Universidad de las Ciencias Informáticas

Facultad 2

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Módulo Firewall para el software Segurmática antivirus para GNU/Linux

Autor:

Nayeli Garciga Rodríguez

Tutor(es):

Ing. Leydis Rodríguez Zamora

Ing. Javier Ricardo Ponce Pérez

La Habana

Junio del 2018

DECLARACIÓN DE AUTORÍA

Se declara ser la autora del trabajo de diploma con título Módulo Firewall para el software Segurmática antivirus para GNU/Linux y reconocemos a la Universidad de las Ciencias Informáticas sus derechos patrimoniales, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2018

_____

Nayeli Garciga Rodríguez

_____ _____

Ing. Javier Ricardo Ponce Pérez **Ing. Leydis Rodríguez Zamora**

DATOS DE CONTACTO

Autor

Nayeli Garciga Rodríguez

Correo electrónico: ngarciga@estudiantes.uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba

Tutores

Ing. Leydis Rodríguez Zamora

Correo electrónico: lrzamora@uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba

Ing. Javier Ricardo Ponce Pérez

Correo electrónico: jrponce@uci.cu

Universidad de las Ciencias Informáticas, La Habana, Cuba

Resumen

Entre las principales funcionalidades que debe estar presente en cada computadora personal (en lo adelante, PC) se encuentra la restricción del acceso no deseado desde otro dispositivo, es una de las características que más requiere las entidades con el fin de evitar la pérdida o robo de información vital. Actualmente el sistema de antivirus Segurmática, perteneciente a la Empresa de Consultoría y Seguridad Informática, trabaja en cómo lograr la detección de este tipo de delito informático mediante el estudio del funcionamiento y filtrado de paquetes en la red. Ocasionalmente es complejo detectar a tiempo la interrupción de algún usuario no deseado, así como aplicar reglas que limiten las conexiones y logren evitar la usurpación de datos imprescindibles.

Con el objetivo de solucionar las deficiencias detectadas, se decide llevar a cabo el desarrollo de una herramienta que sea capaz de detectar y bloquear las posibles interrupciones al sistema, mediante un módulo firewall para el antivirus Segurmática. Para esto se realizó un estudio del arte sobre los distintos sistemas de detección de intrusos, se seleccionaron las herramientas para la implementación obteniendo como resultados el modelo conceptual y diagrama de la base de datos, entre otros artefactos, además se realizaron pruebas para verificar la efectividad de la solución.

El módulo de firewall desarrollado contiene todas las funcionalidades necesarias para definir que equipos y cuales no tienen acceso a la computadora. Los clientes del sistema de antivirus podrán trabajar con un mayor nivel de protección en sus ordenadores.

Palabras clave

Antivirus, firewall, filtrado de paquetes, Segurmática.

Summary

Among the main features that must be present in each personal computer is the restriction of unwanted access from another device, is one of the features that most entities require in order to avoid loss or theft of vital information. Currently the antivirus system Segurmática, belonging to the Computer Security and Consulting Company, works on how to achieve the detection of this type of computer crime by studying the operation and filtering of packets in the network. Occasionally it is difficult to detect the interruption of any unwanted user in time, as well as to apply rules that limit connections and avoid the usurpation of essential data.

With the aim of solving the deficiencies detected, it is decided to carry out the development of a tool that is capable of detecting and blocking possible interruptions to the system, by means of a firewall module for the Segurmática antivirus. For this, a study of the art on the different intrusion detection systems was carried

out, the tools for the implementation were selected, obtaining as a result the conceptual model and diagram of the database, among other artifacts, in addition tests were carried out to verify the effectiveness of the solution.

The developed firewall module contains all the necessary functionalities to define which computers and which do not have access to the computer. The clients of the antivirus system will be able to work with a higher level of protection on their computers.

Keywords:

Antivirus, firewall, packet filtering, Segurmática.

ÍNDICE

INDICE DE TABLAS

INDICE DE FIGURAS

Introducción

En la actualidad, el uso de las computadoras es común en la vida cotidiana, pero cada computadora conectada a internet puede ser víctima de un ataque informático, por lo que ha pasado de ser una simple preocupación a una realidad. Actualmente personas con escasos conocimientos informáticos son capaces de realizar ataques desde las redes causando molestias y pérdidas, tanto simples como de millones de dólares en recursos y tiempo invertido. Ello se debe al desarrollo de herramientas informáticas que detectan vulnerabilidades, además son accesibles para el público y fáciles de encontrar por principiantes. Ejemplo de estos ataques están los sufridos por las grandes compañías (1), pero todos estamos propensos a recibir un ataque malintencionado. Existen programas dedicados a la seguridad, destacándose por su utilidad y popularidad como los siguientes: antivirus, proxies, firewall, monitores de red y sistema detectores de vulnerabilidades, analizadores de logs, sistemas de detección de intrusos, entre otros.

Con el paso del tiempo los sistemas operativos e internet han evolucionado, por lo que se ve más frecuente el uso de dispositivos de seguridad por parte de los usuarios. Entre ellos se encuentran los antivirus, que son programas que no sólo detectan los virus, sino que los bloquean, desinfectan archivos y previenen infecciones. Muchos antivirus tienen entre sus funcionalidades la protección de la información del usuario que puede estar comprometida por el uso de la internet o redes internas, además tienen integrado un firewall. Que no es más que un dispositivo de seguridad para la red que monitorea el tráfico entrante y saliente, permite o bloquea tráfico de información en función de un conjunto de reglas de seguridad definidas. Tiene gran importancia porque establecen una barrera entre las redes internas protegidas y controladas en las que se puede confiar y redes externas que no son de confianza, como Internet (2). Lo más relacionado es una puerta que prohíbe el paso de todos aquellos servicios no autorizados, pero deja pasar a todo aquel que el usuario necesita (si el servicio está autorizado, él lo deja pasar).

Entre los tipos de peligros que puede evitar tener un antivirus con firewall se encuentra el acceso por fallas o errores de configuración de Windows, instalación de publicidad o elementos de seguimiento como las cookies. También se encuentran los troyanos (aplicaciones ocultas que se descargan de la internet y que puede ser usada para extraer información personal por terceras personas), reducción del ancho de banda disponible por el tráfico de banners, sitios no solicitados y otros tipos de datos innecesarios que reutilizan la conexión.

Cuba no se encuentra ajena de la situación y con el aumento de la utilización del internet ve la necesidad de aumentar la seguridad en las redes. Como respuesta del gran avance en el mundo moderno crea la Empresa de Consultoría y Seguridad Informática, en forma abreviada SEGURMÁTICA, creada por la Resolución No.10/1995 del Ministerio de Economía y Planificación, acumula ya más de 20 años de experiencia, tiene como objeto social entre otros, comercializar licencias de uso y tecnologías de seguridad

informática, así como brindar servicios asociados a ellas, cuenta con el único laboratorio de Antivirus existente en el país y es capaz de brindar soporte técnico en todo el país, donde los usuarios siempre tendrán una respuesta técnica especializada en la mayor brevedad posible y la atención a todas sus inquietudes lo que contribuye a la mejora constante del producto, algo imposible de encontrar en los otros antivirus que están en el mercado. El producto líder de esta empresa es el Segurmática antivirus, programa antivirus de gran funcionalidad, lo cual contribuye a elevar la seguridad informática del sistema operativo de las computadoras, cuenta con una interfaz amigable y sencilla, que consume pocos recursos y fácil acceso a las actualizaciones, las cuales son incrementales y funciona en cualquier sistema operativo Windows a partir de XP (3).

Actualmente el antivirus de Segurmática para el sistema operativo Windows tiene implementado un firewall personal. Esta protección es esencial para todo antivirus porque monitorea el trafico entrante y saliente de información que pasa a través de él. El sistema GNU/Linux no tiene implementada dicha funcionalidad por lo que representa una desventaja con respecto al producto el Windows y además una debilidad en la seguridad de este producto. Una de las consecuencias que puede traer ataques de terceros es la obtención de la información personal o documentos importantes de los usuarios sin autorización de los antes mencionados.

Se ha delegado la responsabilidad de crear nuevas funcionalidades de Segurmática antivirus creada para el sistema operativo GNU/Linux a la Universidad de las Ciencias Informáticas (en lo adelante, UCI) que es un centro docente-productor donde especialistas y estudiantes desarrollan aplicaciones y servicios informáticos, orientados a los diversos sectores de la economía y los servicios en el ámbito nacional e internacional. Los productos de software libre de la UCI son avalados por clientes nacionales y extranjeros por su alta calidad.(4)

En el centro de Telemática se está desarrollando actualmente la interfaz visual y por consola para el antivirus SAVUnix para GNU/Linux. Este software presenta varias funcionalidades, como son la protección permanente, actualización, cuarentena, exclusiones, estadísticas, servidor corporativo y búsquedas. Pero una de las inestabilidades de SAVUnix es que no presenta un firewall personal dentro de sus funcionalidades.

Por todo lo antes expuesto, se plantea el siguiente **problema de investigación**: ¿Cómo controlar las conexiones entrantes y/o salientes aumentando la seguridad en el sistema?

Se define como **objeto de estudio** los sistemas de control de conexiones entrantes y/o salientes.

Por tanto, se plantea como **objetivo general** de este trabajo es desarrollar un módulo de firewall para el antivirus Segurmática en GNU/Linux, que controle las conexiones entrantes y/o salientes de la PC basado en reglas previamente configuradas por el usuario.

Se identifica como **campo de acción**: los sistemas de control de conexiones entrantes y/o salientes mediante un firewall para el antivirus Segurmática de GNU/Linux.

Para el cumplimiento de los objetivos se definen las siguientes **tareas de investigación**:

1. Análisis del estado actual de los diferentes sistemas antivirus y firewall para el desarrollo del software.
2. Selección de las herramientas y tecnologías para el desarrollo de software que faciliten el desarrollo del módulo para el control de conexiones entrantes y/o salientes**.**
3. Análisis y diseño del software para guiar el proceso de desarrollo.
4. Ejecución de las pruebas del software para comprobar su correcto funcionamiento.

Entre los **métodos científicos** utilizados destacan:

Como métodos teóricos: el **Analítico-Sintético** que se utilizó en la revisión de documentos, libros, artículos e informes para la extracción de elementos importantes que están relacionados con los antivirus y firewalls existentes. Se aplicó además el método **Inductivo–Deductivo** se empleó para la identificación de la problemática relacionada con el proceso de flujo de información entrantes y salientes, así como las soluciones a este problema.

El método **Histórico-Lógico** se empleó para el estudio crítico de los trabajos anteriores relacionados con los firewalls para GNU/Linux; y el método de **Modelación** se empleó para modelar teóricamente el sistema para el módulo firewall para el software antivirus Segurmática en GNU/Linux y de esa manera comprender mejor el negocio, para ello se utilizaron los diagramas de casos de uso del sistema y el modelo conceptual.

El presente trabajo consta de 3 capítulos estructurados de la siguiente manera:

Capítulo 1: Fundamentos teóricos sobre el desarrollo de un firewall. Abordan los principales conceptos relacionados con los firewalls, así como una descripción de las herramientas, tecnologías y la metodología a emplear en la solución del problema planteado.

Capítulo 2: Análisis y diseño de un firewall. Se describen los pasos de la metodología utilizada, con el objetivo de facilitar la construcción del firewall. En principio se identifican los requisitos del sistema tanto funcionales como no funcionales, se confeccionan los casos de uso, los diagramas de base de datos y el de clases, además se define la arquitectura a utilizar.

Capítulo 3: Implementación y validación del firewall. Se describe el proceso de implementación del firewall, mediante la utilización de las herramientas descritas en el capítulo primero. Además, se realiza la validación del firewall, a partir de pruebas que permitirá comprobar la veracidad y factibilidad del sistema a desarrollar.

Capítulo 1: Fundamentos Teóricos sobre el Desarrollo de un Firewall

Durante este capítulo, se realiza un estudio de la fundamentación teórica de los firewalls relacionados fundamentalmente con GNU/Linux. Se abordará los conceptos generales relacionados con el firewall para expandir el conocimiento general, se explicará la metodología AUP-UCI y las herramientas empleadas en el desarrollo del módulo.

1.1 Conceptos generales

Los conceptos asociados al problema que se explican a continuación permiten tener un dominio teórico del proceso de protección de software. Este epígrafe es de gran importancia para el desarrollo del trabajo investigativo la información adquirida en la investigación servirá de base para los resultados.

1.1.1 Seguridad Informática

Según CCM[1], la seguridad informática consiste en garantizar que el material y los recursos de software de una organización se usen únicamente para los propósitos para los que fueron creados y dentro del marco previsto. La seguridad informática se resume, por lo general, en cinco objetivos principales: **integridad**, que garantiza que los datos sean los que se supone que son; c**onfidencialidad**, que asegura que solo los individuos autorizados tengan acceso a los recursos que se intercambian; **disponibilidad**, que garantiza el correcto funcionamiento de los sistemas de información; **evitar el rechazo**, que garantiza de que no pueda negar una operación realizada; **autenticación**, que asegura que solo los individuos autorizados tengan acceso a los recursos. (5)

1.1.2 Firewall

Según CISCO un firewall es un dispositivo de seguridad de la red que monitorea el tráfico de red entrante y saliente y decide si permite o bloquea tráfico específico en función de un conjunto definido de reglas de seguridad.(2)

Según CCM un firewall es un sistema que protege a un ordenador o a una red de ordenadores contra intrusiones provenientes de redes de terceros (generalmente desde Internet). Un sistema de firewall filtra paquetes de datos que se intercambian a través de Internet. Por lo tanto, se trata de una pasarela de filtrado que comprende al menos las siguientes interfaces de red: una interfaz para la red protegida (red interna) y una interfaz para la red externa.(6)

Como se puede apreciar en las bibliografías consultadas, existen varias coincidencias en cuanto a que es un firewall, pero cada una de ellas tienen sus diferencias, por lo que nos vamos a adherir al concepto dado por CISCO ya que es el que más se ajusta según la decisión del autor.

---

[1] En español “modelo de madurez de capacidades”, en inglés: "Capability Maturity Model (CMM).

1.1.3 Firewall Personal

El término firewall personal se utiliza para los casos en que el área protegida se limita al ordenador en el que el firewall está instalado.

Un firewall personal permite controlar el acceso a la red de aplicaciones instaladas en el ordenador y prevenir notablemente los ataques de programas como los troyanos, es decir, programas dañinos que penetran en el sistema para no permitir que un hacker controle el ordenador en forma remota. Los firewall personales permiten subsanar y prevenir intrusiones de aplicaciones no autorizadas a conectarse a su ordenador.(7)

1.2 Sistemas similares

En términos generales, el panorama mundial muestra que se están desarrollando soluciones de seguridad informática que se caracterizan por ser cada vez más fiables y fáciles de gestionar. Un antivirus es esencial para resguardar su seguridad, su privacidad y asegurarte que estas protegidos de virus malware, spyware y otras amenazas cibernéticas. Se pueden aplicar a distintos sistemas operativos por lo que puede haber un proveedor de antivirus en distintas plataformas con funcionalidades completamente diferentes. A continuación, se presentas diferentes antivirus tanto de GNU/Linux como de Windows para comparar el comportamiento y presencia de firewall en ellos:

1.2.1 ZoneAlarm Antivirus

ZoneAlarm Antivirus es gratuito, confiable y efectivo. Entre sus diferentes servicios para proteger al usuario se encuentra la de antivirus y antispyware donde se detenta y eliminan virus, spyware, troyanos, gusanos, bots y otras amenazas maliciosas. El Firewall Personal tiene como función desactivar instantáneamente los programas maliciosos y mantener una protección proactiva contra los ciberataques, detiene los ataques de Internet apenas se producen. Contiene servicio de protección de identidad y respaldo en línea donde crea copias de respaldo automatizadas cuando le resulte cómodo y restaure sus archivos y datos en caso de pérdida, robo, eliminación accidental o falla de un disco. Todos sus datos se almacenarán con una clave de cifrado protegida y una contraseña que solo sabrá usted.

Entre sus productos se encuentra **ZoneAlarm** Pro Antivirus & Firewall 2018 que contiene una gran gama de servicios entre los que se encuentra motor de análisis antivirus/anti-software espía, antivirus en tiempo real avanzado, firewall avanzado y el firewall bidireccional que detiene los ataques de internet en la puerta principal. Protege proactivamente contra ataques entrantes y salientes, al mismo tiempo que te hace invisible para los hackers. Vigila y bloquea el tráfico de las amenazas, tanto entrante como saliente. Modo totalmente silencioso, para hacer su PC invisible a los piratas informáticos y los controles de eliminación desactivan inmediatamente los programas dañinos. Este es un producto de pago por tener mayor número de funcionalidades y propiedades específicas, personalizado en la protección tanto antivirus como

firewall.(8)

1.2.2 Comodo Internet Security

Comodo Internet Security ofrece protección completa de virus, troyanos, gusanos, desbordamientos de búfer, ataques de día cero, spyware y hackers. Comodo Internet Security le avisa cuando un potencial malware intenta atacar o acceder a su sistema. El programa combina un potente software de protección contra virus, un firewall de filtrado de paquetes de clase empresarial (protege su sistema contra todos los ataques entrantes y salientes), prevención de intrusiones de host avanzada, control de aplicaciones y antispyware en una aplicación supremamente poderosa.

Construido desde cero con su seguridad en mente, Free Internet Security ofrece protección de 360 ° combinando una poderosa protección Antivirus, un firewall de filtrado de paquetes de clase empresarial, prevención avanzada contra intrusiones de host y sandboxing automático de archivos desconocidos. La suite de seguridad en Internet de Comodo difiere del software antivirus tradicional en que también incluye otras capas de protección, incluyendo antispyware, controles parentales, protección de privacidad y mucho más. Esta suite de seguridad en Internet gratuita es un paquete completo que puede descargar e instalar para su tranquilidad. Este software gratuito de seguridad en Internet, defiende su PC de software malicioso como virus y evita que se robe su información personal.(9)

1.2.3 Bitdefender Total Security

Bitdefender ofrece una seguridad sólida con una infraestructura de prestación de seguridad a escala mundial, soluciones visionarias y gran cantidad de galardones, llevamos desde 2001 siendo un proveedor de seguridad de total confianza. Colabora con organizaciones gubernamentales, grandes empresas, pymes y particulares en más de 150 países. Proporciona protección para un solo dispositivo, un hogar inteligente, su pequeña empresa, infraestructuras híbridas o un centro de datos corporativo, ofrece seguridad, buen rendimiento y una increíble facilidad de uso. Mantiene una seguridad completa para Windows, MacOS, iOS y Android.

Entre sus funcionalidades se encuentra una completa protección de datos, defensa contra amenazas avanzadas, protección multicapa contra ransomware, Anti-Phishing, antifraude, archivos seguros, navegación segura, modo de rescate, antirrobo, destructor de archivos, firewall de la privacidad, protección de redes sociales y evaluación de vulnerabilidad.

1.2.4 Comodo Antivirus

Comodo Antivirus para GNU/Linux (CAVL) ofrece la misma gran protección contra virus que el software de Windows con el beneficio adicional de un sistema antispam completamente configurable. Con escáneres de virus on-access y bajo demanda, CAVL también utiliza el análisis de comportamiento basado en la nube de archivos desconocidos para proporcionar una protección inigualable contra el malware de día cero. El

potente motor de AV se complementa con una puerta de enlace de correo altamente configurable para filtrar el correo no deseado y bloquear las amenazas transmitidas por correo electrónico.

Entre sus características se encuentran: La protección antivirus proactiva intercepta todas las amenazas conocidas. Actualizaciones automáticas para la protección antivirus más actualizada. Incluye un programador de escaneo, un visor de eventos detallado y perfiles de escaneo personalizados. El filtro de correo es compatible con Postfix, qmail, Sendmail y Exim MTA's. Instalar y olvidar. Sin falsas alarmas molestas, solo protección de virus sólida.(10)

1.2.5 ESET NOD32

El antivirus ESET NOD32 para el sistema operativo GNU/Linux ofrece protección antimalware, multiplataforma y medios de almacenamiento externos. Entre sus funcionalidades se encuentra antivirus y anti-espía: donde protege tu identidad y tu información privada. Detecta amenazas multiplataforma independientemente del sistema al que se dirijan: Windows, GNU/Linux o Mac OS.

