Universidad de Cienfuegos "Carlos Rafael Rodríguez"

Facultad de Ingeniería

Carrera de Ingeniería Informática

Título: "Módulo web para administrar el Sistema InterFirewall perteneciente a la División Territorial de ETECSA Cienfuegos".

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

Autor(es):

David Valdaliso López

Tutor(es):

Msc. Denis Morejón López

Msc. Anay Carrillo Ramos

Consultante(es):

Ing. Darian Enrique Martínez Pomar

Cienfuegos, Cuba

Curso 2016 - 2017

Agradecimientos

A todas las personas que me ayudaron en la realización de esta investigación y en especial:

A mis tutores Denis Morejón López y Anay Carrillo Ramos por haberme brindado su apoyo, su tiempo, esfuerzo y confianza.

A mis mejores amigos de estos años de estudio Richard, Dayron, Yairo por haber compartido conmigo todo este tiempo.

A mi madre Josefina López y a mi padre Jorge Bernardo Valdaliso, porque gracias a ellos he logrado ser quien soy.

Un agradecimiento especial a mi hermana que fue mi luz todo este tiempo y siempre me dio su apoyo incondicional.

Dedicatoria

Dedico este trabajo a todas las personas que al igual que yo luchan por un título de ingeniero informático y con el mismo doy constancia que si se puede lograr ese gran sueño de ser ingeniero

Resumen

Esta investigación realizada en la Empresa de Telecomunicaciones de Cuba S.A (ETECSA), en Cienfuegos, desarrolla el módulo de la interfaz web del Sistema InterFirewall, este sistema, conformado por tres módulos, facilita la administración de los cortafuegos que se utilizan en la entidad. Con este módulo el cliente manipula el tráfico de los diferentes servicios de la red, permitiéndolo o negándolos, según los requerimientos de seguridad, sin la intervención directa sobre firewalls de diversas tecnologías. La aplicación se realiza con lenguajes de programación de alto nivel, JavaScript con su framework de desarrollo del módulo web Extjs6 y la comunicación entre cliente y servidor se ejecuta con JSON, se utiliza Ajax para realizar las peticiones de la interfaz web al servidor y se propondrán un conjunto de reglas a considerar por los administradores de red.

Abstract

This research carried out in the Cuban Telecommunications Company S.A (ETECSA), in Cienfuegos, develops the InterFirewall System web interface module. This system, made up of three modules, facilitates the administration of the firewalls that are used in the entity. With this module the client manipulates the traffic of the different services of the network, allowing or denying them, according to the security requirements, without the direct intervention on firewalls of diverse technologies. The application is made with high level programming languages, JavaScript with its Extjs6 web module development framework and the communication between client and server is executed with JSON, Ajax is used to make the requests of the web interface to the server, and will propose a certain amount of rules for the easy management of the administrator.

.

