Universidad Católica de Santa María

Facultad de Ciencias e Ingenierías Físicas y Formales

Escuela Profesional de Ingeniería de Sistemas

"PROPUESTA METODOLÓGICA PARA LA IDENTIFICACIÓN DE PATRONES DE FALLA DE CAPA ENLACE DE DATOS DEL MODELO OSI Y RECUPERACIÓN PROACTIVA DE EQUIPOS DE COMUNICACIÓN APLICANDO ALGORITMOS DE APRENDIZAJE AUTOMÁTICO"

Tesis presentada por el Bachiller:
Esquivel Rodríguez, Jorge Armando
Para optar por el título profesional:
Ingeniero de Sistemas: Especialidad en Ingeniería de Software

Asesor:
Mg. Rosas Paredes, Karina

Arequipa - Perú
2020

UCSM-ERP

UNIVERSIDAD CATÓLICA DE SANTA MARÍA

INGENIERIA DE SISTEMAS

DICTAMEN APROBACIÓN DE BORRADOR DE TESIS

Arequipa, 20 de Julio del 2020

Dictamen: 000461-C-EPIS-2020

Visto el borrador de tesis del expediente 000461, presentado por:

2014223491 - ESQUIVEL RODRIGUEZ JORGE ARMANDO

Titulado:

PROPUESTA METODOLÓGICA PARA LA IDENTIFICACIÓN DE PATRONES DE FALLA DE CAPA ENLACE DE DATOS DEL MODELO OSI Y RECUPERACIÓN PROACTIVA DE EQUIPOS DE COMUNICACIÓN APLICANDO ALGORITMOS DE APRENDIZAJE AUTOMÁTICO

Nuestro dictamen es:

APROBADO

1425 - MARTINEZ MUÑOZ JORGE LUIS
DICTAMINADOR

1568 - ROSAS PAREDES KARINA
DICTAMINADOR

PRESENTACIÓN

Señor Decano de la Facultad de Ciencias e Ingenierías Físicas y Formales.

Señor director del Programa Profesional de Ingeniería de Sistemas.

Sres. Miembros del Jurado.

De conformidad con las disposiciones del Reglamento de Grados y Títulos de la Escuela Profesional de Ingeniería de Sistemas, pongo a vuestra consideración el presente trabajo de investigación titulado:

"PROPUESTA METODOLÓGICA PARA LA IDENTIFICACIÓN DE PATRONES DE FALLA DE CAPA ENLACE DE DATOS DEL MODELO OSI Y RECUPERACIÓN PROACTIVA DE EQUIPOS DE COMUNICACIÓN APLICANDO ALGORITMOS DE APRENDIZAJE AUTOMÁTICO"

El presente trabajo de investigación fue realizado gracias a los conocimientos que fui adquiriendo durante mi formación universitaria y laboral, el mismo que de ser aprobado me permitirá optar por el Título Profesional de Ingeniero de Sistemas.

ESQUIVEL RODRIGUEZ JORGE ARMANDO.

DEDICATORIA

El presente trabajo de investigación lo dedico principalmente a Dios, por ser el inspirador y darme la fuerza necesaria para poder continuar en este proceso de obtener uno de los anhelos más deseados.

A mis padres por todo el amor, trabajo y sacrificio en todos estos años, gracias a ustedes pude llegar hasta aquí y convertirme en lo que soy. Estoy muy orgulloso de ser su hijo, siempre serán los mejores padres y los llevare en mi corazón toda la eternidad.

A mis hermanos por estar siempre presentes, acompañándome y apoyándome moralmente a lo largo de esta etapa de mi vida.

A mis amigos y futuros colegas por todo el apoyo dado para poder seguir adelante, muchas gracias por toda su ayuda y buena voluntad.

Agradezco a todos los docentes que me enseñaron a lo largo de la vida universitaria que gracias a su conocimiento, sabiduría y apoyo me motivaron a desenvolverme como persona y profesional.

AGRADECIMIENTO

Agradezco la Ing. Jorge Martínez y a la Mg. Karina Rosas por su asesoría y por su apoyo que me brindaron y a todas las personas que me ayudaron durante la elaboración de este proyecto.

RESUMEN

En este trabajo de investigación se ha propuesto una metodología que ayude a un administrador de red a poder predecir de manera rápida un fallo en la red. Se tomó una modelo de minería de datos como base para poder implementar la metodología, esta metodología abarca desde poder identificar los problemas que se vienen presentando habitualmente en la red hasta la solución definitiva de los mismos, por lo tanto, se tuvo que extraer datos de equipos de infraestructura de red mediante el protocolo simple de administración de red de una manera automatizada aplicando un script desarrollado en el lenguaje de programación Python.

A los datos extraídos se les aplicaron diferentes técnicas de aprendizaje automático supervisado determinando de esta manera que la máquina de soporte vectorial es la más adecuada para poder predecir fallas en base a los datos analizados dado que se obtuvo un 97.8 % de probabilidad de predicción.

Para poder realizar todas las pruebas necesarias sin afectar el estado de la red se propuso el utilizar un emulador grafico de red conocido como GNS3, es aquí donde se emuló una topología compuesta por equipos de comunicación de capa enlace de datos que pueda presentar fallas.

Además, se propusieron políticas de configuración con la finalidad de que al aplicar un mecanismo automatizado para solucionar las fallas este no cometa errores y genere problemas al desactivar algunas características de equipos de red que pueda desencadenas más fallas.

Para finalizar se propusieron 3 maneras de cómo aplicar una solución frente a la identificación de una posible falla a ocurrir señalando a un administrador de red y un software como los principales actores que intervendrían.

Palabras clave

Capa de Enlace de Datos, Protocolo Simple de Administración de Red, Emulador Gráfico, Aprendizaje Supervisado, Políticas de Configuración.

ABSTRACT

In this research work, a methodology has been proposed that helps a network administrator to quickly predict a network failure. A data mining model was used as the basis to implement the methodology, this methodology ranges from being able to identify the problems that are commonly occurring in the network to the final solution of these, for this, data had to be extracted from network infrastructure equipment using the simple network management protocol in an automated way applying a script developed in the Python programming language.

Different supervised machine learning techniques were applied to these extracted data determining in this way that the support vector machine is the most suitable to be able to predict failures based on the analyzed data, given that a 97.8% prediction probability was obtained.

In order to carry out all the necessary tests without affecting the state of the network, it was proposed to use a graphical network emulator known as GNS3, It is here that a topology composed of data link layer communication equipment that may have failures was emulated.

In addition, configuration policies were proposed therefore, when applying an automated mechanism to solve the faults, this does not make mistakes and generates problems by deactivating some features of network equipment that can trigger more faults.

To finish, 3 ways were proposed of how to apply a solution against the identification of a possible failure to occur, pointing to a network administrator and software as the main actors that would intervene in these scenarios.

Keywords

Data link layer, Simple Network Management Protocol, Graphical Emulator, Supervised Learning, Configuration Policies.

INTRODUCCIÓN

Desde su aparición en 1957 las redes de computadoras han facilitado la vida del ser humano para poder comunicarse de una forma rápida y sencilla. Mientras van pasando los años van mejorando para ofrecer una calidad de servicio muy alta para los que hacen uso de la misma.

Hoy en día el tamaño de las redes de computadoras empresariales ha crecido de tal forma de poder conectar diferentes sucursales a lo largo de una ciudad o incluso de manera intercontinental, y es aquí donde surge un tema muy importante que es la administración de estas si bien una red pequeña es fácil de administrar una red grande es más compleja.

Se dice que una red está administrada cuando se está monitoreando constantemente los equipos por la que está compuesta con la finalidad de poder detectar algún cambio anómalo que pueda perjudicar todo su correcto funcionamiento.

Las fallas en red es un tema que ha venido ligado a las redes de computadoras desde la creación de estas y mientras una red crece más es muy probable que esté sujeta a fallas, al día de hoy urge con gran importancia algún sujeto que esté al tanto del correcto funcionamiento de la red para poder detectar alguna falla y es aquí donde está el problema ya que puede significar un costo mayor el tener que contratar a más personal para que pueda administrar la red.

Gracias a los avances que se vienen desarrollando desde los inicios de la primera conexión es que al día de hoy se han creado protocolos de comunicación, dichos protocolos son procesos que se deben seguir para poder establecer un correcto canal de comunicación y transmitir información a través de los mismos, un ejemplo es el protocolo TCP que permite establecer un canal de comunicación orientado a la conexión, lo que significa que tanto emisores como receptores tienen que conectarse antes de poder transmitir información.

Sin duda un protocolo que permitió el desarrollo de este trabajo de investigación es el protocolo simple de administración de red (SNMP) este permite la transferencia de información de administración como ancho de banda, cantidad de paquetes transmitidos, cantidad de paquetes recibidos, etc. entre dispositivos de red.

En este trabajo de investigación se hace uso del protocolo simple de administración de red (SNMP) para poder extraer información de equipos de red y analizarlos mediante técnicas de aprendizaje automático con la finalidad de poder detectar un patrón de alguna falla próxima a ocurrir y poder evitar dicha falla antes de que desencadene una falla total en toda la red de computadoras.

El trabajo de investigación está constituido por 4 capítulos en los que se tratan los siguientes temas:

En el capítulo 1, Planteamiento del problema, objetivos del trabajo de investigación, las bases teóricas de la investigación además de una solución propuesta para el problema a tratar.

En el capítulo 2, El marco teórico del proyecto, así como conceptos relacionados a varios aspectos a tratar para más adelante argumentar sobre los mismos.

En el capítulo 3, Diseño e implementación de la propuesta metodológica, fases de la metodología propuesta y aplicación de la metodología tratada.

En el capítulo 4, Se muestran los resultados de haber aplicado la metodología planteada además de una discusión sobre los resultados obtenidos.

Finalmente se definen las conclusiones y recomendaciones para trabajos futuros

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

CAPITULO I

1. PLANTEAMIENTO TEÓRICO

1.1. PLANTEAMIENTO DE LA INVESTIGACION

1.1.1. Planteamiento del Problema

Una red de computadoras o red de ordenadores es un conjunto de nodos conectados entre sí gracias a un medio por el cual se puede transmitir información de manera segura y confiable.

Según Ariganello (2016), una red de área local y/o red de área amplia conectan a los usuarios dentro y fuera de una organización y permite gran cantidad y diversos tipos de comunicación.

Sin embargo, el tema central de las redes no se basa en los dispositivos ni en los medios si no en los protocolos que indican y especifican la manera en cómo se envían y transmiten los mensajes, cómo se direccionan y reenvían a través de la red y cómo se interpretan en dispositivos de destino.

Según IBM Knowledge Center. (2018), un protocolo de comunicación es un conjunto de reglas y/o formatos que se deben cumplir para la correcta comunicación entre el emisor y el receptor, hoy en día hay bastantes y diferentes protocolos de red que permiten un correcto y óptimo funcionamiento en la red.

Sin embargo, los protocolos de comunicación están sujetos a fallas. Es por eso por lo que urge un método, un procedimiento o una técnica capaz de identificar una posible falla que puedan producirse con los protocolos antes de que ocurran ya que de presentarse el problema en el momento menos oportuno causará varios problemas en la infraestructura de red.

Según Deivid, D. (2018), un bucle de capa 2 ocurre cuando la red se satura de tramas enviadas principalmente por un switch, este envía un broadcast y ya que si una red no tiene configurado el protocolo STP, la trama seguirá dando vueltas y multiplicándose en toda la red.

Según Cisco Systems. (2016), los bucles de capa 2 (enlace de datos) son los problemas más comunes de encontrar en una red, los mismos pueden llevar a saturar la red y dejarla inservible. Por lo tanto, se tomará este problema como ejemplo para poder aplicar algoritmos de aprendizaje automático para poder así

identificar la falla antes de que ésta ocurra y así poder dar una solución lo más rápido posible preferiblemente antes que ocurra.

Según Analytics, Business, Intelligence and Data Management. (2018), es posible generar modelos de aprendizaje de una forma muy rápida y sencilla que puedan analizar grandes cantidades de datos y complejos y dar resultados muy precisos.

Se aplicaran técnicas para poder extraer datos constantemente de los equipos de red para poder así monitorear y poder encontrar un patrón que nos ayude a identificar qué problemas causan los bucles que se presentarán en ese momento luego gracias a los datos que se proporcionarán en el patrón encontrado se mostrarán opciones para que el personal de redes pueda resolver este problema o con ayuda del protocolo SNMP reconfigurar los equipos de red para que el problema pueda ser solucionado en primera instancia.

1.1.2. Objetivos de la Investigación

a) General

Proponer una metodología que identifique patrones de falla en capa enlace de datos y permita la recuperación de equipos de comunicación usando algoritmos de aprendizaje automático.

b) Específicos

- Proponer configuraciones adecuadas para aplicar en equipos de comunicación con el fin de poder generar un canal de extracción constante de datos.
- Identificar y aplicar las técnicas de aprendizaje automático que permitan predecir con gran exactitud una futura falla de equipos de red y a su vez que permitan realizar un correcto preprocesamiento de los datos previamente extraídos.
- Determinar el lenguaje de programación adecuado para poder implementar los algoritmos de aprendizaje automático.
- Proponer el uso de un emulador para poder implementar la topología real de una empresa y así poder realizar las pruebas adecuadas en el mismo.

- Establecer políticas de configuración para poder proponer formas de recuperación de equipos de comunicación.

1.1.3. Preguntas de Investigación

- ¿La metodología propuesta podrá ayudar a identificar patrones de falla en equipos de comunicación?
- ¿Qué configuración se debe aplicar a los equipos de red para poder establecer un canal de comunicación que permita extracción de datos en todo momento?
- ¿Cuáles son las técnicas de aprendizaje automático adecuadas para poder analizar los datos extraídos?
- ¿Qué lenguaje programación facilitará el análisis de los datos?
- ¿Es adecuado el uso de un emulador o un simulador para poder hacer pruebas antes de aplicar la configuración en un ambiente real?

1.1.4. Línea y Sub-Línea de Investigación

- Línea de Investigación: Redes y Telemática
- Sub-línea de Investigación: Gestión de servicios de red

1.1.5. Palabras Clave

Capa de enlace de datos, protocolo simple de administración de red, emulador gráfico, aprendizaje supervisado, políticas de configuración.

1.1.6. Solución Propuesta

a) Justificación e Importancia

Hoy en día existen diferentes sistemas de información y tecnologías que funcionan bajo una misma red capaz de interconectar todo entre sí con la facilidad de poder transmitir información de manera segura y confiable, es por eso por lo que el ingeniero en redes debe apostar, por el correcto funcionamiento de la red ya que es parte indispensable de la infraestructura empresarial. Sin embargo, los problemas nunca están de más ya que pueden originarse de diferentes formas que incluso a veces pueden escapar del

conocimiento del personal en turno, una de las causas podría ser el crecimiento rápido de la red dando paso a diferentes problemas como errores de configuración, problemas con equipos de proveedores diferentes, problemas de enrutamiento, etc.

Por lo tanto, urge un método o proceso que nos ayude a identificar los problemas que ocurren día a día en una red empresarial.

b) Descripción de la Solución

La metodología planteada describe una serie de procesos a seguir por el personal calificado en el área para poder identificar de manera rápida, sencilla y correcta los problemas que pueden presentarse en los equipos que componen la red empresarial.

Existirán de una a varias formas de solucionar o alertar los diferentes tipos de problemas que se podrán presentar, entre los cuales pueden ser una reconfiguración del protocolo simple de administración de red (SNMP), bloqueo de puertos de equipos, reconfiguración de la cantidad de paquetes que pueden pasar por un puerto, bloqueo de nodos de red, entre otros. Además, se tendrán alertas que indiquen que un problema próximo está por ocurrir, lo cual se podrá prevenir gracias a una red neuronal previamente entrenada o algún algoritmo de aprendizaje automático previamente entrenado.

1.2. FUNDAMENTOS TEÓRICOS

1.2.1. Estado del Arte

El correcto funcionamiento de una red es un tema muy crítico para las empresas, es por eso por lo que se necesitan métodos y/o técnicas capaces de ayudar a identificar diferentes problemas de forma rápida y eficaz y dar una solución inmediata.

Según Bhuyan et all. (2014), cualquier red que pueda ser implementada por alguna persona calificada o entidad a menudo está sujeta a eventos anómalos que degradan la calidad de servicio de ésta.

Las redes pueden ser de suma importancia para la infraestructura empresarial hoy en día, por lo que es esencial su correcto funcionamiento, así como la

calidad de servicio es importante para la entidad que ofrece el servicio de conexión.

Hoy en día se usan herramientas de monitoreo cómo SolarWinds, PRTG, WhatsUp Gold, etc. capaces de alarmar a los analistas de red sobre algún evento que se esté presentando en algún equipo previamente configurado, según INCIBE. (2019), el monitoreo de redes hoy en día es un pilar muy importante en el ámbito de las tecnologías de información, ya que permite conocer el comportamiento de las comunicaciones.

Pero las herramientas de monitoreo solo emitirán la alarma cuando se sobrepase el umbral de fallo previamente configurado de algún nodo de red. Puede ser que en algún momento un incidente diferente a los que se tiene previsto que pase en algún nodo de red ocasione problemas, sin embargo, la herramienta de monitoreo mostrará que se ha producido un problema, más no especificará qué es lo que probablemente pudo ocasionar el problema.

Según ReporteDigital (2019), el monitoreo en red permite analizar la información de la conexión que se realiza entre una computadora y la red ya sea de área local (LAN) o externa (WAN), entre otras operaciones que se pueden realizar están identificar direcciones IP, cantidad de información que transita en la red, resolución de direcciones MAC, etc.

Según Pandora FMS. (2019), una buena herramienta de monitoreo no solo debe ser capaz de monitorear redes, sino que tenga la capacidad de escalar y monitorear más elementos de la empresa.

Una buena herramienta de monitoreo debe ser fácil de usar y de entender, sin embargo, una de las principales desventajas al momento de usar un sistema de monitoreo en red es el tener que elegir la herramienta adecuada para que cumpla dicha función. Hay diferentes sistemas en la web ya sean pagados o de código abierto, es muy difícil cubrir las expectativas de todos los usuarios para poder ofrecer la mejor herramienta, lo que lleva al fabricante a tratar de agregar todas las funciones posibles al sistema de monitoreo que venderá, como resultado final se obtiene una herramienta que en muchos casos no termina de configurarse por la empresa que la adquiere de forma completa al momento de su instalación o en el peor de los casos una herramienta difícilmente de aprender a usar.

Según Khan, R., & Khan, S. U. (2013), el término "sistema de monitoreo de red" describe a un software que analiza constantemente el tráfico en una topología de red para detectar interferencias o fallas en los componentes y alertar a la persona responsable de monitorear la red vía correo electrónico, SMS u otros. También indican que un sistema de monitoreo ideal deberá contener lo siguiente:

- Monitoreo continuo de la red.
- Notificaciones inmediatas en bien se presente el incidente.
- Debe ser capaz de identificar el problema principal y la ubicación precisa de este en la topología de red.
- Deberá mantener un registro sobre los cambios que se hacen en la red en tiempo real.
- Deberá tener configurado autenticación y autorización remota para poder acceder desde cualquier parte.

Sin embargo, se deben añadir algunos puntos que hoy en día muchos consideran importantes para beneficio propio los cuales son:

- No deberá tener un costo elevado.
- Deberá ser fácil de configurar y utilizar.
- Continuamente deberá ser parchado para no recibir ataques.
- Capacidad de poder personalizarlo como sea posible.
- Deberá generar informes automáticamente para reducir el tiempo de generarlos.

Por lo tanto, según los investigadores un sistema de monitoreo de red solo permitirá informar sobre las fallas de red una vez se haya alcanzado el umbral configurado y en muy pocos casos nos ayudará a analizar cuál fue el punto de partida que ocasionó el problema principal que desencadenó otros más. Urge un método y un sistema que ayude al responsable de la red a analizar en tiempo real si cabe la posibilidad de que ocurra un error en los próximos minutos y de qué

punto partiría este error para poder así evitar en el mejor de los casos que ocurra el incidente.

La solución a este problema se lograría al encontrar patrones en los datos extraídos de los equipos de red de una topología empresarial, gracias a los datos extraídos y con la experiencia que tenga el equipo de haber analizado otros errores anteriormente se podrá alertar de manera inmediata cuando y donde se generará el incidente.

Existen diferentes protocolos que facilitan el intercambio de información entre dispositivos de red, así como el tráfico generado, las interfaces usadas, el uso del CPU, etc.

Dichos protocolos pueden ayudar a extraer los datos específicos que se necesitan para poder analizar el estado de salud de los equipos de red.

La extracción de datos en tiempo real de los equipos en red se lograría gracias a los protocolos SNMP, Netflow o Syslog ya que son los más adecuados para extraer información detallada. Dado que SNMP lleva más tiempo en actividad que los otros protocolos y ya que permite extraer datos que los demás protocolos no pueden dar se tomará este protocolo para ser usado en la metodología.

Según Martínez, C. Y. (2016), el protocol SNMP puede funcionar en la mayoría de los equipos de red mientras Netflow solo estaría funcionando en equipos Cisco y otros pocos equipos de otras marcas comerciales, en el caso del protocolo Syslog solo sirve para enviar información sobre el registro de eventos que tiene un dispositivo.

Según Roughan, M., & Rocky, C. (2013), el Protocolo Simple de Administración de Red (SNMP) se usa ampliamente para proporcionar datos de los componentes de red. Dichos datos sin detallarse completamente proporcionan una fuente valiosa de información para los administradores de red, lo que permite ayudar a tomar decisiones sobre el enrutamiento, aprovisionamiento y la configuración de infraestructura de red.

Los datos proporcionados por SNMP son fáciles de obtener y almacenar lo que implica el bajo uso de espacio en disco donde se almacenará la data extraída.

Según Montenegro, D. F. Z. (2016), usando Netflow se pueden extraer datos más precisos de muchos hosts a la vez, la información que es más detallada y mejorada proporcionada por Netflow requiere un costo de procesamiento elevado para los equipos que los van a ejecutar y así mismo para su recolección.

Es por eso por lo que para no generar costos elevados de procesamiento se extraen muestras grandes y pesadas, pero puede traer diferentes tipos de complicaciones para los equipos implicados en el monitoreo como la red misma.

La muestra grande y pesada extraída mediante Netflow en intervalos de tiempo largos puede afectar significativamente la precisión de los datos y más adelante puede ser posible que no se puedan usar los datos en su totalidad.

Según Choi, J., Hu, K., & Antoniades, D. (2013), los datos proporcionados por el protocolo Netflow son más detallados y se puede obtener información mucho más precisa sobre grandes flujos de información de extremo a extremo, rendimiento de transferencias, etc. Sin Embargo, los datos que proporciona Netflow no son del todo seguros.

Un ejemplo claro sería al momento de vulnerar la información, en los grandes flujos de datos hay información crítica cómo direcciones IP o direcciones MAC lo que implicaría serios problemas de privacidad si se logra capturar algunos paquetes por algún atacante infiltrado en la red ya sea de forma interna o remota.

Según 1&1 IONOS España S.L.U. (2019), el protocolo SNMP cuenta con diferentes versiones siendo la última (SNMPv3) la más segura ya que se adiciona a las funciones anteriores la seguridad criptográfica, la identificación de entidades para facilitar el transporte de datos sólo entre entidades conocidas, además de añadir soporte para modelos de seguridad basado en usuarios, entre otros.

Una vez que se tengan los datos extraídos por SNMP y almacenados en el formato deseado se necesitará un algoritmo de aprendizaje automático que sea el adecuado para poder tratar los datos.

El algoritmo será el encargado de encontrar un patrón que indique una falla próxima y el inicio de la misma, existen muchos algoritmos de aprendizaje automático, clasificándolos tenemos a los supervisados y los no supervisados, se

puede aplicar cualquier método para tratar a los datos, pero será el correcto el que logre predecir con mayor exactitud el problema a tratar.

Según Techtarget (2017), aprendizaje automático es un tipo de inteligencia artificial que da a la computadora la capacidad de poder aprender sin necesidad de ser programadas intencionadamente. Este se centra en el desarrollo de software que puede cambiar cuando se es expuesto a nuevos conjuntos de datos.

Según Gómez, M. D. P. (2016), los sistemas que se basan en inteligencia artificial tienen una representación del conocimiento que permite tomar decisiones de manera autónoma.

Martin, J. (2018), el aprendizaje automático y la minería de datos son casi similares, en ambos casos se buscan entre los datos algún patrón en específico que responda a algún problema. Sin embargo, la minería de datos se centra en extraer datos para que puedan ser comprendidos por los usuarios y luego dar proporcionar una decisión por parte de los mismos, mientras que el aprendizaje automático usa los datos para detectar algún patrón y ajustar las acciones de un programa configuradas en un principio.

Según Invid. (2019), el aprendizaje supervisado enseña a las computadoras mediante el uso de ejemplos o conjunto de datos de capacitación, el aprendizaje no supervisado utiliza conjunto de datos sin etiquetar ni clasificar pues el sistema tendrá la misión de identificar un patrón en fragmentos de datos.

Los algoritmos supervisados pueden aplicar los puntos aprendidos con anterioridad a nuevos datos mientras que los no supervisados pueden deducir algo a partir de otra información otorgada.

Según Cajamarca, M. (2019), detrás de todos los algoritmos se encuentran análisis estadístico y análisis predictivos que ayudan a analizar los patrones y llega a una conclusión.

La detección de anomalías en red, así como fallas, intrusiones, etc. consiste en identificar muestras de un conjunto de datos que se desvían de otras muestras como para suponer que hay algún mecanismo diferente a lo que se está analizando.

Las anomalías son diferentes a los ruidos que se encuentran en los datos, estas pueden definirse como un fenómeno que no es interés para el encargado de monitorear la red, pero actúa como un obstáculo para poder analizar de manera correcta los datos.

Es por eso por lo que se deben tratar los datos de manera correcta para que ningún dato que nos reste importancia a lo que estamos analizando interfiera en la investigación que se está haciendo.

Sin duda una metodología adecuada que contenga una buena topología de red para extraer datos, un protocolo adecuado para poder ayudar a extraer datos, un buen algoritmo de aprendizaje automático y sobre todo una buena seguridad ayudará bastante al momento de identificar fallas de forma rápida y sencilla además de poder dar solución inmediata a dicho problema.

1.3. MARCO METODOLÓGICO

1.3.1. Bases Teóricas de la Investigación

a) Protocolo de Comunicación

Tolosa, G. (2014), indica que un protocolo de comunicación es un conjunto de reglas y formatos establecidos para que una comunicación entre un emisor y receptor sea posible y factible. Las reglas indican una serie de pasos a respetar para poder transmitir los datos. Mientras que los formatos serían maneras en cómo se va a encapsular la información antes de enviarse.

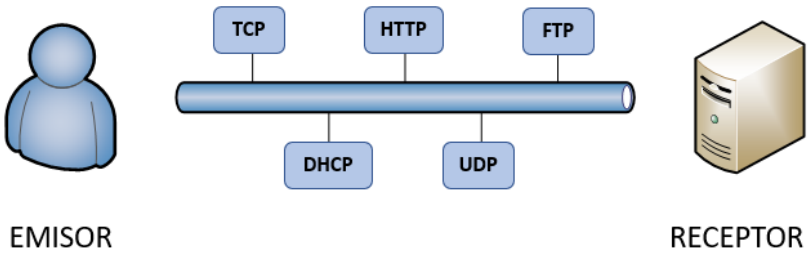

Figura 1. Canal de comunicación

Fuente: Elaboración propia

b) Protocolo Simple de Administración de Red (SNMP)

Hidalgo, F., & Gamess, E. (2014), indica que el protocolo simple de administración de red es un protocolo creado para poder gestionar la red, este se usa ampliamente y es fácil de usar. SNMP es un protocolo perteneciente

a la capa de aplicación que facilita el intercambio de información entre los agentes y sistemas de gestión de redes.

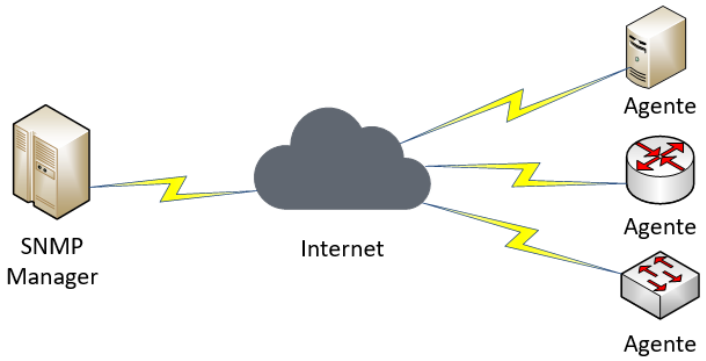

Figura 2. Arquitectura de SNMP

Fuente: Elaboración propia

c) Base de Información Gestionada (MIB)

Rai, N. A. (2014), indica que la base de información gestionada es un conjunto de datos ordenados de manera jerárquica. La base de información gestionada define los objetos que serán administrados por un administrador SNMP.