Bajo consumo de recursos: Diseñado para consumir pocos recursos, puedes confiar en un arranque de tu equipo rápido y un funcionamiento sin ralentizaciones. Fácil de usar: Contiene una interfaz simple y con un diseño que pone toda la información vital y las funciones de seguridad al alcance de tu mano. Se ejecuta en cualquier sistema con Debian, RedHat, Ubuntu, SuSe, Fedora, Mandriva y la mayoría de las distribuciones de RPM y DEB. (11)

1.2.6 Panda Endpoint Protection Plus

Las soluciones tradicionales de seguridad informática no son capaces de bloquear el sofisticado malware actual. Nuestra tecnología inteligente, basada en Big Data y la Inteligencia Artificial, monitoriza todas las aplicaciones que se ejecutan en los sistemas, clasificando absolutamente todo. A diferencia de las soluciones antivirus tradicionales, que sólo actúan cuando un proceso es malicioso, nuestra tecnología detecta los ataques incluso antes de que se produzcan.

Como características principales para su producto dispone de: Big Data y Machine Learning: Nuevo modelo de Panda basada en la inteligencia de comportamiento. Antivirus de nueva generación: Detecta los malware conocidos, ataques sin archivo, así moco cualquier otro comportamiento malicioso antes que se produzca. Monitorización continua: monitorización, registro y clasificación continua del 100% de los procesos ejecutados, ofreciendo protección contra cualquier tipo de amenaza. Prevención: Prevención, detención y remediación. Información detallada: Información forense detallada, auditorias de seguridad, y alarmas en tiempo real. Visualidad granulada y exhaustiva. (12)

De los diferentes antivirus se han visto sus características principales en dos sistemas operativos diferentes, se ve como los antivirus de Windows contiene como una funcionalidad importante la incorporación de un firewall mientras que en GNU/Linux no se incluye esa funcionalidad, ello se puede ver como una desventaja

desde el punto de vista del usuario para los antivirus de este sistema operativo en particular. Por ello se estudiará los diferentes firewalls en el sistema operativo GNU/Linux capaces de proteger al usuario de ataques de terceras personas ya que la solución propuesta por el cliente es la integración de este en el antivirus Segurmática para GNU/Linux.

1.2.7 Firewall

Se escogieron para comparar los firewalls expuestos ya que cumplen con las siguientes características como que sea fácil de utilizar, poderlos instalar en la PC y que tengan un entorno seguro a través de internet. Seguidamente se realiza un análisis de las características de algunos firewalls para llegar a una mejor solución.

IPCop Firewall

El Firewall de IPCop es un firewall de GNU/Linux que está dirigido a los usuarios del hogar y SOHO (pequeñas oficinas) La interfaz web IPCop es muy fácil de usar y facilita el uso. Puede configurar una PC como una red privada virtual (VPN[2]) segura para proporcionar un entorno seguro a través de Internet. Incluye información de uso frecuente para proporcionar una mejor experiencia de navegación web a los usuarios.

Dentro de sus principales características tenemos: cuenta con una interfaz Web codificada por color la cual nos permite monitorear los gráficos de rendimiento para CPU[3], memoria y disco, así como el rendimiento de la red, visualiza y rota automáticamente registros, soporte de múltiples idiomas y proporciona una actualización estable y fácilmente implementable segura y agrega parches actuales a nivel de seguridad (13)

Shorewall

Shorewall es una herramienta de configuración de gateway[4] o firewall para GNU/Linux. Con Shorewall tenemos una herramienta de alto nivel para configurar filtros de red ya que nos permite definir requisitos de firewall mediante entradas en un conjunto de archivos de configuración definidos.

Shorewall puede leer los archivos de configuración y gracias a la ayuda de las utilidades iptables, iptables-restore, ip y tc. Shorewall puede configurar Netfilter y el subsistema de red GNU/Linux para que coincida con los requisitos establecidos. Shorewall se puede utilizar en un sistema de firewall dedicado, un enrutado, un servidor multi-función o en un sistema GNU/Linux autónomo.

Sus principales características son: usa las instalaciones de seguimiento de conexión de Netfilter para filtrado de paquetes con estado, soporta una amplia gama de aplicaciones de routers, firewall y gateway.

---

[2] En español “red privada virtual” (RPV), en inglés: Virtual Private Network (VPN).

[3] En español “unidad central de procesamiento” , en inglés: Central Processing Unit (CPU).

[4] En español "puerta de enlace".

Administración de firewalls centralizada, iinterfaz GUI con el panel de control de Webmin, soporta Masquerading[5] y reenvío de puertos, ssoporta VPN. (14)

UFW – Uncomplicated Firewall

UFW se ha posicionado actualmente como uno de los firewalls más útiles, dinámicos y simples de usar en ambientes GNU/Linux. UFW significa Uncomplicated Firewall, y es un programa desarrollado para administrar un firewall de Netfilter. Proporciona una interfaz de línea de comandos y tiene como objetivo ser sencillo y fácil de usar, de allí su nombre UFW proporciona un marco simple para administrar Netfilter, así como nos proporciona una interfaz de línea de comandos para controlar el firewall desde la terminal, es especialmente adecuado para firewalls basados en host.

Dentro de sus características encontramos: compatible con IPV6[6], opciones de registro extendido con conexión y desconexión, supervisión del estado del firewall, marco extensible, puede ser integrado con aplicaciones y permite añadir, eliminar o modificar reglas de acuerdo a las necesidades.(15)

Vuurmuur

Vuurmuur es un administrador de firewall construido sobre iptables en GNU/Linux. Cuenta con una configuración simple y fácil de usar la cual permite configuraciones simples y complejas. La configuración se puede configurar completamente a través de una GUI[7] de Ncurses, la cual permite la administración remota segura a través de SSH[8] o en la consola.

Vuurmuur es compatible con la configuración del tráfico, cuenta con potentes funciones de monitoreo las cuales permiten al administrador ver los registros, las conexiones y el uso del ancho de banda en tiempo real. Vuurmuur es un software de código abierto y se distribuye bajo los términos de la GNU GPL[9].

Dentro de sus características encontramos: no requiere de amplios conocimientos de iptables, posee sintaxis de reglas legibles por humanos, soporta IPv6 (experimental), incluye modelado del tráfico, hace muy simple, fácil de configurar con NAT[10] e incluye política predeterminada segura. Totalmente manejable a través de SSH y desde la consola (incluyendo desde Windows usando PuTTY), scriptable[11] para la integración con otras herramientas, incluye características anti-spoofing, visualización de la conexión en

---

[5] En español "suplantar identidad"

[6] En español "protocolo del internet" en inglés: Internet Protocol (IP).

[7] En español " interfaz gráfica de usuario ", en inglés: graphical user interface (GUI).

[8] En español " intérprete de órdenes seguro " en inglés: Secure SHell (SSH).

[9] En español " licencia Pública General ", en inglés: General Public License (GNU).

[10] En español " traducción de direcciones de red ", en inglés: Network Address Translation (NAT).

[11] En español “programable”.

tiempo real, cuenta con registro de auditoría: todos los cambios se registran, registro de nuevas conexiones y malos paquetes y contabilidad del volumen de tráfico en tiempo real.(16)

pfSense

pfSense es una distribución de firewall de red gratuita, basada en el sistema operativo FreeBSD con un kernel personalizado e incluye paquetes de software gratuitos de terceros para una funcionalidad adicional. El software pfSense, con la ayuda del sistema de paquete, puede proporcionar la gran funcionalidad o más de firewalls comerciales comunes, sin ninguna de las limitaciones artificiales. Ha sustituido con éxito a todos los firewalls comerciales de gran renombre que pueda imaginar en numerosas instalaciones en todo el mundo, incluidos Check Point, Cisco PIX, Cisco ASA, Juniper, Sonicwall, Netgear, Watchguard, Astaro y más.

Dentro de sus características encontramos: altamente configurable y actualizado desde su interfaz basada en Web, se puede desplegar como firewall perimetral, enrutador, servidor DHCP y DNS. Se puede configurar como punto de acceso inalámbrico y punto final VPN, ofrece *Traffic shaping* [12]e información en tiempo real sobre el servidor y balanceo de carga entrante y saliente. (17)

Se presenta una tabla comparativa donde se ven como parámetros las características que debe utilizar nuestro firewall de Segurmática antivirus:

Sistemas	**IPCop Firewall**	**pfSense**	**UFW – Uncomplicated Firewall**	**Vuurmuur**	**Shorewall**
instalar en la PC					
fácil de usar		**X**			**X**
permite gestionar reglas	**X**				**X**
supervisión del estado del firewall					
Entorno seguro en internet					
Implementado en C++	**X**	**X**	**X**	**X**	**X**
Iptables/netfilter	**X**				

[12] En español “modelado de tráfico”.

Como se ve en la tabla no se puede utilizar ninguno de los antivirus antes propuestos ya que no cumple la mayoría de los requisitos propuestos, entre ellos los más funcionales serian Vuurmuur y Uncomplicated Firewall ya que cumplen la mayoría de los requisitos, entre ellos configurar las iptables que es una herramienta de firewall ampliamente utilizada que interactúa con el marco de filtrado de paquetes netfilter del kernel. Posteriormente se explicará en que consiste netfilter y iptables:

1.2.8 Netfilter

Netfilter es un framework del kernel de GNU/Linux que permite realizar diversas operaciones con la red tales como filtrado de paquetes, traducción de direcciones y puertos (NATP), rastreo de conexiones (Connection Tracking) y otro tipo de operaciones de manipulación de paquetes.

Las funciones de firewall en los sistemas operativos GNU/Linux son soportadas por este framework y es de gran importancia que se entienda su funcionamiento para el transcurso de este proyecto fin de carrera.

Netfilter es una parte importante del kernel de GNU/Linux, y como la mayoría de funcionalidades, está programado en lenguaje C. Netfilter proporciona al sistema operativo la funcionalidad necesaria para encaminar paquetes entre redes y para bloquear flujos de paquetes destinados a segmentos sensibles de la red [38].

Netfilter comprende un conjunto de “hooks” dentro del kernel de GNU/Linux, lo cual permite a los módulos específicos del kernel realizar llamadas al sistema con la pila de red del kernel.

Dichas llamadas, por lo general, se invocan cada vez que un paquete es procesado por un “hook” perteneciente a la pila. La herramienta de control de Netfilter (iptables) junto con el sistema de seguimiento de conexiones (Connection Tracking System), el subsistema de traduccion de direcciones y puertos (NATP) y el propio core de Netfilter conforman la mayor parte del framework Netfilter.

Siendo un poco más especıfico, las características generales del framework Netfilter son:

- Filtrado de paquetes sin control de estado.
- Filtrado de paquetes con control de estado.
- Traducción de direcciones y puertos NAT/NATP (18).

1.2.9 Iptables

Iptables es la herramienta del espacio de usuario que permite configurar Netfilter en base a un conjunto de reglas a través de la línea de comandos. Es una herramienta especialmente pensada para el uso por parte de administradores de sistemas para facilitarles la configuración. Como la traducción de direcciones de red también se configura desde el conjunto de reglas de filtro de paquetes, también se usa iptables. El paquete de iptables también incluye ip6tables. ip6tables se usa para configurar el filtro de paquetes IPv6.

Iptables requiere un kernel que tenga el filtro de paquetes ip_tables. Esto incluye todas las versiones del kernel 2.4.x y posteriores. Entre sus principales características están:

- Enumerar los contenidos del conjunto de reglas del filtro de paquetes
- Agregar / eliminar / modificar reglas en el conjunto de reglas del filtro de paquetes
- Enumeración / puesta a cero de contadores por regla del conjunto de reglas del filtro de paquetes. (19)

El Antivirus Segurmática tiene como objetivo crear un firewall fácil de usar en el sistema operativo GNU/Linux para mejorar la seguridad de sus usuarios en la red como resultado de la investigación se estudiaron diferentes firewalls presentes en el mercado para GNU/Linux, **por lo que se llega a la conclusión** de crear un módulo para el antivirus Segurmática en GNU/Linux y como base para ello utilizar iptables/netfilter ya que es un firewall integrado al kernel de Linux. Los estudios de los diferentes antivirus con firewall fueron de ayuda para el diseño de la interfaz y entre las características que ellos cumplen se escogieron para la creación del nuevo módulo las siguientes: el usuario no requerirá de amplios conocimientos informáticos para trabajar con el firewall, permitirá añadir, eliminar o modificar reglas de acuerdo a las necesidades y supervisar el estado del firewall. Además de poder definir cuáles serán las conexiones permitidas y las denegadas según reglas definidas por el cliente. Revisar los registros de cambios y listar todas las reglas añadidas al sistema incluyendo las predefinidas por GNU/Linux.

1.3 Metodología

Para el diseño y desarrollo de proyectos de software se aplican metodologías, modelos y técnicas que permiten resolver los problemas que se presenten durante el desarrollo. Una metodología impone un proceso de forma disciplinada sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente. Las metodologías definen una representación que permite facilitar la manipulación de modelos, y la comunicación e intercambio de información entre todas las partes involucradas en la construcción de un sistema. (20)

Para desarrollar el módulo firewall personal en el Antivirus Segurmática para GNU/Linux el autor decidió utilizar una metodología de desarrollo de software ágil, variación del Proceso Unificado Ágil (AUP) propuesto por la Universidad de las Ciencias Informáticas (en adelante AUP-UCI) en su escenario 2. Se decide utilizar esta metodología porque es la empleada en el centro de desarrollo de Telemática al cual pertenece este proyecto. Ella describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en el Proceso Unificado Relacional (RUP). Se caracteriza por estar dirigida por casos de uso, centrada en la arquitectura y por ser iterativo e incremental.

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando su objetivo, se unifican las restantes 3 fases de AUP en una sola, llamada Ejecución y se agrega la fase de Cierre.

AUP propone 7 disciplinas (modelo, implementación, prueba, despliegue, gestión de configuración, gestión de proyecto y entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 8 disciplinas, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: modelado de negocio, requisitos y análisis y diseño en AUP están unidos en la disciplina modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina implementación, en el caso de prueba se desagrega en 3 disciplinas: pruebas internas, de liberación y aceptación y la disciplina despliegue se considera opcional. (21)

1.4 Herramientas

Para el desarrollo del módulo Firewall Personal en el Antivirus Segurmática para GNU/Linux debe garantizar el control del tráfico entrante y saliente de la PC del usuario se utilizan herramientas y tecnologías libres, teniendo en cuenta la soberanía e independencia tecnológica del país y llevando a cabo las políticas de migración a software libre.

1.4.1 Lenguaje de Modelado

UML v2.1 (Lenguaje Unificado de Modelado) como su nombre indica es un lenguaje de modelado para describir métodos o procesos. Con el objetivo de detallar los artefactos del sistema, construir y describir un modelo en sí. Incluye conceptos como funciones del sistema, procesos de negocio además de aspectos concretos como esquemas de bases de datos y lenguajes de programación. En esencia permite visualizar, documentar, especificar y construir software orientado a objetos.(22) Se utiliza para la realización de los diagramas lógicos y físicos de la base de datos, el de despliegue y el de clase.

1.4.2 Herramienta CASE

CASE (Computer Aided Software Engineering) comprende un amplio abanico de diferentes tipos de programas que se utilizan para ayudar a las actividades del proceso de software, como el análisis de requerimientos, el modelado del sistema, la depuración y las pruebas. Las herramientas CASE también incluyen un generador de código que automáticamente genera código fuente a partir del modelado del sistema y de algunas guías de procesos para los ingenieros de software.(20)

Visual Paradigm v8.0: Es una herramienta CASE, que utiliza el lenguaje UML profesional, que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Su diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad. Uso de un lenguaje estándar común a todo el equipo de desarrollo para facilitar la

comunicación. (23) Esta herramienta es utilizada para el modelado los diagramas lógicos y físicos de la base de datos, el de despliegue y el de clase.

1.4.3 Lenguaje de programación

Como lenguaje de programación orientado a objetos se utilizó **C++** ya que este es el lenguaje utilizado en el desarrollo de los otros módulos que componen el software Segurmática Antivirus para GNU/Linux.

Es importante destacar que C++ es un lenguaje híbrido, es decir, es posible programar en estilo imperativo (como C) o en estilo orientado a objeto como java, además es un lenguaje amigable, flexible, muy potente para el programador ya que combina la flexibilidad de los lenguajes de alto nivel con el control y la funcionalidad que ofrecen los lenguajes ensambladores. Brinda la posibilidad de crear clases, plantillas, sistema de espacios de nombres y funciones en línea. Permite la sobrecarga de operadores y los utiliza para el manejo de memoria.(24)

1.4.4 Entorno de Desarrollo

Un Entorno de Desarrollo Integrado (IDE) puede incluir uno o varios lenguajes de programación. Tiene el objetivo de ganar fiabilidad y tiempo en los proyectos de software. Proporciona al programador una serie de componentes con igual interfaz gráfica, con la consiguiente comodidad, aumento de eficiencia y reducción de tiempo de codificación. (25)

QT es una SDK[13] multi-plataforma y un framework de interfaz de usuario utilizado para el desarrollo de aplicaciones con soporte a más de una docena de plataformas. El framework Qt se compone de módulos multi-plataformas de clases C ++, bibliotecas Qt y un entorno de desarrollo integrado con Qt Creator IDE y varias herramientas. (26)

QtCreator v5.2 es un IDE (Entorno de desarrollo integrado) para el desarrollo de aplicaciones. Es soportado por sistemas operativos como GNU/Linux, Mac OS X, Windows XP, Vista, Windows 7, por lo que es multiplataforma y además software libre. Permite construir interfaces de usuario complejas de una forma visual y rápida, ya que incluye un editor de texto con autocompletado, diseñador de interfaces gráficas, gestión de proyectos, sistema de depuración e integración con sistemas de control de versiones.(27)

Es empleada esta herramienta ya que es fácil de utilizar por los programadores por su comodidad, además es la utilizada por el centro de desarrollo y es el Entorno de desarrollo donde esta implementado el software Segurmática antivirus.

[13] En español " Kit de desarrollo de software", en inglés: Software Development Kit (SDK).

1.4.5 Base de Datos

SQLite III es una biblioteca en proceso que implementa un motor de base de datos SQL transaccional independiente, sin servidor y de configuración cero. El código para SQLite es de dominio público y, por lo tanto, es gratuito para cualquier uso, permite almacenar información en dispositivos móviles y fijos de una forma eficaz, potente, rápida y en equipos con pocas capacidades de hardware. SQLite implementa el estándar para la sintaxis y semántica de los lenguajes de bases de datos SQL y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo. Esto permite que SQLite soporte desde las consultas más básicas hasta las más complejas del lenguaje SQL, ya que existe compatibilidad al 100% entre las diversas plataformas disponibles, haciendo que la portabilidad entre dispositivos y plataformas sea transparente. SQLite está construida en C, lo cual facilita la migración a diversas plataformas de sistemas operativos y de dispositivos. Dado que una base de datos de SQLite se almacena por completo en un solo archivo, está puede ser exportada a cualquier otra plataforma y tener interoperabilidad al 100% sin ningún requerimiento de programación adicional o cambios de configuración. Se utilizará como gestor de base de datos ya que es fácil de utilizar y por todas las ventajas anteriores.(28)

Se utiliza ya que es una base de datos interna, independiente, sin servidor, permite realizar desde consultas básicas hasta las más complejas en SQL por lo que es la base de datos necesaria para guardar los datos internos de la aplicación.

