ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

Desarrollo de un aplicativo para caracterizar tramas 802.11 en redes inalámbricas utilizando software libre

Lara Hidalgo, Erick Danilo

Departamento de Eléctrica, Electrónica y telecomunicaciones

Carrera de Ingeniería en Electrónica y Telecomunicaciones

Trabajo de titulación previo a la obtención del título de

Ingeniero en Electrónica y Telecomunicaciones

M.Sc. Romero Gallardo, Carlos Gabriel

9 de septiembre del 2020

URKUND

Urkund Analysis Result

Analysed Document:	Tesis_Danilo_Lara.docx (D78522064)
Submitted:	9/3/2020 3:37:00 AM
Submitted By:	cgromero@espe.edu.ec
Significance:	2 %

Sources included in the report:

AEC2 Análisis de WLAN y WPAN con Wireshark.pdf (D45442722)
https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/2819/1/AGP10.pdf
https://riuma.uma.es/xmlui/bitstream/handle/10630/8409/pfc.pdf?sequence=1&isAllowed=y

Instances where selected sources appear:

3

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que el trabajo de titulación, "**Desarrollo de un aplicativo para caracterizar tramas 802.11 en redes inalámbricas utilizando software libre**" fue realizado por el señor **Lara Hidalgo, Erick Danilo** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 9 de septiembre del 2020

Firma:

Firmado electrónicamente por:
CARLOS GABRIEL
ROMERO GALLARDO

Ing. Romero Gallardo, Carlos Gabriel

C. C: 1712198066

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

RESPONSABILIDAD DE AUTORÍA

Yo, **Lara Hidalgo, Erick Danilo**, con cédula de ciudadanía n°1600469470, declaro que el contenido, ideas y criterios del trabajo de titulación**: Desarrollo de un aplicativo para caracterizar tramas 802.11 en redes inalámbricas utilizando software libre** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 9 de septiembre de 2020

Firma

Lara Hidalgo, Erick Danilo

C.C.: 1600469470

DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES

AUTORIZACIÓN DE PUBLICACIÓN

Yo **Lara Hidalgo, Erick Danilo,** con cédula de ciudadanía n° 1600469470, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Desarrollo de un aplicativo para caracterizar tramas 802.11 en redes inalámbricas utilizando software libre** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 9 de septiembre de 2020

Firma

Lara Hidalgo, Erick Danilo

C.C.: 1600469470

Dedicatoria

Esta tesis se la dedico en primera instancia a mis queridos padres Bolívar y Miriam por su amor incondicional y su apoyo infinito siendo mi fortaleza en todas las etapas de mi vida, por enseñarme valores y principios que me ayudaron a cumplir todas mis metas, por ser mi soporte.

A mis hermanos Jairo y Salomé por brindarme su cariño, a mi abuelita quien es mi ejemplo a seguir, estoy eternamente agradecido por sus enseñanzas y en general a toda mi familia que fueron pieza fundamental en obtener este importante logro en mi vida profesional.

A Jazmín mi novia, por ser parte de mi vida y estar a mi lado en las buenas y malas, fue tu motivación y esfuerzo el que me ayudó alcanzar este logro académico.

Erick Danilo Lara Hidalgo

Agradecimiento

Agradezco a Dios y a mi familia por darme la fortaleza necesaria para afrontar las adversidades que se presentaron para concluir esta etapa académica con éxito.

A la Universidad de las Fuerzas Armadas “ESPE” por formarme académicamente, así como a mi director de tesis, Ing. Carlos Romero por confiar en mi para realizar este proyecto, por ser un guía y por compartirme sus conocimientos.

Erick Danilo Lara Hidalgo.

Índice de contenidos

Índice de tablas

Índice de figuras

Resumen

El uso de herramientas adecuadas para gestionar redes inalámbricas es de gran importancia para el monitoreo, identificar amenazas y sobre todo para tomar decisiones adecuadas. Es por cuanto, el presente proyecto tiene como fin la creación de un aplicativo que permite obtener parámetros determinados de forma automatizada del proceso de caracterización de redes inalámbricas en tiempo real utilizando software libre, permitiendo escoger una característica determinada del tráfico de red, capturando las tramas y recopilando la información para posteriormente ser desplegada dentro de una interfaz gráfica creada en Python. El funcionamiento del aplicativo se basa en un Software libre que permite la caracterización de tramas de redes inalámbricas en tiempo real, este sistema admite que las personas que se encargan de la administración de redes mejoren el proceso de análisis de tramas 802.11.
En este sentido, compararlo con las características obtenidas en un sniffer llamado Wireshark. a partir de los resultados obtenidos en las pruebas se pretende realizar el análisis de resultados y las conclusiones correspondientes del trabajo realizado. Por lo cual, la aplicación pretender ser amigable con los usuarios y que la información obtenida mediante sea más eficiente y arroje datos útiles, tomando en cuenta cada una de las premisas idóneas que se hayan adoptado los beneficiarios directos e indirectos a través de un sistema fácil de manejar y con resultados oportunos.

Palabras clave:

- **TRAMA**
- **PYTHON**
- **ESTÁNDAR 802.11**
- **SNIFFER**

Abstract

The use of adequate tools to manage wireless networks is of great importance for monitoring, identifying threats and, above all, for making appropriate decisions. Therefore, the purpose of this project is to create an application that allows automated parameters to be obtained from the characterization process of wireless networks in real time using free software, allowing you to choose a specific characteristic of network traffic, capturing the frames and collecting the information to be later displayed within a graphical interface created with Python.

The operation of the application is based on free software that allows the characterization of frames of wireless networks in real time, this system admits that the people who are in charge of network administration improve the process of analyzing frames 802.11.

In this sense, compare the application with the characteristics obtained in a sniffer called Wireshark. From the results obtained in the tests, it is intended to carry out the analysis of results and the corresponding conclusions of the work carried out.

Therefore, the application pretends to be user-friendly and that the information obtained through it is more efficient and yields useful data, taking into account each of the ideal premises that direct and indirect beneficiaries have adopted through an easy system. to handle and with timely results.

Key words:

- **FRAME**
- **PYTHON**
- **STANDARD 802.11**
- **SNIFFER**

Capítulo I

Introducción

Antecedentes

En la actualidad el desarrollo de las redes inalámbricas en el área local (WLAN) se ha relacionado directamente a través del uso del estándar 802.11, es por cuanto su distribución de chip sets WIFI ha sido global donde se han registrado alrededor de 20 mil millones de estos dispositivos en estos últimos 5 años, por lo cual la gran demanda y el acelerado crecimiento de aplicaciones que requieren grandes descargas, así como transmisiones de video 4K y el uso de realidad virtual y las interferencias entre los dispositivos WIFI y no WIFI mismo que operan en un mismo canal, han llevado a que en los últimos 20 años la velocidad de las redes haya tenido un creciente aumento de 11 Mbps a varios Gbps, de manera que se ha realizado una comprensión, evaluación y optimización continua de las redes 802.11.

En este sentido, los canales variables permiten que las redes 802.11 tengan una conductividad factible, aunque los dispositivos físicos no la experimentan de esta forma; Adebi, A (2017) asevera que existen varias metodologías que proporcionan un intervalo de confianza y en algunos casos generan datos falaces, por este motivo se plantea la creación de una metodología de valoración empírica con el fin de evaluar ambientes repetitivos, utilizando trazas que unen indagaciones reales con las características que brindan los simuladores, dichas trazas se agrupan de acuerdo a los posibles dispositivos para posteriormente examinar algoritmos y sistemas, del mismo modo, incluye la evaluación de las velocidades de transmisión en el estándar 802.11n, para obtener la unión exitosa de la singularidad de la capa física, para encontrar como resultado final la existencia de conexión entre las tasas de transmisión en largos periodos de tiempo en ambientes con interferencia y movilidad (Adebi., A 2017).

Del mismo modo, como se había destacado anteriormente las redes 802.11 poseen un crecimiento exponencial a nivel mundial, dado a que sus características que son ajustables a todo tipo de infraestructuras y sobre todo son flexibles, por otro lado poseen un gran inconveniente que es ser vulnerable a ataques informáticos lo que ocasionan pérdidas monetarias y de datos, es por ello que, Medina & Rivas en su investigación "Evaluación del Rendimiento de un Sistema de Detección de Intrusos para Redes Inalámbricas 802.11 Contra Ataques Informáticos"; buscan estimar la capacidad de un sistema que permite hallar intrusos en redes 802.11, realizado en un SO dedicado a la seguridad informática como Kali Linux, para a continuación realizar un sin número de ataques a dichas redes y evaluar según los IDS puestos en marcha, para de este modo obtener como resultado un análisis comparativo que permita reconocer el mejor sistema en identificar intrusos en diversos entornos (Medina & Rivas, 2019).

En lo que concierne al tema abordado, en los últimos años el aumento masivo de redes inalámbricas debido a sus beneficios tales como, costos inferiores, sencilla instalación, factible operación y oportunidades de transmisión, por dichas razones es la tecnología más factible para ser utilizada en ambientes en el que resulta casi imposible dar servicio de Ethernet, se emplea tanto para transmitir datos entre equipos como para acceder a internet, sin embargo estas redes inalámbricas poseen riesgos ya que viajan por medio del aire y son vulnerables a ser atacadas por personas que poseen los estudios y equipos necesarios, es por cuanto al instalar medidas de seguridad , podemos evitar ataques invasivos que obtengan información relevante.

Angón, E (2014) señala todos los protocolos de seguridad para redes inalámbricas, así como ventajas y desventajas para obtener un sistema seguro, desde los métodos más básicos hasta sistemas con un elevado grado de dificultad, de modo que los usuarios puedan elegir el más adecuado con el fin de poder proteger sus redes (Angón E., 2014).

Un ejemplo claro del desempeño de estas redes, dan a conocer Álvarez, Isbarbo, & Rivas en su estudio "Sistema de monitoreo de capacidad de redes inalámbricas 802.1"; en el que diseñan e implementan un aplicativo que obtiene en tiempo real un señalizador del desempeño de las redes 802.11 en el Plan Ceibal, mediante la elaboración de un modelo matemático que permite estimar el Throughput TCP máximo de bajada que se obtiene entre usuarios que se conecten a la red, utilizando como característica principal las tasas de transmisión de los usuarios, por consiguiente dar resultados errores estimados inferiores al 7%, para posteriormente utilizar el modelo para lograr el Throughput en tráfico de bajada, este último modelo se lo aplica y realizan pruebas en la red Ceibal en una institución educativa alcanzando resultados con errores menores del 10% en el 90% de la población total evaluada, al analizar estos datos se desarrolla una herramienta de monitoreo que permita desplegar en forma real los indicadores de los modelos ejecutados (Álvarez, Isbarbo, & Rivas, 2016).

Páez, T (2015) asevera la implementación de un analizador portable de tráfico de redes wifi del estándar 802.11, siendo la principal herramienta una Raspberry Pi, con la que se obtiene un menor consumo de energía, una mayor velocidad al momento de procesar datos y permite una ágil conexión, con el apoyo del programa Kismet permite el diseño e implementación de este analizador de redes en tiempo real dentro de los laboratorios de la Universidad de las Fuerzas Armadas - ESPE, localizando los AP mediante modelos de georeferencia para examinar el desempeño del prototipo por medio del roaming del cliente, mediante los datos obtenidos en la captura del tráfico se identifica la distribución de la infraestructura de red en la universidad, así como las características indispensables para una correcta construcción de una red como por ejemplo: los estándares usados, seguridad de las redes, velocidades de los AP, eficiente entre otras características (Páez., T 2015).

En consecuencia, se monitorea y analiza redes wifi con el objetivo de adquirir información de la red y el uso de los usuarios para la implementación de un software de monitoreo, el cual labora en forma pasiva para realizar la captura de tramas para a continuación analizar dichas tramas y obtener información de las características más importantes las mismas que serán almacenadas en bases de datos para realizar su análisis. Para su metodología se realizaron 4 etapas:

En primera instancia la inmersión, que trata de una investigación teórica de herramientas para estudiar redes inalámbricas, estándar 802.11, bases de datos y Python; seguido del desarrollo, que pretende la creación de un sniffer en Python para el análisis de las redes; continuando con pruebas de monitoreo en oficinas realizadas durante dos meses, y finalmente el análisis respectivo de manejo de la información. (Gaitan., V 2017)

Justificación

En la actualidad existen diversas herramientas para el análisis y monitoreo de tráfico de redes inalámbricas que son de gran importancia para gestionar sistemas de Telecomunicaciones tales como Wireshark, Tcpdump, OmniPeek, NetStumbker, entre otras. Dichos sniffers permiten anticipar así como reparar problemas, detectar amenazas y tomar decisiones correctas a la planificación de la red (Salazar, 2016).

Dichas herramientas permiten el análisis de trafico de redes inalámbricas previo la captura de datos, suben la información a la plataforma y entregan resultados, esta característica no permite que la información sea en tiempo real. Analizar el tráfico en tiempo real es fundamental para la correcta gestión de red (Thomas, 2006).

Por otro lado, su aplicación es cuantiosa ya que implica la compra de diversas herramientas que permitan su correcto funcionamiento en las diferentes plataformas.

Los analizadores de protocolos muestran los datos de los paquetes, puesto que la captura y visualización de paquetes por sí solos pueden ser ineficientes, es por cuanto, para lograr comprender su comportamiento de la red, es necesario los paquetes de datos y la información del flujo de tráfico, ejemplificando las aplicaciones y servicios disponibles, la utilización de los recursos de ancho de banda y las anomalías en materia de seguridad (Quiñones, Coya, & Marichal, 2012).

Las redes inalámbricas tienen tramas de control y tramas de gestión que comprenden una serie sucesiva de bits, que se encuentran organizados en forma cíclica, lo mismos que transportan información y permiten su extracción, conocido como paquete de red modelo OSI (Gaitan, 2017), dicha información permite reconocer problemas de seguridad, cobertura o alcance.

No obstante, para la realización de dicho análisis es de vital importancia la intervención de un usuario que mediante una orden permita la ejecución de este proceso.

Es por cuanto, la presente investigación pretende la creación de un aplicativo, que permita al usuario poder escoger los parámetros a ser analizados mediante la captura de tramas en tiempo real que admita desplegar características del tráfico que captura las premisas antes determinadas guardando los datos relevantes permitiendo un breve análisis (Gaitan, 2017).

Los beneficiarios de esta investigación como herramienta de forma directa sería aquellos usuarios que se encuentren dentro del campo del análisis de tráfico, administradores de red o de seguridad, y de forma indirecta todos aquellos usuarios que hagan uso de redes inalámbricas.

Objetivos

General

Desarrollo de un aplicativo para caracterizar las tramas 802.11 en una red inalámbrica

por medio del desarrollo de una aplicación usando herramientas de software libre.

Específicos

- Levantar el estado del arte acerca de caracterización de tráfico en redes, así como el uso de herramientas de software libre para el efecto.
- Analizar el tráfico de redes inalámbricas para escoger las características y parámetros a utilizar.
- Seleccionar las herramientas de software libre para la automatización del proceso de caracterización del tráfico de redes inalámbricas.
- Crear un aplicativo para la caracterización del tráfico de datos en redes inalámbricas utilizando las herramientas de software libre escogidas.
- Realizar pruebas de funcionamiento de la aplicación en casos reales.
- Comparar los resultados con relación a al sniffer Wireshark.

Capítulo II

Marco teórico

Redes WiFi

WiFi (Wireless Fidelity) es la tecnología que permite la comunicación inalámbrica de equipos o dispositivos, que se encuentran conectados a internet, dicha red se basa en el estándar IEEE 802.11 de la cual se define su clasificación en los tipos a, b, c, d, f, entre otros (García et al., 2018).

Características de las redes WiFi

Reducen el uso de cables permitiendo de esta manera una instalación sencilla, por otro lado, se puede conectar en áreas con dificultad de conexión cableada, son de fácil acceso comercial; además se puede conectar a la red múltiples equipos sin necesidad de invertir en infraestructura adicional.

Cabe mencionar que la WiFi Alliance certifica que todos los dispositivos que trabajan con wifi son compatibles en su totalidad en cualquier parte del mundo.

Estándar IEEE 802.11

A lo largo de los años algunas organizaciones se han encargado de estandarizar las redes inalámbricas, por ejemplo, en Estados Unidos es el Instituto de Ingeniería Eléctrica y Electrónica (IEEE) y en Europa el Instituto Europeo de Normas de Telecomunicaciones (ETSI). Hoy en día la organización encargada de promover, estandarizar y certificar la tecnología con el estándar 802.11 es la WiFi Alliance.

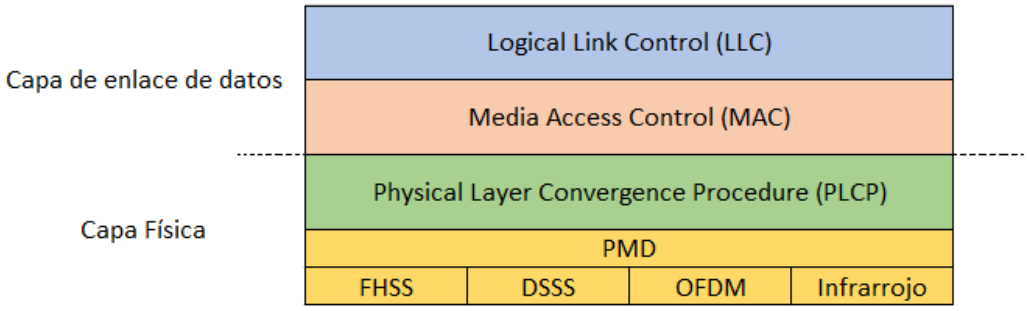
Inicialmente en 1997 la IEEE, definió el primer estándar IEEE 802.11, pero fue hasta 1999 que se empezó a comercializar mundialmente. El estándar IEEE 802.11 es aquel que define las redes WiFi, su objetivo es trabajar en las bandas de frecuencia 2.4 GHz (ISM) y 5GHz (U-NII), además, sigue como esquema de referencia el modelo OSI y sus capas: física y de enlace de datos (Campo et al., 2019, p. 322).

Dentro de la capa física se encuentran 2 subcapas: la subcapa inferior o PMD, que es el conjunto de declaraciones de los sistemas para transmitir a nivel físico, en el cual se definen 4 tipos: FHSS, DSSS, OFDM e infrarrojo y la subcapa superior o PLCP, que es responsable de adecuar las declaraciones de la subcapa inferior a la subcapa MAC.