Índice de Contenido

Introducción 1

Capítulo I 9

1. Fundamentación teórica 9

1.1. Introducción 9

1.2. Interfaz Web 9

1.3. Cortafuego 10

1.4. Buenas prácticas para configurar las reglas del Cortafuego 11

1.5. ACL 12

1.5.1. Aspectos que se deben tener en cuenta 13

1.5.2. Políticas del cortafuego 14

1.6. Sistemas similares existentes 14

1.6.1. Zentyal 14

1.6.2. PfSense 15

1.6.3. IPCop 15

1.7. Framework de JavaScript 16

1.7.1. AngularJs 16

1.7.2. JQuery 16

1.7.3. Nodejs 17

1.7.4. ExtJS 17

1.8. Tendencias, metodologías y/o tecnologías actuales 19

1.8.1. Etapa de diseño y desarrollo 19

1.8.2. Uso de lenguaje y tecnologías Web 20

1.8.2.1. Ajax 20

1.8.2.2. JavaScript 21

Índice de Contenido

1.8.2.3. JSON (JavaScript Object Notation) 22

1.8.2.4. Servidor Web 23

1.8.2.5. Patrón Modelo-Vista-Controlador (MVC) 23

1.8.3. Herramientas a Utilizar 24

1.8.3.1. Visual Paradigm 24

1.8.3.2. WebStorm 25

1.9. Conclusiones 26

Capítulo II 27

2.1 Introducción 27

2.2 Modelo de Dominio 27

2.2.1. Definición de las entidades y los conceptos principales 28

2.2.2. Reglas del Negocio 28

2.2.3. Representación del Modelo del Dominio 29

2.3 Requisitos 29

2.3.1. Concepción general del sistema 29

2.3.2. Requerimientos funcionales: 29

2.3.3. Requisitos no funcionales: 32

2.3.4. Modelo de caso de uso del Sistema 34

2.3.4.1 Actor del Sistema 35

2.3.4.2 Diagrama de caso de uso del sistema 35

2.3.4.3 Descripción de los casos de uso del sistema. 36

2.4 Diseño 38

2.4.1 Diagrama de Clase de Diseño Web 38

2.4.2 Diagrama de implementación 39

2.4.3 Principios de Diseño del Sistema (Arreglar Revisar) 40

2.4.4 Patrón de Diseño 40

Índice de Contenido

2.4.6 Ayuda 42

2.4.8 Tratamiento de errores 42

2.5 Propuestas de Reglas 42

2.6 Conclusiones 45

Capítulo III 46

3.1 Introducción 46

3.1.1 Escenario #1 46

3.1.2 Escenario #2 46

3.2 Verificación y Validación de Software 47

3.2.1. Caso de Prueba "Registrar Usuario" 47

3.2.2. Caso de Prueba "Crear Puerto" 48

3.2.3. Caso de Prueba "Cambiar Contraseña" 48

3.2.4. Caso de Prueba "Cuando no existe conexión con el servidor" 48

3.2.5. Caso de Prueba "Asignar Zona" 49

3.2.6. Caso de Prueba "Servicio Configuración" 49

3.2.7. Caso de Prueba "Member" 49

3.2.9. Caso de Prueba "Firewall" 50

3.2.10 Caso de prueba Adicionar reglas a la ACL 50

3.3 Estimación por puntos de casos de uso 51

3.3.1 Cálculo de Puntos de Casos de Uso sin Ajustar 51

3.3.1 Factor de Peso de los Actores sin ajustar. 52

3.4 Cálculo de Puntos de Casos de Uso Ajustados 55

3.4.1 Factor de complejidad técnica (TCF) 56

3.4.2 Factor Ambiente (EF) 58

3.5 De los Puntos de Casos de Uso a la estimación del esfuerzo: 60

Índice de Contenido

3.6 Cálculo de Costo 61

3.7 Análisis de los costos y beneficios 61

3.8 Conclusiones 62

Conclusiones 63

Recomendaciones 64

Referencias Bibliografías 65

Bibliografía 68

Índice de Tablas

Tabla 1 Actor Del Sistema *35*

Tabla 2 Descripción de los casos de uso del sistema *36*

Tabla 3 Diagrama de clases web *39*

Tabla 4 Red Usuario *43*

Tabla 5 Red Administrador *44*

Tabla 6 Red Gestnet *44*

Tabla 7 Red DMZ *45*

Tabla 8 Clasificación de los Actores de Sistema *52*

Tabla 9 Descripción de los casos de uso del sistema según el tipo *54*

Tabla 10 Factor de Complejidad Técnica *58*

Tabla 11 Factor ambiente *59*

Tabla 12 Estimación de Esfuerzo *61*

Índice de Figuras

Figura 1 Como funciona AJAX.....*21*

Figura 2 Modelo del Dominio *29*

Figura 3 Caso de Uso del Sistema.....*36*

Figura 4 Diagramas de implementación *40*

Introducción

La expansión de las Tecnologías de la Información y las Comunicaciones (TIC), conjuntamente con Internet, representan un motor de impulso al desarrollo científico y tecnológico de cualquier nación, siendo soporte y fundamento de la sociedad del conocimiento. Esta nueva situación ha propiciado un cambio acelerado hacia una sociedad que basa su riqueza en la gestión de la información y el conocimiento disponibles en la Red Global.

En Cuba los avances alcanzados en los últimos años en la informatización de la sociedad con el incremento de tecnologías de la información en todos los sectores y en particular de las redes informáticas y sus servicios asociados, y el impulso orientado por la dirección del país al desarrollo acelerado de programas que multipliquen dichos logros, requieren la adopción de medidas que garanticen un adecuado nivel de seguridad para su protección y ordenamiento. La seguridad de las organizaciones, sistemas y redes de información están constantemente amenazadas por diversas fuentes que incluyen ataques de distintos tipo y origen; la ocurrencia de catástrofes, errores de operación y negligencias, aumentan los riesgos a que están expuestos los servicios y protocolos utilizados, así como el contenido de la información tratada en dichos sistemas, todo lo cual puede afectar severamente la confidencialidad, integridad y disponibilidad de la información. A partir de las vulnerabilidades y debilidades propias de los sistemas informáticos y de las dificultades y limitaciones que se presentan para detectar y neutralizar oportunamente las posibles acciones del enemigo en esta esfera se implementa en el país un basamento legal que establece los requerimientos de seguridad en el empleo de las tecnologías de la información a partir de criterios de racionalidad y utilidad, que

resulten susceptibles de verificación y propendan a la disminución de los riesgos en la seguridad informática.

La seguridad es un aspecto indispensable a tratar cuando una organización conecta su red privada a Internet, sin tomar en cuenta el tipo de negocios, los administradores de red tienen que incrementar la seguridad de sus sistemas, debido a que se expone toda la infraestructura de la red, los servicios y la información a las amenazas del uso de esta red. Para proveer el nivel de protección requerida, la organización necesita establecer políticas de seguridad para prevenir el acceso no autorizado de usuarios a los recursos propios de la red privada, y protegerse contra la exportación de información privada, aún si una organización no está conectada al internet, esta debe establecer una política de seguridad interna para administrar el acceso de usuarios a determinados servicios de la red y proteger la información.

Un Cortafuego es un sistema o grupo de sistemas que impone una política de seguridad entre la organización de red la privada e Internet. El Cortafuego determina cuál de los servicios de red pueden ser accedidos dentro de esta por los que están fuera, es decir quién puede entrar para utilizar los recursos de red pertenecientes a la organización. Para que un Cortafuego sea efectivo, todo tráfico de información deberá pasar a través del mismo para ser inspeccionado. El Cortafuego podrá únicamente autorizar el paso del tráfico, y el mismo podrá ser inmune a la penetración, desafortunadamente, este sistema no puede ofrecer protección alguna una vez que el agresor lo traspasa o permanece en torno a este, un Cortafuego de Internet no es justamente un enrutador, un servidor de defensa, o una combinación de elementos que proveen seguridad para la red, el cortafuego es parte de la política de seguridad diseñada para proteger los recursos informáticos. Esta política podrá incluir publicaciones con las guías de ayuda donde se informe a los usuarios de sus responsabilidades, normas de acceso a la red, política de servicios en la red, política

de autenticidad en acceso remoto o local a usuarios propios de la red, normas de dial-in y dial-out, reglas de encriptación de datos y discos, normas de protección de virus, y entrenamiento.

Los predecesores de los cortafuegos para la seguridad de la red fueron los routers que mantenían a las redes separadas unas de otras. En 1988 comienzan las publicaciones sobre los sistemas de filtro conocidos como cortafuegos de filtrado de paquetes. El filtrado de paquetes actúa mediante la inspección de los paquetes, si un paquete coincide con el conjunto de reglas del filtro, el paquete se reducirá o será rechazado enviando una respuesta de error al emisor. Este tipo de filtrado de paquetes no presta atención a si el paquete es parte de una secuencia existente de tráfico, en su lugar, se filtra cada paquete basándose únicamente en la información contenida en el paquete en sí. Los protocolos TCP y UDP comprenden la mayor parte de comunicación a través de Internet, utilizando por convención puertos bien conocidos para determinados tipos de tráfico, por lo que un filtro de paquetes puede distinguir entre ambos tipos de tráfico (ya sean navegación web, impresión remota, envío y recepción de correo electrónico, transferencia de archivos u otros); a menos que las máquinas a cada lado del filtro del paquete estén utilizando puertos no estándar. El filtrado de paquetes llevado a cabo por un cortafuego actúa en las tres primeras capas del modelo de referencia OSI, lo que significa que todo el trabajo se realiza entre la capa de red y las capas físicas. Cuando el emisor origina un paquete y es filtrado por el cortafuego, comprueba las reglas de filtrado de paquetes que tiene configurado, aceptando o rechazando el paquete en consecuencia. Cuando el paquete pasa a través de cortafuegos, éste filtra el paquete mediante un protocolo y un número de puerto base. Por ejemplo, si existe una norma en el cortafuego para bloquear el acceso telnet, bloqueará el protocolo TCP para el número de puerto 23.

