Informe final proyecto NetBy

Johan Eduardo Cala Torra
Facultad de ingeniería de Sistemas e Informática
Universidad Pontificia Bolivariana
Bucaramanga, Colombia
johan.cala.2019@upb.edu.co

Duvan Andres Diaz Montañez
Facultad de ingeniería de Sistemas e Informática
Universidad Pontificia Bolivariana
Bucaramanga, Colombia
duvan.diaz.2019@upb.edu.co

Angela Sofia Remolina Gutiérrez
Facultad de ingeniería de Sistemas e Informática
Universidad Pontificia Bolivariana
Bucaramanga, Colombia
angela.remolina.2019@upb.edu.co

***Resumen* — En este informe se presenta el proyecto NetBy, que es un servicio web que permite generar capturas de red por parte de los usuarios (clientes y administradores) para después analizar a detalle el tráfico de red doméstica que ocurrió durante esa captura, la cantidad de información transitante y el consumo de ancho de banda de algunos servicios.**

A continuación, se da un contexto de donde nace la idea del proyecto, los objetivos planteados para resolver la problemática, la metodología que se llevó a cabo para el desarrollo del aplicativo web NetBy y los resultados que se pudieron obtener a partir del producto final desarrollado y un conjunto de pruebas realizadas en un ambiente controlado.

Palabras claves — Red de datos, tráfico de red, sniffing de paquetes de red

***Abstract* — This document presents the Project NetBy, which is a web service that allows users to generate network captures (clients and administrators) to later analyze in detail the home network traffic that occurred during that capture, the amount of information transit and bandwidth consumption of some services.**

Next in the document, a context is given from which the idea of the project was born, the objectives set to solve the problem, the methodology that was carried out for the development of the web application NetBy and the results that could be obtained from the final product developed and a set of tests performed in a controlled environment.

Keywords — Computer network, network traffic, network packet sniffing

I. INTRODUCCIÓN

La situación de salud pública actual ha motivado enormemente los esquemas de teletrabajo y educación con apoyo de medios virtuales, todo esto directamente involucrado con el consumo de canal de datos o Internet. Debido a la pandemia se estimó que durante la cuarentena en aumento en un 38% el consumo de internet [1]. Con Colombia lo anterior, se ha visto forzado a optimizar el uso del ancho de banda con el que se cuenta en cada residencia, donde han ingresado dispositivos IoT o móviles que podrían afectar negativamente nuestro trabajo académico (meeting, aula virtual, VPN, entre otros).

De qué manera un usuario podrá visualizar sus estadísticas de red donde este pueda modificar parámetros, configuraciones y pueda observar lo más detallado posible lo que está ocurriendo en su red. Para dar solución a esta problemática se planteó esta pregunta de investigación ¿Cómo identificar y clasificar las conexiones de servicios de red en un rango de tiempo especificado, según acciones (consumo, peticiones, etc.) realizadas por los usuarios?

Para responder esta pregunta se propuso el proyecto NetBy que presentará una plataforma amigable para que el usuario pueda consultar detalles del tráfico de su red local. En este documento se presentan los objetivos propuestos para cumplir a cabalidad este proyecto, y los avances alcanzados hasta la fecha 5 de septiembre 2021.

II. MARCO CONCEPTUAL

Para el desarrollo del aplicativo NetBy es necesario tener los conocimientos, Redes de Datos donde se tienen en cuenta los siguientes protocolos.

El protocolo de Control de Transmisión (TCP), el cual es el Protocolo de comunicaciones orientado a la conexión que facilita el intercambio de mensajes entre dispositivos informáticos en una red [2]. Es el protocolo más común en las redes que utilizan el Protocolo de Internet (IP); juntos se denominan a veces TCP/IP.

El protocolo de datagramas de usuario (UDP) que funciona sobre el protocolo de Internet (IP) para transmitir datagramas a través de una red. [3] El UDP no requiere que el origen y el destino establezcan un apretón de manos de tres vías antes de que se produzca la transmisión. Además, no es necesaria una conexión de extremo a extremo.

El protocolo HTTP es cual la base de la World Wide Web y se utiliza para cargar páginas web mediante enlaces de hipertexto [4]. HTTP es un protocolo de capa de aplicación diseñado para transferir información entre dispositivos en red y se ejecuta sobre otras capas de la pila de protocolos de red. Un flujo típico a través de HTTP implica que una máquina cliente haga una petición a un servidor, que luego envía un mensaje de respuesta.

Dirección IP la cual es la abreviatura de la dirección del protocolo de Internet; es un número de identificación que se asocia con un ordenador o una red de ordenadores específicos [5]. Cuando se conecta a Internet, la dirección IP permite a los ordenadores enviar y recibir información.

Además, para el desarrollo de una captura de Red se hará uso de un Sniffer el cual es una aplicación especial para redes informáticas, que permite como tal capturar los paquetes que viajan por una red. Este es el concepto más sencillo que podemos dar al respecto, pero profundizando un poco más podemos decir también que un sniffer puede capturar paquetes dependiendo de la topología de red [6].

Luego se debe analizar la información para identificar en que capa del modelo OSI se ubica, el modelo OSI es un marco conceptual utilizado para describir las funciones de un sistema de red. El modelo OSI caracteriza las funciones informáticas en un conjunto universal de reglas y requisitos con el fin de apoyar la interoperabilidad entre diferentes productos y software. En el modelo de referencia OSI, las comunicaciones entre un sistema informático se dividen en siete capas de abstracción diferentes: Física, Enlace de Datos, Red, Transporte, Sesión, Presentación y Aplicación.[7].

También se necesitó indagar sobre la Ing. De Software donde se tiene en cuenta los Casos de Uso, que es un artefacto que define una secuencia de acciones que da lugar a un resultado de valor observable [8]. Los casos de uso proporcionan una estructura para expresar requisitos funcionales de nuestro aplicativo.

Se tiene en cuenta el Modelo Entidad Relación es cual es una representación pictórica o visual de la clasificación de grupos o entidades de interés común y la definición de la relación entre estos grupos [9]. Por lo tanto, se crea una estructura con varios símbolos de diferentes formas y tamaños para que pueda ser utilizado como un modelo para representar la estructura interna y la relación.

Y por último se hizo una consulta sobre de base de datos, donde tenemos nuestra base de datos la cual es la implementación de modelo entidad relación, y sus consultas se ven realizadas por SQL (lenguaje de consulta estructurado), que es un lenguaje informático para almacenar, manipular y recuperar datos almacenados en una base de datos relacional. [10] SQL es el lenguaje estándar para los sistemas de bases de datos relacionales. Y como sistema de gestión de base de datos tenemos el Oracle el cual utiliza SQL como lenguaje de base de datos estándar.

Para realizar la conexión entre el aplicativo y la base de datos se utiliza NodeJs que es una plataforma construida sobre el tiempo de ejecución de JavaScript de Chrome para construir fácilmente aplicaciones de red rápidas y escalables [11]. Node.js utiliza un modelo de E/S basado en eventos y sin bloqueos que lo hace ligero y eficiente, perfecto para aplicaciones en tiempo real con gran cantidad de datos que se ejecutan en dispositivos distribuidos.

III. ESTADO DEL ARTE

Se hizo un estudio del estado del arte con el fin de identificar investigaciones, estudios o aplicaciones similares a NetBy que implementen una herramienta de sniffing dentro de las redes de datos para identificar el tráfico de red y analizar el funcionamiento de éstas.

