TraficAnalyzer

Herramienta para el análisis y simulación de redes de datos, ayuda en la docencia de redes

Edwin Alberto Hernández Basco

Resumen — En Linux existen diferentes utilidades para el análisis del tráfico de red y muchas carecen de interfaz gráfica, por ello, se ha desarrollado una herramienta para estas distribuciones, con una interfaz que integra funcionalidades y simulaciones que permiten el análisis de datos de redes y el refuerzo de conceptos teóricos de redes. Las funcionalidades integran algunas utilidades nativas de Linux y permiten el análisis de datos de red de forma sencilla y, las simulaciones, son para que los alumnos refuercen de forma práctica los conceptos aprendidos en clase. Para este trabajo se ha buscado información sobre las diferentes utilidades para integrarlas gráficamente y también las tecnologías a utilizar para el desarrollo de esta, después se ha definido la planificación y se han escrito los requisitos. En el desarrollo se han definido los bocetos para determinar la apariencia de la interfaz y, utilizando una metodología incremental, se realizó la implementación componente a componente, obteniendo la herramienta que se ha denominado TraficAnalyzer.

Palabras clave — GUI, Linux, Utilidades Linux, Análisis de Red, Redes, Interfaz Gráfica de Usuario, Simulación TCP, Vue, Interfaz Gráfica de Redes de Datos, ElectronJS, Network Simulation

Abstract — In Linux there are different utilities for network traffic analysis and many lack a graphical interface, for this reason, a tool has been developed for these distributions, with an interface that integrates features and simulations that allow the analysis of network data and the reinforcement of theoretical concepts of networks. The functionalities integrate some native Linux utilities and allow the analysis of network data in an easy way, and the simulations are for the students to reinforce in a practical way the concepts learned in class. For this work, information about the different utilities has been searched to be integrated graphically and also the technologies to be used for the development of this, then the planning has been defined and the requirements has been written. In the development, the sketches have been defined to determine the appearance of the interface and, using an incremental methodology, the implementation was carried out component by component, obtaining the tool which has been called TraficAnalyzer.

Index Terms — GUI, Linux, Linux Utilities, Network Analysis, Networking, Graphical User Interface, TCP Simulation, Vue, Graphical Data Network Interface, ElectronJS, Network Simulation

1 Introducción - Contexto del trabajo

Desde la creación del primer núcleo Linux, este ha ido creciendo constantemente, partiendo de tener un sistema operativo que funciona por consola y línea de comandos, hasta la creación de diferentes distribuciones con entornos gráficos que pueden ser utilizados en computadoras de escritorio y servidores como lo son Red Hat, CentOS o incluso uno de los más reconocidos por su facilidad de uso que es Ubuntu. Aún habiendo muchos avances a nivel gráfico de los diferentes programas en Linux, existen utilidades nativas potentes que carecen de interfaz gráfica y que, para muchos usuarios, puede llegar a ser de gran ayuda tenerlas para poder utilizarlas de manera fácil, además de que pueden tener como añadido ciertas ayudas como información adicional o estadísticas.

Aunque para alguna utilidad de Linux ya existen alternativas con interfaz, muchas de estas son complejas, con largas configuraciones previas en el sistema, pesadas, con problemas de compatibilidad al instalar o, una gran parte, son de pago. En la **Figura 1,** se muestra un ejemplo de cómo es la salida de tcpdump [1] en la consola de Linux, donde puede ser tedioso poder encontrar información específica a menos que se utilicen los filtros pertinentes que nos da la aplicación, esto implica saber diferentes tipos de opciones y formatos de filtros de la aplicación y que hacen que, cada vez, pueda necesitarse más conocimientos sobre el mismo programa y de todos los diferentes comandos y combinaciones que pueden aplicarse sobre este.

```
[redacted]:~ # tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
20:38:29.013544 IP text-lb.esams.wikimedia.org.http > 192.168.198.128.54543: F
 659455554:659455554(0) ack 3797162379 win 64240
20:38:29.013961 IP 192.168.198.128.54543 > text-lb.esams.wikimedia.org.http: F
1:1(0) ack 1 win 51120
20:38:29.014324 IP text-lb.esams.wikimedia.org.http > 192.168.198.128.54543: .
ack 2 win 64239
20:38:29.015350 IP 192.168.198.128.40566 > 192.168.198.2.domain: 16599+ PTR? 1
8.198.168.192.in-addr.arpa. (46)
20:38:29.043034 IP 192.168.198.2.domain > 192.168.198.128.40566: 16599 NXDomai
 0/1/0 (95)
20:38:29.043533 IP 192.168.198.128.57315 > 192.168.198.2.domain: 34584+ PTR? 1
2.174.198.91.in-addr.arpa. (45)
20:38:29.101802 IP 192.168.198.2.domain > 192.168.198.128.57315: 34584 1/0/0 P
R[|domain]
20:38:29.102332 IP 192.168.198.128.57083 > 192.168.198.2.domain: 41740+ PTR? 2
198.168.192.in-addr.arpa. (44)
20:38:29.128455 IP 192.168.198.2.domain > 192.168.198.128.57083: 41740 NXDomai
 0/1/0 (93)
20:38:29.177013 IP bits-lb.esams.wikimedia.org.http > 192.168.198.128.53928: F
 929274665:929274665(0) ack 3446487441 win 64240
20:38:29.177279 IP 192.168.198.128.53928 > bits-lb.esams.wikimedia.org.http: F
1:1(0) ack 1 win 58220
20:38:29.177618 IP bits-lb.esams.wikimedia.org.http > 192.168.198.128.53928: .
ack 2 win 64239
20:38:29.178046 IP 192.168.198.128.33843 > 192.168.198.2.domain: 3084+ PTR? 20
```

Figura 1. Salida de tcpdump en consola.

Por otro lado, existen diferentes simuladores de redes que pueden utilizarse para preparar diferentes entornos virtuales con ordenadores y redes interconectadas como lo es Cisco Packet Tracer [2] o Graphical Network Simulator (GNS3) [3], pero estos y la gran mayoría son simulaciones a nivel topológico y no a nivel detallado de protocolos de redes, de este último existen muy pocos, por no decir que son casi inexistentes para que los estudiantes puedan aplicar y reforzar la teoría aprendida en las asignaturas de redes, algunas de las pocas simulaciones que se pueden encontrar son TCP Congestion Control [4], Flow Control [5] o TCP Sliding Window [6], además son simulaciones puntuales de conceptos de protocolos y algunas no son tan fáciles de entender debido a que, la simulación, no muestra gráficamente el procedimiento de la manera en que se hace en las clases de problemas de redes como se aprecia en la **Figura 2**.

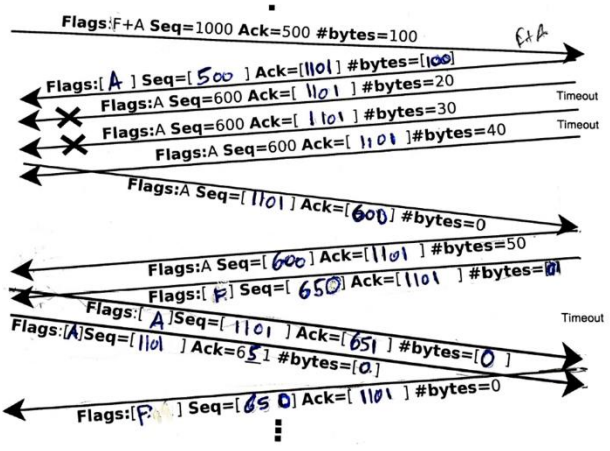


Figura 2. Problema de envío y recepción de segementos TCP en clase de problemas de redes.