Luego se difundieron los cortafuegos de inspección de estado de paquetes que utilizan la colocación de cada paquete individual dentro de una serie de paquetes, mantiene registros de todas las conexiones que pasan por el cortafuegos, determina si un paquete indica el inicio de una nueva conexión, es parte de una conexión existente, o es un paquete erróneo. Este tipo de cortafuegos pueden ayudar a prevenir ataques contra conexiones en curso o ciertos ataques de denegación de servicio.

Los cortafuegos de aplicación actúan sobre la capa de aplicación del modelo OSI, son la tercera generación, manipulan otras aplicaciones y protocolos (por ejemplo: protocolo de transferencia de ficheros, DNS o navegación web), y permite detectar si un protocolo no deseado se coló a través de un puerto no estándar o si se está abusando de un protocolo de forma perjudicial. Un cortafuego de aplicación es mucho más seguro y fiable cuando se compara con un cortafuego de filtrado de paquetes, ya que interviene en las siete capas del modelo de referencia OSI. Puede filtrar protocolos de capas superiores tales como FTP, TELNET, DNS, DHCP, HTTP, TCP, UDP y TFTP. Por ejemplo, si una organización quiere bloquear toda la información relacionada con una palabra en concreto, puede habilitarse el filtrado de contenido para bloquear esa palabra en particular. No obstante, los cortafuegos de aplicación resultan más lentos que los de estado.

En la entidad ETECSA Cienfuegos se enfrentan a una situación problémica con cortafuegos implementados sobre diversas tecnologías. Es justo esta diversidad el primer obstáculo para implementar políticas de control de acceso que abarquen toda la red. Se está realizando un sistema InterFirewall que se comunica con diferentes tecnologías de cortafuegos y este no cuenta con una Interfaz web que gestione automáticamente las reglas, ACL y cortafuegos de la red, capaz de detectar el tráfico malicioso y pase al cortafuego los datos precisos y la indicación del bloqueo al equipo invasor. Además, este sistema al no tener interfaz web, el administrador de red debe

contar con conocimientos sobre el lenguaje con el que está implementado la interfaz línea de comando.

Problema

En la división territorial de ETECSA Cienfuegos se presenta el problema de que el sistema informático denominado InterFirewall, que permite interactuar con cortafuegos de distintas tecnologías para aplicar políticas globales en la red, carece de una interfaz web de administración que facilite esa labor.

Objeto de Estudio

Definimos como objeto de estudio de la presente investigación la gestión de las tecnologías utilizadas en los cortafuegos de la dirección territorial de ETECSA Cienfuegos para el sistema InterFirewall mediante interfaz web, siendo el campo de acción el proceso de elaboración de la interfaz web para el sistema InterFirewall.

Idea a defender

El desarrollo de una interfaz web para el sistema de InterFirewall facilitará la gestión del mismo, para que los administradores de la red de ETECSA Cienfuegos logren garantizar el nivel de seguridad, control, monitorización, y respuesta deseado en la organización.

Objetivo General

Desarrollar una interfaz web para el sistema InterFirewall, así como realizar propuestas de reglas de control de acceso (ACL) para la seguridad de la Red LAN ETECSA Cienfuegos.

Objetivos Específicos

1. Analizar las tendencias de desarrollo web que se ajusten a las necesidades de esa institución.
2. Diseñar una Interfaz web que gestione los router en la entidad ETECSA utilizando las tecnologías de desarrollo de software adecuada.
3. Proponer reglas para las ACL
4. Implementar una Interfaz web que facilite el manejo de las ACL de los diferentes router.
5. Validación del sistema

Tareas a realizar

1. Entrevista con el personal especializado en Seguridad Informática.
2. Estudio de los conceptos asociados al dominio del problema**.**
3. Análisis crítico de los sistemas de cortafuegos que se utilizan en Cuba y en el mundo.
4. Dominio de las diferentes tecnologías de cortafuegos incluyendo los router existentes en ETECSA
5. Selección de herramientas, metodología, lenguajes y tecnologías de desarrollo mediante una revisión bibliográfica.
6. Desarrollo de la interfaz web para facilitar la gestión las ACL.
7. Implementación de las pruebas con los diferentes navegadores que utilizan los clientes de la empresa

Aporte Práctico

El aporte práctico reside en la funcionalidad que le aporta el módulo interfaz web al InterFirewall que posibilita al administrador que pueda controlar varios cortafuegos para realizar acciones de control y de consultas de información asociadas a la infraestructura de la red.

Estructuración del contenido del trabajo de diploma

El presente trabajo, está estructurado en 3 capítulos, a continuación, se muestra

una breve descripción de cada uno de ellos:

Capítulo I: Fundamentación teórica

En este capítulo se determinan los antecedentes y el estado actual de los sistemas informáticos para la gestión de los cortafuegos, presentando los aspectos teóricos y conceptos asociados con al tema a desarrollar y se describen las metodologías, lenguajes y tecnologías a utilizar.

Capítulo II: Construcción de la solución propuesta

Este capítulo presenta una descripción detallada del diagrama del modelo de objeto del dominio y descripción de las reglas del dominio para garantizar las restricciones que existen en el mismo. Se describe el sistema propuesto identificando los requerimientos funcionales y no funcionales, los casos de uso del sistema, actores, paquetes y sus relaciones, la descripción de caso de uso y los diagramas de clases Web. Se propondrán ACL y reglas más avanzada que se utilizarán en la interfaz Web.

Capítulo III: Análisis de factibilidad y validación de la solución propuesta

En este capítulo se diseñan los casos de pruebas funcionales para el sistema, así con sus respectivas validaciones. Además, se detalla el estudio de la factibilidad de la aplicación que se propone, teniendo en cuenta el análisis de los costos, beneficios y planificación para el desarrollo del mismo.