Ya existen aplicaciones para el analisis del tráfico de red, lo que diferencia a NetBy de todas estás es que está planteada para que los mismos usuarios de la red sean quienes puedan ver sus capturas. Por esto se plantea como una herramienta web para que sea de facil acceso a todos. Actúa como una red social en la que se puede registrar, y compartir sus capturas de red con otros usuarios de la aplicación. En la tabla 1 se presenta una tabla comparativa entre las aplicaciones más usadas para estos fines.

TABLA I

COMPARATIVO BÁSICO ENTRE APLICACIONES EXISTENTES PARA EL ANÁLISIS DE TRÁFICO DE RED

Aplicación	**Sistemas operativos soportados**	**Open Source**	**Muestra el protocolo en capa modelo OSI**	**Interfaz de usuario**
WireShark	- Windows - Unix	Sí	Sí	GUI / CLI
Ettercap	- Windows - Unix based	Sí	No	CLI
TCPDump	- Unix based	Sí	No	CLI
Capsa	- Windows	No	Sí	GUI

Dadas las limitaciones establecidas para el desarrollo de NetBy, se analizó el uso de estas aplicaciones ya existentes, sus ventajas o desvetajas frente al desarrollo de una nueva. Vale la pena entonces, resaltar las librerías más usadas para esta técnica, si se quiere desarrollar una herramienta de rastreo propia.

Existen dos grandes librerías principalmente usadas para el fin de capurar la red: Libpcap y Libnet. A continuación, se presentarán algunas diferencias entre ellas:

TABLA II

LIBPCAP VS LIBNET

Característica	**Libpcap**	**Libnet**
Nivel de captura	- Captura los paquetes en nivel bajo. - Extrae el paquete de modo que el kernel sin tratos	- Manipula un tráfico de alto nivel. - Puede manipular varias rutinas de redes de bajo nivel.
Modo de uso	- Modo conectado (TCP) - Modo sin conexión (UDP)	- Modo conectado (TCP) - Modo sin conexión (UDP)
Filtrado	- Filtrado compatible con el filtro BPF. - Inicializa y configura filtros. - Recibe los paquetes mediante un bucle.	- Las dosis no proporcionan un filtrado de paquetes
Protocolos admitidos	- Admite casi todos los protocolos de red.	- Inyecta cualquier tipo de paquete IP. - Manipula un firewall de red (filtro IP, ipfw, ipchains, pf, PktFilter, ...). - Ofrece las funciones de manipulación de direcciones, la caché ARP y tablas de enrutamiento. - Manipula un túnel IP (tun BSD / Linux, Universal TUN / Dispositivo TAP).
Plataformas compatibles	Compatible con todas las plataformas.	- BSD (OpenBSD, FreeBSD, NetBSD, BSD / OS) - Linux (Redhat, Debian, Slackware, etc.) -MacOS X (Windows NT / 2000 / XP) - SolarisIRIX - HP-UX - Tru64

Tabla II. Adaptada de [14]

En esta investigación de rastreadores (sniffers) realizados desde cero, se encontraron varios proyectos similares a NetBy, que busca hacer un análisis del tráfico de red.

Entre ellos está un proyecto que tenía de objetivo la búsqueda de paquetes para el monitoreo del rendimiento de la red, utilizando Python [13]. La limitación que este presenta es que es solo servirá para el monitoreo de una red dada y este no se presenta al usuario de red, sino a quien tenga acceso a esta herramienta.

IV. OBJETIVOS

Objetivo General

Desarrollar una aplicación web que permita obtener la información general de la red conectada para visualizar un tablero de control que muestre la información mediante un panel compuesto de gráficos y tablas estadísticas construido con tecnologías de software libre.

Objetivos Específicos

- Definir los requerimientos funcionales y no funcionales del software, mediante un documento en forma de acta que muestre las limitaciones del sitio.
- Modelar la estructura del sitio web mediante un diagrama de entidad-relación que muestre las bases de datos para el almacenamiento de la información de la red y otro diagrama de bloques de etapas del software para mantener un flujo claro en el desarrollo.
- Desarrollar el back-end y front-end de la aplicación para que capture y almacene los datos de una base de datos SQL y después mostrarlos mediante gráficas y otros elementos web.
- Establecer un conjunto de casos de prueba para implementar el control de la funcionalidad del software mediante pruebas unitarias y de integración.

V. METODOLOGÍA

Se seguirá la metodología de desarrollo ágil Kanban: “Kanban es una palabra japonesa formada por Kan, que quiere decir visual, y Ban, que significa tarjeta.” [8]. La metodología consiste en redactar una lista de todas las tareas que se deben realizar en el proyecto y ponerlo en tarjetas de colores que se pondrán en un tablero con diferentes columnas dependiendo del estado de la actividad. En un inicio todas las actividades estarán en la columna de “Por hacer”. Los integrantes del equipo podrán tener este tablero a la vista, y cada uno se encargará de una tarjeta. Una vez seleccionada la actividad, esta tarjeta se pondrá en la columna de “Haciendo”, y cuando el integrante del equipo termine la actividad, la tarjeta, finalmente, pasará a la columna de “Hecho”.

Para la implementación de Kanban se utilizará GitHub - Projects, una herramienta que permite crear y organizar las tarjetas en un tablero y clasificarlas en las columnas de pendientes por hacer, las que se están haciendo, las que deben evaluar y las hechas, lo que permitirá mantener un registro de todas las actividades realizadas.

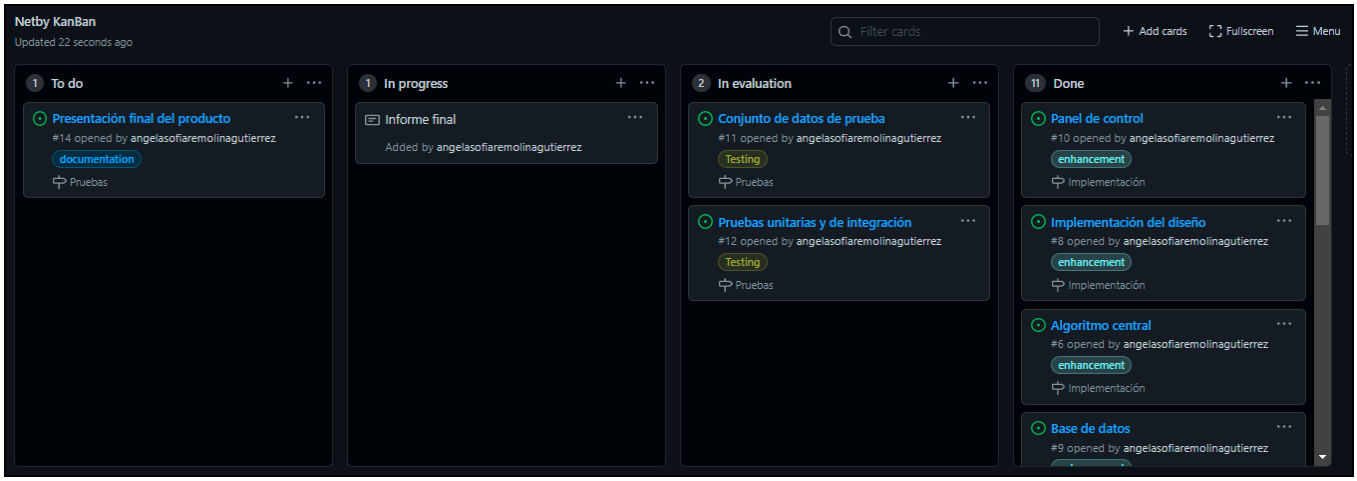

Fig.1. Tablero de tarjetas en GitHub

Para seguir este marco de trabajo (Kanban), se plantearon las siguientes etapas cada una con su lista de actividades a realizar.

