Universidad de Cienfuegos "Carlos Rafael Rodríguez"
Facultad de Ingeniería
Carrera de Ingeniería Informática

"Módulo de interfaz de comandos del sistema InterFirewall que se desarrolla para manejar cortafuegos de distintas tecnologías en ETECSA Cienfuegos"

Trabajo de diploma para optar por el título de Ingeniería en Informática

Autor:

Richard Darian Sánchez Rivero.

Tutores:

Msc. Denis Morejón López.
Msc. Anay Carrillo Ramos.

Cienfuegos, Cuba
Curso 2016 - 2017

Agradecimientos

A todos aquellos que me ayudaron a alcanzar esta meta, en especial:

A mi abuela Fermina que entregó todo en vida.

A mi mamá que siempre me apoya.

A mis dos hermanos Daylín y Aray que me han enseñado muchas cosas.

A Michel, el capitán de la familia, que siempre se puede contar con él.

A mis tío Carlitos y mi tía Cari que siempre me complacen.

A mi tía Toty, mi prima Adriana y mis primitos Michel y Dany que siempre han estado presente.

A mis dos tutores Denis Morejón López y a Anay Carrillo Ramos por confiar en mí.

A todos muchísimas gracias.

Dedicatoria

A toda mi familia y a todas las personas que quieren llegar a ser ingenieros informáticos.

Resumen

En la Empresa de Telecomunicaciones de Cuba S.A (ETECSA), en Cienfuegos, es necesario programar un sistema para el manejo dinámico de cortafuegos. El sistema manejaría en un principio el cortafuego que ya se utiliza en ETECSA para en un futuro incluir soporte para más tipos de cortafuegos. En dicha empresa se utilizan las ACLs de un Switch capa 3 de fabricante Huawei, específicamente el modelo Quidway s5328. Se propuso un módulo de línea de comandos, programado en el lenguaje de alto nivel Python, para manejar desde la red las reglas necesarias y comunes en todos los cortafuegos estudiando las características comunes de los mismos tanto como sus diferencias. Esto creó una capa de abstracción sobre las interioridades sintácticas de cada cortafuego para poder realizar las tareas de forma centralizada.

Palabras claves: manejo dinámico, módulo, cortafuegos.

Abstract

In the Telecommunications Company of Cuba S.A (ETECSA), in Cienfuegos, it is necessary to program a system for dynamic management of firewalls. The DTCF uses the ACLs of a Layer Switch 3, manufacturer Huawei model Quidway s5300. It was proposed a module programmed in Python to manage from the network the necessary and common rules in all the firewalls by studying the common characteristics. This created a layer of abstraction on the syntactic insights of each firewall to be able to perform tasks centrally.

.

Key words: dynamic management, module, firewall

Índice

Índice

Índice de tablas

Índice de figuras

Introducción

Hoy en día se hace más fácil adquirir un equipo de cómputo que 20 años atrás. Las organizaciones se arman de más microcomputadoras, y a su vez las interconectan en redes que cada vez son más complejas y numerosas. Esto trae consigo que también tengan que evolucionar los mecanismos de seguridad de las redes tanto para detectar como para bloquear el tráfico a usuarios que entorpezcan con el objetivo de la organización.

En la actualidad existen y se desarrollan en el mundo diversas medidas para el aseguramiento de las redes locales, una de ellas es implementar cortafuegos perimetrales. Estos son elementos claves en este escenario, ya que permiten a un administrador de red detener el tráfico de determinada PC si un sistema detector de intrusos (IDS) lo alertara sobre el comportamiento sospechoso de la misma. Las empresas cuentan con cortafuegos que son implementados de diversas maneras. Pueden configurarse tanto con hardware especializado, como con listas de control de acceso (ACLs) en los Router, como con software diseñado para ese propósito.

Una organización de hecho pudiera contar con cortafuegos de distintos tipos y fabricantes. Es justo esta diversidad de tecnologías el primer obstáculo para implementar políticas de control de acceso que abarquen toda la red de una manera sencilla de configurar. O sea, impedir el paso de una PC por varios segmentos de la red donde yacen distintos cortafuegos es tarea compleja porque hay que enfrentarse a la manera en que cada uno de ellos configura sus políticas.

Por otro lado, la decisión de bloquear una PC parte de la acción de un administrador y no se produce automáticamente por la indicación de un IDS (Sistema de Detención de Intruso) bien ajustado que detecte el tráfico malicioso y pase a los cortafuegos precisos los datos y la indicación del bloqueo al equipo invasor. Esto implica que el administrador tendrá que conocer cómo se crean las políticas de restricción del tráfico de la red para cada tipo de cortafuego y su fabricante correspondientemente.

Situación Problémica:

ETECSA cuenta con cortafuegos de distintos tipos y fabricantes. Hay que enfrentarse a la manera en que cada uno de ellos configura la regla ya que cada cortafuego usa un lenguaje propio a la hora de implementar las ACL. Cuando se quiere impedir el paso de información de una PC (computadora personal) por varios segmentos de la red donde yacen distintos cortafuegos es tarea compleja. Esto hace más engorroso el trabajo de los administradores de red.

Problema a resolver:

¿Cómo desarrollar un sistema para el manejo dinámico de cortafuegos en la división territorial de ETECSA en Cienfuegos?

Objeto de estudio:

La red informática en la división territorial de ETECSA en Cienfuegos.

Campo de acción:

Los cortafuegos implementados para controlar el tráfico de paquetes en la red.

Idea a defender:

Con la implementación del módulo de línea de comando desarrollada en el lenguaje de alto nivel Python, se logrará una mayor eficiencia y rapidez en el proceso de administración de la red informática de ETECSA en la DTCF.

Objetivo general:

Desarrollar una Interfaz de Línea de Comandos para manejar cortafuegos de distintas tecnologías y un driver para el Switch Huawei modelo Quidway S5328.

Objetivos específicos.

- Analizar las políticas de control de acceso de distintas tecnologías de firewall.
- Implementar un módulo de interfaz de comandos.
- Validar la propuesta mediante pruebas funcionales y escenarios de pruebas.

Tareas a desarrollar.

- Entrevistas al personal encargado de la seguridad de la red.
- Análisis de la implementación de ACL en los firewall Huawei, Iptables, Cisco y Juniper.
- Análisis del lenguaje de programación de alto nivel Python 2.7 y se haciendo énfasis en el Módulo PEXPECT.
- Programación de una Interfaz de Línea de Comandos y un driver para el Switch Huawei S5328
- Investigación sobre los protocolos de comunicación utilizados para la intercomunicación entre dispositivos de una red.
- La realización de un conjunto de escenarios de pruebas y pruebas funcionales.

Descripción de los capítulos.

Para una mejor comprensión de la presente investigación, la misma ha sido estructurada en capítulos, los cuales se describen a continuación.

Capítulo 1- Fundamentación Teórica:

El contenido de este capítulo es la base de la fundamentación teórica del tema que se va desarrollar. Se expone una descripción de los conceptos relacionados con el dominio del problema. Se describe el objeto de estudio, el flujo actual de los procesos, así como un análisis, describiendo la solución propuesta.

Capítulo 2- Construcción de la solución propuesta:

Se realiza un estudio para identificar los procesos principales. Se definen los actores del sistema. Se describe la aplicación propuesta identificando los requerimientos funcionales y no funcionales. La descripción de los casos de uso, diagrama de clases y el diagrama de implementación.

Capítulo 3- Validación y Estudio de Factibilidad:

En este capítulo se realiza un procedimiento de validación de la aplicación a través de escenarios de pruebas y pruebas funcionales. Se describe el proceso de planificación del estudio de factibilidad y se lleva a cabo la determinación de los costos, así como, los beneficios tangibles e intangibles asociados a la aplicación.

1 Fundamentación Teórica.

1.1 Introducción.

En el presente capítulo se lleva a cabo la fundamentación teórica del tema que se va desarrollar. Se realiza un estudio de los sistemas existentes relacionados con el campo de acción y el objeto de estudio. También se detallan los lenguajes y herramientas a utilizar en las que se basa el trabajo.

1.2 Descripción del dominio del problema.

En esta sección se da una descripción de los principales conceptos asociados al dominio del problema que son necesarios para comprender el objeto de estudio.

1.2.1 Dispositivos de interconexión.

Los dispositivos de interconexión tienen dos ámbitos de actuación en las redes telemáticas. En un primer nivel se encuentran los más conocidos, los router, que se encargan de la interconexión de las redes. En un segundo nivel estarían los Switch, que son los encargados de la interconexión de equipos dentro de una misma red, o lo que es lo mismo, son los dispositivos que, junto al cableado, constituyen las redes de área local o LAN. Un Switch o conmutador es un dispositivo de interconexión utilizado para conectar equipos en red formando una red de área local. [1]

Diferencias entre un Switch capa 2 y un Switch capa 3. [2]

Características	Switch Capa 2	Switch Capa 3
Control de tráfico	No hay control de tráfico de paquetes broadcast o multicast, en cuando se presencie una descarga se este tipo la red puede colapsar	Este tipo de Switch previene el colapso de la red, ante la presencia de descargas de broadcast y manejan eficientemente el trafico multicast
Rendimiento en el	Este Switch conectado a	Este Switch identifica si

manejo de tráfico de red	un Switch central de backbone, transfiere todos los paquetes a el Switch de backbone consumiendo innecesariamente recurso y tiempo en el backbone	el tráfico debe ser switcheado en capa 2 o en capa 3, y si debe de ser de manera local o al backbone, maneja sus propios recursos sin consumir ancho de banda ni generar tráfico innecesario
Tolerancia a fallas	No tiene muchos mecanismos para tolerancia a fallas, no cuenta con enlaces redundantes y si los tiene solo puede hacer uso de spanning tree	Cuenta con varios mecanismos de control de fallas y de respaldo tanto en capa 2 como de capa 3, los Switch capa 3 participan de los mecanismos de control de fallos en los enlaces, junto con los enrutadores para recuperar rápida e inteligentemente la conexión entre los recursos de la red
Seguridad	No cuenta con mecanismos de seguridad en la red. Cualquiera puede conectarse a sus puertos y generar cualquier tipo de tráfico e incluso puede escuchar información	Tiene todos los niveles de control y seguridad con los que un ruteador normalmente cuenta. Existen mecanismos de seguridad para prevenir que un usuario indeseado se conecte a la red, incluso a nivel

	sensible que este viajando en la red	físico. Estos switches pueden filtrar información no deseada incluso de los usuarios que tienen permitido el acceso a la red, cuentan con mecanismos de protección para evitar que un usuario no deseado pueda infiltrarse a la configuración del switch