1.5 Conclusiones del capítulo

En el estudio sobre el estado del arte relacionado con los antivirus con firewall y los firewalls en GNU/Linux, permitió definir los conceptos necesarios para completar la investigación concluyó utilizar iptables/netfilter. Se seleccionaron las herramientas y tecnologías necesarias para el desarrollo del módulo del firewall de Segurmática antivirus para GNU/Linux. Para la confección de la propuesta de solución se utilizó la metodología AUP-UCI para guiar el proceso de desarrollo del software. Se definió como lenguaje de programación: C++, como framework de desarrollo QT Creator en su versión 5.2, como sistema gestor de base de datos SQLite III, como herramienta CASE Visual Paradigm en su versión 8.0 y lenguaje de modelado UMLv2.1.

Capítulo 2: Propuesta de Solución

Para el desarrollo del sistema informático es importante entender la lógica del negocio para dar la solución que más se ajuste a las necesidades del cliente y a sus exigencias. En este capítulo se describe la propuesta de solución a desarrollar, por lo que se expone sus características principales, realizando los artefactos relacionados con la fase de ejecución definidos por la metodología seleccionada. Asimismo, se efectúa las descripciones de requisitos por caso de uso referente a las funcionalidades que presenta la solución.

2.1 Propuesta de solución

Para darle solución al problema planteado se decide realizar un módulo firewall para Segurmática Antivirus utilizando el marco de trabajo QT Creator. El Segurmática Antivirus es un software orientado a la protección contra los programas malignos existentes en los sistemas operativos. Este producto está conformado por dos componentes fundamentales, el Servidor destinado a la comunicación con las aplicaciones antivirus y su gestión y la Consola de Administración con una interfaz típica de estos productos. A pesar de lo antes expuesto, el software no cuenta con una herramienta capaz de controlar y monitorear los paquetes que viajan a través de la red. Por lo que se plantea, la necesidad de crear un módulo Firewall que permita añadirle estas características claves para el incremento de las funcionalidades básicas de un antivirus, incrementando de esta forma la seguridad del cliente y su información. Inicialmente el usuario debe acceder al módulo firewall por medio del menú en el antivirus, si es administrador debe autentificarse para obtener los permisos del sistema. Luego en dicho módulo se mostrará la gestión de reglas, el registro de cambios y el habilitar o deshabilitar el firewall. La gestión de reglas es un proceso mayor que trae consigo la adición, modificación y eliminación de información de las reglas en la base de datos, así como listar las reglas. El registro de cambios es donde se muestran las modificaciones realizadas a las reglas, como son la adicción, modificación o eliminación y a qué hora y día se hizo. El habilitar pone a el firewall a funcionar y las reglas en práctica, mientras el deshabilitar lo desactiva.

2.2 Modelo Conceptual

En el modelo conceptual se describe como se relacionan los conceptos de un problema. Se utiliza para representar un problema de manera gráfica a través del diagrama de entidad relación, diccionarios/glosarios y diagrama de clases, entre otros.(29) A continuación, se muestra el modelo conceptual donde se describe desde una forma general la relación conceptual entre el firewall que contiene una lista de reglas y un registro de cambios que son configurados en el sistema. El firewall permite gestionar estas reglas que actualizan el registro de cambios.

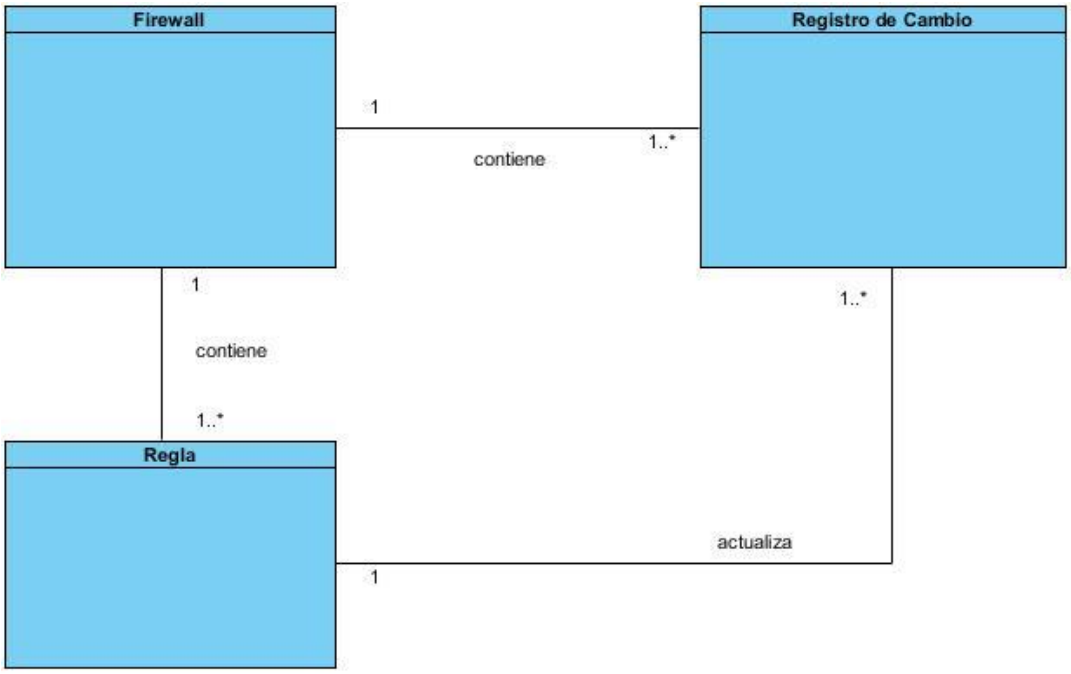


Figura 1. Modelo Conceptual

Diccionario de datos

- **Reglas**

Descripción	La tabla reglas describe cuales son las reglas que tiene el sistema implementadas y guarda los valores que debe tomar una regla.					
Atributos						
Nombre	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	**Clases no válidas**
Fecha	Fecha en que es creada la regla	Cadena de caracteres	No	SI	Cadena de Caracteres	
Nombre regla	Es el nombre que recibe la regla por	Cadena de caracteres	Si	No	Cadena de Caracteres	

	parte del usuario					
Autorizar regla	Permite o deniega el flujo de información implementada en la regla	Booleano	No	No	Toma valor verdadero o falso	
Tipo flujo	Permite saber el tipo de flujo, si es entrante, saliente o ambos	Cadena de caracteres	No	No	Toman valor Entrante, Saliente o Entrante y saliente	
Protocolo	Es el protocolo utilizado en la regla	Cadena de caracteres	Si	Si	TCP, UDP	
Direcciones IP	Las direcciones que se pueden denegar o permitir en la regla	Cadena de caracteres	Si	Si	Direcciones IP en formato IPv4	

Tabla 1. Descripción de Regla

- ✓ **Firewall**

Descripción	Describe la entidad en sí, y engloba todo lo que se hace en el negocio				
Atributos					
Nombre	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**

					Clases válidas	Clases no válidas
Activo	Dice si el firewall está habilitado o no.	Booleano	No	Si	Verdadero o falso	

Tabla 2. Descripción de Firewall

- ✓ **Registro cambios**

Descripción	Describe la entidad en sí, y engloba todo lo que se hace en el negocio					
Atributos						
Nombre	**Descripción**	**Tipo**	**¿Puede ser nulo?**	**¿Es único?**	**Restricciones**	
					Clases válidas	**Clases no válidas**
Fecha	Fecha en que es creada la regla	Cadena de caracteres	No	SI	Cadena de Caracteres	
Nombre de la regla	Lo que genero ese registro	Cadena de caracteres	Si	No	Cadena de Caracteres	
Componente	Lo que genero ese registro	Cadena de caracteres	No	No	Toma valor verdadero o falso	
Detalles	Permite saber qué fue lo que genero el registro	Cadena de caracteres	No	No	Cadena de caracteres	

Tabla 3. Descripción de Registro de cambios

2.3 Requisitos funcionales

Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En un sistema describen lo que el sistema debe hacer. Estos requerimientos dependen del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque general tomado por la organización al redactar requerimientos.(30) A continuación, se listan los **requisitos funcionales** que se identificaron:

RF1. Mostrar las reglas iptables configuradas.

RF2. Mostrar el estado del firewall.

RF3. Mostrar la fecha en que se creó la regla.

RF4. Deshabilitar el firewall de Segurmática antivirus.

RF5. Habilitar el firewall de Segurmática antivirus.

RF6. Modificar reglas del sistema firewall de Segurmática antivirus

RF7. Añadir reglas al sistema firewall de Segurmática antivirus.

RF8. Eliminar reglas del sistema firewall de Segurmática antivirus.

RF9. Exportar reglas iptables configuradas

RF10. Acceder al firewall desde la bandeja del sistema.

2.4 Descripción de requisitos no funcionales

Los requerimientos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares, a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema.

Los requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De forma alternativa, definen las restricciones del sistema como la capacidad de los dispositivos de entrada/salida y las representaciones de datos que se utilizan en las interfaces del sistema.(30)

Atributo de Calidad	Usabilidad
Sub-atributos/Sub-características	Comprensibilidad
Objetivo	Facilitar de uso del sistema: el sistema debe presentar una interfaz amigable que permita la fácil interacción y llegar de manera rápida y

	efectiva a la información buscada.
Origen	El usuario
Artefacto	El sistema
Entorno	El sistema está funcionando correctamente
Estímulo	**Respuesta: Flujo de eventos (Escenarios)**
NA	NA
Medida de respuesta	
NA	

Tabla 4. RNF1. Facilidad de uso del sistema

Atributo de Calidad	Soporte
Sub-atributos/Sub-características	Integralidad
Objetivo	Compilar de forma integrable al sistema Segurmática antivirus: el sistema debe permitir una fácil integración con el sistema de Segurmática antivirus
Origen	El usuario
Artefacto	El sistema
Entorno	El sistema está funcionando correctamente
Estímulo	**Respuesta: Flujo de eventos (Escenarios)**
NA	NA
Medida de respuesta	
NA	

Tabla 5. RNF2. Compilar de forma integral el sistema

Atributo de Calidad	Eficiencia
Sub-atributos/Sub-características	Rapidez
Objetivo	Obtener una respuesta rápida
Origen	El usuario
Artefacto	El sistema
Entorno	El sistema está funcionando correctamente

Estímulo	**Respuesta: Flujo de eventos (Escenarios)**
NA	NA
Medida de respuesta	
El sistema debe brindar un tiempo de respuesta menor o igual de 5 segundos	

Tabla 6.RNF3. Respuesta rápida

Atributo de Calidad	Interfaz
Sub-atributos/Sub-características	Diseño
Objetivo	Realizar la identificación con el negocio: empleo de imágenes y colores identificados con el negocio donde se incluirá.
Origen	El usuario
Artefacto	El sistema
Entorno	El sistema está funcionando correctamente
Estímulo	**Respuesta: Flujo de eventos (Escenarios)**
NA	NA
Medida de respuesta	
NA	

Tabla 7. RNF4. Identificación con el negocio

Atributo de Calidad	Interoperabilidad
Sub-atributos/Sub-características	Compatibilidad
Objetivo	Tener instalado en su PC el Ubuntu 16.04, debían 8 , nova 5.0 u openSUSE desde estas versiones en adelante.
Origen	El usuario
Artefacto	El sistema
Entorno	El sistema está funcionando correctamente
Estímulo	**Respuesta: Flujo de eventos (Escenarios)**
NA	NA
Medida de respuesta	
NA	

Tabla 8. RNF5 Características del software

Atributo de Calidad	Funcionalidad
Sub-atributos/Sub-características	
Objetivo	Poseer 1GB de RAM como mínimo y un espacio en disco de 150 MB.
Origen	El usuario
Artefacto	El sistema
Entorno	El sistema está funcionando correctamente
Estímulo	**Respuesta: Flujo de eventos (Escenarios)**
NA	NA
Medida de respuesta	
NA	

Tabla 9. RNF6 Características del hardware

2.5 Diagrama de caso de uso del sistema

Un caso de uso es una herramienta que sirve para representar la forma como un cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en la cual, los elementos interactúan. La aplicación principal de los casos de uso es en el proceso de análisis y diseño, pero de manera particular en la definición de requerimientos del usuario.(31) A continuación, se muestra el diagrama de caso de uso del sistema:

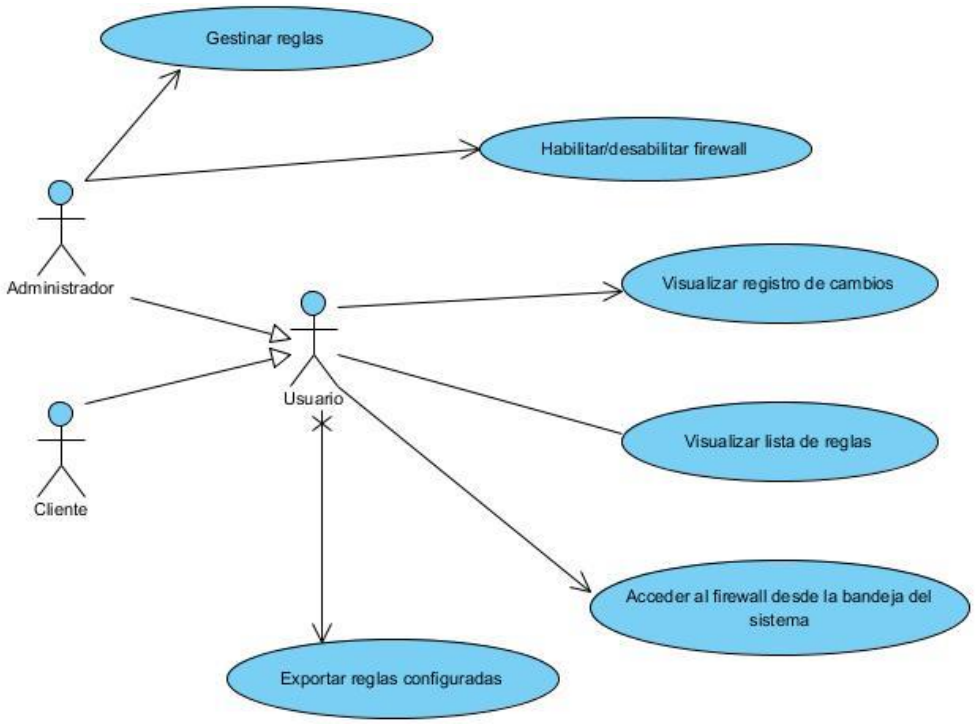


Figura 2. Diagrama de caso de uso del sistema

Descripción de actores

Actor	Objetivos
Administrador	Es quien interactúa con el firewall, configurándolo según le convenga.
Cliente	Visualiza la configuración del firewall

Tabla 10. Descripción de actores del sistema

Descripción de casos de uso del sistema

- ✓ **CU 1. Gestionar reglas**

Objetivo	El objetivo es adicionar, modificar o eliminar una regla al firewall	
Actores	Administrador	
Resumen	Consiste en permitir al usuario adicionar, modificar o eliminar una regla	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	El administrador se ha autentificado en el sistema	
Postcondiciones	Se ha modificado la lista de reglas. Se ha modificado el registro de cambios.	
Flujo de eventos		
Flujo básico : Gestionar listado de usuarios		
	Actor	**Sistema**
	Seleccionar en la interfaz la opción gestionar reglas	
		Se mostrará la interfaz que permite realizar varias acciones con una regla del listado: - Si desea añadir una nueva regla, ver Sección 1 “Adicionar regla”. - Si desea modificar datos de una regla, ver Sección 2 “Modificar regla”. - Si desea eliminar una regla, ver Sección 3 “Eliminar regla”.
	Selecciona una de las opciones que aparecen en el menú: Adicionar regla	

	Modificar regla Eliminar regla	
Sección 1: “Adicionar regla”		
Flujo básico : Adicionar regla		
	Actor	**Sistema**
1.	Selecciona la opción Adicionar	
2.		Brinda la posibilidad de introducir los datos de una regla Nombre Acción Tipo de Flujo Protocolo Tipo de Dirección Dirección y permite: Aceptar y se registra la regla Cancelar operación. Ver **Flujo Alterno 1:** “Cancelar operación.”
3.	Introduce los datos de la regla: Nombre Acción Tipo de Flujo Protocolo Tipo de Dirección Dirección Selecciona la opción Aceptar	

4.		Valida los datos. Si hay datos incompletos, ver **Flujo Alterno 2:** “Existen campos vacíos”. Si hay datos incorrectos, ver **Flujo Alterno 3:** “Existen campos incorrectos.”
5.		Registra la regla.

Sección 2: “Modificar regla”		
Flujo básico : Modificar regla		
	Actor	**Sistema**
1.	Selecciona la opción Modificar	
2.		Brinda la posibilidad de modificarla regla seleccionada: Nombre Acción Tipo de Flujo Protocolo Tipo de Dirección Dirección y permite: Aceptar y se modifican los datos de la regla. Cancelar operación. Ver **Flujo Alterno 1**: “Cancelar operación.”
3.	Modifica los datos del Usuario del listado	
4.		Valida los datos. Si hay datos incompletos, ver **Flujo Alterno 2:** “Existen campos vacíos”.

		Si hay datos incorrectos, ver **Flujo Alterno 3:** "Existen campos incorrectos."
5.		Actualiza los datos del Usuario
Sección 3: "Eliminar regla"		
Flujo básico : Eliminar regla		
	Actor	**Sistema**
1.	Selecciona la opción Eliminar	
2.		Muestra el mensaje de advertencia "¿Está seguro que desea eliminar la regla? " y permite: Aceptar Cancelar Cancelar la operación. Ver **Flujo Alterno 1:** "Cancelar operación."
3.	Selecciona Aceptar	
4.		Elimina la regla
Flujos alternos		
1.2, 2.2, 3.2: Cancelar operación		
	Actor	**Sistema**
1.	Selecciona la opción Cancelar.	Elimina todos los datos de entrada y regresa a la vista anterior.
2.		El caso de uso termina.
Flujos alternos		
1.3, 2.3: Existen campos vacíos		
	Actor	**Sistema**
1.		Muestra el mensaje de error "Existen campos vacíos"
2.		Muestra un indicador al lado de los campos vacíos.
Flujos alternos		

1.4, 2.4: Existen campos incorrectos		
	Actor	**Sistema**
1.		Muestra el mensaje de error “Existen campos incorrectos”
2.		Muestra un indicador sobre los campos incorrectos.

