001061000000000 180938069 19947051  192.168.1.13  0
1712001061000000000 3698589   14724634  192.168.1.1   0
1712001121000000000 182091913 20082665  192.168.1.13  0
1712001121000000000 3721657   14748343  192.168.1.1   0
1712001121000000000 1650884   26534621  192.168.0.110 0
1712001181000000000 3748829   14774822  192.168.1.1   0
1712001181000000000 1716033   27105030  192.168.0.110 0
1712001181000000000 183404192 20282277  192.168.1.13  0
1712001241000000000 1729480   27226756  192.168.0.110 0
> SELECT * FROM "host:total_flows"
name: host:total_flows
time          flows_as_client flows_as_server host      ifid
----          -----
1712001061000000000 0           117        192.168.0.110 0
1712001061000000000 5468       37         192.168.1.13  0
1712001061000000000 28         3490       192.168.1.1   0
1712001121000000000 5523       37         192.168.1.13  0
1712001121000000000 28         3526       192.168.1.1   0
1712001121000000000 0          127        192.168.0.110 0
1712001181000000000 28         3564       192.168.1.1   0
1712001181000000000 0          132        192.168.0.110 0
1712001181000000000 5591       37         192.168.1.13  0
1712001241000000000 0          132        192.168.0.110 0
> SELECT * FROM "host:tcp_packets" LIMIT 10
name: host:tcp_packets
time          bytes_rcvd bytes_sent host      ifid
----          -----

```

Figura 32: InfluxDb: Consulta de los datos almacenados dentro de la base de datos ntop vinculada a ntopng.

2. **Configuración de Ntopng:** Una vez configurado InfluxDB, se configura Ntopng pa-

ra que envíe los datos recopilados a InfluxDB. Esto generalmente implica configurar Ntopng para que utilice InfluxDB como un destino de almacenamiento para sus datos de monitoreo de red. Esto se realiza mediante la configuración de Ntopng para que envíe datos a través del protocolo de línea de comandos de InfluxDB o mediante la configuración de una interfaz específica en Ntopng para enviar datos a InfluxDB.

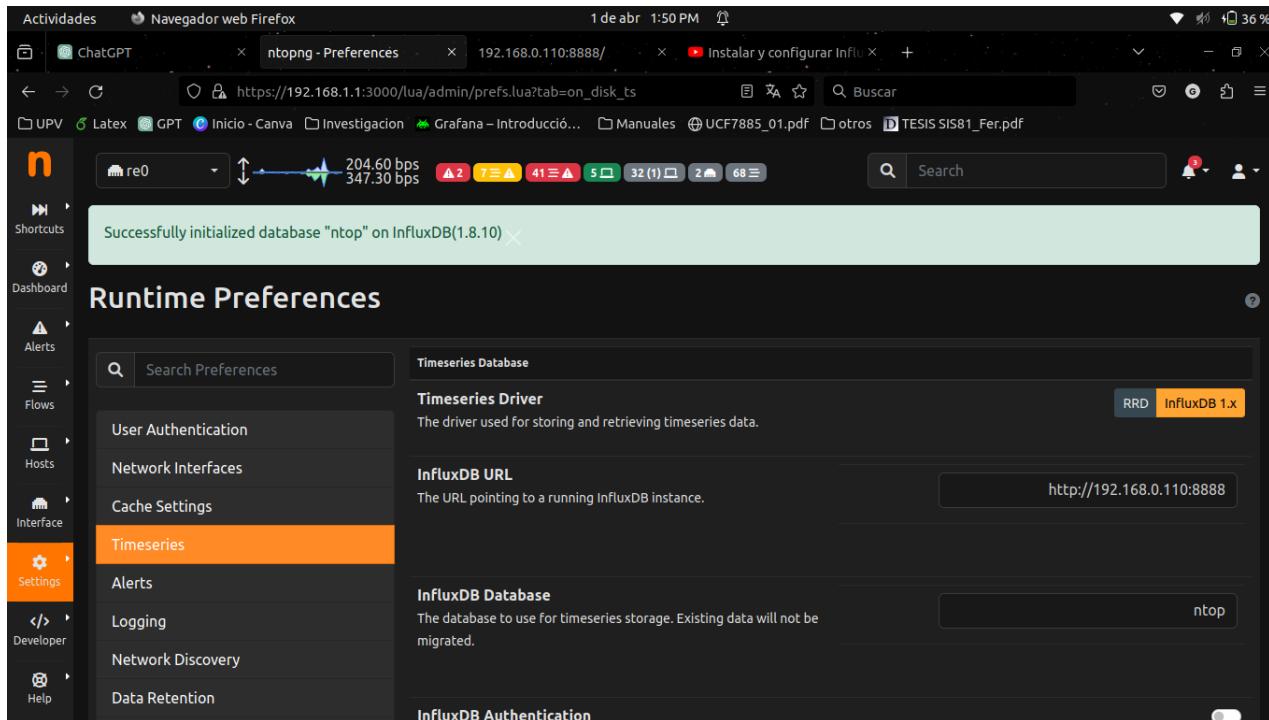


Figura 33: Ntopng: Configuración de InfluxDB dentro de Ntopng en el apartado de preferencias.

3. **Pruebas y ajustes:** Despues de configurar ambas aplicaciones, es importante realizar pruebas para asegurarse de que la conexión entre Ntopng e InfluxDB esté funcionando correctamente. Esto implica monitorear el flujo de datos desde Ntopng hasta InfluxDB y realizar ajustes según sea necesario para garantizar una conexión estable y confiable.