Capítulo I

1. Fundamentación teórica

1.1. Introducción

En este capítulo se fundamenta teóricamente el proceso de implementar las políticas de seguridad a través de cortafuegos mediante el Sistema InterFirewall específicamente en la gestión del módulo de interfaz web. Se destacan los principales conceptos del dominio del problema, descripción de tendencias, tecnologías y metodologías actuales selecciona das para dar solución a la problemática propuesta.

1.2. Interfaz Web

La interfaz web está constituida por elementos gráficos que permiten al usuario acceder a los contenidos, navegar e interactuar. El diseño de la interfaz es importante para mantener la motivación en los usuarios. Cuando accedemos a alguna de las muchas fuentes de información, desde los simples libros hasta los ordenadores, encontramos que la información contenida está organizada de forma tal que puede accederse a los contenidos con facilidad. En algunos casos, como en los libros, por ejemplo, el uso continuo por generaciones ha hecho que dicha organización se encuentre estandarizada y sea muy sencillo para los lectores encontrar cada uno de los elementos que lo constituyen.[1]

El concepto de interfaz es muy amplio y se refiere a todo sistema que permite el contacto y la funcionalidad entre dos sistemas diferentes. Por ejemplo, los botones y la pantalla del teléfono celular conforman la interfaz, ya que permite que el usuario pueda emplear las funciones que este aparato ofrece.

En definitiva, la interfaz web es el conjunto gráfico que permite la presentación y la navegación del sitio. Esto se consigue con la inclusión de elementos gráficos comunes a toda la web que son estándares, haciendo que los usuarios tengan completo control sobre las funcionalidades del sitio desde el momento mismo de entrar a él sin que para ello deba tener amplios conocimientos ni preparación anterior alguna.

Una página web puede contar con los mejores contenidos en el género que se desarrolla, pero fracasará si la interfaz no permite un rápido y cómodo acceso a los mismos por parte de los usuarios. Por el contrario, una página web cuyos contenidos sean de menor calidad (sin que éstos sean malos, por supuesto) pero cuya interfaz permite que sus usuarios naveguen en forma sencilla, tengan acceso en forma inmediata al contenido que dese an e interactúen en forma sencilla, tendrá un mayor éxito.[1]

1.3. Cortafuego

Un Cortafuego es un sistema o un grupo de sistemas que implementan una política de
control de acceso entre dos o más redes, compuesto por dos grandes módulos; uno destinado a bloquear los accesos y el otro a permitirlos.

Los cortafuegos sirven para múltiples propósitos, entre estos la restricción de entrada de usuarios a puntos cuidadosamente controlados de la red interna, la prevención ante los intrusos que tratan de ganar espacio hacia el interior de la red y los otros esquemas de defensas establecidos, la restricción de uso de servicios tanto a usuarios internos como externos y determina cuáles de los servicios de red pueden ser accedidos dentro de ésta por los que están fuera, es decir, quién puede entrar a utilizar los recursos de red pertenecientes a la organización.

El cortafuego es la primera línea de defensa ante un ataque a la red desde Internet y debe ser capaz de repeler un acceso no autorizado antes de que el atacante pueda llegar a la red local o al ordenador y, al mismo tiempo, permitir el normal intercambio de datos entre el ordenador y los servicios que le está permitido recibir de la red local y de internet.

Para ello se utilizan una serie de reglas que establecen como deben ser tratados los elementos que se envían o reciben en el ordenador y si estos elementos pueden continuar su camino hasta su destino o deben ser interceptados y devueltos a su origen.[2]

1.4. Buenas prácticas para configurar las reglas del Cortafuego.

La primera regla que debe tener cualquier administrador es que el Cortafuego no es suficiente para proteger la red y la información de su entidad. Para configurar un Cortafuego se debe pasar por un proceso de configuración exhaustivo de las reglas y/o políticas para garantizar la máxima seguridad.

- No utilice “any” o cualquiera en una regla especifica del Cortafuego.

Esto evitara problemas de seguridad y de control del flujo de tráfico. Una regla que diga cualquier servicio de cualquier fuente a cualquier destino, esto no permite que el Cortafuego actué como punto de control fuerte para el tráfico que fluya a través de él, debe asegurarse que el Cortafuego solo permita el tráfico necesario.

- Añadir comentarios a las reglas del Cortafuego.

Con el tiempo las reglas del Cortafuego crecen y se va haciendo difícil recordar porque una regla se creó, un comentario ayudaría mucho ya que se evita crear reglas

redundantes. Además, si otro administrador inicia sesión estos comentarios le ayudaría a entender rápidamente las reglas y cuál es su propósito.

- Planificar la adición de nuevas reglas.

Cada vez que se desee implementar una nueva política en el Cortafuego, este proceso de cambios de las reglas del Cortafuego se debe programar en momentos oportunos si no se puede afectar el flujo de tráfico de paquetes, los cambios no se deben realizar durante horas o días pico de trabajo porque las conexiones pueden caer y algunos servicios se verán afectados.

- Mantener actualizado el firmware del Cortafuego

Los fabricantes corrigen los problemas de configuración, de seguridad y mejoras a través de parches o nuevas versiones de firmware, por esto es muy importante mantener instalado la última versión de firmware liberado por el fabricante.

1.5. ACL

Una lista de Control de acceso ACL (Access control list) es usada para fomentar la separación de privilegios, es una forma de determinar los permisos de acceso apropiado a un objeto. Las ACL permiten controlar el flujo del tráfico de equipos de redes, tales como enrutadores y conmutadores (switch), su principal objetivo es filtrar el tráfico permitiendo o denegándolo en la red de acuerdo a alguna condición. Estas permiten controlar el tráfico entrante y saliente.

Las ACL especifican un solo par de direcciones de referencia, contra el que se comparan todos los paquetes que entren o salgan de la interfaz en el que se instale la ACL, una ACL filtra tráfico con base en la dirección ip origen de los paquetes.