Prototipo de Interfaz gráfica de usuario gestionar reglas y adicionar regla

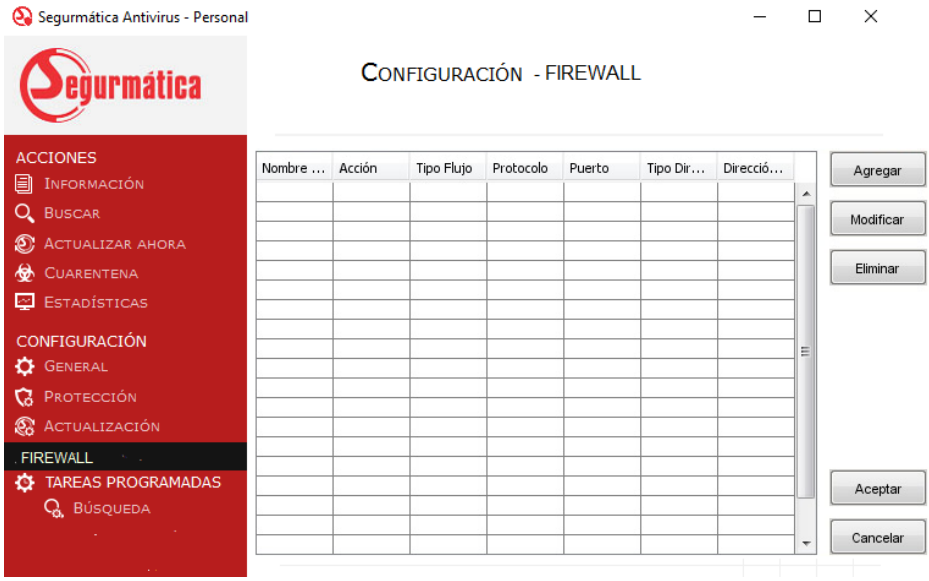


Tabla 11. Descripción del caso de uso gestionar reglas

✓ **CU 2. Visualizar lista de reglas**

Objetivo	El objetivo es mostrar al usuario todas las reglas implementadas	
Actores	Usuario	
Resumen	El usuario accede al sistema firewall. Se muestra las reglas implementadas.	
Complejidad	Media	
Prioridad	Critica	
Precondiciones	Haya sido configurada al menos una regla.	
Postcondiciones		
Flujo de eventos		
Flujo básico: Visualizar lista de reglas		
	Actor	**Sistema**
		Muestra el listado de reglas existentes.
Relaciones	**CU incluidos**	Gestionar. Ver CU gestionar reglas. Registro de cambios. Ver CU Visualizar Registros de Cambios.
Prototipo elemental de interfaz gráfica de usuario		

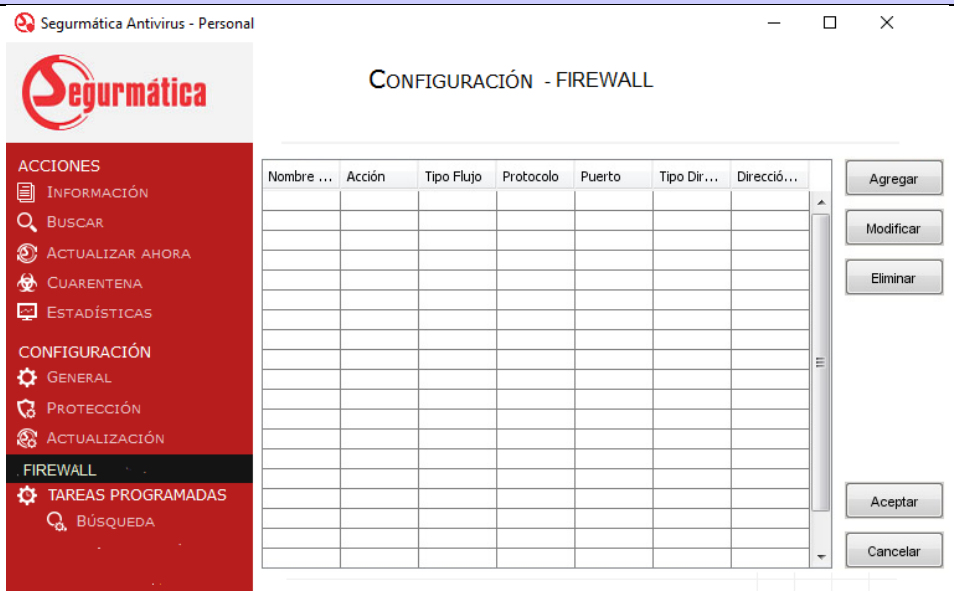


Tabla 12. Descripción de visualizar lista de reglas

✓ **CU 3. Visualizar Registros de Cambios**

Objetivo	El objetivo es mostrar al usuario todos los cambios
Actores	Usuario.
Resumen	El usuario hace clic en Registro de Cambios y se muestra todos los registros de

	las reglas
Complejidad	Media
Prioridad	Media
Precondiciones	Haya sido configurada al menos una regla.
Postcondiciones	
Flujo de eventos	
Flujo básico: Visualizar Registros de Cambios	

	Actor	**Sistema**
	Hacer clic en Registro de Cambios	
		De muestran los registros de Cambios de las reglas
	Seleccionar Aceptar	

Prototipo elemental de interfaz gráfica de usuario

Tabla 13. Descripción de Visualizar registro de cambios

✓ **CU 4. Habilitar/Deshabilitar firewall**

Objetivo	El objetivo es habilitar o deshabilitar el firewall
Actores	Administrador
Resumen	El usuario habilita o deshabilita el firewall por medio de un botón

<table>
<tr><td>Complejidad</td><td colspan="3">Media</td></tr>
<tr><td>Prioridad</td><td colspan="3">Media</td></tr>
<tr><td>Precondiciones</td><td colspan="3">El administrador se ha autentificado en el sistema</td></tr>
<tr><td>Postcondiciones</td><td colspan="3"></td></tr>
<tr><td colspan="4">Flujo de eventos</td></tr>
<tr><td colspan="4">Flujo básico: Visualizar Registros de Cambios</td></tr>
<tr><td></td><td colspan="2">Actor</td><td>Sistema</td></tr>
<tr><td></td><td colspan="2">Hacer clic en habilitar</td><td></td></tr>
<tr><td></td><td colspan="2"></td><td>Se muestran la lista de reglas</td></tr>
<tr><td>Relaciones</td><td>CU incluidos</td><td colspan="2">Gestionar. Ver CU gestionar reglas.
Registro de cambios. Ver CU Visualizar Registros de Cambios.</td></tr>
<tr><td colspan="4">Prototipo elemental de interfaz gráfica de usuario</td></tr>
<tr><td colspan="4"> 
</td></tr>
</table>

Tabla 14. Descripción de Habilitar/Deshabilitar firewall.

✓ **CU 5. Acceder al firewall desde la bandeja del sistema**

Objetivo	Acceder al firewall desde la bandeja del sistema
Actores	Administrador
Resumen	El usuario accede al firewall por medio de la bandeja del sistema
Complejidad	Media

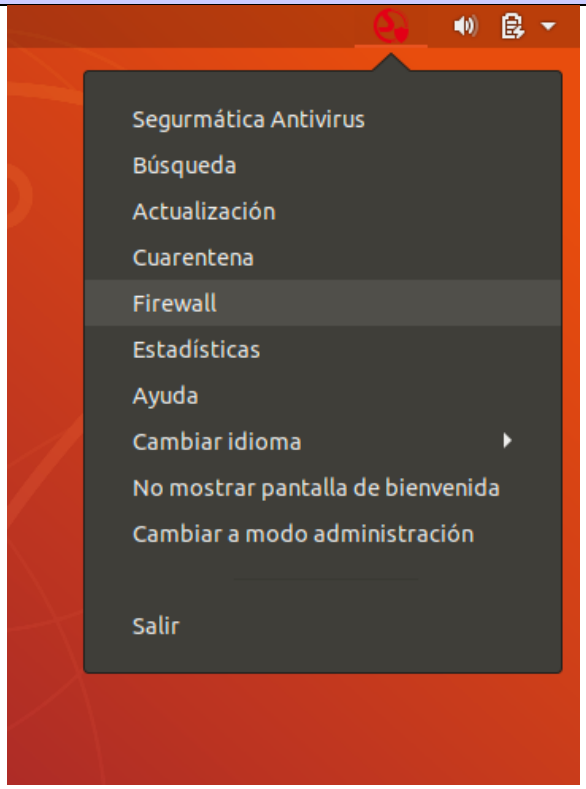
<table>
<tr><td colspan="2">Prioridad</td><td colspan="3">Media</td></tr>
<tr><td colspan="2">Precondiciones</td><td colspan="3">El administrador se ha autentificado en el sistema</td></tr>
<tr><td colspan="2">Postcondiciones</td><td colspan="3"></td></tr>
<tr><td colspan="5">Flujo de eventos</td></tr>
<tr><td colspan="5">Flujo básico: Visualizar Registros de Cambios</td></tr>
<tr><td></td><td colspan="2">Actor</td><td colspan="2">Sistema</td></tr>
<tr><td></td><td colspan="2">Hacer clic en la bandeja del sistema y hacer clic en la opción firewall.</td><td colspan="2"></td></tr>
<tr><td></td><td colspan="2"></td><td colspan="2">Se muestran el firewall</td></tr>
<tr><td colspan="2">Relaciones</td><td>CU incluidos</td><td colspan="2">Gestionar. Ver CU gestionar reglas.
Registro de cambios. Ver CU Visualizar Registros de Cambios.</td></tr>
<tr><td colspan="5">Prototipo elemental de interfaz gráfica de usuario</td></tr>
<tr><td colspan="5"> 
</td></tr>
</table>

Tabla 15. Descripción de Habilitar/Deshabilitar firewall.

- ✓ **CU 6. Exportar reglas configuradas**

Objetivo	El objetivo es exportar las reglas configuradas
Actores	Usuario.
Resumen	El usuario hace clic en salvar y se muestra una ventana emergente para guardar

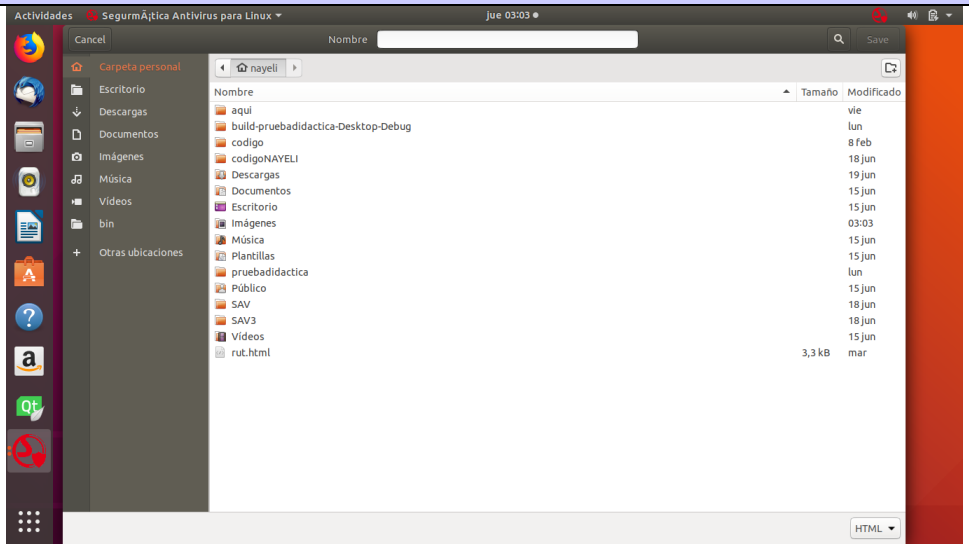
<table>
<tr><td colspan="2">Complejidad</td><td>Media</td></tr>
<tr><td colspan="2">Prioridad</td><td>Media</td></tr>
<tr><td colspan="2">Precondiciones</td><td></td></tr>
<tr><td colspan="2">Postcondiciones</td><td></td></tr>
<tr><td colspan="3">Flujo de eventos</td></tr>
<tr><td colspan="3">Flujo básico: Visualizar Registros de Cambios</td></tr>
<tr><td></td><td>Actor</td><td>Sistema</td></tr>
<tr><td></td><td>Hacer clic en Salvar</td><td></td></tr>
<tr><td></td><td></td><td>De muestran la una ventana emergente para salvar las reglas configuradas.</td></tr>
<tr><td></td><td>Seleccionar Aceptar</td><td></td></tr>
<tr><td colspan="2"></td><td></td></tr>
<tr><td colspan="2"></td><td></td></tr>
<tr><td colspan="3">Prototipo elemental de interfaz gráfica de usuario</td></tr>
<tr><td colspan="3"> 
</td></tr>
</table>

Tabla 16. Descripción de Visualizar registro de cambios

2.6 Diagrama de clases

El diagrama de clases proporciona una visión general del sistema de destino al describir los objetos y las clases dentro del sistema y las relaciones entre ellos. Proporciona una amplia variedad de usos; desde modelar la estructura de datos específica del dominio hasta el diseño detallado del sistema objetivo.(32) A continuación, se muestra el diagrama de clases referente a las entidades existentes en el sistema (anexo 2):

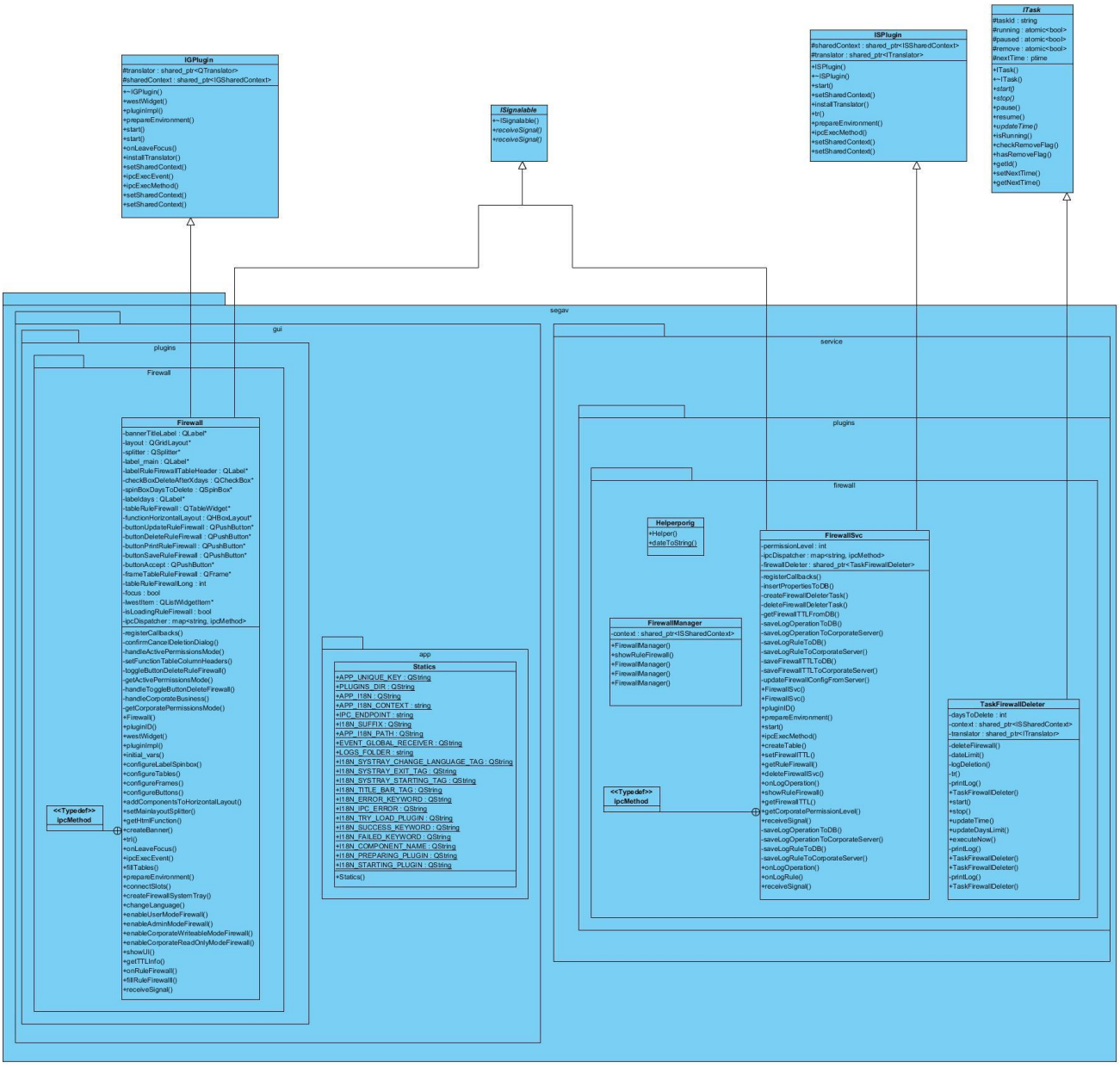


Figura 3. Diagrama de Clase

2.7 Modelo de base de datos

En el modelo de base de datos se determina la estructura lógica de una base de datos, el modo de almacenar, organizar y manipular los datos. Se utiliza para representar la base de datos de manera gráfica a través del diagrama de entidad relación. A continuación, se muestra el modelo de base de datos, donde se exponen los datos y la relación existente entre ellos para el desarrollo del sistema:

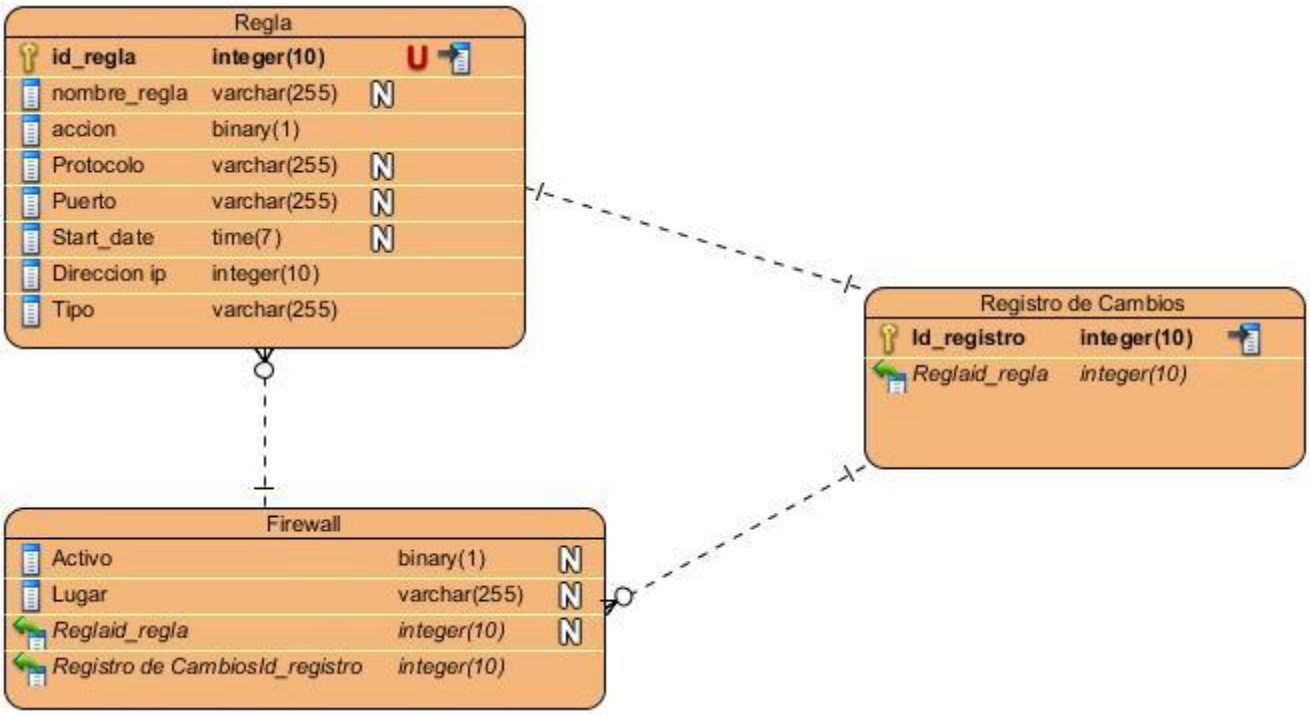


Figura 4. Diagrama de base de datos

2.8 Arquitectura

La arquitectura basada en plugins es la utilizada en Segurmática Antivirus donde la interfaz gráfica de usuario y el servicio a consumir están estructurado en forma de plugins, todo interno en la propia computadora.

SavUnix emplea un sistema de dependencias entre plugins, que no permite la carga de un plugins si las dependencias no fueron satisfechas. Esto proporciona robustez y la posibilidad de crear plugins que cooperan conjuntamente. Los plugins en SavUnix permiten comunicarse entre sí y define un mecanismo interno de comunicación entre plugins mediante IPC[14]. Un plugins puede solicitar un método de otro, para utilizar la información devuelta para su uso interno.

En vista de todos los datos anteriores, el enfoque adoptado consiste en desarrollar la aplicación como un conjunto de plugins que trabajan cooperativamente para proporcionar la funcionalidad deseada. Por tanto, el subsistema se divide en dos capas: una capa dedicada a la interfaz gráfica del sistema que se muestra al usuario, y otra dedicada a proveer los servicios del sistema. Con este diseño, se consigue que, para ampliar la aplicación con nuevas funcionalidades, bastará con añadir un plugins que diera soporte a esta. A continuación, se muestra la arquitectura a utilizar:

[14] En español " comunicación entre procesos ", en inglés: Inter Processes Communication (IPC).