```

influxdb:exported_points
influxdb:exports
influxdb:rtt
influxdb:storage_size
mac:arp_rqst_sent_rcvd_rpls
mac:ndpi_categories
mac:traffic
redis:hits
redis:keys
redis:memory
> SELECT * FROM <nombre_de_la_serie_de_tiempo> LIMIT 10
ERR: error parsing query: found <, expected identifier at line 1, char 15
> SELECT * FROM host:traffic LIMIT 10
ERR: error parsing query: found ;, expected ; at line 1, char 19
> SELECT * FROM traffic LIMIT 10
> SELECT * FROM "host:traffic" LIMIT 10
name: host:traffic
time          bytes_rcvd bytes_sent host      ifid
----          -----       -----   ----
1712001061000000000 1573360    25446072 192.168.0.110 0
1712001061000000000 180938069  19947051 192.168.1.13 0
1712001061000000000 3698589   14724634 192.168.1.1 0
1712001121000000000 182091913  20082665 192.168.1.13 0
1712001121000000000 3721657   14748343 192.168.1.1 0
1712001121000000000 1650884   20534621 192.168.0.110 0
1712001181000000000 3748829   14774822 192.168.1.1 0
Data:1712001181000000000 1716033   27105030 192.168.0.110 0
1712001181000000000 183404192  20282277 192.168.1.13 0
1712001241000000000 1729480   27226756 192.168.0.110 0
>

```

Figura 34: InfluxDB: Consulta de los datos almacenados dentro de la base de datos ntop vinculada a ntopng.

4.5.1. Configuración Influx con Grafana.

La configuración de InfluxDB con Grafana implica integrar la base de datos de series temporales de InfluxDB [23] con la plataforma de visualización de datos de Grafana para poder analizar y visualizar los datos almacenados en InfluxDB.

- Configuración de Grafana:** Despues de configurar InfluxDB, se procede a configurar Grafana para que pueda conectarse a la base de datos de InfluxDB y visualizar los datos almacenados. Esto implica agregar un origen de datos en Grafana y especificar los detalles de conexión, como la dirección del servidor InfluxDB, el puerto y el nombre de la base de datos previamente hecha en influxDB[24].
- Creación de paneles y gráficos:** Una vez que Grafana está conectado a InfluxDB, se pueden crear paneles y gráficos personalizados para visualizar los datos de la serie temporal almacenados en InfluxDB. Grafana ofrece una amplia gama de opciones de visualización [25] y configuración que permiten crear paneles de control personalizados, como los que se muestran en la imagen 42.

En la imagen 35 observamos el caso de uso de un usuario en la red, y cómo la alarma funciona a través de consultas a InfluxDB, que a su vez recibe sus datos de ntopng y de pfSense.

Caso de Uso

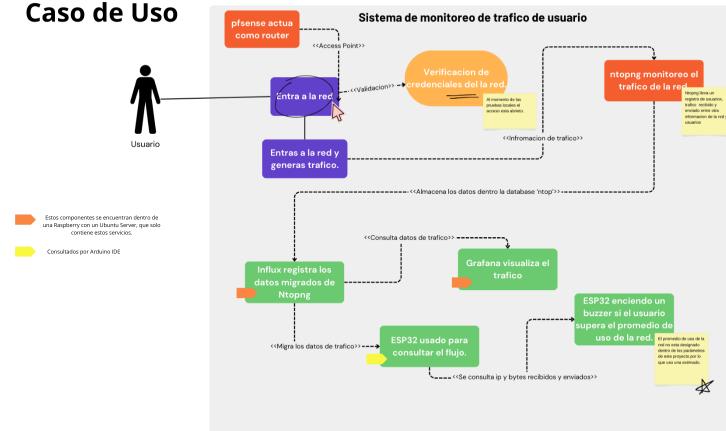


Figura 35: Caso de uso: Monitoreo de un usuario en la red.

Y de nuestro sistema, basándonos en sus componentes como se muestran en la imagen 36.

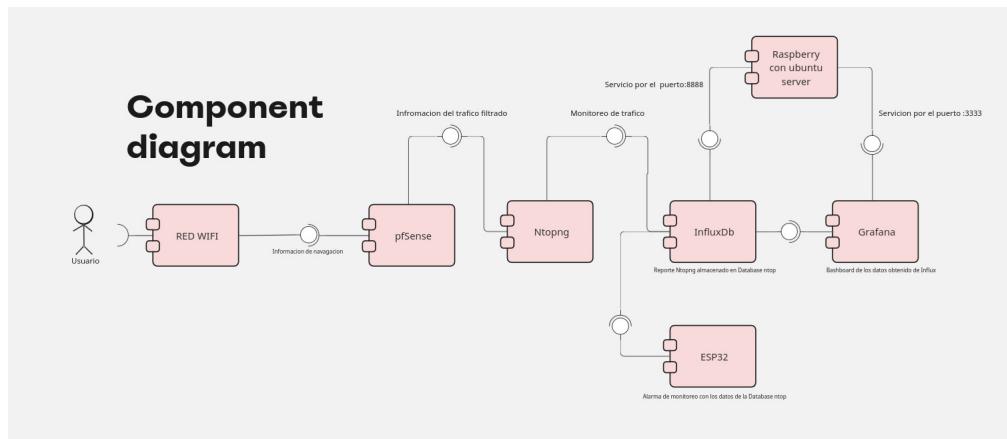


Figura 36: Diagrama de Componentes.

4.6. Pruebas.

Nuestra prueba de consulta a través de un ESP32 fue para verificar el host que ocupa más bytes, que está registrado en InfluxDB.

4.6.1. Comunicación entre InfluxDB y el ESP32.

La comunicación con InfluxDB se llevó a cabo utilizando la librería de InfluxDB para las versiones v1 y v2. Dentro de la documentación de la librería, se encuentra información y comentarios sobre cómo adaptarla según la versión que se esté utilizando y la placa. Utilizando uno de los ejemplos de consultas que se incluyen en la librería, se realizaron pruebas de conexión a la base de datos con éxito. Las siguientes tablas (2) representan las columnas de la consulta a la tabla 'host:traffic':

Cuadro 2: Descripción de los datos consultados a través de la ESP.

result	table	_start	_stop	_time	_value	_field	_measurement	host	ifid
14:23:32.559	string	long	dateTime:RFC3339	dateTime:RFC3339	dateTime:RFC3339	double	string	string	string

Así como se muestra en la siguiente imagen (ver Figura 37), la información dentro de la tabla se desglosa.