De igual forma, la capa de enlace de datos está compuesta de las subcapas MAC en la que se define el protocolo de acceso al medio, así como él envió de ACKs, fragmentación de tramas y la encriptación de los datos y por último la subcapa LLC encargada de brindar el transporte para cualquier tecnología, como se observa en la Figura 1 (Anguís, 2008).

Figura 1

Modelo de referencia del estándar IEEE 802.11



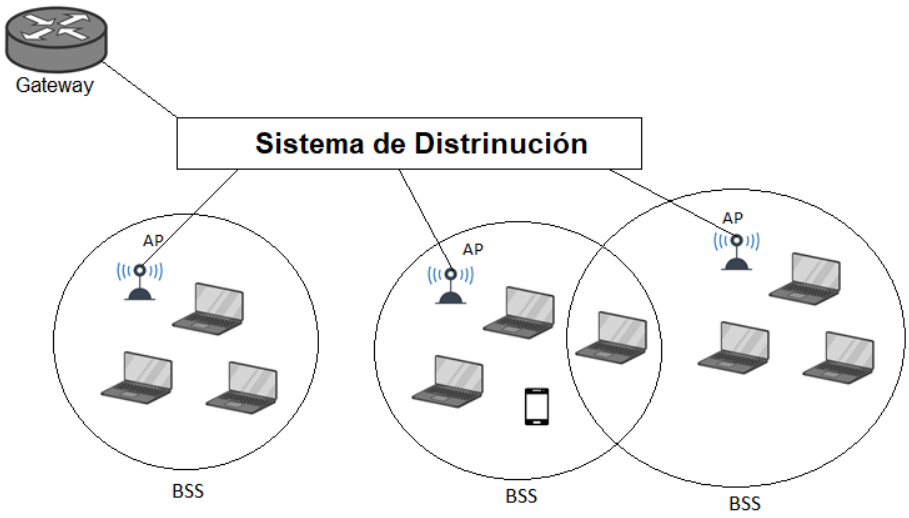
Nota. Tomado de *Redes de Área Local Inalámbricas: Diseño de la WLAN de Wheelers Lane Technology College*, por J. Anguís, 2008.

Componentes de la red

- Estación: Elemento encargado de conectarse al ambiente inalámbrico.
- AP (Access Point): Es el dispositivo al cual se conectan los terminales inalámbricos (WT) para comunicarse entre sí, así como se encarga de coordinar la transmisión mediante los clientes.
- BSS (Basic Service Set): Conjunto de estaciones que forman una comunicación entre ellas, cuando se incluye una AP al BSS se lo conoce como BSS de infraestructura.
- IBSS (Independent Basic Service Set): Se denomina IBSS a las estaciones BSS que mantienen una comunicación sin necesidad de una red cableada, por lo general se los crea con un propósito en particular de forma temporal en la que se encuentran pocas estaciones.
- ESS (Extended Service Set): Es una configuración que interconecta a 2 o más BSS, cada BSS posee un AP y estos se comunican entre sí mediante un sistema de distribución (DS) que permite optimizar el tráfico entre estaciones (Arizo, 2016).

Figura 2

Extended Service Set (ESS)



Nota. Tomado de *Repositorio Digital - EPN: Estudio, pruebas y simulación del estándar IEEE 802.11ac basándose en MU-MIMO (MIMO Multiuser)*, por L. F. A. Arizo, 2016.

Tipos de trama

Las tramas se clasifican según la función que vayan a realizar, el estándar 802.11 tiene 3 tipos de tramas:

- Tramas de datos: Como su nombre lo indica sirve para transportar datos o información hacia las capas superiores.
- Tramas de control: Son aquellas que se encargan de controlar el acceso al medio, con la entrega de las tramas de datos hacia las estaciones.
- Tramas de gestión o administración: Permiten mantener las comunicaciones con las estaciones transportando información de gestión, pero no a capas superiores.

En el presente caso de estudio las tramas que se analizarán son las de administración y control (Páez, 2015).

Tramas de administración

Trama Beacon.

Este tipo de trama envía el AP de manera constante para informar a las estaciones cercanas de la aparición de redes wifi con sus respectivas características y poder conectar con dicho AP, de igual forma permite a la estación enlistar los AP disponibles en todos los canales 802.11.

Trama Probe Request.

Son enviadas por los clientes cuando requieren información de algún AP, que se encuentre en el rango de cobertura, especificando su SSID en broadcast (Páez, 2015).

Trama Probe Response.

Es la respuesta que da la estación cuando recibe un Probe Request, esta trama solo la recibe el cliente que realizó el pedido (unicast) (Páez, 2015).

Trama Authentication.

El cliente envía al AP una trama de autenticación para comprobar su identidad para unirse a la red, existen dos tipos de autenticación:

Autenticación abierta. En este caso el cliente envía la trama de autenticación y el AP responde con una trama de autenticación indicando si acepta o no a la estación.

Autenticación con clave compartida. En este caso la estación debe conocer la clave para poner unirse a la red. El AP comprueba si la estación conoce la llave correcta enviando una trama de texto de respuesta y la estación deberá enviar la llave encriptada una vez validada la identidad de la estación el AP envía una trama aceptando a dicha estación (Páez, 2015).

Trama Association Request.

Este tipo de trama es usada por la estación cliente para de esta forma empezar el proceso de asociación para que el AP reserve recursos y finalmente se sincronice estableciendo un ID de asociación con la estación que realizó la petición (Páez, 2015).

Trama Association Response.

Es enviada por el AP como respuesta a una trama Association Request e informa la aceptación o rechazo de la estación solicitante (Páez, 2015).

Tramas de control

Trama Request To Send (RTS).

La estación envía una (RTS) para empezar con la comunicación de dos vías con el fin de enviar tramas, disminuir colisiones siempre y cuando se tenga dos estaciones asociadas al mismo AP. (Páez, 2015)

Trama Clear To Send (CTS).

Es la respuesta de la estación a una trama RTS para informar que el canal queda libre de transmisiones, estas tramas ayudan en el control de colisiones ya que contienen un valor de tiempo, por ende si la estación solicitante está transmitiendo las demás estaciones no pueden ejecutar esta acción. (Páez, 2015)

Trama Acnknowledgement (ACK).

La finalidad de ACK es asegurar el recibimiento de tramas, el cliente receptor envía una ACK al emisor en caso de no encontrar ningún error, por otro lado, si el cliente emisor no adopta una ACK en un tiempo determinado se envía nuevamente. (Páez, 2015)

Formato de las tramas

Las Trama 802.11 están compuestas de la siguiente estructura:

Figura 3

Formato de la trama 802.11

Nota. Tomado de *ESTUDIO Y ANÁLISIS PARA LA ACTUALIZACIÓN DE RED WLAN DE LA SEPS UTILIZANDO TECNOLOGÍAS BASADAS EN EL ESTÁNDAR IEEE 802.11 AC,* por S. Estrella, 2017.

Preámbulo

- Synch (sincronizar): Es una secuencia de 80 bits entre 1 y 0 alternados usada para sincronizarse con los paquetes recibidos o para escoger la antena adecuada (Estrella, 2017).

- SDF (Start Frame Delimiter): Como su nombre lo indica es un delimitador que se emplea para precisar la temporización de trama, es un patrón definido de 16 bits 0000 1100 1011 1101 (Estrella, 2017).

Cabecera PLCP

Es transmitida siempre a 1 Mbps y almacena información lógica que será usada por la capa PHY con el fin de decodificar las tramas, está formada por:

- Longitud de palabra: Almacena el total de bytes que conforman el paquete para identificar el final del mismo.

- Campo de señalización PLCP: Encargado de contener las velocidades de datos.

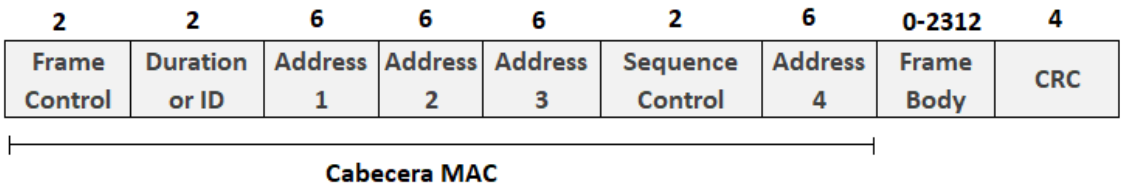
- HEC (Header Error Check Field): Campo encargado de identificar errores CRC, formado por 16 bits (Estrella, 2017).

MAC Data

En la Figura 4, se muestra la estructura general de la trama MAC, los campos cabecera MAC y CRC se encuentran en bytes y por otro lado Body Frame en bits (Estrella, 2017).

Figura 4

Estructura general trama 802.11



Nota. Tomado de *ESTUDIO Y ANÁLISIS PARA LA ACTUALIZACIÓN DE RED WLAN DE LA SEPS UTILIZANDO TECNOLOGÍAS BASADAS EN EL ESTÁNDAR IEEE 802.11 AC,* por S. Estrella, 2017.

Frame Control.

La información contenida en cada campo se encuentra en bytes.

Figura 5

Campos Frame Control

Nota. Tomado de *ESTUDIO Y ANÁLISIS PARA LA ACTUALIZACIÓN DE RED WLAN DE LA SEPS UTILIZANDO TECNOLOGÍAS BASADAS EN EL ESTÁNDAR IEEE 802.11 AC,* por S. Estrella, 2017.

Protocolo Version. Se la usa para definir la versión del estándar 802.11, por ende, identificar si es a, b, entre otros (Estrella, 2017).

Type y Subtype.

Se utilizan estos 6 bytes para identificar si es una trama de administración, control o datos con sus subtipos respectivos. En la Tabla 1 se muestra los tipos y subtipos usados para este estudio (Estrella, 2017).

Tabla 1

Tipo y subtipo tramas 802.11

Valor de Type	Tipo	Valor de Subtype	Subtipo
00	Administración	0000	Association Request
00	Administración	0001	Association Response
00	Administración	00010	Reassociation Request
00	Administración	00011	Reassociation Response
00	Administración	0100	Probe Request
00	Administración	0101	Probe Response
00	Administración	1000	Beacon
00	Administración	1011	Authentication
01	Control	1011	RTS
01	Control	1100	CTS
01	Control	1101	ACK

Nota. Tomado de *Estudio y análisis para la actualización de red WLAN de la seps utilizando tecnologías basadas en el estándar IEEE 802.11 ac,* por S. Estrella, 2017, Repositorio PUCE (http://repositorio.puce.edu.ec/bitstream/handle/22000/14425/Caso%20de%20Estudio%20San

tiago%20Estrella.pdf?sequence=2&isAllowed=y).

ToDS / FromDS. Indica si las tramas se envían o reciben al sistema de distribución, en la Tabla 2 se muestra la agrupación de estos bits. (Cano, 2012)

Tabla 2

Interpretación bits ToDS y FromDS

	ToDS = 0	**ToDS = 1**
FromDS = 0	Tramas de administración y control, las tramas de datos dentro de un IBSS.	Trama de datos de una estación en una red con infraestructura.
FromDS = 1	Trama de datos para una estación en una red con infraestructura.	Trama de datos sobre un bridge inalámbrico.

Nota. Tomado de *configuración y evaluación de redes 802.11n,* por L. Cano, 2012, UGR (http://dtstc.ugr.es/it/pfc/proyectos_realizados/downloads/Memoria2012_LuisCano.pdf).

More Fragments. Este bit tiene como finalidad señalar que existen más fragmentos de la misma trama, es decir que existen más fragmentos entrantes (Estrella, 2017).

Retry. Por medio de este bit se identifica si el fragmento es una retransmisión de un fragmento ya transmitido (Estrella, 2017).

Power Managment. Es usado por las estaciones para señalar que cambia de modo ahorro de energía a administrador de energía o viceversa (Estrella, 2017).

More Data. Se la usa por el AP para señalar que existen más tramas almacenadas en dicha estación, para decidir si usar esta información o cambiar de estado (Estrella, 2017).

WEP. Es el bit encargado de indicar si la trama tiene encriptación o no.

Como medios de seguridad dentro del estándar 802.11 se puede encontrar.

WEP (Privacidad equivalente por cable). Por medio de aircrack-ng se logró vulnerar las contraseñas WEP, además su vida útil es muy corta, se la usó para el cifrado y a 802.1x para la autenticación (Stafanick, 2018).

WPA (Acceso Inalámbrico de protección). Se la encuentra de 2 formas: 802.1X (Enterprise) la cual usa EAP (Protocolo de autenticación extensible) y PSK utiliza una clave compartida entre el AP y el radio del cliente, WPA utiliza TKIP (Protocolo de integridad de clave temporal) como medio de cifrado (Stafanick, 2018).

La ID del elemento WPA está denotado por 0x221 y es la misma que el ID del elemento del proveedor, entonces la estación deberá comprobar la OUI (Identificador Único de la Organización) para identificar si la información encriptada pertenece a WPA, es decir deberá estar establecida en 00-50-f2. En la Tabla 3 se muestra los cifrados compatibles (Holla, 2017).

Tabla 3

Lista de cifrados admitidos

OUI	Tipo de suite	Sentido
00-50-f2	0 0	Use Group Cipher Suite
00-50-f2	1	WEP-40
00-50-f2	2	TKIP
00-50-f2	3	Reservado
00-50-f2	4 4	Reservado
00-50-f2	5 5	WEP-104

Nota. Tomado de *WPA Information Element | Hitch Hiker's Guide to Learning,* por V. Holla,

2017. Hitch Hiker's Guide to Learning (http://www.hitchhikersguidetolearning.com/2017/09/17/wpa-information-element/).

Donde:

TKIP es el cifrado que pertenece a WPA.

AKM es el número de conjuntos de administración de claves de autenticación que son compatibles, estos se muestran en la Tabla 4. (Holla, 2017)

Tabla 4

Mecanismos de autenticación

OUI	Tipo de suite	sentido
00-50-f2	0 0	Reservado
00-50-f2	1	802.11x
00-50-f2	2	PSK

Nota. Tomado de *WPA Information Element | Hitch Hiker's Guide to Learning,* por V. Holla, 2017, Hitch Hiker's Guide to Learning (http://www.hitchhikersguidetolearning.com/2017/09/17/wpa-information-element/).

WPA2. Conocido también como 802.11i, es similar a WPA de igual forma tiene 2 versiones 802.1X y PSK. Para el cifrado WPA2 usa CCMP, mientras que TKIP se lo usa de forma opcional (Stafanick, 2018).

Con WPA y WPA2 se mejora a gran medida la protección de los datos y el control del acceso a las redes inalámbricas, con ello se tiene 2 formas de autenticación, PSK en la que se autentica mediante una contraseña compartida, se usa para redes pequeñas o servicios

domésticos y la segunda forma IEEE 802.1X con el protocolo EAP en la que se establece una contraseña común (Páez, 2015).

Order. Indica que la trama es enviada usando la clase de servicio estrictamente ordenada (Estrella, 2017).

Duration or ID.

- Duration: Se la utiliza para calcular el NAV.
- ID: Es de la estación para encuestas en PSM (Estrella, 2017).

Address 1.

Es la dirección de destino en la cual se especifica si:

ToDS = 0, entonces es de la estación final.

ToDS = 1 y FromDS = 0, entonces es la dirección del BSSID y si ToDS = 1 y FromDS = 1, entonces es bridge (Estrella, 2017).

Address 2.

Es la dirección del transmisor en la cual se especifica si:

FromDS = 0, entonces es de la estación fuente.

FromDS = 1 y ToDS = 0, entonces es la dirección del BSSID y si FromDS = 1 y ToDS = 1, entonces es bridge (Estrella, 2017).

Address 3.

Se especifica si:

FromDS = ToDS = 0, entonces es la dirección del BSSID.

FromDS = 0 y ToDS = 1, entonces es de la estación final.

FromDS = 1 y ToDS = 0, entonces es de la estación fuente.

FromDS = 1 y ToDS = 1, entonces es de la estación final (Estrella, 2017).

Address 4.

Es la dirección de la fuente, se establece únicamente cuando se envían tramas desde un AP a otro, FromDS = ToDS =1 (Estrella, 2017).

En la Tabla 5, se muestra un resumen de los address según sus bits ToDs y FromDS.

Tabla 5

Address según sus bits ToDS Y FromDS

ToDS	FromDS	Address 1	Address 2	Address 3	Address 4
0	0	DA	SA	BSSID	N/A
0	1	DA	BSSID	SA	N/A
1	0	BSSID	SA	DA	N/A
1	1	RA	TA	DA	SA

Nota. Tomado de *Estudio y análisis para la actualización de red WLAN de la seps utilizando tecnologías basadas en el estándar IEEE 802.11 AC,* por S. Estrella, 2017, Repositorio PUCE (http://repositorio.puce.edu.ec/bitstream/handle/22000/14425/Caso%20de%20Estudio%20Santiago%20Estrella.pdf?sequence=2&isAllowed=y).

Sequence Control.

Se utiliza para indicar el orden de los fragmentos de una misma trama, consta de dos subcampos, fragmento número y número de secuencia que definen la trama y los fragmentos numerados (Estrella, 2017).

CRC.

Almacena el control de redundancia cíclica de 32 bits (Estrella, 2017).

Analizador de paquetes de red inalámbrica

También conocido como Sniffer, es un software especializado en monitorear, capturar y analizar tramas o paquetes que circulan en una red de datos que no están necesariamente dirigidas hacia él, son los motores para los sistemas encargados de detección de intrusos (Silva, 2017).

Cabe mencionar que son herramientas de doble filo porque mientras los profesionales la usan con fines educativos, institucionales o para uso comercial también se los puede usar con fines maliciosos.

Usos del analizador de tráfico red

Uso correcto

- Transformar datos binarios que entregan las tramas en información legible.
- Resolver e identificar problemas en la red.
- Descubrir cuellos de botella por medio de un análisis de la red.
- Identificar tarjetas de red defectuosas.
- Analizar las operaciones que realizan aplicaciones específicas.
- Medio de aprendizaje de protocolos, entre otras funciones.

Uso incorrecto

- Descubrir los patrones de usuarios en una red.
- Mapeo y escaneo de la distribución de la red.
- Robo de contraseñas e información confidencial.
- Receptar mensajes de correo electrónico, entre otras.