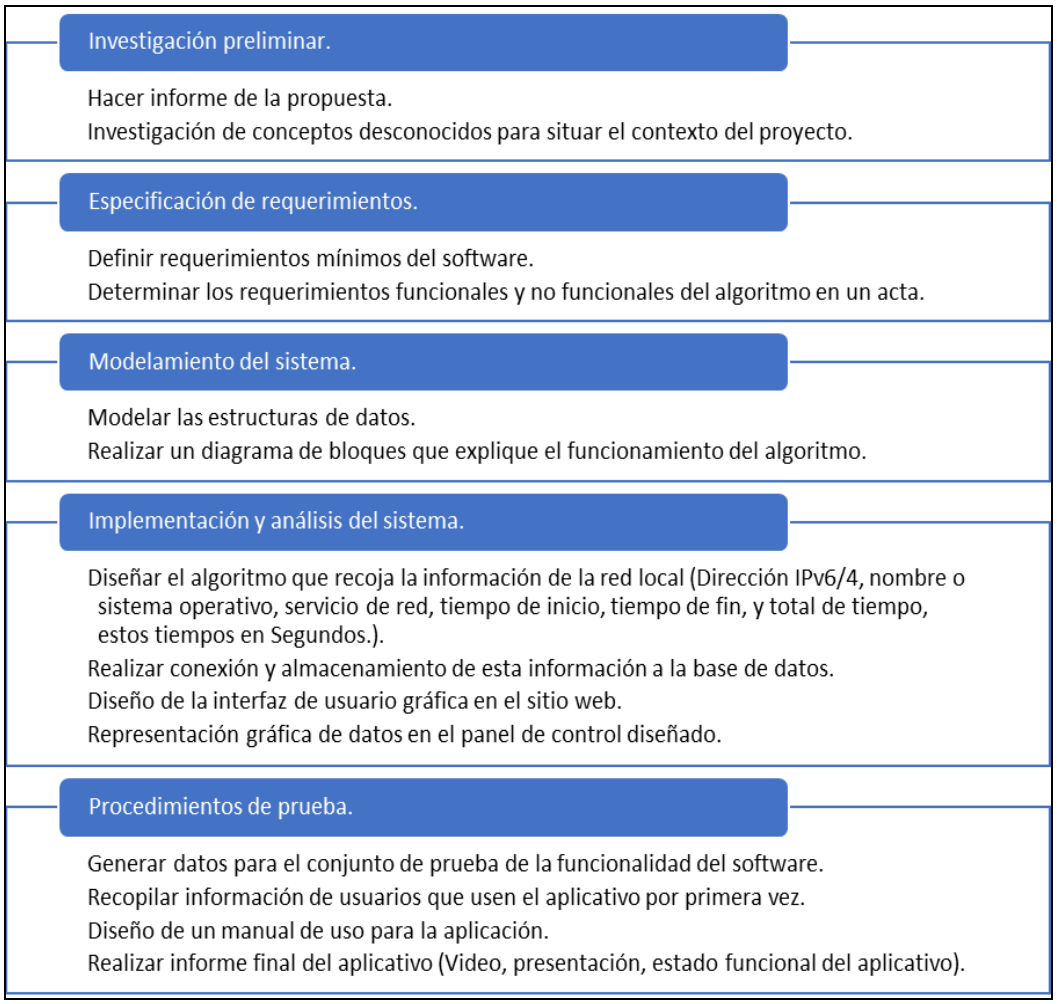

Fig.2. Diagrama de presentación de actividades por hitos para el seguimiento de la metodología.

VI. Resultados

En esta sección se presentarán y explicarán los resultados obtenidos por cada entregable planteado por objetivo.

Para el objetivo de levantamiento de requerimientos se realizó como primera parte un diagrama de casos de uso (anexo 1) el cual muestra el comportamiento que se espera que se cumpla en el aplicativo de NetBy, donde se puede evidenciar la presencia de dos actores, Usuario como actor primario y Base de Datos como actor secundario. Con el diagrama de casos de uso se consideraron necesarios 12 requerimiento funcionales (RF) y 9 requerimientos no funcionales los cuales se documentaron en el acta de requerimientos (anexo 2).

Como parte del resultado del objetivo de modelado se elaboró un diagrama entidad relación (anexo 3)

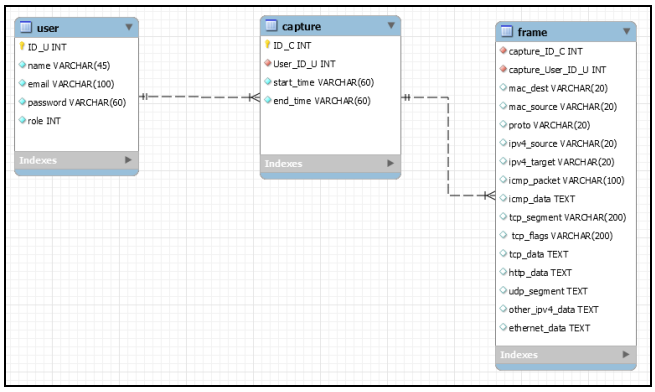

Fig.3. Diagrama entidad-relación (anexo 3)

Este diagrama presenta tres entidades que se relacionan de la siguiente manera: la entidad de *user* que tiene los siguientes campos ID que es el que identifica el usuario, un email y su respectiva password y por ultimo un atributo role que le identifica el tipo de rol que tiene el usuario en el sistema (cliente o administrador), esta entidad presenta una relación de uno a muchos con la entidad de *capture* con los siguientes atributos ID_C el cual es el número que identifica la captura, una llave foránea para identificar a que usuario pertenece con nombre USER_ID_U y por último se encuentran los campos de tiempo de inicio y tiempo de fin de la captura los cuales permiten saber el tiempo total de la captura. Y por último una entidad *frame* que la tabla donde se guarda la información de la captura, y tiene dos llaves foráneas las cuales indican a que capturan y usuario pertenecen, y adicional a esto las direcciones IPV4 y MAC, y también campos con la data de los respectivos protocolos los cuales son HTTP, HTTPS, SMTP, POP3, DNS Y DHCP con el siguiente formato proto_data, esta entidad tiene una relación de uno a mucho con *capture*. Este diagrama entidad relación tiene su correspondiente diccionario de datos (anexo 4). Éste contiene las características lógicas de los datos que se van a utilizar en el sistema de NetBy y permite una mejor visibilidad y entendimiento del diagrama Entidad-Relación.