La motivación de este trabajo es el de implementar una herramienta con tecnologías modernas para el análisis y simulación de redes de datos que ayude a los estudiantes a comprender y reforzar el conocimiento de diferentes conceptos impartidos en las asignaturas de redes. La herramienta ha de permitir la escalabilidad de esta, pudiendo añadir o mejorar cada vez más los diferentes componentes que tiene de manera fácil y, aprovechando otros componentes ya creados o, incluso, utilizando componentes existentes en Internet y diseñados con la misma tecnología utilizada, pudiendo obtener una herramienta completa de monitorización y simulación de redes.

El trabajo está estructurado de la siguiente manera, en la sección 2 el estado del arte donde se expone el desarrollo investigativo del trabajo, en la sección 3 los objetivos planteados del trabajo, la sección 4 la metodología y planificación realizada en el desarrollo del trabajo, en la sección 5 el desarrollo y resultados de la herramienta, en la sección 6 los problemas surgidos, la sección 7 planificación futura del trabajo y la sección 8 las conclusiones del trabajo, finalmente se encuentra la sección de agradecimientos, referencias bibliográficas y el apéndice.

2 Estado del arte

Se ha buscado por Internet soluciones de software parecidas al enfoque de la herramienta a desarrollar a modo de orientación previa, existen diversos programas para el tráfico de red como lo es Wireshark [7], es muy conocido y probablemente muy utilizado por su completa funcionalidad para ver todo el tráfico de la red, pero existen algunas situaciones como por ejemplo, para poder analizar pings, debemos abrir un terminal para realizarlo y Wireshark se encargará de capturar el tráfico, estas acciones son las que interesa en este trabajo para poder integrar, desde un mismo lugar, las diferentes utilidades de análisis de tráfico de red y utilizarlas de manera fácil con la interfaz y desde la misma herramienta, sin necesidad de tener diversas ventanas abiertas, aportando facilidad visual de la información y funciones como lo pueden ser estadísticas de esta información.

Algunos otros softwares como tcpdump [1] vienen integrados en muchas distribuciones de Linux y es una herramienta más simple en funcionalidades que Wireshark, pero igual de potente para analizar el tráfico, esta utilidad a proporcionado el interés de implementar una interfaz gráfica para el uso de las diferentes utilidades que no la tienen porque funcionan directamente por línea de comandos y, además, muchas de estas utilidades suelen ser usadas en asignaturas de redes y de seguridad de los sistemas e información para analizar datos de redes. En la **Figura 3** podemos ver la interfaz de Wireshark donde se aprecia que a nivel de interfaz es mucho mejor que la de tcpdump en la **Figura 1.**

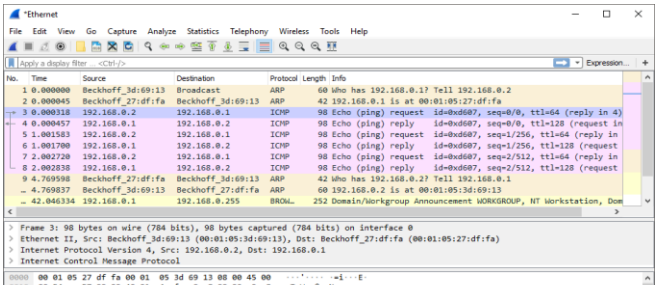


Figura 3. Interfaz de salida de Wireshark.

Existen programas open-source o de pago diseñado específicamente para monitorización de redes en el ámbito de grandes o pequeñas empresas, integradas a nivel de helpdesk o aplicación web, siendo la última la que va en aumento por ofrecer el servicio a nivel web que permite gestionar más fácil la aplicación y sin instalaciones locales para cada usuario, algunas de las herramientas son Solarwinds [8] o Nagios XI [9], estas han servido de base para saber cómo se puede implementar la interfaz gráfica de las diferentes funcionalidades que hay en el programa, en la **Figura 4** se aprecia de manera general algunas de las soluciones que integran estas herramientas.

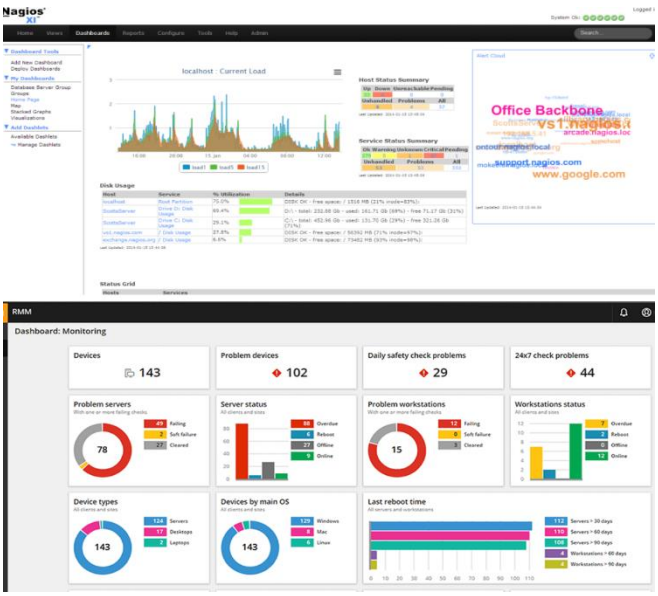


Figura 4. Interfaz de monitorización de Nagios (Imagen de arriba) y So-larwinds (Imagen de abajo).

La existencia de programas nativos de análisis de tráfico de red como lo es Ping, ARP, nslookup, entre otros son algunos de los cuales han sido seleccionados para ser integrados en la herramienta con el objetivo de crear esa interfaz gráfica que aporta además funcionalidades adicionales para el uso de estas.

Para la herramienta se ha utilizado diversas tecnologías, unas de las principales es VueJS [10] que es una herramienta Single Page Application (SPA) y open-source que cada vez está siendo más utilizada en grandes empresas, esta tecnología que lleva desde el 2014 abarcando el mercado junto a ReactJS y Angular tienen una amplia documentación y foros de ayuda para poder resolver cualquier problema. Otra de las ventajas que incluye VueJS es que su funcionalidad es más sencilla que otros frameworks y se basa en el uso de HTML, CSS y JavaScript, además de su propia estructura de código.

Para el diseño de la interfaz gráfica se ha utilizado el framework Vuetify [11] para dar un aspecto moderno y simple a la herramienta, este lleva muchos años en el mercado, se puede encontrar mucha documentación e información en foros a la hora de desarrollar front-end, además es una librería que está en constante desarrollo por lo que añadir o modificar la interfaz en un futuro es viable sin necesidad de tener que comenzar desde cero.

3 Objetivos

El objetivo principal del trabajo es desarrollar una herramienta para distribuciones Linux con interfaz gráfica que implementa funcionalidades y simulaciones de redes, las funcionalidades integran algunas de las utilidades de redes de Linux para usarlas fácilmente y, las simulaciones, ayudan a los estudiantes a comprender y reforzar diferentes conceptos impartidos en las asignaturas de redes, permitiendo sintetizar la docencia de redes a partir de estas y la información dada en las funcionalidades sobre estos conceptos. Para alcanzar este objetivo, se ha de cumplir una serie de subobjetivos que son necesarios:

- Implementación de las funcionalidades que permiten integrar las utilidades de Linux y, además, que contengan funciones extras como estadísticas o visualización de la información pertinente
- Implementación de las simulaciones que permitan la interacción con estas, permitiendo modificar campos para generar diferentes simulaciones y ver cómo afectan a esta según los valores asignados a estos campos
- Se debe poder compilar y distribuir la aplicación para dar utilidad y viabilidad a esta
- La aplicación debe ser intuitiva y fácil, implementar los comandos utilizados de manera transparente para que el usuario no tenga la necesidad de saber en detalle estos al usar la interfaz gráfica.
- La aplicación debe permitir escalabilidad

4 Metodologia y planificación

En este apartado se explica la metodología utilizada para el desarrollo de la herramienta y la planificación, también los retrasos surgidos y la replanificación realizada por los retrasos.