Tabla 1. Diferencias entre switch capa 2 y switch capa 3

1.2.2 Firewall.

Un firewall (llamado también "cortafuego"), es un sistema que permite proteger a una computadora o una red de computadoras de las intrusiones que provienen de una tercera red (expresamente de Internet). El firewall es un sistema que permite filtrar los paquetes de datos que andan por la red. Se trata de un "puente angosto" que filtra, al menos, el tráfico entre la red interna y externa. Un firewall puede ser un programa (software) o un equipo (hardware) que actúa como intermediario entre la red local (o la computadora local) y una o varias redes externas. [3]

1.2.3 Interfaz de líneas de comandos.

Es una Interfaz de línea de comandos (o CLI, por sus iniciales en inglés) para manipular con instrucciones escritas al programa que subyace debajo. A esta interfaz se le acostumbra llamar Consola de sistema o consola de comandos. Se interactúa con la información de la manera más simple posible, sin gráficas ni nada más que el texto crudo. Las órdenes se escriben como líneas de texto (de ahí el nombre), y, si los programas responden, generalmente lo hacen poniendo información en las líneas siguientes. [4]

Casi cualquier programa puede diseñarse para ofrecer al usuario alguna clase de CLI. Por ejemplo, casi todos los juegos de PC en primera persona tienen una interfaz de línea de comandos incorporada, utilizada para diagnóstico y labores administrativas. Como herramienta primaria de trabajo, las líneas de comandos son principalmente usadas por programadores y administradores de sistemas, especialmente en sistemas operativos basados en Unix; en entornos científicos y de ingeniería; y por un subconjunto más pequeño de usuarios domésticos avanzados. [4]

1.2.4 Protocolos de comunicación.

Protocolo de red o también Protocolo de Comunicación es el conjunto de reglas que especifican el intercambio de datos u órdenes durante la comunicación entre las entidades que forman parte de una red. [5]

Al hablar de protocolos no se puede generalizar, debido a la gran amplitud de campos que cubren, tanto en propósito, como en especificidad. No obstante, la mayoría de los protocolos especifican una o más de las siguientes propiedades: [5]

- Detección de la conexión física sobre la que se realiza la conexión (cableada o sin cables)
- Pasos necesarios para comenzar a comunicarse (Handshaking)
- Negociación de las características de la conexión.
- Cómo se inicia y cómo termina un mensaje.
- Formato de los mensajes.
- Qué hacer con los mensajes erróneos o corruptos (corrección de errores)
- Cómo detectar la pérdida inesperada de la conexión, y qué hacer en ese caso.
- Terminación de la sesión de conexión.
- Estrategias para asegurar la seguridad (autenticación, cifrado).

Algunos de los más comúnmente utilizados para la gestión de redes son SNMP, Telnet y técnicas de monitoreo SSH.

Características de los protocolos de comunicación SNMP. Telnet y SSH.

Protocolo Simple de Administración de Red (SNMP).

Permite a los administradores gestionar dispositivos y diagnosticar problemas en la red. Este se basa en dos elementos principales: un supervisor y agentes. El supervisor es el terminal que le permite al administrador de red realizar solicitudes de administración.

La arquitectura de administración de la red propuesta por el protocolo SNMP se basa en tres elementos principales: [6]

- Los dispositivos administrados son los elementos de red (puentes, concentradores, router o servidores) que contienen "objetos administrados" que pueden ser información de hardware, elementos de configuración o información estadística.

- Los agentes, es decir, una aplicación de administración de red que se encuentra en un periférico y que es responsable de la transmisión de datos de administración local desde el periférico en formato SNMP.

- El sistema de administración de red (NMS), esto es, un terminal a través del cual los administradores pueden llevar a cabo tareas de administración.

Telecommunication Network (Telnet)

Es uno de los protocolos más antiguos de Internet, data de la época de ARPANET, y se utiliza para conectar con un equipo remoto a través de la Red, de forma que el ordenador cliente se comporta como una terminal conectada (on-line) con el ordenador remoto.[7]

El protocolo Telnet es un protocolo de Internet estándar que permite conectar terminales y aplicaciones en Internet. El protocolo proporciona reglas básicas que permiten vincular a un cliente (sistema compuesto de una pantalla y un

teclado) con un intérprete de comandos (del lado del servidor). El protocolo Telnet se aplica en una conexión TCP para enviar datos en formato ASCII codificados en 8 bits, entre los cuales se encuentran secuencias de verificación Telnet. Por lo tanto, brinda un sistema de comunicación orientado bidireccional codificado en 8 bits y fácil de implementar. [7]

Interprete de órdenes seguro (SSH).

SSH (Secure SHell, en español: intérprete de órdenes segura) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si tenemos un Servidor X (en sistemas Unix y Windows) corriendo. [8]

SSH trabaja de forma similar a como se hace con telnet. La diferencia principal es que SSH usa técnicas de cifrado que hacen que la información que viaja por el medio de comunicación vaya de manera no legible y ninguna tercera persona pueda descubrir el usuario y contraseña de la conexión ni lo que se escribe durante toda la sesión; aunque es posible atacar este tipo de sistemas por medio de ataques de REPLAY y manipular así la información entre destinos. [8]

De los protocolos antes mencionados el SNMP no cumple las condiciones de automatización de la gestión del Switch, y con respecto al protocolo SSH, este no se encuentra en todos los Switch con los que ETECSA cuenta.

Por las razones anteriores se seleccionó el telnet para establecer la comunicación con el Switch, ya que este está presente en todos los dispositivos de la empresa.

1.3 Descripción de los sistemas existentes.

Redes Definidas por Software (SDN)

SDN es una arquitectura emergente que es dinámica, manejable, rentable y adaptable, lo que lo hace ideal para el alto ancho de banda, la naturaleza

dinámica de las aplicaciones actuales. Esta arquitectura desacopla las funciones de control de red y de reenvío permitiendo que el control de red sea directamente programable y la infraestructura subyacente sea abstraída para aplicaciones y servicios de red. El protocolo OpenFlow es un elemento fundamental para la construcción de soluciones SDN. La arquitectura SDN es: [9]

- **Directamente programable**: El control de red es directamente programable porque está desacoplado de las funciones de reenvío.
- **Ágil**: Resumen de control de reenvío permite a los administradores ajustar dinámicamente el flujo de tráfico de toda la red para satisfacer las necesidades cambiantes.
- **Gestionado de forma centralizada**: la inteligencia de red está (lógicamente) centralizada en controladores SDN basados en software que mantienen una vista global de la red, que aparece a las aplicaciones y los motores de políticas como un único conmutador lógico.
- **Programáticamente configurado**: SDN permite a los administradores de red configurar, gestionar, proteger y optimizar los recursos de red muy rápidamente a través de programas SDN dinámicos y automatizados, que pueden escribir ellos mismos porque los programas no dependen de software propietario.
- **Estándares abiertos basados en proveedores y neutrales**: cuando se implementa a través de estándares abiertos, SDN simplifica el diseño y operación de la red porque las instrucciones son proporcionadas por controladores SDN en lugar de múltiples dispositivos y protocolos específicos del proveedor.

Para poder usar en una red de computadores la metodología SDN es necesario que los dispositivos donde se implementan los firewall soporten el protocolo OpenFlow. Los switches de la empresa ETECSA no cuentan con dicho protocolo, por lo tanto no se puede aplicar esta metodología.

FirewallD.

FirewallD proporciona un firewall gestionado dinámicamente con soporte para zonas de red / firewall para definir el nivel de confianza de las conexiones de red o interfaces. Tiene soporte para IPv4, configuración de firewall IPv6 y para puentes Ethernet y tiene una separación de tiempo de ejecución y opciones de configuración permanente. También admite una interfaz para servicios o aplicaciones para agregar reglas de firewall directamente. [10]

El modelo de firewall anterior con system-config-firewall / lokkit era estático y cada cambio requería un reinicio completo del cortafuegos. Esto incluyó también descargar los módulos del kernel del netfilter del cortafuego y cargar los módulos que son necesarios para la nueva configuración. La descarga de los módulos estaba rompiendo firewalling con estado y conexiones establecidas. [10]

El proyecto de FirewallD está desarrollado sobre iptables, este utiliza una interfaz de líneas de comandos más intuitiva a la hora de configurar el firewall de iptables. Como dicho proyecto es para un solo tipo de firewall y no se puede reutilizar en este trabajo

1.4 Tendencias, metodologías y/o tecnologías utilizadas.

Proceso Unificado de Desarrollo de Software (RUP)

Es una de las más tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodología es más eficaz y necesaria en cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida. Esta divide el desarrollo en 4 fases que definen su ciclo de vida: [11]

- **Inicio:** El objetivo es determinar la visión del proyecto y definir lo que se desea realizar.
- **Elaboración:** Etapa en la que se determina la arquitectura óptima del proyecto.
- **Construcción:** Se obtiene la capacidad operacional inicial.
- **Transmisión:** Obtener el producto acabado y definido.

Lenguaje de Modelado Unificado (UML)

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.[12]

Visual Paradigm

Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades. [13]

Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos. [13]

Text Mate 2(Sublime).

Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. [14]

Sublime Text utiliza un paquete de herramientas de interfaz de usuario personalizado, optimizado para la velocidad y la belleza, aprovechando la funcionalidad nativa de cada plataforma. Cuenta con una potente API de

complementos basada en Python. Junto con la API, cuenta con una consola integrada en Python para experimentar interactivamente en tiempo real. [14]

Python.

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. [15]

Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos. [15]

Su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo, muy rápido y, lo que es más importante, divertido. [15]

Por petición del cliente y para futuras actualizaciones del software se escogió el lenguaje Python para programar los módulos.

ENSP.

ENSP (Plataforma de Simulación de Redes Empresariales) es una plataforma de herramientas de simulación de red gráfica gratuita y extensible proporcionada por Huawei. Principalmente hacer la simulación de hardware para los enrutadores de red de la empresa, interruptores y perfecta presentar una escena de dispositivo real, soporte de simulación de red a gran escala, también nos hacen hacer pruebas experimentales y aprender la tecnología de red en el caso de que no hay dispositivo real.[16]

1.5 Conclusiones

En este capítulo se introdujo a los Switch, los firewall, a las Interfaces de Línea de Comandos y los protocolos más usados para la comunicación. Se definió el concepto de firewall.