Funcionamiento de un analizador de tráfico de red

Los programas necesitan que un computador tenga una tarjeta de red en modo monitor, que de esa forma le permite escuchar, detectar y leer todas las redes a su alrededor incluidos los paquetes que se dirijan a su dirección MAC, una vez que el analizador tiene acceso a los datos empieza a filtrar los paquetes y observa las peticiones de los puertos realizada por los usuarios (Páez, 2015).

Analizadores de paquetes de redes inalámbricas en el mercado

Nagios.

Es considerado el primer software orientado al análisis de las redes, tiene como función monitorear el estado de los equipos y los servicios de la red, de igual forma el monitoreo remoto, es una herramienta muy poderosa que permite consultar casi todos los parámetros de una red y es capaz de notificar mediante correo electrónico o SMS cualquier tipo de fallo (Gaitan, 2017).

Zabbix.

Esta herramienta tiene como fin monitorear el rendimiento y capacidad de los parámetros de una red de igual forma que Nigios notifica por medio de SMS o correo electrónico cualquier tipo de fallo, es amigable con los usuarios ya que tiene una poderosa interfaz gráfica y realiza su monitoreo mediante templates, cabe mencionar que una de las desventajas de este software es que a partir de los 1000 nodos su rendimiento disminuye (Gaitan, 2017).

Nessus.

El fin de esta herramienta es realizar escaneos de forma remota para sistemas Unix como Linux, BSB, Solaris entre otros. Se encuentra basado en plug-in(s) y GTK, realiza más de 1200 pruebas de seguridad remotas. Por otro lado, genera reportes en HTML, XML, LaTeX, y texto ASCII, adicionalmente sugiere soluciones para los problemas de seguridad (Solvetic Sistemas, 2019).

Kismet.

Encargado de analizar el tráfico de una red pueden ser ocultas o SSID que no han sido enviados, está destinado a sistemas UNIX, Windows y OSX.

Kismet trabaja en toca su capacidad con interfaces Wi-Fi, Bluetooth, SDR (radio definida por software). Posee una interfaz gráfica basada en web, dispone de código de captura remota y RAM, esto mejora y facilita el uso de equipos integrados para la captura de paquetes.

Como base indispensable para identificar los bits necesarios para ejecutar la captura de los parámetros de cada trama analizada por el programa realizado. Cabe destacar que para el presente proyecto de grado se usó Wireshark del cual se tiene una breve introducción más adelante (Solvetic Sistemas, 2019).

Herramientas usadas

Sistema Operativo GNU/Linux

Está modelado como Unix, es un sistema multitareas, multiusuarios y libre, Su desarrollo se lo ejecuto de forma altruista, tiene como núcleo a Linux, además el resto del sistema radica en

programas hechos para GNU, algunos de los sistemas operativos basados en Linux son: Ubuntu, CentOs, Debian, Fedora, OpenSUSE, Android, entre otros (Duarte & Paredes, 2016).

Para el desarrollo de este trabajo de grado se utilizó Ubuntu 18.04.4 LTS de 64 bits, dentro de una máquina virtual (VMware Workstation) versión 15.0.2 con memoria de 2GB y 1 procesador.

Aircrack

Es uno de los instrumentos más conocidos que se utiliza para crackear redes WEP/WPA/WPA2, así como generar ataques de direccionamiento y de fuerza bruta. Por lo cual es indispensable tener una tarjeta de red que se use en modo monitor. El paquete que contiene las herramientas de aircrack es aircrack-ng (Rojas et al., 2020).

- Aircrack-ng: Crackea contraseñas de redes inalámbricas.
- Aireplay-ng: Generar tráfico.
- Airodump-ng: Captura paquetes.
- Airbase-ng: Configura puntos de acceso falsos (Rojas et al., 2020).

Python

Es un lenguaje de programación de código abierto que se lo puede ejecutar en infinidad de plataformas, considerado como uno de los lenguajes de programación más amigables con el usuario, elegantes y con gran cantidad de información para su uso, es un lenguaje multiparadigma que brinda diversos estilos para trabajar como distintos tipos de programaciones: orientada a objetos, funcional, web, de igual forma utiliza niveles de organización para su programación como funciones, clases, entre otros.

Características de Python.

- Legible y elegante.

- Soporta objetos de alto nivel como string, directorios, arrays.
- Incluye librerías que tienen un sin número de clases de utilidad.
- Alta velocidad de desarrollo y excelente rendimiento.
- No requiere compilación, entre otras (Python, Python, 2020).

La versión Python usada para este proyecto es 3.7.9

Scapy

Es un programa de Python que ofrece a sus usuarios enviar, diseccionar, decodificar y falsificar paquetes en varios protocolos, gracias a esto se puede crear programas para investigar, escanear y atacar redes, puede suplir a hping, arpspoof, arp-sk, arping, p0f e incluso algunas partes de Nmap, tcpdump y tshark (Scapy, 2008).

Sockets

Son los extremos de un canal de comunicación en dos direcciones y se consiguen comunicar dentro de un proceso o con procesos externos, se los puede implementar por diversos canales como sockets de dominio UNIX, TCP, UDP, entre otros. A través de la librería de sockets se consigue clases que permiten el manejo del transporte común e interfaces para controlar todo lo pertinente (Fernandez, 2020).

Para crear un objeto tipo sockets se hace uso de la función socket().

socket_0 = socket.socket(Socket_family, Socket_type, Protocol=0) (Fernandez, 2020)

donde:

- Socket_family: Son los mecanismos de transporte como: AF_INET, PF_INET, PF_UNIX, PF_X25, entre otros (Fernandez, 2020).

- Socket_type: Identifica el tipo de comunicación, son valores constantes como SOCK_STREAM para protocolos orientados a conexiones y SOCK_DGRAM para protocolos sin conexiones (Fernandez, 2020).
- Protocol: Se la utiliza para identificar la variable de un protocolo y tipo de sockets (Fernandez, 2020).

Wireshark

Es uno de los analizadores de redes más utilizados, su función es capturar paquetes que están circulando en la red y mostrar la información que tiene cada campo de la forma más detallada, se logra su funcionamiento tanto por consola como por su interfaz gráfica, permite analizar los protocolos de red, identificar las direcciones IP y MACs, entre otros.

Además puede cargar datos o información de archivos compatibles como .pcap, pcapng, .cap y .ncf. Cabe añadir, que su instalación en Ubuntu es muy sencilla mediante terminal con el comando sudo apt-get install (Silva, 2017).

SQLite3

Permite crear y gestionar bases de datos ligeras que no ocupe tanto espacio en el disco, asimismo no necesita un proceso de servidor separado, está realizado en C y es un software libre para el uso del público en general. El programa utiliza la funcionalidad de **SQLite** a través de llamadas simples a subrutinas y funciones, esto hace que se reduzcan los tiempos de retardo en el acceso a la base de datos, gracias a que las llamadas a funciones son más eficientes que la comunicación entre procesos y permite generar bases de datos de hasta 2 Terabytes. (ochobitshacenunbyte, 2019)

Para poder hacer uso del objeto se debe como primer caso crear una conexión mediante la función connect() y consecuentemente un cursor con la función cursor() y por medio del comando execute() poder ejecutar los comando SQL (Python, Python, 2020).

En cuanto a instalar SQLite3 se lo hace por medio de terminal con los comandos: sudo apt update, sudo apt install sqlite3.

Adaptador USB inalámbrico TP-LINK-WN722N

Este adaptador tiene una antena desmontable de 4 dBi que brinda mayor potencia de señal, permite conectar una computadora una red inalámbrica con un gran acceso a internet y velocidades. Conforme con el estándar IEEE 802.11n, ofrece una velocidad inalámbrica de hasta 150 Mbps, perfecta para jugar online o reproducir vídeo en streaming. Además, permite la encriptación de la red inalámbrica con tan solo pulsar el botón QSS (Quick Setup Security) previniendo así amenazas externas (Kasa smart, 2020).

Figura 6

Adaptador USB inalámbrico TP-LINK-WN722N

Tabla 6

Especificaciones adaptador TP-LINK-WN722N

Características generales	
Estándares	IEEE 802.11n,IEEE 802.11g, IEEE 802.11b
Tasas de señal inalámbrica	11n: Up to 150Mbps
	11g: Up to 54Mbps
	11b: Up to 11Mbps
Rango de Frecuencias	2.4-2.4835GHz
Potencia de transmisión inalámbrica (EIRP)	20 dBm(MAX. EIRP)
Tipo de modulación	OFDM/CCK/16-QAM/64-QAM
Sensibilidad del receptor (sin antena)	130M: -68dBm@10% PER
	108M: -68dBm@10% PER
	54M: -68dBm@10% PER
	11M: -85dBm@8% PER
	6M: -88dBm@10% PER
	1M: -90dBm@8% PER
Modo de funcionamiento	Ad-Hoc e Infraestructura
Seguridad inalámbrica	64/128 bits WEP
	WPA/WPA2, WPA-PSK/WPA2-PSK (TKIP/AES)
Soporta sistemas operativos	Windows 7(32/64bits), Windows Vista(32/64bits), Windows XP(32/64bits), Windows 2000.

Nota. Tomado de *Adaptador USB Inalámbrico de Alta Ganancia TP-Link 150 Mbps TL-WN722N,* por HyperLink Technologies, 2020, HyperLink Technologies

(http://www.ds3comunicaciones.com/tplink/TL-WN722N.html).

Capítulo III

Diseño del aplicativo para caracterizar tramas 802.11

El objetivo principal del aplicativo es capturar e indagar tramas 802.11, así como extraer los parámetros más importantes para su estudio, es por cuanto para su desarrollo se usó Python.

El programa iniciará y se seleccionará la tarjeta de red para convertirla en modo monitor, se ingresará un tiempo de captura e iniciará el proceso, de igual forma se podrá inyectar varios tipos de tramas con una MAC propia del programa y se visualizará en tablas los datos más relevantes de cada red y sus diferentes tipos de tramas, así como un resumen porcentual de cada trama de los BSSID en toda la captura y el valor de paquetes totales de cada trama.

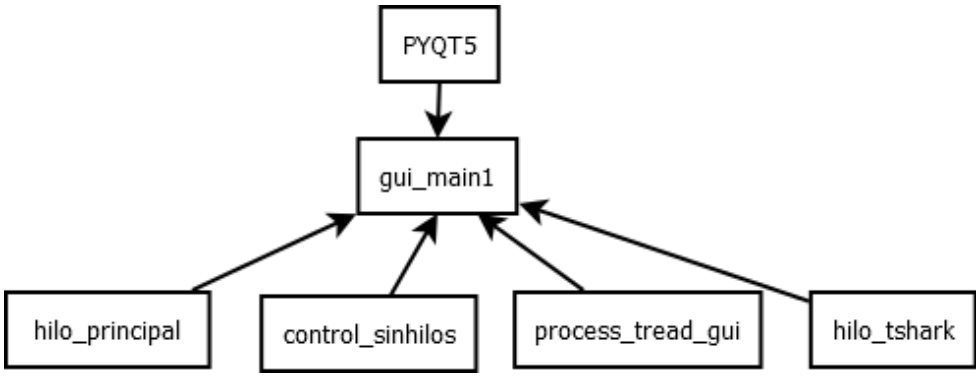
Diseño de la interfaz gráfica para la captura de tramas y procesamiento de datos

Estructura de clases del programa

El programa principal está estructurado en diferentes clases, de la misma manera se utilizan distintas librerías propias de Python.

Figura 7

Clases que conforman el programa principal gui_main1



En la Figura 7, se puede observar la estructura del programa, la clase gui_main1 hereda diferentes clases del paquete PYQT5, con el cual se puede crear objetos como tablas, botones,

líneas de edición, etiquetas, deslizadores y ventanas. De esta forma se crea una interfaz gráfica que permite al usuario interactuar con el aplicativo para realizar la captura, procesamiento y presentación de resultados de los diferentes tipos de tramas.

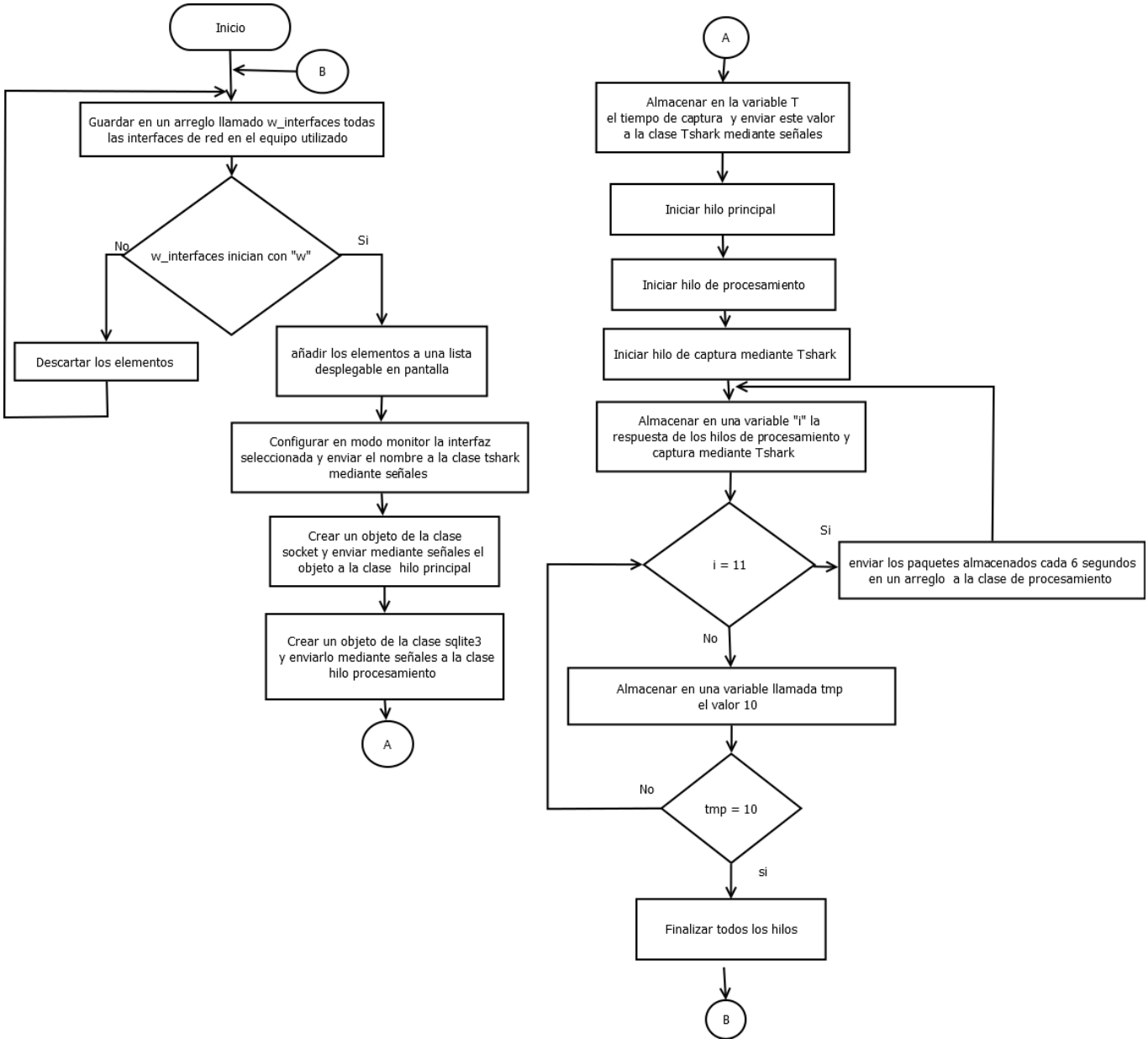
El funcionamiento del programa se logra realizando una comunicación entre clases por medio de señales ya que el flujo normal de ejecución de una clase no se interrumpe utilizando esta forma de compartir información, las mismas que se encuentran en la clase signal (para tshark, proceso principal, procesamiento, finalización de hilos y base de datos) así mismo el trabajo en paralelo de los hilos de procesamiento, tshark y de captura. De esta manera realizar un aplicativo en tiempo real, posee también un botón destinado a la inyección de paquetes y otro a presentar un resumen de los paquetes totales capturados.

La clase hilo_principal se elabora en un hilo cuando el usuario oprima el botón “iniciar captura” y haciendo uso de la librería sockets de Python captura los paquetes que se encuentren en el aire. La clase control_sinhilos se ejecuta en otro hilo y permite realizar el procesamiento de los paquetes capturados y enviar esta información a la base de datos por medio de la clase process_tread_gui, a su vez en otro hilo se ejecuta la clase hilo_tshark que da paso a realizar una captura mediante tsahrk y guardarla en un archivo .pcapng.

Diagrama de bloques del aplicativo

Figura 8

Diagrama de flujo gui_main1



En la Figura 8 se muestra el proceso que sigue el programa principal el cual llama a todas las clases creadas para lograr el funcionamiento adecuado del mismo, de igual forma se crean los objetos sockets, SQLite3 para enviarlas por medio de señales y conseguir una comunicación

completa con todas las clases creadas, así como se crea la interfaz gráfica y por medio de CSS se da el diseño deseado, habilitar, deshabilitar botones y finalizar todos los procesos e hilos cuando sea conveniente.

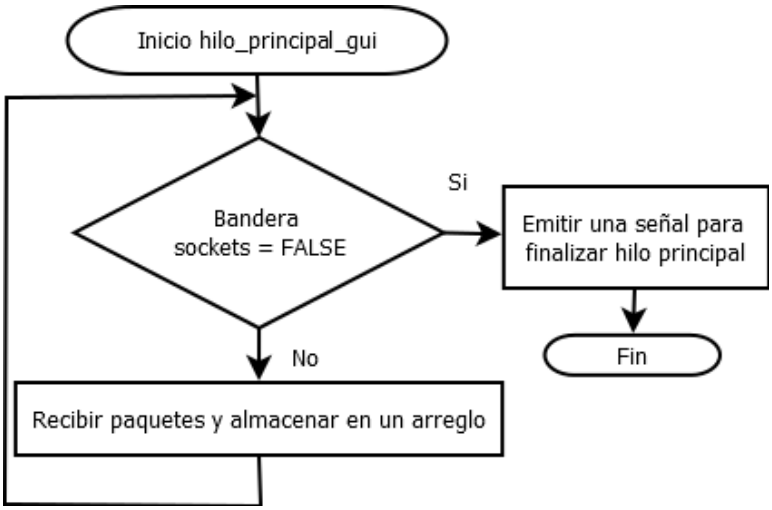
Usando os.system se logra ejecutar comandos de la terminal para eliminar los procesos como wpa-suplicant para evitar generar conflicto al momento de capturar y convertir las tarjetas ingresadas automáticamente a modo monitor.