Se desarrolló un modelo de baja fidelidad de las diferentes ventanas de la web se usó el programa "Balsamiq versión 3". Se tienen en total 10 ventanas la cuales son (anexo 5):

1. Ventana de inicio esta ventana le permite a cualquier persona escoger entre iniciar sesión y registrarse.
2. Ventana de registro esta ventana permite realizar un registro de un nuevo usuario de tipo "cliente".
3. Ventana de inicio de sesión esta ventana le permitirá a cualquier tipo de usuario iniciar sesión.
4. Ventana olvido contraseña esta ventana le permite a cualquier tipo de usuario recuperar su contraseña.
5. Ventana dashboard para "Admin" esta ventana le permite al administrador acceder a otros menús donde solo él está autorizado en entrar.
6. Ventana dashboard para "Cliente" esta venta le permite al cliente acceder a un menú limitado a sus permisos.
7. Ventana de usuarios para el "Admin" esta ventana permite visualizar todos los usuarios registrados en la web donde se observa la información personal de cada usuario y el administrador tiene la autorización de realizar ediciones y eliminar usuarios.
8. Ventana de captura para "Cliente" esta ventana permite crear capturas de red nuevas y visualizar sus capturas hechas en el pasado junto a las capturas que otros usuarios les hayan compartido.
9. Ventana de captura para "Admin" aquí el administrador solo podrá ver todas las capturas de todos los usuarios y podrá editar y eliminar.
10. Ventana de conectividad en esta ventana se puede visualizar los dispositivos conectados a la red junto a diferentes detalles de cada uno de estos dispositivos.

Se modeló un diagrama de bloques (anexo 6) para ilustrar un funcionamiento preliminar de la aplicación. De esta manera se planteó un sniffer (rastreador) en la mitad de la ruta del tráfico entre los dispositivos y el router. Este sniffer, según lo encontrado en la investigación del estado del arte, estuvo propuesto con cuatro alternativas: dos sniffers ya existentes, Ettercap o WireShark. Y otras dos que son librerías de dos lenguajes de programación diferentes, python y JavaScript con NodeJS.

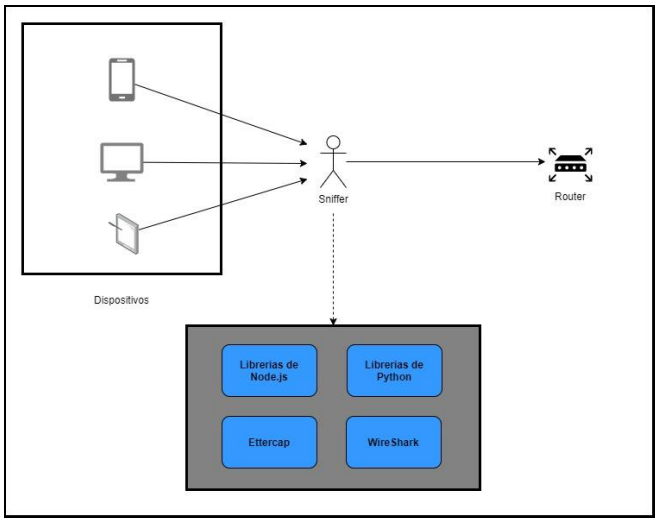

Fig.4. Diagrama de bloques preliminar (anexo 6)

Finalmente se decidió implementar el sniffer con el lenguaje de programación Python y sus librerías Socket y Struct. Los datos obtenidos con este algoritmo fueron enviados con un api restful como lo es flask y servicio principal sobre el que funciona el aplicativo web los lee con una librería de npm llamada node-fetch.

Se redactó un documento que tiene como nombre PoliticasRespaldoRecuperacionEq3 (anexo 7) el cual especifica la forma de recuperación de la información del aplicativo Netby, donde se especifica los pasos que se siguieron para el respaldo tanto del sistema operativo, base de datos, y aplicativo web. Asignando la frecuencia con el que se hacen las copias de seguridad del proyecto.

La arquitectura tecnológica del proyecto se muestra anexo en el archivo DiagramaArquitecturaEq3 (anexo8) o en la fig.4.

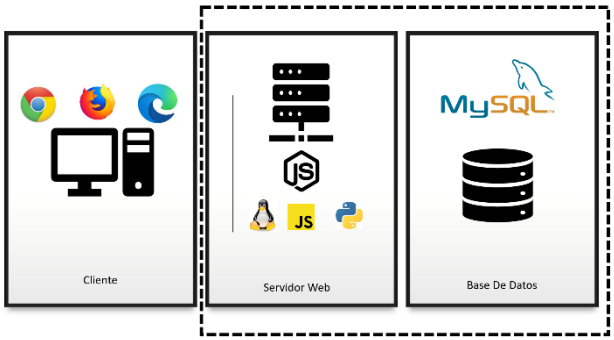

Fig.5. Diagrama de arquitectura tecnológica (anexo 8)

Esta es una arquitectura de tres niveles donde el nivel 1 es conformado por el cliente el cual es un navegador web ya sea Chrome, Edge y Firefox. En el segundo nivel tenemos el servidor web el cual está montando en una terminal de Linux Ubuntu V 18.04, con los lenguajes de programación Python, JavaScript y NodeJs, y por último el tercer nivel en el cual se encuentran nuestra Base de datos con el sistema gestor de bases de datos MySql, cabe aclarar que el Rectángulo Punteado encerrando el nivel 2 y 3 es debido a que nuestro servidor Web y la base de datos se encuentran en el mismo equipo.

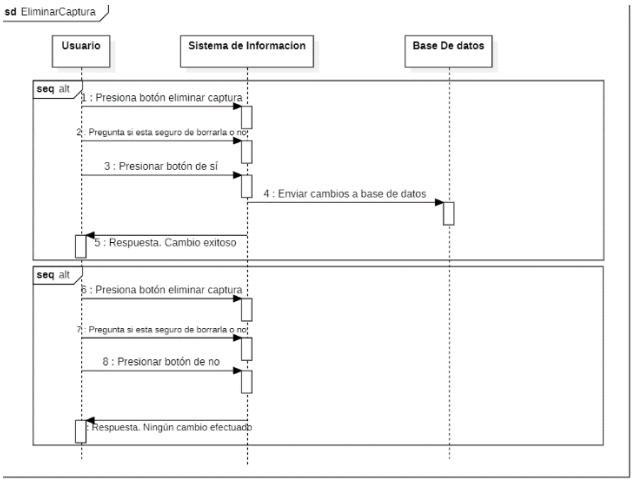

Fig.6. Diagrama de secuencia Eliminar Captura (anexo 9)

Este es un diagrama de secuencia el cual funciona para eliminar capturas, además se implementó en el sitio web y este diagrama logra explicar mejor el funcionamiento del borrado de una captura. Este diagrama tiene cada paso enumerado en orden de acción en el que el usuario deberá presionar el botón de eliminar de la captura que desea eliminar. Luego el sistema le pregunta si quiere borrar la captura, esto como seguridad para evitar equivocaciones en el proceso de borrado de la captura. Como tercer paso deberá presionar en el botón "Si", cuando se presiona el botón de "Si' este deberá eliminar la captura enviando los cambios a la base de datos y finaliza con una respuesta exitosa sobre la eliminación de la captura. Si el usuario presiono el botón de "No", no se hará ningún cambio en la base de datos y la respuesta del sistema será que ningún cambio fue efectuado.