Después de haber hecho un análisis de las tecnologías y tendencias, se seleccionó la metodología RUP como guía para la documentación de la

aplicación propuesta. Se seleccionó el protocolo telnet para la comunicación con el switch y como lenguaje de programación se escogió Python para ejecutar los módulos.

2 Construcción de la Solución Propuesta.

2.1 Introducción.

Este capítulo está dedicado al proceso de desarrollo del dominio utilizando los flujos de trabajo de la metodología RUP. Se realiza la modelación de negocio para tener una mejor perspectiva de las funcionalidades de la aplicación. Se identifican las reglas del negocio, los requisitos funcionales y no funcionales.

2.2 Definición de las entidades y conceptos principales.

En el modelo de dominio se definen las siguientes clases: Firewall, Puerto, Zona, ACL, Regla.

Firewall: Es un dispositivo de hardware que mediante una política bloquea el acceso no autorizado de paquetes que viajan un una red de ordenadores.

Puerto: interfaces Gigabyte-Ethernet que tienen tanto los enrutadores como los conmutadores.

Zona: Subredes conectadas a un router.

ACL: Es una lista de reglas ordenadas que permiten controlar el flujo del tráfico en equipos de redes, tales como enrutadores y conmutadores.

Regla: Es un elemento de control del tráfico en la red, cuenta con clasificador de tráfico (protocolo) y una acción.

2.2.1 Reglas del negocio a considerar.

- Los puertos pueden pertenecer solamente a una interfaz.
- Las interfaces pueden tener asociadas solo dos ACL.
- Las ACL que se asignen a las interfaces solo pueden estar aplicada una en entrada y la otra en salida.
- Las ACL tienen que deshabilitarse para luego ser modificadas.
- Las ACL aplicadas en outbound no pueden no soportan rango de puertos tanto en la fuente como en el destino.

2.2.2 Representación del modelo del dominio.

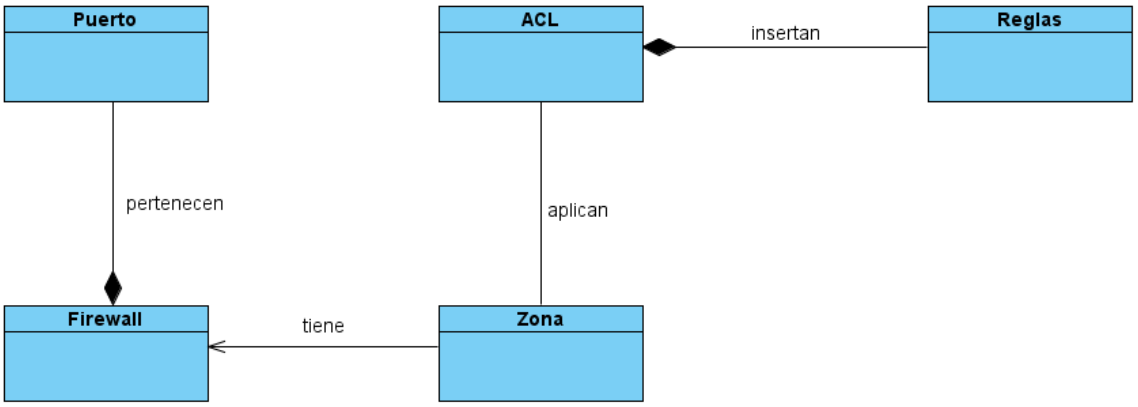


Figura 1. Diagramas de clases del modelo de objetos del dominio.

2.3 Requisitos.

2.3.1 Concepción general del sistema.

El presente proyecto forma parte del proyecto en desarrollo InterFirewall, este está dividido en tres módulos principales, uno es una interfaz web, el segundo es un servicio web y el tercero una aplicación de línea de comandos.

Este proyecto tiene como objetivo implementar el tercer módulo. Su arquitectura cuenta con dos módulos, uno de Interfaz de Línea de Comandos para manejar cortafuegos de distintas tecnologías y el otro, un driver para la configuración del Switch capa 3 de fabricante Huawei, modelo S5328. Esta arquitectura permite que la Interfaz de Línea Comando pueda ser usada por un servicio web o por un administrador de redes pasando así la configuración hacia el driver y este interactuar con el Switch para configurar el cortafuego. También cuenta con una estructura en la que se permite añadir otros drivers para otro tipo de switch.

Teniendo en cuenta los objetivos de los futuros usuarios del dominio y la descripción de cómo debe funcionar el mismo, se pueden inferir los siguientes requerimientos:

2.3.2 Requerimientos funcionales.

R1. Autenticarse.

R2. Interactuar con la interfaz de comando.

R3. Crear ACL.

R4. Deshabilitar ACL.

R5. Borrar ACL.

R6. Mostrar ACL.

R7. Aplicar regla.

R8. Eliminar una regla.

R9. Crear zona.

R10. Asignar puerto a una zona.

R11.Asignar zona a un puerto en modo trunk.

R12. Asignar IP a zona.

R13. Eliminar zona.

R14. Mostrar zona.

R15. Limitar zona en entrada.

R16. Limitar zona en salida.

R17.Delimitar zona en entrada.

R18.Delimitar zona en salida.

R19.Mostrar el estado actual del firewall.

2.3.3 Requerimientos no funcionales.

Los requisitos no funcionales especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad, y fiabilidad. [17 p.110]

Requerimientos de Usabilidad:

La aplicación será utilizada por los administradores y por un servicio web para configurar el tráfico de la red.

Facilidad de uso para las personas que dominen el idioma inglés.

Requerimientos de Rendimiento:

La aplicación deberá responder en el mínimo de tiempo posible ante las solicitudes de información por parte de los usuarios y otros sistemas, además debe permitir el acceso simultáneo a los datos por diferentes usuarios.

Requerimientos de Soporte:

El módulo se documentará con un manual de ayuda con el objetivo de explicar su uso y garantizar el soporte del mismo. Se debe realizar el proyecto de forma versionable, permitiendo mantenimiento al módulo a fin de aumentar las funcionalidades y/o corregir los errores del mismo a través de versiones posteriores.

Requerimientos de Portabilidad:

El módulo propuesto fue desarrollado en la plataforma Linux, pero puede ser ejecutada desde otras plataformas como Windows, que soporte el módulo Pexpect del lenguaje de alto nivel Python.

Requerimientos de Seguridad:

La aplicación cuenta con una política de identificación de usuario con su correspondiente contraseña.

Requerimientos Políticos y Culturales:

El desarrollo de la aplicación debe estar en correspondencia con la cultura organizacional de la empresa.

Requerimientos Legales:

La aplicación propuesta debe cumplir con las regulaciones y normas indicadas oficialmente en la División Territorial de ETECSA en Cienfuegos.

Requerimientos de Ayuda:

La aplicación cuenta con una ayuda disponible sobre las funcionalidades de la misma y su modo de uso.

Requerimientos de Software:

La aplicación necesita el lenguaje de alto nivel Python v2.7 o Python v3 o superior.

Requerimientos de Hardware:

Las terminales solo requerirán estar conectadas a la red y cumplir con los requisitos mínimos de hardware del sistema operativo que se emplee.

2.3.4 Modelo de caso de uso de sistema.

El modelo de casos de uso permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema**.**

2.3.3.1 Actores del sistema.

Actor	**Descripción**
administrador	Es quien utiliza la aplicación y se beneficia de las funcionalidades de la interfaz de línea de comandos.

Tabla 2. Descripción de actores del sistema.

2.3.3.2 Diagramas de caso de uso del sistema.

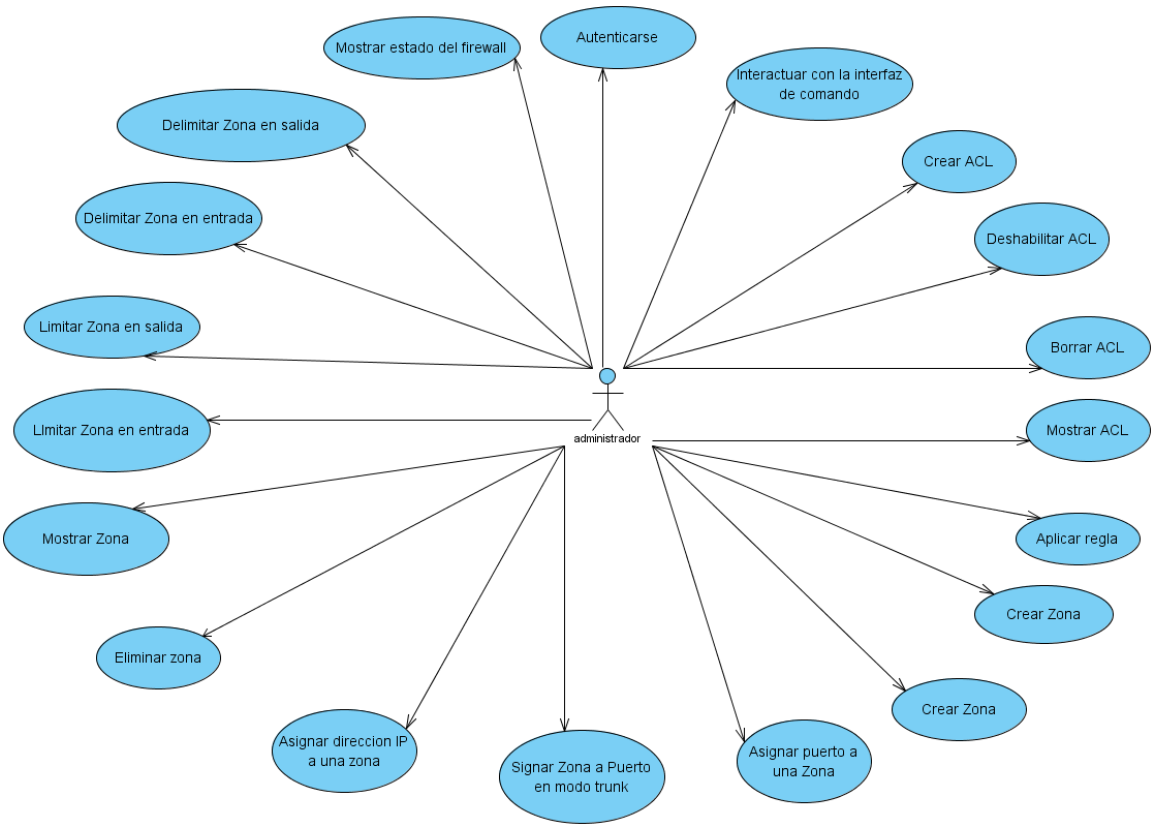


Figura 2. Diagrama de caso de uso del Sistema.