Etapa de captura

Diagrama de flujo hilo_principal_gui.

Figura 9

Diagrama de flujo hilo principal



En la Figura 9 se muestra el proceso general de la clase hilo_principal_gui, el mismo que se encarga de la etapa de captura de paquetes, se crea un bucle con una bandera como condicional que permite identificar cuando termine el proceso de captura, si dicha bandera está en TRUE realiza la captura, caso contrario envía una señal para finalizar el hilo.

Por medio de la librería de Python sockets es posible la captura de paquetes, los cuales

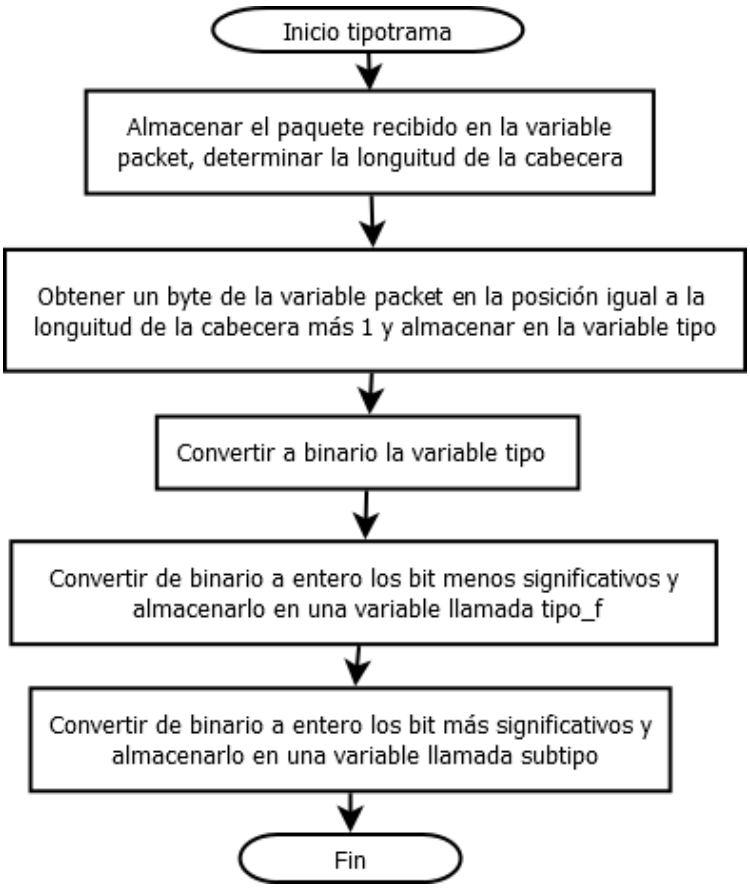
son almacenados en un arreglo y enviados a otra clase para su procesamiento.

Etapa de procesamiento

Diagrama de flujo frames_handle.

Figura 10

Diagrama de flujo clase frames_handle



Este módulo es el encargado del procesamiento de los paquetes, es decir toma los parámetros más importantes del paquete y los guarda en distintas variables para posteriormente enviarlos hacia la base de datos. Este proceso se ejecutó por medio de la identificación de los bytes de la trama, en base a la información extraída del programa Wireshark, para de este modo determinar la información a extraer.

Se inicia recibiendo el paquete enviado desde la clase control_sinhilos, donde se toma

los primeros bytes y mediante la función propia de Python unpack() permite agruparlos y posteriormente realizar conversiones de binario a entero para obtener la información deseada (header length y la potencia).

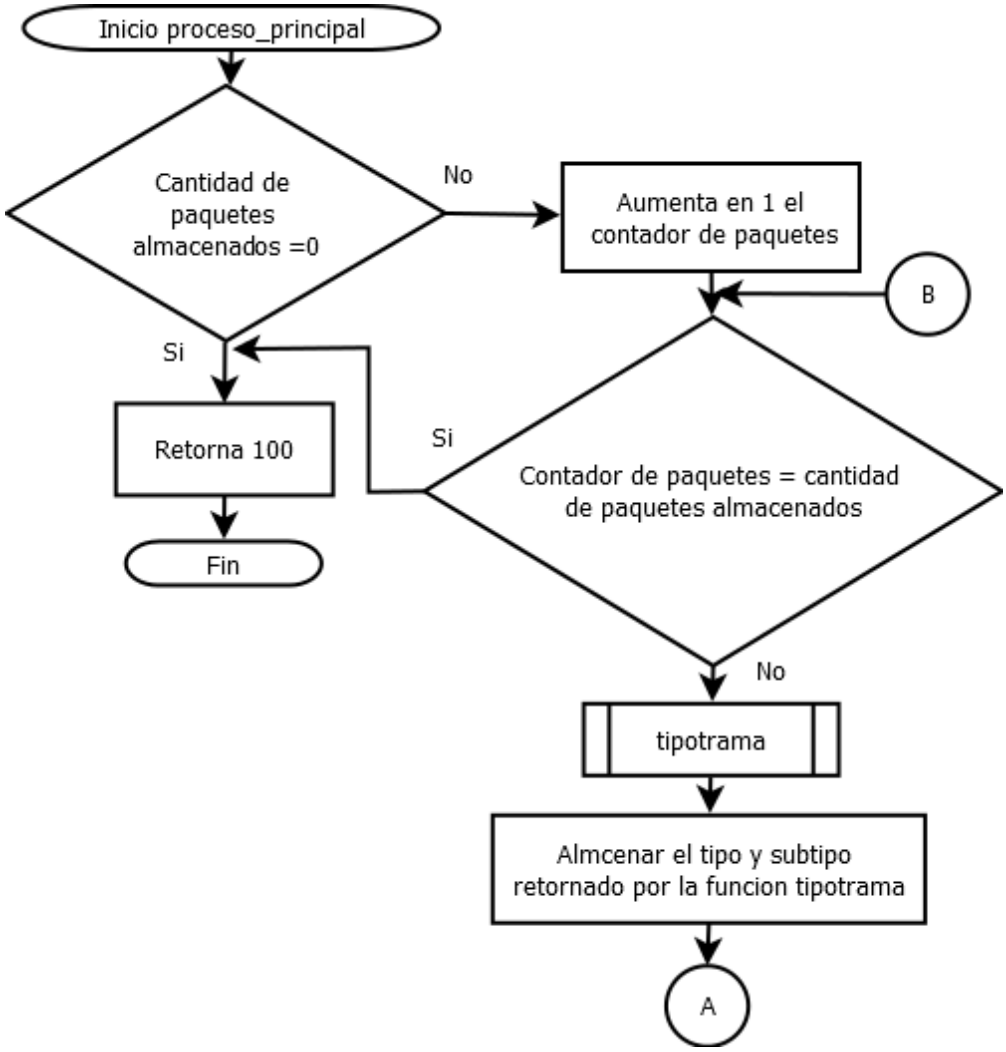
Con la función tipotrama() se fija los bytes apropiados a la variable tipo y se la convierte a binario. De este dato obtenido se toma los bits menos significativos y se los convierte a decimal para guardarlos en la variable tipotrama, de igual forma los bits más significativos se los convierte a decimal que representarían al subtipo.

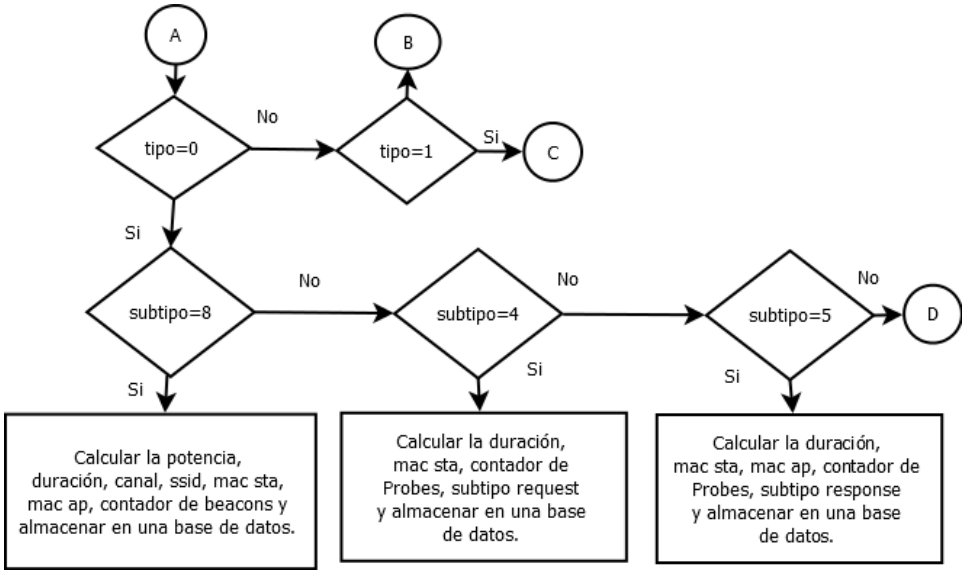
Continuando con la secuencia de identificación de byte en bytes, se obtiene las direcciones MACs, el SSID, frecuencia, canal y duración.

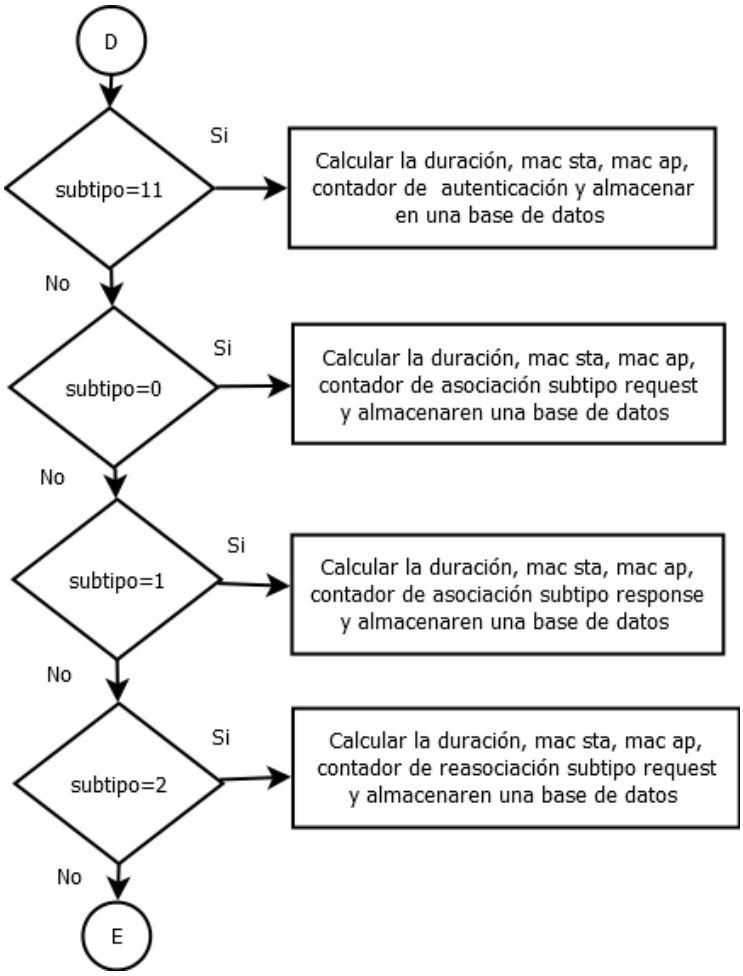
Diagrama de flujo control_sinhilos.

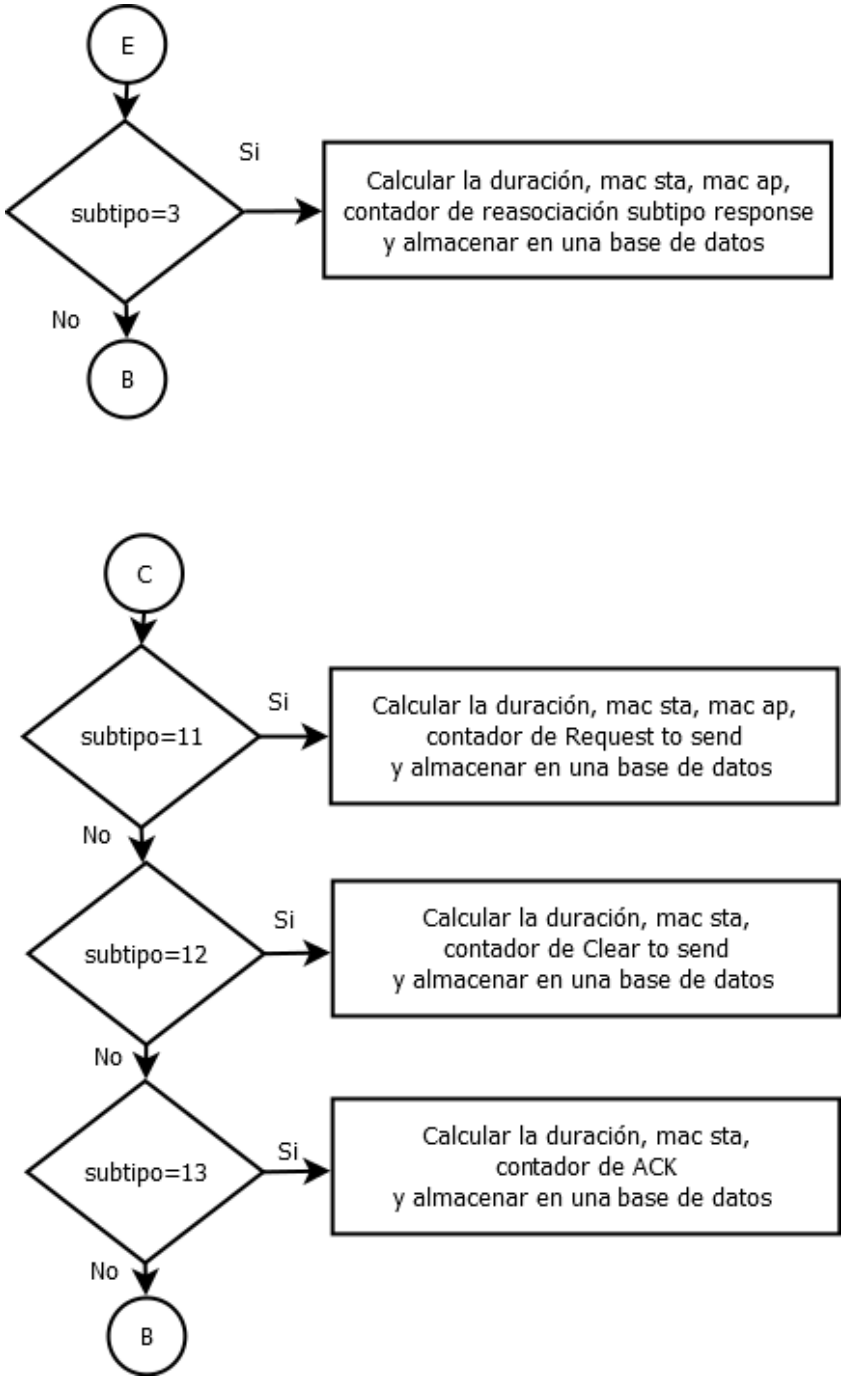
Figura 11

Diagrama de flujo clase control_sinhilos




A
B
tipo=0
No
tipo=1
Si
C
Si
subtipo=8
No
subtipo=4
No
subtipo=5
No
D
Si
Si
Si
Calcular la potencia, duración, canal, ssid, mac sta, mac ap, contador de beacons y almacenar en una base de datos.
Calcular la duración, mac sta, contador de Probes, subtipo request y almacenar en una base de datos.
Calcular la duración, mac sta, mac ap, contador de Probes, subtipo response y almacenar en una base de datos.


D
subtipo=11
Si
Calcular la duración, mac sta, mac ap, contador de autenticación y almacenar en una base de datos
No
subtipo=0
Si
Calcular la duración, mac sta, mac ap, contador de asociación subtipo request y almacenaren una base de datos
No
subtipo=1
Si
Calcular la duración, mac sta, mac ap, contador de asociación subtipo response y almacenaren una base de datos
No
subtipo=2
Si
Calcular la duración, mac sta, mac ap, contador de reasociación subtipo request y almacenaren una base de datos
No
E



En esta clase se encuentra la función proceso_principal, se logra identificar por medio de una bandera la continuación o finalización del proceso, además tiene como parámetros los paquetes recibidos y con los objetos dbase y cursor de la base de datos sirven para comunicarse y enviar los resultados del procesamiento a las tablas creadas en la base de datos.

En la Figura 11 Se muestra la secuencia que se sigue para la clasificación de cada tipo de trama. En primera instancia identifica la cantidad de paquetes almacenados, si existen aumenta un contador y lo compara con la longitud de dichos paquetes, si estos son iguales finaliza el proceso de lo contrario inicia con la clasificación, para ello se llama a la función tipotrama, almacena el tipo y subtipo de cada trama capturada para ordenarlas en tipo administración y de control, como se muestra en la Tabla 7.