4.1 Estrategia de desarrollo

Existen diversas metodologías de desarrollo Software y Web, de las cuales finalmente se ha utilizado la metodología incremental [12], el motivo de utilizar esta metodología ha sido porque, al planificar de una manera secuencial los diferentes componentes o funcionalidades de la herramienta, se ha podido enfocar de manera individual la realización de cada uno de estos, seguidamente se ha testeado y se daba por finalizado obteniendo un componente completamente funcional y así pasar al siguiente. También permitía asegurar de que, si iba mal, solamente afectaría a esta última iteración hecha y se podría volver a la versión anterior al punto antes del problema surgido. En la **Figura 5** Se aprecia el ciclo utilizado en esta metodología incremental.

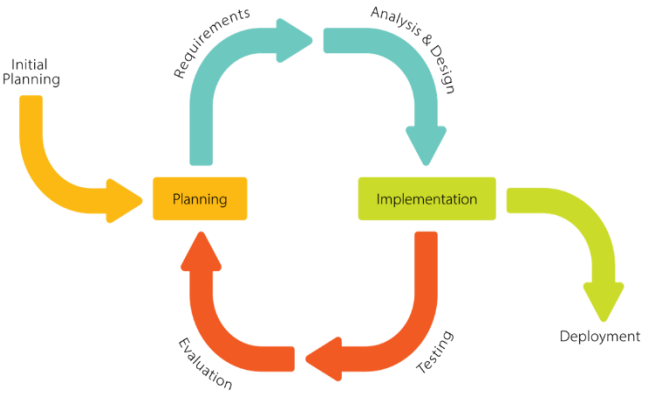


Figura 5. Ciclo de vida de metodologia incremental.

Otro de los motivos para usar esta metodología es que ha sido posible entregar un componente completamente funcional e independiente que se pudiera utilizar desde su primera versión y, de esta manera, cada vez que se iba implementando otra, se iban cumpliendo con las tareas previamente establecidas en el trabajo y la definición de requisitos hasta que finalmente alcanzara el producto final. Esta independencia también hace posible añadir nuevas funcionalidades a cada componente o, mejorar cada una de estos sin que se vean afectados los demás.

4.2 Tecnologías utilizadas

Las tecnologías que se han empleado han sido elegidas teniendo en cuenta diversos factores, en cuanto a librerías y programas, han sido seleccionados basados en utilizar aquellos que pueden asegurar un funcionamiento a largo plazo, es decir, o bien son librerías o programas nativos, o bien son externos, pero que llevan muchos años funcionando y están en constante desarrollo y actualización, por lo que de esta manera se asegura poder tener en funcionamiento los diferentes componentes de la herramienta durante mucho tiempo, preocupándose en principio por modificaciones puntuales que a lo largo de los años puedan surgir por las propias tecnologías como actualizaciones, reestructuración de la sintaxis al codificar o, mejoras de los componentes y no porque las herramientas desaparezcan del mercado.

4.2.1 Utilidades básicas a usar

los programa utilizados para la implementación de las funcionalidades son principalmente las utilidades nativas que podemos encontrar en distribuciones Linux, estas tecnologías son las siguientes:

- **Tcpdump:** Permite analizar el tráfico que hay en la red (captura de paquetes), integra diferentes opciones y filtrado del tráfico por lo que es flexible para poder restringir lo que queremos observar en la red, pero, es necesario saber bien las diferentes opciones de filtrado que se utilizan junto a los comandos.

- **Ping:** Utilizado para ver el estado de comunicación entre computadoras a partir del uso de paquetes Echo Request y Echo Reply.

- **Traceroute:** Utilizado para trazar una ruta de comunicación entre origen y destino a partir de la disminución del Time To Live (TTL) y ayuda a la detección y diagnóstico de posibles problemas en la ruta utilizada.

- **Netstat:** Utilizado para conocer detalles de los sockets activos de entrada y salida como puerto, protocolo utilizado, estado, etc.

- **Nslookup:** Se utiliza para mostrar las IP's y el nombre del dispositivo asociado a cada una de ellas que se encuentra en caché.

- **ARP:** Protocolo de Resolución de Nombres que utiliza un paquete ARP Request para saber la dirección física destino que se envia mediante la respuesta de un paquete ARP Reply.

Algunos de los frameworks y API's externas utilizadas son:

- **LeafletJS [13]:** Librería open-source que nos permite integrar mapas de manera simple y rápida.

- **IPAPI [14]:** Nos permite encontrar las coordenadas de una IP, así como también más información de manera gratis, hasta 10000 consultas al mes, utilizado para integrarlo en una traza de rutas y ver las diferentes ubicaciones por donde pasa un paquete.

- **ApexCharts [15]:** Librería NPM para la utilización de gráficos estadísticos de diversos tipos.

- **CORS [16]:** Es una librería NPM que nos permite hacer peticiones de origen cruzado, esto es debido a que utilizamos el Back-End que estará funcionando a nivel local, pero en un puerto diferente, permitiendo así acceder a recursos de un servidor solicitándolo desde un origen distinto al que pertenece.

Existe la posibilidad que algunas distribuciones de Linux no tengan previamente instalado alguna de las utilidades, por lo que se proveerá una guía de instalación referente a estas, así como también el script necesario para instalarlas.

4.2.2 Tecnología para el Back-End

El Back-End es una de las partes más importantes y críticas en este trabajo porque el Front-End cumple como interfaz gráfica para las funcionalidades, pero todo el trabajo lo realiza el Back-End que hace llamadas al sistema, para emplear las utilidades proporcionadas por Linux, no se puede hacer directamente desde el Front-End, esto es debido a que la tecnología utilizada en la interfaz gráfica es todo a nivel de navegador web, no hay un enlace directo entre la máquina y el navegador para permitir ejecutar diferentes acciones en el sistema desde el navegador directamente y, también, porque sería un riesgo de seguridad para cualquier página web que permita introducir archivos o código malicioso que se puede ejecutar en la máquina.

Por este motivo ha sido importante tener un Back-End que al final es una API que recibe las peticiones HTTP realizadas desde el navegador, procesa la petición y realiza las acciones pertinentes para ejecutar las utilidades y procesar los datos para devolverlos en una respuesta al usuario.

En este caso se ha decidido utilizar NodeJS [17] para el lado del servidor, esto es fundamental para el trabajo, ya que la idea es poder emplear diferentes operaciones sin tener que esperar a finalizar otra. Además, NodeJS proporciona un módulo llamado child_process [18] que nos

permite utilizar varios procesos y no uno único, de esta manera se accede y ejecuta las utilidades a partir de llamadas al sistema que realiza los comandos pertinentes y ejecuta todo en procesos secundarios para posteriormente devolver el flujo de datos de salida hacía la interfaz. Otro de los motivos por el cual se utiliza NodeJS es debido a la facilidad en que podemos programar y nos permite una gran flexibilidad a la hora de crear cada módulo correspondiente a las funcionalidades, en este caso llegando incluso a definir operaciones de manera más simple y sin tantas líneas de código a diferencia de otros lenguajes, esto se ve reflejado también en el uso del módulo de child_process, ya que con unas pocas líneas puedes gestionar todo el flujo de datos creado al ejecutar un comando en el sistema y haciendo mucho más fácil el mantenimiento del código.

4.2.3 Tecnología para el Front-End

El Front-End se utiliza para la interacción del usuario con las diferentes funcionalidades y simulaciones, para el caso de las funcionalidades la comunicación se realiza enviando solicitudes HTTP al Back-End para recuperar los datos de interés solicitados, pudiendo procesar y mostrar en el navegador los diferentes datos de una manera más organizada y que se entienda de manera fácil por el usuario.

El Front-End se ha diseñado con la tecnología de VueJS, permitiendo dividir cada utilidad en módulos o componentes, diferenciando así cada uno de estos y que las modificaciones hechas eviten modificar otros componentes, también nos sirve para poder exportarlos e importarlos en futuros proyectos y, otra de las características importantes, es que utilizamos tecnología ya conocida como HTML y CSS, además de las propias funcionalidades añadidas de VueJS.