2.3.3.3 Descripción de los casos de uso del sistema.

Nombre del caso de uso	Autenticarse
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando se quiere ejecutar el módulo de interfaz de comando. Para esto se envía el IP del firewall, el usuario y la contraseña, si estos son correctos se le permite acceder a las funcionalidades, en caso contrario no se permite acceder culminando así el caso de uso.
Referencias	R1

Precondiciones	En el firewall debe existir un mecanismo de control de acceso para lograr la comunicación con el mismo.
Post-condiciones	El administrador debe acceder a las funcionalidades del módulo de interfaz de comando.

Tabla 3. Descripción del caso de uso: Autenticarse.

Nombre del caso de uso	Interactuar con la interfaz de comando.
Actores	Administrador.
Resumen	El caso de uso inicia cuando se desea acceder a alguna de las funcionalidades del módulo de interfaz de comandos. Para esto el actor ejecuta acciones en forma de comandos pasando parámetros en cada uno de estos; culminando así el caso de uso.
Referencias	R2
Precondiciones	El módulo debe tener permiso de ejecución.
Post-condiciones	

Tabla 4. Descripción del caso de uso: Interactuar con la interfaz de comando.

Nombre del caso de uso	Crear ACL
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando se quiere crear una ACL en el firewall, para esto se interactúa con el módulo de interfaz introduciendo el nombre de la misma.
Referencias	R3
Precondiciones	No puede existir la ACL que se quiere crear.
Post-condiciones	

Tabla 5. Descripción del caso de uso: Crear una ACL.

Nombre del caso de uso	Deshabilitar ACL
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando se quiere deshabilitar una ACL para esto el usuario ingresa el nombre de la zona y en la dirección a la cual esta aplicada la ACL.
Referencias	R4
Precondiciones	La ACL a deshabilitar debe existir y estar aplicada a una zona.
Post-condiciones	

Tabla 6. Descripción del caso de uso: Deshabilitar ACL.

Nombre del caso de uso	Borrar ACL
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando se quiere borrar una ACL del firewall para esto se debe especificar el nombre de una ACL existente.
Referencias	R5
Precondiciones	Debe existir la ACL y se desactiva antes de borrarla.
Post-condiciones	

Tabla 7. Descripción del caso de uso: Borrar ACL.

Nombre del caso de uso	Mostrar ACL
Actores	Administrador(inicia)

Resumen	El caso de uso inicia cuando se quiere mostrar las ACL existentes en el firewall.
Referencias	R6
Precondiciones	Debe existir la ACL de la cual se mostrara la información.
Post-condiciones	

Tabla 8. Descripción del caso de uso: Mostrar ACL.

Nombre del caso de uso	Aplicar regla
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando se quiere aplicar una regla a una ACL, introduciendo el nombre de la ACL y la regla nueva.
Referencias	R7
Precondiciones	En el firewall debe existir la ACL a la cual se quiere aplicar la regla y la ACL tiene que deshabilitarse antes de ser modificada.
Post-condiciones	Se tiene que aplicar la ACL a la zona que pertenecía.

Tabla 9. Descripción del caso de uso: Aplicar regla.

Nombre del caso de uso	Eliminar regla
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando se quiere eliminar una regla entrando por parámetros el identificador de la ACL y el identificador de la regla que se quiera eliminar.

Referencias	R8
Precondiciones	En el firewall tiene que existir la ACL y la regla tiene que pertenecer a dicha ACL, se debe deshabilitar la ACL para poder ser modificada.
Post-condiciones	Se tiene que aplicar la ACL a la zona que pertenecía.

Tabla 10. Descripción del caso de uso: Eliminar regla.

Nombre del caso de uso	Crear Zona
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando el administrador quiere crear una nueva zona
Referencias	R9
Precondiciones	El identificador de la zona no debe coincidir con otro, asegurándonos de que no sobrescriba el contenido de una zona existente.
Post-condiciones	

Tabla 11. Descripción del caso de uso: Crear Zona.

Nombre del caso de uso	Asignar puerto a una zona
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando el administrador quiere asignarle un puerto a una zona
Referencias	R10
Precondiciones	El puerto tiene que estar en modo de acceso para

	realizar esta acción.
Post-condiciones	

Tabla 12. Descripción del caso de uso: Asignarle puerto a Zona.

Nombre del caso de uso	Asignar Zona a un puerto en modo trunk
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando el administrador quiere asignarle una o varias zonas a un puerto en modo trunk.
Referencias	R11
Precondiciones	El puerto tiene que estar en modo trunk.
Post-condiciones	

Tabla 13. Descripción del caso de uso: Asignar Zona a un puerto en modo trunk.

Nombre del caso de uso	Asignar IP a una zona
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando el administrador quiere asignar una o varias direcciones IP a una zona.
Referencias	R12
Precondiciones	La zona debe estar creada. La o las direcciones IP no pueden coincidir ni tener conflictos con otras ya asignadas.
Post-condiciones	

Tabla 14. Descripción del caso de uso: Asignar IP a una Zona.

Nombre del caso de uso	Eliminar Zona
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando el administrador quiere eliminar un nueva zona
Referencias	R13
Precondiciones	La zona tiene que existir para poder ser eliminada
Post-condiciones	

Tabla 15. Descripción del caso de uso: Eliminar Zona.

Nombre del caso de uso	Mostrar Zona
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando el administrador quiere ver la información correspondiente a una determinada zona
Referencias	R14
Precondiciones	La zona debe estar creada para mostrar su contenido
Post-condiciones	

Tabla 16. Descripción del caso de uso: Mostrar Zona.

Nombre del caso de uso	Limitar Zona en entrada
Actores	Administrador(inicia)

Resumen	El caso de uso inicia cuando el administrador quiere limitar una zona en entrada
Referencias	R15
Precondiciones	La zona debe estar creada y no pude tener ninguna política aplicada en entrada.
Post-condiciones	

Tabla 17. Descripción del caso de uso: Limitar Zona en entrada.

Nombre del caso de uso	Limitar Zona en salida
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando el administrador quiere limitar una zona en salida
Referencias	R16
Precondiciones	La zona debe estar creada y no pude tener ninguna política aplicada en salida.
Post-condiciones	

Tabla 18. Descripción del caso de uso: Limitar Zona en salida.

Nombre del caso de uso	Delimitar Zona en entrada
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando el administrador quiere quitar una política de una determinada zona en entrada
Referencias	R17

Precondiciones	La zona debe estar creada y tener una política aplicada en entrada.
Post-condiciones	

Tabla 19. Descripción del caso de uso: Delimitar Zona en entrada.

Nombre del caso de uso	Delimitar Zona en salida
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando el administrador quiere quitar una política de una determinada zona en salida
Referencias	R18
Precondiciones	La zona debe estar creada y tener una política aplicada en salida.
Post-condiciones	

Tabla 20. Descripción del caso de uso: Delimitar Zona en salida.

Nombre del caso de uso	Mostrar estado actual del firewall
Actores	Administrador(inicia)
Resumen	El caso de uso inicia cuando el administrador quiere saber la configuración actual del firewall
Referencias	R19

Tabla 21. Descripción del caso de uso: Mostrar estado actual del firewall.

2.4 Diseño.

2.4.1 Diagrama de clases del diseño.

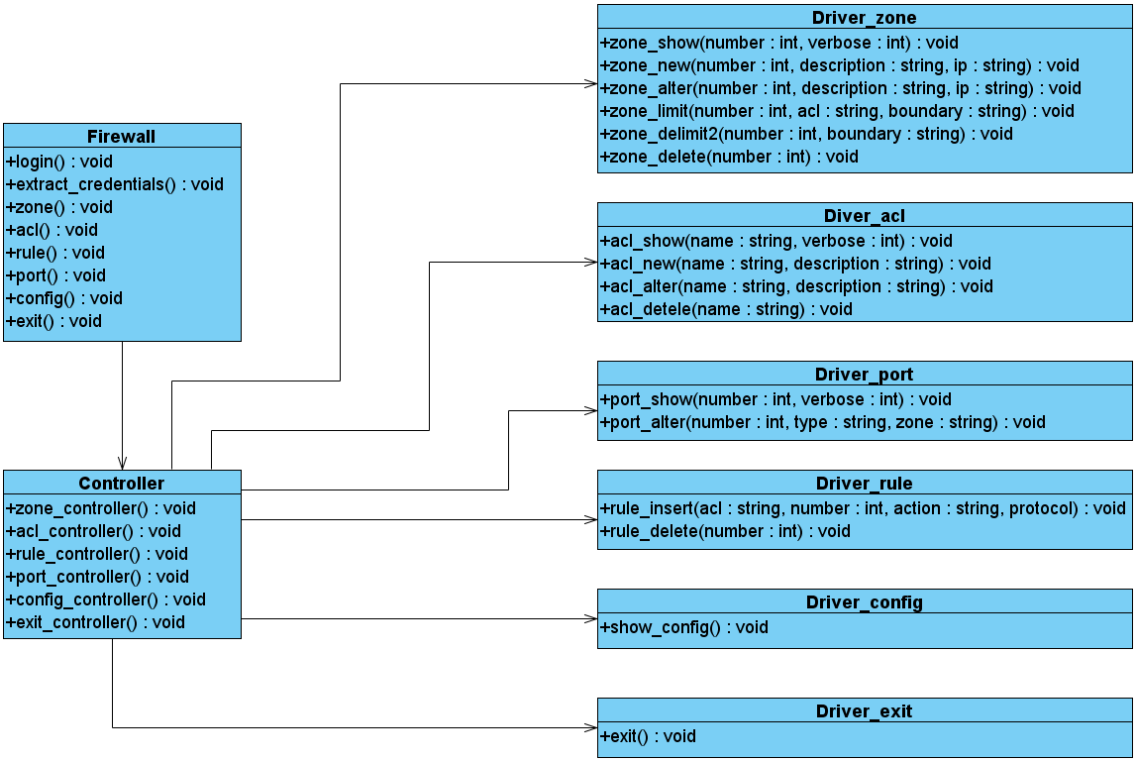


Figura 3. Diagrama de clases.