```

Actividades Arduino IDE 8 de abr 2:16 PM
QueryTable | Arduino IDE 2.3.2
File Edit Sketch Tools Help
✓ → ↻ NodeMCU 1.0 (ESP-12E Module)
QueryTable.ino
140 // Close the result
141 result.close();
142
143 //Wait 10s
144 Serial.println("Wait 10s");
145 delay(10000);
146 }
147
148
Output Serial Monitor ×
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on '/dev/ttyUSB0')
New Line 115200 baud
14:06:21.496 -> === List results ===
14:06:21.496 -> Querying with: from(bucket: "ntop") |> range(start: -24h) |> filter(fn: (r) => r._measurement == "host:traffic") |> limit(n: 10)
14:06:21.656 -> Table:
14:06:21.656 -> result,table,_start,_stop,_time,_value,_field,_measurement,host,ifid,
14:06:21.656 -> string,long,dateTime:RFC3339,dateTime:RFC3339,double,string,string,string,
14:06:21.656 -> _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:40:04Z,0,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.688 -> _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:41:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.688 -> _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:42:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.720 -> _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:43:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.720 -> _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:44:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.720 -> _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:45:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.755 -> _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:46:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.755 -> _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:47:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.788 -> _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:48:00Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.788 -> _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:49:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.788 -> _result,1,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:40:04Z,0,bytes_sent,host:traffic,192.168.1.1,0,
14:06:21.820 -> _result,1,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:42:01Z,1517,bytes_sent,host:traffic,192.168.1.1,0,
14:06:21.820 -> _result,1,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:43:01Z,1517,bytes_sent,host:traffic,192.168.1.1,0,
14:06:21.820 -> _result,1,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:43:01Z,1517,bytes_sent,host:traffic,192.168.1.1,0,
```

Figura 37: Arduino IDE: Consulta a la tabla host:traffic.

5. Implementación del Sistema y Pruebas.

En esta sección, se detalla el proceso de implementación del sistema desarrollado y las pruebas realizadas para verificar su funcionamiento y rendimiento. Se presentan los componentes hardware y software utilizados, así como los pasos seguidos para la configuración y puesta en marcha del sistema. Además, se describen las pruebas realizadas y los resultados obtenidos durante el proceso de validación.

5.1. Designación de Hardware y Software utilizado.

En esta subsección, se enumeran los componentes hardware y software que forman parte del sistema implementado. Se incluyen detalles sobre los dispositivos utilizados, como la Micro SD, el cargador, la Raspberry Pi, el ESP32, entre otros. Asimismo, se mencionan las versiones de software empleadas, como Ubuntu Server, InfluxDB, Grafana, Ntop Community Edition, Pfsense y Arduino IDE. Esta designación es fundamental para comprender la infraestructura del sistema y su entorno de desarrollo.

Hardware	Software
Micro SD Adata v30 A2	Ubuntu Server 23.10
Cargador PWR+ de 5v 3.3A	Influx 1.8.10
Raspberry Pi 3 Model B	Grafana
ESP 32	Ntop Community
	Pfsense 2.7.2
	Arduino IDE

Cuadro 3: Hardware y Software requeridos dentro del proyecto.

5.2. Despliegue y ejecución del sistema.

El sistema fue sometido a pruebas mediante la integración de las herramientas ntop, Influx y Grafana. Estas pruebas se llevaron a cabo a través de la plataforma ESP32, conectándola a la base de datos de Influx designada como ntop, donde se registran los datos pertinentes. Se realizó una consulta específica al apartado 'host:traffic', el cual, en comparación con otras tablas almacenadas en la base de datos de ntop, contiene información detallada sobre los bytes enviados y recibidos, así como la hora y la dirección IP del host correspondiente.

Con el propósito de identificar el host que está consumiendo mayor ancho de banda, se implementó una consulta utilizando Arduino IDE. Esta consulta tenía como objetivo devolver en el monitor serial la identificación del host que presente el mayor consumo de recursos dentro de la red. En el transcurso de estas pruebas, se identificó que el host con la dirección IP 192.168.0.13, correspondiente a un equipo utilizado por un alumno que reproducía un video de YouTube, era el que generaba un mayor tráfico. La información obtenida fue corroborada utilizando el Dashboard de Ntopng [26], el cual confirmó que durante la realización de estas pruebas, el equipo con la dirección IP 192.168.0.13 fue el principal consumidor de recursos en la red.

5.3. Pruebas.

Dentro de las puebas realizadas para la ESP, se desarrollaron dentro del software Arduino IDE donde se dividió en tres pasos, conexión de la ESP a internet, la consulta a la tabla general de host:traffic que es nuestra tabla de interés y la consulta de el host dentro de host:traffic que avarca mas ancho de banda.

5.3.1. Pruebas de conexión a internet.

La compatibilidad de las librerías con la ESP fue un conflicto, especialmente por el uso de conexiones a internet. También se observó que, independientemente del código, este ocuparía un espacio significativo de la RAM de la placa [27]. Este fue un inconveniente que surgió dentro del comportamiento de la placa; sin embargo, no compromete la eficiencia del algoritmo. También hubo problemas con `emptool.py` [28], aunque Arduino comúnmente no utiliza este script debido a que la librería está más influyente a otro tipo de placas tuvo que ser actualizado en el equipo de consulta.

```

Actividades Arduino IDE 8 de abr 2:51 PM sketch_apr8a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
sketch_apr8a.ino NodeMCU 1.0 (ESP-12E Module) ...
sketch_apr8a.ino
11 Serial.println();
12 Serial.print("Conectando a ");
13 Serial.println(ssid);
14
15 WiFi.begin(ssid, password);
16
17 while (WiFi.status() != WL_CONNECTED) {
18   delay(500);
19   Serial.print(".");
20 }
21
22 Serial.println("");
23 Serial.println("Conexión WiFi establecida");
24 Serial.print("Dirección IP: ");
25 Serial.println(WiFi.localIP());
26
27 void loop() {
28   // Tu código aquí
29 }
30
31

```

Output Serial Monitor >

Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on '/dev/ttyUSB0')

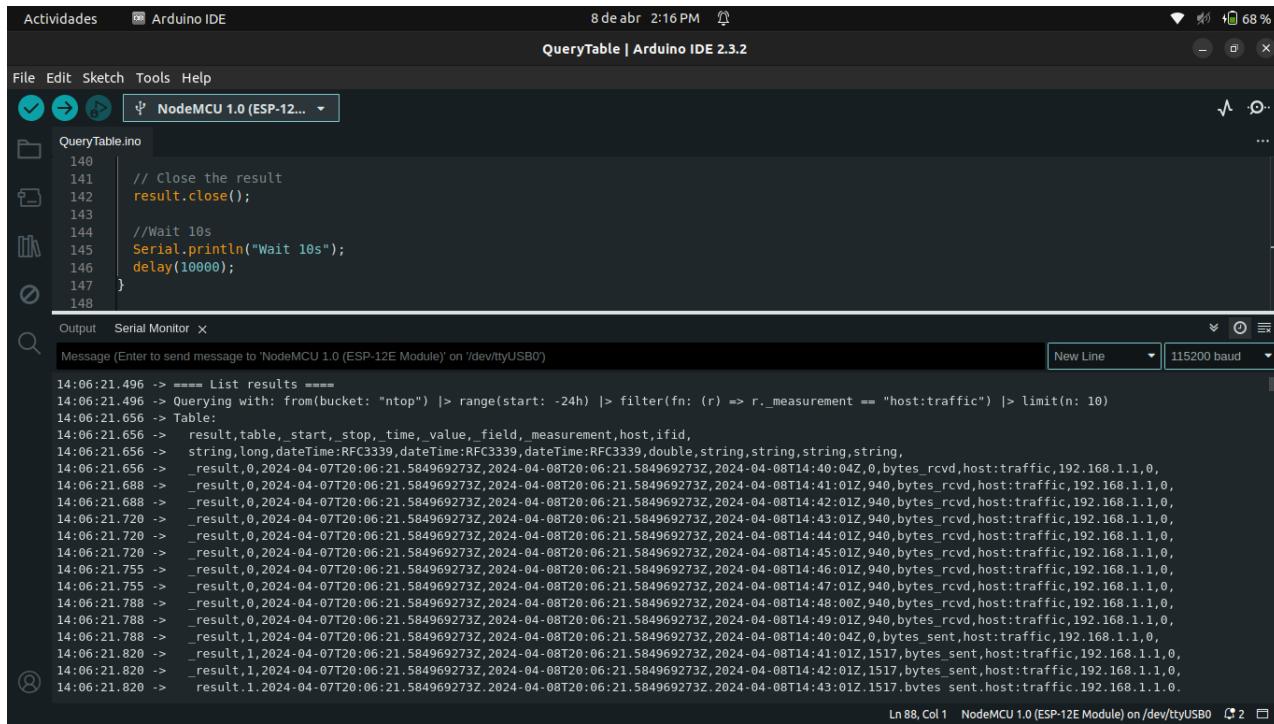
14:51:53.457 ->
14:51:57.592 -> Conexión WiFi establecida
14:51:57.592 -> Dirección IP: 192.168.0.107

Ln 4, Col 36 NodeMCU 1.0 (ESP-12E Module) on /dev/ttyUSB0 115200 baud

Figura 38: Arduino IDE: Conexión WiFi exitosa.

5.3.2. Pruebas de conexión a Influxdb: Tabla Host:traffic.

Dentro de la librería y documentación de InfluxDB se encuentran ejemplos de consultas, así como también cómo adaptarlo a los lineamientos de nuestra investigación. Fue posible acceder con éxito a través de InfluxDB, recordando que la función de consulta por Flux debe estar activada dentro de la configuración de Influx.



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** Actividades, Arduino IDE, 8 de abr 2:16 PM, 68 %
- File Menu:** File, Edit, Sketch, Tools, Help
- Sketch Selection:** NodeMCU 1.0 (ESP-12E)
- Code Editor:** QueryTable.ino