Por otro lado, respecto al diseño de la interfaz, se ha utilizado la tecnología de Vuetify [11], que integra Material Design disponible para su utilización, aportando una interfaz clara y bien definida, de esta manera se ha podido enfocar el desarrollo sobre todo en la funcionalidad de cada uno de los componentes y simulación.

4.2.4 Distribución de la herramienta

Para hacer viable la herramienta, se ha utilizado ElectronJS [20] que es un framework para el desarrollo de aplicaciones gráficas de escritorio y que permite integrar tanto aplicaciones a nivel de servidor como cliente o bien, ambas a la vez.

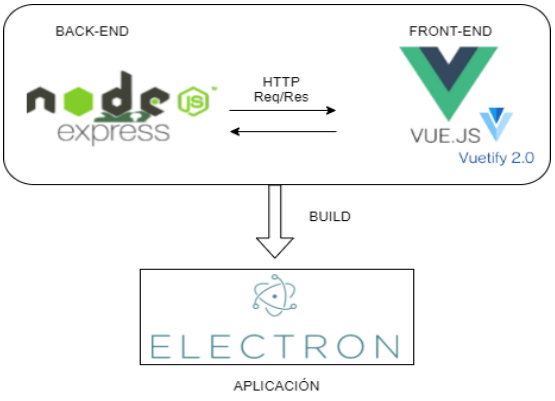


Figura 6. Estructura de la implementación de la aplicación.

En la **Figura 6** podemos ver la estructura de la aplicación, donde tenemos la comunicación entre el Back-End y Front-End que se comunican a través de peticiones HTTP, toda esta estructura se compila y construye en una aplicación con Electron, en este caso se integra tanto el Back-End como el Front-End en la misma aplicación para que pueda ser fácil poder distribuirla y también de usar para los usuarios.

4.3 Planificación

En este apartado se explica la planificación llevada a cabo durante la fase de desarrollo y los cambios que ha habido referente a la planificación inicial.

4.3.1 Planificación del proyecto

La planificación final del proyecto se puede apreciar en la **Figura 7**, ha sido seguida en el orden indicado en esta y teniendo en cuenta algunos cambios que se han hecho por retrasos en el desarrollo del proyecto surgidos.

Nombre de tarea	Comienzo	Fin	% completado	Estado
TraficAnalyzer: Herramienta para el análisis y simulación de redes de datos	**mar 23/02/21**	**sáb 03/07/21**	**80%**	**Retrasada**
Planificación	**mar 23/02/21**	**vie 12/03/21**	**100%**	**Completada**
Informe Inicial	**mar 23/02/21**	**vie 12/03/21**	**100%**	**Completada**
Consultar herramientas a utilizar	mar 23/02/21	vie 26/02/21	100%	Completada
Crear esqueleto del informe	lun 01/03/21	mar 02/03/21	100%	Completada
Redactar informe inicial	**mié 03/03/21**	**vie 12/03/21**	**100%**	**Completada**
Fuentes de información	mié 03/03/21	vie 05/03/21	100%	Completada
Redactar informe inicial (Portada, Introducción, métodología bibliografía)	lun 08/03/21	vie 12/03/21	100%	Completada
Fase de desarrollo	**lun 15/03/21**	**mié 23/06/21**	**84%**	**Retrasada**
Documentación base y especificación de requisitos	lun 15/03/21	mié 17/03/21	100%	Completada
Esqueleto de interfaz gráfica principal	lun 15/03/21	vie 19/03/21	100%	Completada
Funcionalidades	**sáb 20/03/21**	**jue 15/04/21**	**100%**	**Completada**
Comprobar Estado de una máquina	sáb 20/03/21	mar 23/03/21	100%	Completada
Sockets Activos	mié 24/03/21	vie 26/03/21	100%	Completada
Ruta de paquetes	vie 26/03/21	vie 02/04/21	100%	Completada
Capturar paquetes	lun 05/04/21	jue 15/04/21	100%	Completada
Test y correción de funcionalidades	vie 16/04/21	lun 19/04/21	100%	Completada
Informe de progreso I	lun 19/04/21	sáb 24/04/21	100%	Completada
Simulaciones	**lun 26/04/21**	**sáb 22/05/21**	**55%**	**Retrasada**
Conceptos TCP	lun 26/04/21	dom 16/05/21	100%	Completada
The Mother of All Problems	vie 07/05/21	sáb 22/05/21	0%	Retrasada
Test y correción de simulaciones	mié 19/05/21	lun 24/05/21	100%	Completada
Informe de progreso II	lun 24/05/21	sáb 29/05/21	100%	Completada
Cambios de desarrollo (Funcionalidades y simulaciones)	lun 24/05/21	jue 10/06/21	100%	Completada
Test, correción y finalización de desarrollo	vie 11/06/21	dom 13/06/21	80%	Retrasada
Fase de finalización	**lun 14/06/21**	**sáb 26/06/21**	**100%**	**Completada**
Finalizar documentación	lun 14/06/21	sáb 19/06/21	100%	Completada
Propuesta de informe final	lun 14/06/21	sáb 19/06/21	100%	Completada
Propuesta de presentación	lun 21/06/21	sáb 26/06/21	100%	Completada

Figura 7. Planificación del desarrollo del proyecto.

En el primer bloque de desarrollo se ha podido lograr el objetivo que era tener las funcionalidades desarrolladas y funcionales hasta la entrega del informe de progreso I a la fecha indicada. Por otro lado, las simulaciones no se han podido completar en su totalidad, solamente se ha podido implementar la simulación TCP, esto es debido a que hubo un imprevisto externo que ha hecho retrasar los días de desarrollo y ha imposibilitado poder implementar la simulación de *The Mother Of All Problems*, por lo que el desarrollo del proyecto sigue abierto, en el apéndice se adjunta el **Diagrama de Gannt**.

4.3.2 Replanificación y problemas surgidos

La replanificación se ha llevado a cabo en base a los retrasos surgidos durante la fase de desarrollo. A continuación, se indica una lista de los problemas que han influido en la replanificación:

- **Esqueleto de la interfaz gráfica principal:** Dos días más debido a la adaptación de la tecnología y

consultas referentes al desarrollo

- **Ruta de paquetes:** Cuatro días más por la integración del mapa en la tecnología usada del Front-End.
- **Capturar paquetes:** Una semana más debido a la búsqueda de recursos que permitieran procesar en tiempo real los paquetes.
- **Estadísticas**: Dado los retrasos anteriores y otras tareas no referentes a este trabajo, se ha suprimido la funcionalidad de estadísticas, aunque existen estadísticas en algunas funcionalidades. Esta funcionalidad era específica para mostrar el estado de diversos recursos de red de la máquina.
- **The Mother Of All Problems:** La simulación no ha podido ser implementada debido a causas de fuerza mayor externas que han ocupado gran parte del tiempo

Cabe destacar que la planificación inicial era correcta, teniendo en cuenta una estimación de desarrollo para cada tarea y teniendo en cuenta su retraso, pero, la gran afectación, fue en el segundo bloque de desarrollo que fueron las simulaciones ya que las circunstancias por fuerza mayor han ocupado casi la totalidad del tiempo para este bloque.

4.3.3 Estado actual del proyecto

El proyecto actualmente se encuentra en desarrollo, específicamente la fase de desarrollo final, en base a los requisitos redactados al principio y la replanificación realizada, se define una implementación lograda del 80% de la herramienta teniendo en cuenta el nivel de cumplimiento de los requisitos y, que queda faltando la implementación de la simulación de *The Mother Of All Problems*. El porcentaje de cumplimiento se aprecia en la **Figura 7.**

5 Desarrollo y Resultados

En este apartado se explica el diseño e implementación de la herramienta, comenzando por explicar la estructura general de la herramienta, seguido de una explicación breve de la funcionalidad de cada componente y la vista de la interfaz gráfica del mismo.