2.4.2 Diagrama de implementación.

El lenguaje utilizado para especificar una clase del diseño es lo mismo que el lenguaje de programación. Consecuentemente, las operaciones, parámetros, atributos, tipos y demás son especificados utilizando la sintaxis del lenguaje de programación elegido. [17 p.209]

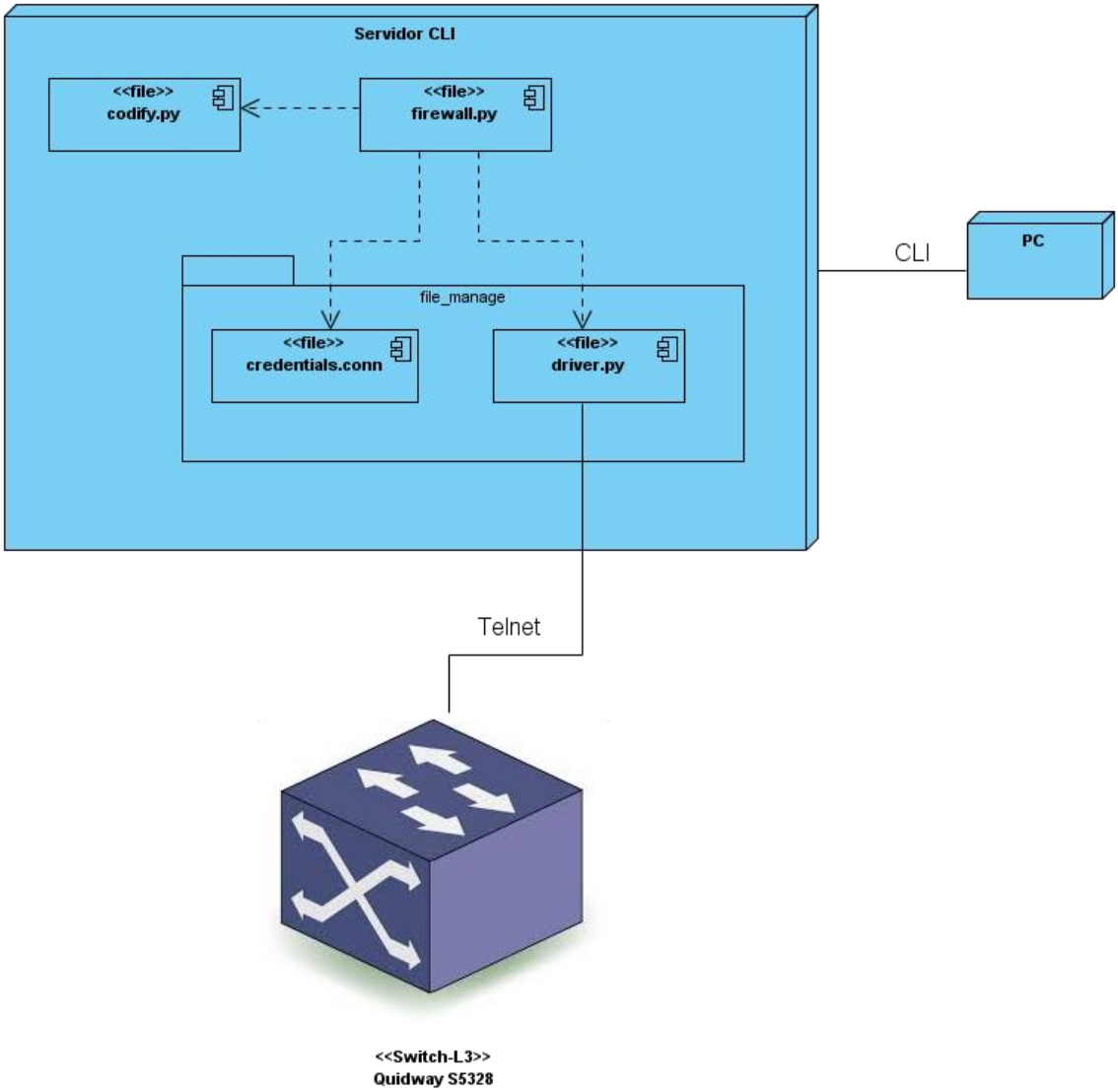


Figura 4. Diagrama de implementación.

2.4.3 Principios de diseño.

Los principios del diseño de este proyecto están orientados a facilitar el uso de la aplicación con eficiencia y mayor rapidez. Se describen a continuación.

2.4.3.1 Estándares en la interfaz de la aplicación

El módulo principal de la aplicación implementa una interfaz de línea de comandos interactiva. Cuenta con varios comandos y a su vez con varios subcomandos y a cada subcomando le corresponde una serie de argumentos. El valor de estos argumentos se teclea en forma de parámetros y se recibe una salida para cada parámetro específico.

2.4.3.2 Tratamiento de errores

Cuando el cliente ejecute una funcionalidad se mostrará una información en pantalla de acuerdo al resultado de la ejecución. Esta información varía de acuerdo al tipo de error o éxito de la petición garantizando el adecuado funcionamiento de la aplicación.

2.4.3.3 Concepción General de la ayuda

Los módulos implementados cuentan con una descripción de las funcionalidades. Cada comando del CLI cuenta con una ayuda para guiar a los usuarios. Además la aplicación cuenta con un manual de ayuda para los usuarios que se inicien con el manejo de líneas de comandos.

2.5 Conclusiones

En el presente capítulo se ha descrito la solución propuesta a través de la representación gráfica de los diagramas de modelo del dominio, de caso de unos del sistema, diagrama de clases del diseño y diagrama de implementación. Se describieron los principios de diseño del sistema, específicamente, los temas de estándares de la interfaz, concepción del tratamiento de errores y sistema de ayuda.

3. Validación y Estudio de Factibilidad.

3.1 Introducción

En este capítulo se realiza el estudio de factibilidad empleando el análisis de planificación por casos de uso, la validación de la aplicación mediante escenarios de pruebas y pruebas funcionales.

3.2 Validación de la solución propuesta

Para la validación de la aplicación se crearon un conjunto de escenarios poniendo a prueba las funcionalidades del sistema.

3.2.1 Escenario de prueba #1

En el departamento de informática de la empresa ETECSA se introdujo una PC (Computadora Personal)en la red, no perteneciente a ningún trabajador de la empresa. Mediante un Sistema de Detección de Intrusos (IDS) se identificó la dirección IP de la PC y a la subred donde estaba conectado, en este caso la zona Usuarios.

Teniendo ubicado al intruso en la red, se dio paso a configurar el firewall mediante la aplicación, modificando la política aplicada en entrada, a la zona Usuarios, insertándole una regla para bloquear el acceso de dicha PC(ver anexo 1). Luego de hacer las configuraciones anteriores se comprobó que la PC insertada no tenía acceso a la zona de Usuarios.

3.2.2 Escenario de prueba #2

La empresa ETECSA cuenta una infraestructura de red dividida en varias zona, en las cuales se limita el tráfico entre ellas, para una mejor seguridad de la red. Entre esas zonas se encuentra la zona Servidores, la zona Usuarios y la zona Administradores, se quiere establecer una conjunto de reglas para que solo tengan acceso los administradores a los servidores, los cuales serán los únicos autorizados a configurar dichos servidores mediante pruebas de monitorización (ver anexo 2).

Para esto se configuró la política aplicada en entrada, a la zona Servidores, insertándole dos reglas. La primera bloqueando la entrada de usuarios y la

segunda permitiendo el acceso de los administradores el manejo de los servidores de manera remota.

3.3 Pruebas funcionales del sistema.

Se denominan pruebas funcionales, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales, esta etapa suele ser la última etapa de pruebas y al dar conformidad sobre esta el paso siguiente es el pase a producción. [18]

3.3.1 Validación de pruebas funcionales del caso de uso Autenticarse

Caso de uso	Autentificarse
Resumen	EL usuario debe ingresar una serie de parámetros posicionales o sea obligatorios en su orden de inserción.
Sección: Comandos <ÍP> <user> <password> <manufacturer> <model>	
Validaciones: La validación se hace cuando el usuario presiona enter al interactuar con la interfaz de línea de comandos: 1- faltan argumentos por entrar 1- la IP es incorrecta 2- la elección del fabricante no es válida 3- la elección del modelo no es válida **(ver las 4 validaciones anteriores en anexo 3)** 4- no se pudo establecer la conexión por telnet **(ver anexo 4)** 5- error de autenticación **(ver anexo 5)**	

Tabla 22. Validación de pruebas funcionales del caso de uso Autenticarse

3.3.2 Validación de pruebas funcionales de los casos de uso relacionados con zona

<table>
<tr><td>Casos de uso</td><td>-mostrar zona
-crear zona
-modificar zona
-limitar zona
-delimitar zona
-borrar zona</td></tr>
<tr><td>Resumen</td><td>Está involucrado todo lo referente a cualquier subcomando del comando zona con sus respectivos argumentos.</td></tr>
<tr><td colspan="2">Sección de Comandos</td></tr>
<tr><td colspan="2">show -n NUMBER
new <number> [–d DESCRIPTION] [–ip IP]
alter <number> [–d DESCRIPTION] [–ip IP]
limit <zone><acl><boundary>
delimit <zone> <bound>
delete <number></td></tr>
<tr><td colspan="2">Validaciones:
La validación se hace cuando el usuario presiona enter al interactuar con la interfaz de línea de comandos:
1- el número de la zona no es correcto
2- la zona no existe()
(ver las dos validaciones anteriores en anexo 6)
3- la zona ya existe</td></tr>
</table>

4- la descripción es muy larga 5- el IP es incorrecto **(ver las tres validaciones anteriores en el anexo 7)** 6- la ACL no existe 7- la direcciones para limitar la zona son incorrectas **(ver las dos validaciones anteriores en el anexo 8)** 8- la zona ya está limitada en esa dirección 9- faltan argumentos por entrar **(ver las dos validaciones anteriores en el anexo 9)** 10-la zona no está limitada para poder ser delimitada 11-la zona que se quiere borrar no existe **(ver las dos validaciones anteriores en el anexo 10)**

Tabla 23. Validación de pruebas funcionales de los casos de uso relacionados con zona