```

140 // Close the result
141 result.close();
142
143 //Wait 10s
144 Serial.println("Wait 10s");
145 delay(10000);
146
147 }
148
      
```
- Output Window:** Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on '/dev/ttyUSB0')


```

14:06:21.496 -> ===== List results =====
14:06:21.496 -> Querying with: from(bucket: "ntp") |> range(start: -24h) |> filter(fn: (r) => r._measurement == "host:traffic") |> limit(n: 10)
14:06:21.656 -> Table:
14:06:21.656 ->   result,table,_start,_stop,_time,_value,_field,_measurement,host,ifid,
14:06:21.656 ->   string,long,dateTime:RFC3339,dateTime:RFC3339,double,string,string,string,
14:06:21.656 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:40:04Z,0,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.688 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:41:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.688 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:42:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.720 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:43:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.720 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:44:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.720 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:45:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.755 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:46:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.755 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:47:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.788 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:48:00Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.788 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:49:00Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.820 ->   _result,1,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:40:04Z,0,bytes_sent,host:traffic,192.168.1.1,0,
14:06:21.820 ->   _result,1,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:41:01Z,1517,bytes_sent,host:traffic,192.168.1.1,0,
14:06:21.820 ->   _result,1,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:42:01Z,1517,bytes_sent,host:traffic,192.168.1.1,0,
14:06:21.820 ->   _result,1,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:43:01Z,1517,bytes_sent,host:traffic,192.168.1.1,0.
      
```
- Serial Monitor:** New Line, 115200 baud

Figura 39: Arduino IDE: Consulta a la tabla host:traffic.

5.3.3. Pruebas de conexión a Influx: Host que utiliza más recursos.

La prueba del host con los bytes utilizados dentro del servicio fue un proceso más fácil debido a las pruebas anteriores, y a la verificación de los datos de consulta dentro de la tabla *host_traffic* que necesitábamos, tales como la dirección IP, bytes enviados y bytes recibidos, los cuales utilizamos para calcular el host con más bytes utilizados en la red.