5.1 Componentes a desarrollar

TraficAnalyzer se compone de funcionalidades y simulaciones, las funcionalidades son herramientas que permiten tener al alcance diferentes utilidades de análisis de tráfico de red para que el usuario pueda utilizarlos en cualquier momento, de manera rápida y fácil cuando lo necesite, por otro lado, las simulaciones, son un elemento práctico que tiene como objetivo ayudar a reforzar conceptos teóricos visto en clases de redes, pudiendo interactuar en la simulación y poder reforzar estos conocimientos adquiridos. De esta manera se integra todo en la misma herramienta, además de lograr una abstracción sobre las utilidades que se usan internamente y que los usuarios no tengan por qué saber los comandos de estos.

5.1.1 Funcionalidades

Se implementan las siguientes funcionalidades con el objetivo de que puedan ser accesibles de manera rápida y, utilizando formularios para aportar los datos necesarios para ejecutar el programa sin necesidad de saber comandos ni parámetros de las utilidades. La implementación contiene las opciones que se han considerado más básicas y/o utilizadas para no convertirlo en un programa mucho más complejo que requiera más tiempo del límite impuesto en el trabajo. Las diferentes funcionalidades que se implementan son las siguientes:

- **Ruta de paquetes:** Se utiliza para mostrar la ruta tomada por los paquetes o diagnosticar posibles problemas de comunicación en la ruta tomada por estos, de esta manera podemos ver desde un mapa en cuál punto puede haber un fallo de entrega de paquetes o latencias altas, además incorpora funcionalidades de resolución de nombre de dominio e IP.
- **Capturar paquetes:** Se utiliza para la captura de paquetes y se envía la información de cada paquete al Front-End donde se visualizará de mejor manera estos.
- **Comprobar Estado de una máquina:** Permite saber el estado de una máquina remota o local, se muestra en pantalla información de sí la máquina está activa o no, además de una pequeña estadística del valor del Round Trip Time (RTT) y cantidad de paquetes enviados y perdidos.
- **Encontrar dispositivos conectados en la red:** Se utiliza para hacer un escaneo mediante el envio de paquetes ICMP con ping en el rango de IP's disponibles y poder mostrar la lista de IP's de dispositivos conectados, así como también su nombre de dominio utilizando ARP.
- **Sockets activos:** se utiliza para mostrar los sockets activos, en qué puerto están escuchando y su estado, así como también una estadística de la cantidad de conexiones que hay y la posibilidad de matar los procesos de manera fácil y rápida.

5.1.2 Simulaciones

Las simulaciones se implementan dentro de la herramienta con el objetivo de ayudar a los usuarios en el aprendizaje de los conceptos básicos de redes, a partir de la interacción de estas, el objetivo es poder reforzar lo aprendido en clase configurando diferentes opciones para cada simulación. Debido a que la simulación de "The Mother Of All Problems" no se ha podido realizar, se obviará en este apartado.

• **Simulación de envíos y recepción de segmentos TCP**: Con esta simulación es posible configurar diferentes campos de la cabecera TCP vistos en clase como por el ejemplo el ACK, tamaño de ventana y el tamaño máximo del segmento. A partir de aquí se puede simular los diferentes envíos y recepciones de los segmentos enviados entre dos máquinas para ver como sus campos afectan o no en el envio del segmento, permite también introducir una probabilidad de perdida para ver la retransmisión provocada, envío bidireccional y también utiliza un algoritmo adaptativo para el Round Trip Time (RTT).

5.2 Requisitos del sistema

En esta sección se muestra a modo de resumen los diferentes requisitos de cada una de las funcionalidades que se han definido en la herramienta con el objetivo de mostrar de manera general lo que se espera que haga cada uno de los componentes.

RF-01 Captura de paquetes	• Escanear y mostrar paquetes de red • Mostrar información de ayuda • Buscador en la tabla
RF-02 Comprobar Estado de una máquina	• Diagnosticar estado de una máquina • Mostrar estadísticas del RTT (Round-Trip Time) • Mostrar número de paquetes recibidos y perdidos
RF-03 Ruta de paquetes	• Realizar la traza de ruta • Mostrar geolocalización de una IP • Mostrar los resultados de la traza • Resolver nombres de dominio
RF-04 Encontrar dispositivos conectados en la red	• Mostrar dispositivos conectados • Escanear red completa o un rango específico
RF-05 Ver sockets activos	• Escanear diferentes tipos de conexiones • Terminar/matar socket • Mostrar gráficos de las conexiones
RF-06 Simulación TCP	• Mostrar el proceso de comunicación entre dos máquinas con TCP • Plantear una simulación • Mostrar información de los paquetes
RF-07 The Mother of All Problems	• Mostrar el proceso de resolución del ejercicio • Mostrar detalles de los mensajes al pasar por las diferentes redes • Mostrar información de los paquetes • Plantear una simulación

Figura 8. Requisitos funcionales del Sistema.

5.3 Diseño e implementación

Para el diseño de interfaz se ha creado un Sketch de las diferentes vistas que tendría cada funcionalidad y simulación, de esta manera se ha podido implementar la interfaz teniendo un mínimo de conocimiento de cómo quedaría, se adjunta el **Sketch inicial de interfaz** en el ápendice. A continuación, se muestra la implementación final de la interfaz, así como también una explicación breve del funcionamiento:

- **Pantalla principal:** Menú de navegación desplegable en el lateral izquierdo donde se listan dos apartados, funcionalidades y simulaciones.

Figura 9. Interfaz principal de la herramienta.

- **Captura de paquetes:** Consiste en una tabla donde se muestra la información más importante de las cabeceras de los paquetes. Proporciona un formulario donde puedes elegir entre tres opciones, capturar paquetes TCP, UDP o ICMP y la cantidad de paquetes a capturar. Se ha planteado capturar principalmente estos porque son algunos de los más importantes en cuantos a la docencia de redes, de esta manera se puede apreciar los campos que contiene y su funcionamiento.
  El Back-end utiliza estos datos y ejecuta la utilidad de tcpdump que captura los paquetes, el tiempo de procesamiento de estos va en función de la cantidad de tráfico que hay en el momento y sobre todo del tráfico específico a capturar, una vez capturado los N paquetes, se envían a la vista y se muestra en forma de tabla.

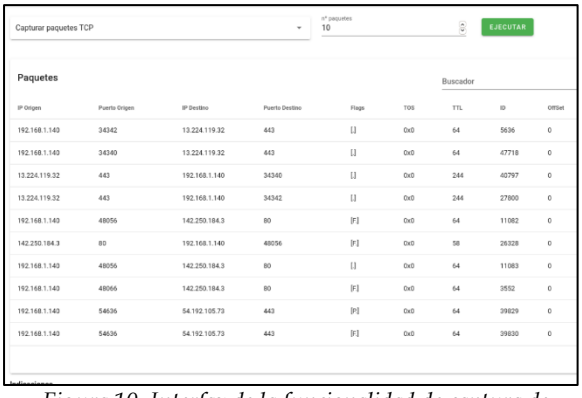


Figura 10. Interfaz de la funcionalidad de captura de paquetes.

- **Comprobar estado de una máquina:** Formulario al cual se le pasa la IP de la máquina a comprobar y el número de paquetes que se enviarán para comprobar el estado. Los datos se envían al Back-End y posteriormente envía los resultados, se gráfica los RTT obtenidos por cada paquete, se muestra una tabla con datos estadísticos del RTT (mínimo, máximo y media), muestra la cantidad de paquetes perdidos y recibidos e informa al usuario si la máquina está activa o no y, en caso de no estarlo, se

listan posibles causas del este.