3.3.3 Validación de pruebas funcionales de los casos de uso relacionados con ACL

Casos de uso	-mostrar ACL -crear ACL -modificar ACL -borrar ACL
Resumen	Está involucrado todo lo referente a cualquier subcomando del comando ACL con sus respectivos argumentos.
Sección de Comandos	

show -n NAME new <name> [–d DESCRIPTION] alter <name> [–d DESCRIPTION] delete <name>
Validaciones: La validación se hace cuando el usuario presiona enter al interactuar con la interfaz de línea de comandos: 1- el tamaño del nombre es muy largo 2- faltan argumentos **(ver las validaciones anteriores en el anexo 11)** 3- la ACL no existe 4- la ACL ya existe **(ver las validaciones anteriores en el anexo 12)** 5- el tamaño de la descripción es muy largo 6- el IP es incorrecto **(ver las validaciones anteriores en el anexo 13)**

Tabla 24. Validación de pruebas funcionales de los casos de uso relacionados con ACL

3.3.4 Validación de pruebas funcionales de los casos de uso relacionados con puerto

Casos de uso	-mostrar puerto -modificar puerto
Resumen	Está involucrado todo lo referente a cualquier subcomando del comando puerto con sus respectivos

	argumentos.
Sección de Comandos	
show [-n NUMBER] alter <number> <type> <zone>	
Validaciones: La validación se hace cuando el usuario presiona enter al interactuar con la interfaz de línea de comandos: 1- el número del puerto es incorrecto 2- faltan argumentos por entrar 3- el modelo Quidway S5328 solo tiene habilitado 24 puertos 4- el tipo de puerto es incorrecto 5- si el puerto en modo Access solo puede vincularse una zona **(ver las validaciones anteriores en el anexo 14)**	

Tabla 25. Validación de pruebas funcionales de los casos de uso relacionados con puerto

3.3.5 Validación de pruebas funcionales de los casos de uso relacionados con regla

Casos de uso	-insertar regla -eliminar regla
Resumen	Está involucrado todo lo referente a cualquier subcomando del comando regla con sus respectivos argumentos.
Sección de Comandos	
insert <ACL><id><action><protocol>[-s SOURCE] [-s_port SOURCE_PORT][-	

d DESTINY][-d_port DESTINY_PORT] [-log] [-desc DESCRIPTION] delete <ACL> <id>
Validaciones: La validación se hace cuando el usuario presiona enter al interactuar con la interfaz de línea de comandos: 1- faltan argumentos 2- la ACL no existe 3- el id es incorrecto 4- las tipo de acción es incorrecta 5- el tipo de protocolo es incorrecto **(ver validaciones anteriores en el anexo 15)** 6- la dirección IP no es valida 7- la subred es incorrecta 8- el primer IP tiene que ser menor que el segundo 9- el rango no puede encerrar más de 255 IP 10-la ACL no tiene insertada la regla que pretendes borrar **(ver validaciones anteriores en el anexo 16)**

Tabla 26. Validación de pruebas funcionales de los casos de uso relacionados con regla

3.4 Estimación.

Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.[19]

3.4.1 Cálculo de Puntos de Casos de Uso sin Ajustar

UUCP: Puntos de Casos de Uso sin Ajustar

UAW: Factor de Peso de los Actores sin Ajustar

UUCW: Factor de Peso de los Casos de Uso sin Ajustar

UUCP = UAW + UUCW

Cálculo de UAW:

ACTOR	**Tipo de Actor**	**Factor de peso**
administrador	complejo	3

Tabla 27. Clasificación de los Actores del sistema

UAW = Cantidad de actores * Factor de peso

UAW = 1 * 3

UAW = 3

Cálculo del UUCW:

Caso de Uso	**Clasificación**	**Peso**
Autenticarse	Simple	5
Interactuar con la interfaz de comandos	Simple	5
Crear ACL	Simple	5
Deshabilitar ACL	Simple	5
Borrar ACL	Simple	5
Mostrar ACL	Simple	5
Aplicar Regla	Medio	10
Eliminar regla	Medio	10
Crear zona	Simple	5
Asignar puerto a zona	Simple	5
Asignar zona a puerto	Simple	5

en modo trunk		
Asignar IP a zona	Medio	
Eliminar zona	Simple	5
Mostrar zona	Simple	5
Limitar zona en entrada	Simple	10
Limitar zona en salida	Simple	10
Delimitar zona en entrada	Simple	10
Delimitar zona en salida	Simple	10
Mostrar estado actual del firewall	Simple	5

Tabla 28. Factor de peso de los casos de uso sin ajustar

En la tabla de clasificación anterior se muestra que el sistema está compuesto por 19 Casos de Uso, de ellos hay 3 con clasificación media y 16 con clasificación simple. Por lo que el factor de Peso de los Casos de uso sin ajustar es igual a:

UUCW = 16 *5 + 3 * 10

UUCW = 80 + 30

UUCW = 110

Al sustituir en la fórmula nos queda el siguiente resultado:

UUCP = 3 + 110

UUCP = 103

3.4.2 Cálculo de Casos de uso ajustados

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin Ajustar.

TCF: Factor de Complejidad Técnica.

EF: Factor de Ambiente

Obtenidos los Puntos de Casos de Uso sin ajustar, se debe ajustar este valor mediante la siguiente ecuación:

UCP = UUCP * TCF *EF

Cálculo de TCF:

Factor	**Descripción**	**Peso**	**Valor**	**Peso * Valor**
T1	Sistema distribuido.	2	0	0
T2	Objetivos de performance o tiempo de respuesta.	1	3	3
T3	Eficiencia del usuario final.	1	4	4
T4	Procesamiento interno complejo.	1	4	4
T5	EL código debe ser reutilizable	1	5	5
T6	Facilidad de instalación.	0.5	4	2
T7	Facilidad de uso.	0.5	4	2
T8	Portabilidad.	2	5	10
T9	Facilidad de cambio.	1	4	4
T10	Concurrencia.	1	3	3
T11	Incluye objetivos especiales de seguridad.	1	2	2
T12	Provee acceso directo a terceras partes.	1	0	0
T13	Se requiere facilidades especiales de entrenamiento a usuarios.	1	2	2

Tabla 29. Factor de complejidad Técnica

Por lo que el Factor de Complejidad Técnica resulta:

TCF = 0.6 + 0.01 * **Σ** (Peso (i) * Valor asignado)

TCF = 0.6 + 0.01 * (0 + 3+ 4 +4 + 5 + 2 + 2 + 10 + 4 + 3 + 2 + 0 + 2)

TCF = 0.6 + 0.01 * 41

TCF = 1.01

Cálculo de EF:

Factor	Descripción	Peso	Valor	Peso * Valor
E1	Familiaridad con el proyecto utilizado.	1.5	4	6
E2	Experiencia en la aplicación.	0.5	4	2
E3	Experiencia en orientación a objetos.	1	4	4
E4	Capacidad del analista líder.	0.5	5	2.5
E5	Motivación.	1	5	5
E6	Estabilidad de los requerimientos.	2	5	10
E7	Personal part-time.	-1	0	0
E8	Dificultad del lenguaje de programación.	-1	1	-1

Tabla 30. Factor Ambiente.

Por lo que el Factor de Ambiente resulta:

EF = 1.4 – 0.03 * **Σ** (Peso * Valor asignado)

EF = 1.4 – 0.03 * (6 + 2 + 4 + 2.5 + 5 +10 + 0 + -1)

EF = 1.4 – 0.03 * 28.5

EF = 0.545

Los puntos de Caso de Uso Ajustados resultan:

UCP = UUCP * TCF * EF

UCP = 103 * 1.01 * 0.545

UCP = 56.696

3.4.3 Cálculo del esfuerzo

El esfuerzo en horas/hombre viene dado por:

E = UCP * CF

Donde:

E: Esfuerzo estimado en horas/hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: factor de conversión.

Cálculo de CF:

CF = 20 horas/hombre si EF ≤ 2

CF = 28 horas-hombre (si EF = 3 o EF = 4)

CF = abandonar o cambiar proyecto (si EF ≥ 5)

El Esfuerzo estimado en horas/hombres resulta:

E = 56.696 * 20

E = 1133.92 horas/hombres

3.4.4 Determinación del costo del proyecto

Salario básico: $1200.00

Cantidad de hombres: 1

Duración del proyecto: 1133.92 horas, Trabajando8 horas diarias durante 5 días semanales, el proyecto tendrá una duración de 7 meses y un costo de $8400.

3.5 Beneficios tangibles e intangibles.

Los beneficios obtenidos con la implementación de la aplicación permiten configurar de manera menos compleja y más intuitiva un cortafuego, permitiendo mejorar el control y la seguridad de la red de computadoras de ETECSA en División Territorial de Cienfuegos.

3.6 Análisis de costos y beneficios.

El presente trabajo de diploma no implica costo alguno para ETECSA, sin embargo, toda investigación tiene asociada un costo y su justificación económica viene dada por los beneficios tangibles e intangibles que esta produce. La implementación de la aplicación incluye las operaciones comunes para el manejo de un cortafuego, en función del mejoramiento de la seguridad de la red de dicha empresa mediante la utilización de una interfaz de comandos.

3.7 Conclusiones

Se realizó el estudio de factibilidad basada en puntos de casos de uso de la aplicación donde se estima un tiempo de desarrollo de aproximadamente 7 meses y con un costo por concepto de salario aproximado de 8400 pesos, trabajando 25 días al mes y tomando como salario promedio 1200 pesos. Además se realizaron los casos de pruebas funcionales para verificar el correcto funcionamiento de la aplicación implementada. El desarrollo de la validación del producto mostró resultados favorables a partir de la realización de Escenarios de pruebas y pruebas funcionales.

Conclusiones

- Se analizaron las políticas de control de acceso de distintos firewall permitiendo esto determinar las características comunes entre ellos.
- Se implementó una aplicación de comandos con un nivel mayor de abstracción para manejar cortafuegos de distintas tecnologías y un driver para el switch Quidway S5328 para realizar los cambios en el mismo de manera dinámica.
- Se validó la aplicación mediante pruebas funcionales y escenarios de pruebas permitiendo elevar la calidad de la misma.

Recomendaciones

- Probar al máximo las funcionalidades de la aplicación durante un período amplio de tiempo.
- Generalizar el uso del producto en otras Divisiones Territoriales de la empresa ETECSA.

Referencias bibliográficas.