1.5.1. Aspectos que se deben tener en cuenta

1. Sin ACL: de forma predeterminada, cuando se crea un extremo, se permite todo en este extremo.

2. Permitir: al agregar uno o más intervalos "permitidos", se deniegan todos los demás intervalos de forma predeterminada. Solo los paquetes del intervalo de IP permitidas podrán comunicarse con el extremo del destino.

3. Denegar: al agregar uno o más intervalos "denegados", se permiten todos los demás intervalos del tráfico de forma predeterminada.

4. Combinación de permitir y denegar: puede utilizar una combinación de "permitir" y "denegar" si desea definir un intervalo de direcciones IP específico para permitir o denegar.

Sintaxis General de Reglas

Generalmente se expresan como condiciones y acciones que se consulta en orden hasta encontrar una regla que permita tomar una decisión sobre el bloqueo o el reenvío de la trama. Se debe tener presente el orden de análisis de las condiciones para implementar la política de seguridad de forma correcta, el orden de reglas especifica la prioridad.

<IDREGLA>	<ACCIÓN>	<SERVICIO>	<FUENTE>	<DESTINO>

Donde:

IDREGLA: numero para identificar la regla

ACCIÓN: permitir o denegar

SERVICIO: puertos con protocolos

FUENTE: de donde viene el paquete

DESTINO: hacia donde se dirige el paquete

1.5.2. Políticas del cortafuego

Hay dos políticas básicas en la configuración de un cortafuego y que cambian radicalmente la filosofía fundamental de la seguridad en la organización:

> **Política restrictiva:** Se deniega todo el tráfico excepto el que está explícitamente permitido. El cortafuego obstruye todo el tráfico y hay que habilitar expresamente el tráfico de los servicios que se necesiten.
>
> **Política permisiva:** Se permite todo el tráfico excepto el que esté explícitamente denegado. Cada servicio potencialmente peligroso necesitará ser aislado básicamente caso por caso, mientras que el resto del tráfico no será filtrado.

La política restrictiva es la más segura, ya que es más difícil permitir, por error, tráfico potencialmente peligroso, mientras que en la política permisiva es posible que no se haya contemplado algún caso de tráfico peligroso y sea permitido por defecto. [4]

1.6. Sistemas similares existentes

1.6.1. Zentyal

Zentyal es una aplicación web que usa el servidor web Apache, principalmente escrito en Perl orientado a objetos. Su interfaz de usuario usa CSS y AJAX, e incluye varios componentes Mason, como una tabla genérica usada para configurar servicios los errores y avisos se manejan a través de excepciones software

Zentyal (anteriormente conocido como eBox Platform) es un servidor de red unificada de código abierto (o una plataforma de red unificada) para las Pymes. Zentyal puede actuar gestionando la infraestructura de red, como puerta de enlace a Internet (Gateway), gestionando las amenazas de seguridad (UTM), como servidor de oficina, como servidor de comunicaciones unificadas o una combinación de estas[9].Además, Zentyal incluye un marco de desarrollo (un framework) para facilitar el desarrollo de nuevos servicios basados en Unix.[5]

1.6.2. PfSense

Pfsense es una distribución personalizada de FreeBSD (Sistema Operativo) para usarlo en servicios de redes LAN y WAN tales como Cortafuego, enrutador, servidor de balanceo de carga, entre otras. El modelo de desarrollo de Pfsense es de código abierto, el núcleo de Pfsense es basado en el sistema operativo libre llamado BSD, el tipo de núcleo de Pfsense es de tipo monolítico, el usado en UNIX. [6]

Pfsense cuenta con un gestor de paquetes desde su interfaz gráfica accedida remotamente para ampliar sus funcionalidades, al elegir el paquete deseado el sistema lo descarga y lo instala automáticamente. Existen 60 módulos disponibles para descargar al Pfsense e instalarlos entre estos son el proxy Squid Inspector, Snort, ClamAV [6]

1.6.3. IPCop

IPCop tiene como objetivos ser un cortafuego sencillo, con pocos requerimientos hardware orientado a usuarios domésticos o a pequeñas empresas (SOHO), administrado a través de una interfaz web, con funcionalidades básicas y avanzadas, yendo (a manera de ejemplo) desde el simple filtrado de paquetes hasta la asignación

de ancho de banda fijo a cada puesto de trabajo o la configuración de redes virtuales VPN. IPCop se actualiza desde el Interfaz Web de manera muy sencilla, incluyendo actualizaciones del Kernel. [7]

La configuración y administración de estos cortafuegos requiere de conocimientos medianamente avanzados, los menús y opciones no son intuitivos con la complejidad de los conceptos técnicos. Todos estos cortafuegos están limitados a manejar un solo medio y no pueden configurar routers de distintas tecnologías.

1.7. Framework de JavaScript

1.7.1. AngularJs

AngularJs es JavaScript. Es un proyecto de código abierto, realizado en JavaScript que contiene un conjunto de librerías útiles para el desarrollo de aplicaciones web y propone una serie de patrones de diseño para llevarlas a cabo. En pocas palabras, es lo que se conoce como un framework para el desarrollo, en este caso sobre el lenguaje JavaScript con programación del lado del cliente. [8]

1.7.2. JQuery

jQuery es considerado un Framework de JavaScript, o ambiente de desarrollo. Lo que no es más que un conjunto de utilidades las cuales ya fueron programadas, probadas y podemos utilizarlas de una manera muy simplificada. En otras palabras, se podrá lograr los mismos resultados, en menos tiempo sin necesidad de programar una funcionalidad completamente. [9]

1.7.3. Nodejs

Es un entorno JavaScript del lado del servidor, basado en eventos. Node ejecuta JavaScript utilizando el motor V8, desarrollado por Google para uso de su navegador Chrome. Aprovechando el motor V8 permite a Node proporcionar un entorno de ejecución del lado del servidor que compila y ejecuta javascript a velocidades increíbles. El aumento de velocidad es importante debido a que V8 compila JavaScript en código de máquina nativo, en lugar de interpretarlo o ejecutarlo como bytecode. Node es de código abierto, y se ejecuta en Mac OS X, Windows y Linux.[10]

1.7.4. ExtJS

Sencha Ext JS

Sencha Ext JS (anterior ExtJS) es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas que además de flexibilizar el manejo de componentes de la página como el DOM, Peticiones AJAX, DHTML, tiene la gran funcionalidad de crear interfaces de usuario bastante funcionales. Creado por Jack Slocum, Brian Moeskau, Aaron Conran, Rich Waters. [11]

Ventajas

Una de las grandes ventajas de utilizar ExtJS es que nos permite crear aplicaciones complejas utilizando componentes predefinidos.

