Para le Informe de avance 1: Avance 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 especifica un contexto general del proyecto, los objetivos, la metodología para llevar a cabo el desarrollo del aplicativo web NetBy y los resultados hasta la fecha actual (5/09/2021). NetBy en general es un proyecto para generar capturas por parte de los usuarios (clientes y administradores) donde se podrá evidenciar su información de red.**

Palabras claves — Red de datos, tráfico de red.

***Abstract* — This document specifies a general context of the project, the objectives, the methodology to execute along with the development of the web application NetBy and the results to date (09/05/2021). NetBy in general is a project to generate network captures by users (clients and administrators) where their network information can be seen.**

Keywords — Computer network, network traffic.

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, nos hemos visto forzados 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 nos 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.

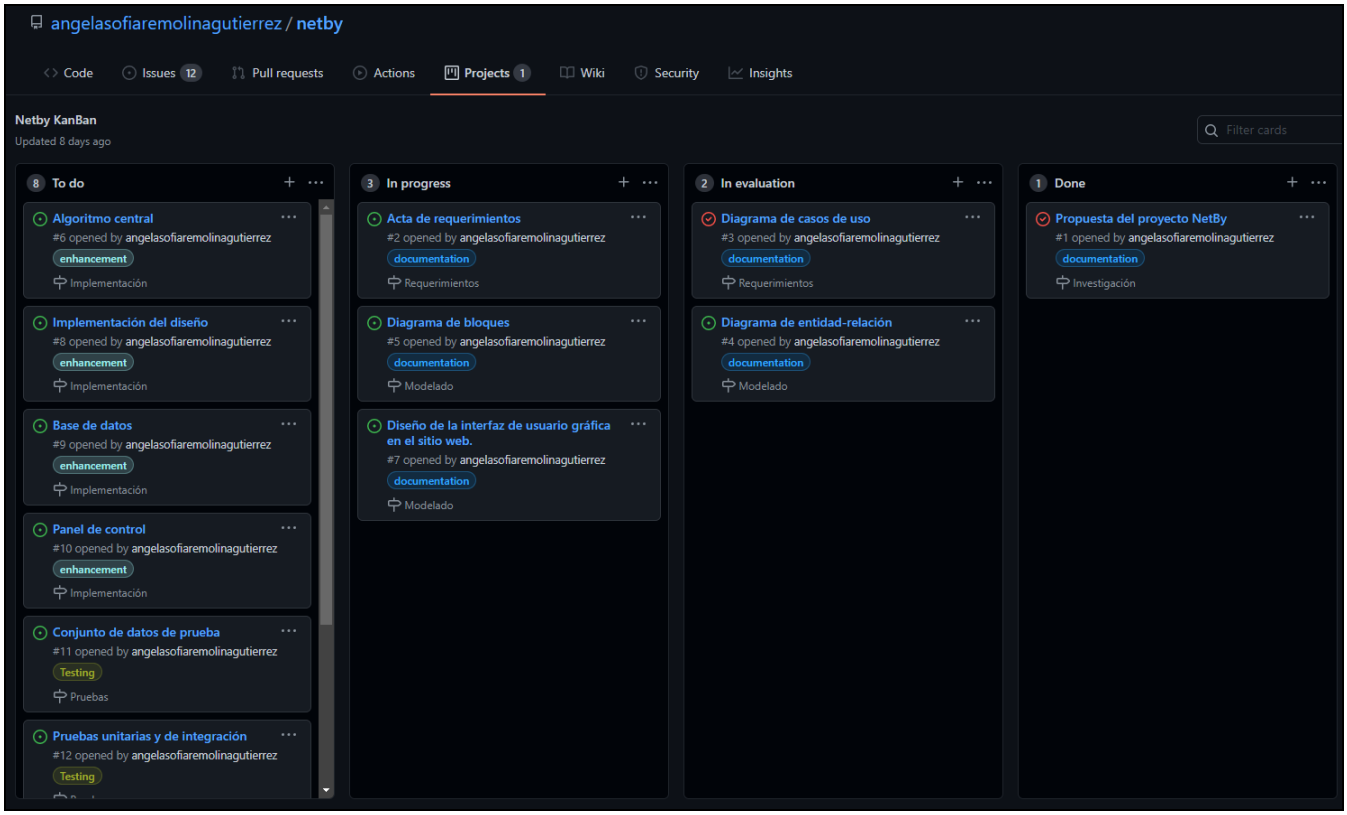

Fig.1. Tablero de tarjetas en GitHub (actualizado a la fecha de este informe de avance)