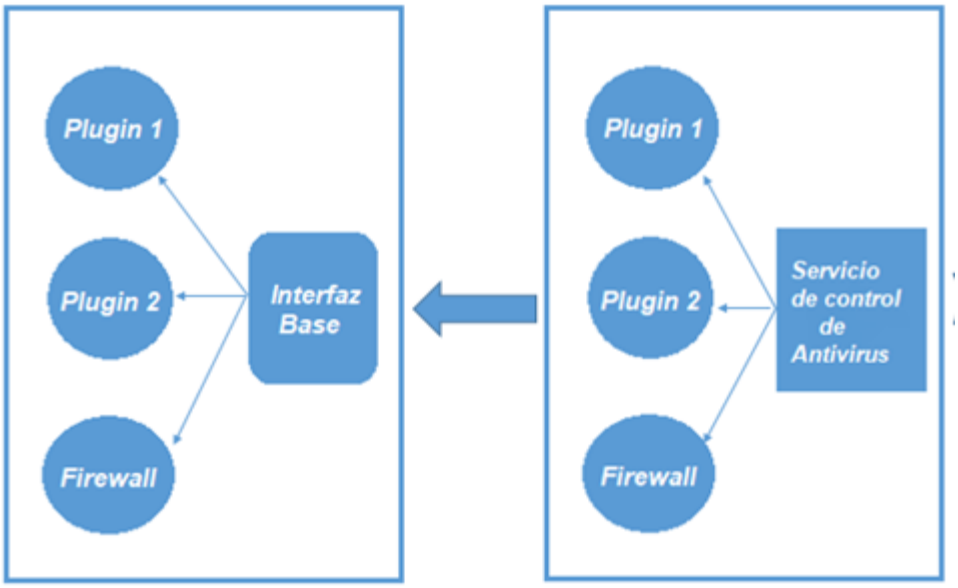


Figura 5.Arquitectura basada en plugins

Descripción de la estructura de la aplicación:

El proyecto se encuentra dividido en subproyectos, los cuales quedan de la siguiente forma:

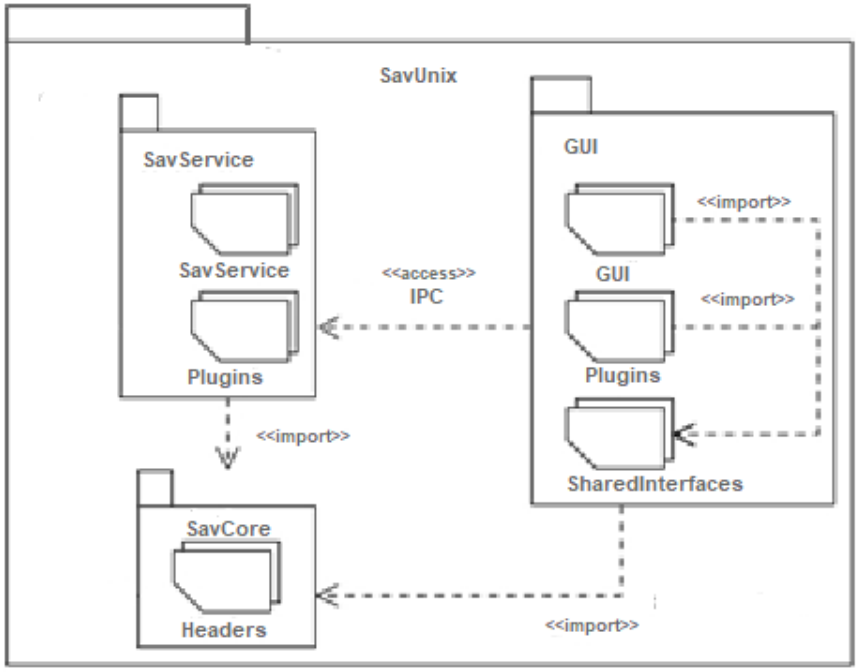


Figura 6. Estructura de la aplicación

- **GUI**: Subproyecto de interfaz gráfica de usuario.
  - GUI: Aplicación principal de la interfaz.
  - Plugins: Plugins de la interfaz.

- SharedInterfaces: Interfaces para la comunicación entre la aplicación principal y sus plugins.

- **SavService**: Subproyecto de servicio
  - SAVService: Aplicación principal del servicio antivirus.
  - Plugins: Plugins de los servicios del antivirus.
- **SavCore**: Subproyecto de componentes base comunes para el resto de los subproyectos. Contiene interfaces, implementaciones genéricas y modelos útiles. Se emplea durante la compilación de los otros subproyectos. Una vez compilados este subproyecto ya no es necesario y no forma parte de la aplicación compilada.

  La comunicación con SavService se realiza mediante IPC, una vez que la aplicación esta compilada. Cuando se compila el subproyecto SavCore deja de existir, ya que los componentes que requieren los otros subproyectos y que están contenidos en él se transfieren a estos al compilarlos.

2.9 Conclusiones del capítulo

En este capítulo se diseñó el marco teórico para lograr el entendimiento de los procesos que se llevan a cabo en la propuesta de solución, se identificaron como actores al cliente y el administrador que ambos son usuarios del sistema. Se Identificaron 10 requisitos funcionales y 6 requisitos no funcionales que se utilizan en la elaboración del módulo firewall para Segurmática antivirus en GNU/Linux, los cuales fueron representados mediante el diagrama de caso de uso con sus respectivas descripciones. Se procedió al diseño del diagrama de clases usando como base los casos de uso dentro del cual se destacó el gestionar regla por su gran importancia, además se definió la arquitectura del software, quedando las bases sentadas para avanzar al próximo flujo de trabajo implementación y prueba.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

Al finalizarse el proceso de análisis y diseño del firewall, se procederá a adentrarse en los temas de implementación y validación del módulo. Luego de identificar el modelo conceptual, los requisitos funcionales y no funcionales entre otros diagramas y modelos necesarios, se procede a desarrollar el módulo de firewall para el Antivirus Segurmática. Finalizada la fase de implementación, se someterá al sistema a pruebas de veracidad y factibilidad se las reglas configuradas. Esto permitirá validar la propuesta de solución planteada, así como verificar el correcto funcionamiento.

3.1 Patrones de Diseño

Los patrones GRASP constituyen un apoyo para la enseñanza que ayuda a uno a en tender el diseño de objetos esencial, y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Este enfoque para la comprensión y utilización de los principios de diseño se basa en los patrones de asignación de responsabilidades.

Los desarrolladores orientados a objetos con experiencia (y otros desarrolladores de software) acumulan un repertorio tanto de principios generales como de soluciones basadas en aplicar ciertos estilos que les guían en la creación de software. Estos principios y estilos, si se codifican con un formato estructurado que describa el problema y la solución, y se les da un nombre, podrían llamarse patrones.

Para utilizar los diferentes patrones hay que tener en cuenta que una asignación habilidosa de responsabilidades es extremadamente importante en el diseño de objetos. Además, decisión acerca de la asignación de responsabilidades, a menudo, tiene lugar durante la creación de los diagramas de interacción y con seguridad durante la programación. Los patrones son pares problema/solución con un nombre que codifican buenos consejos y principios relacionados con frecuencia con la asignación de responsabilidades.

Experto en Información (o Experto)

Solución: Asignar una responsabilidad al experto en información (la clase que tiene la información necesaria para realizar la responsabilidad).

Problema: ¿Cuál es un principio general para asignar responsabilidades a los objetos?

Un Modelo de Diseño podría definir cientos o miles de clases software, y una aplicación podría requerir que se realicen cientos o miles de responsabilidades. Durante el diseño de objetos, cuando se definen las interacciones entre los objetos, tomamos decisiones sobre la asignación de responsabilidades a las clases software. Si se hace bien, los sistemas tienden a ser más fáciles de entender, mantener y ampliar, y existen más oportunidades para reutilizar componentes en futuras aplicaciones.

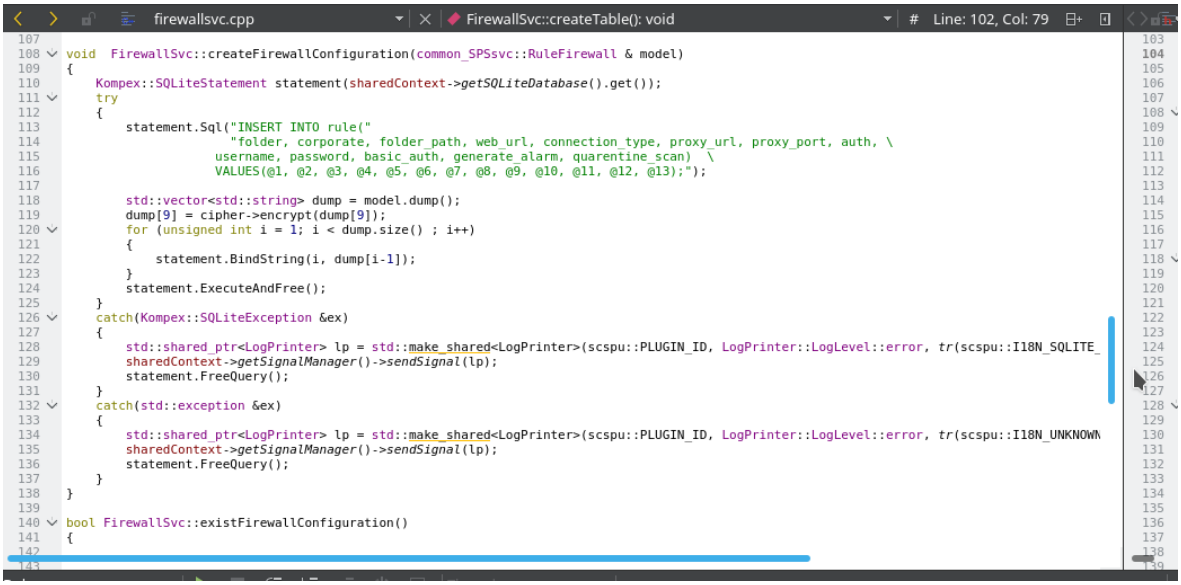


Figura 7.Patrón experto

Este patrón es utilizado para saber cuáles responsabilidades va a tener cada clase por ejemplo la creación de una nueva regla la realizará la clase firewallsvc.cpp ya que es la clase del servidor que está en relación con los atributos obtenidos de la clase interfaz firewall.cpp

Creador

Problema: ¿Quién debería ser el responsable de la creación de una nueva instancia de alguna clase?

La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia, es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien, el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización.

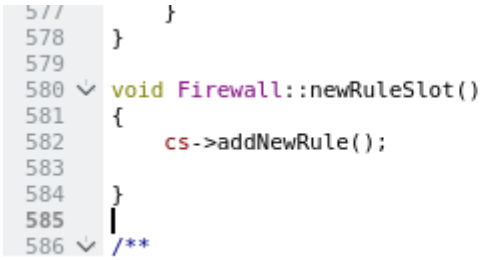


Figura 8.Patrón Creador

Se utilizó para crear la interfaz de las reglas iptables ya que es la clase firewall.cpp llama a una interfaz que recoge los datos de las reglas por lo que ella crea un objeto de la clase newrule.cpp para que ella los recopile.

Bajo Acoplamiento

Solución: Asignar una responsabilidad de manera que el acoplamiento permanezca bajo.

Problema: ¿Cómo soportar bajas dependencias, bajo impacto del cambio e incremento de la reutilización?

El acoplamiento es una medida de la fuerza con que un elemento está conectado a, tiene conocimiento de, confía en, otros elementos. Un elemento con bajo (o débil) acoplamiento no depende de demasiados otros elementos; "demasiados" depende del contexto, pero se estudiará. Estos elementos pueden ser clases, subsistemas, sistemas, etcétera.

Una clase con alto (o fuerte) acoplamiento confía en muchas otras clases. Tales clases podrían no ser deseables; algunas adolecen de los siguientes problemas:

- ✓ Los cambios en las clases relacionadas fuerzan cambios locales.
- ✓ Son difíciles de entender de manera aislada.
- ✓ Son difíciles de reutilizar puesto que su uso requiere la presencia adicional de las clases de las que depende.

Por ejemplo, la clase firewallsvc_shared.h se relaciona con un pequeño número de clases, ya que la mayoría de las clases están relacionadas con la clase firewall.cpp y a firewallsvc.cpp que es la que controla.

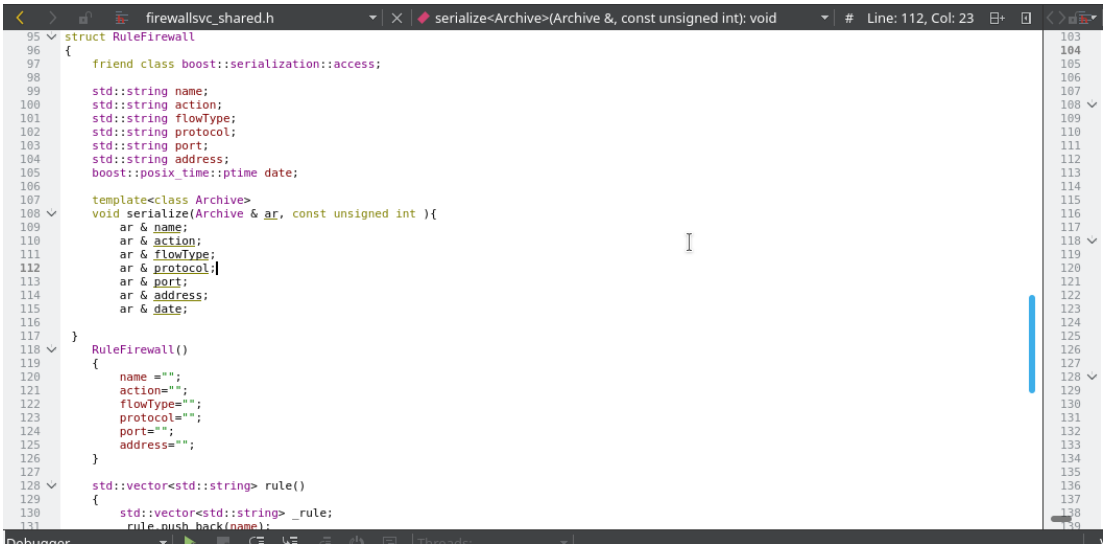


Figura 9.Patrón Bajo Acoplamiento

Alta Cohesión

Solución: Asignar una responsabilidad de manera que la cohesión permanezca alta.

Problema: ¿Cómo mantener la complejidad manejable?

En cuanto al diseño de objetos, la cohesión (o de manera más específica, la cohesión funcional) es una medida de la fuerza con la que se relacionan y del grado de focalización de las responsabilidades de un elemento. Un elemento con responsabilidades altamente relacionadas, y que no hace una gran cantidad de trabajo, tiene alta cohesión. Es tos elementos pueden ser clases, subsistemas, etcétera.

Una clase con baja cohesión hace muchas cosas no relacionadas, o hace demasiado trabajo. Tales clases

no son convenientes; adolecen de los siguientes problemas:

- ✓ Difíciles de entender.
- ✓ Difíciles de reutilizar.
- ✓ Difíciles de mantener.
- ✓ Delicadas, constantemente afectadas por los cambios.

Se aplica a la mayoría de las clases del diseño, ya que en cada una solo se implementan las funcionalidades que le corresponden.

Controlador

Solución: Asignar la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase que representa una de las siguientes opciones:

- ✓ Representa el sistema global, dispositivo o subsistema (controlador de fachada).
- ✓ Representa un escenario de caso de uso en el que tiene lugar el evento del sistema, a menudo denominado <NombreDelCasoDeUso>Manejador, <NombreDel CasoDeUso>Coordinadoro <NombreDelCasoDeUso>Sesion (controlador de sesión o de caso de uso).
- Utilice la misma clase controlador para todos los eventos del sistema en el mismo escenario de caso de uso.
- Informalmente, una sesión es una instancia de una conversación con un actor. Las sesiones pueden tener cualquier duración, pero se organizan a menudo en función de los casos de uso (sesiones de casos de uso).

Problema: ¿Quién debe ser el responsable de gestionar un evento de entrada al sistema?

Un evento del sistema de entrada es un evento generado por un actor externo. Se asocian con operaciones del sistema (operaciones del sistema como respuesta a los eventos del sistema), tal como se relacionan los mensajes y los métodos.(33)

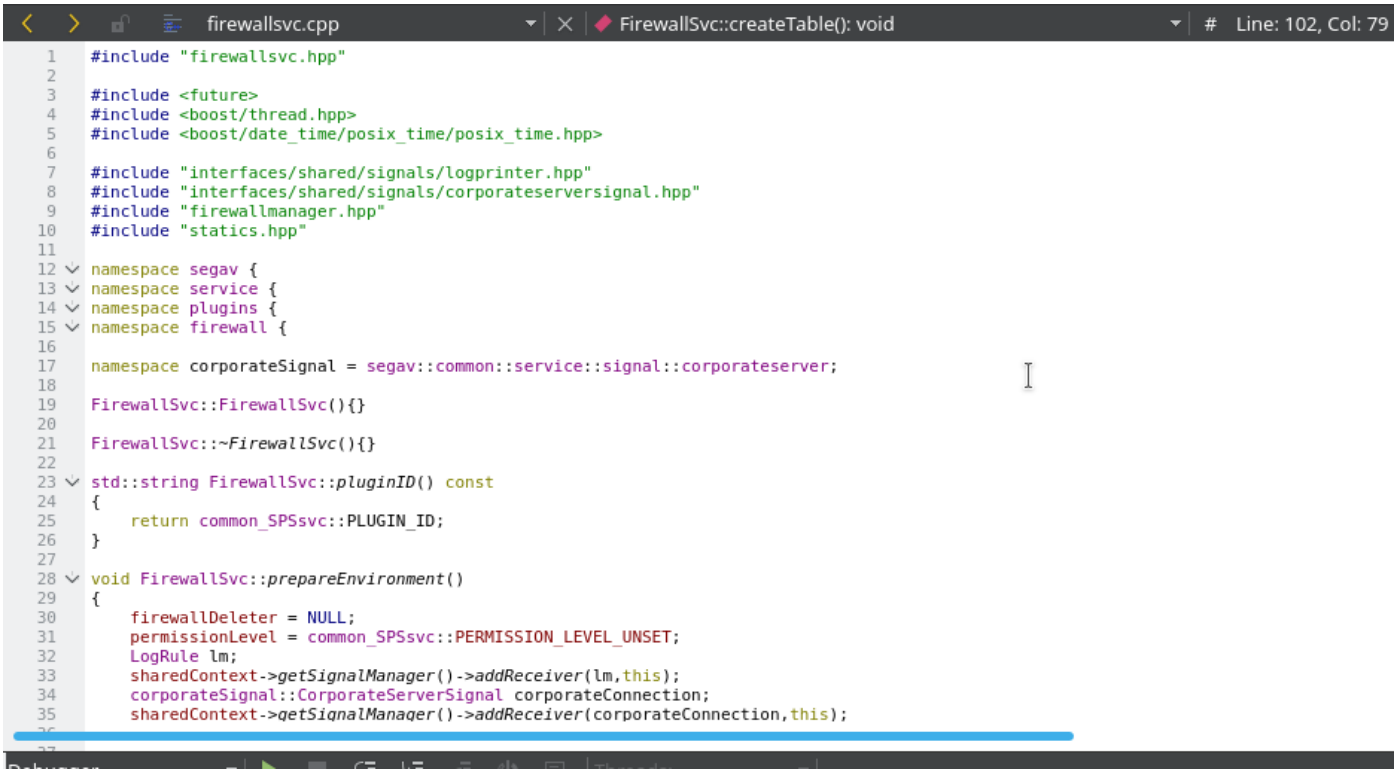


Figura 10.Patrón controlador

La clase firewallsvc.cpp es la clase controladora de la aplicación la que provee los servicios y guarda en la base de datos las diferentes configuraciones de las reglas.

3.2 Diagrama de despliegue

El diagrama de despliegue ayuda a modelar el aspecto físico de un sistema de software orientado a objetos. Modela la configuración del tiempo de ejecución en una vista estática y visualiza la distribución de componentes en una aplicación. En la mayoría de los casos, implica el modelado de las configuraciones de hardware junto con los componentes de software que perduraron.(34) A continuación, se muestra el diagrama de despliegue en el cual se describe el despliegue físico de la información generada por el programa de software en los componentes de hardware.