Evita el problema de tener que validar el código para que funcione bien en cada uno de los navegadores (Firefox, IE, Safari, Opera etc.).

El funcionamiento de las ventanas flotantes lo pone por encima de cualquier otro.

Relación entre Cliente-Servidor balanceada: Se distribuye la carga de procesamiento, permitiendo que el servidor pueda atender más clientes al mismo tiempo.

Eficiencia de la red: Disminuye el tráfico en la red pues las aplicaciones cuentan con las posibilidades de elegir que datos desea trasmitir al servidor y viceversa (Criterio este que puede variar con el uso de aplicaciones de pre-carga).

Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.

Para desarrollar este módulo web se utilizará el framework ExtJS que es la más poderosa biblioteca de desarrollo Web construido en JavaScript, tiene muy buena performance, un modelo de componentes limpio y arquitectura MVC. Esta librería Javascript que permite construir aplicaciones complejas en internet además de flexibilizar el manejo de componentes de la página como el DOM, Peticiones AJAX, DHTML, tiene la gran funcionalidad de crear interfaces de usuario bastante funcionales.

Esta librería nos provee de compatibilidad cross-browser (Internet Explorer 6+, Firefox 3.6+, Safari 4+, Chrome + y Opera) para crear aplicaciones de un atractivo visual sorprendente y, además, de una sólida estructura por detrás.

También es un requerimiento de la entidad ya que existe experiencia acumulada con este Framework.

1.8. Tendencias, metodologías y/o tecnologías actuales

1.8.1. Etapa de diseño y desarrollo

Metodología RUP

El Rational Unified Process o Proceso Unificado de Racional. Es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta y de mayor calidad para satisfacer las necesidades de los usuarios que tienen un cumplimiento al final dentro de un límite de tiempo y presupuesto previsible. Es una metodología de desarrollo iterativo que es enfocada hacia “diagramas de los casos de uso, y manejo de los riesgos y el manejo de la arquitectura” como tal.

El RUP mejora la productividad del equipo ya que permite que cada miembro del grupo sin importar su responsabilidad específica pueda acceder a la misma base de datos incluyendo sus conocimientos.[12]

UML: El lenguaje unificado de modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG).[13]

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. [13]

El lenguaje de modelado unificado (UML) fue creado para forjar un lenguaje de modelado visual común, semántico y sintácticamente rico para la arquitectura, diseño e implementación de sistemas de software complejos tanto estructural como de comportamiento. UML tiene aplicaciones más allá del desarrollo de software, como el flujo de procesos en la fabricación.

Es análogo a los planos utilizados en otros campos, y consta de diferentes tipos de diagramas. En conjunto, los diagramas UML describen el límite, la estructura y el comportamiento del sistema y los objetos dentro de él.[13]

1.8.2. Uso de lenguaje y tecnologías Web

1.8.2.1. Ajax

Ajax (Asynchronous JavaScript and XML) es un método de construcción de aplicaciones interactivas para la Web que procesan las solicitudes de los usuarios de inmediato. Ajax combina varias herramientas de programación incluyendo JavaScript, HTML dinámico (DHTML), Extensible Markup Language (XML), hojas de estilo en cascada (CSS), el DOM (Document Object Model) y el objeto Microsoft, XMLHttpRequest. [14]

Ajax permite que el contenido de las páginas Web se actualice inmediatamente cuando un usuario realiza una acción, a diferencia de una solicitud HTTP, durante la cual los usuarios deben esperar a que se cargue una página completamente nueva. [5]

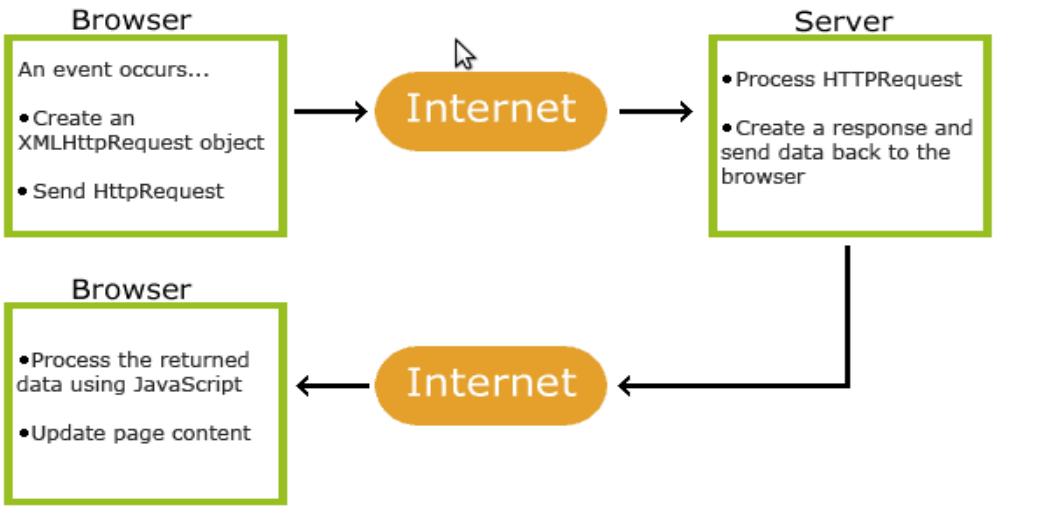


Figura 1 Como funciona AJAX

1.8.2.2. JavaScript

JavaScript, al igual que Flash, Visual Basic Script, es una de las múltiples maneras que han surgido para extender las capacidades del lenguaje HTML (lenguaje para el diseño de páginas de Internet). Al ser la más sencilla, es por el momento la más extendida. JavaScript no es un lenguaje de programación propiamente dicho como C, C++, Delphi, etc. Es un lenguaje script u orientado a documento, como pueden ser los lenguajes de macros que tienen muchos procesadores de texto y planillas de cálculo. No se puede desarrollar un programa con JavaScript que se ejecute fuera de un Navegador, aunque en este momento se expande a otras áreas, como la programación en el servidor con Node.js. JavaScript es un lenguaje interpretado que se embebe en una página web HTML. Un lenguaje interpretado significa que a las instrucciones las analiza y procesa el navegador en el momento que deben ser ejecutadas.[15]

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo y dinámico.

Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web. Desde el 2012, todos los navegadores modernos soportan completamente. JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

1.8.2.3. JSON (JavaScript Object Notation)

Es un formato para el intercambio de datos, básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos. JSON nació como una alternativa a XML, el fácil uso en JavaScript ha generado un gran número de seguidores de esta alternativa. Una de las mayores ventajas que tiene el uso de JSON es que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías. JSON es una sintaxis para serializar objetos, arreglos, números, cadenas, booleanos y nulos. Está basado sobre la sintaxis JavaScript. [16]

1.8.2.4. Servidor Web

La definición más sencilla de servidor web, es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc). Estos servidores web utilizan el protocolo http.

Los servidores web están alojados en un ordenador que cuenta con conexión a Internet. El web server, se encuentra a la espera de que algún navegador le haga alguna petición, como por ejemplo, acceder a una página web y responde a la petición, enviando código HTML mediante una transferencia de datos en red.[17]

Apache HTTP: Es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento.[17]

1.8.2.5. Patrón Modelo-Vista-Controlador (MVC)

MVC es una propuesta de diseño de software utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.[18]

Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores, o lo que es lo mismo, Model, Views & Controllers, en inglés. [18]

Las grandes aplicaciones del lado del cliente han sido siempre difíciles de escribir, difícil de organizar y difícil de mantener. Tienden a crecer rápidamente sin control a medida que agrega más funcionalidad y desarrolladores a un proyecto. Ext JS viene

con una nueva arquitectura de aplicación que no sólo organiza su código, sino que también reduce la cantidad de código que se debe escribir.

La arquitectura de una aplicación que sigue un patrón como el MVC con modelos, vista y controladores definen:

- Un modelo como una colección de campos y sus datos (por ejemplo, un modelo de usuario con nombre de usuario y las contraseñas). Los Modelos saben cómo se persisten a través del paquete de datos, y puede estar vinculado a otros modelos a través de asociaciones. Los Modelos se utilizan normalmente con Stores para presentar los datos en las grillas y otros componentes.

- La Vista con cualquier tipo de componente – grids, trees and panels son todos ellos vistas.

- Controladores que son lugares especiales para poner todo el código que hace que la aplicación funcione – ya sea el renderizado de vistas, creación de instancias de Modelos, o cualquier otra lógica de aplicación.

1.8.3. Herramientas a Utilizar

1.8.3.1. Visual Paradigm

Visual Paradigm de UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML* ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. [19]

1.8.3.2. WebStorm

JetBrains Webstorm es un IDE profesional JavaScript que es compatible con una amplia gama de tecnologías modernas relacionadas con el lenguaje de programación JavaScript, HTML y CCS, además ofrece la experiencia completa para el desarrollo Web productivo.[20]

Entorno de desarrollo inteligente

El IDE entiende su proyecto y le ayuda a producir código de alta calidad de manera más eficiente, gracias al completamiento de código se detectan errores sobre la marcha, la navegación poderosa y refactorizaciones automatizadas.[19]

Soporte para las recientes tecnologías

WebStorm funciona bien con la mayoría de las tecnologías modernas y populares para el desarrollo web. Ofrece soporte de primera clase para AngularJS, ECMAScript 6, Compass y mucho más.[20]

WebStorm agiliza el flujo de trabajo mediante la integración con todo lo necesario para el desarrollo productivo. Puede analizar el depurador, VCS, terminales y otras herramientas desde el IDE. [20]

IDE Multiplataforma

WebStorm funciona en Windows, Mac OS o Linux con una única clave de licencia. No importa que sistema operativo prefiera, WebStorm garantiza la mejor experiencia de desarrollo.

1.9. Conclusiones

En este capítulo se investigó sobre los conceptos relacionados al desarrollo de la interfaz web, el uso de los cortafuegos, algunos sistemas que los implementan. Se estudiaron los lenguajes, tecnologías, herramientas y metodologías necesarias para el desarrollo del sistema, concluyendo utilizar RUP como metodologías de desarrollo de software, UML como lenguaje de modelado, Visual Paradigm como herramienta para modelar la aplicación, JavaScript como lenguaje de programación, JSON como lenguaje de intercambio de información con el Servidor.

Capítulo II

2.1 Introducción

Este capítulo está dedicado al proceso de desarrollo de software de la solución propuesta a través de los flujos de trabajos de RUP. Se elaborará el modelo del dominio para tener mejor perspectiva del proceso. Se definen los casos de uso del sistema y la descripción detallada de casos de uso, se extraen los requisitos funcionales y no funcionales, reglas del negocio. Se describen los diagramas de clases web, la interfaz del usuario, el tratamiento de errores, el diagrama de implementación y las propuestas de Reglas.

2.2 Modelo de Dominio

Un *Modelo de Dominio* es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea construcción del modelo de dominio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física.[21]

Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir.[21]

2.2.1. Definición de las entidades y los conceptos principales

En el modelo de dominio se definen las siguientes clases: Objeto, Zona, Firewall, ACL, Regla.
Objeto: Es un conjunto de las estaciones de red que van hacer objetos de control

Zona: Subredes conectadas a un Swicth o router

Firewall: es un dispositivo de hardware que permite la interconexión de ordenadores en red.

ACL son un conjunto de regla ordenadas que permiten controlar el flujo del tráfico en equipos de redes, tales como enrutadores y conmutadores.

Regla: Un elemento de control del tráfico en la red, tiene clasificador de tráfico (protocolo), tiene acción y clasificación.

2.2.2. Reglas del Negocio

Una regla de negocio posee varios parámetros de entrada, una validación entre éstos y de acuerdo a dicha validación, realiza una acción de salida.[22]

1. Una zona que pertenezca a un firewall puede tener dos ACL una de entrada y una de salida.
2. Para asignar una ACL el firewall tiene que estar asignado a una Zona.
3. Un objeto luego de asignarse a una zona no puede ser asignado a otra zona.
4. Un puerto que está en modo “acces” no puede existir en diferentes zonas.
5. Un firewall tiene que tener una ip única.