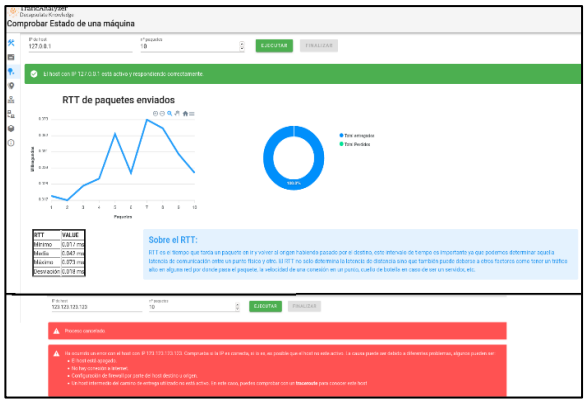


Figura 11. Interfaz de Comprobar estado de una máquina.

- **Ruta de paquetes:** Se divide en dos aspectos, el primero en la traza de ruta de paquetes y geolocalización de los saltos hechos y, el segundo, resolución de nombre de dominios e IP inverso. El desplegable contiene:

  o Trazar ruta de paquete: Solicita la IP de una máquina a trazar la ruta del paquete que se envía, de esta manera el Back-End envía los resultados a la vista, se mostrará en una tabla los diferentes saltos hechos con la respectiva IP de la máquina de cada salto y los diferentes RTT obtenidos.

  o Mostrar geolocalización de una IP: Petición HTTP desde la vista para recuperar los datos de la localización de la IP que el usuario proporciona y se muestra en el mapa con diferentes campos de interés.

  o Resolver nombre de dominio de una IP: Se envía al back-end la IP a la cual el usuario desea saber su nombre de dominio y utiliza la utilidad de nslookup para resolverla y devolver el nombre a la vista.

  o Encontrar IP a partir de un nombre de dominio: En este caso se procede de la misma manera que la opción anterior, pero se utiliza la librería dns de NPM para resolver la IP del dominio.

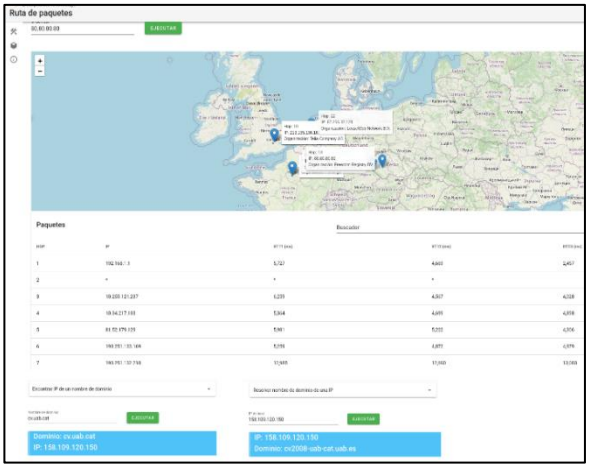


Figura 12. Interfaz de funcionalidad de ruta de paquetes.

- **Dispositivos en red:** Se proporciona la IP de la red y la interfaz para escanear la red y encontrar los dispositivos que están activos en ese momento en la red, se puede hacer una búsqueda a partir de cierto rango o bien hacer una búsqueda completa en todo el rango de IP's disponible.

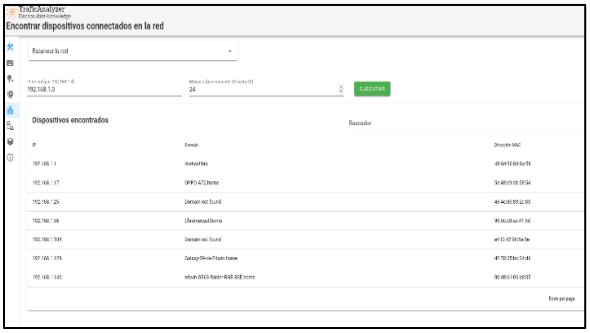


Figura 13. Interfaz de funcionalidad de encontrar dispositivos conectados en la red.

El usuario introduce la IP de la red en la cual está conectado, la máscara de red o un entero que específica hasta cuál dirección IP quiere realizar la búsqueda, se escanea todas las IP's parcial o total de esa red a partir de envio de paquetes ICMP para recuperar las IP's activas y así también obtener con ARP las direcciones físicas, se envía los datos a la vista en donde se muestra en una tabla la IP, el nombre de dominio y su dirección física.

- **Sockets activos:** Consiste en escanear todas las diferentes conexiones activas en la máquina, se muestra cada conexión en una tabla con los campos importantes como el estado de la conexión, IP's, puertos e identificador del proceso.

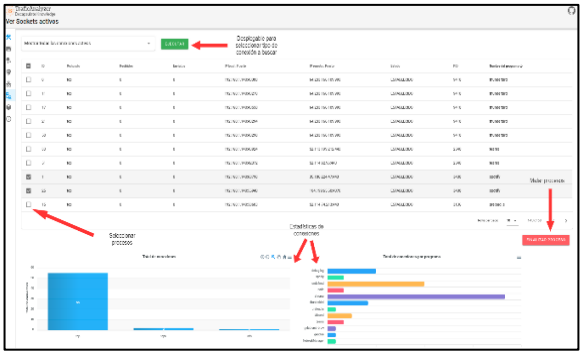


Figura 14. Interfaz de funcionalidad de ver conexiones activas.

En este caso se puede mostrar todas las conexiones activas, las de estado "listening", TCP o UDP, como tal no tienen un estado ya que no establece una conexión como TCP, aparece en la lista si está activo, pero el campo de estado está vacío, también se puede seleccionar conexiones activas y cerrarlas.

- **Simulación TCP [19][20][21]:** consiste en un formulario donde se específica algunos de los campos más importantes del protocolo TCP, los cuales son el Máximo Tamaño del Segmento (MSS), el tamaño de la ventana y el número de bytes totales que un host quiere enviar. También se ha incluido el campo de especificar la tasa de perdida de segmentos para que el estudiante pueda simular casos en que se pierden estos.

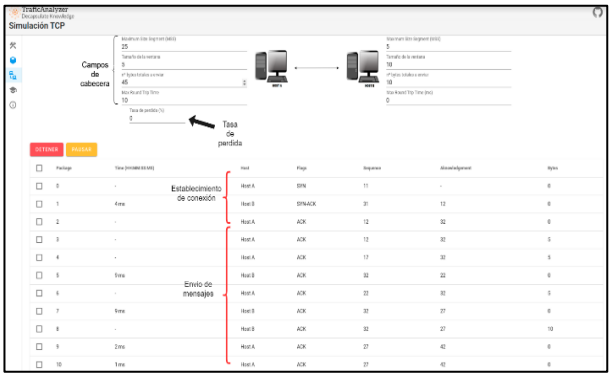


Figura 15. Interfaz de simulación de TCP.

Al comenzar la simulación, la tabla va mostrando todos los segmentos que viajan de un host a otro cada segundo, el estudiante podrá ser capaz de pausar la simulación y reanudarla si así lo desea, existen diferentes conceptos que se aplican en la simulación como lo es el establecimiento y cierre de la conexión, la ráfaga de segmentos, la perdida de segmentos, el envio bidireccional de ambos hosts, el control de la ventana, tamaño de segmentos y el cálculo del RTT a partir de un algoritmo adaptativo con estrategia de backoff del timeout.

6 Problemas surgidos

- **Tiempo real de los paquetes con tcpdump**: La idea ha sido capturar paquetes en tiempo real con el objetivo de que, a cada paquete capturado, se reflejaba uno a uno en tiempo real. Se ha intentado el uso de eventos de JavasScript, funcionaba de manera correcta para varias utilidades, pero tcpdump no enviaba el flujo de datos, se intentó ejecución síncrona y asíncrona con procesos hijos sin éxito, por tanto, la funcionalidad se ha quedado en que primero los captura y luego los muestra con el fin de no retrasar el desarrollo del resto de la herramienta.