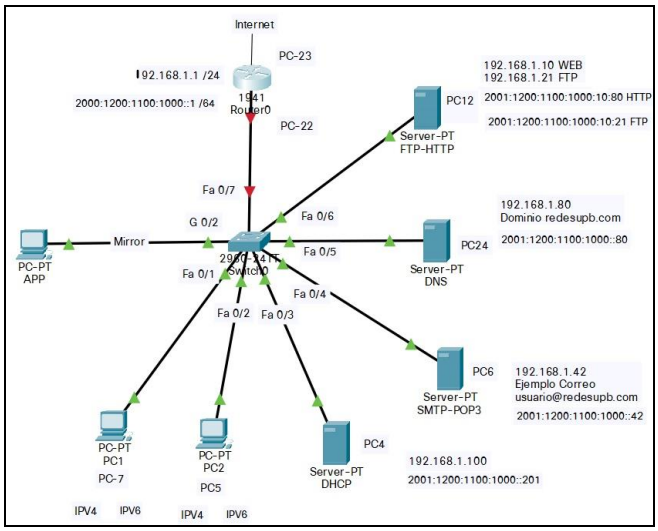

Fig.7. Modelo de la topología del entorno de pruebas (anexo 10)

En el modelo de la topología se pueden observar los 4 servidores los cuales están conectados a la red 192.168.1.1 mediante un switch que permite la conexión con 2 computadores de cliente las cuales son PC-7 y PC-5. También un puerto mirror está conectado al switch para efectuar el correcto recibimiento de los datos por parte de los servidores y de los computadores clientes, también respuestas del router el cual se encarga de proporcionar el internet a toda la red.

El servidor DHCP se encarga de asignar todas las IPv4 e IPv6 de toda la red, el servidor DNS proporciona el dominio por cual toda la red podrá comunicarse en este caso redesupb.com, SMTP-POP3 se encargan de enviar-recibir correos electrónicos, mediante sus computadores cliente PC-7 y PC-5 y además asignan las direcciones de correo electrónico, y FTP-HTTP se encargan de enviar información de este tipo mediante FTP y sitios web que se estén navegando en ese momento con HTTP. El puerto mirror se encarga de recibir toda esta información y enviarla a la aplicación previamente programada mediante un sniffer para recibir todos los datos que estén pasando sobre esta red.

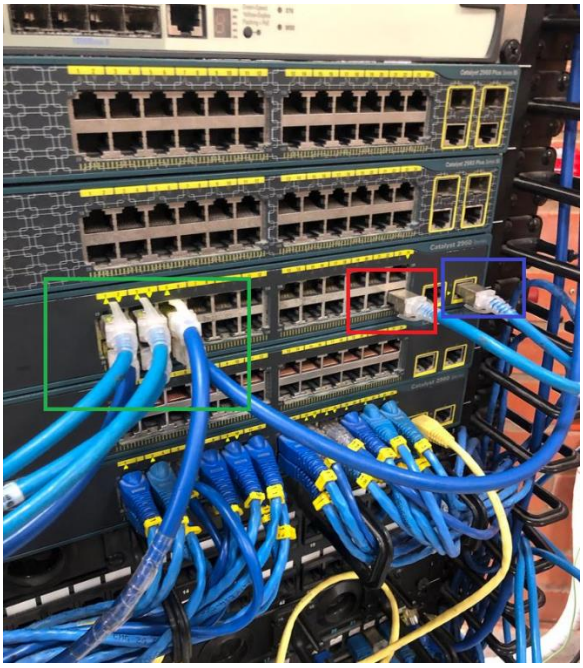

Fig.8. Topología Física (anexo 11)

La topología física fue efectuada en el aula de redes donde todo lo que se mencionó en el modelo de la topología se realizó con éxito, físicamente se plasmó lo contado y en el recuadro verde podemos ver los 4 servidores y 2 computadores cliente conectados al switch, en el recuadro rojo se observa el Router y en el recuadro azul se ve conectado el puerto mirror el cual escucha toda la red y se la envía a la aplicación que esta sobre ese mismo puerto mirror.

A continuación, se presentará la interfaz final obtenida de la aplicación web:

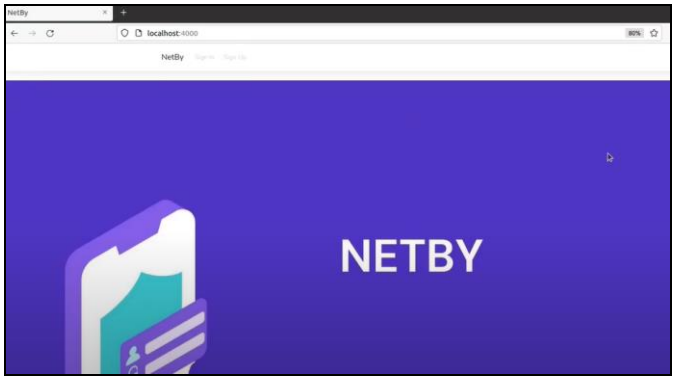

Fig.9. Pantalla de inicial NetBy

En esta pantalla (fig.9) se muesta la pantalla principal con una barra de navegación superior que tiene dos botones: *Sign up* para registrarse y *Sign in* para ingresar al sistema.

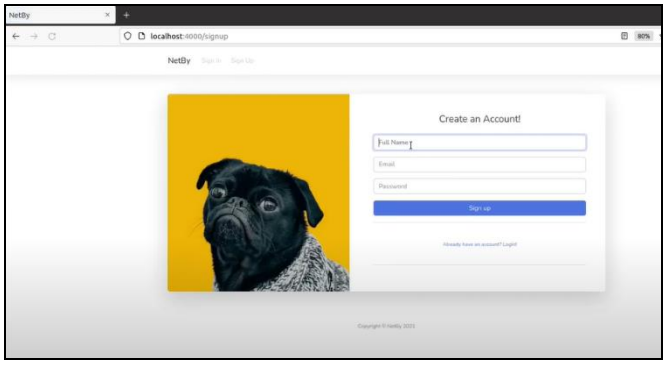

Fig.10. Pantalla de registro (Sign up) NetBy

Aquí se le presenta al usuario un formulario con 3 campos que debe llenar para crear una cuenta: nombre, correo y contraseña.

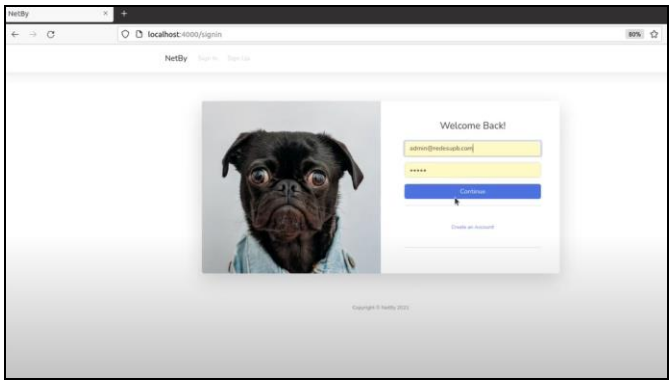

Fig.11. Pantalla de inicio de sesión (Sign in) NetBy

Aquí se le presenta al usuario otro formulario donde, una vez creada la cuenta, puede poner el correo con el que se registró y su contraseña, tras llenar esa información correctamente se muestra la siguiente pantalla.

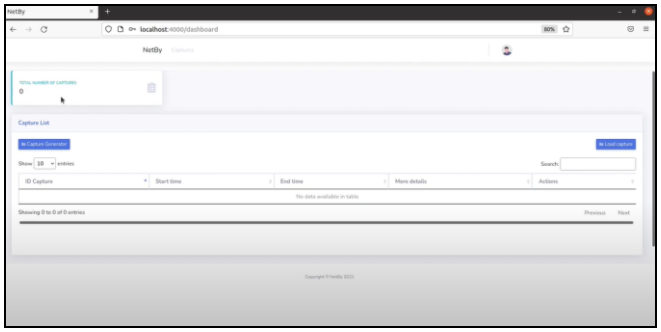

Fig.12. Dashboard vista cliente nuevo NetBy

El usuario puede observar el dashboard donde se ven sus capturas listadas en una tabla y arriba de esto dos botones, uno para generar una nueva captura y otro para cargarla a la vista. Si es un usuario nuevo no saldrá ninguna captura como se ve en la figura 12.