```

Actividades Arduino IDE 8 de abr 7:03 PM sketch_apr8b | Arduino IDE 2.3.2
File Edit Sketch Tools Help
sketch_apr8b.ino NodeMCU 1.0 (ESP-12E ... )
135 // Imprime la IP con la mayor cantidad de bytes
Output Serial Monitor
Not connected. Select a board and a port to connect automatically.
19:01:15.212 -> 192.168.1.13 3786102 3786102
19:01:15.212 -> 192.168.1.13 3786570 3786570
19:01:15.212 -> 192.168.1.13 3787686 3787686
19:01:15.212 -> 192.168.1.13 0 0
19:01:15.212 -> 192.168.1.13 7228 7228
19:01:15.212 -> 192.168.1.13 34242 34242
19:01:15.212 -> 192.168.1.13 299361 299361
19:01:15.244 -> 192.168.1.13 332605 332605
19:01:15.244 -> 192.168.1.13 332605 332605
19:01:15.244 -> 192.168.1.13 333399 333399
19:01:15.244 -> 192.168.1.13 334107 334107
19:01:15.244 -> 192.168.1.13 334551 334551
19:01:15.244 -> 192.168.1.13 334923 334923
19:01:15.277 -> 192.168.1.14 0 0
19:01:15.277 -> 192.168.1.14 6595 6595
19:01:15.277 -> 192.168.1.14 6595 6595
19:01:15.277 -> 192.168.1.14 6595 6595
19:01:15.277 -> 192.168.1.14 6595 6595
19:01:15.277 -> 192.168.1.14 6595 6595
19:01:15.277 -> 192.168.1.14 0 0
19:01:15.309 -> 192.168.1.14 3134 3134
19:01:15.309 -> 192.168.1.14 3134 3134
19:01:15.309 -> 192.168.1.14 3134 3134
19:01:15.309 -> 192.168.1.14 3134 3134
19:01:15.309 -> 192.168.1.14 3134 3134
19:01:15.309 -> IP with max bytes: 192.168.1.13
19:01:15.309 -> Wait 10s

```

Ln 142, Col 99 NodeMCU 1.0 (ESP-12E Module) on /dev/ttyUSB0 [not connected] 2

Figura 40: Arduino IDE: Consulta del Host con mas bytes utilizados.

5.4. Resultados.

A continuación se presentan los resultados de las configuraciones obtenidas durante el período de estadía.

5.4.1. Ntop con Influx.