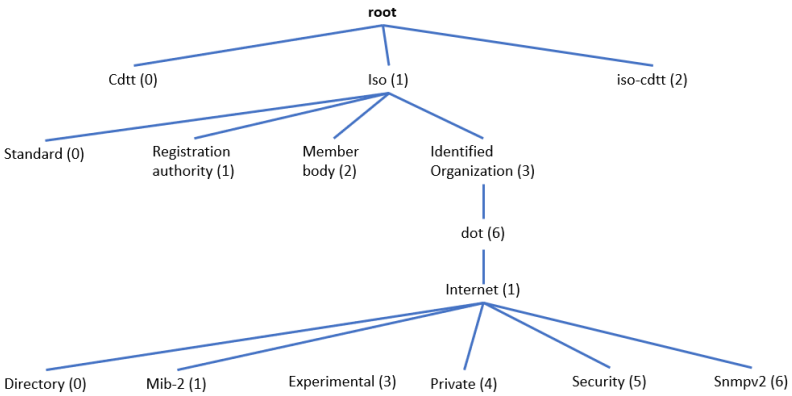

Figura 3. Estructura de una MIB

Fuente: Elaboración propia

d) Identificador de Objeto (OID)

Broadcom. (2019), indica que un identificador de objeto es un identificador de un objeto específico en la base de información gestionada.

e) Aprendizaje Automático

Rodríguez Tapia, S., & Camacho-Cañamón, J. (2018), indican que el aprendizaje automático es una rama de la inteligencia artificial que se dedica a desarrollar técnicas para que sea posible el aprendizaje por parte de los ordenadores sin necesidad de volver a programarlos.

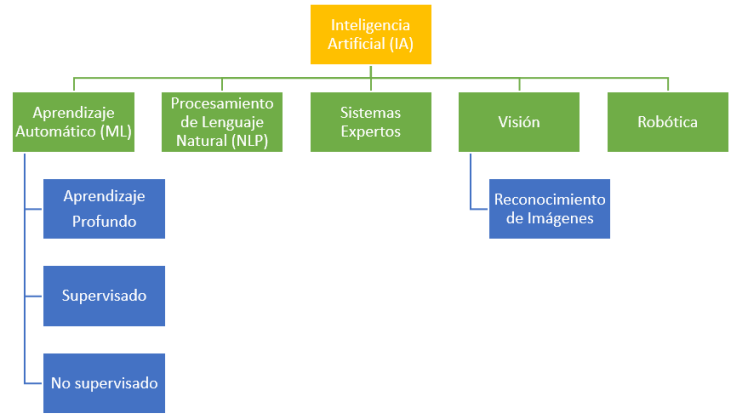

Figura 4. Ramas de la inteligencia artificial

Fuente: Elaboración propia

f) Redes Neuronales

Sprockel, J. J., Diaztagle, J. J., Alzate, W., & González, E. (2014), indican que las redes neuronales son estructuras computacionales que están formadas por modelos matemáticos que siguen un proceso algorítmico basado en el cálculo que se basan por analogía en la manera en cómo funciona el sistema nervioso central usando nodos conectados entre sí que hacen de neuronas y conexiones entre los mismos.

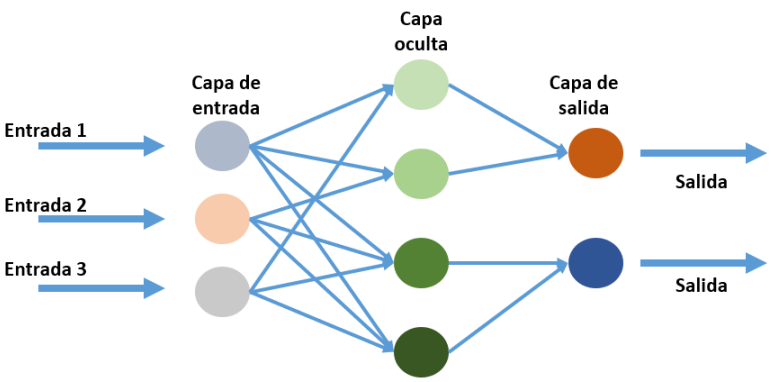

Figura 5. Estructura de una red neuronal

Fuente: Elaboración propia

g) Árboles de Decisión

Jaramillo, A., & Arias, H. P. P. (2015), indican que los árboles de decisión son modelos que permiten la predicción y a su vez son un conjunto de decisiones que permiten clasificar datos.

h) Propagación hacia Atrás

Burgueño, L., Cabot, J., & Gérard, S, (2015) indican que la propagación hacia atrás es un método de cálculo en la que las salidas de las neuronas retroalimentan de nuevo a la red.

i) K – Medias

Felipe, V., & Ramírez, E. (2014), indican que k – medias es un método que permite agrupar datos dado un número de grupos y un conjunto de datos que serán agrupado, permite determinar para cada elemento del conjunto de datos el clúster que más se aproxima.

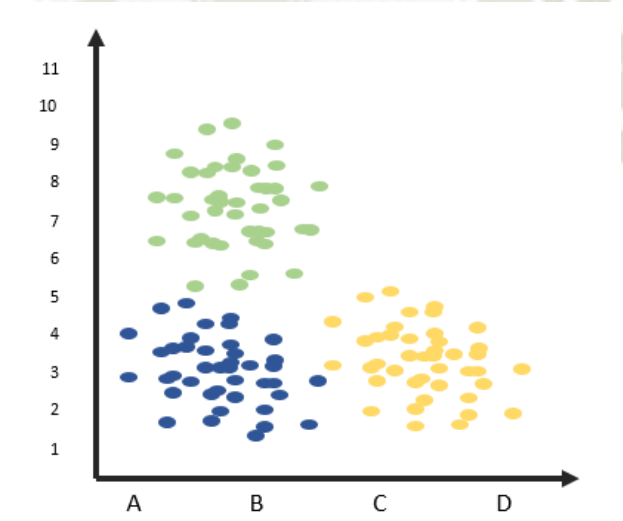

Figura 6. Ejemplo de un agrupamiento de datos con k-medias

Fuente: Elaboración propia

j) Análisis Discriminante Lineal

Rpubs (2016), indica que el análisis discriminante lineal es un método de clasificación supervisado de variables cualitativas muy utilizado para encontrar combinaciones lineales.

1.4. Alcances y Limitaciones

1.4.1. Alcances

El trabajo de investigación tiene la finalidad de implementar una metodología que permita la extracción constante de información de equipos de red para un análisis continuo usando algoritmos de aprendizaje automático para la predicción de fallas futuras.

La investigación tomará en cuenta el protocolo simple de administración de red (SNMP) en su versión 2 que será el que permitirá que se pueda extraer la información necesaria.

Además de contar con una topología con las respectivas configuraciones que nos permita extraer los datos sin problema.

1.4.2. Limitaciones

La investigación está limitada a analizar el tráfico proporcionado por los equipos de red, de la capa de enlace de datos, de red y transporte del modelo OSI ya que en estas capas del modelo es donde ocurren mayormente las fallas.

Además, está limitada a extraer información de equipos de red mediante el protocolo SNMP en su versión 2 ya que este protocolo y la versión fueron las elegidas para implementar en este trabajo de investigación.

Finalmente se limita al uso de las configuraciones recomendadas para aplicar en equipos de red de capa enlace de datos debido a que estas son las que se requieren para poder extraer los datos.

1.4.3. Tipo y Nivel de Investigación

a) Tipo de Investigación

La presente investigación es de tipo aplicada , ya que el propósito principal es el de actuar sobre el funcionamiento de la infraestructura de red analizando constantemente la información necesaria proveniente de los equipos de red, utilizando el protocolo simple de administración de red (SNMP), aplicando los algoritmos de aprendizaje automático necesarios para encontrar algún patrón, logrando identificar de manera temprana una posible falla futura e

identificando el motivo que pudo desencadenar dicha falla además de sugerir o dar una solución al problema de manera inmediata.

b) Nivel de Investigación

La presente investigación cuenta con dos tipos:

Experimental, la cual indica una metodología con la finalidad de poder encontrar patrones que indiquen posibles fallas y dar soluciones antes y luego de haber ocurrido el problema.

Correlacional, la cual permite el análisis de la relación que tienen las variables dependientes con la independientes.

1.4.4. Población y Muestra Metodológica

a) Población

Trafico del protocolo simple de administración de red (SNMP) de una red empresarial proveniente de equipos de comunicación.

b) Muestra

Un conjunto de 700 a más registros provenientes del protocolo simple de administración de red (SNMP) extraídos de cada equipo de infraestructura de red conectados entre sí que actualmente se encuentren operando extraídos mediante un script programado en Python usando la librería pySNMP.

1.4.5. Métodos, Técnicas e Instrumentos Empleados

Para la implementación de la metodología se utilizaron los siguientes métodos, técnicas e instrumentos de recolección de datos:

a) Métodos

- Observación Experimental: Ya que se van a elaborar los datos en condiciones parcialmente controladas por la persona que investigará los sucesos a ocurrir y dado que este manipulará las variables a estudiar.

b) Técnicas

- Selección: Del tráfico de red a analizar.
- Identificación: De todos los equipos de red que cuenten con el protocolo SNMP en su configuración

- Observación y Medida: De las variables a analizar.

c) Instrumentos de Recolección de datos

- Discusión y análisis: Nos permitirá analizar diferentes protocolos de los cuales se tendrá que elegir el más adecuado que proporcione la información deseada.
- Hoja o ficha de registro de datos: Nos permitirá almacenar todos los datos extraídos de las variables elegidas por la persona que va a investigar el suceso.
- Diccionario de datos: Permitirá dar a conocer el listado de las variables a extraer y las definiciones de cada una.

CAPITULO II

2. MARCO TEORICO

En este capítulo se desarrollará el marco teórico, este permitirá al lector una mejor comprensión acerca de la investigación que se está llevando a cabo.

Así mismo en este apartado se encontrarán los conceptos necesarios que se refieren a la propuesta metodológica que se está desarrollando en este documento.

2.1. CONCEPTOS DE REDES DE COMUNICACIONES

2.1.1. Protocolo de Comunicación

Es un conjunto reglas y/o normas que permiten que diferentes entidades en un sistema de comunicación puedan establecer un canal para poder comunicarse y así poder transmitir información a través de este (Díaz, 2013).

2.1.2. Trama de Red

Es una serie sucesiva de bits, además considerada una unidad de envío de datos. Dichos bits están ordenados de forma cíclica y ordenada lo que permite al receptor extraer la información de manera correcta.

Preámbulo	Inicio de delimitador de trama	Dirección destino	Dirección origen	Tipo	Datos	Secuencia de verificación de trama

Figura 7. Estructura y composición de una trama de red

Fuente: Elaboración propia

2.1.3. Dato

Es una representación simbólica de algo, puede ser de variables, atributos, etc.

Sin embargo, el concepto que se tendrá en cuenta para este trabajo de investigación indica que el dato es una porción de la información de algo en concreto.

2.1.4. Conexión

Es un sistema con el que cuenta un dispositivo que necesita conectarse a la red para transmitir información, este puede ser alámbrica o inalámbrica.

2.1.5. Protocolo de Internet (IP)

Es un protocolo de comunicación de datos que se encuentra en la capa de red del modelo osi.

Su función principal es la de comunicar un origen y un destino bidireccionalmente mediante un protocolo no orientado a la conexión (1&1 Ionos España S.L.U., 2019).

Tabla 1

Clasificación de las direcciones IP

Clase	Desde	Hasta	Cantidad de redes	Cantidad de hosts
A	0.0.0.0	127.255.255.255	128	16777214
B	128.0.0.0	191.255.255.255	16384	65534
C	192.0.0.0	223.255.255.255	2097152	254
D	224.0.0.0	239.255.255.255	-	-
E	240.0.0.0	255.255.255.255	-	-

Fuente: Elaboración propia

2.1.6. Switch de capa 3

Equipo de comunicación encargado de conmutar tramas en función a la información contenida en capa de red. Un switch de capa 3 tiene todas las funciones de un switch de capa enlace de datos la diferencia radica en que este es capaz de enrutar tráfico de red.

2.1.7. Modelo OSI

El modelo OSI también conocido como modelo de conexión de sistemas abiertos, es un estándar que hace referencia a los protocolos de comunicación.

Este modelo permite identificar la función de cada protocolo en diferentes sectores de una red de computadoras.

Su principal función es interconectar diferentes sistemas de procedencia distintas.

El modelo OSI tiene 7 capas descritas a continuación:

Tabla 2

Comparación entre Python y Java

Capa	**Unidad de datos**
Capa Aplicación	Datos
Capa Presentación	Datos
Capa Sesión	Datos
Capa Transporte	Segmentos
Capa de Red	Paquetes
Capa Enlace de Datos	Tramas
Capa Física	Bits

Fuente: Elaboración propia

2.1.8. Capa de Enlace de Datos

Es el segundo nivel del modelo OSI, en este nivel se encuentran las tramas como unidades de datos, recibe paquetes de la capa superior (capa de red) para ser reenviados por los servicios de la capa física a otro medio de comunicación.

2.1.9. Netflow

Es un protocolo de red que tiene la finalidad de recolectar datos de la red, fue desarrollado por Cisco Systems. Está diseñado para recolectar datos de tráfico IP.

2.1.10. Syslog

Es un estándar que sirve para él envió de información de registro dentro de una red informática. Tiene por finalidad registrar todos los eventos que suceden en determinados equipos.

2.1.11. Base de Información Gestionada (MIB)

La base de información gestionada es un tipo de base de datos que contiene OID's (identificador de Objetos) estructurados de manera jerárquica en forma de árbol.

2.1.12. Identificador de Objeto (OID)

Los OID´s sirven para gestionar y controlar diferentes componentes en una red.

Se encuentra compuesto por una serie de números que indican el nivel jerárquico en el que se encuentra el objeto.

2.1.13. Comunidad

Es un nombre que se emplea para la autenticación al momento de enviar una petición SNMP.

Generalmente el nombre es "public", se recomienda no usar este nombre ya que es poco seguro, hay una comunidad para la operación de lectura y otra para escritura.

2.1.14. Puerto Lógico

Es una zona en la memoria principal que está asociada a un protocolo o un puerto físico que permite almacenar información temporalmente de la información que va a ser transferida.

Hay un total de 65536 puertos lógicos que van enumerados desde el 0.

2.1.15. Puerto Físico

Un puerto físico es una conexión física la cual se usa para poder conectar dispositivos lo que permite el intercambio de información, de esta manera pueden interactuar uno o más dispositivos conectados simultáneamente.

2.1.16. Administrador SNMP

Cumple la función de enviar y extraer información de los agentes SNMP.

Generalmente es un host en el que se está ejecutando un sistema de monitoreo de red el cual emitirá una alerta si se pasa algún umbral previamente configurado.

2.1.17. Agente SNMP

Son los equipos de infraestructura de red previamente configurados de donde se va a extraer la información.

Hoy en día la mayoría de equipos de red traen consigo el protocolo SNMP instalado.

2.1.18. Versión de SNMP

Existen 3 versiones de protocolo SNMP, cada una trae consigo mejoras con las que no contaba la anterior versión. La última versión trae consigo autenticación

y autorización, lo cual permite una mayor seguridad al momento que se ejecute el protocolo.

2.1.19. Sistema de Monitoreo

Un sistema de monitoreo es un software que permite la extracción de datos de equipos de red por un determinado periodo, la función principal de este es detectar fallas una vez que hayan ocurrido para que algún otro agente intervenga en la solución de este problema.

2.1.20. Protocolo de Árbol de Expansión (STP)

Es un protocolo de la capa enlace de datos que permite la disponibilidad de la red cuando se presente algún bucle que pueda ocasionar congestión en toda la red.

2.1.21. Bucle de capa 2

Es un problema que como su nombre indica solo ocurren en capa enlace de datos, consiste en la generación de bucles en la red, lo que ocasiona que se envíen tramas a través de todos los puertos ocasionando congestión y sobrecarga en el CPU del equipo de red.

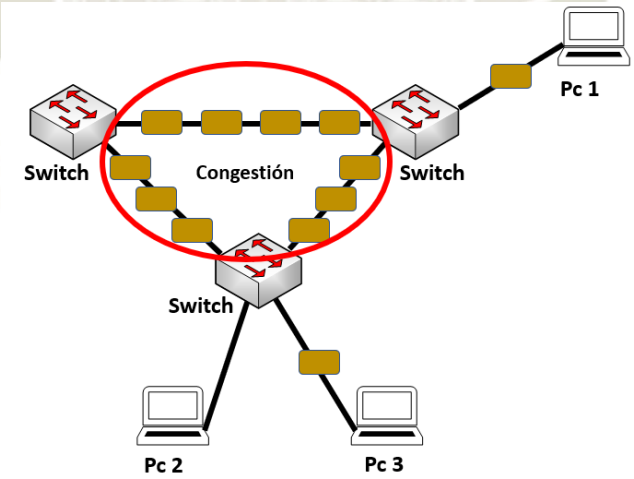

Figura 8. Ejemplificación de un bucle de capa enlace de datos

Fuente: Elaboración propia

2.1.22. GNS3

Es un software que sirve para emular dispositivos de red y para diseñar topologías de red, permite la combinación de equipos reales y virtuales en un solo entorno.

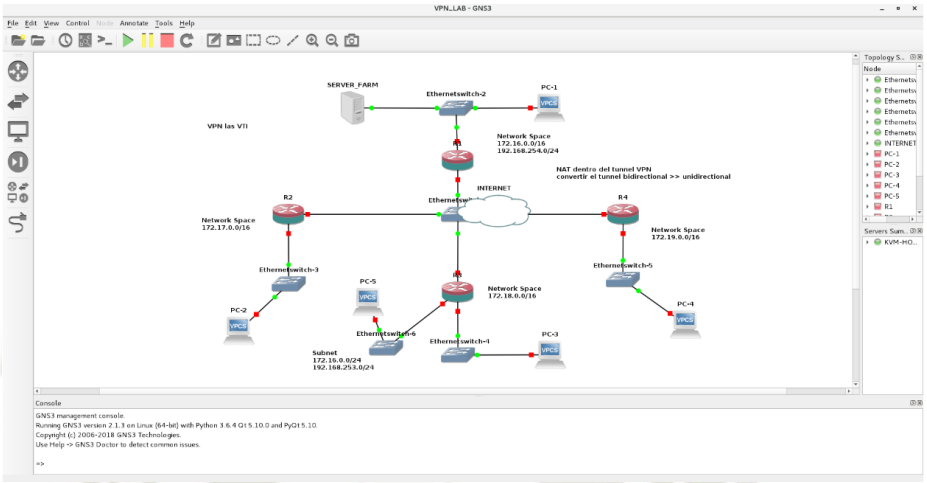

Figura 9. Ambiente de emulación de topologías de red

Fuente: Elaboración propia

2.1.23. IOUL2

Es el nombre con el que se le conoce a una herramienta de Cisco Systems.

Sirve para poder integrar imágenes de switches en un ambiente emulado.

2.1.24. Protocolo simple de administración de red (SNMP)

El Protocolo Simple de Administración de Red es un protocolo de capa 7 (aplicación) que se usa para administrar redes basadas en TCP/IP.

Este protocolo este compuesto por un conjunto de normas para gestionar la red, este permite a los administradores de red diagnosticar y gestionar dispositivos.

Este protocolo usa dos puertos para su funcionamiento los cuales son 161 y 162 sobre UDP.

Entre las principales tareas que puede realizar se encuentra la escritura y lectura de datos en equipos de infraestructura. Este protocolo puede funcionar tanto en administradores como agentes.

Los administradores son los encargados de leer y escribir la información en los agentes, mientras que los agentes son los equipos administrados, los equipos que serán monitoreados.

En los dispositivos snmp hay dos funciones principales las cuales son pooling y traps.

La función de polling cumplen con la tares de enviar consultar ejecutando operaciones síncronas de consultas.

Las traps son mensajes que envían los dispositivos snmp cuando ha habido un cambio o cuando un umbral previamente configurado haya sido sobre sobrepasado, estas operaciones se ejecutan de forma asíncrona.

Este protocolo funciona lanzando una consulta snmp a una dirección ip estática previamente configurada, pero también se requiere un parámetro adicional que es la comunidad, ésta es una cadena alfanumérica que autoriza la consulta snmp en el agente.

El protocolo simple de administración de red trabaja sobre UDP lo que significa que usa los servicios no orientados a la conexión de UDP para realizar sus operaciones.

El fin de usar el servicio no orientado a la conexión es el de no afectar el rendimiento de la red (Rodriguez, 2020).

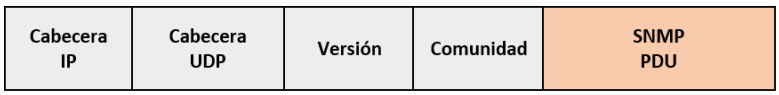

Figura 10. Estructura y composición de un paquete SNMP

Fuente: Elaboración propia

Snmp cuenta con varias operaciones las cuales son simples envíos de paquetes (peticiones) pero con respuestas o información más detallada, lo que permite que el agente snmp responda de diferentes maneras.

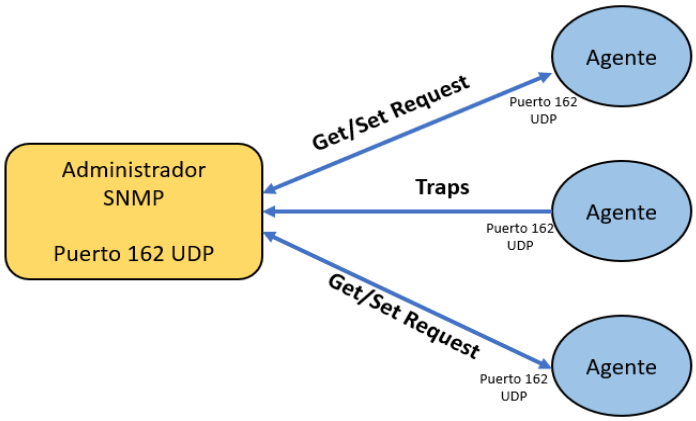

Figura 11. Mensajes SNMP

Fuente: Elaboración propia

- La petición GET permite extraer la información de una variable, el administrador envía una petición y el agente responde con la información y un valor indicando que hubo éxito en la operación.
- La petición GETnext permite extraer la información de la siguiente variable disponible en el agente según el orden alfabético de la variable que se encuentra en la MIB.

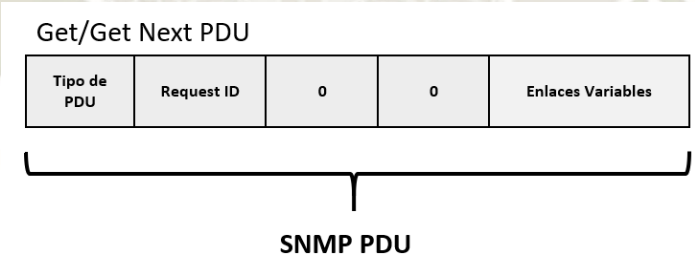

Figura 12. Estructura y composición de la PDU del mensaje Get/Get Next

Fuente: Elaboración propia

- La petición GETBulk nos permite hacer varias iteraciones entre variables. El agente retornará una respuesta con múltiples variables entrelazadas.

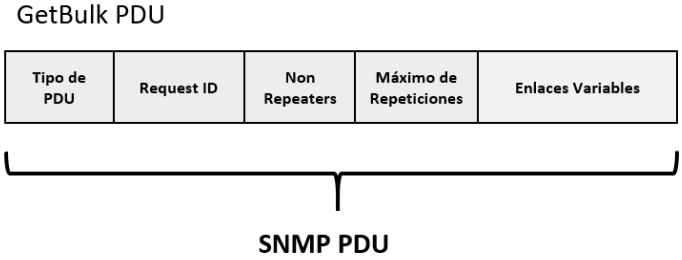

Figura 13. Estructura y composición de la PDU del mensaje GetBulk

Fuente: Elaboración propia

- La solicitud SET permite enviar información a los agentes con el fin de cambiar el valor de una variable, previamente debe haber una comunidad y un puerto preconfigurado, dicha solicitud podría considerarse critica ya que un cambio erróneo en la variable de un equipo puede generar fallas en toda una red.

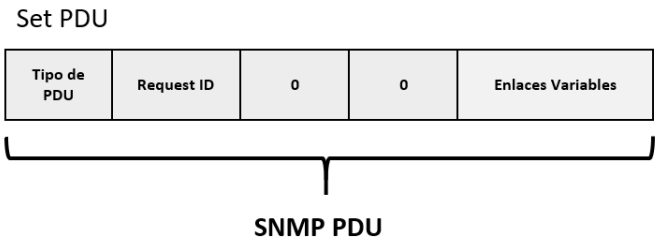

Figura 14. Estructura y composición de la PDU del mensaje Set

Fuente: Elaboración propia

- La respuesta es un paquete que puede tener varias funciones, entre las cuales destacan las pruebas de conectividad snmp, respuesta a una operación exitosa o respuesta a una operación fallida.

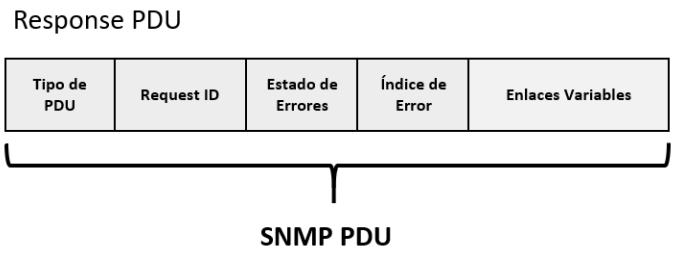

Figura 15. Estructura y composición de la PDU de la respuesta SNMP

Fuente: Elaboración propia

Las variables que se consultaran se conocen como OID (Identificador de objeto) estas variables se encuentran en una base de información gestionada (MIB) que está escrita en una sintaxis llamada ASN.1, las OID's se encuentran jerárquicamente organizadas en forma de árbol.

Las Mib's contienen información de los datos de todos los dispositivos de la red entre algunas de sus funciones se encuentran la asignación simbólica de nombres, tipos de datos, descripción y parámetros de nodos OID's.

Se cuenta con tres versiones de este protocolo las cuales son v1, v2 y v3 la cuales ofrecen diferentes mejoras que la anterior versión.

a) Snmp v1

La versión 1 fue la primera implementación que se tuvo de este protocolo, opera a través de UDP, dicha versión carece de una seguridad significativa, la autenticación de cliente se da a través de la cadena de comunidad que prácticamente podría ser un tipo de "Contraseña" la cual se transmite por la red como un texto plano.

La versión 1 no es perfecta ya que solo se creó con el fin de tener una solución temporal a la gestión de la red, se esperaba que se creen protocolos mucho más eficaces y útiles.

Esta versión no está diseñada para gestionar grandes cantidades de dispositivos lo que la limita demasiado, es por eso que se propusieron mejoras en el protocolo ya que mediante avanzaba el tiempo se creyó que se podía mejorar sin necesidad de crear un protocolo diferente.

b) Snmp v2

La versión 2 implementa notables mejoras con respecto a la versión anterior, dicha versión añade mecanismos de seguridad además una mejora en el detalle de la definición de variables, también se añaden estructuras de tablas para facilitar el manejo de datos, se añadieron mejoras en el rendimiento y la confidencialidad. Con esta versión es posible gestionar una mayor cantidad de datos. Se añadieron más funciones como GetBulkRequest y GetNextRequest con el fin de poder extraer grandes cantidades de datos en el menor tiempo posible y sin sobrecargar la red. Hubo varios problemas para aceptarlo ya que se habían creado bastantes NMS (Network Management Systems) que ya trabajaban con la versión anterior y hacer algunos cambios requeriría esfuerzo y sería muy costoso, sin embargo, terminó siendo aceptado. Se creo una subversión llamada V2c que aplicaba un par de mejoras en la versión V2, esta versión era más simple y un poco más segura que la versión inicial.

c) Snmp v3

La versión 3 de snmp es la última que se lanzó, esta versión incluyó diferentes mecanismos que mejoraron la seguridad en varios aspectos.

Se agregaron los mecanismos de seguridad como, Integridad de mensaje, Autenticación y Encriptación.

- Integridad de Mensaje: Este mecanismo nos asegura que el paquete no fue interferido y modificado en su camino hacia su destino.
- Autenticación: Este mecanismo permite identificar si los paquetes se envían desde una fuente confiable. Se hace mediante firmas digitales como MD5 y SHA1.
- Encriptación: Este mecanismo permite encriptar al paquete de tal forma que si es intervenido en su camino hacia el destino este o pueda ser leído o entendido por el atacante. Se hace como una medida de prevención. La encriptación de los paquetes se hace mediante el cifrado DES y AES para así garantizar una mayor privacidad.