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

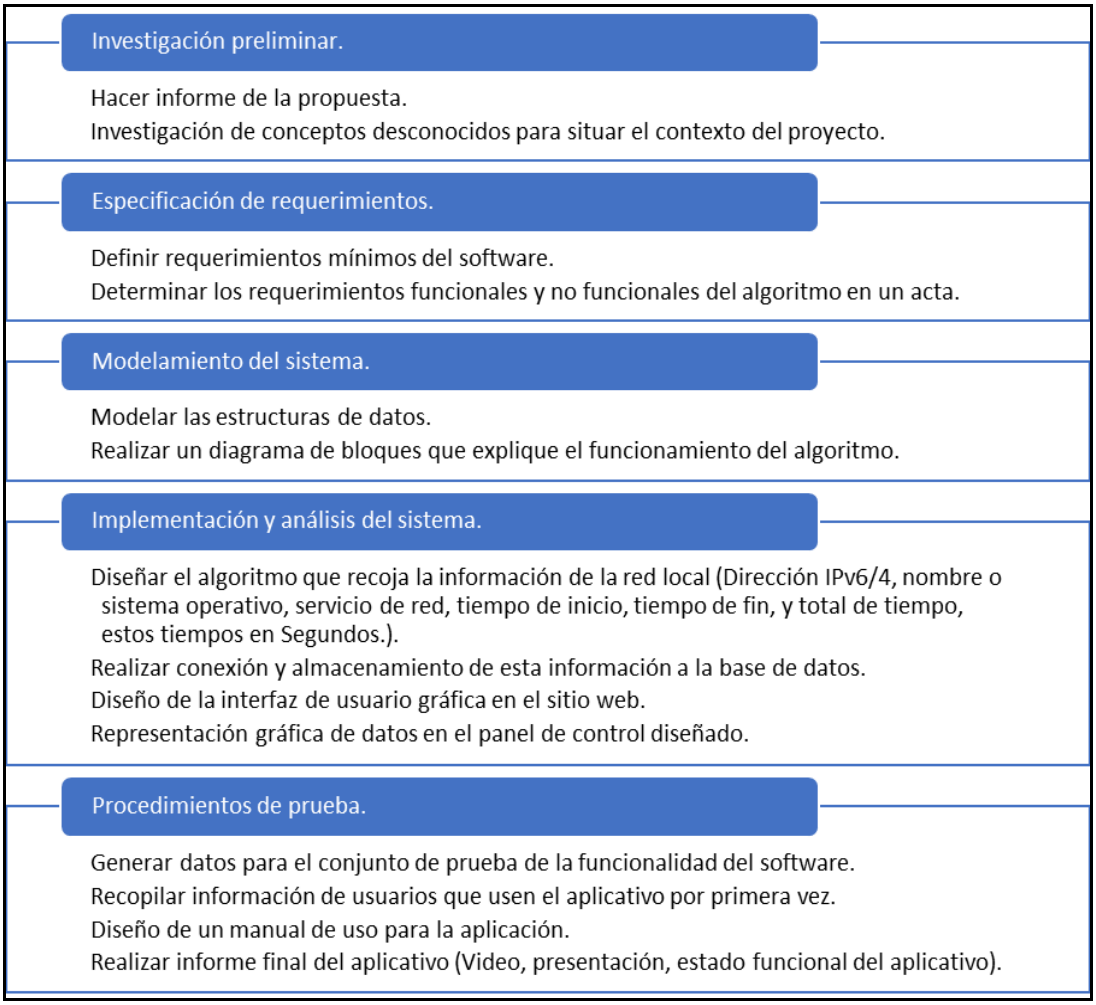

Fig.2. Diagrama de presentación de actividades por hitos para el seguimiento de la metodología.

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) el cual tiene cuatro entidades las cuales se relacionan de la siguiente manera, la entidad de Personas presenta una relacion de uno a muchos con las entidades de Captura y Tipo_Personas, la entidad de capturas presenta una relacion de muchos a muchos en cual genera una nueva entidad que tiene como nombre Captura_has_Dispositivo. Este diagrama entidad relacion tiene su correpodiente 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 el 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, está 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.

VII. Cumplimiento del cronograma

En la Fig.3 se muestra el cronograma con las diferentes actividades planteadas marcadas como terminadas o no terminadas. Como se puede observar en esta imagen, hasta el momento se han logrado completar 6 actividades de 13 llevando al proyecto a estar al 46% de completado.

Las actividades terminadas son: el informe de la propuesta, investigación de conceptos desconocidos para situar el contexto del proyecto, determinar los requerimientos funcionales y no funcionales del software en un acta, modelado de las estructuras de datos mediante un diagrama entidad relación y el diagrama de bloques que explique el funcionamiento del algoritmo. Adicional a esto, se agregó al hito de los requerimientos un modelo de casos de uso que complemente la acción de cada tipo de usuario en el sistema.

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.3. Cumplimiento del cronograma.

VIII. Discusión

IX. Conclusiones preliminares

A continuación, se presentan algunas conclusiones que han resultado del desarrollo y la investigación para el proyecto hasta la fecha de este primer informe de avance.

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.

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. Vale aclarar que los atributos o columnas que se encuentran en este modelo preliminar pueden cambiar según la herramienta de rastreo o sniffing de red que se escoja, por las limitaciones que estas puedan presentar.

El diseño planteado de las ventanas que tendrá NetBy es de gran utilidad para el inicio de la implementación del front-end. Con este modelo de baja fidelidad, el desarrollador tendrá la base más importante para empezar que es la estructura de la vista para los clientes y para los administradores (que es diferente para cada tipo de usuario).

X. Trabajo futuro

XI. Agradecimientos

XII. Anexos

Anexo 1: DiagramaCasosDeUsoEq3.mdj

Anexo 2: ActaDeRequerimientosEq3.docx

Anexo 3: EntidadRelacionEq3.png

Anexo 4: DiccionarioDatosEq3.html

Anexo 5: Modelo_interfaz_de_baja_fidelidad.pdf

Anexo 6: DiagramaDeBloques.jpg

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.