La conexión de ntop a Influx a través de una Raspberry Pi fue exitosa, configurada directamente desde las preferencias de configuración de ntop [29]. Este logro representa un avance significativo en la integración y el aprovechamiento de recursos tecnológicos para mejorar la monitorización y el análisis de datos en el sistema de red. La capacidad de configurar esta conexión de manera sencilla y efectiva desde la interfaz de usuario de ntop demuestra la versatilidad y la accesibilidad de estas herramientas para la administración de redes.

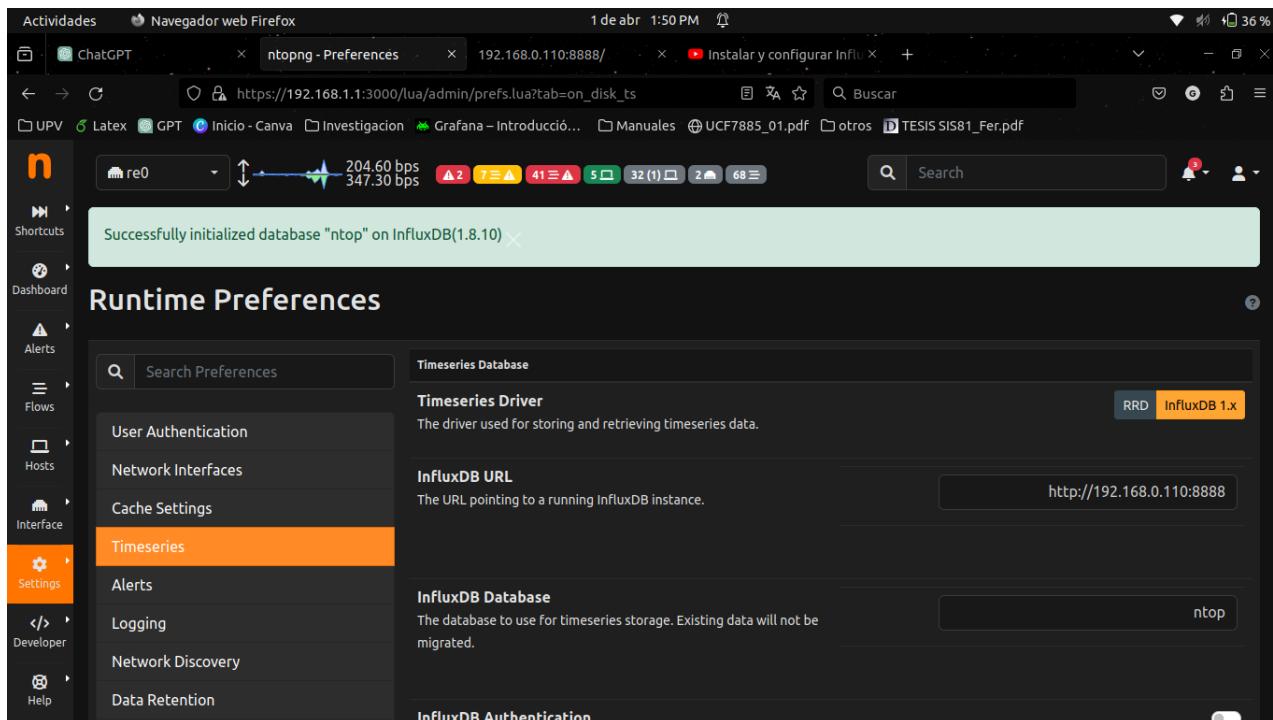


Figura 41: Ntopng: Configuración de influxDb dentro de Ntopng dentro del apartado de preferences.

5.4.2. Influx con Grafana.

La exitosa conexión de Influx a Grafana a través de los puertos 8888 y 3333. Esta integración permite aprovechar las capacidades de Grafana para crear paneles dinámicos y personalizados que muestran los datos almacenados en Influx de manera clara y efectiva.

La configuración dentro de Grafana para establecer esta conexión demuestra la flexibilidad y la capacidad de adaptación de la plataforma para trabajar con diferentes fuentes de datos, lo que facilita la creación de visualizaciones personalizadas y la realización de análisis detallados.

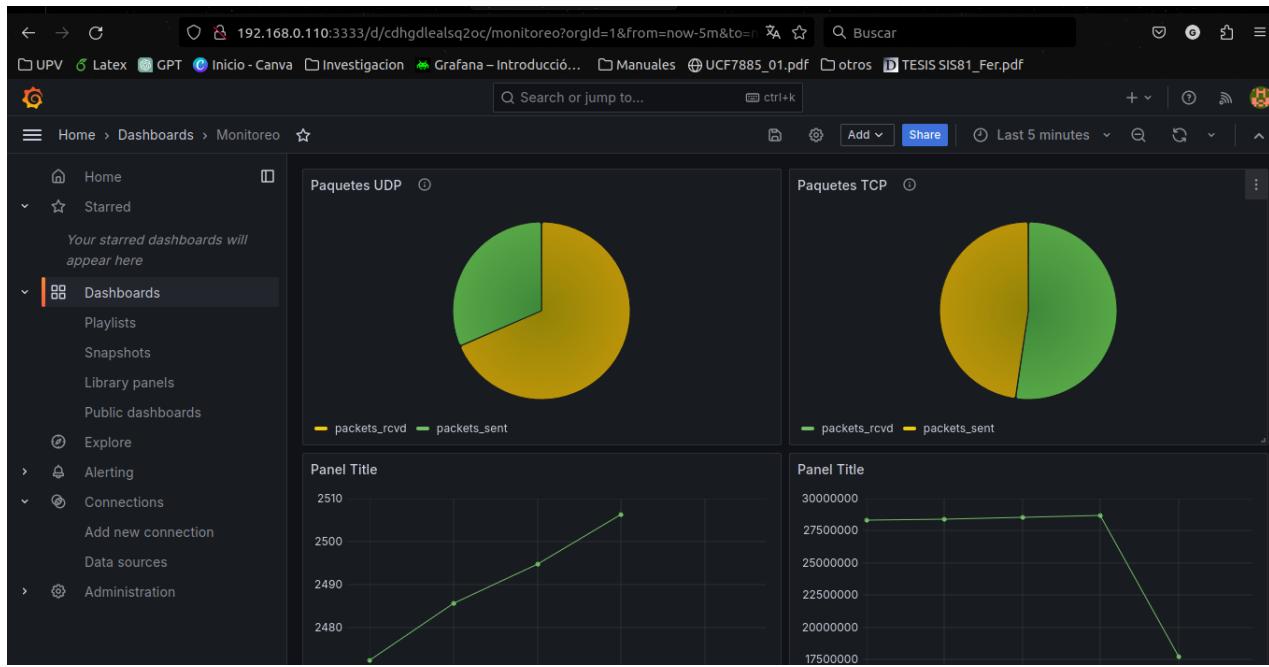
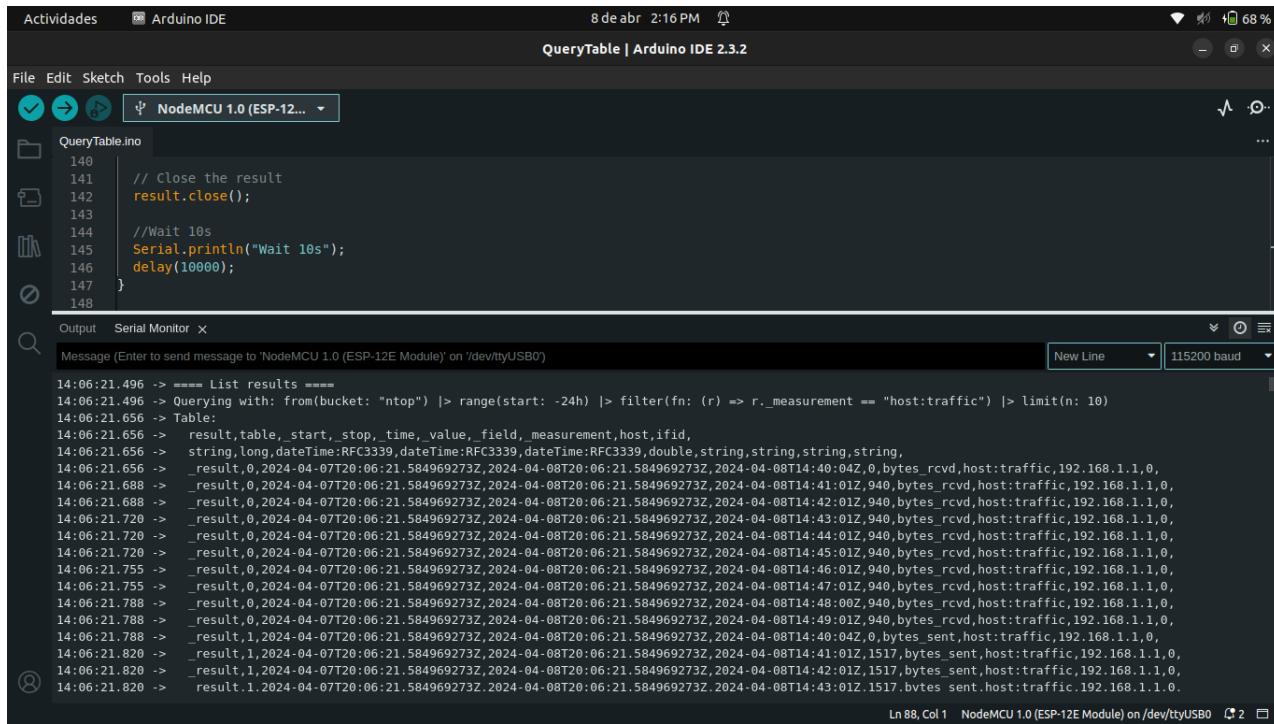


Figura 42: Grafana: Dashboard de las conexiones a Influx.

5.4.3. Consulta Influx desde ESP32.

La exitosa conexión de Influx a un dispositivo ESP a través de la biblioteca oficial de InfluxDB para consultas representa un avance significativo en la capacidad de integración de datos en tiempo real entre dispositivos IoT y sistemas de gestión de datos. Esta integración permite que el dispositivo ESP acceda a datos almacenados en InfluxDB [30] y los utilice para diversos fines, como el monitoreo de sensores, el control de dispositivos, o la recopilación de información para análisis posterior.