Tabla 3

Propiedades de las versiones de SNMP

Propiedad	**SNMP v1**	**SNMP v2c**	**SNMP v3**
Nivel	Sin Autenticación Sin Privacidad	Sin Autenticación Sin Privacidad	Autenticación Privacidad
Autenticación	Comunidad	Comunidad	MD5 / SHA
Encriptación	No	No	DES
Traps	Incluido	Incluido	Incluido
Informes	No Incluido	No Incluido	No Incluido

Fuente: Elaboración propia

Actualmente el protocolo simple de administración de datos (SNMP) se encuentra en las mayorías de dispositivos de red existentes lo que facilita la

aplicación de esta metodología ya que lo que se quiere es abarcar la monitorización de la mayoría de los equipos de red.

2.1.25. NETFLOW

Es un protocolo desarrollado por Cisco Systems, tiene la finalidad de recolectar información de tráfico IP. Actualmente se encuentra presente en varias plataformas además de Cisco como por ejemplo equipos propietarios de Linux, Juniper, FreeBSD, etc.

Cuando se activa la función de netflow en los equipos empiezan a generar "registros netflow".

Los registros netflow contienen información que será entregada a un administrador de red que este usando el servicio de netflow, este dispositivo puede ser un servidor, una computadora, etc. El dispositivo se encargará de almacenar y procesar la información.

La información que se transmite mediante el protocolo netflow usa los servicios de UDP o SCTP.

Stream Control Transmition Protocol (SCTP) es un protocolo de comunicación que se ubica en la capa 4 del modelo OSI (Capa Transporte).

SCTP surge como alternativa a TCP y UDP puesto que este ofrece transporte confiable, control de flujo y secuenciación. SCTP además permite que los paquetes de mensajes se envíen fuera de orden, este protocolo está orientado al mensaje.

Todos los paquetes de netflow que se transmiten hacia un destino contienen una capacidad pequeña de información, es muy importante ya que en sí no envía payload hacia el destino, en cambio solo enviará porciones de información de datos estadísticos.

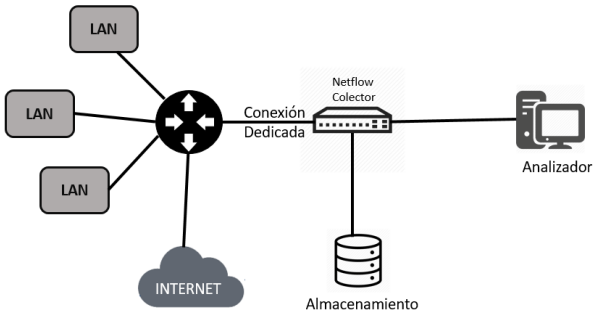

Figura 16. Estructura del funcionamiento de Netflow

Fuente: Elaboración propia

Los paquetes netflow permitirán al administrador de red conocer el origen y destino del tráfico además del origen y causas de la saturación en una red.

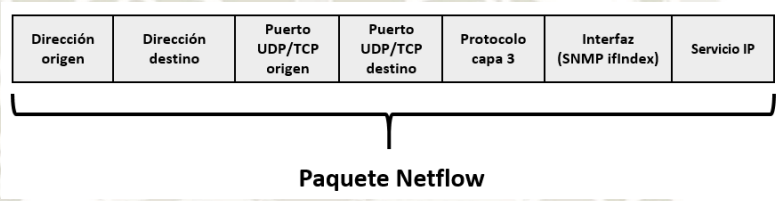

Figura 17. Estructura y composición del paquete de Netflow

Fuente: Elaboración propia

El flujo de datos es una secuencia unidireccional de paquetes que son enviados desde un origen a un destino, los puntos extremos del flujo se identifican por la dirección IP y el número del puerto usado.

La configuración típica de Netflow implica tres componentes fundamentales para un correcto funcionamiento y manejo.

1. **Exportadores de flujo de red**

   Se encarga de agregar paquetes a los flujos y exporta los registros de los flujos a múltiples recolectores, generalmente los exportadores suelen ser Switches y Routers, también pueden ser los cortafuegos.

2. **Recolectores de flujo de red**

   Este es el que se encarga de recibir los flujos de datos además de almacenarlos para luego procesarlos que son enviados por un exportador, generalmente puede ser un servidor o un pc.

3. Sistema de Análisis

Se encarga de analizar todo el flujo de red que fue almacenado en el contexto de detección de intrusos o congestión de red

Netflow cuenta con 8 versiones cada una de estas ofrece mejoras significativas que la anterior.

Netflow "exporta" los datos en datagramas UDP en cualquier formato de cualquiera de las 8 versiones, en todas las versiones los datagramas se componen por una cabecera y uno o más registros de flujos.

El primer campo de la cabecera contiene la versión del datagrama.

El segundo campo de la cabecera contiene el número de registros que el datagrama contiene. Este se usa como un índice de registros.

Los datagramas de las versiones 5,6,7 y 8 contienen un campo llamado número de secuencia, este campo es usado por aplicaciones que consumen y procesan datos de netflow entra las que podemos encontrar dispositivos colectores con la finalidad de detectar si existen datagramas perdidos o fuera de secuencia.

Los errores en netflow pueden ocurrir ya que emplea el protocolo no orientado a la conexión UDP que a su vez no es confiable para transportar los datagramas (Zambrano Montenegro, D, 2015).

2.1.26. IP SLA

Internet Protocol Service Level Agreement (IP SLA) es un protocolo que permite el monitoreo y medición de una red de computadoras. Fue desarrollado por Cisco Systems.

Tiene como finalidad reunir y monitorear información acerca de un tipo de tráfico en específico de lado a lado dentro de una red.

Con este protocolo podemos monitorear que el servicio que se le está ofreciendo a un cliente está cumpliendo los requisitos ofrecidos.

Este protocolo de monitoreo continuo de tráfico de red brinda un método confiable de análisis de red.

IP SLA ofrece diferentes operaciones para algunos protocolos existentes.

Tabla 4

Operaciones para protocolos en IP SLA

Protocolo	**Operación**
HTTP	Medición de ida y vuelta para acceder a páginas web
FTP	Medición de ida y vuelta para enviar archivos
DNS	Medición de tiempo de búsqueda de DNS
DHCP	Medición de tiempo para recuperar una dirección IP
TCP	Medición de tiempo de conexión.
UDP JITTER	Medición del retraso de ida y vuelta unidireccional y perdida de paquetes
VoIP UDP JITTER	Medición de métricas de rutas de llamadas VoIP
ICMP ECHO	Medición de retraso de ida y vuelta
UDP ECHO	Medición de tiempo de respuesta de nodos IP SLA
ICMP PATH ECHO	Medición de rutas desde un enrutador a otro nodo
ICMP PATH JITTER	Medición de variación de retraso entre paquetes a lo largo de una ruta trazada

Fuente: Elaboración propia

IP SLA consiste en un responder que puede ser un Switch o un Router que previamente configurado enviará información a un sistema administrador de red.

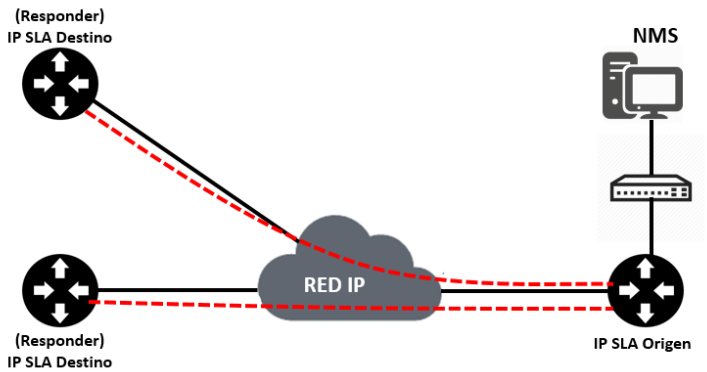

Figura 18. Estructura del funcionamiento de IP SLA

Fuente: Elaboración propia

IP SLA usa una serie de pasos para poder establecer comunicación entre el responder y el NMS.

1. El dispositivo de origen inicia la operación especificando un dispositivo de destino, una operación y un número de puerto, tal como se define en la configuración de las operaciones IP SLA.
2. El dispositivo fuente envía un mensaje de control al puerto 1967 del IP SLA Responder, con el número de puerto y la duración especificados.
3. Si la autenticación del mensaje MD5 está habilitada, la suma de comprobación MD5 se envía con el mensaje de control.
4. Si la autenticación de mensajes MD5 está habilitada, el Respondedor la verifica. Si la autenticación falla, el Respondedor devuelve un mensaje de falla de autenticación.
5. Si el dispositivo fuente no recibe una respuesta del Respondedor, retransmite el mensaje de control hasta tres veces y finalmente agota el tiempo de espera.
6. Si el Respondedor no puede procesar el mensaje de control, devuelve un mensaje de error. Si el Respondedor procesa con éxito el mensaje de control, envía una respuesta correcta al enrutador de origen y comienza a escuchar en el puerto especificado. Tenga en cuenta que el Respondedor puede responder a múltiples operaciones desde múltiples fuentes que se conectan al mismo número de puerto.
7. Si el código de retorno del mensaje de control es correcto, el dispositivo fuente envía paquetes de prueba IP SLA al Respondedor.
8. Según el tipo de operación, el Respondedor agrega marcas de tiempo en los paquetes de devolución para una medición precisa. El dispositivo fuente realiza el cálculo de las mediciones del tiempo de respuesta.
9. Después de responder a los paquetes de prueba o después de que expire el temporizador de duración del mensaje, el Respondedor deshabilita el puerto de monitoreo especificado (Lightsys, 2012).

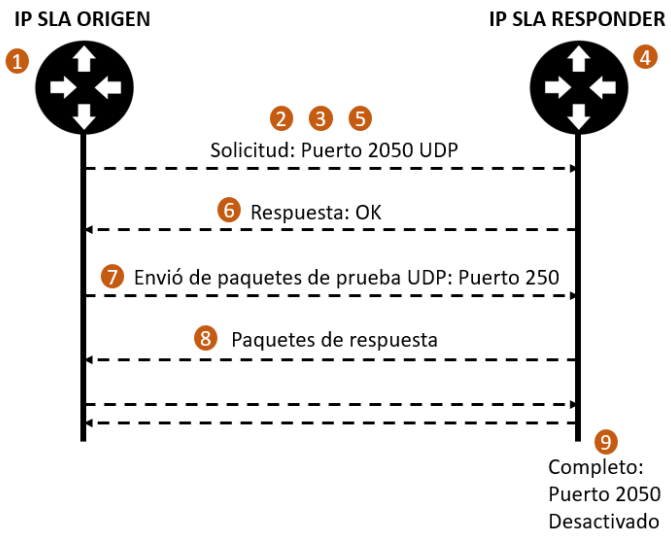

Figura 19. Pasos para establecer un canal de comunicación mediante IP SLA

Fuente: Elaboración propia

2.2. CONCEPTOS DE PROGRAMACIÓN

2.2.1. Pysnmp

PySNMP es una librería multiplataforma programada en el lenguaje de programación Python.

Cuenta con un motor SNMP completamente funcional el cual puede actuar en los roles de agente, administrador, Proxy.

Es compatible con las tres versiones de protocolo SNMP y tiene soporte para trabajar con IPV4 e IPV6.

La librería sigue muy de cerca todas las características detalladamente que ofrece el protocolo SNMP en sus tres versiones, para que de esta forma se pueda aprovechar en casi un 100% todas las operaciones que se pueden realizar.

La versión que se usará es la 4.4 que es la estable y funciona con Python 2.4 a 3.7.

Además, una vez que es descargada trae consigo un conjunto de herramientas de líneas de comando con operaciones básicas de snmp como get, set y walk, lo cual permitirá un trabajo mucho más fácil para implementar lo que se desea.

Esta librería es gratuita y de código abierto, su código fuente está alojado en un repositorio de GitHub.

La librería ofrece un amplio soporte significativo en el aspecto relacionado con la seguridad, ya que la información que se transmite desde un administrador hacia un agente y viceversa está encriptada, además posee la autenticación de los datos.

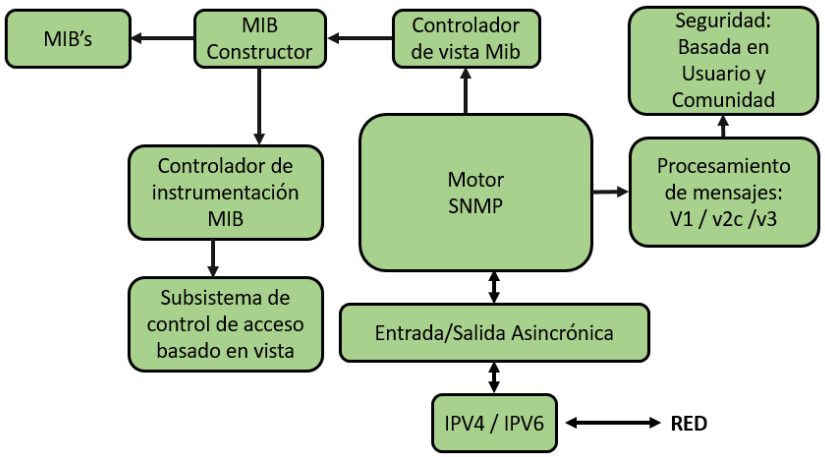

Figura 20. Arquitectura de la librería PySNMP

Fuente: Elaboración propia

Los componentes internos de PySNMP son:

- Motor SNMP

  Tiene la función de controlar los demás componentes de la librería, es muy difícil de reconfigurar si se desea hacer cambios.

- Subsistema de transporte

  Se usa con el fin de enviar y recibir paquetes SNMP por la red. El subsistema de E/S consta de un Despachador abstracto y una o más clases de transporte abstractas. SNMP utiliza con frecuencia el transporte UDP, pero también son posibles otros.

- Mensajes y Despachador de PDU

  En este apartado se procesan los mensajes SNMP. Sus principales funciones se basan en el envío de PDU desde aplicaciones SNMP a través de varios subsistemas hasta el Despachador, y pasar mensajes SNMP procedentes de la red a aplicaciones SNMP

- Módulos de procesamiento de mensajes:

  Manejan las operaciones de protocolo a nivel de mensaje para las versiones presentes y posiblemente futuras del protocolo SNMP. Lo más importante es que incluyen análisis y construcción de mensajes y posiblemente invocar servicios de seguridad cuando sea necesario.

- Módulos de seguridad:

  Realizan la autenticación y/o encriptación de mensajes.

- Subsistema de Control de Acceso:

  Es el encargado de autorizar el acceso remoto a los objetos gestionados. Solo se usa cuando se usa el rol de agente.

- Colección de MIB:

  Conjunto de bases de datos y objetos los cuales son fundamentales para consultar información en los dispositivos gestionados (Recio Recio, J, 2010).

2.2.2. Xlsxwriter

Es una librería hecha en Python que permite la escritura y lectura de letras, números, formulas e hipervínculos en varias hojas de cálculo en archivos Excel.

Además, permite trabajar con gráficos, imágenes, autofiltros y muchos más.

Tiene un alto grado de fidelidad con los archivos que son producidos con Excel. En la mayoría de los casos los archivos producidos por esta librería son 100 por ciento equivalentes a archivos generados por Microsoft Excel.

Esta librería es compatible con Python desde la versión 2.7 y solo usa bibliotecas estándar.

Cuenta con una amplia documentación, ejemplos y pruebas. Lo que facilita su comprensión.

Una de las desventajas más considerables que se tiene es el no poder leer ni modificar archivos existentes de Microsoft Excel. Este es un grave problema ya que nosotros deseamos escribir datos en archivos Excel e ir guardándolos en cada momento lo que implicaría el tener que leer y escribir este archivo (Castro Flores, C. A., Guillen Asencio, J. M., & Riera Barraza, J. J, 2010).

```
hello World.py
1  import xlsxwriter
2
3  workbook = xlsxwriter.Workbook('hello.xlsx')
4  worksheet = workbook.add_worksheet()
5
6  worksheet.write('A1', 'Hola Mundo')
7
8  workbook.close()
```

Figura 21. Implementación de código para poder crear un archivo con Xlsxwriter

Fuente: Elaboración propia

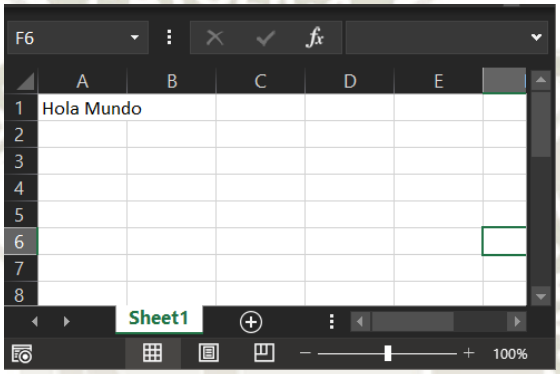

Figura 22. Resultado de la ejecución del código de la figura 21

Fuente: Elaboración propia

2.2.3. Openpyxl

Al igual que la xlsxwriter esta librería está diseñada en el lenguaje de programación Python, además es multiplataforma.

Está diseñada para escribir archivos Excel a partir de la versión 2010 en formatos xlsx, xlsm, xltx y xltm.

Fue creada a falta de una librería que sea capaz de leer y escribir archivos de forma nativa desde el lenguaje de programación Python.

Esta librería abarca muchas funciones como manejo de gráficos, formato de números, condicionales, etc. Además, tienen incluido un tokeneiser para analizar formulas en archivos Excel.

Una desventaja considerable de esta librería es al momento de manejar archivos demasiado grandes pues es bastante lento para procesarlos (Cajamarca, M, 2019).

```
hello World.py
1  from openpyxl import Workbook
2  wb = Workbook()
3
4  ws = wb.active
5
6  ws['A1'] = "Hola Mundo"
7
8
9  wb.save('hello.xlsx')
```

Figura 23. Implementación de código para poder crear un archivo con Openpyxl

Fuente: Elaboración propia

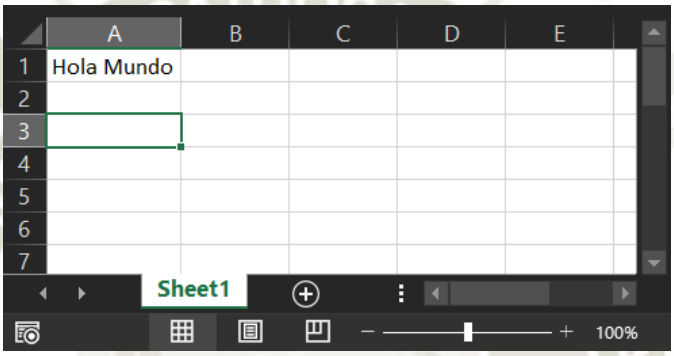

Figura 24. Resultado de la ejecución del código de la figura 23

Fuente: Elaboración propia

2.2.4. Java

Java es un lenguaje de programación basado en clases y orientado a objetos, fue desarrollado por Sun Microsystems.

Además, es conocida como una tecnología que trae consigo una máquina virtual con el propósito de compilar código fuente en cualquier plataforma ya sea hardware o software.

Uno de los objetivos principales que tiene el lenguaje es que el código fuente sea escrito una sola vez y pueda ser ejecutado en cualquier lugar. Por lo tanto implica que el código sea compilado una sola vez y pueda ejecutarse las veces que se desea en cualquier computador que admita java.

Las aplicaciones programadas en java generalmente se compilan en bytecode's que pueden ejecutarse en cualquier máquina virtual conocida como (JVM) independientemente de la arquitectura informática.

Entre las características de java encontramos las siguientes:

- Simple y ordenado
- Orientado a objetos
- Robusto y seguro
- Portátil
- Con ejecuciones de alto rendimiento
- Interpretado
- Dinámico
- Distribuido
- Multihilo
- De arquitectura neutral
- De alto rendimiento
- Recolección de basura

Java tiene la característica de ser interpretado y compilado.

Java incluye bibliotecas para varias funciones que son indispensables. Algunas bibliotecas principales que trae consigo este lenguaje son:

- IO/NIO
- Networking
- Genérica
- Concurrencia
- Programación funcional
- Colecciones
- Seguridad
- Internacionalización y localización
- Java Database Connectivity (JDBC)

- JMX

Java además genera una carga significativamente alta ya que los paquetes de red suelen ser un poco grandes, y dado el tráfico que genere en la red por las consultas que se haga puede generar inconvenientes.

Además, java cuenta con una librería para poder realizar operaciones Snmp llamada SNMP4J.

2.2.5. Python

Python es un lenguaje de programación de alto nivel Creado por Guido van Rossum.

Es un lenguaje de propósito general lo que significa que no está orientado a un fin concreto

Este lenguaje es interpretado lo que significa que no se compila el código fuente a código de maquina si no que existe un interpretador que es el encargado de ejecutar el programa basándose en el código fuente directamente.

Es multiplataforma lo que permitirá poder ejecutarse en diferentes sistemas operativos.

Entre sus objetivos principales se encuentra ayudar a los programadores a escribir código fuente claro y lógico para proyectos de pequeña y gran escala.

Python se escribe dinámicamente y al igual que java permite la recolección de basura.

Es un lenguaje de programación que admite múltiples paradigmas de programación, incluidas la programación orientada objetos y la funcional.

Entre sus características principales encontramos:

- Lenguaje interpretado
- No compilado
- De tipado dinámico
- Fuertemente tipado
- Multiplataforma

- Multiparadigma

Los desarrolladores recomiendan Python por la facilidad que se tiene al momento de desarrollar aplicaciones web gracias a frameworks que permiten la creación de las mismas.

Algunas frameworks que trabajan con Python son:

- Pyramid
- Bottle
- Django

Además, la llegada del Big Data y la Inteligencia Artificial hicieron que Python se vuelva el lenguaje de programación favorito para la creación de algoritmos.

Python cuenta con una carga baja con respecto al rendimiento de red dado que los paquetes son de menor tamaño.

Cuanta cpn una librería para el procesamiento de consultar SNMP llamada PySNMP.

2.2.6. Php

Hypertext Preprocessor (Php) es un lenguaje de programación de propósito general del lado del servidor que se usa para el desarrollo de aplicaciones web. Creado por Rasmus Lerdorf.

Generalmente el código php se procesa en un servidor web por un intérprete php.

Es un lenguaje multiplataforma, lo que permite que se ejecute en diferentes sistemas operativos.

Generalmente las consultas que se hacen al programa php en un servidor web son devueltas mediante el protocolo HTTP.

Php no solo sirve para crear aplicaciones web, fuera de este contexto se pueden crear aplicaciones graficas además de poder crear aplicaciones que permitan el control de bot´s o drones.

Entre sus principales características encontramos:

- Estabilidad

- Velocidad
- Seguridad
- Simplicidad
- Multiplataforma

El código de programación de este lenguaje es invisible para el navegador web y al cliente.

Tiene una capacidad de conexión bastante grande con motores de bases de datos, entre los cuales destacan Mysql y PostgreSQL.

Además, no requiere de definición de tipo de variables, el contenido de las variables se evalúa en tiempo de ejecución.

La carga que origina este lenguaje en la red puede ser significativa, todo depende de la cantidad de consultas que se hagan.

2.2.7. C++

Es un lenguaje de programación de propósito general, facilita la adaptación a los recursos limitados como los que se encuentran en arquitecturas de software.

C++ es un lenguaje de tipado estático, lo que significa que los tipos de variables se declaran de manera explícita y se determinan en tiempo de ejecución. Además, es un lenguaje de programación multiparadigma.

Es un lenguaje de programación con características de bajo nivel.

Entre sus características principales tenemos:

- Alto rendimiento
- Multiplataforma
- Extendido
- Rápido
- Didáctico

Para que C++ pueda enviar paquetes a través de la red se requiere del framework .NET

Microsoft .Net es una respuesta al entorno de desarrollo de aplicaciones orientadas a la web, dado que C++ no cuenta con esta caracteristica, Microsoft. net permite enviar paquetes a través de la red. Además, .NET genera una carga de red considerable si se desea desarrollar aplicaciones para entornos web

2.2.8. Pycharm

Pycharm es un entorno de desarrollo utilizado para la programación de computadores, fue creado con la finalidad de poder utilizarse para programar en el lenguaje de programación Python.

Pycharm ofrece una amplia gama de herramientas entre las cuales están depuraciones avanzadas, autocompletado de código, inspección, etc. Además, permite la interacción con Django y Data Science con Anaconda.

Pycharm es una herramienta multiplataforma , con versiones de Windows , macOS y Linux . La versión comunitaria trae consigo las herramientas necesarias para poder realizar programación.

2.2.9. Spyder

Spyder es un entorno de desarrollo interactivo con diferentes variables en múltiples sistemas operativos, fue diseñado con la finalidad de poder trabajar con el campo de ciencia de datos. Está programado en su totalidad en Python.

Cuenta con múltiples características entre las cuales encontramos puntos de interrupción de ejecución, visualización de variables, edición de variables, navegación por bloques, visualización de gráficos, explorador de archivos, etc.

2.3. CONCEPTOS DE APRENDIZAJE AUTOMATICO

2.3.1. Preprocesamiento de Datos

Es una etapa en el proceso de análisis de datos, consiste en limpiar los datos, así como su transformación, reducción e integración de los mismos, preparándolos así para la siguiente etapa que consiste en aplicar las operaciones necesarias para poder generar resultados más adelante.

Hay varias técnicas para preprocesar los datos, a continuación, se detallan estas:

Tabla 5

Técnicas de preprocesamiento de datos

Técnica de preprocesamiento	**Operaciones**
Transformación de datos	Normalización Smmothing Agregación Generalización
Limpiado de datos	Valores desaparecidos Incoherencia de datos Ruidos de datos
Reducción de datos	Cubos de agregación Reducción dimensional Comprensión de datos Selección de propagación
Integración de datos	Fuentes de información Resolución de problemas y codificación Integración de datos de diferentes tablas

Fuente: Elaboración propia

2.3.2. Regresión

La regresión es un modelo matemático generalmente empleado en estadística, se usa para la aproximación de la relación entre una variable dependiente y una variable independiente y un término aleatorio (SAS Institute, 2019).

A continuación, la explicación de la ecuación de regresión lineal:

$$Y_t = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$$

Figura 25. Ecuación de regresión

Fuente: Elaboración propia

y_t: Variable dependiente.

$x_1, x_2, \dots, x_p$: Variables independientes.

$\beta_0, \beta_1, \beta_2, \dots, \beta_p$: Parámetros.

2.3.3. Clasificación

Es una técnica para analizar los datos comúnmente utilizado en inteligencia artificial, se usan generalmente cuando el resultado que se va a obtener es una etiqueta discreta.

Es muy útil cuando se sabe que el resultado caerá en un conjunto finito de resultados probables.

2.3.4. Clustering

Es una técnica que implica la recolección de objetos de datos y agrupación de los mismo con características similares ya sea grupo, clase, categoría y se diferencian de objetos de otros clústers.

El Clustering es una de las formas más conocidas de aplicar aprendizaje automático no supervisado.

Este campo estudia las interacciones entre las máquinas y el lenguaje humano.

2.3.5. Deep Learning

Es parte de un conjunto de técnicas del aprendizaje automático basado en el uso de las redes neuronales artificiales, este puede ser supervisado, semi supervisado o sin supervisar (Herran Arias, 2019).

2.3.6. Redes neuronales Artificiales

Las redes neuronales artificiales abarcan una serie de algoritmos que sirven para reconocer patrones en un conjunto de datos.

Las redes neuronales artificiales tratan de emular el comportamiento de los seres humanos, específicamente el comportamiento para tomar decisiones. Lo que permite a las computadoras hacer cálculos casi precisos y tomar decisiones en base a los cálculos (Herran Arias, 2019).

2.4. CONCEPTOS DE PROCESOS METODOLOGICOS

2.4.1. METODOLOGÍA CRISP-DM

La metodología realizada para este trabajo de investigación se basa en la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining), esta metodología está diseñada para proporcionar normalización en todo el ciclo de vida de un proyecto de análisis de datos. La metodología CRISP-DM considera el proceso de análisis de los datos como proyecto profesional. La secuencia de las fases de esta metodología no es unidireccional, el resultado final de cada fase determina con que fase o tarea en específico se debe seguir (Gallardo Arancibia, J, 2009).

Esta metodología es de naturaleza cíclica, el proyecto a implementar no acabará si la solución no se ve desplegada.