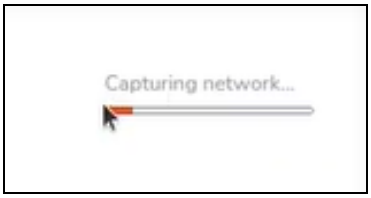

Fig.13. Barra de carga

Tras presionar el botón de generar captura, el usuario puede observar una barra de carga mientras la captura está en proceso. Una vez esta barra cargue, el usuario puede presionar el botón de cargar captura, el sitio se recargará y la captura aparecerá en la tabla (figura 14)

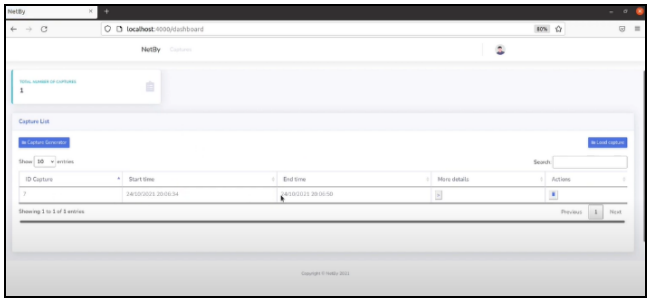

Fig.14. Dashboard vista cliente con una captura NetBy

La figura 14 muestra la pantalla una vez se cargó una captura generada anteriormente en la tabla. Si se observa con detenimiento, la barra de navegación solo presenta un botón que dice capturas y es la pantalla que se muestra por defecto al iniciar sesión, pero si se inicia sesión con un usuario administrador, la vista cambiará y la barra de navegación se mostrarán 3 opciones: capturas, usuarios y estadísticas (fig.15)

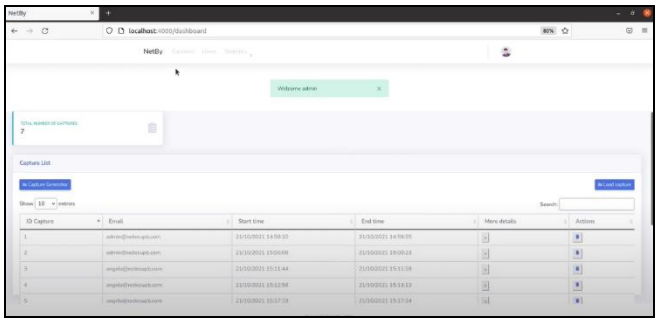

Fig.15. Dashboard vista administrador NetBy

La primera opción de captura, al igual que para los clientes muestra las capturas listadas en una tabla, pero la diferencia es que el administrador puede ver las capturas de él mismo y la de todos los demás usuarios. La captura se puede eliminar con el botón que tiene icono de basura y para ver más detalles de esta captura está el botón con icono de flecha (fig. 16).

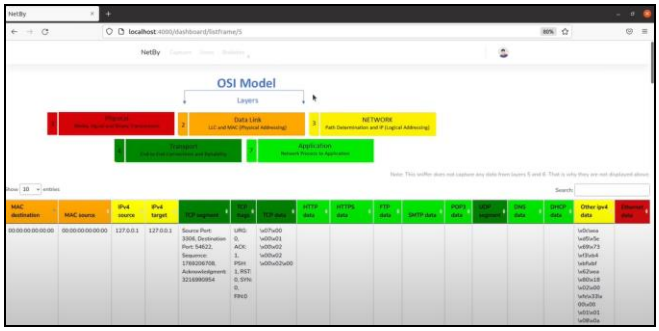

Fig.16. Vista de más detalles de la captura NetBy

Esta pantalla presenta una tabla en donde se listan todas tramas de red que fueron capturadas. Por cada trama se clasifican sus protocolos en las capas del modelo OSI con una convención de colores de la siguiente manera: rojo: capa física, naranja: capa de datos, amarillo capa de red, verde oscuro: capa de transporte y verde claro: capa de aplicación.

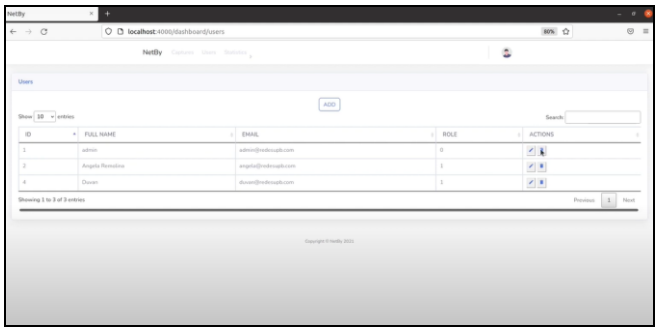

Fig.17. Vista de administrador lista usuarios NetBy

Esta pantalla presenta una tabla en donde se listan todos los usuarios registrados en la plataforma, allí el administrador puede editar un usuario para cambiar su rol, nombre de usuario y/o correo, o también puede eliminarlo.

Seleccionando la opción de estadísticas tenemos 3 gráficos, entre ellos un gráfico de barras de consumo de ancho de banda por protocolo

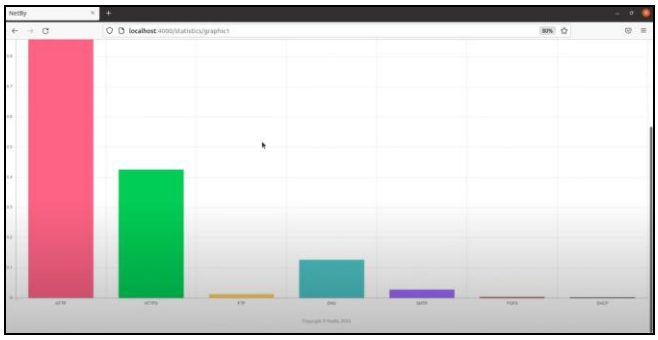

Fig.18. Gráfico de barras de consumo de ancho de banda por protocolo

Este gráfico permite ver que es lo que más consume en nuestra red doméstica. Por ejemplo, como se ve en este resultado el protocolo que más consume ancho de banda fue el HTTP.

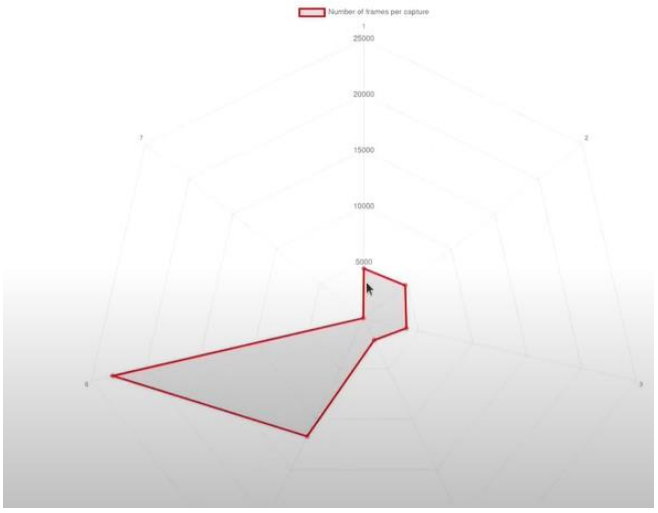

Fig.19. Gráfico de radar de tramas por captura

Este grafico permite identificar la cantidad de tramas que fueron capturadas por captura. Sirve para sacar conclusión de lo concurrido que está la red en el momento de hacer la captura.