[1] «El switch: cómo funciona y sus principales características | Redes Telemáticas». [En línea]. Disponible en: http://redestelematicas.com/el-switch-como-funciona-y-sus-principales-caracteristicas/. [Accedido: 24-may-2017].
[2] «Diferencia Entre Switch Capa 2 y Capa 3». [En línea]. Disponible en: https://www.scribd.com/doc/87087243/Diferencia-Entre-Switch-Capa-2-y-Capa-3. [Accedido: 24-may-2017].
[3] «Que es un Firewall y como funciona». [En línea]. Disponible en: http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-un-Firewall-y-como-funciona.php. [Accedido: 26-may-2017].
[4] «Línea de comandos - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/L%C3%ADnea_de_comandos#Implementaciones_de_CLI. [Accedido: 24-may-2017].
[5] «Protocolos de redes - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/Protocolos_de_redes. [Accedido: 24-may-2017].
[6] «Protocolo Simple de Administración de Red (SNMP) - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/SNMP. [Accedido: 24-may-2017].
[7] «Telnet - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/Telnet. [Accedido: 24-may-2017].
[8] «SSH - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/SSH. [Accedido: 24-may-2017].
[9] «Software-Defined Networking (SDN) Definition - Open Networking Foundation». [En línea]. Disponible en: https://www.opennetworking.org/sdn-resources/sdn-definition. [Accedido: 25-may-2017].
[10] «Firewalld - FedoraProject». [En línea]. Disponible en: https://fedoraproject.org/wiki/Firewalld?rd=FirewallD. [Accedido: 25-may-2017].
[11] «Metodologías de desarrollo de software - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/Metodolog%C3%ADas_de_desarrollo_de_software#Filosof.C3.ADa_RUP. [Accedido: 25-may-2017].
[12] «UML - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/UML. [Accedido: 25-may-2017].
[13] «Visual Paradigm - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/Visual_Paradigm. [Accedido: 25-may-2017].
[14] «Sublime text - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/Sublime_text. [Accedido: 25-may-2017].
[15] Raul Gonzales Duque, *Python para todos*. 2008.
[16] «Huawei Enterprise Network Simulation Platform eNSP Global Shock Release - Enterprise Service Support». [En línea]. Disponible en: http://support.huawei.com/enterprise/NewsReadAction.action?newType=0301&contentId=NEWS1000001140. [Accedido: 25-may-2017].
[17] James Rumbaugh Ivar Jacobson Grady Booch, *El Proceso Unificado de Desarrollo de Software*, Pearson Educación S.A. Madrid, 2000.
[18] L. Gonzélez Palacio, *Método para generar Casos de prueba funcional en el desarrollo del softwre*, Revista Ingenierías Universidad de Medellín. 2009.
[19] M. Peralta, *Estimación del Esfuerzo bazada en Casos de Uso*.

Bibliografía

[1] «7.2. re — Regular expression operations — Python 2.7.13 documentation». [En línea]. Disponible en: https://docs.python.org/2/library/re.html. [Accedido: 29-may-2017].

[2] «15.1. os — Miscellaneous operating system interfaces — Python 2.7.13 documentation». [En línea]. Disponible en: https://docs.python.org/2/library/os.html. [Accedido: 29-may-2017].

[3] «16.4. argparse — Parser for command-line options, arguments and sub-commands — Python 3.6.1 documentation». [En línea]. Disponible en: https://docs.python.org/3/library/argparse.html. [Accedido: 29-may-2017].

[4] «Diferencia Entre Switch Capa 2 y Capa 3». [En línea]. Disponible en: https://www.scribd.com/doc/87087243/Diferencia-Entre-Switch-Capa-2-y-Capa-3. [Accedido: 24-may-2017].

[5] James Rumbaugh Ivar Jacobson Grady Booch, *El Proceso Unificado de Desarrollo de Software*, Pearson Educación S.A. Madrid, 2000.

[6] «El switch: cómo funciona y sus principales características | Redes Telemáticas». [En línea]. Disponible en: http://redestelematicas.com/el-switch-como-funciona-y-sus-principales-caracteristicas/. [Accedido: 24-may-2017].

[7] M. Peralta, *Estimación del Esfuerzo bazada en Casos de Uso*. .

[8] «Examples — Pexpect 4.2.1 documentation». [En línea]. Disponible en: http://pexpect.readthedocs.io/en/stable/examples.html. [Accedido: 29-may-2017].

[9] «Firewalld - FedoraProject». [En línea]. Disponible en: https://fedoraproject.org/wiki/Firewalld?rd=FirewallD. [Accedido: 25-may-2017].

[10] «Huawei Enterprise Network Simulation Platform eNSP Global Shock Release - Enterprise Service Support». [En línea]. Disponible en: http://support.huawei.com/enterprise/NewsReadAction.action?newType=0301&contentId=NEWS1000001140. [Accedido: 25-may-2017].

[11] «Inmersión en Python», 2005. [En línea]. Disponible en: http://es.diveintopython.net/toc.html. [Accedido: 25-may-2017].

[12] I. G. Andrés Marsal, *Instroducción a la programación con Python*. 2003.

[13] «Línea de comandos - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/L%C3%ADnea_de_comandos#Implementaciones_de_CLI. [Accedido: 24-may-2017].

[14] «Metodologías de desarrollo de software - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/Metodolog%C3%ADas_de_desarrollo_de_software#Filosof.C3.ADa_RUP. [Accedido: 25-may-2017].

[15] L. Gonzélez Palacio, *Método para generar Casos de prueba funcional en el desarrollo del softwre*, Revista Ingenierías Universidad de Medellín. 2009.

[16] «Protocolos de redes - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/Protocolos_de_redes. [Accedido: 24-may-2017].

[17] «Protocolo Simple de Administración de Red (SNMP) - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/SNMP. [Accedido: 24-may-2017].

[18] «Python 2.7.13 documentation». [En línea]. Disponible en: https://docs.python.org/2/. [Accedido: 29-may-2017].

[19] Raul Gonzales Duque, *Python para todos*. 2008.

[20] «Que es un Firewall y como funciona». [En línea]. Disponible en: http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-un-Firewall-y-como-funciona.php. [Accedido: 26-may-2017].

[21] «RFC 2647 - Benchmarking Terminology for Firewall Performance». [En línea]. Disponible en: file:///media/rich/Hewlett-Packard/13/RFC%202647%20-

%20Benchmarking%20Terminology%20for%20Firewall%20Performance.html. [Accedido: 30-may-2017].
[22] «RFC 2979 Behavior of and Requirements for Internet Firewalls». [En línea]. Disponible en: https://www.ietf.org/rfc/rfc2979.txt. [Accedido: 26-may-2017].
[23] «RFC 3093 Traduccin al español». [En línea]. Disponible en: file:///media/rich/Hewlett-Packard/13/RFC%203093%20Traducci%EF%BF%BDn%20al%20espa%EF%BF%BDol.html. [Accedido: 30-may-2017].
[24] «RFC 3511 - Benchmarking Methodology for Firewall Performance». [En línea]. Disponible en: https://tools.ietf.org/html/rfc3511. [Accedido: 26-may-2017].
[25] «Software-Defined Networking (SDN) Definition - Open Networking Foundation». [En línea]. Disponible en: https://www.opennetworking.org/sdn-resources/sdn-definition. [Accedido: 25-may-2017].
[26] «SSH - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/SSH. [Accedido: 24-may-2017].
[27] «Sublime text - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/Sublime_text. [Accedido: 25-may-2017].
[28] «Telnet - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/Telnet. [Accedido: 24-may-2017].
[29] «UML - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/UML. [Accedido: 25-may-2017].
[30] Martin Flower, *UML gota a gota*, Addison Wesley. México, 1999.
[31] «Visual Paradigm - EcuRed». [En línea]. Disponible en: https://www.ecured.cu/Visual_Paradigm. [Accedido: 25-may-2017].

Glosario de términos.

Firewall: es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas. Se trata de un dispositivo o conjunto de dispositivos configurados para permitir, limitar, cifrar, descifrar, el tráfico entre los diferentes ámbitos sobre la base de un conjunto de normas y otros criterios.

Switch: es el dispositivo digital lógico de interconexión de equipos que opera en la capa de enlace de datos del modelo OSI. Su función es interconectar dos o más segmentos de red.

Administrador de red: Una persona responsable de la configuración y administración de la red. El administrador generalmente configura la red, asigna contraseña y permisos, y ayuda a los usuarios.

Hardware: Término que hace referencia a cada uno de los elementos físicos de un sistema informático (pantalla, teclado, ratón, memoria, discos duros, microprocesador, entre otros.)

IP: Internet Protocol (Protocolo de Internet). Uno de los protocolos más representativos del estándar TCP/IP, es el responsable del esquema de direccionamiento utilizado en Internet y define la estructura y el intercambio de los datagramas IP entre redes distantes, definido en el RFC 791.

IDS: Sistema Detector de Intrusos.

PC: Personal Computer (PC). Computadora personal, es una computadora que puede ser una laptop o una computadora de escritorio.

Protocolo: Es el sistema, reglas y conjunto de normas que establece, gobierna y permite la comunicación entre dispositivos informáticos (la transferencia de datos)**.**

Python: Es un lenguaje de programación interpretado creado por Guido van Rossum en el año 1991.

Software: Son los ficheros, programas, aplicaciones y sistemas operativos que nos permiten trabajar con el ordenador o sistema informático. Se trata de los elementos que hacen funcionar al hardware.

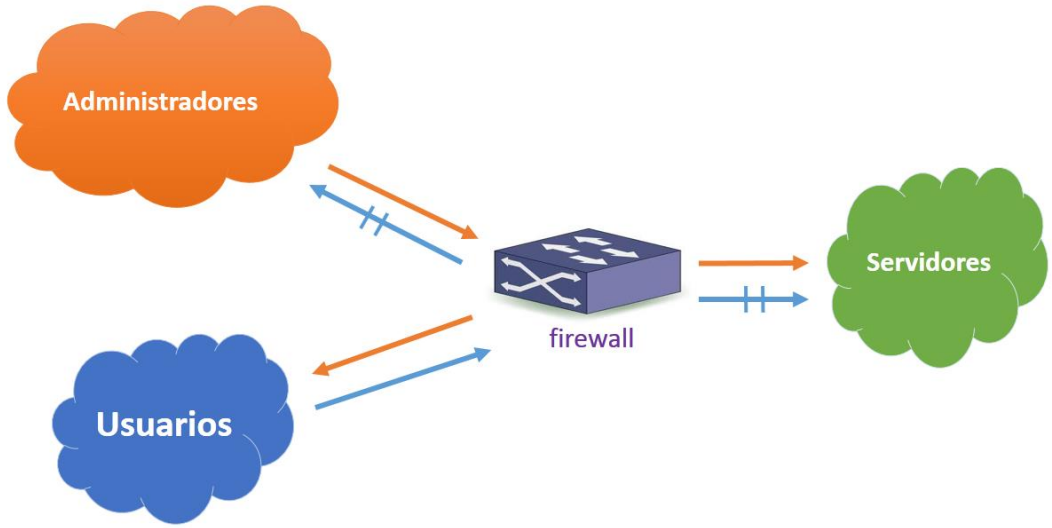
UML: (Unified Modeling Language): Es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema. Fue originalmente concebido por la Corporación Rational Software. El lenguaje ha ganado un significante soporte en la industria de varias organizaciones. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para plasmar un sistema de software antes de escribir cualquier línea de código.