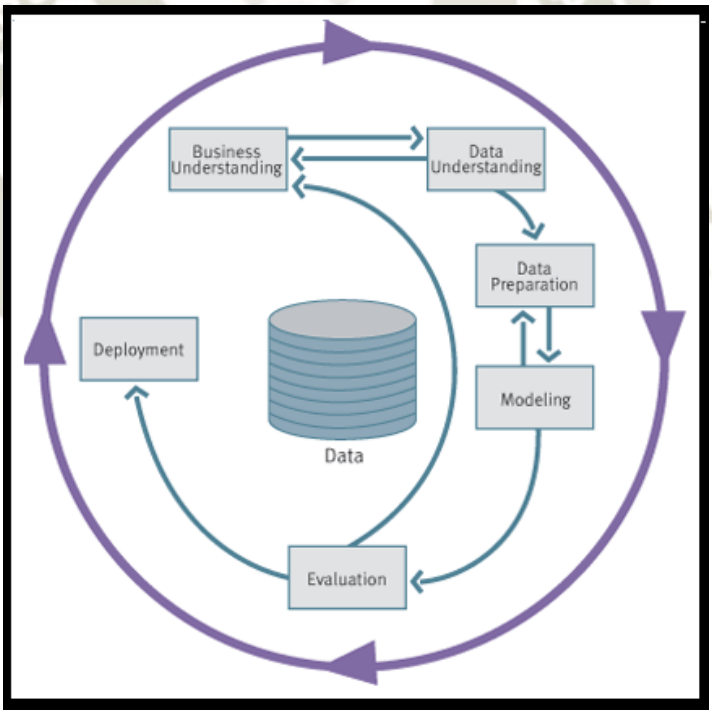

Figura 26. Fases de la metodología CRISP - DM

Fuente: Elaboración propia

a) Fases de la metodología

La metodología CRISP-DM consta de 6 fases es su proceso cíclico bidireccional. Cada fase contiene actividades que mientras van terminando van abriendo las puertas para que se realicen nuevas tareas de la misma o diferentes fases.

A continuación, se detallan las fases de la metodología CRISP-DM con las tareas específicas que se deben realizar:

1. **Comprensión del negocio**

   Esta fase inicial se enfoca en la comprensión de las necesidades del negocio, luego se procede con la identificación del problema de minería de datos para luego diseñar un plan preliminar para alcanzar los objetivos deseados.

2. **Comprensión de los datos**

   En esta fase se inicia con la recolección de los datos a analizar, luego se procede con la familiarización de los datos, además se procede a verificar si las variables extraídas son las que se necesitan para poder entender el problema.

3. **Preparación de los datos**

   Esta fase prepara los datos para que la fase de modelamiento los reciba sin problemas para poder realizar las operaciones necesarias, en esta fase se hace el preprocesamiento de los datos obtenidos además de la transformación y la limpieza de los mismos.

4. **Modelamiento**

   En esta fase se seleccionan las técnicas de modelamiento que se aplicaran a los datos previamente preparados. Cabe mencionar que entre más técnicas se seleccionen será más factible el resultado final, las diferentes técnicas aplicadas proporcionarán diferentes tipos de resultados, el que más se acerque a resultado deseado se tomará para luego proceder con la evaluación.

5. **Evaluación**

   Para esta fase se han debido generar uno o varios modelos que parecen haber alcanzado una tasa alta de calidad en sus resultados.

   Antes de proceder al despliegue es importante que el resultado del modelo elegido de una respuesta a la mayoría de los problemas observados en un principio, si es que hay alguno que pudo pasar

desapercibido o no se tomó en cuenta otro es recomendable retornar a la fase de comprensión del problema.

6. **Despliegue**

Muchos creen que la generación del modelo es la etapa final, pero no. A veces el modelo aplicado a los datos previamente recolectados nos permite aumentar el conocimiento de los mismos. Por lo tanto, se tendrá que reevaluar las operaciones a realizar.

El despliegue puede ser tan simple como la elaboración de un informe final con los resultados obtenidos o tan complejo como la implementación automática de un proceso (Gallardo Arancibia, J, 2009).

Tabla 6

Fases y actividades de la metodología CRISP-DM

Fases	**Actividades**
Comprensión del Negocio	• Determinar objetivos de negocio • Evaluación de la situación • Establecer Objetivos de minería de datos
Comprensión de los datos	• Generar plan de proyecto • Recopilación de los datos • Descripción de datos • Exploración de datos • Verificación de datos
Preparación de los datos	• Selección de datos • Limpieza de datos • Construcción de datos • Integración de datos • Formateo de datos
Modelado	• Selección de la técnica de modelado • Diseño de la evaluación • Construcción del modelo • Evaluación del modelo
Evaluación	• Evaluación de resultados • Revisión del proceso • Determinación de siguientes pasos
Despliegue	• Desplegar el plan • Monitorear y mantener • Generación de informe final • Revisión de proyecto

Fuente: Elaboración propia

CAPITULO III

3. DISEÑO E IMPLEMENTACIÓN DE LA PROPUESTA METODOLOGICA

3.1. DESCRIPCIÓN DE LA PROPUESTA METODOLÓGICA

La propuesta metodológica descrita a continuación fue diseñada basada en CRISP-DM, con el fin de fomentar la interoperabilidad de las herramientas a través de todo el proceso de minería de datos, además de que es posible una mejor gestión y planificación del proyecto a aplicar.

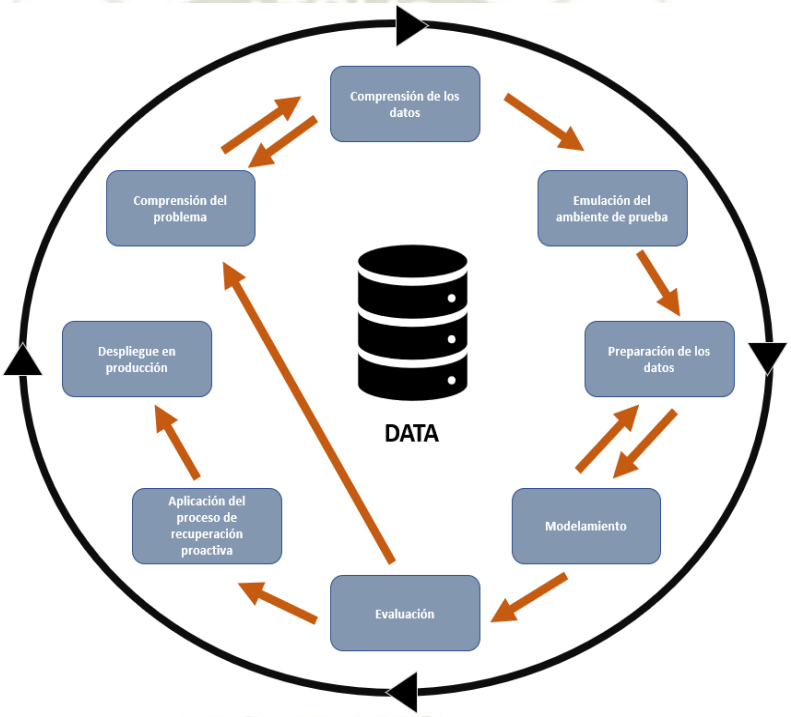

Figura 27. Fases de la metodología propuesta

Fuente: Elaboración propia

Esta metodología sigue un proceso cíclico bidireccional, cada fase de la metodología planteada tiene como finalidad dar un resultado de calidad para que la siguiente tarea o fase a realizar tenga los elementos adecuados para poder desempeñar su operación.

3.2. FASES DE LA PROPUESTA METODOLÓGICA

La metodología consta de 8 fases las cuales no podrán ejecutar sus respectivas tareas a menos que la fase anterior haya culminado todas sus actividades.

Se realiza esta operación con el fin de obtener un buen funcionamiento y aplicación de la metodología.

A continuación, se describen cada una de las fases de la metodología planteada.

1. **Comprensión del problema**

   Posiblemente la fase más importante ya que en esta fase se identificará el problema que ocasiona la incertidumbre y se planificará el resto del proyecto para llevar a cabo una solución capaz de beneficiar al negocio afectado.

   En esta fase se da la comprensión de los objetivos y requerimientos del proyecto para solucionar el problema.

   Además, se identifican los costos al tratar de aplicar el diseño de infraestructura que nos permitirá extraer la información requerida.

2. **Comprensión de los datos**

   En esta fase se procede con la identificación de variables que se van a extraer y almacenar.

   Se identifican los protocolos pueden servir para extraer datos.

   Además, se investiga a cerca de una librería que facilitar la comunicación del agente extractor de datos con los equipos de donde se van a extraer los datos.

   Finalmente se genera un script capaz de extraer los datos de los equipos de red y almacenarlos para su futuro análisis.

   Entre esta fase y la anterior hay un camino bidireccional ya que, si no se identifican las variables necesarias, los protocolos que se usaran, así como las librerías de programación se tendrá que realizar un nuevo análisis de la comprensión del problema. De esta forma nos aseguramos de tener todo en correcto funcionamiento una vez que se vayan a extraer los datos más adelante.

3. Emulación del ambiente de prueba

Es muy importante la emulación de un ambiente de pruebas ya que aquí se podrán realizar pruebas de conectividad y extracción de datos entre el script generado anteriormente y los equipos de red.

Además, se podrán identificar errores que puedan poner en riesgo los equipos en producción.

Esta fase tiene el fin de optimizar los recursos generados anteriormente hasta que se tenga una seguridad casi al 100 por ciento de que una vez que se aplique el script de extracción de datos en un ambiente real este no falle.

Es muy importante que se hagan todas las pruebas necesarias en esta fase ya que si hay algún problema que haya pasado desapercibido ocasionará algunos inconvenientes más adelante cuando se hagan las pruebas en equipos de producción.

4. Preparación de los datos

En esta fase se procede con la extracción y almacenamiento de los datos que se van a necesitar para que en la siguiente fase se aplique el modelamiento de los mismos.

Además, se aplica el preprocesamiento de los datos etapa que permite su limpieza, transformación y reducción. Dejándolos listos para la siguiente fase.

5. Modelamiento

En esta fase se probarán varios algoritmos de aprendizaje automático, con el fin de poder obtener uno o algunos que se acerquen a la predicción casi exacta de lo que se quiere conocer.

Esta fase también tiene un camino bidireccional con la fase anterior ya que si no se hizo un correcto preprocesamiento de datos anteriormente puede causar dificultad para aplicarles los algoritmos que se piensan usar.

6. Evaluación

En esta fase se procede a evaluar los resultados obtenidos con los algoritmos de aprendizaje automático puede que los resultados sean buenos y que logren predecir en gran parte alguna falla que se va a producir o puede que no ya que pudo haber un problema al inicio al momento de analizar las variables que serían necesarias para el correcto análisis.

Por lo tanto tiene una conexión con la primera fase que es comprensión del problema, hasta este punto habrá que repetir de la fase 1 a la 6 hasta obtener los resultados esperados.

7. Aplicación del proceso de recuperación proactiva

Una vez que se tenga conocimiento del algoritmo que logro predecir con gran exactitud el problema se pasa a aplicar esta fase, la mencionada fase permitirá solucionar los problemas que se estén presentando en la red mediante el uso de un protocolo de comunicación que permita configuración de equipos.

Cabe resaltar que hasta el momento los pasos fueron puestos en marcha en un ambiente emulado.

Si todos los pasos anteriores tuvieron éxito y ya no hay nada que mejorar se procederá a la fase más crítica.

8. Despliegue en producción

Una de las fases más críticas ya que se pondrá en marcha todo lo que se probó en el ambiente emulado, se supone que si nos encontramos en esta fase es porque se hicieron todas las pruebas y se emularon ciertos problemas que podrían ocurrir.

Se procederá a aplicar todo lo anterior realizado en el ambiente de producción, en tiempo real. Con el fin de empezar a detectar, predecir y solucionar los problemas futuros que se aproximen (Arias Paredes, Á, 2019).

Tabla 7

Fases y actividades de la propuesta metodológica

N°	Fase	Actividades
1	Comprensión del problema	• Determinación de objetivos • Evaluación de la situación • Identificación de las fallas • Determinar objetivo del aprendizaje automático
2	Comprensión de los datos	• Propuesta del lenguaje de programación • Propuesta de protocolo para extraer datos • Propuesta del IDE de desarrollo • Búsqueda y elección de librerías • Identificar variables • Describir datos a extraer
3	Emulación del ambiente de prueba	• Desarrollo de script • Búsqueda de emulador gráfico • Búsqueda de IOS • Diseñar de topología • Configuración de equipos
4	Preparación de los datos	• Extracción de datos • Almacenamiento de datos • Preprocesamiento de datos
5	Modelamiento	• Elección de técnicas de aprendizaje automático • Aplicación de las técnicas elegidas
6	Evaluación	Evaluación de resultados

7	Aplicación del proceso de recuperación	• Verificación de parámetros a modificar • Aplicación del proceso de recuperación • Identificación de equipos
8	Despliegue en producción	• Aplicación de configuración simulada • Monitoreo del levantamiento de la operación

Fuente: Elaboración propia

Tabla 8

Resultado de las actividades aplicadas a cada fase

Fase	Resultado
1	Descripción de la situación que se afronta y problema en específico a solucionar.
2	Creación del script con las funcionalidades requeridas.
3	Levantamiento de topología con la configuración de equipos virtuales de red.
4	Hoja de cálculo con datos preprocesados.
5	Lista de algoritmos aplicados con resultados de cada uno.
6	Elección de el o los algoritmos adecuados.
7	Recuperación proactiva de los equipos afectados por las fallas.
8	Logro satisfactorio de la predicción de la falla y la recuperación proactiva.

Fuente: Elaboración propia

3.3. APLICACIÓN DE LA PROPUESTA METODOLÓGICA

3.3.1. Comprensión del problema

a) Determinación de objetivos

Los objetivos principales se centran en la solución de fallas en equipos de capa 2 y la recuperación proactiva de los mismos.

b) Evaluación de la situación

Se cuentan con tres equipos de capa 2 (Switches) de la serie 2960 de la marca Cisco Systems ubicados en un laboratorio de redes conectados de forma cíclica uno con otro. A los equipos de comunicación se encuentra conectadas algunas computadoras que envían datos entre ellas a través de la red.

La red además cuenta con un equipo de capa 3 (Router) de la serie 2900 que permite el enrutamiento hacia otros nodos dentro de la red y hacia el exterior.

Dicha red en ciertas horas se empieza a saturar ya que al parecer no cuenta con la configuración adecuada.

La solución que se da cuando ocurre la saturación consiste en el apagado total de todos los equipos que componen la red.

Al apagar todos los equipos que componen la red afectan a todos los demás equipos que estaban haciendo uso de este servicio.

c) Identificación de las fallas

En esta etapa se hace varias pruebas para identificar el origen del problema, conectando computadores personales directamente al equipo de capa 3 (Router) y proporcionándole una dirección ip estática, con la finalidad de identificar si la saturación de red tiene algo que ver con el proveedor de servicios.

Una vez hecho el descarte nos percatamos que no hay ningún problema en el servicio de internet que se está ofreciendo.

Se da por hecho que las fallas que se están presentando ocurren dentro de la red diseñada.

d) Determinación del objetivo del aprendizaje automático

Con esta etapa se toma la decisión de aplicar algunas técnicas de aprendizaje automático para detectar los errores anticipadamente, con la finalidad de degradar la calidad de servicio de la red.

Además, otro de los objetivos será la rápida recuperación de las fallas que puedan estar presentando los equipos de infraestructura de red que ofrecen el servicio.

El fin de esta actividad es que se predigan con anticipación las fallas que ocurrirán y se pueda solucionar de manera rápida y eficaz los errores que se vayan a presentar. Si el problema ya ha ocurrido entra en acción el fin de la metodología que sería solucionar de manera inmediata el problema que se está presentando.

Con todos los objetivos bien definidos y las fallas identificadas repodemos pasara a la siguiente fase de la metodología.

Es importante haber identificado de manera correcta los factores principales en esta fase, ya que en base a este análisis las demás fases se basarán para tratar de llegar al objetivo deseado (Herran Arias, 2019).

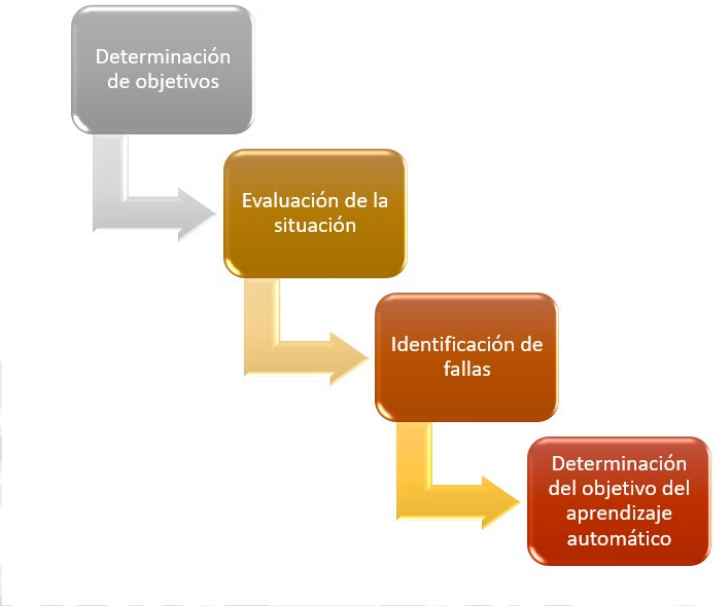

Figura 28. Actividades de la primera fase de la metodología propuesta

Fuente: Elaboración propia

3.3.2. Comprensión de los datos

a) Propuesta del lenguaje de programación

Se requiere de un lenguaje de programación que permita trabajar con protocolos de red, además de poder ser multiplataforma y fácil de utilizar.

Se realizó la búsqueda del lenguaje de programación más adecuado para poder trabajar y se eligieron 4 siendo los cuales son Java, Python, Php y C++.

A continuación, se detalla un resumen de la información necesaria de los lenguajes de programación que se investigaron:

Tabla 9

Características de los lenguajes de programación evaluados

Lenguaje	**Paradigma**	**Sistema de tipos**	**Tipo**	**Seguridad**	**Librerías SNMP**
Java	Orientado a objetos	Fuerte, estático	Compilado	Media	Si
Python	Multiparadigma	Fuertemente tipado, dinámico	Interpretado	Alta	Si
Php	Multiparadigma	Dinámico, débil	Interpretado	Baja	Si
C++	Multiparadigma	Fuertemente tipado	Compilado	Alta	Si

Fuente: Elaboración propia

Tabla 10

Carga de red que origina cada lenguaje evaluado

Lenguaje	**Carga de red**
Java	Mayor
Python	Menor
Php	Media
C++	Mayor

Fuente: Elaboración propia

Figura 29. Interés por los lenguajes de programación

Fuente: Guru99

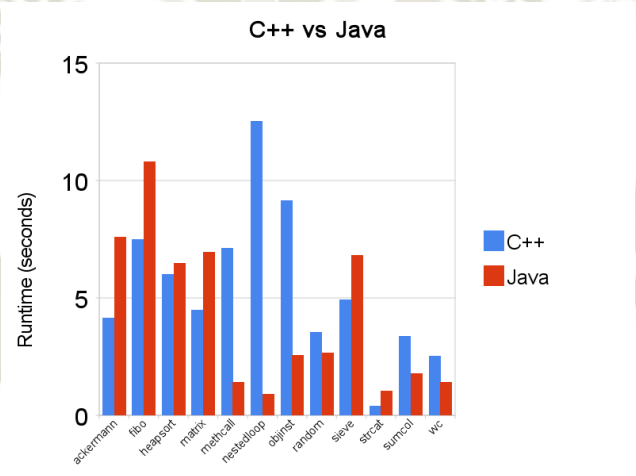

Figura 30. Tiempo de ejecución entre Java y C++

Fuente: UnPocodeJava

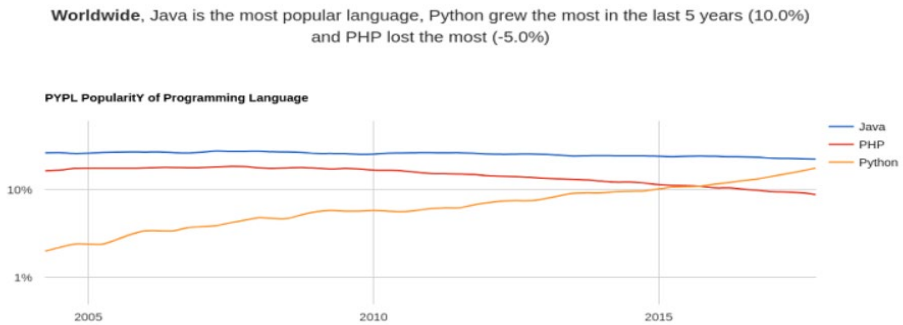

Figura 31. Lenguajes más populares en los últimos 5 años

Fuente: Pypl, 2015

- La comunidad de python crece cada día más y más este es un aspecto positivo ya que las librerías programadas en este lenguaje tendrán un amplio soporte por muchos años más.
- En cuanto a la comunidad de Java, esta se mantiene ya que es un lenguaje antiguo y uno de los más preferidos en sus inicios, aún tiene mucho soporte para los siguientes años.
- Php va en picada, ya que dejó de usarse para hacer operaciones en el lado del servidor y está siendo sustituido por Python.
- Hoy C++ cuenta con una gran comunidad ya que tiene mucho tiempo de antigüedad, el inconveniente es que se necesita del framework .NET para poder realizar proyectos web.
- Lo más recomendable es usar php para proyectos web.

Tabla 11

Comparación de los lenguajes elegidos

Parámetros	**Python**	**Php**	**Java**	**C++**
Facilidad de aprendizaje	Lenguaje de propósito general, fácil de aprender	No fue desarrollado como un lenguaje de propósito general. Es difícil de aprender	Lenguaje de propósito general, fácil de aprender	Lenguaje de propósito general
Soporte Comunitario	La comunidad sigue creciendo	Tiene una gran cantidad de soporte comunitario ya que fue muy popular en un tiempo	Su comunidad ofrece un gran apoyo ya que es antiguo	Tiene una amplia comunidad ya que lleva tiempo de creación y uso.
Soporte para librerías	Soporte de biblioteca excepcionalmente bien desarrollado para casi todos los tipos de aplicaciones.	Este es un aspecto que va en contra de PHP, pero dispone de Packagist que es un repositorio de paquetes	Soporte de biblioteca excepcionalmente bien desarrollado para casi todos los tipos de aplicaciones.	Contiene bibliotecas estándar que trae consigo.
Velocidad	Rápido	Muy Rápido	Rápido	Rápido

Fuente: Elaboración propia

Como se observa en la tabla Python y Java poseen muchos pros en comparación que PHP y C++, cabe resaltar que el soporte a las librerías es un aspecto muy importante ya que se trabajará con estas para poder realizar operaciones más adelantes como consultas de datos SNMP o análisis de datos para lo cual las librerías deben tener un soporte continuo en caso se necesite hacer cambios luego de su implementación y puesta en marcha en producción.

En el caso de C++ se va a requerir el framework .NET para poder realizar aplicación web o scripts que permitan mandar información a través de la red

De esta manera descartamos Php y C++ de la lista.

Tabla 12

Comparación entre Python y Java

Tecnología	**Python**	**Java**
Sintaxis	Fácil de aprender	Difícil de aprender
Performance	Relativamente más lento que java	Muy rápido
Multiplataforma	Si	Si con JVM
Librerías de aprendizaje automático	Tensorflow Pandas Scikit learn Numpy Scipy keras	Weka Mallet Deeplearning4j

Fuente: Elaboración propia

Dado que Python se considera como un lenguaje para principiantes, no tiene una curva de aprendizaje pronunciada, e incluso un desarrollador con conocimientos básicos puede trabajar con él.

Además, los desarrolladores tampoco tienen que pensar en las limitaciones de ingeniería de software o el tiempo dedicado a la depuración de códigos en

Python. El tiempo consumido es menor en comparación con Java. Como resultado, los desarrolladores pueden dedicar más tiempo a sus algoritmos y heurísticas relacionadas con IA y ML.

Python viene con una gran cantidad de bibliotecas incorporadas para el aprendizaje automático y la inteligencia artificial. Algunas de las bibliotecas más populares son Tensorflow, Pandas, Scikit learn, Numpy, Scipy, keras, etc

El soporte de la comunidad de desarrolladores y una gran cantidad de características es lo que hace que Python sea adecuado para aplicaciones de aprendizaje automático. Por otro lado, Java se creó principalmente para la programación general, no para el cálculo de números, un campo donde R y Python son más preferidos.

En el tema de velocidad y rendimiento java es el ganador, se ha venido demostrando muchos años que java es mucho más rápido que Python, pero donde Python carece de velocidad lo compensa con su flexibilidad.

Se eligió el lenguaje de programación Python dado que es un lenguaje de programación que cuenta con una buena seguridad, además de poder correr en múltiples plataformas.

Tiene una ejecución menos pesada.

Otra característica importante es que tiene bastantes librerías para poder usar más adelante con la implementación del protocolo y el análisis de los datos.

Otro aspecto importante que se tomó es que la librería con la que cuenta Python para la extracción de tráfico SNMP llamada PySNMP es fácil de implementar y aún cuenta con soporte.

Además, la carga de red que pueda originar es la más adecuada ya que no genera paquetes muy pesados.

Python es una buena alternativa en el caso que se quiera analizar datos y aplicar algoritmos de machine learning ya que existen muchas librerías que cuentan con un soporte al día de hoy

b) Propuesta de protocolo para extraer datos

Para esta actividad se hizo una búsqueda con el fin de encontrar el mejor protocolo que nos permita extraer los datos que necesitamos.

Se tuvo pensado desde un principio usar protocolos de capa 7 (Capa aplicación) ya que la mayoría de los protocolos cuentan con aplicaciones que permiten interactuar al usuario con los equipos de una manera amigable, y sencilla.

La capa 7 (capa Aplicación) es la última capa del modelo OSI y la capa más cercana al usuario es por eso por lo que esta capa es la que tiene el mayor número de protocolos que existen debido a que los usuarios tienen distintas necesidades para interactuar con equipos y servicios que ofrece la red.

Algunos de los protocolos que existen en esta capa son http, https, snmp, smtp, ssh, telnet, pop3, imap, etc.

Según Seaccna. (2015). Este nivel es responsable por convertir las diferencias que existen entre los varios sistemas operativos y aplicativos para un padrón, es decir, esta capa recibe las informaciones que viene del usuario que llamamos SDU (Service Data Unit) y adiciona la información de control que llamamos de PCI (Protocol Control Information) para que tengamos como salida la conocida PDU (Protocol Data Unit).

Se hizo una comparativa entre los tres protocolos descritos anteriormente, encontrando ventajas y desventajas unos con los otros.

Se eligió el protocolo simple de administración de red ya que este protocolo es mucho más fácil de usar y se encuentra en la mayoría de los equipos de comunicación que hoy existen. Además de ser mucho menos pesado que los demás protocolos.

Este protocolo además cuenta con mayor seguridad en su última versión implementada.

Netflow se centra en analizar el tráfico que entra por una interfaz y sale por otra para indicar información como IP origen, IP destino, protocolo, calidad de datos, etc.

IP SLA cuenta con funciones mejoradas que permiten extraer muchísimos más datos que SNMP, lo cual puede servir para un análisis más detallado, pero con la información que nos ofrece SNMP es suficiente para poder realizar las operaciones que deseamos. Permite configurar dispositivos para ejecutar pruebas desde su ubicación en la red.

Necesitamos un protocolo que permita extraer información no solo de las interfaces si no de componentes internos del equipo, además que no sobrecargue la red y que sea fácil de configurar, además añade la seguridad, ya que es un factor importante dado que no queremos intrusiones mientras estemos extrayendo datos sumamente importantes para el análisis de los mismos.

Tabla 13

Características de los protocolos evaluados

Protocolo	**Compatibilidad**	**Sobrecarga de red**	**Seguridad**	**Configuración**
SNMPv2	Todos los equipos de red	Menor	Buena	Fácil
NETFLOW	Mayoría de equipos Cisco	Mayor	Buena	Media
IP SLA	Mayoría de equipos Cisco	Mayor	Buena	Media/Alta

Fuente: Elaboración propia

c) Propuesta del IDE de desarrollo

Antes de poder usar cualquier IDE debemos tomar en cuenta el lenguaje de programación elegido anteriormente, hoy en día existen IDE desarrollados para el uso de un lenguaje de programación en especifico

Para esta actividad se permite el uso de cualquier IDE que nos facilite la programación de los algoritmos que se piensan desarrollar más adelante.

Se eligió pycharm para el desarrollo del script y Spyder para realizar la fase de análisis de datos.

Ambos ofrecen muchas herramientas que permitirán desarrollar de una manera más fácil y rápida los algoritmos necesarios

d) Búsqueda y elección de librerías

En esta actividad se procede a elegir una librería que sea fácil de implementar usar y a su vez muy útil previamente hecha una búsqueda exhaustiva.

Para el uso del protocolo elegido anteriormente se optó por una librería que nos permita extraer los datos. Se eligió la librería PySNMP, ya que esta es la más conocida para poder implementar operaciones con el protocolo simple de administración de red además de estar programada en python.

Para el manejo de hojas de cálculo se encontraron 2 las cuales son xlsxwriter y openpyxl.

Ambas librerías permiten el manejo de hojas de cálculo, pero cada una trae funcionalidades en específico.