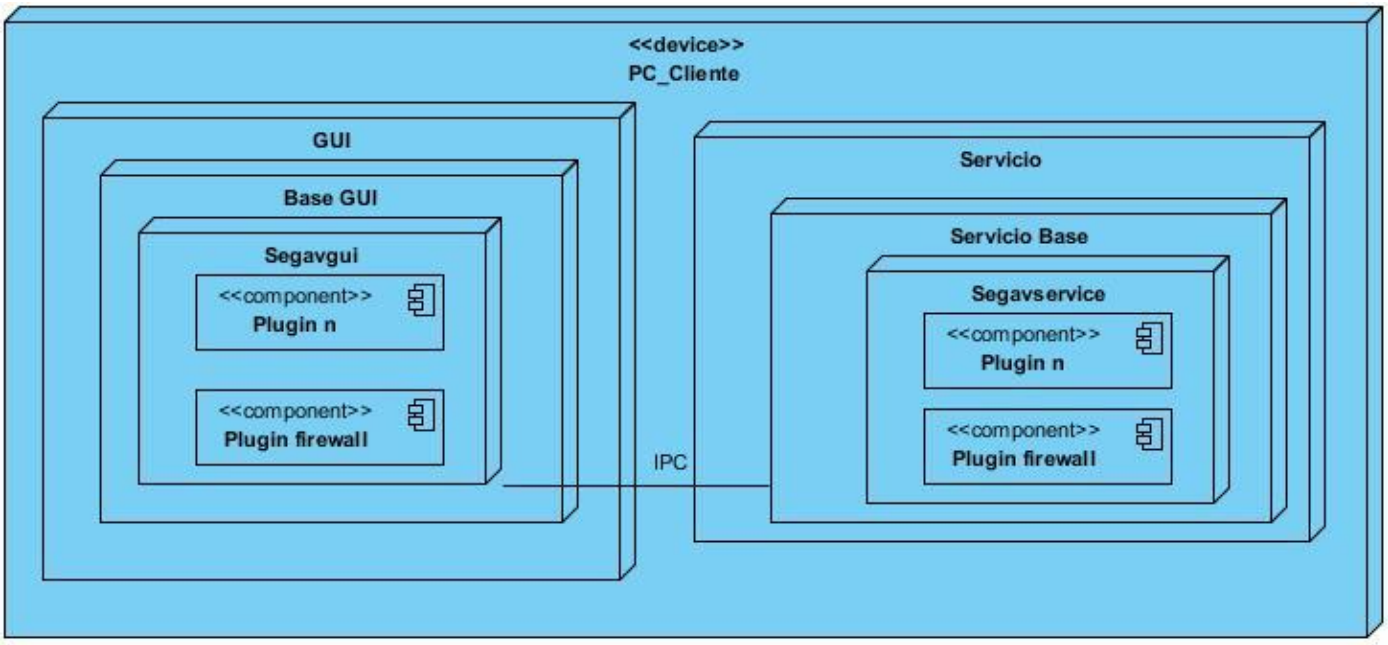


Figura 11. Diagrama de Despliegue

3.3 Casos de Pruebas

Para desarrollar productos software de calidad, la prueba es una de las tareas más importantes y, cuando se aplica linealmente en el ciclo de vida del producto, desempeña un papel crucial en la gestión de proyectos. Las pruebas evalúan el producto para determinar que cumple con el objetivo previsto, por lo que es necesario diseñar un plan de pruebas que se adapte y sea coherente con la metodología de desarrollo, que proporcione un enfoque de fácil acceso a la estructura para verificar los requisitos y cuantificar su rendimiento, y que identifique las diferencias entre los resultados previstos y los reales —errores o fallas—; es el proceso por medio del cual se evalúa la correcta interpretación y aplicación de los requisitos especificados. Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, se trata de demostrar que las funciones del software son operativas, que las entradas se manejan de forma adecuada y que se produce el resultado esperado. Las pruebas de caja blanca se centran en la estructura lógica interna del software. Se basan en un examen de los procedimientos y caminos lógicos del sistema. (36)

Método de pruebas

Se explica a continuación el método de prueba a emplear en la solución del componente:

Prueba de Caja Negra

Las Pruebas de Caja Negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software. Se enfoca solamente en las entradas y salidas del sistema, sin preocuparnos en tener conocimiento de la estructura interna del programa de software. Para obtener el detalle de cuáles

deben ser esas entradas y salidas, nos basamos en los requerimientos de software y especificaciones funcionales. No utilizan ninguna información interna de los componentes de software o sistemas que se van a probar, sino que consideran el comportamiento del software desde el punto de vista de un observador externo, es decir, tal y como lo “viven” los usuarios del sistema

Para desarrollar estas pruebas existen varias técnicas:

- Técnica de partición de equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica de análisis de valores límites: esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de causa-efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

La técnica a emplear para desarrollar las pruebas es la de partición de equivalencia. Esta técnica permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico.(37)

Casos de pruebas

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. Siempre es ejecutada como una unidad, desde el comienzo hasta el final.

Debe verificar:

- Si el producto satisface los requerimientos del usuario, tal y como se describe en las especificaciones de los requerimientos.
- Si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.(37)

Se describieron los casos de pruebas para cada caso de uso. A continuación, se presenta el caso de prueba para el caso de uso Gestionar regla, especificando el escenario, la descripción, las variables, la respuesta del sistema y el flujo central, así como los resultados obtenidos una vez ejecutado el caso de prueba y las condiciones que deben cumplirse para que este se ejecute. El resto de los casos de pruebas se encuentran anexados al final del documento **(Ver anexo 1).** Las celdas de la tabla contienen los valores V, I, NA. V indica válido, I indica inválido y NA, indica que no es necesario proporcionar un valor del dato ya que es irrelevante.

Caso de Prueba 1:

Descripción general:

- Permite gestionar las reglas

Condiciones de ejecución:

- El usuario debe estar autentificado en el sistema

SC1 Gestionar reglas

Escenario	Descripción	Nombre	Acción	Tipo de Flujo	Protocolo	Puerto	Dirección IP	Respuesta del sistema	Flujo central
EC 1.1 Adicionar regla iptables satisfactoriamente.	Se debe adicionar una nueva regla iptables al sistema.	V Regla1	V Permitir	V Entrante	V TCP	V 8080	V 10.0.0.1	Adiciona las reglas iptables al sistema.	1. Seleccionar en el menú la opción Adicionar regla. 2. Introduce los datos de la nueva regla. 3. Selecciona la opción Aceptar.
		V	V Denegar	V Saliente	V	V	V		
		V Puerto	V Permitir	V Entrante/Saliente	V UDP	V 2222	V		
		V Dener	V Denegar	V Saliente	V	V 8080	V 129.0.4.6-129.0.4.8		
		V Regla3	V Permitir	V Saliente	V TCP	V	V 10.0.9.4		
EC 1.2 Seleccionar la opción Cancelar.	El usuario cancela la opción de	NA	NA	NA	NA	NA	NA	El sistema muestra la lista de las	1. Seleccionar en el menú la opción

	adicionar regla iptables.							reglas iptables.	Nueva regla. 2. Presionar el botón Cancelar.
EC 1.3 Campos incorrectos.	Se valida que no existan campos incorrectos.	NA	NA	NA	NA	I 80.80	V 10.0.0.9	Muestra un mensaje de error: “Existen valores incorrectos”.	1. Seleccionar en el menú la opción Nueva regla. 2. Llenar los datos de forma incorrecta. 3. Presiona el botón Aceptar.
		NA	NA	NA	NA	2222	10.22		

Tabla 17. Caso de Prueba, SC 1 Adicionar regla

SC2 Eliminar regla

Escenario	Descripción	Nombre	Acción	Tipo de Flujo	Protocolo	Puerto	Dirección IP	Respuesta del sistema	Flujo central
EC 2.1 Eliminar regla configurada.	Se debe eliminar una regla iptables del sistema.	NA	NA	NA	NA	NA	NA	Muestra el mensaje: "Está seguro que desea eliminar la regla". Una vez aceptada la operación muestra el mensaje: "la regla ha sido eliminado satisfactoriamente".	1. Seleccionar de la lista de reglas configuradas la regla a eliminar. 2. Seleccionar en el menú la opción "eliminar regla". 3. Presionar el botón Aceptar para confirmar su eliminación.
EC 1.2 Seleccionar la opción Cancelar.	El administrador cancela la opción de	NA	NA	NA	NA	NA	NA	Muestra el mensaje "Está seguro que desea eliminar la regla".	1. Seleccionar de la lista de reglas

	eliminar una regla iptables.							Se cancela la eliminación. Muestra el listado de reglas configuradas.	configuradas la regla a eliminar. 2. Seleccionar en el menú la opción "eliminar regla". 3. Presionar el botón Cancelar para no eliminar la regla.

Tabla 18. Caso de Prueba Gestionar regla, SC 2 Eliminar regla.

SC 3 Modificar proveedor

Escenario	**Descripción**	**Nombre**	**Acción**	**Tipo de Flujo**	**Protocolo**	**Puerto**	**Dirección IP**	**Respuesta del sistema**	**Flujo central**
EC 3.1 Modificar reglas configuradas.	Se debe modificar una regla del sistema, con los datos correctos.	V Regla1	V Permitir	V Entrante	V TCP	V 8080	V 10.0.0.1	Modifica las reglas del sistema y muestra el mensaje: "La regla ha sido modificado satisfactoriamente".	1. Seleccionar en el menú la opción Nueva regla. 2. Seleccionar de la lista de reglas la opción Editar
		V	V Denegar	V Saliente	V	V	V		
		V Puerto	V Permitir	V Entrante/Saliente	V UDP	V 2222	V		

		V Dener	V Denegar	V Saliente	V	V 8080	V 129.0.4.6 - 129.0.4.8		correspondiente a la regla que se desea modificar. 3. Introducir los nuevos datos de la regla. 4. Presionar el botón Aceptar.
		V Regla3	V Permitir	V Saliente	V TCP	V	V 10.0.9.4		
EC 1.2 Seleccionar la opción Cancelar.	El usuario cancela la opción de modificar una regla.	NA	NA	NA	NA	NA	NA	El sistema muestra la lista de las reglas iptables.	1. Seleccionar en el menú la opción Nueva regla 2. Seleccionar de la lista de proveedores la opción Editar correspondiente al proveedor que se desea modificar. 4. Presionar el botón Cancelar.

EC 1.3 Campos incorrectos.	Se valida que no existan campos incorrectos.	NA	NA	NA	NA	I 80.80	V 10.0.0.9	Muestra un mensaje de error: "Existen valores incorrectos".	1. Seleccionar en el menú la opción Nueva regla. 2. Llenar los datos de forma incorrecta. 3. Presiona el botón Aceptar.
		NA	NA	NA	NA	2222	10.22		

Tabla 19. Caso de Prueba, SC 3 Modificar regla

SC 4 Listar reglas

Escenario	**Descripción**	**Nombre**	**Acción**	**Tipo de Flujo**	**Protocolo**	**Puerto**	**Dirección IP**	**Respuesta del sistema**	**Flujo central**
EC 4.1 Listar las reglas iptables.	Se debe mostrar una lista con las reglas configuradas.	V	V	V	V	V	V	Muestra los datos de las reglas guardadas en el sistema.	1. Se muestra al habilitar el firewall

Tabla 20. Caso de Prueba, SC 3 Listar reglas

Descripción de las variables

No	**Nombre de campo**	**Clasificación**	**Valor nulo**	**Descripción**
1	Nombre	Campo de texto	Si	Solo admite letras y números
2	Acción	Campo de texto	No	Admite letras
3	Tipo de Flujo	Campo de texto	No	Solo admite letras
4	Protocolo	Campo de texto	Si	Solo admite números

5	Puerto	Campo de texto	Si	Solo admite números
6	Dirección IP	Campo de texto	Si	Solo admite números y puntos

Tabla 21. Descripción de las variables

Resultado de las pruebas

A continuación, se muestran los resultados de las pruebas de caja negra que fueron realizadas al componente para el módulo Firewall, en las que se puede observar el número de iteraciones que se le aplicaron a los casos de pruebas, el total de no conformidades encontradas y las que fueron resueltas, así como la cantidad que no procedieron.

No. de iteraciones	Total de no conformidades	Resueltas
1ra iteración	10	8
2da iteración	3	3
3ra iteración	0	0

Tabla 22. Resultados de las pruebas de Caja Negra aplicadas.

Prueba de Caja Blanca

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplican al software, logrando como resultado que disminuya en un gran porciento el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.(37)

Casos de pruebas

La aplicación que se va a utilizar para realizar las pruebas es CppUnit, que es el puerto C ++ del famoso marco JUnit para pruebas unitarias. El resultado de la prueba está en formato XML o texto para pruebas automáticas y GUI basado en pruebas supervisadas. Se basa en la jerarquía de un conjunto de pruebas que comprende casos de prueba unitaria que prueban funciones de clase.

Se realizaron 3 iteraciones para un total de 10 pruebas que se llevaron a cabo de forma satisfactoria de las cuales se obtuvieron los siguientes resultados:

Primera iteración

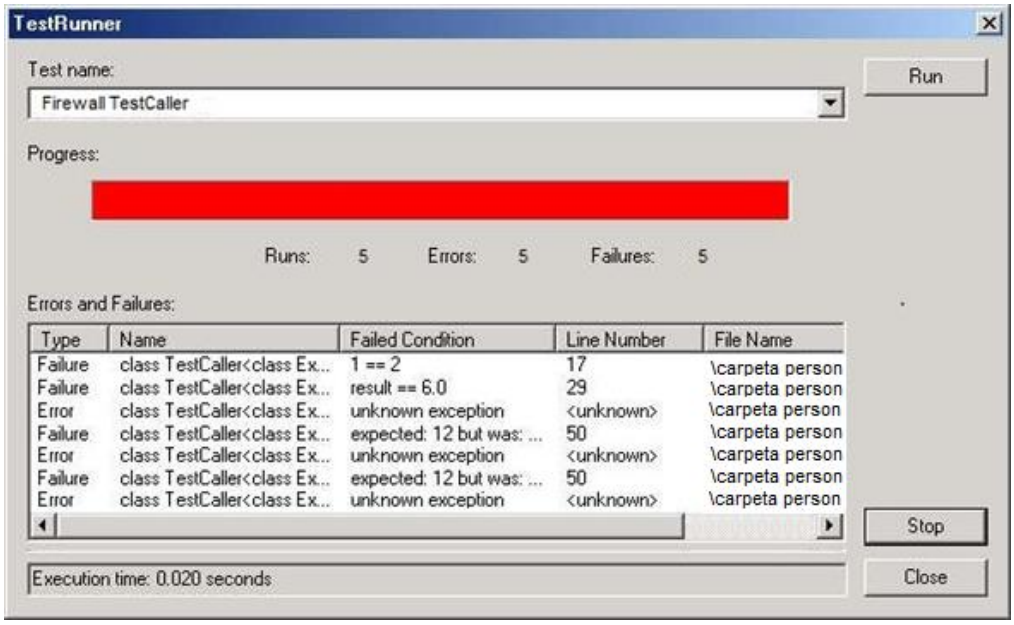


Figura 12. Primera Iteración en CppUnit

Se realizaron 5 pruebas donde los resultados fueron que había 5 errores y 5 fallos en el código de los cuales todos se solucionaron satisfactoriamente.

2da Iteración

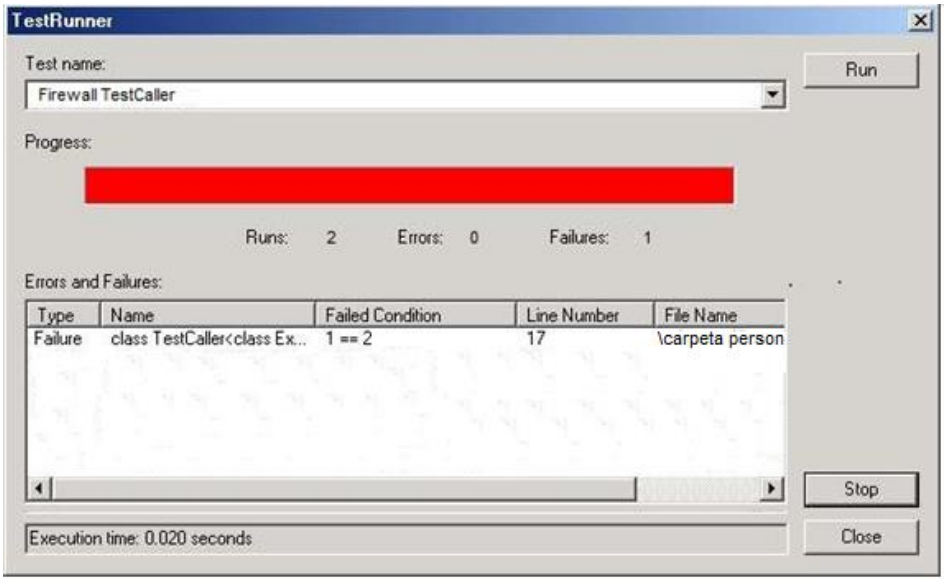


Figura 13. Segunda Iteración en CppUnit

Se realizaron 2 pruebas donde los resultados fueron que había 5 fallos y ningún error en el código de los cuales todos se solucionaron satisfactoriamente.

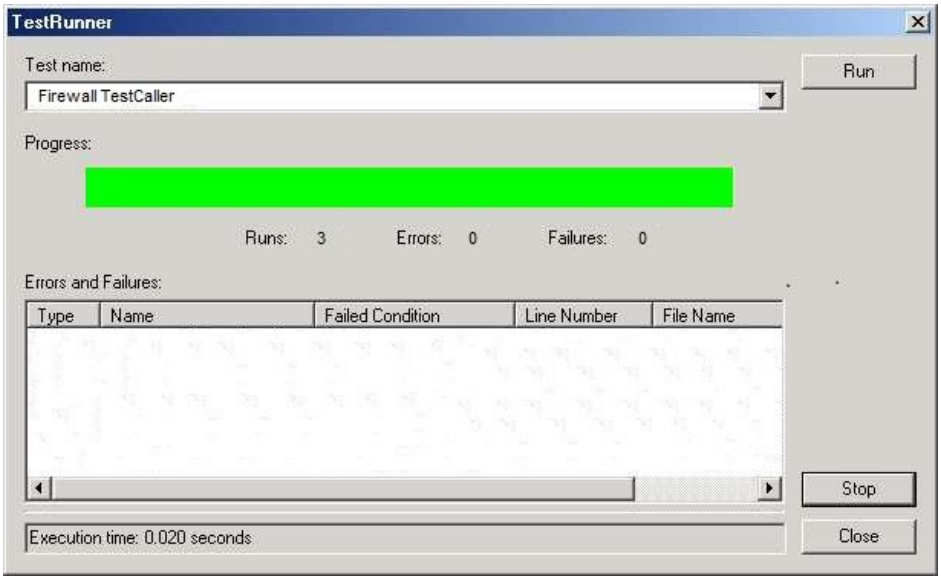


Figura 14. Tercera Iteración en CppUnit

Se realizaron 3 pruebas donde los resultados fueron satisfactorios y no hubo ningún error o fallo.

3.4 Conclusiones del capítulo

En este capítulo se realizaron las pruebas tanto de caja negra se realizaron 3 iteraciones donde se obtuvieron varias inconformidades de las cuales todas fueron resueltas, se efectuaron las pruebas de caja blanca obteniéndose resultados satisfactorios y se expusieron los diferentes patrones de diseño utilizados.