Tabla 7

Tipo y subtipo de tramas

Trama	**Tipo**	**Subtipo**
Association Response	0	1
Reassociation Request	0	2
Reassociation Response	0	3
Probe Request	0	4
Probe Response	0	5
Association Request	0	7
Beacon	0	8
Authentication	0	11
RTS	1	11
CTS	1	12
ACK	1	13

Nota. Tomado de *ESTUDIO Y ANÁLISIS PARA LA ACTUALIZACIÓN DE RED WLAN DE LA SEPS UTILIZANDO TECNOLOGÍAS BASADAS EN EL ESTÁNDAR IEEE 802.11 AC,* por S. Estrella, 2017, Repositorio PUCE

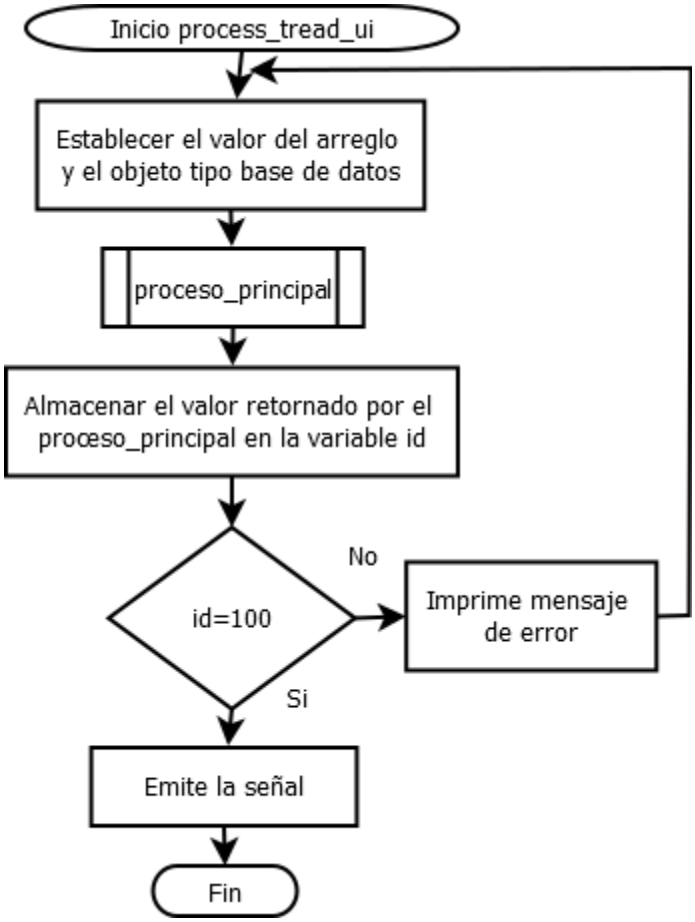
(http://repositorio.puce.edu.ec/bitstream/handle/22000/14425/Caso%20de%20Estudio%20Santiago%20Estrella.pdf?sequence=2&isAllowed=y).

Dentro de cada uno de estos tipos de tramas se llama a las funciones de la clase frames_handle donde se realiza el procesamiento de los datos, de igual forma se llama a la función tipoencriptacion, el cual realizando filtros en el OUI del cifrado y se consiguen los tipos de encriptación y a la función Update_Add de la clase sqlite3_python para almacenar la información obtenida en la base de datos.

Diagrama de flujo process_tread_ui.

Figura 12

Diagrama de flujo clase process_tread_ui

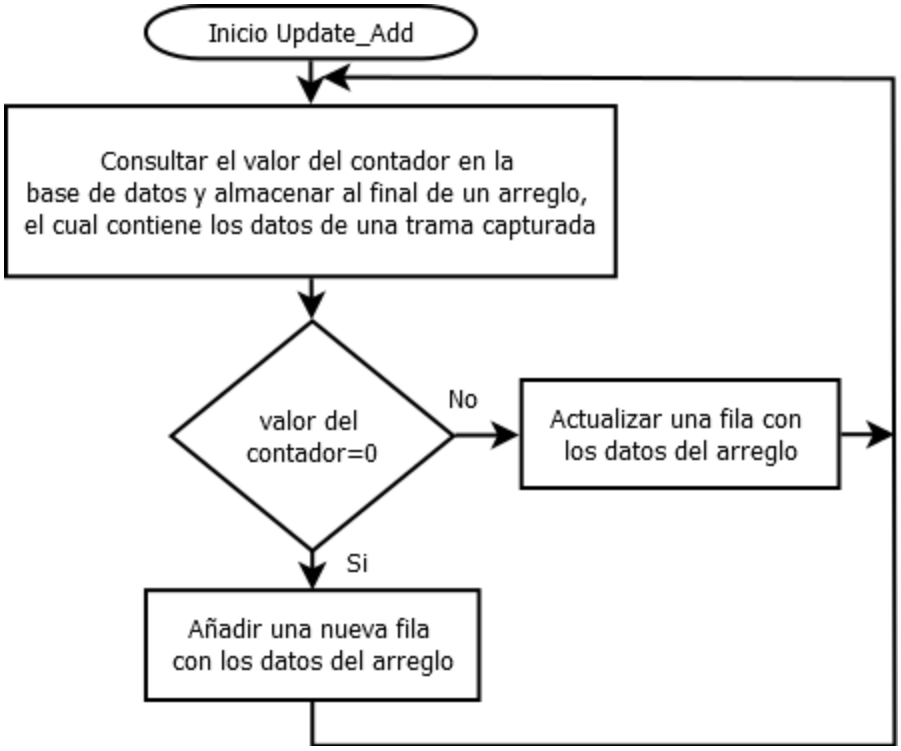


La función de este módulo es crear un vínculo hacia la base de datos emitiéndole una señal para enviar los valores obtenidos en el procesamiento realizado en la función proceso_principal del módulo control_sinhilos.

Diagrama de flujo sqlite3_python.

Figura 13

Diagrama de flujo clase sqlite3_python



Por medio de esta clase se puedo poner en forma automática la ubicación del archivo .db realizado en SQLite3, se diseña la base de datos usando comandos propios de SQL con los nombres de las tablas, los campos a añadir y los objetos tipo base de datos para poder manipular las bases.

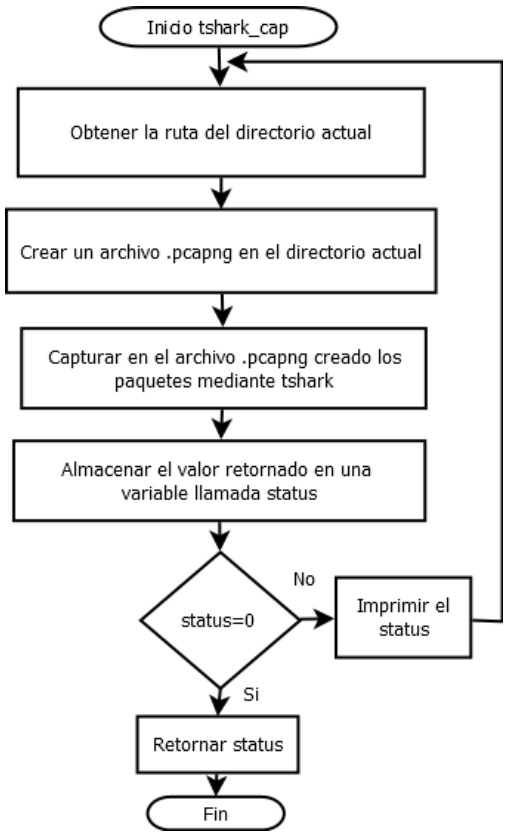
Se realiza el proceso a añadir, borrar individualmente las tablas, borrar por completo la base de datos y actualizar los registros, por ultimo escoger los parámetros adecuados para la visualización de las tablas en la interfaz gráfica.

Etapa tshark

Diagrama de flujo tshark_sniff.

Figura 14

Diagrama de flujo clase tshark_sniff

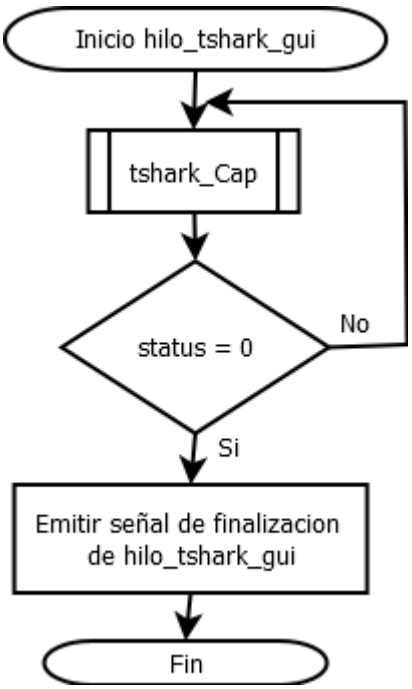


En la Figura 14 se observa la secuencia que cumple la función tshark_Cap que se encuentra en la clase tshark_sniff, que tiene como finalidad crear el archivo .pcapng de forma automática en la ruta en la que se vaya a utilizar el software y realizar la captura de paquetes mediante tshark que tenga la misma duración y capture con la interfaz que utiliza el programa desarrollado, a su vez guardar la información en el archivo .pcapng creado.

Diagrama de flujo hilo_tshark_gui.

Figura 15

Diagrama de flujo clase hilo_tshark



En esta clase se establece los valores que tendrá por defecto el programa, si el usuario no escoge una interfaz en modo monitor y no se ingresa un tiempo de captura, dicho tiempo será de 4 horas en las que el usuario puede finalizar el proceso cuando crea conveniente siempre y cuando escoja una interfaz.

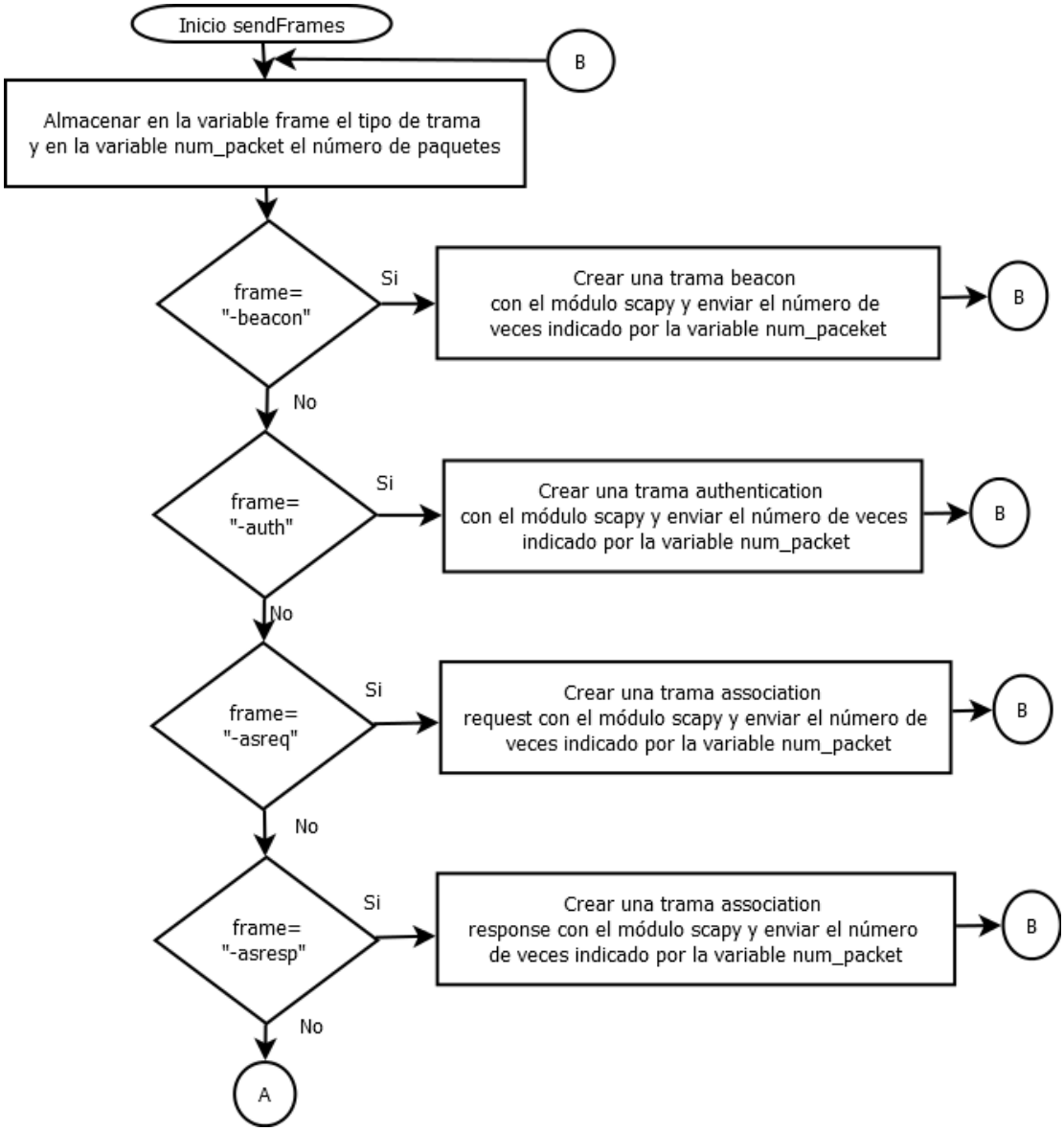
Utiliza la función tshark_Cap de la clase tshark_sniff, para realizar la captura por medio de tshark. Por último, se establece la variable status para identificar el momento que se envía la señal de finalización de la captura.

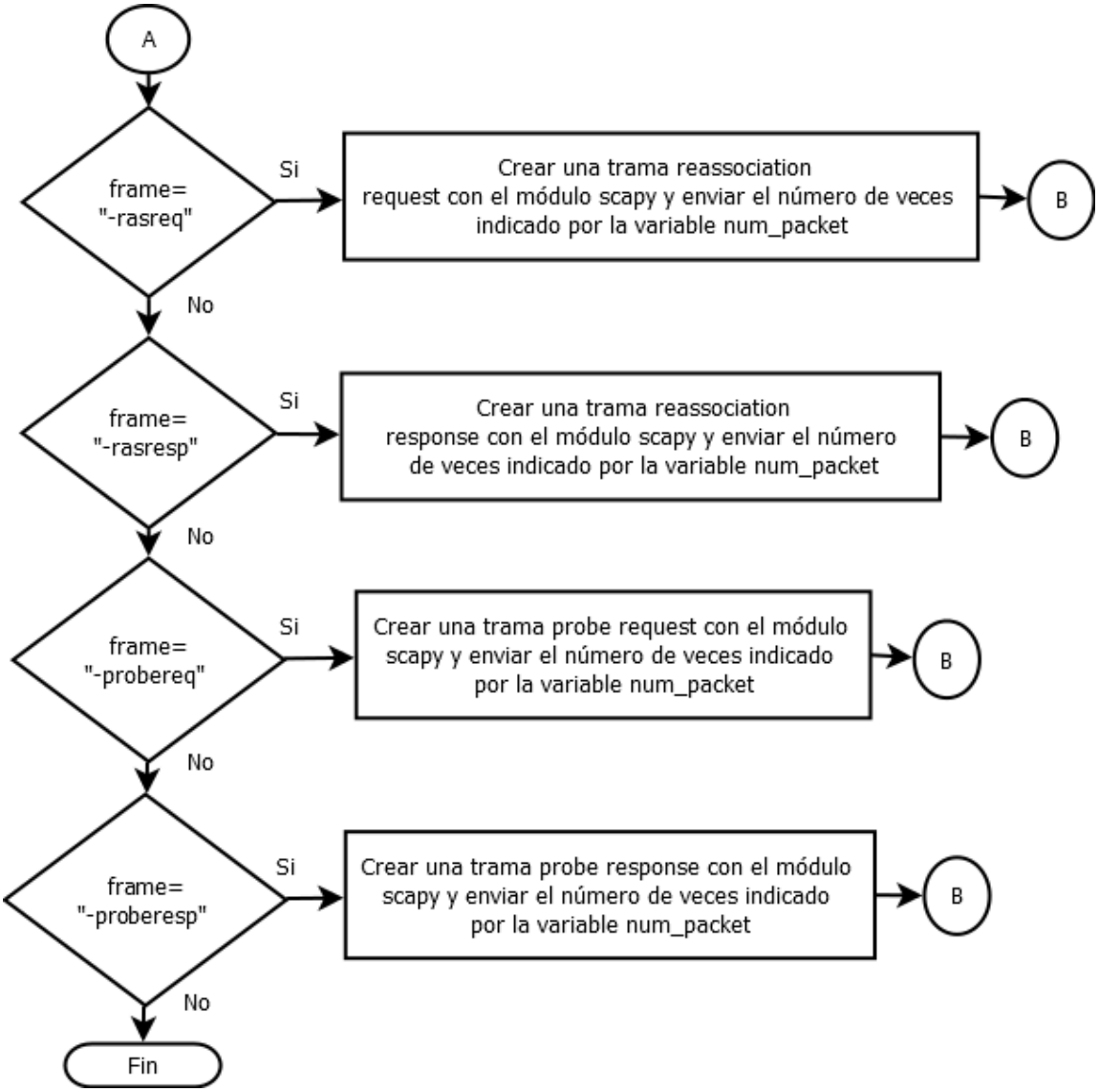
Etapa de inyección

Diagrama de flujo inject.

Figura 16

Diagrama de flujo clase inject


Inicio sendFrames
B
Almacenar en la variable frame el tipo de trama
y en la variable num_packet el número de paquetes
frame=
"-beacon"
Si
Crear una trama beacon
con el módulo scapy y enviar el número de
veces indicado por la variable num_paceket
B
No
frame=
"-auth"
Si
Crear una trama authentication
con el módulo scapy y enviar el número de veces
indicado por la variable num_packet
B
No
frame=
"-asreq"
Si
Crear una trama association
request con el módulo scapy y enviar el número de
veces indicado por la variable num_packet
B
No
frame=
"-asresp"
Si
Crear una trama association
response con el módulo scapy y enviar el número
de veces indicado por la variable num_packet
B
No
A



Por medio de esta clase se establece las direcciones MAC que utilizará el software cuando inyecte tramas de los diferentes tipos, así como el SSID y la cantidad de paquetes que se desea enviar.

MAC AP: dc:d2:fc:64:3a:11

MAC Estación: dc:d2:fc:64:3a:12

SSID: Test-Network

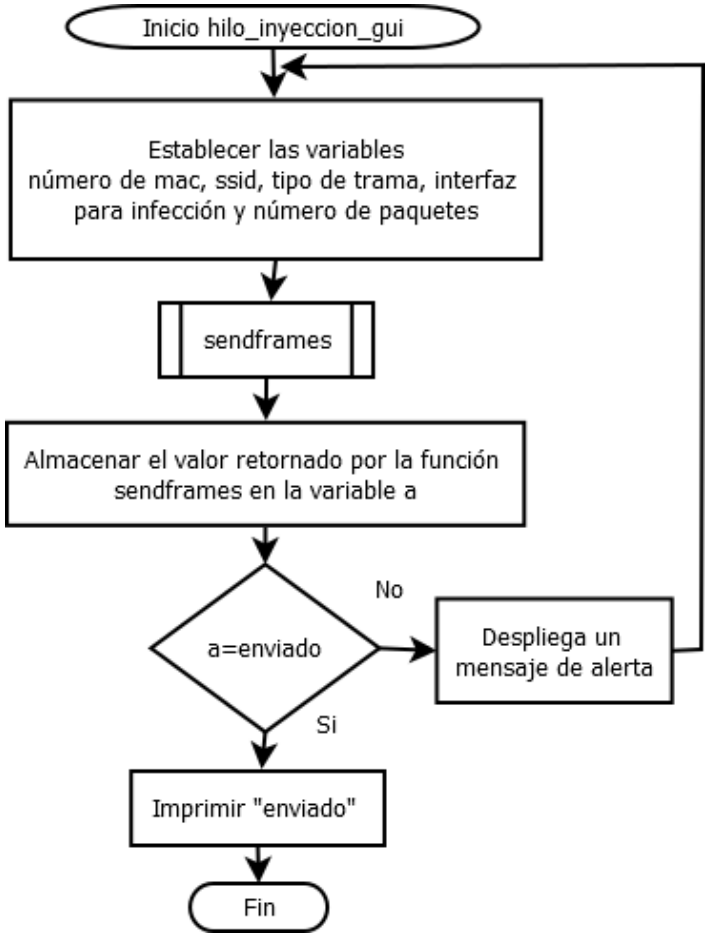
Con la función sendFrames de esta clase y por medio de la librería Scapy se construye

por capas cada paquete teniendo en cuenta el BSSID, tipo y subtipo de trama, las direcciones MAC y a su vez se los envía a través de sockets.