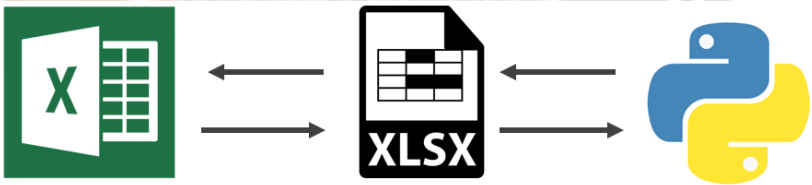

Figura 32. Funcionamiento de las librerías evaluadas para el manejo de archivos

Fuente: Elaboración propia

Tabla 14

Características de las librerías elegidas

Operación	**Xlsxwriter**	**Openpyxl**
Creación	Si	Si
Lectura y Escritura	Si	Si
Escritura de archivos creados	No	Si
Manejo de grandes cantidades de datos	Si	No

Fuente: Elaboración propia

Se optó por elegir Openpyxl para la lectura y escritura de hojas de cálculo, esta librería es una de las pocas que nos permite escribir datos en el archivo nuevamente una vez cerrado, en otras palabras, nos permite editar hojas de cálculo. Este es un punto muy importante ya que el script que se piensa desarrollar deberá escribir y cerrar el documento un determinado número de veces.

e) Identificar variables

Dado que se eligió el protocolo snmp para poder realizar la comunicación con los equipos y extraer la información de los mismos se tomarán las variables proporcionadas por las bases de información gestionada conocidas como OID's para poder identificar los parámetros que nos interesan.

Las variables principales que se necesitan para un análisis inicial son:

- Nombre de dispositivo
- Nombre del puerto
- Tipo de puerto
- Velocidad de transmisión
- Dirección física
- Estado del puerto
- Numero de paquetes enviados
- Número de paquetes recibidos
- Tiempo transcurrido

Las variables obtenidas de la base de información gestionada (MIB) denominada IF-MIB que cumplen con las necesidades anteriores son las mencionadas a continuación, además se encontraron otras más que nos proporcionaron información más detallada de algunos otros aspectos importantes que se necesitan para el análisis:

Tabla 15

Variables elegidas para la extracción de datos

Variable	Object ID (OID)
ifDescr	.1.3.6.1.2.1.2.2.1.2
ifType	.1.3.6.1.2.1.2.2.1.3
ifMtu	.1.3.6.1.2.1.2.2.1.4
ifSpeed	.1.3.6.1.2.1.2.2.1.5
ifPhysAddress	.1.3.6.1.2.1.2.2.1.6
ifOperStatus	.1.3.6.1.2.1.2.2.1.8
ifInOctets	.1.3.6.1.2.1.2.2.1.10
ifInUcastPkts	.1.3.6.1.2.1.2.2.1.11
ifInUnknownProtos	.1.3.6.1.2.1.2.2.1.15
ifOutOctets	.1.3.6.1.2.1.2.2.1.16
ifOutUcastPkts	.1.3.6.1.2.1.2.2.1.17
ifInMulticastPkts	.1.3.6.1.2.1.31.1.1.1.2
ifInBroadcastPkts	.1.3.6.1.2.1.31.1.1.1.3
ifOutMulticastPkts	.1.3.6.1.2.1.31.1.1.1.4
ifOutBroadcastPkts	.1.3.6.1.2.1.31.1.1.1.5
tiempo/msec	-

Fuente: Elaboración propia

f) Describir datos a extraer

En esta actividad se procederá a describir todas las variables que fueron elegidas para su posterior análisis.

A continuación, la descripción de las variables:

- **ifDescr**

  Esta variable proporciona el nombre de todas las interfaces que existen en el equipo, tanto físicas como lógicas ordenadas ascendentemente.

- **ifType**

  Esta variable indica el tipo de interfaz ya sea física o lógica

- **ifMtu**

  Esta variable indica la unidad máxima de trama que se puede mandar por cada interfaz.

- **ifSpeed**

  Devuelve la velocidad de transmisión de datos por interfaces, ya sea lógica o física

- **ifPhysAddress**

  Esta variable nos indica la dirección física de cada interfaz tanto lógica como física.

- **ifOperStatus**

  Indica el estado de cada puerto del equipo, ya sea físico o lógico.

  Los estados de cada interfaz pueden ser:

  - up
  - down
  - testing
  - unknown
  - dormant
  - notPresent
  - lowerLayerDown

- **ifInOctets**

  Esta variable indica el número total de paquetes que ingresó por el interfaz representado en octetos.

- **ifInUcastPkts**

  El número total de paquetes que se entregan a una capa superior.

- **ifInUnknownProtos**

  Esta variable da a conocer la cantidad de paquetes recibidos por la interfaz ya sea lógica o física que fueron descartados debido a un protocolo desconocido o incompatible.

- **ifOutOctets**

  Esta variable indica el número total de paquetes que salieron por el interfaz representado en octetos.

- **ifOutUcastPkts**

  El número total de paquetes que se entregan a una capa inferior.

- **ifInMulticastPkts**

  El número total de paquetes que se entregan a una capa superior y que serán dirigidos a una multidifusión.

- **ifInBroadcastPkts**

  El número total de paquetes de broadcast que fueron transmitidos por la interfaz.

- **ifOutMulticastPkts**

  El número total de paquetes que se reciben de una capa superior y que serán dirigidos a una multidifusión.

- **ifOutBroadcastPkts**

  El número total de paquetes de broadcast que fueron recibidos por la interfaz.

- **tiempo/msec**

  Tiempo transcurrido en milisegundos proporcionado por el reloj del administrador SNMP que extraerá los datos.

g) Desarrollo de script

En esta actividad se desarrolló un script con la finalidad de extraer los datos para ellos se usaron las siguientes librerías:

- PySNMP
- OpenPyXL
- Pathlib
- Datetime
- Time
- Threading

Además, se hizo uso del siguiente IDE:

- Pycharm

Se empezó con la creación del proyecto llamado "ScriptExtracciónDeDatos".

A continuación, se procedió con la creación de tres archivos con extensión Python, los cuales fueron:

- Script.py
- Interval.py
- App.py

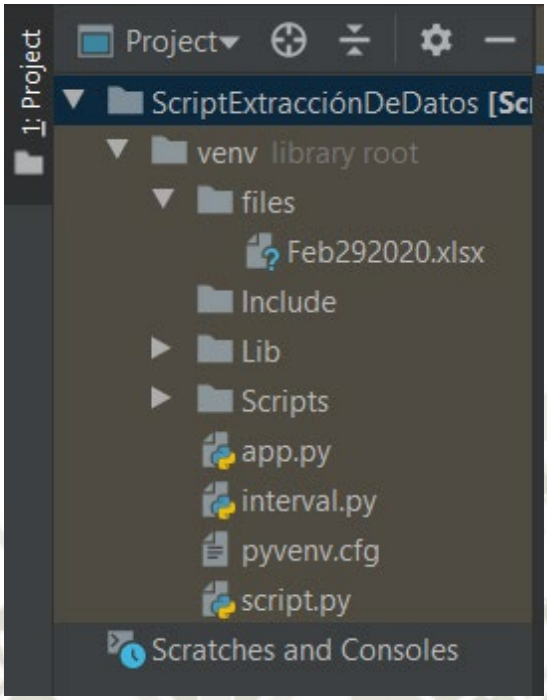

Figura 33. Estructura del código fuente del script programado en Python

Fuente: Elaboración propia

i. Archivo Interval.py

interval.py

```
from threading import Timer

class RepeatedTimer(object):
    def __init__(self, interval, function, *args, **kwargs):
        self._timer     = None
        self.interval   = interval
        self.function   = function
        self.args       = args
        self.kwargs     = kwargs
        self.is_running = False
        self.start()

    def _run(self):
        self.is_running = False
        self.start()
        self.function(*self.args, **self.kwargs)

    def start(self):
        if not self.is_running:
            self._timer = Timer(self.interval, self._run)
            self._timer.start()
            self.is_running = True

    def stop(self):
        self._timer.cancel()
        self.is_running = False
```

Figura 34. Código fuente del archivo internal.py

Fuente: Elaboración propia

ii. Archivo app.py

```python
from pathlib import Path
from datetime import date
from interval import RepeatedTimer
from script import archivoExcel, Dispositivo
from time import sleep

path = (str(Path().absolute()) + "\\files\\")

today = date.today()

arcExcel = archivoExcel(today.strftime("%b%d%Y"), "xlsx", path)
arcExcel.crearArchivoExcel()


def getData(count):
    (Dispositivo(161, 0, 'ciscolab', arcExcel, count)).consultarDispositivo()


print("Archivos guardados en:" + path + "\r\nConsultando...")
# Aqui pones el intervalo de tiempo en el que deseas que se ejecute
rt = RepeatedTimer(0.25, getData, 1)
try:
    sleep(120)
    arcExcel.closeArchivo()
finally:
    rt.stop()
```

Figura 35. Código fuente del archivo app.py

Fuente: Elaboración propia

iii. Archivo script.py

```python
class Dispositivo():
    def __init__(self, puerto, version, comunidad, archivo, ip=Null):
        self.comunidad = comunidad
        self.puerto = puerto
        self.version = version
        self.archivo = archivo
        self.ip = ip
        self.data = []
    def consultarDispositivo(self):
        print("Leyendo ...")
        for (errorIndication, errorStatus, errorIndex, varBinds) in nextCmd(SnmpEngine(),
                                                                            CommunityData(self.comunidad, mpModel=self.version),
                                                                            UdpTransportTarget( ('192.168.0.10', self.puerto)),
                                                                            ContextData(),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifDescr')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifType')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifMtu')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifSpeed')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifPhysAddress')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifOperStatus')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifInOctets')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifInUcastPkts')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifInUnknownProtos')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifOutOctets')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifOutUcastPkts')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifInMulticastPkts')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifInBroadcastPkts')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifOutMulticastPkts')),
                                                                            ObjectType(ObjectIdentity('IF-MIB', 'ifOutBroadcastPkts')),
                                                                            lexicographicMode=False):
            if errorIndication:
                print(errorIndication)
                break
            elif errorStatus:
                print('%s at %s' % (errorStatus.prettyPrint(),
                                    errorIndex and varBinds[int(errorIndex) - 1][0] or '?'))
                break
            else:
                if (errorIndication): break
                row_data = []
                for oid, val in varBinds:
                    row_data.append(val.prettyPrint())
                self.data.append(row_data)
        self.guardarnformacion()

    def guardarnformacion(self):
        self.archivo.escribirArchivo(self.data)
```

Figura 36. Código fuente del archivo script.py

Fuente: Elaboración propia

3.3.3. Emulación del ambiente de prueba

a) Búsqueda de emulador gráfico

En esta etapa se procede con la búsqueda de un emulador grafico para poder realizar el levantamiento de la topología con la configuración debida para de esta forma hacer las pruebas necesarias en el ambiente diseñado antes de proceder a hacerlo en un ambiente real.

Hoy en día existen varias herramientas que nos pueden ayudar a diseñar ambientes de pruebas, se procede a elegir un emulador y no un simulador ya que este último puede ofrecer equipos y características limitadas que puedan impedir las operaciones que se quieren realizar. Un emulador grafico nos proporciona casi el total de características necesarias para las operaciones que se desean realizar (Tejada, 2019).

GNS3 es una herramienta que permite el diseño de topologías de red complejas, además permite una combinación de equipos reales y virtuales de tal forma que pueden interactuar todos entre sí.

Esta herramienta es multiplataforma lo que permite su instalación y uso en diferentes sistemas operativos, GNS3 trae consigo una nueva máquina virtual (GNS3 VM) con el sistema operativo base Ubuntu 18.04 LTS, y ahora tendremos compatibilidad con **VMware, Virtual Box**, y una de las novedades es que tendremos también **compatibilidad con Hyper-V** de Microsoft, ideal para poder utilizar conjuntamente GNS3 VM y Docker, todo ello en el mismo ordenador sin tener problemas de virtualización entre ellos.

Estas nuevas características que trae consigo nos permite simular equipos de capa 2 con todas sus funcionalidades cosa que otros emuladores no pueden realizar.

Trae consigo incorporada la herramienta de análisis de protocolos llamada wireshark para poder realizar un análisis de todos los paquetes que se envían a través de la red tanto en un ambiente emulado como en un ambiente real.

Todas estas características y funcionalidades que trae consigo la herramienta GNS3 permite que sea el emulador indicado para poder levantar un ambiente de prueba con una infraestructura casi 100% similar a la que se tiene en un

ambiente real, lo cual nos permitirá extraer datos parecidos en su totalidad a la red real y poder analizarlos luego.

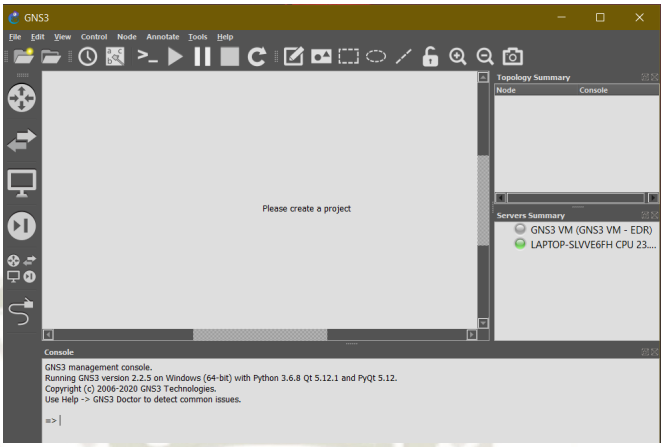

Figura 37. Interfaz inicial de GNS3

Fuente: Elaboración propia

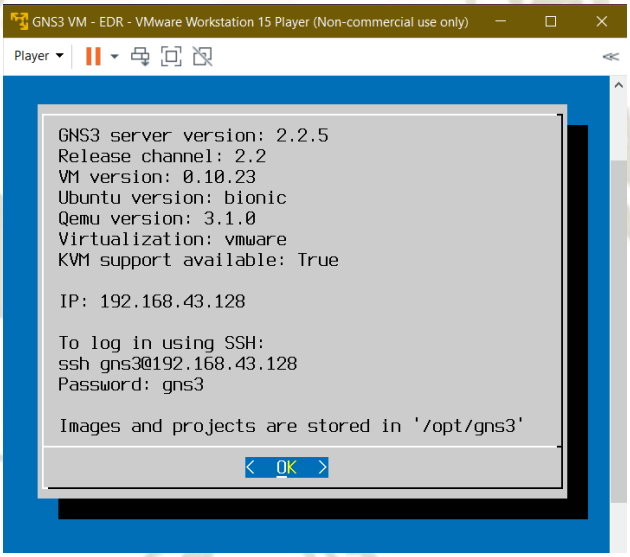

Figura 38. Características de la máquina virtual de GNS3

Fuente: Elaboración propia

b) Búsqueda de IOS

IOS (Internetwork Operating System) es el nombre como se le conoce al software usado en la mayoría de routers y switches de Cisco Systems. Es un paquete de funciones de enrutamiento, conmutamiento, trabajo de internet y telecomunicaciones que se integra estrechamente con un sistema operativo multitarea. Trae consigo una CLI (Interfaz de línea de comandos) que proporciona un conjunto de diferentes tipos de comandos para poder

interactuar con el dispositivo donde esté instalado. Se puede acceder a los diferentes tipos de comandos contando con un "Modo" y un "Nivel de privilegios del usuario".

El modo "Global configuration" proporciona un conjunto de comandos para cambiar la configuración del sistema.

El modo "Interface configuration" proporciona comandos para cambiar la configuración de una interfaz o un conjunto de las mismas.

A todos los comandos se les asigna un nivel de privilegios, de 0 a 15, y pueden ser accedidos por usuarios con los privilegios necesarios.

Hace un par de años no era posible emular switches de Cisco Systems en una topología de GNS3 porque su motor de virtualización Dynamips estaba limitado al uso de imágenes de routers de Cisco Systems, lo cual era un gran problema, pero había un par de maneras de solucionarlo:

La primera manera de solucionarlo era crear una topología e incluir un router de Cisco Systems virtualizado que venía por defecto en GNS3 e incluirle una tarjeta con 8 o 16 puertos ethernet, pero no se acercaba a las funciones de un switch real.

La otra manera era montar un laboratorio real con equipos cisco, pero era muy tedioso y costoso.

Por suerte los problemas mencionados forman parte del pasado ya que hoy en día hay también dos maneras de virtualizar switches de Cisco Systems en una topología de GNS3.

La primera opción se llama Cisco IOU (IOS Over Unix), este es un simulador de uso interno de Cisco Systems que conciste en una imagen que funciona sobre un sistema Unix de Solaris, existe una imagen de capa dos que puede simular de manera correcta y completa el funcionamiento de un switch de Cisco Systems.

La segunda opción se llama Cisco VIRL y es una opción muy potente ya que cuenta con todo el respaldo de cisco y su comunidad. VIRL pone una gran cantidad de dispositivos Cisco a nuestra disposición y se puede integrar con equipos virtuales de otros fabricantes. La única desventaja es que se tiene

que pagar 200 dólares anuales para poder descargar las imágenes de los equipos, al final no resulta mucho dado que nos ofrece muchas ventajas más que las dichas anteriormente. Pero como este trabajo de investigación trata de consumir recursos económicos, usaremos la primera opción de Cisco IOU, además nos ofrece todo lo necesario que se necesita para desarrollar el proyecto de investigación.

Tabla 16

Características de los emuladores evaluados

Opción	Complejidad	Multiplataforma	Costo
Cisco IOU	Alta	Si	Libre
Cisco VIRL	Alta	Si	199 USD

Fuente: Elaboración propia

i. IOU en GNS3 VM

GNS3 VM (GNS3 Virtual Machine) es una máquina virtual con un sistema Linux que se ejecuta en el interior de esta, será capaz de albergar todos nuestros dispositivos de red virtuales de forma eficiente y transparente para el usuario.

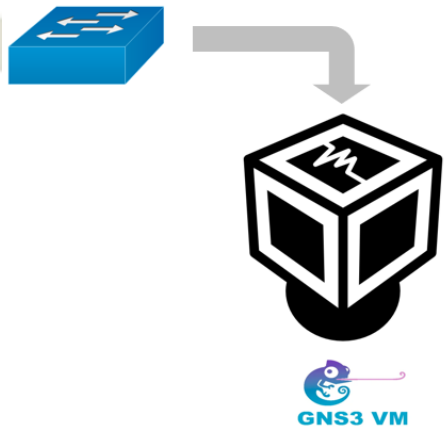

Figura 39. Arquitectura de la emulación de un switch en la máquina virtual de GNS3

Fuente: Elaboración propia

ii. I86bi-linux-l2-adventerprisek9-15.1a

Es la imagen que se recomienda para poder emular dispositivos de capa dos, está basada en Linux y está diseñada por Cisco Systems, proporciona casi todas las funcionalidades de la serie de switches 2900. Requiere de una licencia que es fácil de encontrarla en repositorios de IOS de Cisco Systems.

c) Diseño de topología

En esta etapa se debe plasmar el diseño real y la respectiva configuración con el fin de poder realizar las pruebas de manera correcta.

Adicionalmente se propone agregar un canal dedicado por el cual se pueda extraer la información sin ningún problema tanto en el ambiente real como el virtualizado.

El fin de agregar el canal dedicado es el de poder extraer la información en tiempo real a pesar de cualquier circunstancia que se esté presentando dentro de la red, este canal si bien va conectado a todos los equipos a analizar cumple su labor como si estuviese en una red aparte, con la finalidad de no involucrarlos en los problemas que ocurran más adelante en dicha red.

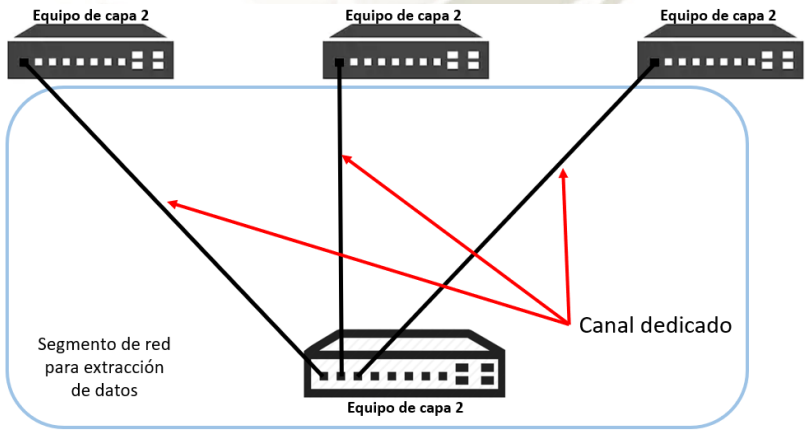

Figura 40. Propuesta de implementación de canales para la extracción de datos

Fuente: Elaboración propia

Los canales dedicados pueden ser alámbricos o inalámbricos siempre y cuando a través de ellos solo circulen los datos necesarios para su posterior almacenamiento.

Entre los datos que circularán por los canales se encuentran paquetes de los protocolos ICMP y SNMP.

d) Configuración de equipos

Esta etapa se debe desarrollar una vez que se haya implementado la topología de red, se deberán añadir todas las configuraciones tal y como se tienen en la topología real.

Adicionalmente se deben añadir las siguientes configuraciones para crear el canal dedicado por donde se pasará los datos que necesitamos.

i. Switch Principal

Es el equipo de donde saldrán los enlaces para formar el canal dedicado hacia los equipos que se van a analizar.

La configuración que necesita el equipo debe ser la siguiente:

#Switchport mode Access

#Switchport Access vlan 10

Todos los puertos que tengan una conexión directa con algún otro equipo de red deberán está configurado en modo “Access” y debe estar incluido en la Vlan 10.

De esta manera nos aseguramos de que solo el tráfico que se genere este dentro de la Vlan 10 previamente creada.

ii. Switches Directamente Conectados

Son todos los equipos que van a ser analizados, de los cuales se va a extraer información para almacenarla y luego analizarla.

La configuración que necesitan los equipos debe ser la siguiente:

#switchport mode Access

#switchport access vlan 10

#ip address <ip dentro del rango> <mascara de red>

#snmp-server community R1 ro

#snmp-server community R1rw rw

Comandos opcionales para aumentar seguridad

#snmp-server community ciscolab ro SNMP_ACL

#snmp-server location <DataCenter >

#snmp-server contact <Adminitrador >

#snmp-server host <Servidor SNMP > version 2c <Comunidad>

#snmp-server enable traps

#ip access-list standard SNMP_ACL

#permit <Servidor SNMP >

El puerto al cual esté conectado un enlace proveniente del Switch principal deberá estar configurado en modo "Access" y debe estar incluido en la Vlan 10.

Además todos los equipos deberán contar con las vlan10 creada y además la Vlan debe tener una dirección ip y una máscara subred dentro del mismo rango.

Además, se deberá configurar el protocolo SNMP para poder extraer los datos mediante el mismo. Se debe establecer una comunidad para lectura y una para escritura, para que de esta forma la comunidad de lectura nos permita extraer información mientras que la de lectura permita cambiar parámetros de configuración (Tejada, 2019).

3.3.4. Preparación de los datos

a) Extracción de datos

En esta etapa se debe ejecutar el script desarrollado para la fase anterior, además se deben ir ajustando los parámetros necesarios hasta lograr extraer los que se necesita.

El Script cuenta con una serie de tareas entre las cuales se encuentran:

- Prueba de comunicación con el equipo de red
- Asignación de tiempo de consultas
- Variables por extraer, etc.

Todos los parámetros influyen en la etapa de extracción de datos por lo que es necesario un correcto ajuste de los mismos.

b) Almacenamiento de datos

Los datos deben ir almacenados en un lugar seguro y con amplia capacidad para almacenar archivos pesados y es que esa etapa es muy crítica ya que se deben conservar los datos de manera correcta y en el formato correcto.

Si bien el script propuesto en la fase anterior guarda los datos extraídos en hojas de cálculo es sumamente importante que estas se vayan guardando en un sector del disco fácil de acceder y con el espacio suficiente para que el archivo crezca.

c) Preprocesamiento de datos

Para esta actividad se debe tener un conocimiento de las características de datos que tenemos almacenados, para nuestro conjunto de datos se procedieron a aplicar diferentes técnicas de pre-procesado con el fin de poder obtener datos limpios, ordenados y claros con los cuales podamos trabajar.

Se deben aplicar las técnicas adecuadas al conjunto de datos que se obtengan.

Para este conjunto de datos se aplicaron las siguientes técnicas:

i. Eliminación de filas y columnas

Se usó esta técnica ya que se cuentan con filas que contienen datos que según el criterio personal no servirán de mucho.

Se procede a quitar estas filas con la finalidad de tener un resultado más preciso con las técnicas de aprendizaje automático que se aplicarán más adelante.

Las siguientes variables son las que fueron eliminadas por criterio propio:

Tabla 17

Variables descartadas del análisis

Variable	Tipo
ifType	Cadena
ifMtu	Entero
ifSpeed	Entero
ifPhysAddress	String
ifOperStatus	String
ifInUnknownProtos	Entero

Fuente: Elaboración propia

Además, se procedió a eliminar todas las filas que contenían en la columna ifDescr el valor vlan10 ya que ese registro pertenece a la vlan que se configuró para extraer los datos. Por lo tanto, aportará un dato que no estamos interesados en analizar.

```
import pandas as pd
dataset = pd.read_excel("datos.xlsx")

#Eliminación de columnas
dataset = dataset.drop(['ifType', 'ifMtu'], axis=1)
dataset =dataset.drop(['ifSpeed'], axis=1)
dataset =dataset.drop(['ifPhysAddress'], axis=1)
dataset =dataset.drop(['ifOperStatus'], axis=1)
dataset =dataset.drop(['ifInUnknownProtos'], axis=1)

#Eliminación de la fila con valor de 'vlan10'
dataset = dataset.drop(dataset[dataset['ifDescr']=='Vlan10'].index)
```

Figura 41. Eliminación de las variables mediante la función drop

Fuente: Elaboración propia

ii. Relleno de datos faltantes

Dado que cada registro de información que se tiene es crucial para poder analizar los datos se optó por rellenar datos faltantes para evitar eliminar todo un registro.

Los datos faltantes pueden representarse en símbolos como los siguientes:

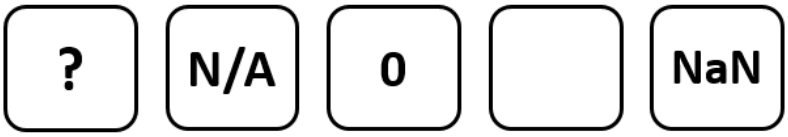

Figura 42. Símbolos que representan datos faltantes

Fuente: Elaboración propia

Se reemplazaron los valores faltantes por el valor promedio de toda la columna. De esta forma se evita eliminar registro (Tejada, 2019).

```
prom_IfInOctets = dataset["ifInOctets"].mean()
dataset["ifInOctets"].replace(np.nan, prom_ifInOctets)

prom_ifInBroadcastPkts = dataset["ifInBroadcastPkts"].mean()
dataset["ifInBroadcastPkts"].replace(np.nan, prom_ifInBroadcastPkts)

prom_ifOutBroadcastPkts = dataset["ifOutBroadcastPkts"].mean()
dataset["ifOutBroadcastPkts"].replace(np.nan, prom_ifOutBroadcastPkts)

```

Figura 43. Código fuente del relleno de datos faltantes

Fuente: Elaboración propia

3.3.5. Modelamiento

a) Elección de las técnicas de aprendizaje automático

El objetivo principal del aprendizaje automático es crear un modelo que permita resolver una tarea encomendada. Este modelo será entrenado con grandes cantidades de datos, lo que permitirá que el modelo aprenda y pueda realizar predicciones

Esta parte es importante ya que se debe seleccionar el algoritmo adecuado según la tarea que se quiere realizar.

Elegir la técnica adecuada no es una tarea fácil, pero con algunas pautas se puede lograr aplicar la más indicada para nuestro problema.

En primero lugar debemos definir cuál es el objetivo de esta fase, pues el objetivo es lograr predecir cuándo va a fallar un equipo de red, en este caso se estaría hablando de un problema de regresión.

Luego debemos tener claro con que información contamos para poder conseguir el objetivo planteado anteriormente, en este caso tenemos los datos de equipos que estuvieron funcionando con normalidad y de un determinado momento a otro fallaron, claramente con los datos podremos deducir que pudo influir en los equipos para que fallen, pues es aquí donde podemos

inferir que para poder elegir las técnicas nos basaremos en técnicas de aprendizaje automático supervisadas.

Dado que hoy en día las redes neuronales ayudan a predecir con mayor exactitud datos, se ha optado por aplicar esta técnica

Entonces se procede con la elección de las técnicas en base a lo detallado anteriormente, las técnicas son las siguientes:

- Regresión polinómica
- Support Vector Machine (SVM)
- Gradient Boosting (XGBoost)
- Redes Neuronales Artificiales.

b) Aplicación de las técnicas elegidas

Se utilizó el lenguaje de programación Python y el IDE Spyder, para el procesamiento de los datos se ha utilizado un computador con procesador Core I7 de 8va generación.

i. Regresión Polinómica

Los modelos de regresión tienen una serie de restricciones que uno debe de comprobar antes de calcular el modelo de regresión.