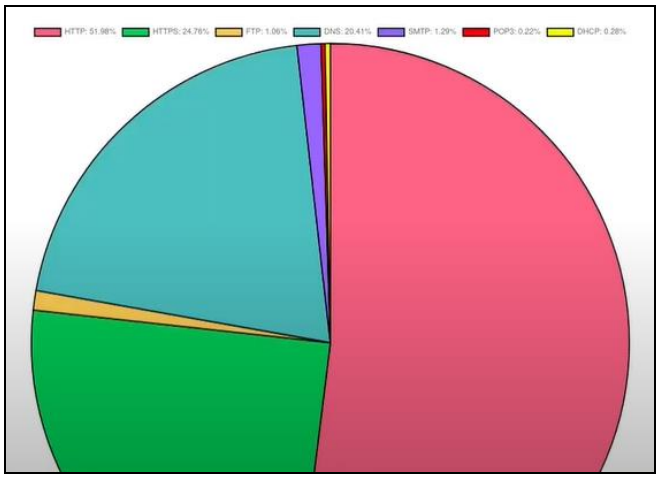

Fig.20. Gráfico de torta de tramas por protocolo

Este gráfico permite ver la cantidad de tramas de red clasificadas por cada protocolo capturado. Permite ver cuáles son los protocolos más concurridos o capturados en general por todos los usuarios.

Para ver un reporte detallado y general de todo el proyecto puede referirse al anexo 13 donde se presenta un video de presentación o al anexo 14 y 15 que es un archivo comprimido .zip y el link al repositorio con todo el código del proyecto.

VII. CUMPLIMIENTO DEL CRONOGRAMA

El proyecto como tal quedo desarrollado en un 90% siendo así un gran avance para la culminación de este proyecto, se realizaron la gran mayoría de actividades, las únicas que no se cumplieron como tal en nuestro cronograma fueron: Recoger información de usuarios que usen el aplicativo por primera vez y Diseño de un manual de uso para la aplicación. Esto pudo haber faltado por varios factores, entre esos estuvo poder lograr las capturas y lograr finalizar de manera correcta el sitio web con sus factores más importantes. Esto pudo conllevar a la falta de poder finalizar estas 2 actividades.

El cumplimiento de este cronograma permitió llevar a cabo varias actividades de manera organizada y con un buen control de seguimiento de las cosas funcionales y no funcionales.

OBJETIVOS ESPECIFICOS	ACTIVIDADES	Terminado	No terminado	ENTREGABLES
Definir los requerimientos funcionales y no funcionales del software, mediante un documento en forma de acta que muestre las limitaciones del sitio.	Hacer informe de la propuesta.	X		Propuesta y acta de requerimientos.
	Investigación de conceptos desconocidos para situar el contexto del proyecto.	X		
	Determinar los requerimientos funcionales y no funcionales del software en un acta.	X		
Modelar la estructura del sitio web mediante un diagrama de entidad-relación que muestre las bases de datos para el almacenamiento de la información de la red y otro diagrama de bloques de etapas del software para mantener un flujo claro en el desarrollo.	Modelar las estructuras de datos mediante un diagrama entidad relación.	X		Informe de avance, diagramas.
	Realizar un diagrama de bloques que explique el funcionamiento del algoritmo.	X		
Desarrollar una herramienta (back-end) que capture la información de la red para almacenar los datos obtenidos en una base de datos SQL.	Diseñar el algoritmo que recoja la información de la red local (Dirección IPv6/4, nombre o sistema operativo, servicio de red, tiempo de inicio, tiempo de fin, y total de tiempo, estos tiempos en Segundos.).	X		Software 50%
	Realizar conexión y almacenamiento de esta información a la base de datos.	X		
Implementar una interfaz gráfica (Front-end) que muestre los datos leídos de la base de datos mediante gráficas y otros elementos web.	Diseño de la interfaz de usuario gráfica en el sitio web.	X		Software 70%
	Representación gráfica de datos en el panel de control diseñado.	X		
Establecer un conjunto de casos de prueba para implementar el control de la funcionalidad del software.	Generar datos para el conjunto de prueba de la funcionalidad del software.	X		Software funcional 100%
	Recoger información de usuarios que usen el aplicativo por primera vez		X	Video de presentación, manual de uso, informe final.
	Diseño de un manual de uso para la aplicación.		X	
	Realizar informe final del aplicativo (Video, presentación, estado funcional del aplicativo).	X		

Fig.6. Cumplimiento del cronograma. (anexo 12)

VIII. Discusión

Haciendo un análisis acerca de lo planteado, lo obtenido y el funcionamiento de otras plataformas similares como las que fueron consultadas en el estado del arte, se hace una discusión con lo obtenido.

Si se compara el funcionamiento de un network sniffer como WireShark con la aplicación resultante que se obtuvo, se tiene una limitación en los protocolos capturados. NetBy captura datos principales de una trama de red por capas. Captura datos de TCP como HTTP, HTTPS, FTP, FTPS, SMTP y POP3, por parte de UDP también captura DHCP y DNS. Otros datos de la red también los obtiene, pero no los clasifica y esta es una desventaja frente a WireShark que tiene una gran variedad de filtros por sus protocolos, soportando todos los anteriores y muchos más.

También vale resaltar que la aplicación estaba planteada inicialmente para cumplir con sniffing de protocolos para plataformas de streaming como Netflix y esto no se logró para esta versión del aplicativo. Así mismo con otros requisitos planteados como: poder compartir capturas con otros usuarios, un grafo que muestre el flujo de la red desde el dispositivo que la emitió hasta el router. Hubo algunos requisitos que no se cumplieron por practicidad y replanteamiento de su utilidad en el aplicativo, como el requerimiento de editar captura, esto se descartó por parte del equipo de desarrollo al analizar que es permitir que el usuario tergiverse la información original que fue capturada en la red.

Respecto lo que fue planteado en un inicio con los modelos de baja fidelidad y lo que se obtuvo como front-end de la aplicación hubo una mejora en cuanto al diseño de experiencia de usuario. Con el uso de Bootstrap y plantillas libres, se logró cumplir con algunas de las leyes de UX.

Algunas de las leyes de UX que cumple son:

- Aesthetic-Usability Effect: Diseño estéticamente agradable.
- Jakob's Law: Es muy parecido a los sitios que los usuarios ya conocen.
- Law of Prägnanz: Se interpretarán imágenes ambiguas o complejas como la forma más simple posible.

IX. Conclusiones

En primer lugar, durante la investigación del estado del arte, se encontraron varias opciones de herramientas para el sniffing o rastreo del tráfico de red. Entre estas se encontró la librería network-sniffer de npm [15], pero esta opción fue descartada por falta de documentación en la herramienta. Es una librería poco usada y con poco soporte de la comunidad. Por otra parte, se dejaron abiertas opciones como: Ettercap, WireShark, algunas librerías de Python como Scapy, y otras de JavaScript.