Diagrama de flujo hilo_inyeccion_gui.

Figura 17

Diagrama de flujo clase hilo_inyeccion_gui



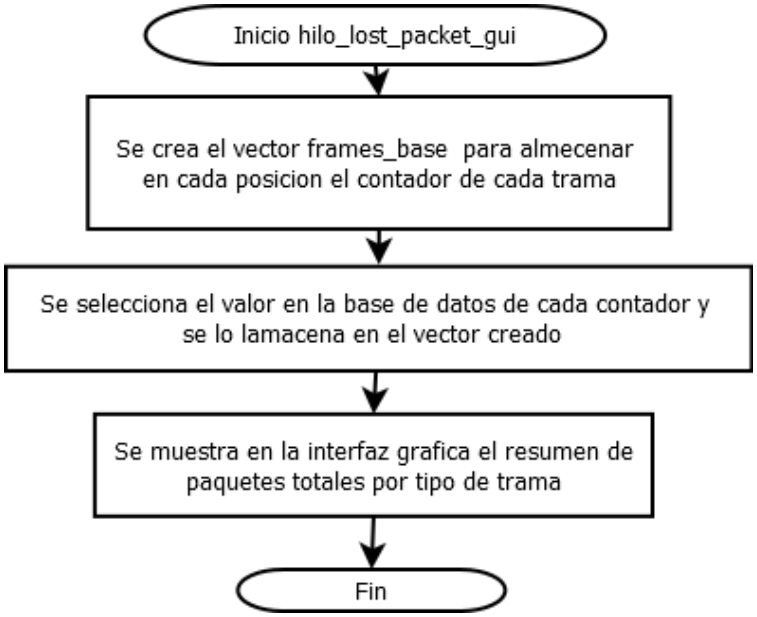
En la Figura 17 se muestra el proceso a seguir para la inyección de paquetes en el que ya creada la interfaz gráfico y por medio de la función sendFrames de la clase inject permite enviar y capturar el tipo de trama seleccionado, si este paquete se envió correctamente se muestra en la interfaz gráfica un mensaje con información de la dirección MAC del paquete enviado, caso contrario se despliega una ventana de alerta, si no se escoge un tipo de trama y no se ingresa el

número de paquetes el programa por defecto enviaría 10 Tramas Beacons.

Diagrama de flujo hilo_lost_packet_gui.

Figura 18

Diagrama de flujo clase hilo_lost_packet_gui



De igual forma se desarrolla la interfaz gráfica de la ventana Paquetes Totales, en la que muestra un resumen con el contador total de cada tipo de trama, seleccionando de la base de datos el valor del contador de cada trama para mostrarlo por pantalla.

Funcionamiento del aplicativo para captura de tramas 802.11

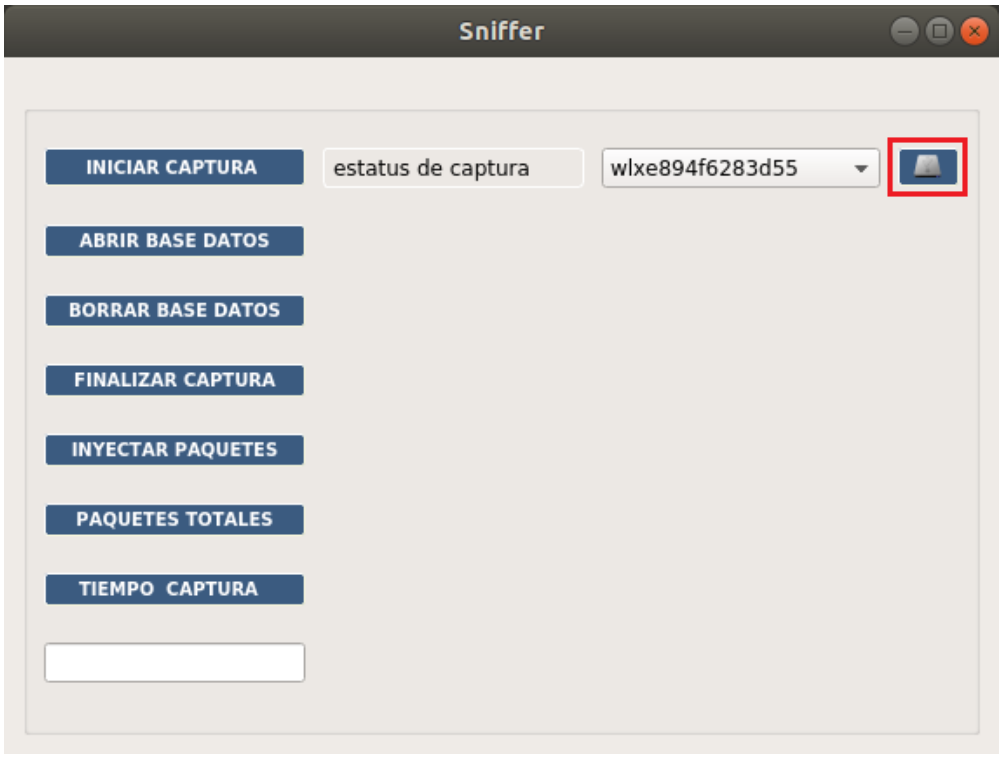
Para realizar la caracterización de paquetes se utilizó el programa creado en Python, como se observa en la Figura 19 en la cual, al lado superior derecho se encuentra una lista desplegable que presenta las tarjetas de red conectadas al computador. En primera instancia si el ordenador no tiene conectado ninguna tarjeta de red y se abre el aplicativo la lista desplegable saldrá vacía, es por cuanto al momento de ingresar una tarjeta se oprime el botón señalado en el recuadro rojo para actualizar la lista, por otro lado, si se conectan más tarjetas

solamente se presiona la lista desplegable para su actualización, al momento de cerrar el programa las tarjetas de red vuelven a estado manager.

Al seleccionar una de las tarjetas de red se convierte automáticamente en modo monitor, si la tarjeta se encuentre ya en modo monitor solo se selecciona y continua con la captura.

Figura 19

Interfaz Gráfica, ventana principal del aplicativo

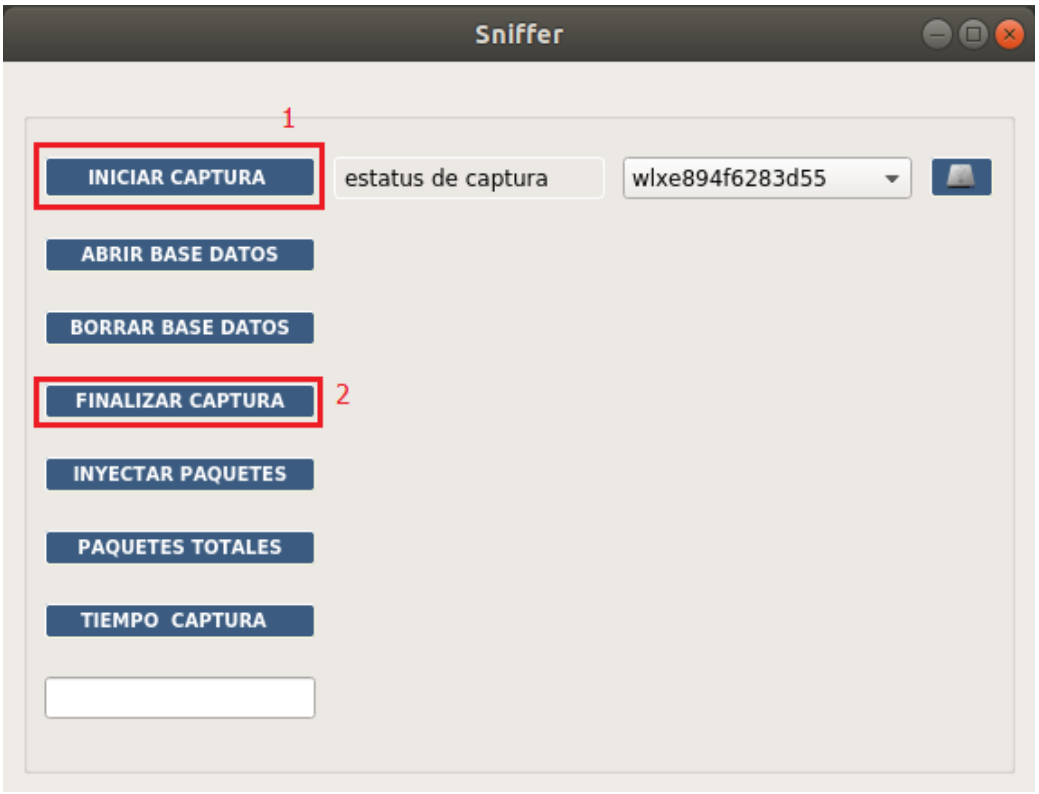


La captura de paquetes se la puede realizar de 2 formas:

- Sin tiempo definido de captura, donde el usuario finaliza el proceso cuando crea conveniente seleccionando los botones señalados en los recuadros rojos, como se muestra en la Figura 20.

Figura 20

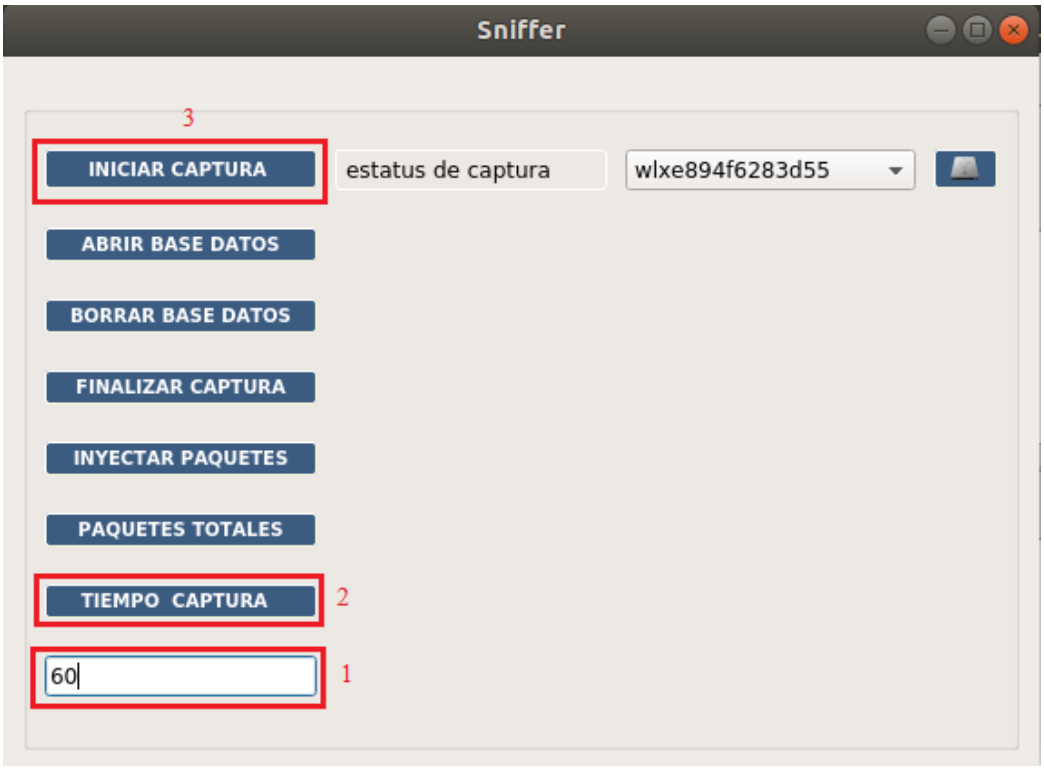
Capturar sin tiempo ingresado



- Con tiempo de captura, donde el usuario escribe el tiempo total de captura, a continuación, presiona el botón TIEMPO CAPTURA para su confirmación e INICIAR CAPTURA y espera que finalice el tiempo ingresado, como se observa en la Figura 21.

Figura 21

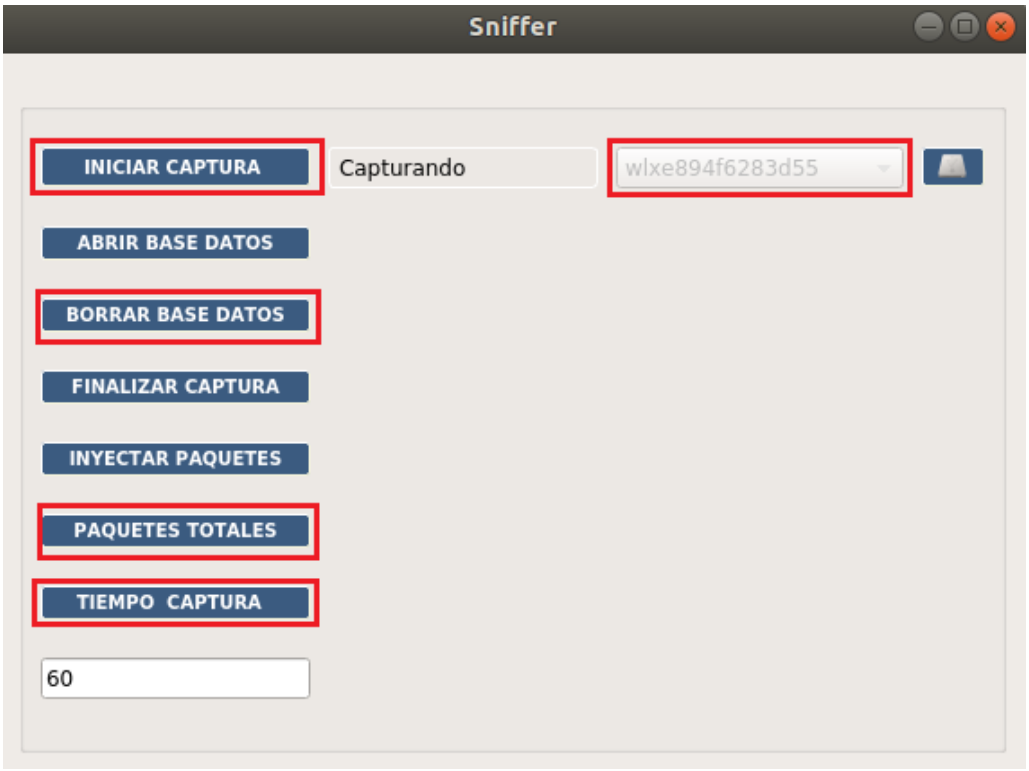
Capturar con tiempo ingresado



En los 2 casos durante la captura de paquetes se bloquean algunos botones como se muestra en la Figura 22 con el objetivo de no causar conflicto en el programa y mantener un orden lógico en el proceso.

Figura 22

Bloqueo de botones



Al presionar el botón ABRIR BASE DATOS se abre una nueva ventana Figura 23. En la parte superior se encuentra una lista desplegable con los tipos de tramas y resumen, al seleccionar cualquiera de ellos se observan tablas con la información más relevante de cada uno, así como un resumen porcentual de cada trama, de igual forma se tiene las opciones de eliminar una fila de la tabla, borrar la tabla seleccionada y cerrar la ventana.

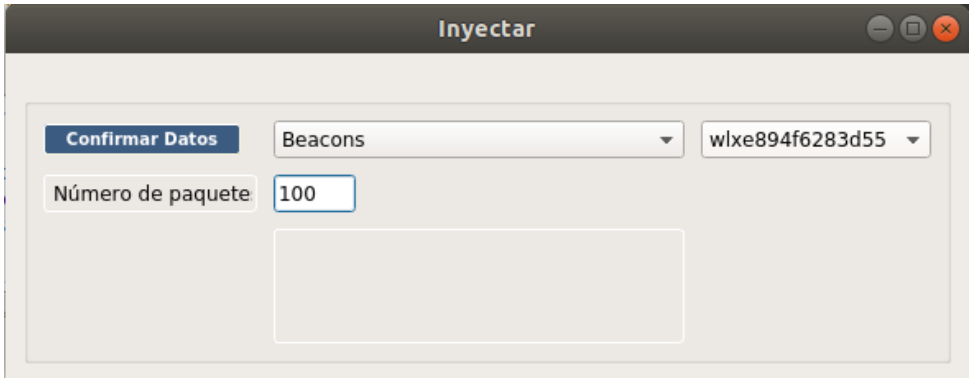
Figura 23

Ventana base de datos

Al regresar a la ventana principal del programa se encuentra el botón INYECTAR PAQUETES, el mismo que abre una nueva ventana en la que se escoge la tarjeta inalámbrica con la cual se va a realizar la inyección, se elige en una lista el tipo de trama y se escribe la cantidad de paquetes a inyectar.

Figura 24

Ventana inyectar



Ya culminada la captura de paquetes se puede eliminar por completo los valores almacenados en la base de datos, así como observar un resumen por medio del botón

PAQUETES TOTALES, como su nombre lo indica muestra la totalidad de paquetes capturados en todo el proceso y los divide por tipo de trama.

Figura 25

Ventana paquetes totales

Capítulo IV

Captura y análisis comparativo de los resultados obtenidos

El presente capítulo tiene como fin plasmar el escenario de pruebas y la ejecución del aplicativo para caracterizar tramas 802.11, además exponer las características del mismo y conseguir un análisis comparativo del software desarrollado y Wireshark.