```

Actividades 8 de abr 2:16 PM
Arduino IDE
File Edit Sketch Tools Help
QueryTable | Arduino IDE 2.3.2
File Edit Sketch Tools Help
QueryTable.ino
140 // Close the result
141 result.close();
142
143 //Wait 10s
144 Serial.println("Wait 10s");
145 delay(10000);
146
147 }
148
Output Serial Monitor x
Message (Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on '/dev/ttyUSB0')
New Line 115200 baud
14:06:21.496 -> ===== List results =====
14:06:21.496 -> Querying with: from(bucket: "ntp") |> range(start: -24h) |> filter(fn: (r) => r._measurement == "host:traffic") |> limit(n: 10)
14:06:21.656 -> Table:
14:06:21.656 ->   result,table,_start,_stop,_time,_value,_field,_measurement,host,ifid,
14:06:21.656 ->   string,long,dateTime:RFC3339,double,string,string,string,
14:06:21.656 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:40:04Z,0,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.688 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:41:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.688 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:42:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.720 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:43:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.720 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:44:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.720 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:45:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.755 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:46:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.755 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:47:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.788 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:48:00Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.788 ->   _result,0,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:49:01Z,940,bytes_rcvd,host:traffic,192.168.1.1,0,
14:06:21.820 ->   _result,1,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:40:04Z,0,bytes_sent,host:traffic,192.168.1.1,0,
14:06:21.820 ->   _result,1,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:41:01Z,1517,bytes_sent,host:traffic,192.168.1.1,0,
14:06:21.820 ->   _result,1,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:42:01Z,1517,bytes_sent,host:traffic,192.168.1.1,0,
14:06:21.820 ->   _result,1,2024-04-07T20:06:21.584969273Z,2024-04-08T20:06:21.584969273Z,2024-04-08T14:43:01Z,1517,bytes_sent,host:traffic,192.168.1.1,0.

```

Figura 43: Arduino IDE: Consulta a la tabla host:traffic.

5.4.4. Prueba de alarma.

Se realiza la validación del host que consume más bytes consultando la tabla "host:traffic" dentro de InfluxDB, la cual recopila la información de Ntop. Este proceso se lleva a cabo utilizando el umbral rojo definido dentro del código. Si el host que ocupa más bytes está dentro de ese parámetro, se encenderá el LED azul; de lo contrario, se encenderá el LED rojo como se ve en la imagen 45.

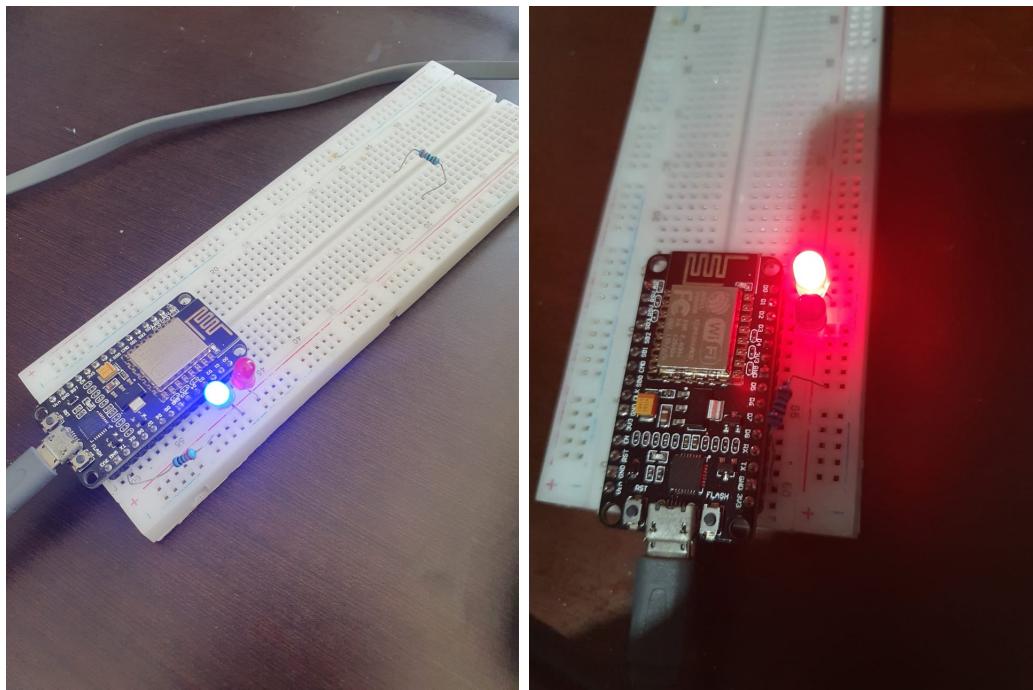


Figura 44: Prueba de la alarma con LED's rojo y azul para determinar cuándo un host está usando más recursos o si el host sigue los parámetros designados.

La última modificación realizada, según la observación del evaluador empresarial, consistió en reemplazar los LEDs por un buzzer, con el propósito de proporcionar una reacción sonora para la alarma, tal como lo solicitaron los responsables de la red.

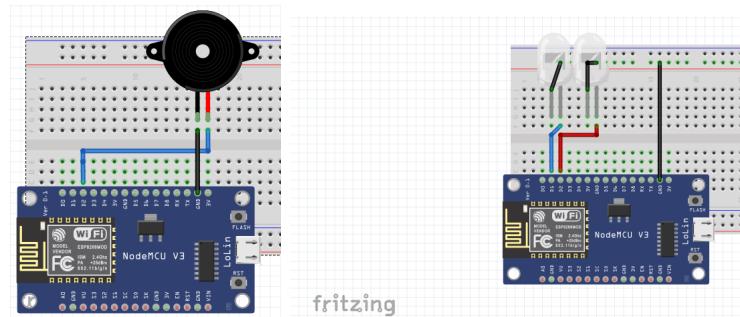


Figura 45: Conexión de la alarma con buzzer y con led's.

6. Conclusiones y Trabajo Futuro.

En la siguiente sección se mencionan las conclusiones obtenidas de acuerdo a la realización del presente proyecto de estadía, así como también el trabajo futuro, toda prueba que no se llegó a realizar y se planea que se puede implementar para el mejoramiento del sistema.

6.1. Conclusiones.

En el marco de este proyecto, se llevó a cabo el desarrollo de una alarma de tráfico de red destinada al sistema de red de la Universidad Politécnica de Victoria. Esta iniciativa se apoyó en la utilización de sistemas de monitoreo como NTopng y Pfsense, además de emplear herramientas como Influx, Grafana y Arduino IDE para el procesamiento y consulta de datos. La placa ESP8266 se desplegó con el propósito de ejecutar las consultas pertinentes y adaptar el modelo de alarma a su forma física, permitiendo así una integración efectiva entre el entorno digital y el mundo real.

El proyecto se llevó a cabo con éxito, cumpliendo cabalmente con cada uno de los objetivos establecidos desde su fase inicial. A pesar de los desafíos encontrados durante el proceso, tales como la detección de materiales defectuosos y errores en las instalaciones, que provocaron contratiempos y periodos de inactividad, estos obstáculos fueron superados con eficacia, lo que permitió alcanzar una culminación satisfactoria del proyecto. La vinculación de NTop con Influx posibilitó la explotación de informes de consulta que no podían ser aprovechados desde su reporte principal de host en un formato .json, ofreciendo así una mayor amplitud de análisis y perspectivas en el estudio del tráfico de red.

En conclusión, la confluencia de tecnologías y la resolución efectiva de desafíos técnicos permitieron la realización exitosa de la alarma de tráfico de red para la Universidad Politécnica de Victoria.

6.2. Trabajo Futuro

Se han identificado varias áreas para futuras mejoras que podrían enriquecer y ampliar la funcionalidad del sistema desarrollado. Una de estas áreas clave es la realización de pruebas en redes más extensas para designar los promedios utilizados por sectores, como los edificios de la institución, con el fin de monitorear un mayor número de usuarios y determinar aquellos que podrían estar generando congestión en la red. Este enfoque permitiría identificar patrones de uso y posibles cuellos de botella en el tráfico de la red, lo que a su vez podría contribuir a una gestión más eficiente y a una mejor optimización del ancho de banda disponible.

Otra área de importancia para el desarrollo futuro del sistema es la implementación de herramientas especializadas en la detección de comportamientos anómalos en el consumo de datos. Se sugiere explorar métodos que permitan diferenciar entre un consumo normal y anormal por parte de los dispositivos conectados a la red. Esto podría abarcar desde la identificación de actividades maliciosas, como la presencia de malware, hasta la detección de tráfico sospechoso. La capacidad de identificar y mitigar tales amenazas sería fundamental para garantizar la seguridad y la integridad de la red.

Índice de figuras

1.	Conexión de dispositivos totalmente activos y pasivos correspondientes	6
2.	Version de Ntopng utilizada en el proyecto.	8
3.	Demostración general de cómo funciona Ntop.	9
4.	Versión de NTopng del proyecto de Evaluación de herramientas TIC para gestionar el monitoreo y análisis de la red de datos del Recinto de Golfito de la Universidad de Costa Rica.	10
5.	Version de Ntop donde se muestra la forma de reporte del Top Talkers Relationship.	11
6.	Raspberry utilizada en el proyecto.	13
7.	Raspberry imager.	17
8.	Interfaz de Arduino IDE	18
9.	Arquitectura de la sistema de red local.	19
10.	Peticiones NMS.	20
11.	Arquitectura tipica de un DSMS	21
12.	Diferencias entra los DSMS y DBMS.	22
13.	Etapas del proceso de clasificación haciendo uso de técnicas de Machine Learning	24