Conclusiones

La presente investigación se centró en el desarrollo de un componente para el módulo firewall para el Segurmática antivirus en GNU/Linux, con la finalidad de facilitar el proceso de gestión de las reglas iptables que no posee el Segurmática antivirus. Durante el avance se cumplieron los objetivos propuestos y se arribó a las siguientes conclusiones:

- A partir de la elaboración del marco teórico de la investigación se enriqueció el conocimiento acerca del módulo firewall para Segurmática antivirus en GNU/Linux.
- El estudio realizado como parte de la investigación sirvió de apoyo a la toma de decisiones con vista al desarrollo eficiente del módulo firewall para Segurmática antivirus en GNU/Linux.
- Se implementó el módulo a partir de los requerimientos de software establecidos, haciendo uso de la arquitectura y las tecnologías definidas por el equipo de desarrollo de software de dicho sistema.
- Se diseñaron y aplicaron las pruebas definidas, probando cada una de las funcionalidades del módulo para verificar su correcto funcionamiento.

Referencias

1. CEBRIÁN, María R. Sahuquillo, Belén Domínguez. Un potente ciberataque afecta a grandes empresas de todo el mundo. In: *EL PAÍS* [online]. 28 junio 2017. [Accessed 1 febrero 2018]. Available from: https://elpais.com/internacional/2017/06/27/actualidad/1498568187_011218.html.
2. CISCO. ¿Qué es un firewall? - Cisco. In: [online]. [Accessed 9 mayo 2018]. Available from: https://www.cisco.com/c/es_mx/products/security/firewalls/what-is-a-firewall.html.
3. EMPRESA SEGURMÁTICA. segav – Empresa Segurmática. In: [online]. [Accessed 27 junio 2018]. Available from: https://segurmatica.cubava.cu/tag/segav/.
4. UCI. Productos | Universidad de las Ciencias Informáticas. In: [online]. [Accessed 9 mayo 2018]. Available from: http://www.uci.cu/investigacion-y-desarrollo/productos.
5. CCM. Introducción a la seguridad informática. In: *CCM* [online]. [Accessed 27 junio 2018]. Available from: https://es.ccm.net/contents/622-introduccion-a-la-seguridad-informatica.
6. CCM. «Firewall». In: [online]. [Accessed 9 mayo 2018]. Available from: https://es.ccm.net/contents/590-firewall.
7. ¿Qué es un Firewall personal? : Área de Sistemas de la Información y las Comunicaciones : UPV. In: [online]. [Accessed 1 febrero 2018]. Available from: http://www.upv.es/entidades/ASIC/seguridad/353811normalc.html.
8. ZONEALARM. Software antivirus y de firewall por ZoneAlarm - Protección contra virus para su PC. In: *ZoneAlarm* [online]. [Accessed 20 junio 2018]. Available from: https://www.zonealarm.com/es/software/antivirus-firewall/.
9. COMODO. Free Internet Security | Why Comodo Internet Security Suite for PC? In: *comodo.com* [online]. [Accessed 20 junio 2018]. Available from: https://www.comodo.com/home/internet-security/free-internet-security.php.
10. COMODO. Antivirus for Linux | Free Linux Antivirus for Virus Protection. In: *comodo.com* [online]. [Accessed 20 junio 2018]. Available from: https://www.comodo.com/home/internet-security/antivirus-for-linux.php.
11. ESET NOD32. ESET NOD32 antivirus 4 para Linux | ESET. In: *ESET NOD32* [online]. [Accessed 20 junio 2018]. Available from: https://www.eset.es/productos/nod32-antivirus-linux/#.
12. PANDA SECURITY. La Revolución Antivirus - Panda Security. In: *pandasecurity.com* [online]. [Accessed 21 junio 2018]. Available from: /usa-es/.

13. CABRERA V., César y NARVÁEZ, Paola. FIREWALLS y seguridad en internet mediante IPCcoop. In: [online]. 2006, [Accessed 4 diciembre 2017]. Available from: http://dspace.uazuay.edu.ec:8080/handle/datos/2251.

14. SHOREWALL. Introduction. In: [online]. [Accessed 9 mayo 2018]. Available from: http://shorewall.org/Introduction.html.

15. UBUNTU. UncomplicatedFirewall - Ubuntu Wiki. In: [online]. [Accessed 7 diciembre 2017]. Available from: https://wiki.ubuntu.com/UncomplicatedFirewall.

16. VUURMUUR. Vuurmuur Firewall. In: [online]. [Accessed 7 diciembre 2017]. Available from: https://www.vuurmuur.org/trac/.

17. PFSENSE®. pfSense® - World's Most Trusted Open Source Firewall. In: [online]. [Accessed 11 diciembre 2017]. Available from: https://www.pfsense.org/.

18. NETFILTER. netfilter/iptables project homepage - The netfilter.org project. In: [online]. [Accessed 27 junio 2018]. Available from: https://www.netfilter.org/.

19. NETFILTER. netfilter/iptables project homepage - The netfilter.org «iptables» project. In: [online]. [Accessed 27 junio 2018]. Available from: https://www.netfilter.org/projects/iptables/index.html.

20. SOMMERVILLE, Ian. *Ingeniería de Software* [online]. Séptima edición. Madrid: Pearson Educacion, 2005. [Accessed 9 mayo 2018]. ISBN 84-7829-074-5. Available from: http://zeus.inf.ucv.cl/~bcrawford/AULA_ICI_3242/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20Ian%20Sommerville.pdf.

21. RODRÍGUEZ SÁNCHEZ, Tamara. *Metodología de desarrollo para la Actividad productiva de la UCI.* Carretera a San Antonio Km 2 1⁄2 . Torrens. Boyeros. La Habana. Cuba. UCI, 2015.

22. IBM SOFTWARE GROUP. *Introduction to UML 2* [online]. S.l.: s.n., 2005. Available from: http://www.omg.org/news/meetings/workshops/MDA-SOA-WS_Manual/00-T4_Matthews.pdf.

23. VISUAL PARADIGM. *Software Desing Tools for Agile Teams, whit UML, BPMN and More* [online]. S.l.: s.n., 2015. [Accessed 11 diciembre 2017]. Available from: http://www.visual-paradigm.com/.

24. GOMEZ, Ruiz. *Introducción al lenguaje C/C++* [online]. Málaga: s.n., 2013. [Accessed 5 enero 2018]. Available from: http://informaticagamarra.weebly.com/.

25. COMESAÑA CABEZA, José Luis. *Instalación y uso de los entornos de desarrollo* [online]. S.l.: s.n., 2012. Available from: http://docplayer.es/3976439-Instalacion-y-uso-de-entornos-de-desarrollo.html.

26. THE QT COMPANY. Application Development with Qt | Qt. In: [online]. [Accessed 9 mayo 2018]. Available from: https://www.qt.io/qt-for-application-development/.

27. THE QT COMPANY. Qt Creator Manual. In: [online]. [Accessed 9 mayo 2018]. Available from: http://doc.qt.io/qtcreator/index.html.

28. SQLITE. About SQLite. In: [online]. 2014. [Accessed 11 diciembre 2017]. Available from: https://www.sqlite.org/about.html.

29. Modelos conceptuales - Trabajos finales. In: [online]. [Accessed 9 mayo 2018]. Available from: http://www.buenastareas.com/ensayos/Modelos-Conceptuales/23455.html.

30. Requerimientos funcionales y no funcionales. In: [online]. [Accessed 9 mayo 2018]. Available from: https://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales.

31. VISCENCIO, Escrito por Isidro Leos. Casos de Uso - Introducción. In: [online]. [Accessed 9 mayo 2018]. Available from: http://softtlan.blogspot.com/2007/01/casos-de-uso-introduccin.html.

32. UML MODELING. Class Diagram - UML Diagrams - Unified Modeling Language Tool. In: *UML Modeling - Unified Modeling Language Tool.* [online]. [Accessed 9 mayo 2018]. Available from: https://www.visual-paradigm.com/VPGallery/diagrams/Class.html.

33. LARMAN, Craig. *UML y Patrones. Una introducción al análisis orientado a objetos y al proceso unificado*. Segunda edición. S.l.: s.n., [no date]. ISBN 978-970-17-0261-1.

34. UML MODELING. Deployment Diagram - UML 2 Diagrams - UML Modeling Tool. In: *UML Modeling - Unified Modeling Language Tool.* [online]. [Accessed 26 febrero 2018]. Available from: https://www.visual-paradigm.com/VPGallery/diagrams/Deployment.html.

35. UML Modeling - Unified Modeling Language Tool. In: [online]. [Accessed 26 febrero 2018]. Available from: https://www.visual-paradigm.com/VPGallery/diagrams/index.html.

36. ARISTEGUI O., José Luis. TEST CASES IN SOFTWARE TEST, LOS CASOS DE PRUEBA EN LA PRUEBA DEL SOFTWARE. In: . 2010, no. No.3, pp. 27-34.

37. PRESSMAN, Reger S. *Ingeniería de software. Un enfoque práctico* [online]. Séptima edición. S.l.: s.n., 2010. ISBN 978-607-15-0314-5. Available from: http://cotana.informatica.edu.bo/downloads/ld-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF. 805

BIBLIOGRAFÍA

CCM. «Firewall». In: [online]. [Accessed 9 mayo 2018]. Available from: https://es.ccm.net/contents/590-firewall.

CISCO. ¿Qué es un firewall? - Cisco. In: [online]. [Accessed 9 mayo 2018]. Available from: https://www.cisco.com/c/es_mx/products/security/firewalls/what-is-a-firewall.html.

LARMAN, Craig. *UML y Patrones. Una introducción al análisis orientado a objetos y al proceso unificado*. Segunda edición. S.l.: s.n., [no date]. ISBN 978-970-17-0261-1.

NETFILTER. netfilter/iptables project homepage - The netfilter.org «iptables» project. In: [online]. [Accessed 27 junio 2018]. Available from: https://www.netfilter.org/projects/iptables/index.html.

NETFILTER. netfilter/iptables project homepage - The netfilter.org project. In: [online]. [Accessed 27 junio 2018]. Available from: https://www.netfilter.org/.

SHOREWALL. Introduction. In: [online]. [Accessed 9 mayo 2018]. Available from: http://shorewall.org/Introduction.html.

Introduction [online]. S.l.: s.n. [Accessed 9 mayo 2018]. Available from: http://shorewall.org/Introduction.html.

SLIDESHAE. Modelos de Datos y Modelado Conceptual. In: [online]. [Accessed 9 mayo 2018]. Available from: https://es.slideshare.net/amruiz/modelos-de-datos-y-modelado-conceptual-presentation-663634.

SQLITE. About SQLite. In: [online]. 2014. [Accessed 11 diciembre 2017]. Available from: https://www.sqlite.org/about.html.

THE QT COMPANY. Application Development with Qt | Qt. In: [online]. [Accessed 9 mayo 2018]. Available from: https://www.qt.io/qt-for-application-development/.

THE QT COMPANY. Qt Creator Manual. In: [online]. [Accessed 9 mayo 2018]. Available from: http://doc.qt.io/qtcreator/index.html.

UBUNTU. UncomplicatedFirewall - Ubuntu Wiki. In: [online]. [Accessed 7 diciembre 2017]. Available from: https://wiki.ubuntu.com/UncomplicatedFirewall.

UCI. Productos | Universidad de las Ciencias Informáticas. In: [online]. [Accessed 9 mayo 2018]. Available from: http://www.uci.cu/investigacion-y-desarrollo/productos.

VUURMUUR. Vuurmuur Firewall. In: [online]. [Accessed 7 diciembre 2017]. Available from: https://www.vuurmuur.org/trac/.

¿Qué es un Firewall personal? : Área de Sistemas de la Información y las Comunicaciones : UPV. In: [online]. [Accessed 1 febrero 2018]. Available from: http://www.upv.es/entidades/ASIC/seguridad/353811normalc.html.
About Bitdefender: 17 Years of Innovation in CyberSecurity. In: [online]. [Accessed 21 junio 2018]. Available from: https://www.bitdefender.com/company/.
About SQLite. In: [online]. [Accessed 1 febrero 2018]. Available from: https://www.sqlite.org/about.html.
Plugins | Complementos y temas para Firefox & Thunderbird. In: [online]. [Accessed 8 mayo 2018]. Available from: https://addons.firefoxmania.uci.cu/plugins/.
Productos | Universidad de las Ciencias Informáticas. In: [online]. [Accessed 4 diciembre 2017]. Available from: http://www.uci.cu/investigacion-y-desarrollo/productos.
Requerimientos funcionales y no funcionales. In: [online]. [Accessed 9 mayo 2018]. Available from: https://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales.
Shoreline Firewall (Shorewall). In: [online]. [Accessed 4 diciembre 2017]. Available from: http://shorewall.org/#WhatIs.
UncomplicatedFirewall - Ubuntu Wiki. In: [online]. [Accessed 7 diciembre 2017]. Available from: https://wiki.ubuntu.com/UncomplicatedFirewall.
Vuurmuur Firewall. In: [online]. [Accessed 7 diciembre 2017]. Available from: https://www.vuurmuur.org/trac/.
ARISTEGUI O., José Luis. TEST CASES IN SOFTWARE TEST, LOS CASOS DE PRUEBA EN LA PRUEBA DEL SOFTWARE. In. 2010, no. No.3, pp. 27-34.
ARTEAGA PEÑA, Isaac Patricio y LITUMA VELÍN, Gonzalo Iván. Gestión de Firewall bajo Linux mediante Shorewall. In: [online]. 2006, [Accessed 27 noviembre 2017]. Available from: http://dspace.uazuay.edu.ec:8080/handle/datos/2239.
BELLOVIN, S. M. y CHESWICK, W. R. Network firewalls. In: *IEEE Communications Magazine*. septiembre 1994, Vol. 32, no. 9, pp. 50-57. DOI 10.1109/35.312843.
CABRERA V., César y NARVÁEZ, Paola. FIREWALLS y seguridad en internet mediante IPCcoop. In: [online]. 2006, [Accessed 4 diciembre 2017]. Available from: http://dspace.uazuay.edu.ec:8080/handle/datos/2251.
CCM. Introducción a la seguridad informática. In: *CCM* [online]. [Accessed 27 junio 2018]. Available from: https://es.ccm.net/contents/622-introduccion-a-la-seguridad-informatica.

CEBRIÁN, María R. Sahuquillo, Belén Domínguez. Un potente ciberataque afecta a grandes empresas de todo el mundo. In: *EL PAÍS* [online]. 28 junio 2017. [Accessed 1 febrero 2018]. Available from: https://elpais.com/internacional/2017/06/27/actualidad/1498568187_011218.html.

COMESAÑA CABEZA, José Luis. *Instalación y uso de los entornos de desarrollo* [online]. S.l.: s.n., 2012. Available from: http://docplayer.es/3976439-Instalacion-y-uso-de-entornos-de-desarrollo.html.

COMODO. Antivirus for Linux | Free Linux Antivirus for Virus Protection. In: *comodo.com* [online]. [Accessed 20 junio 2018]. Available from: https://www.comodo.com/home/internet-security/antivirus-for-linux.php.

COMODO. Free Internet Security | Why Comodo Internet Security Suite for PC? In: *comodo.com* [online]. [Accessed 20 junio 2018]. Available from: https://www.comodo.com/home/internet-security/free-internet-security.php.

DESAI, Mayur S., RICHARDS, Thomas C. y VON DER EMBSE, Thomas. System insecurity – firewalls. In: *Information Management & Computer Security*. agosto 2002, Vol. 10, no. 3, pp. 135-139. DOI 10.1108/09685220210431890.

EMPRESA SEGURMÁTICA. segav – Empresa Segurmática. In: [online]. [Accessed 27 junio 2018]. Available from: https://segurmatica.cubava.cu/tag/segav/.

ESET NOD32. ESET NOD32 antivirus 4 para Linux | ESET. In: *ESET NOD32* [online]. [Accessed 20 junio 2018]. Available from: https://www.eset.es/productos/nod32-antivirus-linux/#.

GARCIA-ALFARO, Joaquin. *Mecanismos de prevención*. S.l.: s.n., 2017.

GOMEZ, Ruiz. *Introducción al lenguaje C/C++* [online]. Málaga: s.n., 2013. [Accessed 5 enero 2018]. Available from: http://informaticagamarra.weebly.com/.

HAYAJNEH, Thaier, MOHD, Bassam, ITRADAT, Awni y QUTTOUM, Ahmad Nahar. Performance and Information Security Evaluation with Firewalls. In: *International Journal of Security and Its Applications*. 30 noviembre 2013, Vol. 7, no. 6, pp. 355-372. DOI 10.14257/ijsia.2013.7.6.36.

HERNÁNDEZ, Aguirre y ALEJANDRO, John. Análisis e implementación del firewall forefront TMG 2010. In: [online]. 4 julio 2013, [Accessed 27 noviembre 2017]. Available from: http://repository.ucatolica.edu.co/handle/10983/900.

IBM SOFTWARE GROUP. *Introduction to UML 2* [online]. S.l.: s.n., 2005. Available from: http://www.omg.org/news/meetings/workshops/MDA-SOA-WS_Manual/00-T4_Matthews.pdf.

PANDA SECURITY. La Revolución Antivirus - Panda Security. In: *pandasecurity.com* [online]. [Accessed 21 junio 2018]. Available from: /usa-es/.

PFSENSE®. pfSense® - World's Most Trusted Open Source Firewall. In: [online]. [Accessed 11 diciembre 2017]. Available from: https://www.pfsense.org/.

PRESSMAN, Roger S. *Ingeniería de software. Un enfoque práctico* [online]. Séptima edición. S.l.: s.n., 2010. ISBN 978-607-15-0314-5. Available from: http://cotana.informatica.edu.bo/downloads/ld-Ingenieria.de.software.enfoque.practico.7ed.Pressman.PDF.

805

RODRÍGUEZ SÁNCHEZ, Tamara. *Metodología de desarrollo para la Actividad productiva de la UCI.* Carretera a San Antonio Km 2 1⁄2. Torrens. Boyeros. La Habana. Cuba. UCI, 2015.

SOMMERVILLE, Ian. *Ingeniería de Software* [online]. Séptima edición. Madrid: Pearson Educación, 2005. [Accessed 9 mayo 2018]. ISBN 84-7829-074-5. Available from: http://zeus.inf.ucv.cl/~bcrawford/AULA_ICI_3242/Ingenieria%20del%20Software%207ma.%20Ed.%20-%20Ian%20Sommerville.pdf.

UML MODELING. Class Diagram - UML Diagrams - Unified Modeling Language Tool. In: *UML Modeling - Unified Modeling Language Tool.* [online]. [Accessed 9 mayo 2018]. Available from: https://www.visual-paradigm.com/VPGallery/diagrams/Class.html.

UML MODELING. Deployment Diagram - UML 2 Diagrams - UML Modeling Tool. In: *UML Modeling - Unified Modeling Language Tool.* [online]. [Accessed 26 febrero 2018]. Available from: https://www.visual-paradigm.com/VPGallery/diagrams/Deployment.html.

VIEIRA, Marcio. 5 ventajas del uso del firewall Open Source pfSense. In: *HopeMedia* [online]. [Accessed 11 diciembre 2017]. Available from: https://hopemedia.es/5-ventajas-firewall-pfsense/.

VISCENCIO, Escrito por Isidro Leos. Casos de Uso - Introducción. In: [online]. [Accessed 9 mayo 2018]. Available from: http://softtlan.blogspot.com/2007/01/casos-de-uso-introduccin.html.

VISUAL PARADIGM. *Software Desing Tools for Agile Teams, whit UML, BPMN and More* [online]. S.l.: s.n., 2015. [Accessed 11 diciembre 2017]. Available from: http://www.visual-paradigm.com/.

VISUAL PARADIGM. Visual Paradigm - Leading UML, BPMN, EA, Agile and Project Management Software. In: [online]. [Accessed 9 mayo 2018]. Available from: https://www.visual-paradigm.com/.

VUGT, Sander van. Setting Up the Netfilter Firewall with iptables and ufw. In: *Beginning Ubuntu LTS Server Administration* [online]. S.l.: Apress, 2008. pp. 351-361. [Accessed 4 diciembre 2017]. ISBN 978-1-4302-1082-5. Available from: https://link.springer.com/chapter/10.1007/978-1-4302-1081-8_12.