Estas restricciones son las siguientes:

- Linealidad
- Homocedasticidad
- Normalidad multivariable
- Independencia de los errores
- Ausencia de multicolinealidad

Estas restricciones son muy importantes y si al menos alguna de estas no se cumple el modelo no tendrá sentido.

No porque un modelo tenga una gran cantidad de variables independientes quiere decir que sea mejor, es por eso que se procederá a

eliminar algunas variables para tener una predicción mucho más exacta, estas variables no se eliminaran por criterio propio ya que se estaría cometiendo un error grave, por lo tanto se aplicará una técnica llamada eliminación hacia atrás.

La eliminación hacia atrás es un proceso que permite seleccionar las variables adecuadas para el análisis, primero se parte de un conjunto de datos que contienen todas las variables seguidamente estas se van eliminando poco a poco mientras se va calculando un valor. Aquella variable que tenga la menor correlación parcial con la variable dependiente será la primera en ser considerada para su eliminación.

Agregamos una columna de valores "1" al principio de nuestro conjunto de datos para poder realizar la eliminación hacia atrás

Utilizamos la librería Statsmodels para poder aplicar el proceso

Damos un valor P inicial de 0.05, cualquier variable que sobrepase ese valor será eliminada.

```
# Eliminación hacia atrás
import statsmodels.api as sm
X = np.append(arr = np.ones((500,1)).astype(int), values = X, axis = 1)
SL = 0.05


X_opt = X[:, [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]]
regression_OLS = sm.OLS(endog = y, exog = X_opt.tolist()).fit()
regression_OLS.summary()

X_opt = X[:, [0, 1, 2, 3, 4, 6, 7, 8, 9]]
regression_OLS = sm.OLS(endog = y, exog = X_opt.tolist()).fit()
regression_OLS.summary()

X_opt = X[:, [0, 1, 2, 4, 6, 7, 8, 9]]
regression_OLS = sm.OLS(endog = y, exog = X_opt.tolist()).fit()
regression_OLS.summary()

X_opt = X[:, [0, 1, 4, 6, 7, 8, 9]]
regression_OLS = sm.OLS(endog = y, exog = X_opt.tolist()).fit()
regression_OLS.summary()
```

Figura 44. Proceso de eliminación mediante el valor P

Fuente: Elaboración propia

Repetimos el proceso hasta que haya se tenga un valor 0 en el valor "P" de todas las variables.

	coef	std err	t	P>|t|	[0.025	0.975]
const	-4.0356	0.667	-6.053	0.000	-5.346	-2.726
x1	-3.948e-10	1.18e-10	-3.341	0.001	-6.27e-10	-1.63e-10
x2	-8.444e-06	1.55e-06	-5.434	0.000	-1.15e-05	-5.39e-06
x3	3.602e-06	2.55e-07	14.142	0.000	3.1e-06	4.1e-06
x4	0.0136	0.002	7.300	0.000	0.010	0.017
x5	-3.517e-06	2.99e-07	-11.757	0.000	-4.1e-06	-2.93e-06
x6	0.0010	0.000	7.853	0.000	0.001	0.001

Omnibus:	33.845	Durbin-Watson:	0.189
Prob(Omnibus):	0.000	Jarque-Bera (JB):	130.452
Skew:	0.032	Prob(JB):	4.71e-29
Kurtosis:	5.502	Cond. No.	9.71e+09

Figura 45. Representación del valor P en las variables

Fuente: Elaboración propia

Con las variables que quedan ya podemos aplicar el modelo.

	0	1	2	3	4	5	6
0	1	5.92131e+06	18712	178	367	141	1.25
1	1	240559	447	156	361	119	1.5
2	1	276182	432	147	358	110	1.75
3	1	78787	344	166	355	129	2.75
4	1	5.93e+06	18739	178	367	141	8.5
5	1	240559	447	156	361	119	8.75
6	1	276182	432	147	358	110	9
7	1	78787	344	166	355	129	10
8	1	5.93488e+06	18755	178	367	141	15.75

Figura 46. Datos resultantes mediante eliminación de variables

Fuente: Elaboración propia

Importamos la librería PolynomialFeatures, se le da un valor de 2 al parámetro degree para que pueda realizar las combinaciones polinómicas 2 veces a cada variable

```
from sklearn.preprocessing import PolynomialFeatures
X_train_poly = PolynomialFeatures(degree=2).fit_transform(X_train)
X_test_poly = PolynomialFeatures(degree=2).fit_transform(X_test)
```

Figura 47. Creación de variables de entrenamiento con función PolynomialFeatures

Fuente: Elaboración propia

Importamos la librería LinearRegression para aplicar las respectivas operaciones, y procedemos a entrenar el modelo, luego usamos la función Score () para poder predecir el porcentaje de aprendizaje.

```
from sklearn.linear_model import LinearRegression
model_lr = LinearRegression().fit(X_train_poly,y_train)
predict = model_lr.predict(X_test_poly)
Porcentaje = model_lr.score(X_train_poly, y_train)
print('Porcentaje de predicción:', Porcentaje)
```

Figura 48. Implementación de la función LinearRegression

Fuente: Elaboración propia

El modelo logra predecir en un 95 por ciento si un equipo va a fallar.

```
In [30]: from sklearn.preprocessing import PolynomialFeatures

In [31]: X_train_poly = PolynomialFeatures(degree=2).fit_transform(X_train)
    ...: X_test_poly = PolynomialFeatures(degree=2).fit_transform(X_test)

In [32]: from sklearn.linear_model import LinearRegression

In [33]: model_lr = LinearRegression().fit(X_train_poly,y_train)
    ...: predict = model_lr.predict(X_test_poly)

In [34]: Porcentaje = model_lr.score(X_train_poly, y_train)

In [35]: print('Porcentaje de predicción:', Porcentaje)
Porcentaje de predicción: 0.9590152864660692
```

Figura 49. Resultado de la aplicación del modelo de regresión polinómica

Fuente: Elaboración propia

ii. Support Vector Machine (SVM)

La máquina de soporte vectorial ofrece una mayor precisión en comparación a otros clasificadores, es conocida por su truco de Kernel para manejar espacios de entrada no lineales.

Se procede con el escalado de variables haciendo uso de la plataforma StandardScaler, se escalan las variables independientes de entrenamiento y test.

```
# Escalado de variables
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

Figura 50. Escalamiento de variables

Fuente: Elaboración propia

Importamos la librería SVC, creamos una instancia y a al parámetro Kernel especifica el tipo de núcleo que se utilizará por lo tanto utilizaremos un Kernel lineal que permitirá separar con una línea recta el conjunto de datos.

El para metro random_state se inicializa en 0 se utiliza para inicializar el generador interno de números aleatorios, que decidirá la división de datos en índices de entrenamiento y prueba en su caso.

Finalmente se llama a la función fit (), la cual creará el modelo o sea la máquina de soporte vectorial.

```
# Ajustar el SVM en el Conjunto de Entrenamiento
from sklearn.svm import SVC
classifier = SVC(kernel = "linear", random_state = 0)
classifier.fit(X_train, y_train)
```

Figura 51. Implementación de la SVM

Fuente: Elaboración propia

Se usó la función score() que nos devuelve el porcentaje de predicción de la técnica supervisada.

```
Porcentaje = classifier.score(X_test, y_test)
print('Porcentaje de predicción:', Porcentaje)
```

Figura 52. Uso de la función score

Fuente: Elaboración propia

Finalmente se obtiene una predicción de un total de 97.8 por ciento, lo que indica que el modelo aprendió de manera correcta.

```
In [9]: Porcentaje = classifier.score(X_train, y_train)
   ...: print('Porcentaje de predicción:', Porcentaje)
Porcentaje de predicción: 0.9786666666666667
```

Figura 53. Resultado de la aplicación del modelo SVM

Fuente: Elaboración propia

iii. Gradient Boosting (XGBoost)

Esta técnica nos permite trabajar con grandes cantidades de datos a una velocidad considerable, XGBoost permite aumentar la gradiente distribuida con el fin de que sea altamente eficiente flexible y portátil.

Se procede con la importación de la librería xgboost y la clase XGBClassifier, se crea una instancia de esta, y se ajustan los datos con la función fit(), a la cual se le envían las variables de entrenamiento.

```
# Ajustar el modelo XGBoost al Conjunto de Entrenamiento
from xgboost import XGBClassifier
classifier = XGBClassifier()
classifier.fit(X_train, y_train)
```

Figura 54. Implementación de XGBoost

Fuente: Elaboración propia

La consola nos muestra todos los parámetros configurados con los valores por defecto.

Se indica que se crearon 100 árboles de clasificación a partir de nuestros datos para aumentar el grado de clasificación.

Además, se creó una red neuronal con 3 capas de profundidad y una ratio de aprendizaje de 0.1.

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0,
              learning_rate=0.1, max_delta_step=0, max_depth=3,
              min_child_weight=1, missing=None, n_estimators=100, n_jobs=1,
              nthread=None, objective='multi:softprob', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=1, verbosity=1)
```

Figura 55. Resultado de inicializar los parámetros de XGBClassifier

Fuente: Elaboración propia

Se procede a invocar a la función score() para que nos devuelva el porcentaje de aprendizaje que tuvo la técnica.

```
#Predicción de los resultados con el Conjunto de Testing
y_pred  = classifier.predict(X_test)

Porcentaje = classifier.score(X_test, y_test)
print('Porcentaje de predicción:', Porcentaje)
```

Figura 56. Aplicación de la predicción y la variable score

Fuente: Elaboración propia

La precisión que nos proporciona esta técnica es de un 89.3 por ciento.

```
In [18]: Porcentaje = classifier.score(X_test, y_test)
    ...: print('Porcentaje de predicción:', pr)
Porcentaje de predicción: 89.3
```

Figura 57. Resultado de la aplicación del modelo XGBoost

Fuente: Elaboración propia

iv. Redes Neuronales Artificiales

Comenzamos usando el escalado de nuestras variables independientes tanto las de entrenamiento como las de test.

Con la finalidad de hacer un cálculo mucho más fácil. El escalado consiste en transformar los valores a valores cercanos a cero.

```
# Escalado de variables
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

Figura 58. Escalamiento de variables

Fuente: Elaboración propia

Procedemos a importar la librería keras, dicha librería nos proporciona una capa sobre tensorflow para que sea mucho más fácil el poder configurar la red neuronal.

```
# Importar Keras y librerías adicionales
import keras
from keras.models import Sequential
from keras.layers import Dense
```

Figura 59. Importación de la librería keras

Fuente: Elaboración propia

Se procede a crear el objeto classifier que será la red neuronal mediante el uso de la clase Sequential ().

```
# Inicializar la RNA
classifier = Sequential()
```

Figura 60. Creación del objeto classifier

Fuente: Elaboración propia

Procedemos a señalar cuantas variables de entrada usaremos con la función input_dim(),luego se procede a usar el parámetro Kernel_initializer y se le asigna el valor uniform para distribuir de manera aleatoria los pesos al inicio, asignamos a la función de activación el valor de rectificador lineal unitario ,además creamos la primera capa ocultar que estará compuesta por 5 nodos.

Luego se procede a crear la segunda capa oculta también compuesta por 5 nodos.

Finalizamos con la capa de salida compuesta de 3 nodos, ya que tenemos 3 variables dependientes.

```
# Añadir las capas de entrada y primera capa oculta
classifier.add(Dense(units = 5, kernel_initializer = "uniform",
                     activation = "relu", input_dim = 9))

# Añadir la segunda capa oculta
classifier.add(Dense(units = 5, kernel_initializer = "uniform",
                     activation = "relu"))

# Añadir la capa de salida
classifier.add(Dense(units = 3, kernel_initializer = "uniform",
                     activation = "relu"))
```

Figura 61. Aplicación de las capas a la red neuronal

Fuente: Elaboración propia

Procedemos a compilar la red neuronal y luego se ajusta la red neuronal al conjunto de entrenamiento.

```
# Compilar la RNA
classifier.compile(optimizer = "adam", loss = "binary_crossentropy",
                   metrics = ["accuracy"])

# Ajustamos la RNA al Conjunto de Entrenamiento
classifier.fit(X_train, y_train,  batch_size = 10, epochs = 100)
```

Figura 62. Implementación del número de veces de recorrido de una RNA

Fuente: Elaboración propia

Se logra predecir con un 79.6 por ciento de probabilidad.

```
Epoch 98/100
8000/8000 [==============================] - 2s 226us/step - loss: 0.5060 - accuracy: 0.7960
Epoch 99/100
8000/8000 [==============================] - 2s 215us/step - loss: 0.5060 - accuracy: 0.7960
Epoch 100/100
8000/8000 [==============================] - 2s 216us/step - loss: 0.5060 - accuracy: 0.7960
Out[23]: <keras.callbacks.callbacks.History at 0x20b0f79bd88>
```

Figura 63. Resultado de la aplicación del modelo de red neuronal

Fuente: Elaboración propia

3.3.6. Evaluación

a) Evaluación de resultados

A continuación, se detallan los resultados de predicción obtenidos al aplicar las técnicas de aprendizaje automático aplicadas en el punto 3.4.5 al conjunto de datos.

Tabla 18

Resultados de las técnicas aplicadas

Técnica aplicada	**Resultado de la predicción**
Regresión Polinómica	95.9%
Support Vector Machine (SVM)	97.8%
Gradient Boosting (XGBoost)	89.3%
Redes Neuronales Artificiales	79.6%

Fuente: Elaboración propia

No sirve de mucho elegir una técnica con el porcentaje de predicción muy cercano al 100% ya que puede ocasionarnos problemas al predecir datos futuros, si el resultado de predicción es igual o muy cercano al 100% significa que la técnica de aprendizaje automático fue entrenada de más.

Lo que se debe de evitar es que la técnica se ajuste a unas características muy específicas de los datos de entrenamiento que no tienen relación causal con la función objetivo que estamos buscando para generalizar (Iaarbook.github, 2019).

Dado que la técnica denominada máquina de vector soporte o Support Vector Machine (SVM) nos ofrece un resultado de predicción optimo se procede con la elección de la misma.

Es posible que para trabajos de investigación futuros se piense utilizar otras técnicas de aprendizaje automático, ésta propuesta metodología no obliga a usar las técnicas aplicadas en el proceso anterior.

Se puede usar cualquier técnica que se pueda acomodar de la mejor forma al conjunto de datos.

i. Validación con datos reales de equipos de red

Este proceso se hizo con la finalidad de dar a conocer el resultado final de unos registros al aplicarle la técnica de aprendizaje automática elegida anteriormente.

Tabla 19

Indicadores de un correcto funcionamiento de una interfaz

Variable	Rango correcto en octetos	Promedio	Influencia en CPU
ifInOctets	5921312 - 8681531	7301422	Alta
ifInUcastPkts	18607 - 27052	22830	Baja
ifOutOctets	6371529 - 9411469	7891499	Alta
ifOutUcastPkts	18712 - 27617	23165	Baja
ifInMulticastPkts	2766 - 2700	2739	Media
ifInBroadcastPkts	178 - 1867719	575160	Alta
ifOutMulticastPkts	367 - 417	392	Media
ifOutBroadcastPkts	141 - 1658579	503297	Alta

Fuente: Elaboración propia

Tabla 20

Características del funcionamiento adecuado de una interfaz

Variable	Valor
ifInOctets	278994
ifInUcastPkts	442
ifOutOctets	89732
ifOutUcastPkts	441
ifInMulticastPkts	2460
ifInBroadcastPkts	147
ifOutMulticastPkts	367
ifOutBroadcastPkts	110

Fuente: Elaboración propia

Tabla 21

Características de un fallo próximo en una interfaz

Variable	**Valor**
ifInOctets	180862656
ifInUcastPkts	483
ifOutOctets	180673331
ifOutUcastPkts	481
ifInMulticastPkts	2537
ifInBroadcastPkts	692086
ifOutMulticastPkts	398
ifOutBroadcastPkts	662948

Fuente: Elaboración propia

Tabla 22

Características del fallo de una interfaz

Variable	**Valor**
ifInOctets	452307904
ifInUcastPkts	491
ifOutOctets	452118931
ifOutUcastPkts	490
ifInMulticastPkts	2545
ifInBroadcastPkts	1730154
ifOutMulticastPkts	406
ifOutBroadcastPkts	1521014

Fuente: Elaboración propia

3.3.7. Aplicación del proceso de recuperación

a) Verificación de parámetros a modificar

El protocolo simple de administración de red permite en gran parte el modificar parámetros de configuración de los equipos de red.

Los parámetros deberán contar con la operación "set" que permitirá su respectiva configuración

Lo primero que debemos hacer es identificar las MIB's con las respectivas variables que queremos modificar y luego verificar si estas son "setteables" ósea que permitan escritura.

Las MIB siempre van a tender a variar dependiendo del problema que se esté tratando de solucionar, como esta investigación se centra en la solución de bucles en capa enlace de datos se usaran las siguientes:

Tabla 23

Variables con operaciones de lectura y escritura

Mib	Oid	Nombre	Operación
IF-MIB	.1.3.6.1.2.1.2.2.1.7	ifAdminStatus	Lectura-Escritura
IF-MIB	1.3.6.1.2.1.31.1.1.1.16	ifPromiscuousMode	Lectura-Escritura
IF-MIB	1.3.6.1.2.1.3.1.1.2	atPhysAddress	Lectura-Escritura
IF-MIB	1.3.6.1.2.1.6.13.1.1	tcpConnState	Lectura-Escritura
MAU-MIB	1.3.6.1.2.1.26.2.1.1.11	ifMauDefaultType	Lectura-Escritura

Fuente: Elaboración propia

b) Aplicación del proceso de recuperación

Para poder aplicar los procesos de recuperación es necesario establecer políticas de configuración, en estas políticas deberán detallarse hasta que nivel se puede configurar ciertas interfaces de red por parte de un sistema y un encargado de red.

En primera instancia se deberá establecer una serie de estados para identificar las prioridades.

Se proponen 4 estados los cuales son:

Tabla 24

Prioridades aplicables al proceso de recuperación

N°	Prioridad
1	No modificable
2	Modificable con permiso del encargado
3	Modificable
4	Modificada

Fuente: Elaboración propia

1. **No modificable**

   Este estado hace hincapié en las interfaces de red que tengan equipos críticos directamente conectados como servidores, firewalls, bases de datos, etc.

   Por lo tanto, la configuración de la interfaz no deberá ser modificada por algo o alguien diferente al encargado.

2. **Modificable con permiso del encargado**

   Este estado permitirá hacer modificadores a los equipos y/o interfaces, pero antes de hacerlo el encargado deberá otorgar el consentimiento, este estado será aplicado también a equipos críticos de segundo nivel, hago referencia a segundo nivel ya que pueden ser importantes, pero no tanto como los principales donde puede ocasionarse una pérdida de datos en el caso de una mala configuración en los puertos de red

3. **Modificable**

   Este estado permitirá aplicar cualquier configuración que sea la indicada, será para puertos que tengan conexión directa a equipos que no sean tan importantes.

4. **Modificado**

   Este estado se asigna a las interfaces que ya sufrieron alguna modificación en su configuración.

Esta parte de la metodología propone tres maneras de aplicar un proceso de recuperación, con el fin de poder evadir un posible problema una vez detectado o con el fin de solucionar un problema una vez que este haya sucedido.

i. **Proceso de recuperación con intervención de un software.**

Este proceso deberá darse cuando se cumplan las prioridades 2 y 3, ya que el software podrá aplicar las configuraciones necesarias a las interfaces de red gracias a las variables setteables previamente indagadas. Entre estas configuraciones están, bloqueo o activación de puertos, configuración de velocidad del puerto, eliminación de procesos, etc.

ii. **Proceso de recuperación con intervención de un especialista.**

Este proceso deberá darse cuando se cumplan las prioridades 1 y 2, esta investigación tiene la finalidad de encontrar patrones de falla y poder ayudar a aplicar una recuperación de los equipos de una manera proactiva por lo tanto si se logró identificar el problema que desencadenará muchos más y la interfaz de red tiene conectado un equipo de suma importancia se recomienda que solo la persona indicada proceda a solucionar el problema, dado que el análisis que se hará constantemente mediante el algoritmo de aprendizaje automático contiene registros de las interfaces a fallar será fácil identificar donde se está originando el problema. De esta manera se le permitirá al especialista solucionar el problema antes de que ocurre o identificar donde se originó el problema una vez que haya ocurrido.

iii. **Proceso de recuperación con intervención de ambas partes.**

Este proceso combina los dos anteriores ya que si bien el software puede modificar parámetros primero lo debe informar al encargado para que este le pueda aprobar la reconfiguración en caso sea necesario, generalmente se da en interfaces que tengan prioridad 1 y 2.

3.3.8. Despliegue en producción

Esta fase final permite aplicar en los equipos de infraestructura todo lo que fue previamente emulado en un ambiente de pruebas, solo se podrá llegar a esta fase cuando se tenga la certeza de haber cumplido los objetivos que fueron planteados en un principio.

Es una fase muy importante dado que se procederá a aplicar las configuraciones previamente hechas en el emulador y solo serán aplicables si funcionó todo en su totalidad

a) Identificación de equipos

En primer lugar, se deberán identificar los equipos de infraestructura de red a los que se les aplicará la configuración propuesta en el ambiente de pruebas. Los equipos pueden ser Switch's, Router's incluso Switch's de capa de red (Capa 3).

Para poder realizar una correcta y adecuada identificación de equipos se debe desarrollar un inventariado detallados de los mismos.

Además, se debe tener el conocimiento de la configuración de los equipos antes de aplicarle la configuración que fue emulada.

Los siguientes formatos de inventario son los que se deben aplicar para poder llevar el control de los equipos con sus características y configuración.

La tabla de inventario contendrá toda la información referente a los equipos con los que se van a trabajar los cuales serán en su mayoría equipos de capa enlace de datos.

Los datos que se necesitan son los siguientes:

Código: Compuesto por una nomenclatura creada por la persona encargada de los equipos.

Equipo: Pude ser Switch de capa 2 o Switch de capa 3, incluso algún equipo de capa 3 que se esté analizando en paralelo.

Marca/Modelo: La marca y el modelo de los equipos.

Capa: La capa en la que se encuentran funcionando los equipos.

Área: El área al que proporciona servicio el equipo.

Estado: Puede ser activo o inactivo.

Prioridad: La prioridad que tiene el equipo, puede ser alta si alguna falla puede afectar otros equipos de red o baja si una falla en este no afecta otros equipos.

Tabla 25

Estructura de la tabla de inventario

Fecha:							
Descripción:							
Nombre:							
N°	**Código**	**Equipo**	**Marca/Modelo**	**Capa**	**Área**	**Estado**	**Prioridad**
1							
2							
3							
.							
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
n							

Fuente: Elaboración propia

La tabla de características contendrá las especificaciones de los equipos de red previamente listados en la anterior tabla.

Los datos que se necesitan son los siguientes:

Código: Deberá ser el mismo código de la tabla anterior

Equipo: Pude ser Switch de capa 2 o Switch de capa 3, incluso algún equipo de capa 3 que se esté analizando en paralelo.

Serie: Esta es la misma que tiene el equipo al momento de realizar la compra de este, también se puede verificar por consola.

CPU: CPU con el que cuenta el equipo.

Dram: Descripción de las características de la memoria Dram del equipo

Memoria Flash: Descripción de las características de la memoria Flash del equipo

Puerto: Número de puertos que tiene conexión directa a otros equipos de red de la misma capa

Tipo: Depende de la velocidad ya que puede ser 10/100/1000 mbps por lo tanto puede ser ethernet, fastethernet o gigabitethernet, etc.

Tabla 26

Estructura de la tabla de características

Fecha:								
Descripción:								
Nombre:								
N°	**Código**	**Equipo**	**Serie**	**CPU**	**Dram**	**Memoria Flash**	**Puerto**	**Tipo**
1								
2								
3								
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
n								

Fuente: Elaboración propia

La última tabla mostrada a continuación nos indica de manera estructurada la configuración que se tiene en los equipos que serán afectados con la configuración emulada.

Los datos que tendrán esta tabla estarán en base al problema que se viene tratando desde el principio de la implementación de la metodología.

Tabla 27

Estructura de la tabla de configuración

Fecha:							
Descripción:							
Nombre:							
N°	**Código**	**Listas de acceso**	**IP**	**Puertos**	**MAC**	**Vlan**	**Modo**
1							
2							
3							
.							
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
n							

Fuente: Elaboración propia

La información de estas tablas de inventario y características nos permitirá tener conocimiento de los equipos a modificar y sus características además de la configuración que tienen antes de aplicarle la configuración que se emuló.

b) Aplicación de configuración simulada

Seguidamente se deberá aplicar la configuración que previamente se había hecho en el ambiente de pruebas del emulador, solo deberá ser aplicada la configuración necesaria.

Adicionalmente se procede con la aplicación del script que se encargará de extraer datos para analizarlos de manera continua para así poder buscar un patrón de alguna falla que vaya a ocurrir más adelante.

c) Monitoreo del levantamiento de la operación

Finalmente, todo el proceso deberá ser monitoreado por la persona a cargo en el caso de que se presente algún problema que pueda haber escapado de todos los casos planteados en el ambiente de prueba.

Para finalizar, el monitoreo del levantamiento de la operación se centrará en tener una conectividad constante de los equipos implicados, además de que estén ofreciendo el servicio a los usuarios finales.

La finalidad es que se apliquen los cambio emulados y los equipos no fallen por lo tanto se puede usar pruebas de conectividad simple enviando paquetes a los equipos o se pueden optar por alternativas de monitoreo gratuitas, con la finalidad de poder monitorear los nodos.

PRTG Paessler Router Traffic Grapher es un Sistema de monitoreo de redes, ofrece diferentes herramientas como uso de ancho de banda, conectividad, monitor snmp, estadísticas de conexión, etc.

Este es un sistema comercial que ofrece su versión de paga gratis por 30 días, pasada la versión de prueba algunas herramientas se bloquearan hasta adquirir una licencia.

La versión gratuita es suficiente para poder monitorear la conectividad en varios equipos a la vez.

Figura 64. Logo de PRTG

Fuente: Elaboración propia

Luego de realizar la instalación del sistema de monitoreo procederemos crear un grupo donde agregaremos todos los equipos a analizar.

Para este ejemplo el nombre del grupo es Grupo_Equipos_Analizados.

Figura 65. Creación de grupo de administración de dispositivos

Fuente: Elaboración propia

Automáticamente el grupo se añadirá a la raíz Sonda Local, la cual es la encargada de contener todos los grupos y equipos que serán administrados.

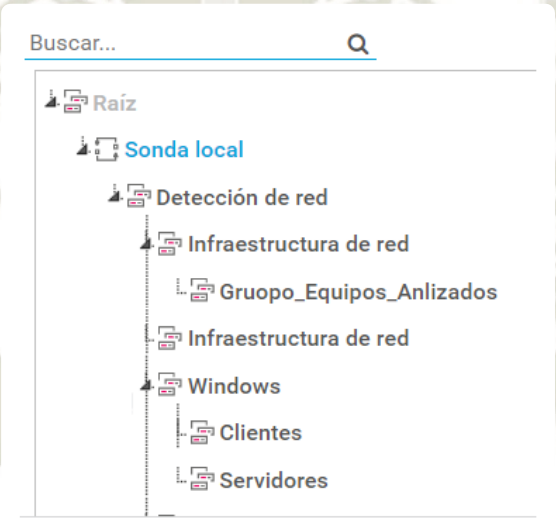

Figura 66. Ramas del nodo raíz

Fuente: Elaboración propia

Seguidamente procederemos a añadir todos los dispositivos que vamos a administrar desde el sistema de monitoreo, en esta sección nos ayudaremos de las tablas de inventario que realizamos anteriormente, de aquí sacaremos la información requerida para agregar los dispositivos.

Normalmente el sistema de monitoreo nos pide un nombre y una dirección de red para poder agregar el dispositivo, de requerir algún campo más se deberá consultar con las tablas de inventario y características.

Figura 67. Integración del equipo L2A1E2 al grupo

Fuente: Elaboración propia

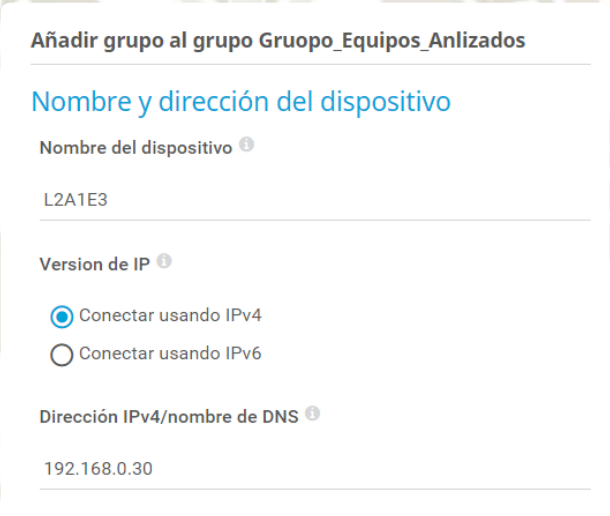

Figura 68. Integración del equipo L2A1E3 al grupo

Fuente: Elaboración propia

Figura 69. Integración del equipo L2A1E4 al grupo

Fuente: Elaboración propia

Una vez agregados todos los equipos que se van a analizar se mostrarán de la siguiente forma, en esta sección podemos ver que los equipos no tienen sensores, lo siguiente que se debe realizar es el agregado de sensores ya que permitirán el monitoreo de los equipos.

Figura 70. Equipos agregados correctamente

Fuente: Elaboración propia

Se procede a agregar los sensores que permitirán el monitoreo de los equipos la cual consta de la operación del protocolo de control de mensajes de internet (ICMP).

Este sensor permitirá enviar paquetes ICMP a los equipos para probar la conectividad, y alertarnos en el caso de que se pierda la conexión o se saturen.

Dado que el problema que se está abordando para solucionar son los bucles de capa enlace de datos se ha optado por agregar solo este sensor, en caso de que se desee solucionar otro problema de enlace de datos se deberán buscar los sensores para poder agregarlos a los equipos.

Figura 71. Configuración del sensor ICMP

Fuente: Elaboración propia

Una vez agregado el sensor de ICMP nos mostrará una etiqueta de color rojo, esta etiqueta permanecerá hasta que se logre tener una conexión sin intermitencias con el equipo que se esté analizando.

Figura 72. Sensor ICMP en equipos analizados

Fuente: Elaboración propia

PRTG permite la creación de mapas topológicos para tener un control grafico sobre los dispositivos administrados, si se desea se pueden agregar las

topologías necesarias, cabe resaltar que también se mostrará una etiqueta en rojo hasta que se haya completado la conectividad vía ICMP con los dispositivos analizados.

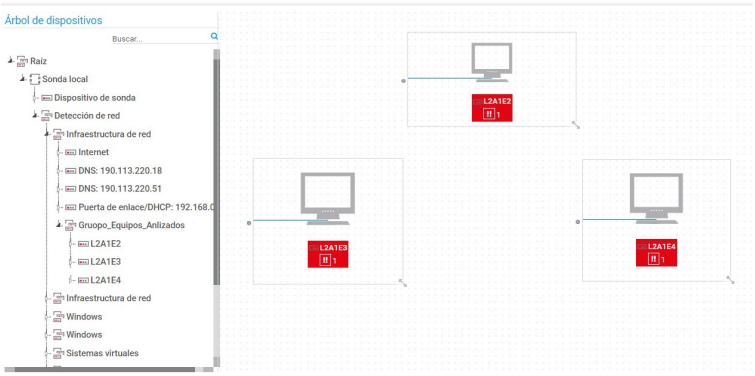

Figura 73. Topología en PRTG de los equipos analizados

Fuente: Elaboración propia

Una vez que se tenga él envió y respuesta correcta vía ICMP se mostrarán las etiquetas en verde, lo indica que el equipo está ofreciendo una conectividad correcta.

Figura 74. Sensores agregados

Fuente: Elaboración propia

Es aquí donde inicia el monitoreo de los equipos, pues la razón principal es ofrecer un servicio no interrumpido de conexión a los usuarios, al aplicar la configuración emulada es muy poco probable que se pueda producir un incidente, es decir no significa que no vaya a ocurrir ningún incidente, el sistema de monitoreo nos ayudará a identificar si ocurre el incidente de esta

manera podremos estar alertas en caso ocurra y trataremos de solucionarlo lo más antes posible.

CAPITULO IV

4. RESULTADOS

En esta sección se muestran los resultados obtenidos de aplicar las técnicas de aprendizaje automático en los datos extraídos de equipos de red, además se muestra el resultado de la emulación y configuración que deberá aplicarse en equipos de red.

Se procedió con la configuración del switch principal al cual irán conectados todos los equipos de red a analizar, y se le aplicó la configuración necesaria para que haya comunicación entre los equipos y este sin mezclar datos de la red interna.

Se configuro la Vlan 10 en modo de acceso en cada una de las interfaces directamente conectadas a los equipos para poder realizar la comunicación.