Proceso de captura

Dicho proceso consiste en recoger todos los datos necesarios para un análisis posterior, el método de captura implica la ejecución del software desarrollado, utilizando el adaptador inalámbrico TP-LINK TL-WN722N ambas en un computador.

La ejecución del software se lo realizó dentro de un ambiente doméstico en el Conjunto habitacional Jardín Sur, el escenario es favorable ya que se encontrarán diversos puntos de acceso distribuidos en toda el área. Las capturas fueron realizadas durante dos semanas de lunes a domingo, por la mañana y por la noche, en cada uno de estos horarios se realizaron capturas de 10 minutos, 40 minutos y 1h30, con el fin de ampliar la recolección de datos.

Análisis de resultados

Con los datos obtenidos de las capturas realizadas dentro de los 14 días, se ejecutó un análisis comparativo con Wireshark para conocer la eficiencia del programa desarrollado mediante la identificación del error obtenido de paquetes perdidos por tipo de trama y el error de pérdidas de paquetes inyectados propios del software, así como saber los fabricantes de los APs, canales, encriptación, entre otros.

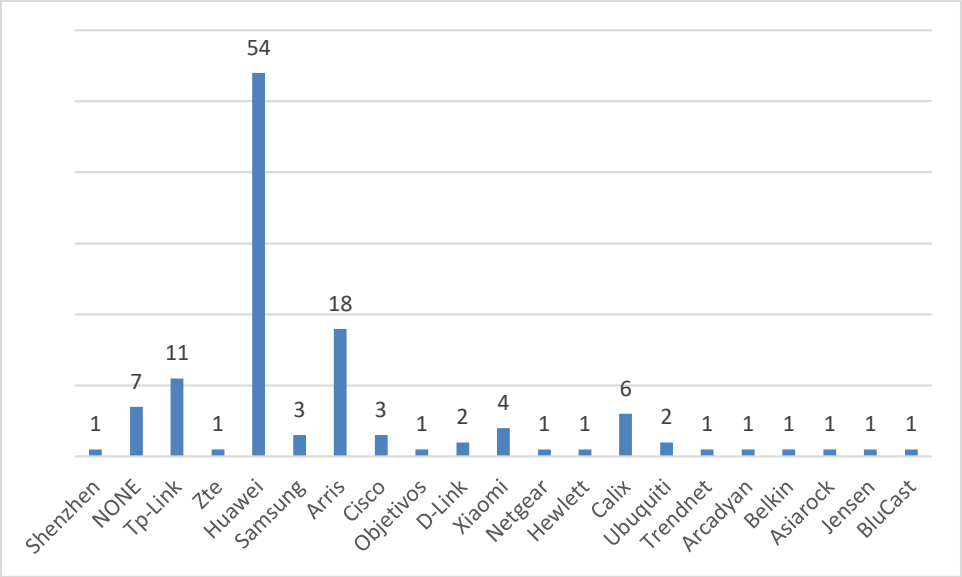
En la etapa de captura se localizó un total de 121.

Fabricantes

Se encontraron 21 marcas diferentes de fabricaste de APs, donde los más utilizados son Huawei con 54 equipos y Arris con 18, por otro lado 7 equipos que no se pudieron reconocer su fabricante.

Figura 26

Fabricantes de AP



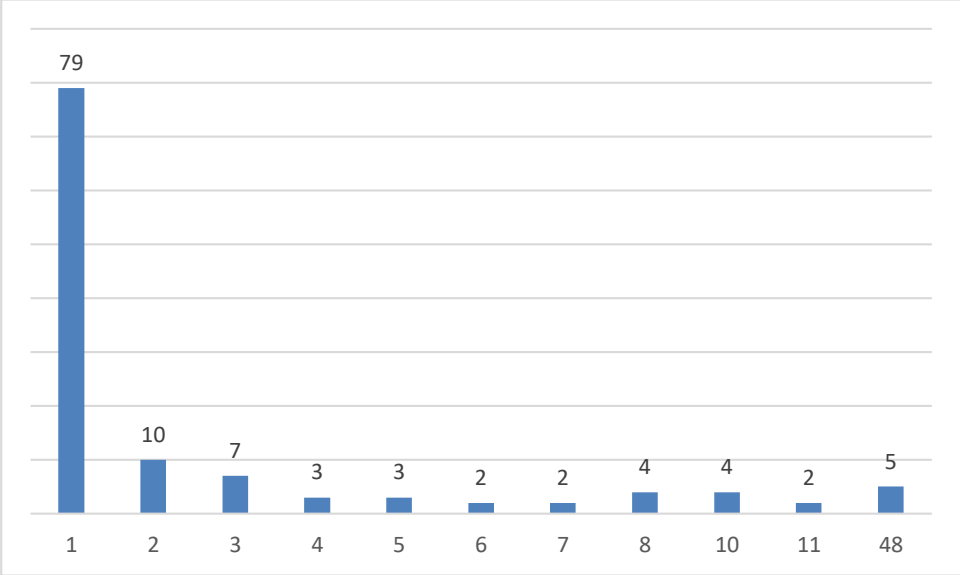
Canales

Para evitar cualquier tipo de interferencia se recomienda configurar los equipos en los canales 1, 6, 11. Ya que estos no tienden a producir solapamiento entre ellos, mientras que si con sus canales contiguos.

Como se observa en la Figura 27, 79 de los 121 APs están configurados en el canal 1, 2 en el canal 6 y 2 en el canal 11, el resto de equipos se encuentran repartidos en los canales 2, 3, 4, 5, 7, 8, 10 y 5 equipos en el canal 48 de la banda de 5GHz.

Figura 27

Canales usados

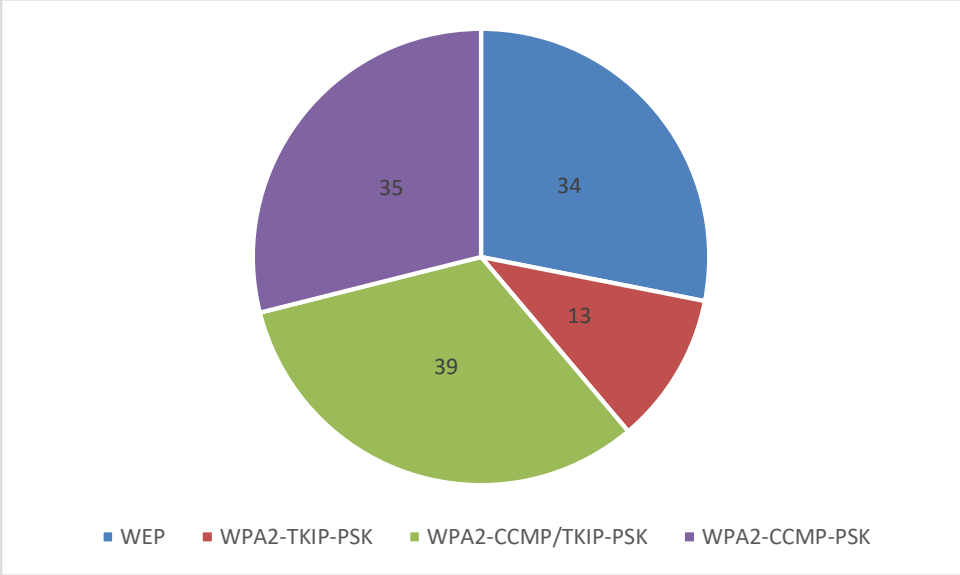


Seguridad

En la Figura 28, se identifica el tipo de encriptación que poseen los equipos detectados, donde a pesar de ser un mecanismo de seguridad no fiable, fácil de vulnerar y más antiguo, se tienen 34 equipos que usan WEP, 13 equipos usan WPA2-TKIP que permite la compatibilidad con equipos más antiguos, mientras que los demás dispositivo usan mecanismos más robustos y modernos, con sistemas más avanzados y seguros como WPA2-CCMP que se lo recomienda con más frecuencias con 35 equipos y 39 redes donde los routers permiten una combinación de WPA2-CCPM-TKIP.

Figura 28

Encriptación usada



Eficiencia

Se calculó el error de cada tipo de trama, teniendo en cuanta que el dato real es el proporcionado por Wireshark y el dato simulado sería del programa desarrollado, de los 14 días de toma de datos se escogió un de la mañana y otro de la tarde ambos de 40 minutos.

A continuación, se muestran tablas de cada tipo de trama con los errores calculados por día y el promedio de cada una de estas, es decir el porcentaje de erro de las tramas perdidos, con el objetivo de identificar la eficiencia del programa.

Tramas de control.

Request to Send.

Tabla 8

Porcentaje de pérdidas RTS

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	3.69	5.23	0.41	1.54	0.86	4.02	3.70	4.43	2.93	0.81	1.22	1.97	1.83	2.01	2.47
Pm	1.75	1.76	0.78	1.55	1.24	1.20	0.92	0.52	2.04	1.00	1.31	1.59	1.62	1.42	1.26

Clear to Send.

Tabla 9

Porcentaje de pérdidas CTS

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	1.94	1.67	0.82	1.5	1.6	1.01	1.23	2.08	1.28	1.45	1.29	1.53	1.27	1.44	1.43
Pm	1.40	1.52	0.80	1.33	2.03	1.11	1.25	0.25	1.45	0.68	3.68	1.22	1.23	1.75	1.4

Acknowledgement.

Tabla 10

Porcentaje de pérdidas ACK

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	1.79	3.15	4.46	0.49	0.81	3.90	2.93	1.36	1.97	2.18	2.10	1.57	2.34	2.56	2.25
Pm	1.64	1.56	1.45	1.22	1.51	1.33	1.21	0.47	1.10	0.95	0.68	1.44	1.26	1.22	1.21

Tramas de gestión.

Beacons.

Tabla 11

Porcentaje de pérdidas Beacons

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	2.36	2.76	3.17	1.89	2.68	2.10	2.78	2.24	2.48	2.55	2.30	2.44	3.09	2.29	2.50
Pm	1.57	1.65	1.54	1.20	1.72	1.43	1.27	0.61	1.23	1.23	1.24	1.24	0.90	1.22	1.11

Probes.

Tabla 12

Porcentaje de pérdidas Probes

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	6.08	2.31	3.68	1.17	1.11	1.44	2.81	0.47	1.24	2.30	3.42	2.29	3.57	1.67	2.39
Pm	1.07	5.76	5.06	1.25	6.53	4.43	1.55	1.92	3.47	0.57	0.42	0.89	2.73	2.47	2.72

Association.

Tabla 13

Porcentaje de pérdidas Association

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Reassocciation.

Tabla 14

Porcentaje de pérdidas Reassociation

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Authentication.

Tabla 15

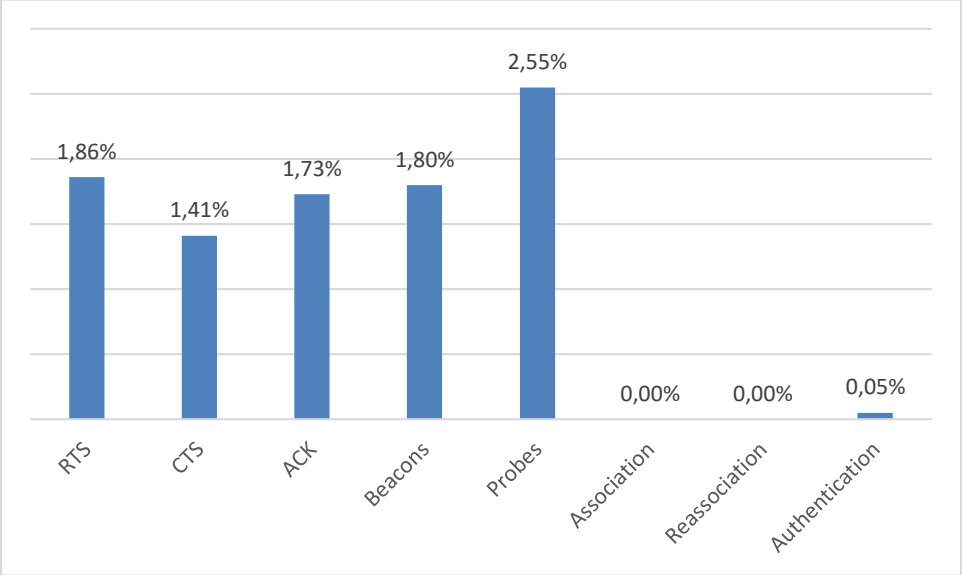
Porcentaje de pérdidas Authentication

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pm	0	0	0	0	0	1.51	0	0	0	0	0	0	0	0	0.10

En la Figura 29, se observa el porcentaje final de error que se obtuvo por cada tipo de trama durante los 14 días de prueba, tomando en cuenta las capturas realizadas por los paquetes inyectados. El error más elevado es el de tramas Probes con 2.55%, es una de las tramas que se envía en tasas de trasferencias bajas con el fin de que todos los dispositivos la escuchen y obtener una respuesta, al enviar grandes cantidades de esta afecta al rendimiento del canal.

Figura 29

Porcentaje de pérdidas por paquetes



Tramas inyectadas.

Se inyectaron tramas de administración desde el programa desarrollado con SSID Test-Network y MAC DC:D2:FC:64:3A:11 y DC:D2:FC:64:3A:12 de estación y cliente respectivamente, la inyección se la realizó aleatoriamente, diferentes cantidades como 200, 400 o 600 paquetes y por medio de una sola tarjeta inalámbrica tanto para inyectar como para capturar.

Beacons.

Tabla 16

Porcentaje de pérdidas Beacons inyectadas

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pm	0	0	0	0	0	0	0	0	0	0	0	0	0	5.83	0.41

Probe Request.

Tabla 17

Porcentaje de pérdidas Probe Request inyectadas

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	0	0	0	0	1.10	0	1.27	0.93	1.73	1.38	0	0	2.41	0	0.63
Pm	2.74	0	3.5	1.96	0	0	0	0	0	0	0	0	0	0	0.58

Probe Response.

Tabla 18

Porcentaje de pérdidas Probe Response inyectadas

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pm	0	0	0	0	0	0	0	0	0	0	3.62	0	0	0	0.25

Association Request.

Tabla 19

Porcentaje de pérdidas Association Request inyectadas

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pm	1.68	0	0	0	0	0	0	0	0	0	0	0	0	0	0.12

Association Response.

Tabla 20

Porcentaje de pérdidas Association Response inyectadas

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pm	2.09	0	0	0	0	0	0	0	2.13	0	0	0	0	0	0.30

Reassociation Request.

Tabla 21

Porcentaje de pérdidas Reassociation Request inyectadas

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pm	0	0	0	2.83	0	0	0	0	0	0	0	0	0	0	0.20

Reassociation Response.

Tabla 22

Porcentaje de pérdidas Reassociation Response inyectadas

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Authentication.

Tabla 23

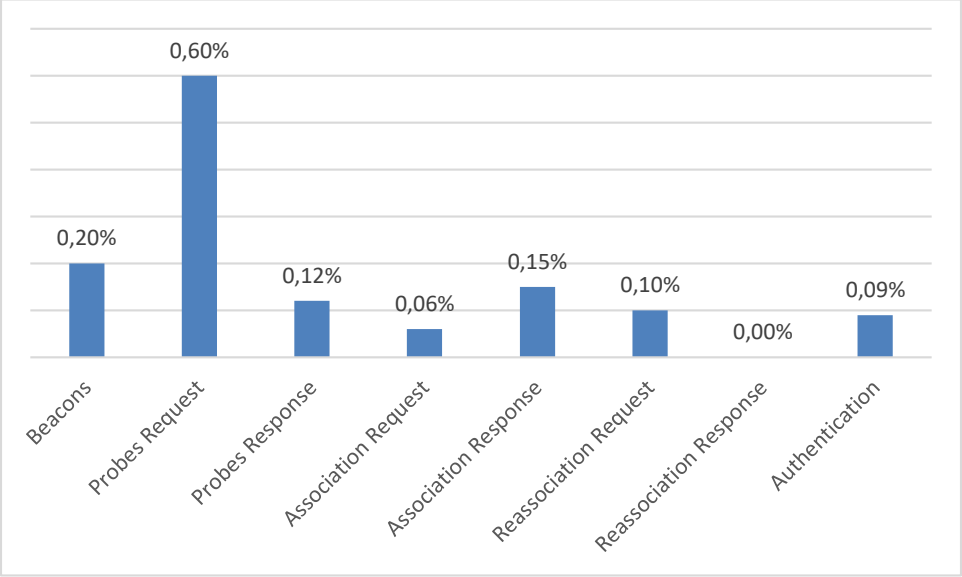
Porcentaje de pérdidas Authentication inyectadas

Día	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Promedio
Am	0	0	0	0	0	0	0	0	0	0	0	0	2.54	0	0.18
Pm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Como se identifica en la Figura 30, de la misma forma que en las tramas capturas normalmente el tipo de trama con más error es Probe Request en este caso de 0.60% durante los 14 días de tomas de datos.

Figura 30

Porcentaje de paquetes perdidos tramas inyectadas

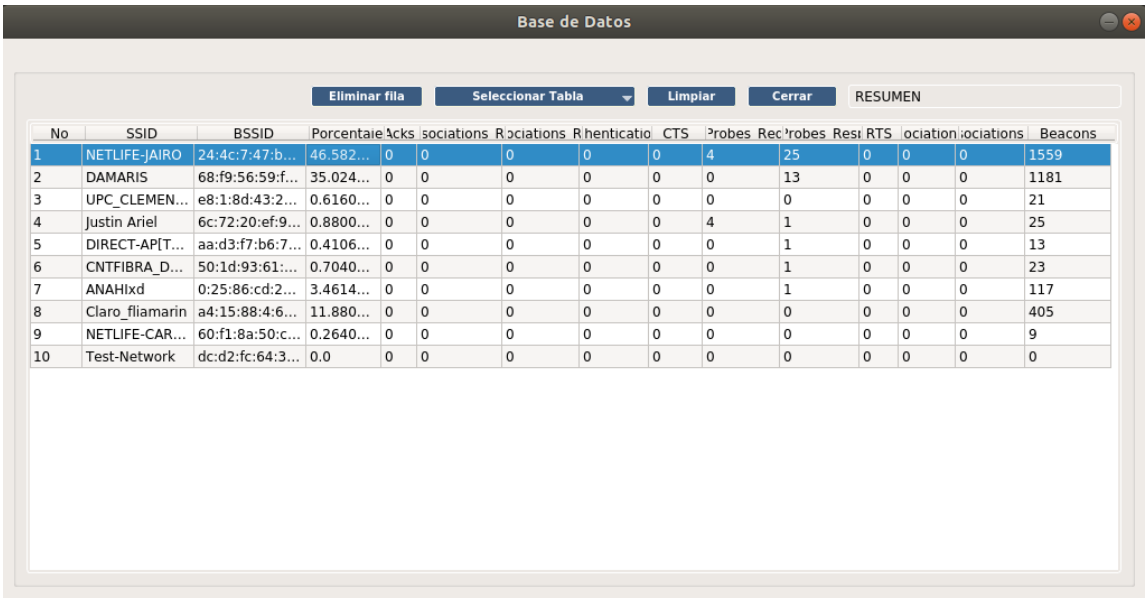


Por último, se realizó una prueba de 10 minutos por la noche para obtener información de la red más específica del SSID NETLIFE-JAIRO con MAC 24:4C:7:47:B5:2C es la red más cercana, en la figura … se puede observar la ventana resumen en la cual se idéntica que dicha

red ocupa el 46.58% del tráfico de la red tomando en cuenta solo tramas de administración ya que las tramas de control por tener una sola MAC no se logró identificar a que BSSID pertenece para tomarlas encuentra para el análisis porcentual.