14.	Interfaz de Ntop de la Evaluacion de herramientas TIC para gestionar el monitoreo y analisis de la red de datos	25
15.	Interfaz de Wireshark de la Evaluacion de herramientas TIC para gestionar el monitoreo y analisis de la red de datos	26
16.	Gráfica de pastel con tiempos distribuidos.	28
17.	Pfsense: Inicio de Proyecto.	30
18.	Pfsense: Package Manager Error solucionado.	31
19.	Pfsense: Configuración de la instalación de Ntopng en el sistema.	32
20.	Pfsense: Especificaciones del Pfsense.	32
21.	Pfsense: Gráfica de tráfico vista desde Pfsense.	32
22.	NTopng: Inicio de Proyecto.	33
23.	Ntopng: Vistazo al Networkchart dentro de Ntopng.	34
24.	Ntopng: Vistazo a la infomacion de host chart dentro de Ntopng.	34
25.	Ntopng: Vistazo de los Top Host dentro del Dashboard de Ntopng.	35
26.	Red del pfSense dada por el access point conectado al pfSense.	35
27.	NTopng: Informacion del reporte de flujo en el formato .json	36
28.	Ntopng: Informacion de un host designado dentro de la red.	37
29.	Ntopng: Información de un host designado dentro de la red.	37
30.	Raspberry y Micro SD que se usaron en el proyecto.	38
31.	Configuración de la Raspberry Pi como sistema final.	38
32.	InfluxDb: Consulta de los datos almacenados dentro de la base de datos ntop vinculada a ntopng.	39
33.	Ntopng: Configuración de InfluxDB dentro de Ntopng en el apartado de preferencias.	40
34.	InfluxDB: Consulta de los datos almacenados dentro de la base de datos ntop vinculada a ntopng.	41
35.	Caso de uso: Monitoreo de un usuario en la red.	42

36.	Diagrama de Componentes	42
37.	Arduino IDE: Consulta a la tabla host:traffic.	43
38.	Arduino IDE: Conexión WiFi exitosa.	45
39.	Arduino IDE: Consulta a la tabla host:traffic.	46
40.	Arduino IDE: Consulta del Host con mas bytes utilizados.	47
41.	Ntopng: Configuración de influxDb dentro de Ntopng dentro del apartado de preferences.	48
42.	Grafana: Dashboard de las conexiones a Influx.	49
43.	Arduino IDE: Consulta a la tabla host:traffic.	50
44.	Prueba de la alarma con LED's rojo y azul para determinar cuándo un host está usando más recursos o si el host sigue los parámetros designados.	51
45.	Conexión de la alarma con buzzer y con led's.	51

Índice de cuadros

1.	Cronograma de actividades	29
2.	Descripción de los datos consultados atraves de la ESP.	43
3.	Hardware y Software requeridos dentro del proyecto.	44

Referencias

- [1] Damián Victorio González y Cruz Quezada Carrasco. “Sistema de monitoreo y análisis de tráfico en la red”. En: *Licenciatura en Ingeniería en Sistemas Computacionales* (2015).
- [2] *pfSense Documentation — pfSense Documentation*. <https://docs.netgate.com/pfsense/en/latest/>. (Accessed on 03/26/2024).
- [3] Christopher M Buechler y Jim Pingle. “pfsense: The definitive guide”. En: *Reed Media Services* (2009).
- [4] Cerritos Magaña y Vidal Enrique. “Firewall PfSense como opción de código abierto para la seguridad de la red informática en centros escolares católicos de Santa Ana”. En: (2018).
- [5] *ntopng Documentation — ntopng 6.1 documentation*. <https://www.ntop.org/guides/ntopng/>. (Accessed on 03/26/2024).
- [6] Emil Gatial, Zoltán Balogh y Ladislav Hluchý. “Concept of energy efficient ESP32 chip for industrial wireless sensor network”. En: *2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES)*. IEEE. 2020, págs. 179-184.
- [7] Oleksii Barybin, Elina Zaitseva y Volodymyr Brazhnyi. “Testing the security ESP32 internet of things devices”. En: *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*. IEEE. 2019, págs. 143-146.
- [8] Raspberry Pi. “Raspberry pi 3 model b”. En: *online].(https://www.raspberrypi.org* (2015).
- [9] Eben Upton y Gareth Halfacree. *Raspberry Pi user guide*. John Wiley & Sons, 2016.
- [10] InfluxDB. *InfluxData Documentation*. <https://docs.influxdata.com/>. (Accessed on 03/26/2024).
- [11] Mainak Chakraborty y Ajit Pratap Kundan. “Grafana”. En: *Monitoring cloud-native applications: Lead agile operations confidently using open source software*. Springer, 2021, págs. 187-240.
- [12] Claudio Peña. *Arduino IDE: Domina la programación y controla la placa*. RedUsers, 2020.
- [13] Omar Jesús Fernández Huaytalla. “Implementación de un servidor como gestión y monitoreo de servicios para la red de datos en la UGEL Huamanga, 2018”. En: (2019).
- [14] Felipe Andrés Luco Pacheco y Brian Valenzuela Ramos. “MONITOREO DE DATOS MEDIANTE UN ADMINISTRADOR DE FLUJO DE DATOS.” En: () .
- [15] Gianpaolo Cugola y Alessandro Margara. “Processing flows of information: From data stream to complex event processing”. En: *ACM Computing Surveys (CSUR)* 44.3 (2012), págs. 1-62.
- [16] Daruin Arley León et al. “Inteligencia artificial para el control de tráfico en redes de datos: Una Revisión”. En: *Entre Ciencia e Ingeniería* 16.31 (2022), págs. 17-24.
- [17] Enrique Colon Ferruzola Gómez, Oscar Xavier Bermeo Almeida, Lissett Margarita Arévalo Gamboa et al. “Análisis de los sistemas centralizados de seguridad informática a través de la herramienta AlienVault Ossim”. En: *Ecuadorian Science Journal* 6.1 (2022), págs. 23-31.

- [18] Alan Martín Corrales Rodríguez. “Evaluación de herramientas TIC para gestionar el monitoreo y análisis de la red de datos del Recinto de Golfito de la Universidad de Costa Rica”. En: (2020).
- [19] Nancy Estefanía Zapata Tapia. “Nodo de gestión y monitoreo de calidad de servicio de la empresa Ajnet en el cantón Latacunga.” B.S. thesis. Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas . . ., 2023.
- [20] Christian Hernández, Raúl H Palacios, Felipe de Jesús Núñez-Cárdenas et al. “Instalación y configuración de Pfsense en máquina virtual con VirtualBox”. En: *Ciencia Huasteca Boletín Científico de la Escuela Superior de Huejutla* 11.22 (2023), págs. 22-31.
- [21] *How To Create a Database on InfluxDB 1.7 & 2.0 – devconnected*. <https://devconnected.com/how-to-create-a-database-on-influxdb-1-7-2-0/>. (Accessed on 04/03/2024).
- [22] *InfluxDB 2.0 Support · Issue 3764 · ntop/ntopng — github.com*. <https://github.com/ntop/ntopng/issues/3764>. [Accessed 04-04-2024].
- [23] *InfluxDB data source — Grafana documentation — grafana.com*. <https://grafana.com/docs/grafana/latest/datasources/influxdb/>. [Accessed 04-04-2024].
- [24] Steve Anthony. “Metrics with Influx and Grafana”. En: (2018).
- [25] Tiia Leppänen. “Data visualization and monitoring with Grafana and Prometheus”. En: (2021).
- [26] *Dashboard &x2014; ntopng 6.1 documentation — ntop.org*. https://www.ntop.org/guides/ntopng/web_gui/dashboard.html. [Accessed 04-04-2024].
- [27] *¿Como hacer consultas a InfluxDB de varias columnas? — forum.arduino.cc*. <https://forum.arduino.cc/t/como-hacer-consultas-a-influxdb-de-varias-columnas/699146>. [Accessed 04-04-2024].
- [28] *Envío de datos por Wifi con ESP32, Influx-DB y Grafana - Smart Open Lab — smartopenlab.com*. <https://smartopenlab.com/proyecto/envio-de-datos-por-wifi-con-esp32-influx-db-y-grafana/>. [Accessed 04-04-2024].
- [29] *Integración de InfluxDB2 con Grafana — de Nandita Sahu — Medio*. https://medium.com.translate.google/@nanditasahu031/integration-of-influxdb2-with-grafana-28b4aebb3368?_x_tr_sl=auto&_x_tr_tl=es&_x_tr_hl=es&_x_tr_hist=true. (Accessed on 04/03/2024).
- [30] *Use the InfluxDB Arduino client library — InfluxDB Cloud (TSM) Documentation*. <https://docs.influxdata.com/influxdb/cloud/api-guide/client-libraries/arduino/>. (Accessed on 04/03/2024).