```
!
interface FastEthernet0/1
 switchport access vlan 10
 switchport mode access
!
interface FastEthernet0/2
 switchport access vlan 10
 switchport mode access
!
interface FastEthernet0/3
 switchport access vlan 10
 switchport mode access
!
interface FastEthernet0/4
 switchport access vlan 10
 switchport mode access
!
interface FastEthernet0/5
 switchport access vlan 10
 switchport mode access
!
```

Figura 75. Configuración del equipo directamente conectado a los demás

Fuente: Elaboración propia

A los equipos a analizar se les activó el protocolo SNMP con las operaciones de lectura y escritura lo que permitirá que el protocolo pueda extraer la información con la operación GET y a la vez también se puedan configurar parámetros del equipo con la operación SET.

A las interfaces que están conectadas al switch principal se les configuró la vlan 10 en modo acceso, además se les colocó una dirección IP a las vlan creadas con el fin de poder indicar al script la dirección de donde debe sacar la información vía SNMP.

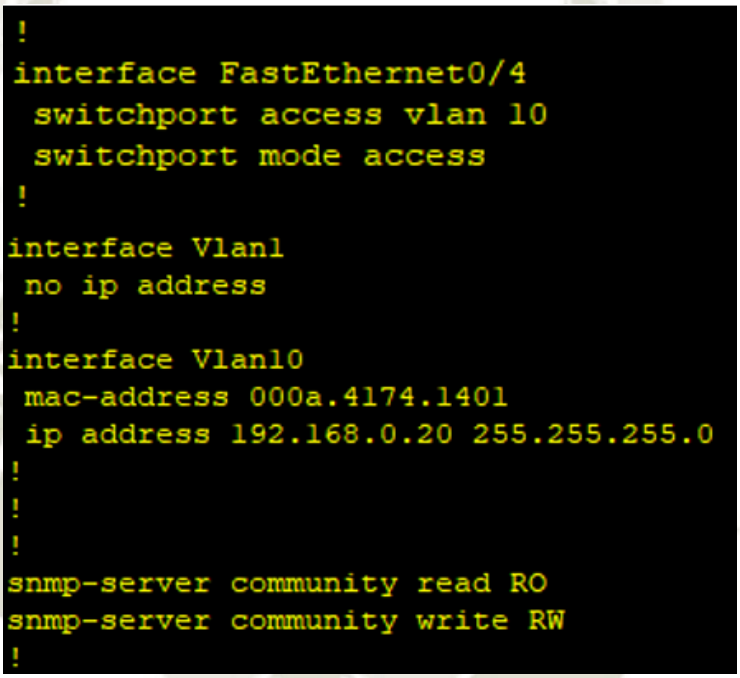

Figura 76. Configuración del equipo que será analizado

Fuente: Elaboración propia

Se llevó acabo el análisis de las versiones de SNMP adecuadas para poder realizar la extracción de datos, del cual se pudo determinar que la más adecuada era la versión 2,

La versión 3 refuerza los aspectos de seguridad, incluye autenticación, privacidad y control de acceso, lo que a su vez hace más pesada la trama de red tanto las de solicitud como las de respuesta, el enlace dedicado no debe saturarse es por eso que se configuró la versión 2 de SNMP que es menos pesada, el canal permitirá mayor cantidad de paquetes incluso cuando el enlace esté saturado.

Los aspectos de seguridad son importantes y no se deben dejar de lado, lo recomendable es aplicar una lista de acceso estándar que permita que una computadora sea la que haga

las solicitudes SNMP, lo que permitirá que solo esa computadora se comunique con los equipos y en el caso de que haya alguna intrusión esta no pueda afectar en lo absoluto a los equipos que está ejecutando SNMP versión 2.

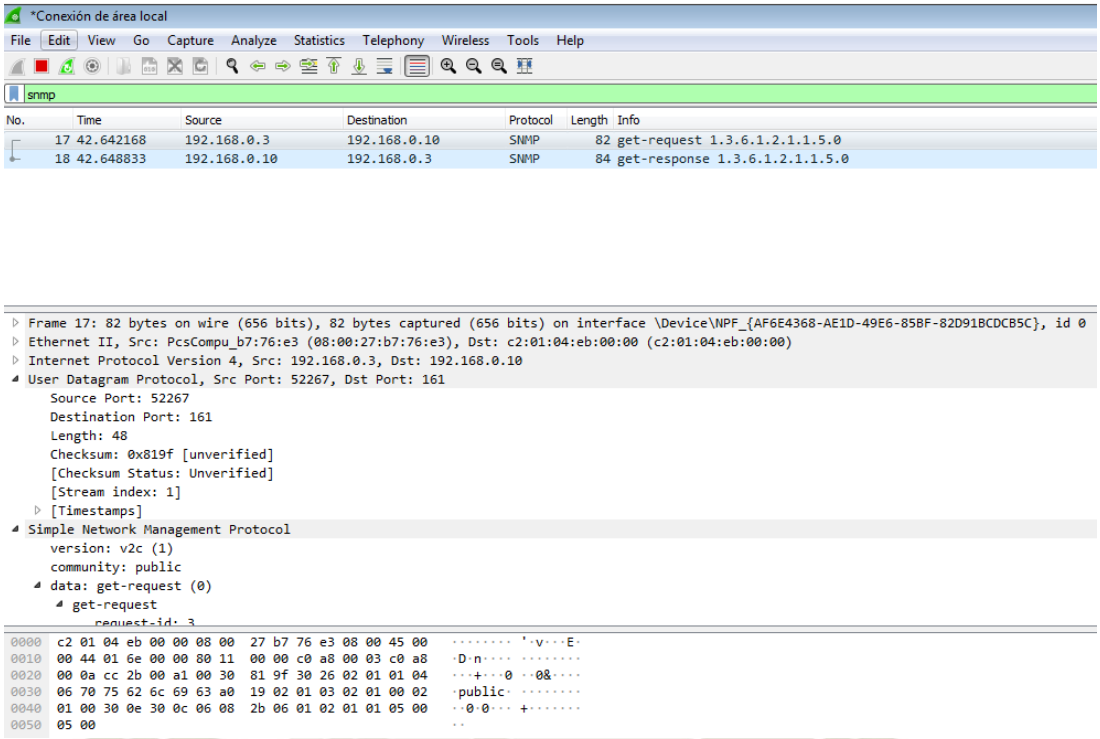

Figura 77. Captura de trama Get-request de snmp versión 2

Fuente: Elaboración propia

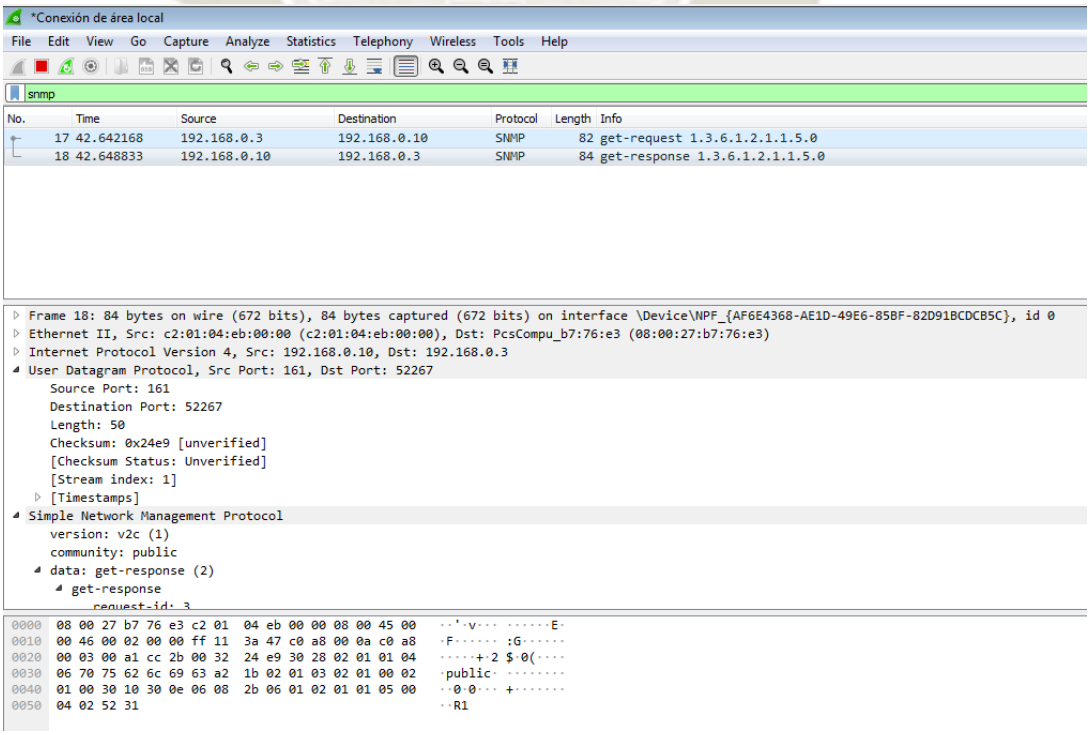

Figura 78. Captura de trama Get-response de snmp versión 2

Fuente: Elaboración propia

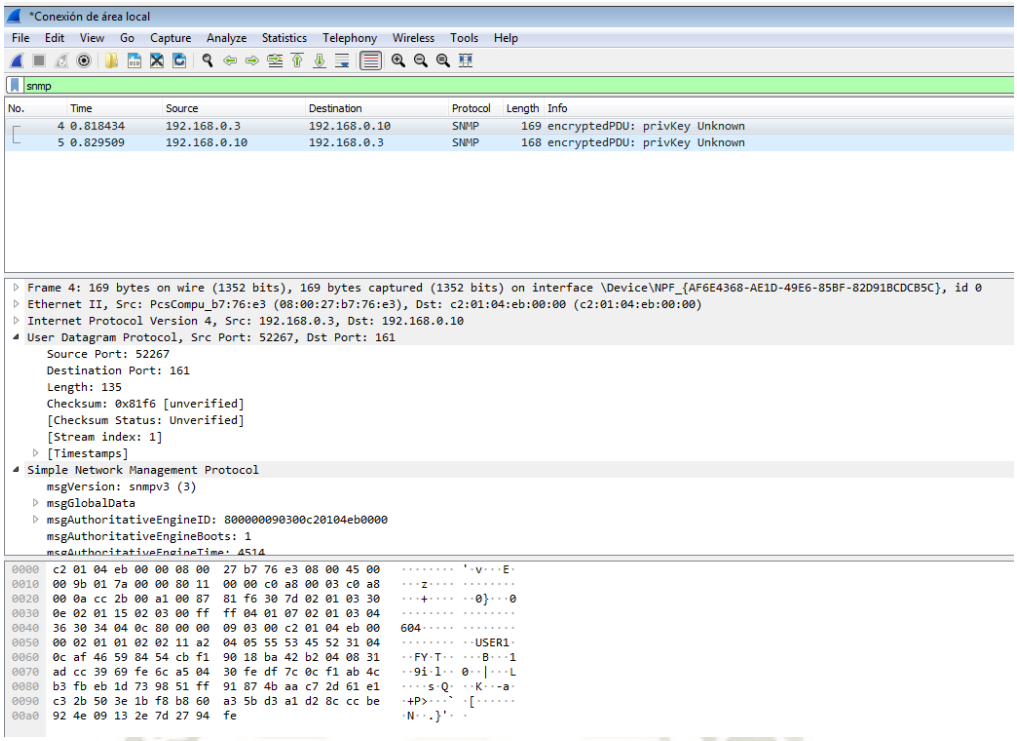

Figura 79. Captura de trama Get-request de snmp versión 3

Fuente: Elaboración propia

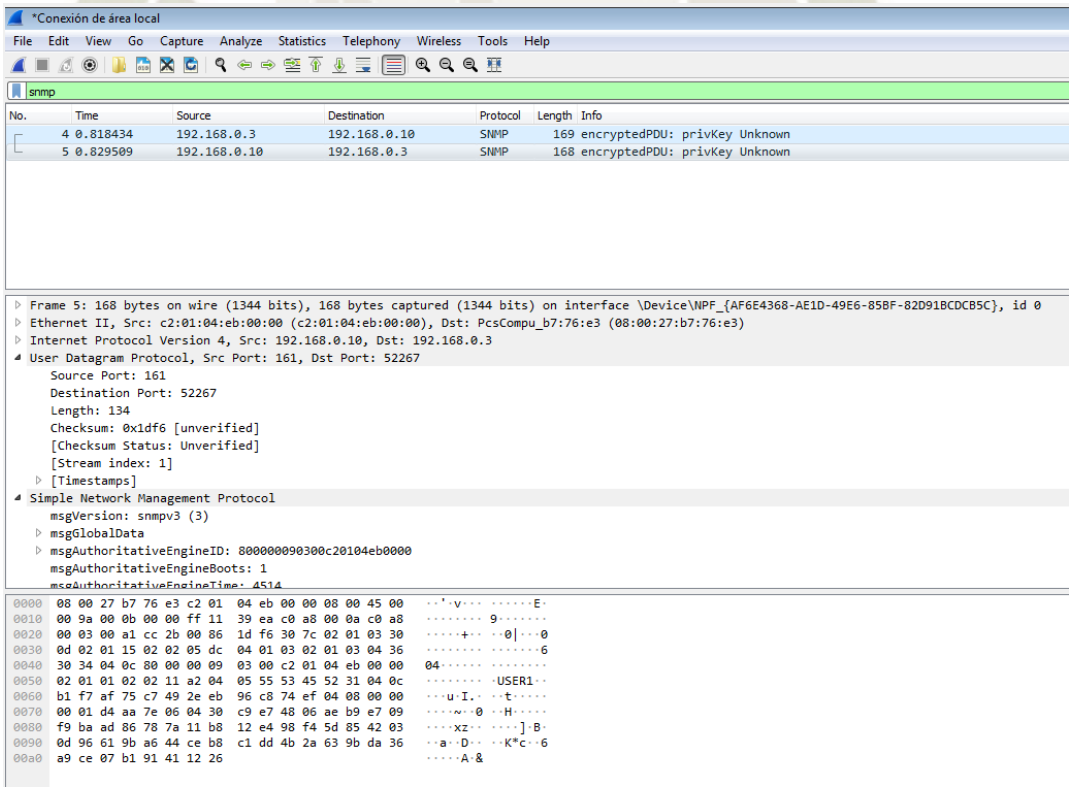

Figura 80. Captura de trama Get-response de snmp versión 3

Fuente: Elaboración propia

Se realizó el inventario correspondiente de los equipos, además se extrajo la información necesaria para colocarla de manera resumida en tablas con la finalidad de saber cuál era la configuración que tenía antes de iniciar el proceso de la aplicación de la configuración simulada.

En la figura 67 se muestra el inventario realizado, también se tomó en cuenta el equipo que será el encargado de extraer la información y analizarla.

Fecha: 09/03/2020							
Descripción: Inventario de equipos							
Nombre: Jorge Esquivel Rodríguez							
N°	**Código**	**Equipo**	**Marca/Modelo**	**Capa**	**Area**	**Estado**	**Prioridad**
1	L2A1E1	Switch	Cisco 2960 Series X	Enlace de datos	Principal (Core)	Activo	Alta
2	L2A1E2	Switch	Cisco 2960 Series X	Enlace de datos	Principal (Core)	Activo	Alta
3	L2A1E3	Switch	Cisco 2960 Series X	Enlace de datos	Principal (Core)	Activo	Alta
4	L2A1E4	Switch	Cisco 2960 Series X	Enlace de datos	Principal (Core)	Activo	Alta
5	L3A1E5	PC	Lenovo Legion y7000p	Red	Analisis	Activo	Baja

Figura 81. Inventario de los equipos analizados

Fuente: Elaboración propia

Fecha: 09/03/2020								
Descripción: Caracteristicas de equipos								
Nombre: Jorge Esquivel Rodríguez								
N°	**Código**	**Equipo**	**Serie**	**CPU**	**Dram**	**Memoria Flash**	**Puerto**	**Tipo**
1	L2A1E1	Switch	FOC103248MJ	APM86392 600 MHz dual core	512 MB	128 MB	24 10/100 2 1000	Fastethernet/ Gigabitethernet
2	L2A1E2	Switch	FOC1033Z1EY		512 MB	128 MB	24 10/100 2 1000	Fastethernet/ Gigabitethernet
3	L2A1E3	Switch	FOC10363GZU		512 MB	128 MB	24 10/100 2 1000	Fastethernet/ Gigabitethernet
4	L2A1E4	Switch	FOC1049VGP1		512 MB	128 MB	24 10/100 2 1000	Fastethernet/ Gigabitethernet
5	L3A1E5	PC	-	Intel Core I7	-	-	1 100	Fastethernet

Figura 82. Características de los equipos analizados

Fuente: Elaboración propia

La información resumida mostrada en la siguiente tabla permitirá identificar el origen que pudo causar algún problema, dado que el análisis hecho con la máquina de soporte vectorial nos indicará el equipo y la característica que influyó más en la falla ocurrida o próxima a ocurrir.

Fecha: 09/03/2020

Descripción: Configuración de equipos

Nombre: Jorge Esquivel Rodríguez

Nº	Código	Listas de acces	Interfaz	IP	Macara	MAC	vlan	Modo
1	L2A1E1	Standard Permit 192.168.1.3	fastethernet 0/1	-	-	0002.4a8c.d304	10	access
			fastethernet 0/2			0060.5cc2.ac04	10	access
			fastethernet 0/3			0050.0f2b.9b04	10	access
			fastethernet 0/4			-	10	access
			fastethernet 0/5			-	10	access
			Vlan 10			-	-	-
			Vlan 1			-	-	-
2	L2A1E2	Standard Permit 192.168.1.3	fastethernet 0/1	-	-	0050.0f2b.9b01	1	-
			fastethernet 0/2			0060.5cc2.ac01	1	-
			fastethernet 0/3				1	-
			fastethernet 0/4			0090.2b77.7401	10	access
			Vlan 10	192.168.0.20	255.255.255.0	000a.4174.1401	-	-
			Vlan 1	-	-	-	-	-
3	L2A1E3	Standard Permit 192.168.1.3	fastethernet 0/1	-	-	-	1	-
			fastethernet 0/2			0060.5cc2.ac02	1	-
			fastethernet 0/3			-	1	-
			fastethernet 0/4			0090.2b77.7403	10	access
			Vlan 10	192.168.0.30	255.255.255.0	000a.f34d.5a01	-	-
			Vlan 1	-	-	-	-	-
4	L2A1E4	Standard Permit 192.168.1.3	fastethernet 0/1	-	-	0002.4a8c.d302	1	-
			fastethernet 0/2			0050.0f2b.9b02	1	-
			fastethernet 0/3			-	1	-
			fastethernet 0/4			0090.2b77.7402	10	access
			Vlan 10	192.168.0.40	255.255.255.0	0009.7cad.7401	-	-
			Vlan 1	-	-	-	-	-

Figura 83. Resumen de la configuración de los equipos analizados

Fuente: Elaboración propia

Una de las fases de la metodología implicaba la creación o configuración de un canal dedicado para la extracción de información de los equipos de red, el canal puede ser alámbrico o inalámbrico, como se muestra en la figura 67 los canales deben ir conectados directamente desde el switch principal a los switches de los cuales se sacara la información necesaria, lo más adecuado fue conectar los enlaces en interfaces que tengan un ancho de banda considerable para que no pueda haber problemas de saturación más adelante.

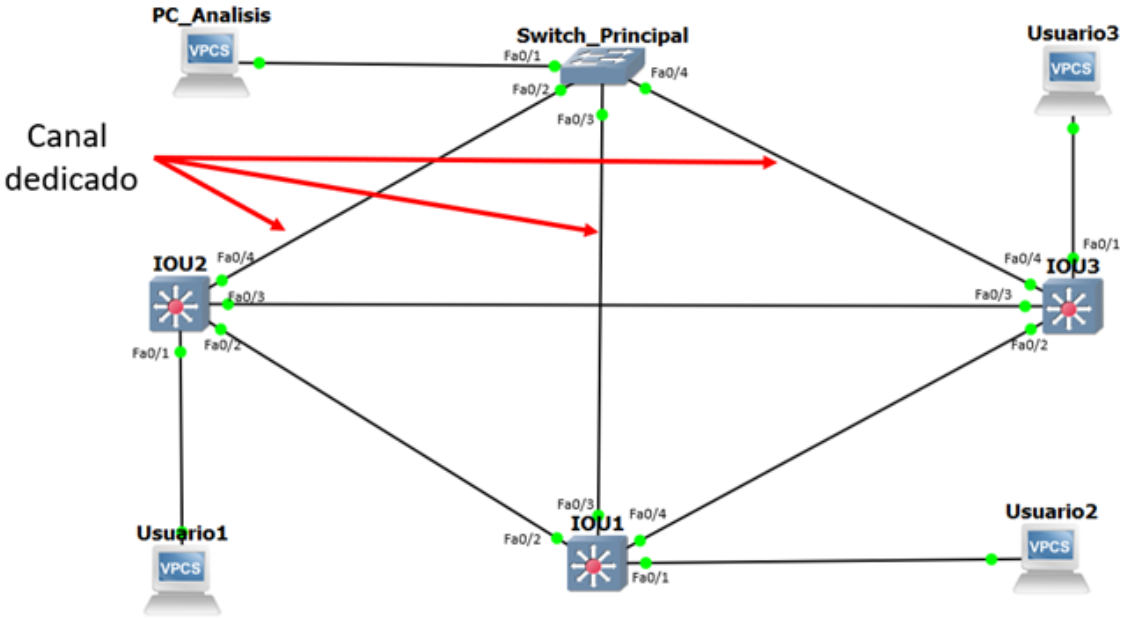

Figura 84. Topología planteada

Fuente: Elaboración propia

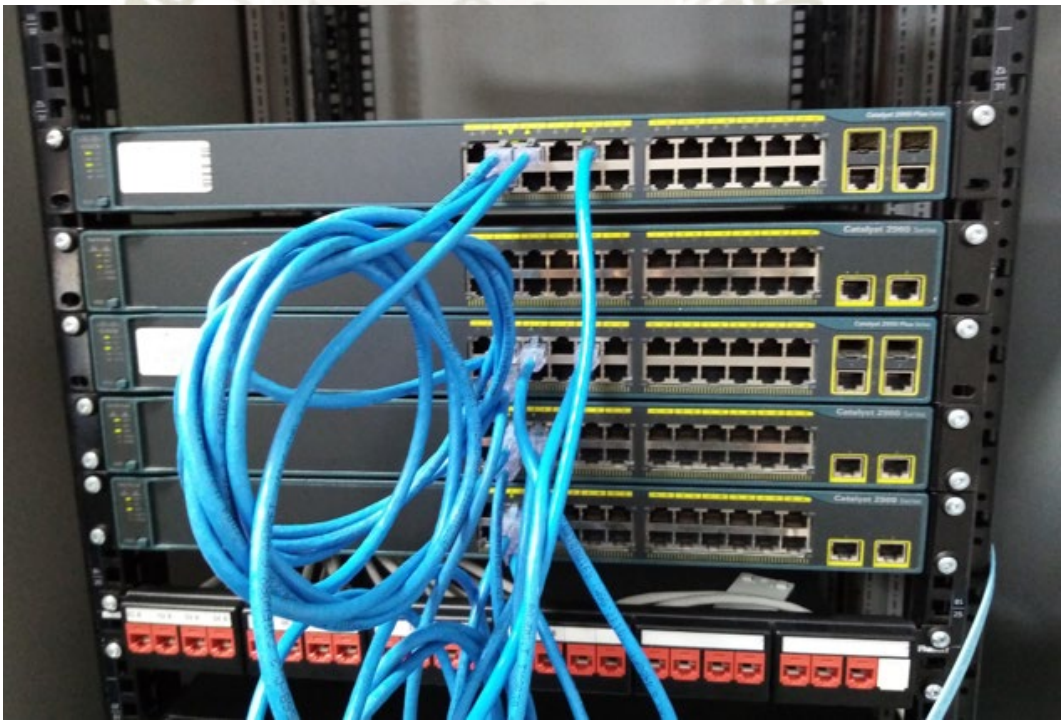

Figura 85. Dispositivos que serán puestos a prueba

Fuente: Elaboración propia

Se procede a inducir a los equipos en una falla anticipada para poder probar el correcto funcionamiento del script desarrollado anteriormente, mediante la técnica de aprendizaje automático elegida se podrá identificar el problema y se dará solución al mismo.

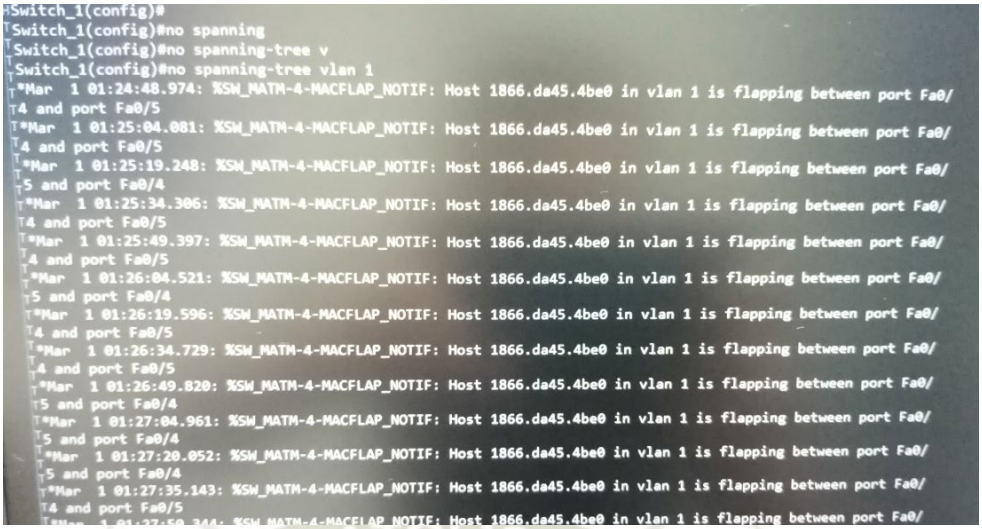

Figura 86. Desactivación del protocolo de árbol de expansión

Fuente: Elaboración propia

Dado que no hay ningún mecanismo que pueda controlar el flujo de tráfico en un bucle los canales de red se empiezan a saturar por lo tanto se empieza a perder conectividad con todos los equipos.

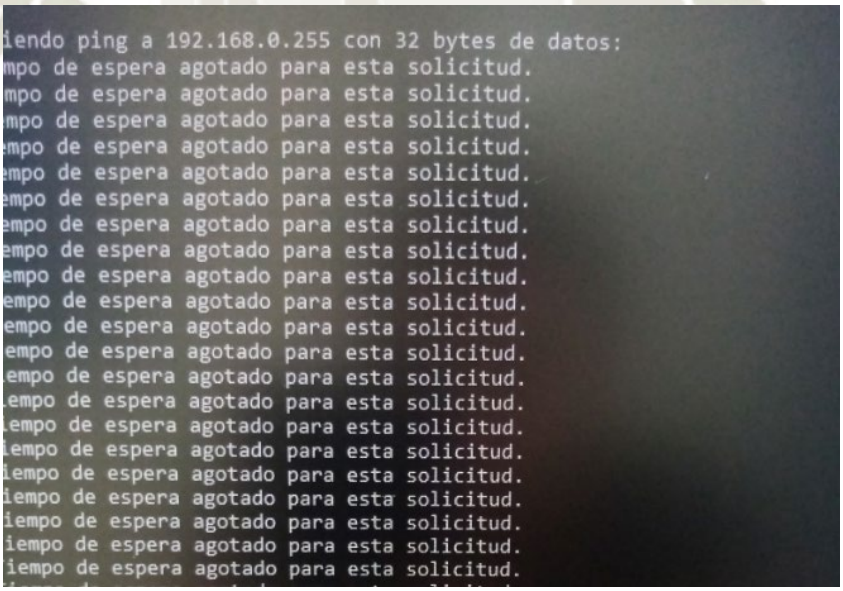

Figura 87. Envío de paquetes comenzando a fallar

Fuente: Elaboración propia

Seguidamente en PRTG se empiezan a activar las alarmas debido a que empiezan a fallar los equipos.

Figura 88. Alarmas debido a la perdida de conectividad

Fuente: Elaboración propia

A continuación, se muestra la tabla con los resultados de las técnicas aplicadas, las técnicas fueron elegidas de acuerdo con el tipo de información que se obtuvo mediante SNMP de los equipos de red, cabe resaltar que fueron técnicas de aprendizaje supervisado, adicionalmente se aplicó el análisis de datos mediante redes neuronales artificiales, finalmente la técnica adecuada fue la máquina de soporte vectorial o SVM.

Tabla 28

Resultados de las técnicas aplicadas

Técnica aplicada	**Resultado de la predicción**
Regresión Polinómica	95.9%
Support Vector Machine (SVM)	97.8%
Gradient Boosting (XGBoost)	89.3%
Redes Neuronales Artificiales	79.6%

Fuente: Elaboración propia

A continuación, se muestran las matrices de confusión donde se confirma el resultado de predicción de las técnicas aplicadas a los datos extraídos.

Podemos ver los valores que serían los verdaderos positivos de manera diagonal, dichos valores son las variables dependientes de los registros obtenidos.

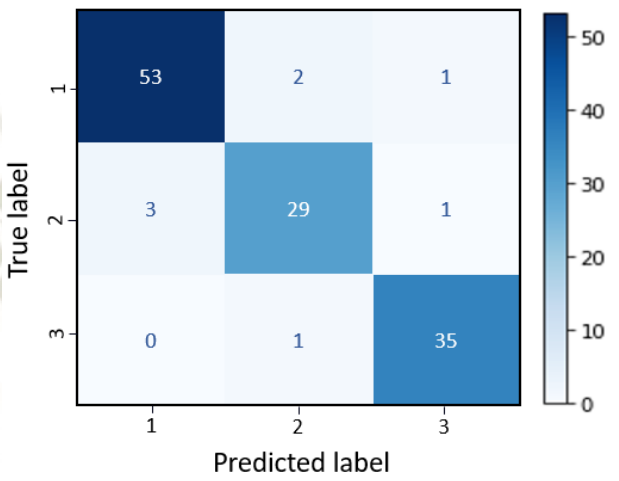

Figura 89. Matriz de confusión de la regresión polinómica

Fuente: Elaboración propia

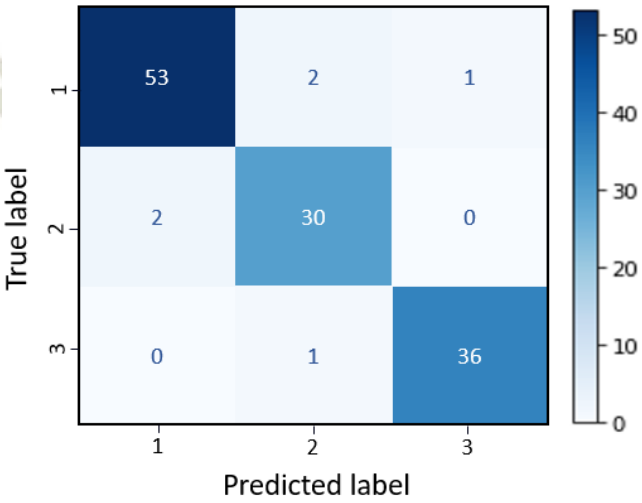

Figura 90. Matriz de confusión de la máquina de soporte vectorial

Fuente: Elaboración propia

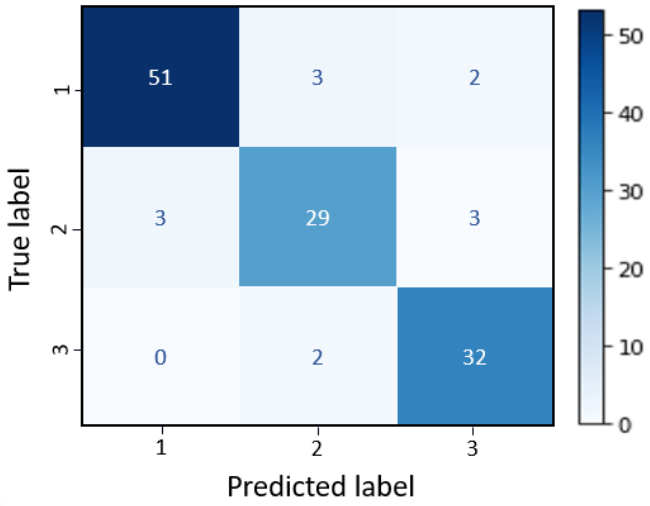

Figura 91. Matriz de confusión de gradient boosting

Fuente: Elaboración propia

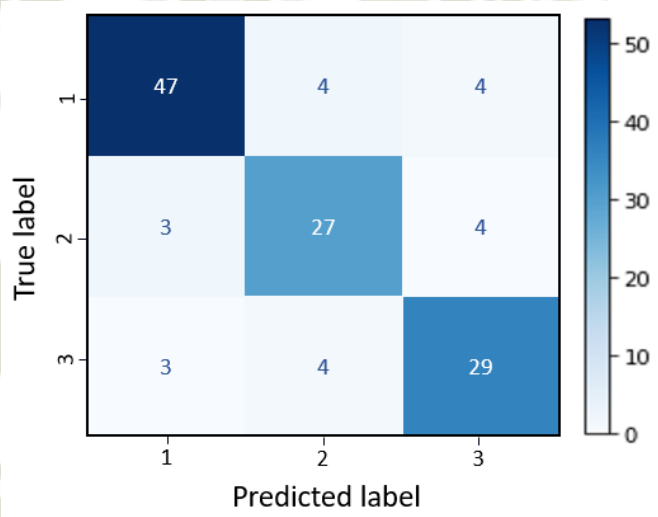

Figura 92. Matriz de confusión de redes neuronales

Fuente: Elaboración propia

Luego de aplicar la configuración necesaria debido a que se produjo una falla los canales de comunicación comienzan a recuperarse, la técnica de aprendizaje automático logró predecir con anticipación la falla que iba a ocurrir.

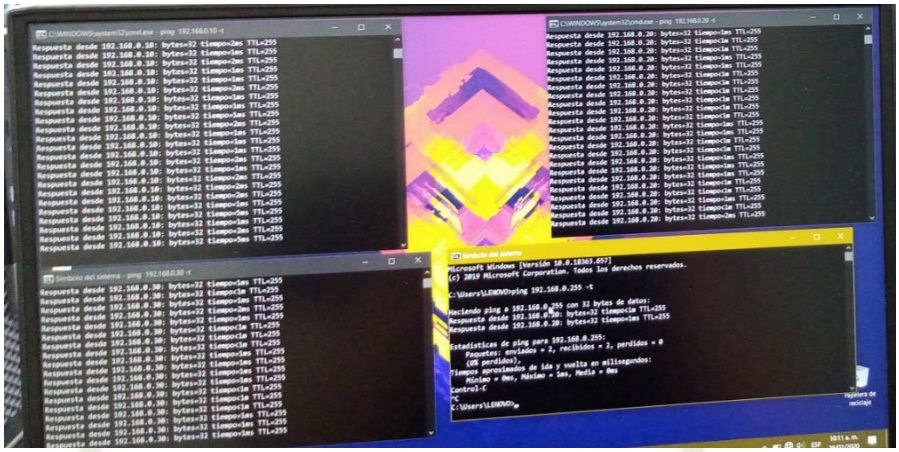

Figura 93. Conectividad sin intermitencia con los equipos de red

Fuente: Elaboración propia

5. Análisis y discusión de los resultados

Al aplicar la configuración propuesta en el capítulo 3 y conectando canales dedicados a los equipos a analizar se pudo extraer la información de los equipos de red incluso cuando estaban fallando, la información fue extraída hasta pocos segundos antes del fallo total lo que indica que esa configuración la necesaria para poder extraer datos a analizar.

Por otro lado, para poder extraer la información mediante tramas que no sean muy pesadas se optó por utilizar el protocolo simple de administración de red en su versión 2 (SNMP v2c), en las imágenes que se muestran en el apartado de resultados se ve que la petición por parte del protocolo simple de administración de red en su versión v2c (get-request) tiene un tamaño mínimo de 82 bytes teniendo un total de 656 bits y la respuesta por parte de este mismo protocolo (get-response) tiene un tamaño de 84 bytes teniendo un total de 672 bits.

El tamaño de las tramas aplicando la configuración del protocolo simple de administración de red en su versión 3 (SNMP v3) es diferente, la petición por parte de este protocolo en la versión 3 (get-request) tiene un tamaño de 169 bytes teniendo un total de 1352 bits y la respuesta (get-response) tiene un tamaño de 168 bytes con un total de 1344 bits haciendo de esta última versión la más pesada.

La máquina de soporte vectorial es la técnica más adecuada para realizar el análisis en el conjunto de datos extraídos, logró una predicción del 97.8%, se elige esta ya que si se elige otra técnica con un porcentaje más elevado significará que esta sobre entrenada por lo tanto no podrá predecir de manera correcta las fallas.

CONCLUSIONES

Dada por concluida la implementación de la metodología propuesta para identificar patrones de falla y aplicar el proceso de recuperación proactiva de equipos de comunicación se llegaron a las siguientes conclusiones:

1. Se logró crear una metodología que permitirá identificar patrones de fallas en equipos de comunicación, además de también permitir un proceso de recuperación proactiva en los mismos gracias a la aplicación de algoritmos de aprendizaje automático.
2. Se logró aplicar las configuraciones necesarias en equipos de comunicación de capa enlace de datos (Capa 2) que permitirán extraer información de los mismos aún incluso cuando ocurra una falla, dado que además se implementaron canales dedicados directamente conectados a los equipos.
3. Se consiguió aplicar diferentes técnicas de aprendizaje automático supervisado, las cuales dieron un resultado optimo al predecir fallas de las cuales la que resultó efectiva fue la máquina de soporte vectorial (SVM) dando una probabilidad del 97.8% de predicción.
4. Se determinó que Python es el lenguaje de programación adecuado que se necesita para poder realizar la aplicación de las técnicas de aprendizaje automático y también para realizar el script que permite extraer datos mediante el protocolo SNMP de equipos de comunicación creando un flujo de paquetes no muy pesados.
5. Se logró proponer un emulador para poder aplicar las configuraciones y realizar las pruebas necesarias en los equipos de red previamente emulados.
6. Se lograron establecer políticas de configuración con 4 estados las cuales permitieron proponer 3 formas de recuperación proactiva de equipos de comunicación.
7. Se constató que la versión 2 del protocolo simple de administración de red es la que se requiere para poder extraer la información necesaria para analizar.
8. La máquina de soporte vectorial pudo identificar una tormenta de broadcast ocasionada por el fallo del funcionamiento del protocolo de árbol de expansión, por lo tanto el número de paquetes de entrada y salida empezaron a subir constantemente segundo a segundo, por lo tanto, el CPU del equipo empezó a sobrecargarse lo que origino que no procese todos los paquetes originando fallos en la red, cabe resaltar que el tráfico de red aumentó considerablemente originando perdida del servicio de red.

RECOMENDACIONES Y TRABAJOS FUTUROS

Se proponen las siguientes recomendaciones:

- Es recomendable disponer de una gran cantidad de bases de información gestionada con la finalidad de poder encontrar un mayor número de OID's, lo cual permitirá extraer una mayor cantidad de información de los equipos analizados.