Figura 31

Ventana Resumen SSID NETLIFE-JAIRO

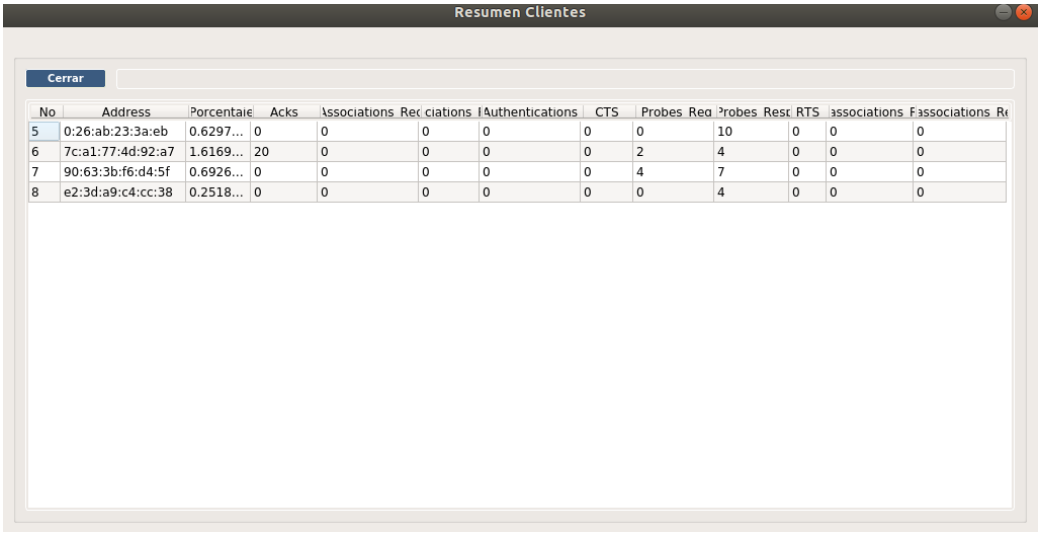


No	SSID	BSSID	Porcentaje	Acks	sociations	ociations	henticatio	CTS	Probes	Probes	RTS	ociation	ociations	Beacons
1	NETLIFE-JAIRO	24:4c:7:47:b...	46.582...	0	0	0	0	0	4	25	0	0	0	1559
2	DAMARIS	68:f9:56:59:f...	35.024...	0	0	0	0	0	0	13	0	0	0	1181
3	UPC_CLEMEN...	e8:1:8d:43:2...	0.6160...	0	0	0	0	0	0	0	0	0	0	21
4	Justin Ariel	6c:72:20:ef:9...	0.8800...	0	0	0	0	0	4	1	0	0	0	25
5	DIRECT-AP[T...	aa:d3:f7:b6:7...	0.4106...	0	0	0	0	0	0	1	0	0	0	13
6	CNTFIBRA_D...	50:1d:93:61:...	0.7040...	0	0	0	0	0	0	1	0	0	0	23
7	ANAHIxd	0:25:86:cd:2...	3.4614...	0	0	0	0	0	0	1	0	0	0	117
8	Claro_fliamarin	a4:15:88:4:6...	11.880...	0	0	0	0	0	0	0	0	0	0	405
9	NETLIFE-CAR...	60:f1:8a:50:c...	0.2640...	0	0	0	0	0	0	0	0	0	0	9
10	Test-Network	dc:d2:fc:64:3...	0.0	0	0	0	0	0	0	0	0	0	0	0

De igual forma se puede identificar los clientes de los distintos BSSID en este caso de NETLIFE-JAIRO en la que se tiene 4 clientes, el correspondiente a la MAC 7C:A1:77:4D:92:A7 con el mayor porcentaje de 1.61%, al comparar el valor de Beacons con Wireshark se obtiene un error de 9.41% que representaría el error de paquetes perdidos de este BSSID.

Figura 32

Clientes asociados al SSID NETLIFE-JAIRO



Capítulo V

Conclusiones y recomendaciones

Conclusiones

Se puede concluir que es factible la realización de un aplicativo que permita caracterizar tramas 802.11 mediante el software libre Linux, con el lenguaje de programación Python que gracias a su gran capacidad, amplios módulos y librerías permitió el desarrollo óptimo del software, podemos precisar que Linux y Python se encuentran en el mercado de forma gratuita para los usuarios, es por cuanto se obtiene un ahorro significativo con relación a sniffers existentes en el mercado ya que su desarrollo tiene costos de elaboración casi nulos por lo cual solo se necesita una computadora y una tarjeta inalámbrica usb.

A través de la información sobre las tramas 802.11 se conoce, que por medio de la cabecera MAC se puede obtener la información más relevante de la trama como el tipo, subtipo, MAC address, encriptación entre otras, así como el número de bytes que ocupa cada una de estas, dicha información es indispensable para el desarrollo del aplicativo.

Se escogió los tipos de tramas más comunes y utilizados como son administración y control, de cada una de estas se obtuvo parámetros como sub tipo, frecuencia, canal, encriptación, duración y contadores que permitirán identificar el total de paquetes por trama. MAC address ya que permiten identificar BSSID y clientes para dar al usuario información completa de cada red.

Se optó por usar el lenguaje de programación Python ya que es una herramienta muy potente, con gran cantidad de información y softwares desarrollados para su entendimiento, además posee librerías como Scapy y sockets indispensables para realizar la captura, así como el procesamiento de los datos. Se exploraron herramientas como airodump para tener una base

sobre la información a seleccionar, aunque existen sin números de sniffers se utilizó como referencia Wireshark para el desarrollo puesto que permite identificar los parámetros de la red según la posición de los bytes en la trama.

Por consiguiente con el uso de Python y con una tarjeta de red en modo monitor que realiza la captura e inyección de tramas, se consiguió crear el aplicativo que cumple con el objetivo de obtener la información más relevante de los equipos en tiempo real, teniendo como principal desafío optimizar el programa para que no existan muchas pérdidas con la captura y procesamiento en tiempo real ya que el módulo PYQT5 con el cual se desarrolló la interfaz gráfica no puede trabajar con multiprocesing, como primera opción se usó 3 hilos 2 de ellos (captura y procesamiento) en paralelo y un tercero (Tshark) en secuencia teniendo intervalos de acuerdo al valor de tiempo ingresado por el usuario, finalmente esta opción se descartó ya que no cumplía con el objetivo y se optó por trabajar con lo los 3 hilos ejecutándose al mismo tiempo y así obtener el software en tiempo real.

Una vez depurado el programa se realizaron pruebas durante 14 días en las redes domésticas del conjunto habitacional Jardín Sur en la que se encontraron 121 redes, de estas 79 usan el canal 1 y se aprecia que existe solapamiento entre canales ya que varias redes usan canales contiguos en las cuales el fabricante que se impuso fue Huawei seguido de Arris, así como la encriptación más usada es WPA2-CCPM-TKIP y 35 equipo usan el mejor mecanismo de encriptación WPA2-CCMP. Al inyectar tramas propias del programa se identificó que la captura de las mismas tanto en el programa como en Wireshark sale duplicada, esto se debe a que se utiliza la misma tarjeta para inyectar y capturar paquetes.

A su vez se obtuvo la eficiencia del programa desarrollado por medio de un análisis de error de paquetes perdidos con relación a los capturados con un sniffer comercial como Wiseshark en el cual se obtuvieron erros en el rango de 0 % a 2.55% con RTS(1.86%), CTS(1.41%),

ACK(1.73%), Beacons(1.80%), Probes(2.55%), authentication(0.05%), Association y Reassociation con 0% de error, mientras que tomando en cuenta solo tramas inyectadas se obtuvo Beacons(0.20%), Probes Request(0.60%), Probes Response(0.12%), , authentication(0.09%), Association Request(0.06%), Association Response(0.15%), Reassociation Request(0.10%) y Reassociation Response(0.15%) con 0% de error, tomando en cuenta que se utilizó una única tarjeta para inyectar y capturar.

Dichos errores están dentro de un margen aceptable y son debido a la programación realizada ya que, al trabajar con hilos, cada uno de estos se turna para su ejecución de 3 a 6 milisegundos en los cuales se puede tener pérdidas tanto en la captura como en el procesamiento ya que se está usando un solo núcleo para obtener información en tiempo real.

Recomendaciones

Se recomienda hacer uso del documento requerimientos.txt, el cual se lo ejecuta desde terminal para instalar todas las dependencias con las versiones usadas para no tener algún problema de compatibilidad con versiones antiguas o más actuales.

Para realizar una comparación con Wireshark se debe filtrar por tipo y subtipo de trama ya que este, captura distintos tipos de paquetes y el software desarrollado procesa solo los paquetes más frecuentes.

Por otro lado, la librería de Python Scapy tiene algunas limitaciones que causaron conflicto al momento de procesar los datos capturados ya que no identifica en su totalidad todas las capas de algunas tramas.

Por lo cual al hacer uso de este módulo se debe consultar la información que tiene cada tipo de trama inyectada con Scapy para que pueda ser detectada por los diferentes analizadores de red en el mercado.

En la capa DotEltRSN de Scapy se llena la información acerca del tipo de encriptación de los paquetes inyectados, si se desea configurar con otro tipo se debe realizar los cambios en esta capa.

A su vez se recomienda trabajar con tarjetas inalámbricas de marcas TP-LINK para obtener el mejor potencial del sistema ya que su desarrollo se basó en el uso de esta marca.

Trabajos futuros

Realizar un estudio más a fondo sobre librerías o módulos que permitan trabajar con multiprocesing para evitar los problemas encontrados al trabajar con hilos de pérdidas de paquetes en tiempo real.

Elaborar el sistema con dos núcleos en diferentes computadores donde la una capture y la otra procese en tiempo real para obtener un mayor rendimiento del sniffer.

Almacenar la información en servidores y diseñar una página web en la cual se pueda encontrar la información de las capturas, así como graficas del comportamiento de cada red.

Referencias

Abedi, A. (2017). *Evaluating and Characterizing.* Tesis de Ingeniería, University of Waterloo, Ciencias de la Computación, Waterloo. Retrieved from https://uwspace.uwaterloo.ca/bitstream/handle/10012/12317/Abedi_Ali.pdf?sequence=3&isAllowed=y

Álvarez, A., Isbarbo, M., & Rivas, B. (2016). *Sistema de monitoreo de capacidad de.* Tesis de Ingeniería, Universidad de la Republica Uruguay, Facultad de Ingeniería, Montevideo. Retrieved from https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/20120/1/AIR16.pdf

Anguís, J. (2008, marzo). *Redes de Área Local Inalámbricas: Diseño de la WLAN de Wheelers Lane Technology College*. Universidad de Sevilla. http://bibing.us.es/proyectos/abreproy/11579/fichero/a.+Portada.pdf

Arizo, L. F. A. (2016, 30 marzo). *Repositorio Digital - EPN: Estudio, pruebas y simulación del estándar IEEE 802.11ac basándose en MU-MIMO (MIMO Multiuser)*. Repositorio Digital EPN. https://bibdigital.epn.edu.ec/handle/15000/15072

Angón, E. (2014). *Mecanismos y estrategias de seguridad en redes Wi-Fi.* Tesis de Ingeniería, Universidad Autónoma del Estado de México, Ingeniería en Computación, Zumpango. Retrieved from http://ri.uaemex.mx/bitstream/handle/20.500.11799/40492/tesinaFinal.pdf?sequence=1&isAllowed=y

Campo, W., López, N., & Chanchí, G. (2019). Análisis de tráfico del servicio de Videostreaming sobre una red WiFi. *Revista Ibérica de Sistemas e Tecnologias de Informação*,

314-326. https://search.proquest.com/openview/2057cc3d20a5c604a2409ce34d16284a/1?pq-origsite=gscholar&cbl=1006393

Cano, L. (2012, diciembre). *CONFIGURACIÓN Y EVALUACIÓN DE REDES 802.11n*. Universidad de Granada.http://dtstc.ugr.es/it/pfc/proyectos_realizados/downloads/Memoria2012_LuisCano.pdf

Estrella, S. (2017). *"ESTUDIO Y ANÁLISIS PARA LA ACTUALIZACIÓN DE RED WLAN DE LA SEPS UTILIZANDO TECNOLOGÍAS BASADAS EN EL ESTÁNDAR IEEE 802.11 AC*. PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR. http://repositorio.puce.edu.ec/bitstream/handle/22000/14425/Caso%20de%20Estudio%20Santiago%20Estrella.pdf?sequence=2&isAllowed=y

Fernandez, R. (2020, 17 abril). *Programación de redes en Python: Sockets*.

Gaitan, V. (2017). *Monitarización y análisis del tráfico en redes inalámbricas 802.11.* Madrid.

García, D., Mosquera, G., & Pérez, M. (2018). *Modelo de optimización de la red Wifi en el politécnico Grancolombiano sede Medellín*. Institución Universitaria Politécnico Grancolombiano. http://alejandria.poligran.edu.co/bitstream/handle/10823/1397/Trabajo%20de%20grado%20ing%20sis.pdf?sequence=1&isAllowed=y

Kasa smart. (2020). *TL-WN722N | Adaptador USB Inalámbrico de Alta Sensibilidad a 150 Mbps | TP-Link Ecuador*. https://www.tp-link.com/ec/home-networking/adapter/tl-wn722n/

Holla, V. (2017, 17 septiembre). *WPA Information Element | Hitch Hiker's Guide to Learning*. Hitch Hiker's Guide to Learning. http://www.hitchhikersguidetolearning.com/2017/09/17/wpa-information-element/

HyperLink Technologies. (2020). *Adaptador USB Inalámbrico de Alta Ganancia TP-Link 150 Mbps TL-WN722N*. http://www.ds3comunicaciones.com/tplink/TL-WN722N.html

Medina, J., & Rivas, Y. (2019). *Evaluación del Rendimiento de un Sistema de Detección de Intrusos para Redes Inalámbricas 802.11 Contra Ataques Informáticos.* Tesis de Ingeniería, UNIVERSIDAD NACIONAL "PEDRO RUIZ GALLO", FACULTAD DE INGENIERÍA CIVIL, DE SISTEMAS Y DE ARQUITECTURA, Lambayaque. Retrieved from http://190.108.84.117/bitstream/handle/UNPRG/8074/BC-4454%20MEDINA%20ROJAS-RIVAS%20MONTALVO.pdf?sequence=1&isAllowed=y

Páez, T. (2015). *Implementación de un prototipo de sistema de análisis de tráfico de redes 802.11 utilizando la minicomputadora Raspberry PI.* Tesis de Ingeniería, Universidad de las Fuerzas Armadas-ESPE, Departamento de Eléctrica y electrónica, Sangolquí. Retrieved from http://repositorio.espe.edu.ec/handle/21000/10672

Python. (2020). *Python*. Retrieved from https://docs.python.org/3/library/sqlite3.html

Python. (2020). *Python*. Retrieved from https://lenguajesdeprogramacion.net/python/

Quiñones, T., Coya, L., & Marichal, L. (2012). Herramientas de monitorización y análisis del tráfico en redes de datos. *Revista Telemática*, 46-59.

ochobitshacenunbyte. (2019, 7 octubre). *Instalación y uso básico de SQLite en Ubuntu 18.04*. https://www.ochobitshacenunbyte.com/2019/10/01/instalacion-y-uso-basico-de-sqlite-en-ubuntu-18-04/

Rojas, M., Deyvi, J., Montalvo, R., & Yajanovic, Y. (2020). Evaluación del Rendimiento de un Sistema de Detección de Intrusos para Redes Inalámbricas 802.11 Contra Ataques Informáticos. *Universidad Nacional Pedro Ruiz Gallo*, 1-97. http://repositorio.unprg.edu.pe/handle/UNPRG/8074

Salazar, J. (2016). Redes Inalámbricas. *TechPedia*, 1-40.

Stafanick, G. (2018, 16 julio). A closer look at WiFi Security IE (Information Elements). Aruba Blogs. https://blogs.arubanetworks.com/industries/a-closer-look-at-wifi-security-ie-information-elements/?fbclid=IwAR2u7pI8lTU7Apr1efNydVBMsOF_VDX2Cq193jmgFW39rqb5qwpogxBPMwM

Silva, L. (2017, septiembre). *Análisis del uso y del tráfico generado en una red inalámbrica, sin autenticación, con conexión a Internet, ubicado en el edificio CIDS de la Universidad Nacional Autónoma de Nicaragua (Unan-León)*. Universidad Nacional Autónoma de Nicaragua, UNAN – LEÓN. http://riul.unanleon.edu.ni:8080/jspui/bitstream/123456789/6555/1/233987.pdf

Solvetic Sistemas. (2019, 22 julio). *Mejores analizadores tráfico de red y Sniffers para Windows y Linux Gratis*. Solvetic. https://www.solvetic.com/page/recopilaciones/s/programas/mejores-analizadores-protocolos-de-red-y-sniffers-para-windows-y-linux-gratis

Scapy. (2008). *Introduction — Scapy 2.4.4 documentation*. https://scapy.readthedocs.io/en/latest/introduction.html#about-scapy

Thomas, G. (2006). Using Etheral for Network Troubleshooting. *Contemporary Control Systems*.