ZONEALARM. Software antivirus y de firewall por ZoneAlarm - Protección contra virus para su PC. In: *ZoneAlarm* [online]. [Accessed 20 junio 2018]. Available from: https://www.zonealarm.com/es/software/antivirus-firewall/.

The Uncomplicated Firewall. In: *Lullabot - Strategy, Design, Development* [online]. 10 febrero 2016. [Accessed 7 diciembre 2017]. Available from: https://www.lullabot.com/articles/the-uncomplicated-firewall. A simple firewall for everyone

¿Qué es un firewall? In: *Cisco* [online]. [Accessed 1 febrero 2018]. Available from: https://www.cisco.com/c/es_mx/products/security/firewalls/what-is-a-firewall.html.

Un firewall es un dispositivo de seguridad de la red que monitorea el tráfico que entra y sale de una red. Permite o bloquea el tráfico en función de un conjunto definido de reglas de seguridad.

Advanced Policy Firewall - R-fx Networks. In: [online]. [Accessed 27 noviembre 2017]. Available from: https://www.rfxn.com/projects/advanced-policy-firewall/.

Bitdefender - Cybersecurity Solutions for Business and Personal Use. In: [online]. [Accessed 21 junio 2018]. Available from: https://www.bitdefender.com/.

Cómo proteger nuestra pc | Observatorio Tecnológico. In: [online]. [Accessed 13 junio 2018]. Available from: http://recursostic.educacion.es/observatorio/web/gl/equipamiento-tecnologico/seguridad-y-mantenimiento/281-eduardo-e-quiroga-gomez.

Design a security firewall policy to filter incoming traffic in packet switched networks using classification methods. In: [online]. [Accessed 29 noviembre 2017]. Available from: http://www.redalyc.org/articulo.oa?id=467546204023.

Modelos conceptuales - Trabajos finales. In: [online]. [Accessed 9 mayo 2018]. Available from: http://www.buenastareas.com/ensayos/Modelos-Conceptuales/23455.html.

Seguridad - Cómo utilizar un firewall. In: [online]. [Accessed 27 junio 2018]. Available from: https://es.ccm.net/faq/191-seguridad-como-utilizar-un-firewall.

ANEXO 1

Caso de Prueba 2:

Descripción general:

- Permite exportar los datos las reglas contenidas en formato .HTML

Condiciones de ejecución:

- Habilitar el firewall.

SC1 Exportar reglas configurada

Escenario	Descripción	Nombre	Acción	Tipo de Flujo	Protocolo	Puerto	Dirección IP	Respuesta del sistema	Flujo central
EC 1.1 Enviar orden de exportar a HTML los datos de las reglas previamente configuradas	El usuario envía la orden de exportar las reglas previamente configuradas a un fichero HTML.	NA	NA	NA	NA	NA	NA	Permite guardar en un archivo en formato HTML llamado por el nombre deseado .HTML los datos exportados serán los valores de las reglas como: Fecha, Nombre, Acción, Tipo de flujo, Puerto y Dirección IP.	Seleccionar la etiqueta "Firewall" en el Menú Izquierdo de la aplicación. Seleccionar el botón "Salvar". Se muestra una ventana que define la dirección se desea guardar y se presiona Aceptar o Cancelar.
EC 1.2 Enviar orden de Exportar a HTML sin ninguna regla predeterminada	El usuario envía la orden de exportar las reglas previamente configuradas	NA	NA	NA	NA	NA	NA	Se permite guardar en un archivo sin embargo este no va a tener datos en la tabla.	

	a un fichero HTML.								

Tabla 23. Caso de Prueba exportar reglas.

Caso de Prueba 3:

Descripción general:

- Habilitar/deshabilitar el firewall en el sistema.

Condiciones de ejecución:

- El antivirus debe estar instalado y ejecutándose.

SC 1 Habilitar/deshabilitar el firewall.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1Habilitar el firewall	El usuario envía la orden de habilitar el firewall	Habilita el firewall y sus opciones	Hacer clic en el botón habilitar el firewall
EC 1.2 Deshabilitar el firewall	El usuario envía la orden de habilitar el firewall	Deshabilita el firewall y sus opciones	

Tabla 24. Caso de Prueba habilitar/deshabilitar el firewall.

Caso de Prueba 4:

Descripción general:

- Visualizar registro de cambio en el sistema.

Condiciones de ejecución:

- El antivirus debe estar instalado y ejecutándose.

SC 1 Visualizar registro de cambio.

Escenario	Descripción	Respuesta del sistema	Flujo central

EC 1.1 Visualizar registro de cambio	El usuario escoge el menú firewall	Visualiza el registro de cambios del sistema	Hacer clic en el menú firewall

Tabla 25. Caso de Prueba visualizar registro de cambio.

Caso de Prueba 5:

Descripción general:

- Visualizar la lista de reglas en el sistema.

Condiciones de ejecución:

- El antivirus debe estar instalado y ejecutándose.

SC 1 Visualizar la lista de reglas.

Escenario	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Visualizar la lista de reglas	El usuario escoge el menú firewall	Visualiza la lista de reglas en del sistema	Hacer clic en el menú firewall

Tabla 26. Caso de Prueba visualizar la lista de reglas.

Caso de Prueba 6:

Descripción general:

- Acceder al firewall desde la bandeja del sistema.

Condiciones de ejecución:

- El antivirus debe estar instalado y ejecutándose.

SC 1 Acceder al firewall desde la bandeja del sistema.

Escenario	Descripción	Respuesta del sistema	Flujo central

EC 1.1 Acceder al firewall desde la bandeja del sistema	Envía una orden de abrir el firewall desde el icono de Segurmática Antivirus	Abre el firewall.	Hacer clic derecho en el icono de Segurmática antivirus y hacer clic izquierdo sobre firewall.

Tabla 27. Caso de Prueba visualizar la lista de reglas.

ANEXO 2

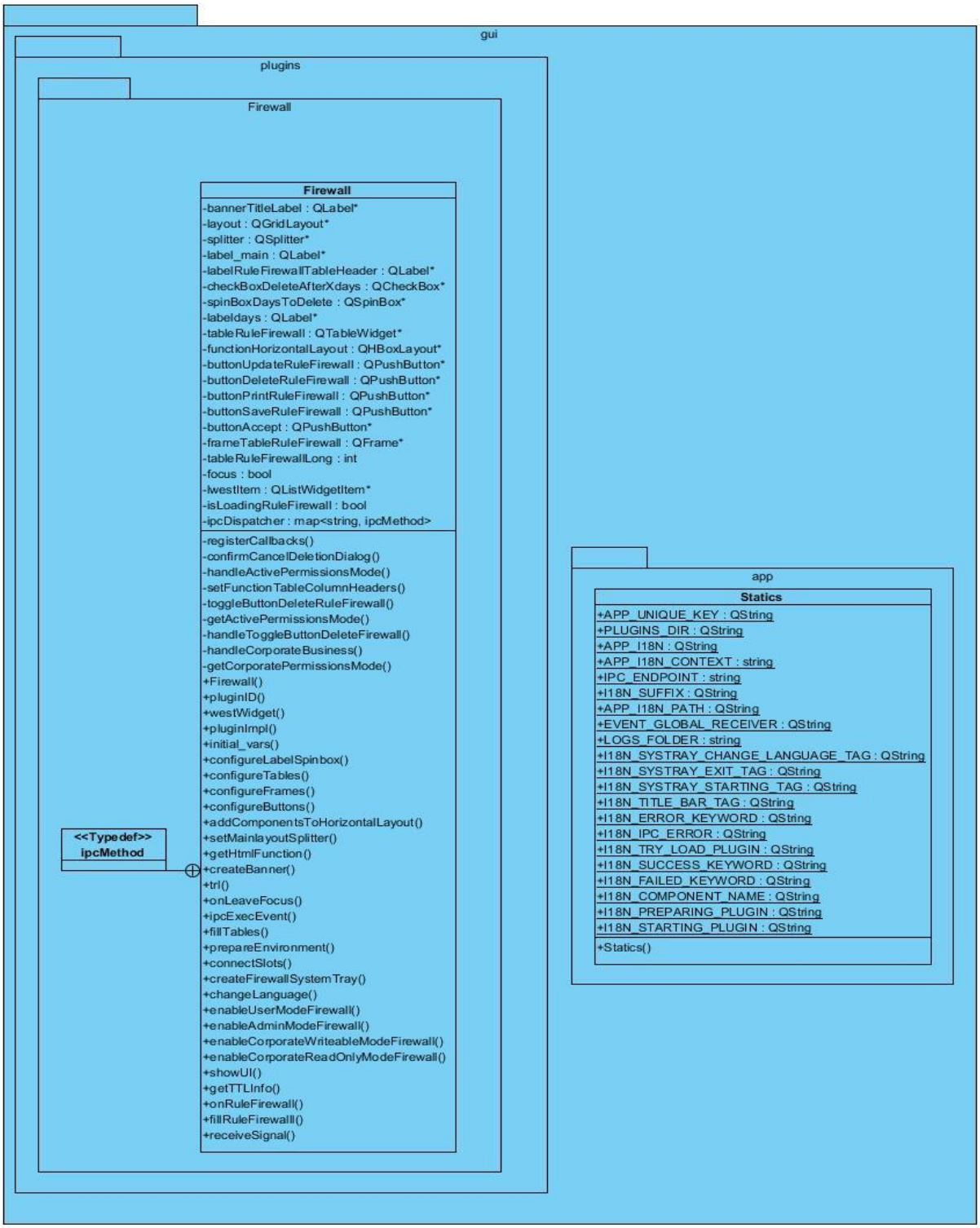

gui
plugins
Firewall
Firewall
-bannerTitleLabel : QLabel*
-layout : QGridLayout*
-splitter : QSplitter*
-label_main : QLabel*
-labelRuleFirewallTableHeader : QLabel*
-checkBoxDeleteAfterXdays : QCheckBox*
-spinBoxDaysToDelete : QSpinBox*
-labeldays : QLabel*
-tableRuleFirewall : QTableWidget*
-functionHorizontalLayout : QHBoxLayout*
-buttonUpdateRuleFirewall : QPushButton*
-buttonDeleteRuleFirewall : QPushButton*
-buttonPrintRuleFirewall : QPushButton*
-buttonSaveRuleFirewall : QPushButton*
-buttonAccept : QPushButton*
-frameTableRuleFirewall : QFrame*
-tableRuleFirewallLong : int
-focus : bool
-lwestItem : QListWidgetItem*
-isLoadingRuleFirewall : bool
-ipcDispatcher : map<string, ipcMethod>
-registerCallbacks()
-confirmCancelDeletionDialog()
-handleActivePermissionsMode()
-setFunctionTableColumnHeaders()
-toggleButtonDeleteRuleFirewall()
-getActivePermissionsMode()
-handleToggleButtonDeleteFirewall()
-handleCorporateBusiness()
-getCorporatePermissionsMode()
+Firewall()
+pluginID()
+westWidget()
+pluginImpl()
+initial_vars()
+configureLabelSpinbox()
+configureTables()
+configureFrames()
+configureButtons()
+addComponentsToHorizontalLayout()
+setMainlayoutSplitter()
+getHtmlFunction()
+createBanner()
+trl()
+onLeaveFocus()
+ipcExecEvent()
+fillTables()
+prepareEnvironment()
+connectSlots()
+createFirewallSystemTray()
+changeLanguage()
+enableUserModeFirewall()
+enableAdminModeFirewall()
+enableCorporateWriteableModeFirewall()
+enableCorporateReadOnlyModeFirewall()
+showUI()
+getTTLInfo()
+onRuleFirewall()
+fillRuleFirewall()
+receiveSignal()
<<Typedef>>
ipcMethod
app
Statics
+APP_UNIQUE_KEY : QString
+PLUGINS_DIR : QString
+APP_I18N : QString
+APP_I18N_CONTEXT : string
+IPC_ENDPOINT : string
+I18N_SUFFIX : QString
+APP_I18N_PATH : QString
+EVENT_GLOBAL_RECEIVER : QString
+LOGS_FOLDER : string
+I18N_SYSTRAY_CHANGE_LANGUAGE_TAG : QString
+I18N_SYSTRAY_EXIT_TAG : QString
+I18N_SYSTRAY_STARTING_TAG : QString
+I18N_TITLE_BAR_TAG : QString
+I18N_ERROR_KEYWORD : QString
+I18N_IPC_ERROR : QString
+I18N_TRY_LOAD_PLUGIN : QString
+I18N_SUCCESS_KEYWORD : QString
+I18N_FAILED_KEYWORD : QString
+I18N_COMPONENT_NAME : QString
+I18N_PREPARING_PLUGIN : QString
+I18N_STARTING_PLUGIN : QString
+Statics()

Figura 15. Clase Firewall y Statics del diagrama de clase

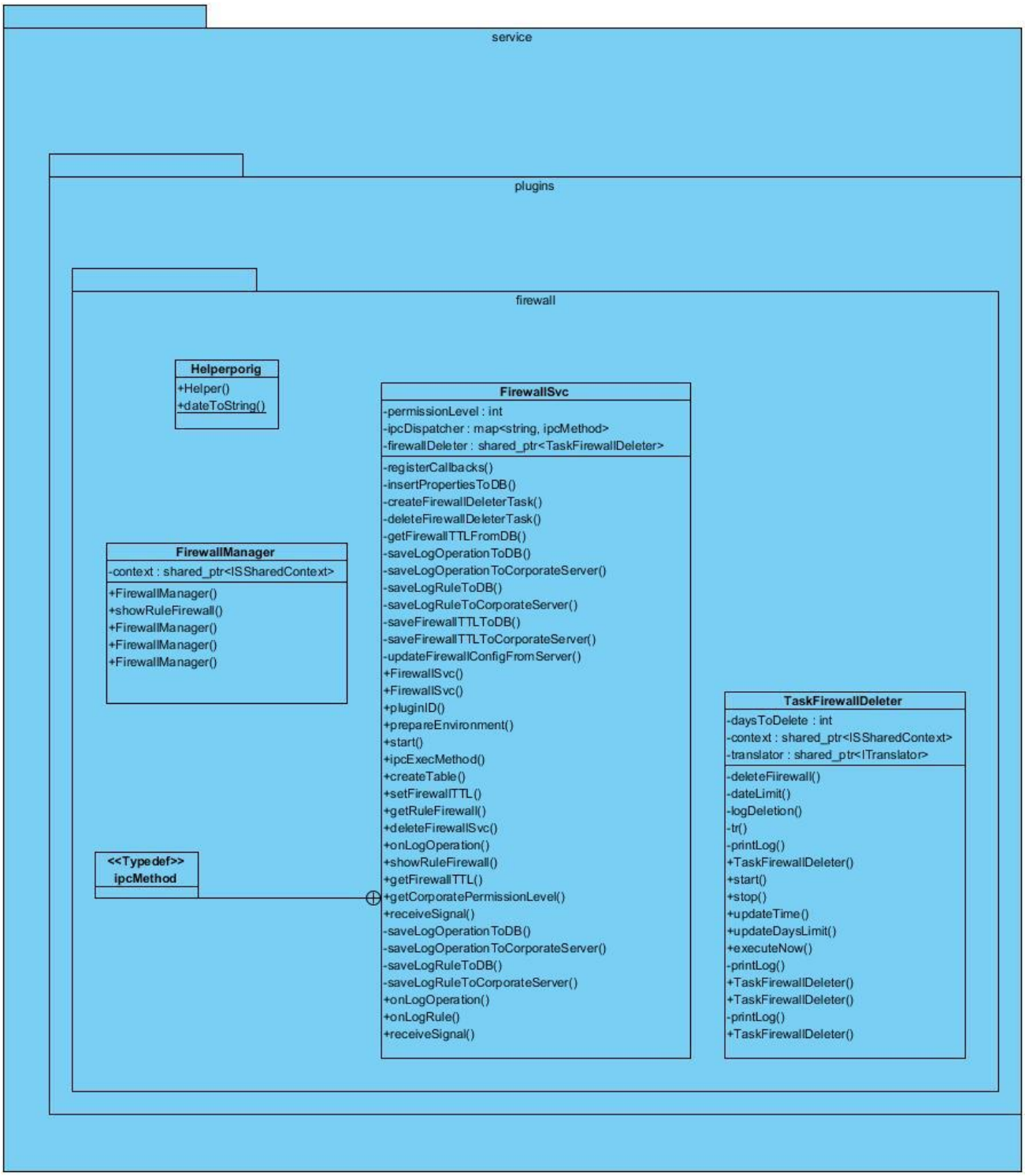


Figura 16. Clase FirewallSvc y sus clases relacionadas del diagrama de clase

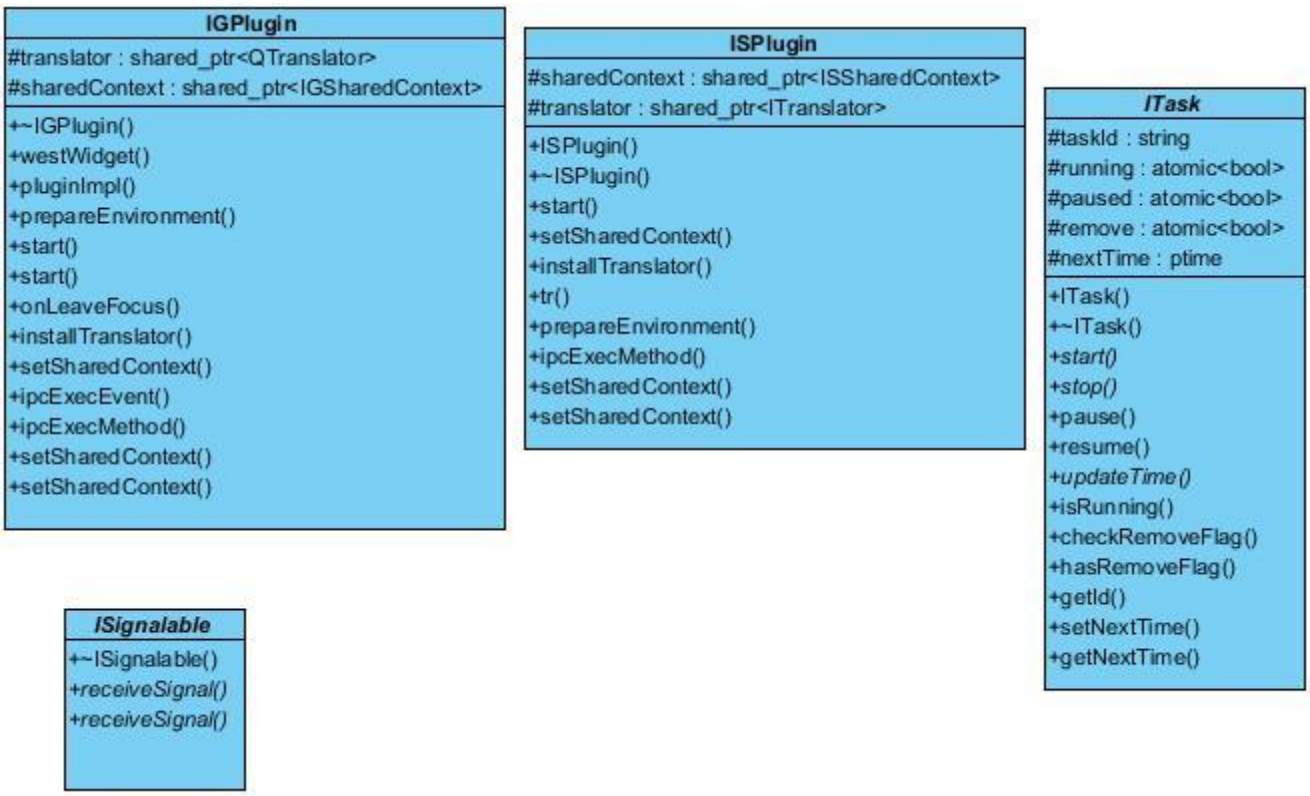


Figura 17. Clases del sistema del diagrama de clase