Anexos

Anexo 1- Escenario de prueba # 1

```
root@rich-PC:/home/rich# firewall.py acl new usuarios_in
The ACL 'usuarios_in' has been created successfully
root@rich-PC:/home/rich# firewall.py zone new 33 -ip 230.220.110.1/28
The address 230.220.110.1/28 has been inserted
root@rich-PC:/home/rich# firewall.py rule insert usuarios_in 3 deny ip -s 10.10.111.2 -desc denegar_PC
This operation may take a few seconds. Please wait for a moment...
The rules have been inserted successfuly
root@rich-PC:/home/rich# firewall.py zone limit 33 usuarios_in in
The zone 33 has been limited in inbound
root@rich-PC:/home/rich#
```

Anexo 2- Escenario de prueba # 2



Anexo 3- Pruebas funcionales del caso de uso Autenticarse, validaciones 1,2 y 3

```
root@rich-PC:/# firewall.py 192.168.81.230 huawei qwerty huawei
usage: firewall.py <ip> <user> <password> <manufacturer> <model> huawei
       [-h] {5328}
firewall.py <ip> <user> <password> <manufacturer> <model> huawei: error: too few arguments
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py 270.168.0.230 huawei qwerty huawei 5328
usage: firewall.py <ip> <user> <password> <manufacturer> <model>
firewall.py: error: argument ipaddress: The ip address 270.168.0.230 is wrong
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py 192.168.81.230 huawei qwerty cisco 5328
usage: firewall.py <ip> <user> <password> <manufacturer> <model>
firewall.py: error: invalid choice: 'cisco' (choose from 'huawei')
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py 192.168.81.230 huawei qwerty huawei 12700
usage: firewall.py <ip> <user> <password> <manufacturer> <model> huawei
       [-h] {5328}
firewall.py <ip> <user> <password> <manufacturer> <model> huawei: error: argument model: invalid choice: '12700' (choose from '5328')
root@rich-PC:/#
```

Anexo 4- Prueba funcional del caso de uso Autenticarse, validación 4

```
root@rich-PC:/# firewall.py 192.168.81.1 huawei qwerty huawei 5328
Exception: Could not connect to telnet service. Please check you ip address
root@rich-PC:/#
```

Anexo 5- Prueba funcional del caso de uso Autenticarse, validación 5

```
root@rich-PC:/# firewall.py 192.168.81.230 huaw qwerty huawei 5328
Exception: Telnet authentication error. Please check your credentials
root@rich-PC:/#
```

Anexo 6- Pruebas funcionales de los casos de uso relacionados con zona, validaciones 1 y 2

```
root@rich-PC:/# firewall.py zone show -n 5987
The zone number must be lower than 4904
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py zone show -n 4000
Not exist the zone 4000
root@rich-PC:/#
```

Anexo 7- Pruebas funcionales de los casos de uso relacionados con zona, validaciones 3, 4 y 5

```
root@rich-PC:/# firewall.py zone new 3 -d esta_es_una_zona_de_prueba_y_esta_es_su_descripcion_la_cual_contiene_informacion_de_la_respectiva_zona
The description must have lower than 80 characters
The zone 3 is already exist
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py zone new 3 -ip 234.214.12.1 -d descripcion_de_la_zona_3
usage: [-h] <number> [-d DESCRIPTION] [-ip(s) IP_ADDRESS]
[-h] <show | new | limit | delimit| alter | delete> [args] new: error: argument -ip/--ipaddress: The subnet 234.214.12.1 is wrong
root@rich-PC:/#
```

Anexo 8 Pruebas funcionales de los casos de uso relacionados con zona, validaciones 6 y 7

```
root@rich-PC:/# firewall.py zone limit 3 acl1 in
There is not an ACL called: acl1
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py zone limit 3 acl1 salida
usage: [-h] <zone> <acl> <in | out>
[-h] <show | new | limit | delimit| alter | delete> [args] limit: error: argument boundary: invalid choice: 'salida' (choose from 'in', 'out')
root@rich-PC:/#
```

Anexo 9- Pruebas funcionales de los casos de uso relacionados con zona, validaciones 8 y 9

```
root@rich-PC:/# firewall.py zone limit 3 acl3 in
Cannot limit the zone 3 be cause is already limited in inbound
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py zone limit 3 acl3
usage: [-h] <zone> <acl> <in | out>
[-h] <show | new | limit | delimit| alter | delete> [args] limit: error: too few arguments
root@rich-PC:/#
```

Anexo 10- Pruebas funcionales de los casos de uso relacionados con zona, validaciones 10 y 11

```
root@rich-PC:/# firewall.py zone delimit 3  out
The zone 3 does not have a policy in outbound
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py zone delete 10
The zone 10 does not exist
root@rich-PC:/#
```

Anexo 11- Pruebas funcionales de los casos de uso relacionados con ACL, validaciones 1 y 2

```
root@rich-PC:/# firewall.py acl show -n este_es_el_nombre_de_la_ACL_de_la_cual_quiero_ver_su_informacion
The name must be lower than 31 characters
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py acl new -d descrpcion_de_la_ACL_60
usage: [-h] <name> [-d DESCRIPTION]
[-h] <show | new | alter | delete> [args] new: error: too few arguments
root@rich-PC:/#
```

Anexo 12- Pruebas funcionales de los casos de uso relacionados con ACL, validaciones 3 y 4

```
root@rich-PC:/# firewall.py acl alter acl10 esta_es_la_nueva_descripcion
There is not an ACL called acl10
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py acl new acl1
The ACL acl1 is already exist
root@rich-PC:/#
```

Anexo 13- Pruebas funcionales de los casos de uso relacionados con ACL, validaciones 5 y 6

```
root@rich-PC:/# firewall.py acl alter acl1 esta_Es_la_nueva_descripcion_de_la_acl_llamada_acl1_esta_se_esta_creardo_para_insertale_como_minimo_73
_reglas_para_crear_una_nueva_politica
The name and description must be between 31 and 127 characters respectively
root@rich-PC:/#
```

Anexo 14- Pruebas funcionales de los casos de uso relacionados con puerto, validaciones 1, 2, 3, 4 y 5

```
root@rich-PC:/# firewall.py port show -n 120
usage:  [-h] <-n | -l> [args]
<show | alter> show: error: argument -n/--number: The number port 120 is wrong
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py port show -n 33
The Switch Quidway S5328 only has 24 ports
root@rich-PC:/#
root@rich-PC:/# firewall.py port alter 23 hybrid 12
usage: <port> <type> <zone(s)>
<show | alter> alter: error: argument type: The port hybrid must be access or trunk
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py port alter 23 access 12,3,23
The port in mode 'access' just can assign one zone
The changes have made successfully
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py port alter 23 access
usage: <port> <type> <zone(s)>
<show | alter> alter: error: too few arguments
root@rich-PC:/#
```

Anexo 15- Pruebas funcionales de los casos de uso relacionados con regla, validaciones 1, 2, 3, 4 y 5

```
root@rich-PC:/# firewall.py rule insert acl1 90 deny
usage: <acl_name> <id_rule> <permit | deny> <ip | tcp | icmp | udp> [args]
<insert | delete> insert: error: too few arguments
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py rule insert acl10 90 deny ip
There is not an ACL called: acl10
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py rule insert acl1 9819909823 deny ip
The rule must be lower than 4294967294
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py rule insert acl1 98 allow ip
usage: <acl_name> <id_rule> <permit | deny> <ip | tcp | icmp | udp> [args]
<insert | delete> insert: error: argument action: invalid choice: 'allow' (choose from 'permit', 'deny')
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py rule insert acl1 98 permit igmp
usage: <acl_name> <id_rule> <permit | deny> <ip | tcp | icmp | udp> [args]
<insert | delete> insert: error: invalid choice: 'igmp' (choose from 'ip')
root@rich-PC:/#
```

Anexo 16- Pruebas funcionales de los casos de uso relacionados con regla, validaciones 6, 7, 8, 9 y 10

```
root@rich-PC:/# firewall.py rule insert acl1 90 deny ip -s 10.10.1234.1
usage: [-s SOURCE] [-d DESTINATION] [-log] [-desc DESCRIPTION]
<acl_name> <id_rule> <permit | deny> <ip | tcp | icmp | udp> [args] ip: error: argument -s/--source: The address 10.10.1234.1 is wrong
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py rule insert acl1 90 deny ip -s 10.10.123/56
usage: [-s SOURCE] [-d DESTINATION] [-log] [-desc DESCRIPTION]
<acl_name> <id_rule> <permit | deny> <ip | tcp | icmp | udp> [args] ip: error: argument -s/--source: The address 10.10.123/56 is wrong
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py rule insert acl1 90 deny ip -s 10.10.100.3-10.10.100.1
usage: [-s SOURCE] [-d DESTINATION] [-log] [-desc DESCRIPTION]
<acl_name> <id_rule> <permit | deny> <ip | tcp | icmp | udp> [args] ip: error: argument -s/--source: The ip address 10.10.100.3 must be lower tha
n 10.10.100.1
root@rich-PC:/#
root@rich-PC:/#
root@rich-PC:/# firewall.py rule insert acl1 90 deny ip -s 10.10.100.1-10.10.101.1
usage: [-s SOURCE] [-d DESTINATION] [-log] [-desc DESCRIPTION]
<acl_name> <id_rule> <permit | deny> <ip | tcp | icmp | udp> [args] ip: error: argument -s/--source: The range ip 10.10.100.1-10.10.101.1 cannot
enclose more than 255 ip address
root@rich-PC:/#
root@rich-PC:/# firewall.py rule delete acl1 33
The ACL acl1 not have any rule whit id 33
root@rich-PC:/#
root@rich-PC:/#
```