- Se recomienda crear uno mismo la base de información gestionada (MIB) usando la librería pysnmp o algún software, de esta manera minimizaremos el procesamiento de búsqueda en diferentes MIB a una sola, lo que traerá muchos beneficios y que también se pueden configurar una amplia gama de variables a modificar gracias a la operación SET.
- Un modelo de aprendizaje automático aprende mucho mejor cuando tiene los datos correctos para modelarse, es por eso que se recomienda inducir a los equipos a diferentes fallas en el emulador de esta manera se podrán extraer más datos de diferentes estados y fallas, y así se podrán solucionar más problemas.
- A mayor cantidad de procesamiento se necesita un equipo que sea capaz de poder realizar todas las operaciones sin ningún problema, para el análisis de datos se recomienda el uso de computadoras con procesadores no menores a I7 de octava generación y una capacidad de memoria ram no menor a 8 GB.
- Hoy en día el procesamiento gráfico está disponible en muchos sistemas de análisis, el procesamiento en paralelo es una opción para este trabajo de investigación ya que permitirá realizar muchos cálculos en paralelo para llegar a una respuesta optima de manera rápida, para lo cual es posible usar la herramienta CUDA que permitirá trabajar al máximo las tarjetas gráficas.
- Para poder recopilar los datos de los equipos se propuso un canal dedicado cableado, es posible usar un método inalámbrico en el caso de que los equipos puedan quedar lejos de la base principal de análisis, en este caso se deberá usar un repetidor de red capaza de poder englobar y alcanzar todos los equipos.

REFERENCIAS BIBLIOGRÁFICAS

1&1 Ionos España S.L.U. (2019). *Protocolos de red, la base de la transmisión electrónica de datos*. Obtenido de https://www.ionos.es/digitalguide/servidores/know-how/los-protocolos-de-red-en-la-transmision-de-datos/

Analytics, Business Intelligence and Data Management. (2018). *Aprendizaje automático: Qué es y por qué es importante.* Recuperado 30 diciembre, 2019, de https://www.sas.com/es_pe/insights/analytics/machine-learning.html.

Arias Paredes, Á. (2019). *Análisis de vulnerabilidades de servidores virtuales, caso práctico servicios web informativos de la ESPOCH.* Chimborazo.

Ariganello, E. (2016). *Redes cisco: guía de estudio para la certificación CCNA Routing y Switching.* (4ª ed.). Bogotá: Ediciones de la U.

Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). *Network Anomaly Detection: Methods, Systems and Tools.* IEEE Communications Surveys & Tutorials, 16(1), 303–336. Recuperado de https://ieeexplore.ieee.org/document/6524462.

Broadcom. (2019). *Identificadores de objetos (OID) del Protocolo simple de administración de redes (SNMP) del Symantec Messaging Gateway.* Recuperado 31 diciembre, 2019, de https://support.symantec.com/es/es/article.TECH227540.html.

Burgueño, L., Cabot, J., & Gérard, S. (2015). *Transformaciones de Datos con Machine Learning.*

Cajamarca, M. (2019). *Inteligencia Artificial, Aprendizaje Automático y Aprendizaje Profundo.* Recuperado 30 diciembre, 2019, de https://planetachatbot.com/inteligencia-artificial-aprendizaje-autom%C3%A1tico-y-aprendizaje-profundo-862ca9790bb9.

Castro Flores, C. A., Guillen Asencio, J. M., & Riera Barraza, J. J. (2010). *Resmatización de un sistema operativo de libre distribución con aplicaciones de carácter pedagógico para ser utilizado en el Área de Educación básica del Colegio Evangélico Centroamérica (CEMCA).*

Choi, J., Hu, K., & Antoniades, D. (2013). *Advanced Performance Modeling with Combined Passive and Active Monitoring (Annual Project Report – 2nd year).* Recuperado de https://sdm.lbl.gov/apm/docs/APM-Yr2-Report.pdf.

Cisco Systems. (2016). *Causas comunes del IntraVLAN lento y Conectividad del InterVLAN en las redes del Campus Switch.* Recuperado de https://www.cisco.com/c/es_mx/support/docs/lan-switching/virtual-lans-vlan-trunking-protocol-vlans-vtp/23637-slow-int-vlan-connect.pdf.

Deivid, D. (2018). *Bucles en la capa 2, o cómo arrancarte el pelo administrando una red.* Recuperado 30 diciembre, 2019, de https://deividsdocs.wordpress.com/2015/10/25/bucles-en-la-capa-2-switching-loops-broadcast-storm/.

Díaz. (2013). *Diseño, construcción y validación del sistema de monitoreo y adquisición de datos (housekeeping) para el foto detector a bordo de la misión euso-balloon.* México: Universidad Nacional Autónoma De México.

Felipe, V., & Ramírez, E. (2014). *K-Medias empleando la GPU. In el Proceedings del III Simposio Científico y Tecnológico en Computación (SCTC), sesión de Posters.*

Gallardo Arancibia, J. (2009). *Metodología para la definición de requisitos en proyectos de data mining.*

Gómez, M. D. P. (2016). *Aprendizaje Profundo El poder del aprendizaje automático unido al poder de cálculo de las computadoras actuales.* Recuperado 30 diciembre, 2019, de http://ccc.inaoep.mx/~pgomez/conferences/PggTSys16.pdf.

Herran Arias. (2019). *Inteligencia artificial*. Obtenido de https://es.scribd.com/document/426958857/kj

Hidalgo, F., & Gamess, E. (2014). Integrating Android Devices into Network Management Systems based on SNMP. *Int. J. Adv. Comput. Sci. Appl, 5*(5), 1-8.

Iaarbook.github. (2019). *Libro online de IAAR*. Obtenido de Introducción al Machine Learning: https://iaarbook.github.io/machine-learning/

IBM Knowledge Center. (2018). *Protocolos TCP/IP.* Recuperado 30 diciembre 2019, de https://www.ibm.com/support/knowledgecenter/es/ssw_aix_72/network/tcpip_protocols.html.

INCIBE. (2019). *Monitorizando redes y eventos en SCI: más información, más seguridad.* Recuperado 30 diciembre, 2019, de https://www.incibe-cert.es/blog/monitorizando-redes-y-eventos-sci-más-informacion-más-seguridad.

Invid. (2019). *Algoritmos de aprendizaje automático - INVID.* Recuperado 30 diciembre, 2019, de https://invidgroup.com/es/algoritmos-de-aprendizaje-automatico/.

IONOS España S.L.U. (2019). *SNMP: el protocolo base para la gestión de redes.* Recuperado 30 diciembre, 2019, de https://www.ionos.es/digitalguide/servidores/know-how/snmp/.

Jaramillo, A., & Arias, H. P. P. (2015). Aplicación de Técnicas de Minería de Datos para Determinar las Interacciones de los Estudiantes en un Entorno Virtual de Aprendizaje. *Revista Tecnológica-ESPOL, 28*(1).

Khan, R., & Khan, S. U. (2014). An Efficient Network Monitoring and Management System. *International Journal of Information and Electronics Engineering, 3*(1). doi:https://doi.org/10.7763/ijiee.2013.v3.280

Lightsys. (2012). *Manual de Instalación y Programación* . Obtenido de http://descargas.hommaxsistemas.com//4.%20INTRUSION/Centrales%20Cableadas%20y%20Via%20Radio/RISCO/LIGHTSYS2/5INxxxx%20LightSYS%20Installer%20Manual%20ES.pdf

Martin, J. (2018). *¿Qué es machine learning?* Recuperado 30 diciembre, 2019, de https://www.cerem.pe/blog/que-es-el-machine-learning.

Martínez, C. Y. (2016). *NetFlow Vs SNMP. ¿Qué método usar?* Recuperado 30 diciembre, 2019, de https://techclub.tajamar.es/netflow-vs-snmp/.

Montenegro, D. (2016). *Propuesta de utilización de herramientas de telemetría, para identificar técnicas de ciberdelitos como watering hole, en redes de infraestructura (Caso de estudio netflow de cisco).* Recuperado 30 diciembre, 2019, de http://repositorio.puce.edu.ec/bitstream/handle/22000/8437/Tesis%20-%20Watering%20Hole.pdf?sequence=1&isAllowed=y.

Pandora FMS. (2019). *Monitoreo de Redes: 16 mejores herramientas de monitorización de redes.* Recuperado 30 diciembre, 2019, de https://pandorafms.com/blog/es/herramientas-de-monitoreo-de-redes/.

Rai, N. (2014). Designing a Network Based Intrusion Detection System using MIB with the aid of SNMP Agents. *International Journal of Engineering Research, 3*(3).

Recio Recio, J. (2010). *Técnicas de extracción de características y clasificación de imágenes orientada a objetos aplicadas a la actualización de bases de datos de ocupación del suelo.* Tesis para optar el título de doctor.

Reportedigital.com. (2019). *Monitoreo de red: características y herramientas | Reporte Digital.* Recuperado 28 diciembre, 2019, de https://reportedigital.com/seguridad/monitoreo-de-red/.

Rodriguez. (2020). *Servicios de red orientados a conexión y sin conexión*. Obtenido de http://dmrodriguez.50megs.com/Internetworking5.html

Rodríguez Tapia, S., & Camacho-Cañamón, J. (2018). *Los métodos de aprendizaje automático supervisado en la clasificación textual según el grado de especialización.*

Roughan, M., & Rocky, C. (2013). *Passive and Active Measurement: 14th International Conference.* PAM 2013, Hong Kong, China, March 18-19, 2013, Proceedings. Berlin, Alemania: Springer Berlin Heidelberg.

Rpubs.com. (2016). *Análisis discriminante lineal (LDA) y Análisis discriminante cuadrático (QDA).* Recuperado 31 diciembre, 2019, de https://rpubs.com/Joaquin_AR/233932.

SAS Institute. (2019). *Qué es y por qué es importante*. Obtenido de https://www.sas.com/es_mx/insights/analytics/data-mining.html

Seaccna. (2015). *Modelo OSI: ¡¡¡la guía definitiva del Modelo OSI!!!* Recuperado 19 marzo, 2020, de https://seaccna.com/modelo-osi-guia-definitiva/.

Sprockel, J. J., Diaztagle, J. J., Alzate, W., & González, E. (2014). Redes neuronales en el diagnóstico del infarto agudo de miocardio. *Revista Colombiana de Cardiología, 21*(4), 215-223.

Techtarget.com. (2017). *¿Qué es Aprendizaje automático (machine learning)?* Recuperado 28 diciembre, 2019, de https://searchdatacenter.techtarget.com/es/definicion/Aprendizaje-automático-machine-learning.

Teixeira, M. M. (2015). *Monitoring Wireless Networks Through Machine Learning Algorithms.* Recuperado de https://fenix.tecnico.ulisboa.pt/downloadFile/563345090414030/METI-68229-Miguel-Teixeira-Dissertacao.pdf.

Tejada. (2019). *Introduccion a las redes informaticas*. Obtenido de https://es.scribd.com/document/416554501/Manual-de-redes-basico

Tolosa, G. (2014). *Protocolos y Modelo OSI.* Recuperado de http://www. tyr. unlu. edu. ar/TYR-publica/02-Protocolosy-OSI. pdf.

Zambrano Montenegro, D. (2015). *Propuesta de utilización de herramientas de telemetría, para identificar técnicas de Ciberdelitos como Watering Hole, en redes de infraestructura (caso de estudio Netflow de Cisco)*. Obtenido de http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/174/A4.pdf?sequence=4

ANEXOS

Anexo A: Glosario de Términos

- **GET:** Mensaje snmp que permite retornar el valor de un objeto consultado.
- **GET-NEXT:** Mensaje snmp que tiene la finalidad de recorrer una tabla de objetos.
- **GET-BULK:** Mensaje snmp utilizado con la finalidad de transportar una gran cantidad de datos como es el caso de las tablas.
- **SET:** Mensaje snmp utilizado con la finalidad de cambiar valores de objetos.
- **HTTP:** Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto), protocolo que permite la transferencia de datos a través de la world wide web.
- **HTTPS:** Hypertext Transfer Protocol Secure (Protocolo de Transferencia de Hipertexto Seguro), protocolo que permite la transferencia de información de manera segura.
- **Nodos:** Referente a los equipos de red en una topología de red.
- **Broadcast:** Forma de transmisión de datos donde un nodo emisor envía mensajes a una multitud de receptores al mismo tiempo.
- **Conmutador:** Dispositivo de interconexión que opera en la capa enlace de datos.
- **Tráfico de red:** Problema que consiste en un gran número de paquetes con un retardo de procesamiento.
- **Conmutación de paquetes:** Forma de agrupar datos en paquetes través de una red.
- **Enrutamiento:** Función que consiste en buscar el camino optimo a través de una red para llegar al destino
- **Aprendizaje automático:** Campo de la inteligencia artificial centrado en desarrollar técnicas con la finalidad del aprendizaje de las computadoras.
- **Aprendizaje supervisado:** Técnica que se centra en la reducción de función a partir de datos de entrenamiento.
- **Proceso cíclico bidireccional:** Sucesión a seguir con la finalidad de retornar al punto de partida desde cualquier punto.

- **Serie 2960:** Serie de conmutadores cisco.
- **SMTP:** Simple Mail Transfer Protocol, protocolo que permite el envío de correos a través de internet.
- **Telnet:** Teletype Network, Protocolo que permite el acceso a otra red para manejarla de manera remota.
- **ACL:** Access Control List, Forma para determinar permisos de acceso y privilegios a diferentes objetos.
- **SDU:** Service Data Unit (Unidad de datos de servicio), nombre que se usa para describir la unidad de datos una vez que se deriva a una capa inferior o superior.
- **PDU:** Protocol Data Unit (Unidad de datos de protocolo), se usa para el intercambio de datos entre unidades.
- **Dynamips:** Software que tiene la finalidad de emular enrutadores y conmutadores de red.
- **Metodología:** Conjunto de procesos utilizados para llegar a un objetivo.
- **LAN:** Local Area Network (Red de Área Local), red de computadoras en un área reducida.
- **WAN:** Wide Area Network (Red de Área Amplia), red de computadoras localizadas en diferentes ubicaciones.
- **Dirección IP:** Conjunto de números de 32 bits que identifica de manera lógica una interfaz de red.
- **Dirección MAC:** Conjuntos de números de 48 bits que corresponde a una interfaz de red.
- **CPU:** Unidad Central de Procesamiento, parte principal de una computadora encargada de realizar múltiples tareas.

- **Nvram:** Memoria de acceso aleatorio no volátil, Tipo de memoria de acceso aleatorio que no pierde la información al cortar el flujo eléctrico.
- **Interfaz de red:** Componente de hardware cuya finalidad es conectar un equipo a la red.
- **CUDA:** Compute Unified Device Architecture, plataforma para el desarrollo de programas con ejecución en paralelo.