- **Electron ejecutable:** Ha surgido grandes problemas al generar una imagen ejecutable de la herramienta con electron, han ocurrido diferentes problemas a nivel de compatibilidad y/o librerías, se ha podido distribuir una versión, pero cada vez que se generaba otra, sucedía lo mismo, hasta que finalmente se pudo resolver.

- **Fase de desarrollo aún abierta:** Debido a un gran problema de causa mayor que imposibilitaba la dedicación completa al desarrollo del trabajo, no se ha podido completar de manera exitosa toda la fase de desarrollo, dejando sin implementar la simulación de "The Mother Of All Problems", por lo que sigue abierto el proyecto.

- **Retrasos de desarrollo:** Retrasos de desarrollo a nivel de código por problemas como bugs, librerías y tarea de investigación de implementación de código que retrasó y replanificó algunas tareas.

7 Planificacion futura del trabajo

El proyecto sigue en fase de desarrollo final por los problemas comentados anteriormente, el principal objetivo a corto plazo es finalizar la planificación en su totalidad y, una vez completado, mejorar cada una de las funcionalidades y simulaciones para que puedan ser completas, con más funcionalidades e integración de estadísticas de redes de datos. Aun así, se resumen en la siguiente lista las mejoras que se consideran pertinentes a realizar y que no se han podido alcanzar por falta de tiempo de desarrollo:

- Implementar la simulación "The Mother Of All Problems" y mejorar la simulación TCP para que sea más completa
- Mejorar interfaz gráfica para tener un diseño más completo
- Integrar nuevos componentes y mejorar los existentes
- Generar animaciones de las simulaciones
- Integrar nuevas simulaciones

8 Conclusión

Durante la realización del proyecto, se ha podido definir la estructura de lo que es la herramienta TraficAnalyzer a partir de la investigación previa sobre otros softwares de monitorización y simulación de redes en el mercado, gracias a esto se ha podido realizar una primera base e implementar una interfaz gráfica que permite aprovechar de forma más rápida las utilidades de redes de Linux y, también, hacer una aproximación a las simulaciones de redes, enfocando así la herramienta como una ayuda para aprender sobre los conceptos de redes y prácticar la teoría aprendida en clase de manera práctica desde una sola herramienta.

El proyecto se ha realizado utilizando una metodología incremental que permitía enfocarse en cada componente hasta su finalización, de esta manera se ha logrado tener poco a poco funcionalidades completas en base a los requisitos y, de esta manera, se iba construyendo cada vez más la herramienta sin riesgo a tener que hacer marcha atrás a una versión en su totalidad, solo parcialmente al componente que se estaba desarrollando sin que afectara a los demás.

Respecto a la planificación, su estimación ha sido la correcta, la holgura por estimación de retraso que se añadia

a cada una de las tareas era el correcto, si bien hubo retraso en algunas tareas, estás entraban en la holgura determinada en el inicio, también se ha podido realizar aproximadamente el 80% de la planificación y en base a los requisitos definidos, aun así, y por motivos de fuerza mayor, se ha retrasado gran parte de la etapa de desarrollo en el segundo bloque correspondiente a las simulaciones porque la situación que se había presentado, requería gran parte del tiempo que era para dedicarlo al trabajo y se ha quedado finalmente abierto en la fase de desarrollo.

A nivel de resultados actuales, se ha logrado implementar una herramienta con formato similar a los investigados en el estado del arte, todas las funcionalidades propuestas se han podido finalizar y que sean funcionales, con la simulación TCP también se ha logrado aproximar una simulación práctica real, también se ha cumplido el objetivo de que todas estas utilidades estuvieran integradas en una misma herramienta y que fueran fáciles de utilizar, todo esto logrando también la portabilidad de la aplicación, permitiendo poder distribuirla para cualquiera que esté interesado en utilizarla y, finalmente, la herramienta permite ser ampliable gracias a las tecnologías utilizadas.

La implementación con tecnologías de desarrollo moderna ha permitido que en el proceso de desarrollo se pudieran generar componentes reutilizables, pudiendo integrarlas en una aplicación con la misma tecnología o bien, integrar los componentes de otra aplicación a esta. Además, la utilización de estas herramientas permite retomar el desarrollo de esta teniendo una curva de aprendizaje rápida porque, son tecnologías que utilizan en gran parte lenguajes básicos de programación, en este caso destacando a VueJS.

Como conclusión general, se puede ver que se han alcanzado gran parte de los objetivos propuestos a pesar de los imprevistos que han surgido, el tiempo dedicado ha generado un esfuerzo grande y personalmente me siento satisfecho por el trabajo realizado y los resultados obtenidos ya que se ha podido aplicar lo aprendido durante el grado y, que el desarrollo de este trabajo también ha servido de preparación a nivel práctico y profesional. Aunque se hayan quedado cosas por implementar y, haya muchas cosas que mejorar, considero que todo tiene una base para comenzar e ir creciendo cada vez más y que esta herramienta puede llegar a ser muy útil a medida que se vaya mejorando, incluso llegando a ser una herramienta Open-Source práctica de monitorización general de redes de datos.

AGRADECIMIENTOS

Antes que nada, agradecer a Juan Antonio que ha sido mi tutor durante todo este tiempo y me ha ayudado en cada momento y situación que se ha presentado, siempre atento y demostrando preocupación por lograr todos los objetivos marcados en este trabajo y hacerlo lo mejor posible. También agradecer a Ian Blanes por dedicar un tiempo a responder los correos con dudas referentes a ciertas tecnologías. Agradecer a mis amigos por todo el apoyo que me dieron en todo momento. Finalmente, agradecer demasiado a mi padre que ya no está con nosotros, por darme todo en esta vida, y apoyarme en todo hasta el final.

BIBLIOGRAFIA

[1] Tcpdump.org. 2021. TCPDUMP/LIBPCAP public repository. [online] Available at: <https://www.tcpdump.org/index.html> [Accessed 4 March 2021].
[2] Networking-Academy.2021. [online] Available at: <https://www.netacad.com/es/courses/packet-tracer/faq#1> [Accessed 20 June 2021].
[3] Docs.gns3.com. 2021. Getting Started with GNS3 | GNS3 Documentation. [online] Available at: <https://docs.gns3.com/docs/> [Accessed 5 March 2021].
[4] Media.pearsoncmg.com. 2021. TCP Congestion Control. [online] Available at: <https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/tcp-congestion/index.html> [Accessed 4 March 2021].
[5] Media.pearsoncmg.com. 2021. Flow Control. [online] Available at: <https://media.pearsoncmg.com/aw/ecs_kurose_compnetwork_7/cw/content/interactiveanimations/flow-control/index.html> [Accessed 5 March 2021].
[6] Programmersought.com. 2021. Analyze the sliding window of TCP (animation demonstration) - Programmer Sought. [online] Available at: <https://www.programmersought.com/article/56586098594/> [Accessed 16 March 2021].
[7] Wireshark.org. 2021. Wireshark ·Go Deep.. [online] Available at: <https://www.wireshark.org/> [Accessed 4 March 2021].
[8] Solarwinds.com. 2021. Software de gestión de redes: principales herramientas de red | SolarWinds. [online] Available at: <https://www.solarwinds.com/es/network-management-software> [Accessed 4 March 2021].
[9] Nagios. 2021. IT Monitoring Software from Nagios. [online] Available at: <https://www.nagios.com/products/> [Accessed 4 March 2021].
[10] Vuejs.org. 2021. Introduction — Vue.js. [online] Available at: <https://vuejs.org/v2/guide/#What-is-Vue-js> [Accessed 6 March 2021].
[11] Vuetify. 2021. Vuetify — A Material Design Framework for Vue.js. [online] Available at: <https://vuetifyjs.com/en/> [Accessed 13 March 2021].
[12] GeeksforGeeks. 2021. Software Engineering | Incremental process model - GeeksforGeeks. [online] Available at: <https://www.geeksforgeeks.org/software-engineering-incremental-process-model/> [Accessed 14 March 2021].
[13] Leafletjs.com. 2021. Leaflet — an open-source JavaScript library for interactive maps. [online] Available at: <https://leafletjs.com/> [Accessed 13 March 2021].
[14] Ipapi.co. 2021. ipapi - API Reference | IP Location Examples. [online] Available at: <https://ipapi.co/api/#complete-location> [Accessed 18 April 2021].
[15] ApexCharts.js. 2021. ApexCharts.js - Open Source JavaScript Charts for your website. [online] Available at: <https://apexcharts.com/> [Accessed 18 April 2021].
[16] npm. 2021. cors. [online] Available at: <https://www.npmjs.com/package/cors> [Accessed 20 June 2021].
[17] Node.js. 2021. About | Node.js. [online] Available at: <https://nodejs.org/en/about/> [Accessed 7 March 2021].
[18] Nodejs.org. 2021. Child process | Node.js v15.11.0 Documentation. [online] Available at: <https://nodejs.org/api/child_process.html> [Accessed 7 March 2021].
[19] 3-end-to-end Protocols, Computer Networks, 2019. Sergi Robles
[20] Comer, D., 2014. Internetworking with TCP/IP. Upper Saddle River: Prentice-Hall International.
[21] Transmission Control Protocol. 2021. [online] Available at: <https://w3.ual.es/~vruiz/Docencia/Apuntes/Networking/Protocols/Level-4/05-TCP/index.html#x1-80004.3> [Accessed 30 May 2021].