Con el modelamiento del diagrama de casos de uso según cada tipo de usuario se logró definir los requerimientos y limitaciones que se seguirán como guía para el desarrollo de lo que queda del proyecto. Estas limitaciones de tipo funcional y no funcional se consignaron en un acta de requerimientos. Según esta acta planteada se logra cumplir con el 90% de los requerimientos.

Por parte del modelamiento del diagrama entidad-relación para la estructura de la base de datos, se estableció claramente las tablas existentes que almacenarán los datos del sistema y las relaciones correspondientes. Para el almacenamiento de la información del sistema (usuarios, capturas, tramas, etc), se optó por una base de datos relacional, ya que fue el tipo de base de datos que más se adaptó para el proyecto, porque al momento de extraer la información se logra mediante consultas ágiles y rápidas.

El diseño planteado de las ventanas de NetBy fue de gran utilidad para la implementación del front-end. Con este modelo de baja fidelidad, el desarrollador tuvo una base para diseñar la estructura de la vista para los clientes y para los administradores (que es diferente para cada tipo de usuario). Con base en esto se logró que la página cumpliera con algunos estándares de diseño de experiencia de usuario básicos que permiten la fácil comprensión de su uso sin necesidad de una guía extra.

Al momento de realizar las pruebas en el PC-APP (de la topología planteada) en la red se estaban, enviando correos, transfiriendo imágenes, navegando en páginas web y entre otras acciones, pero no eran capturadas por el sniffer. En las capturas solo aparecían las acciones del PC-APP. Por esto, fue necesario poner la interfaz de ethernet del puerto mirror en modo promiscuo para tener una captura de toda la red y no solo de host de la app. Con esta configuración las capturas ya tomaban la información de todos los servidores y demás clientes montados en la red.

Con base en el entorno de pruebas implementado, se evidenció que los protocolos de la capa de aplicación más concurridos en el tráfico de red fueron el HTTP y el HTTPS. Hay que tener en cuenta que el entorno de pruebas es controlado y simula tráfico de red real, pero no lo es porque algunas acciones de la cotidianidad (Netflix, youtube, etc) no se realizaron.

X. Trabajo futuro

Con base en lo que se logró obtener en el producto final y lo que fue planteado en un inicio, se tienen algunos numerales que al momento del desarrollo no fueron implementados e hicieron falta.

- Poder compartir capturas con otros usuarios.
- Implementar rutas mediante tokens para mejorar la seguridad del sitio web.
- Sistema de recuperación de la contraseña vía email.
- Implementar en la base de datos la tabla tipo de protocolos para generalizar la trama en caso de que un nuevo protocolo quiera ser agregado (como el de los servicios de streaming)

Estas recomendaciones serán tenidas en cuentas para una próxima versión del aplicativo.

XI. Agradecimientos

Especial agradecimiento a los Ingenieros Lenin Javier Serrano Gil y Elkin Alfredo Albarracín Navas por su compromiso con el proyecto integrador, de esta manera se tuvo la respectiva documentación, información y guía para llevar a cabo este proyecto. Estuvieron también activamente involucrados en la mejora en cada entrega y proporcionando diferentes soportes para culminar con este proyecto mediante documentación, pruebas de la aplicación, videos de cada funcionamiento y entre otros resultados.

XII. Anexos

Anexo 1: DiagramaCasosDeUsoEq3.mdj

Anexo 2: ActaDeRequerimientosEq3.docx

Anexo 3: EntidadRelacionEq3.png

Anexo 4: DiccionarioDatosEq3.html

Anexo 5: ModeloInterfazBajaFidelidad.pdf

Anexo 6: DiagramaDeBloques.jpg

Anexo 7: PoliticasRespaldoRecuperacionEq3.docx

Anexo 8: DiagramaArquitecturaEq3.png

Anexo 9: ModeloSecuenciaEq3.mdj

Anexo 10: Topologia.png

Anexo 11: topologiaFisica.png

Anexo 12: Cumplimiento_Cronograma.xlsx

Anexo 13: https://youtu.be/vPwZfeDlWQc

Anexo 14: codigoEq3.zip

Anexo 15: https://github.com/angelasofiaremolinagutierrez/netby

Referencias

[1] “Durante la cuarentena el tráfico de internet creció 38% en las casas”. (2020, 21 de mayo). Diario La República. https://www.larepublica.co/empresas/durante-la-cuarentena-obligatoria-el-trafico-de-internet-crecio-38-en-las-casas-3008410

[2] "What is Transmission Control Protocol (TCP)? | Security Encyclopedia". HYPR. https://www.hypr.com/transmission-control-protocol-tcp/ (accedido el 29 de julio de 2021).

[3] "User datagram protocol (UDP) - geeksforgeeks". GeeksforGeeks. https://www.geeksforgeeks.org/user-datagram-protocol-udp/ (accedido el 29 de julio de 2021).

[4] "HTTP - Concepto, para qué sirve y cómo funciona". Concepto. https://concepto.de/http/ (accedido el 29 de julio de 2021).

[5] "Qué es una dirección IP, para qué sirve y cómo funciona". Blog HostGator México. https://www.hostgator.mx/blog/que-es-una-direccion-ip/ (accedido el 29 de julio de 2021).

[6] "What is packet sniffing?" NETSCOUT. https://www.netscout.com/what-is/sniffer (accedido el 3 de septiembre de 2021).

[7] A. Froehlich, L. Rosencrance y K. Gattine. "What is the OSI model? The 7 layers of OSI explained". SearchNetworking. https://searchnetworking.techtarget.com/definition/OSI (accedido el 29 de julio de 2021).

[8] "IBM Docs". IBM – Deutschland IBM. https://www.ibm.com/docs/es/elm/6.0.3?topic=requirements-defining-use-cases (accedido el 3 de septiembre de 2021).

[9] “Software Engineering Entity-Relationship Diagram - javatpoint". www.javatpoint.com. https://www.javatpoint.com/software-engineering-entity-relationship-diagrams (accedido el 2 de agosto de 2021).

[10] "Qué es y para qué sirve SQL". Styde.net. https://styde.net/que-es-y-para-que-sirve-sql/ (accedido el 29 de julio de 2021).

[11] "Node.js". Node.js. https://nodejs.org/es/ (accedido el 29 de julio de 2021).

[12] A. Guezzaz, A. Asimi, Y. Sadqi, Y. Asimi, Z. Tbatou. "A new hybrid network sniffer model based on Pcap language and sockets (Pcapsocks)", International Journal of Advanced Computer Science and Applications, Vol. No.2, pp. 8, 2016.

[13] A. A. Adewale, V. O. Matthews, A. A. Adelakun, W. Amase, O. Alashiri, "Packet Sniffer for Users End Network Performance Monitoring using Python Programming". *International Journal of Current Trends in Engineering & Research*, 4 (4). pp. 1-11. ISSN e-ISSN 2455–1392, 2018.

[14] Apd, R. (2020, 29 mayo). ¿En qué consiste la metodología Kanban y cómo utilizarla? APD España. https://www.apd.es/metodologia-kanban/

[15] "network-sniffer". (2015). NPM. https://www.npmjs.com/package/network-sniffer.

[16] Yablonski, "Home | Laws of UX", Laws of UX, 2021. [Online]. Available: https://lawsofux.com/. [Accessed: 19- Oct- 2021