APÉNDICE

A1. ENLACES DE LA HERRAMIENTA

GitHub: edwinHernandezB/TraficAnalyzer (github.com)

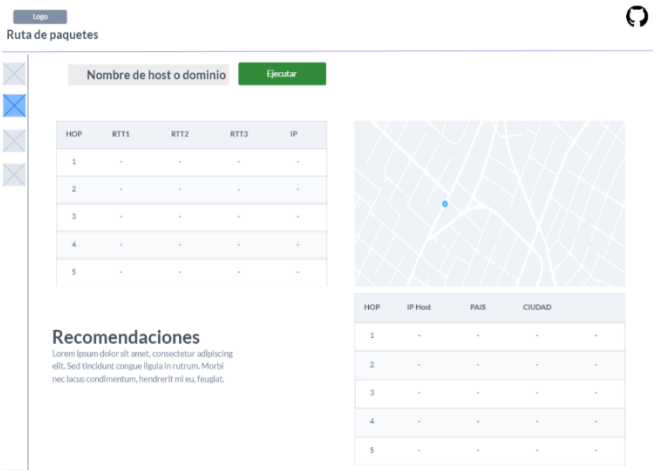
Ejecutable de la aplicación:
https://mega.nz/file/hMpkwCQB#VaMYmfYN6ajxgD8WnsM11_F-Z-l28-8VrGp8f7dnvgY

Guia de uso e instalación de utilidades:
https://mega.nz/file/wE4FAaCZ#_w6pKWJax2JIzh1WawEzflPY--b1gxxjT2C2kmq3x1w

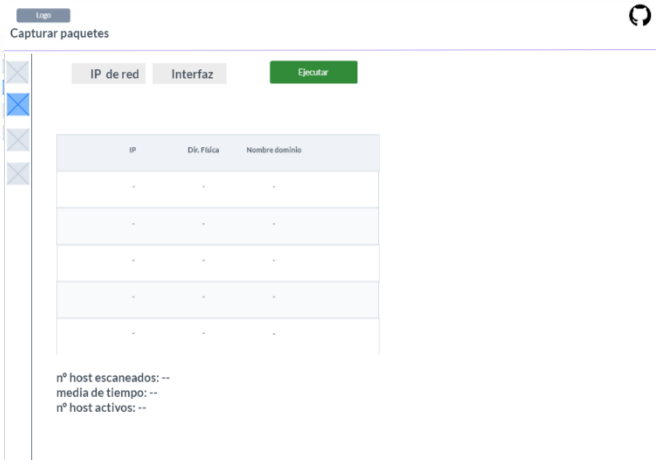
A1. SKETCH INICIAL DE INTERFAZ

Las siguientes figuras son los sketchs diseñados en la fase inicial antes del desarrollo de la herramienta

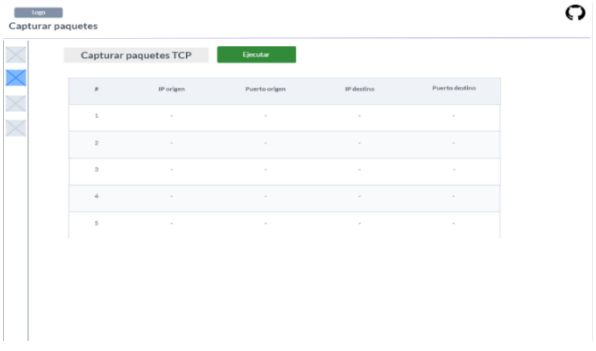
Interfaz principal



Ruta de paquetes

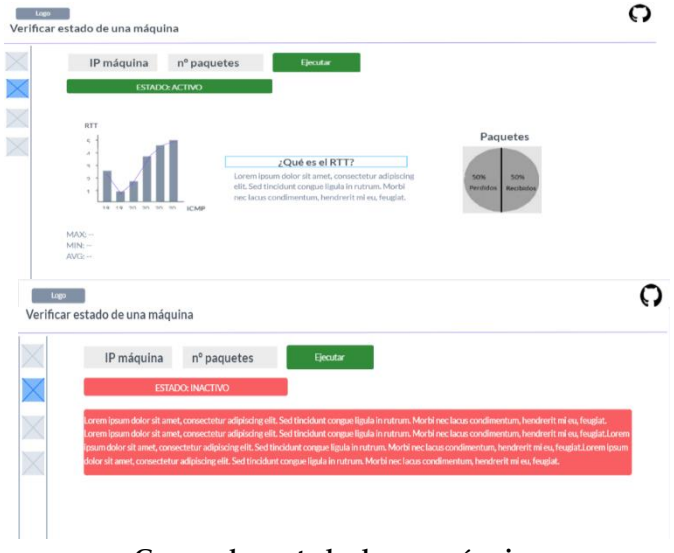


Encontrar dispositivos conectados en la red



Captura de paquetes

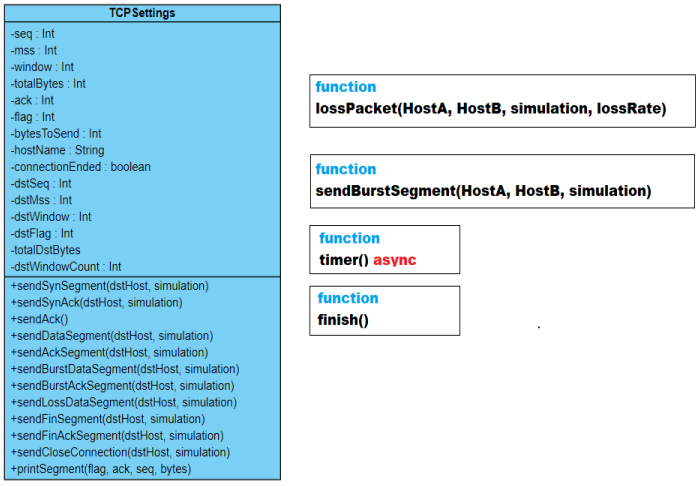
Encontrar sockets activos



Comprobar estado de una máquina

A2. CLASES DE LA SIMULACIÓN TCP

La simulación TCP ha sido creada a partir de código, por lo que se muestra a continuación las funciones y clases utilizadas.



Clase de configuración de paquetes TCP y funciones que permiten la ejecución de la simulación

A3. DIAGRAMA DE GANNT