6. En una ACL el número de la regla no puede repetirse

2.2.3. Representación del Modelo del Dominio

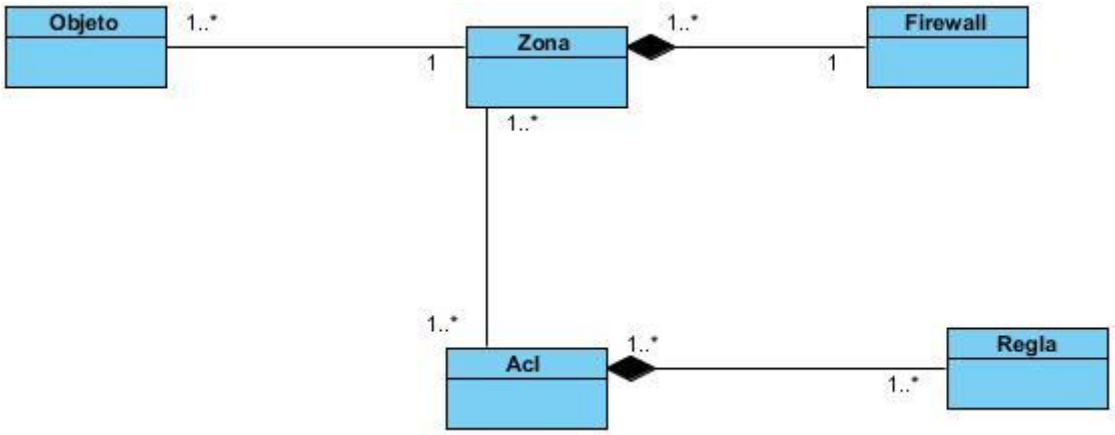


Figura 2 Modelo del Dominio

2.3 Requisitos

2.3.1. Concepción general del sistema

El presente proyecto tiene como objeto crear una Interfaz Web para el InterFirewall que facilite al administrador el control de varios cortafuegos

2.3.2. 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.[23]

Teniendo en cuenta los objetivos de los futuros usuarios del sistema y la descripción de cómo debe funcionar el mismo, se presenta los requerimientos funcionales siguientes.

R1. Autenticarse

R2. Cerrar Sesión

R3. Cambiar contraseña

R4. Insertar firewall

R5. Modificar firewall

R6. Eliminar firewall

R7. Insertar Zona

R8. Modificar Zona

R9. Eliminar Zona

R10. Insertar Objeto

R11. Modificar Objeto

R12. Eliminar Objeto

R13. Insertar Miembro

R14. Modificar Miembro

R15. Eliminar Miembro

R16. Insertar Servicios

R17. Modificar Servicios

R18. Eliminar Servicios

R19. Insertar Servicio Configuración

R20. Modificar Servicio Configuración

R21. Eliminar Servicio Configuración

R22. Insertar ACL

R23. Modificar ACL

R24. Eliminar ACL

R25. Insertar Regla

R26. Modificar Regla

R27. Eliminar Regla

R28. Mostrar lista Firewall

R29. Mostrar lista de zona

R30. Mostrar lista de ACL

R31. Mostrar Lista de Objeto

R32. Mostrar Lista de Miembros

R33. Mostrar lista Regla de una Acl especifica

R34. Asignar Zona a un Firewall

R35. Listar Puertos

R36. Insertar Usuario

R37. Eliminar Usuario

R38. Modificar Usuario

R39. Insertar Puertos

R40. Eliminar Puertos

R41. Modificar Puertos

R42. Listar Usuarios

2.3.3. Requisitos no funcionales:

Son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema[23]

De igual forma pueden ser determinados como los primeros requerimientos no funcionales los siguientes:

Apariencia

El software tiene que tener una interfaz amigable de fácil comprensión, que muestre las funcionalidades, descripción y datos que necesite el cliente para poder hacer uso de estos. Toda la interfaz gráfica, y los mensajes de la aplicación tienen que estar en el idioma Ingles, predominara el color azul, blanco y se mostrara el logo de la Entidad.

Usabilidad.

El sistema estará dirigido a los administradores red para que a través del módulo web puedan conectarse al servidor y controlar el tráfico en la red.

Rendimiento

La aplicación debe cumplir con los requisitos funcionales en el menor tiempo posible ante las solicitudes de informaciones al servidor por parte del administrador.

Soporte.

El sistema se documentará con un manual de ayuda con el objetivo de explicar su uso y garantizar el soporte del mismo. Se debe realizar el proyecto de forma tal que permita darle mantenimientos al sistema a fin de aumentar las funcionalidades y/o corregir los errores del mismo a través de versiones posteriores.

Portabilidad

El sistema debe correr en sistemas operativos Windows, Unix y Linux, utilizando un navegador web que implemente la especificación ECMA Script version5, que permite la ejecución de código javascript.

Legales

El sistema propuesto debe cumplir con las regulaciones y normas indicadas oficialmente en la División Territorial de ETECSA en la provincia de Cienfuegos.

Ayuda y documentación en línea: La ayuda del sistema se ofrece por medio de un manual de usuario , con ejemplos ilustrativos de cada funcionalidad, accesible desde la aplicación y una ayuda rápida en cada componente del subsistema.

Seguridad y privacidad

El sistema estará capacitado para la autenticación de un usuario, controlando la entrada por personas no autorizadas.

La información almacenada deberá ser consistente, y se minimizará la entrada de datos irreales mediante la utilización de validaciones.

La aplicación debe contar con la seguridad necesaria para proteger las configuraciones de la red

Software:

Debe tener un navegador web y la librería ExtJS

Hardware

El servidor tiene que tener una capacidad de memoria RAM no menor de 1GB, capacidad de almacenamiento con no menos de 20GB, un microprocesador de mínimo 2 núcleos a 2,8 MHz.

2.3.4. Modelo de caso de uso del Sistema

El modelo de casos de uso permite que los desarrolladores del software y los clientes lleguen a un acuerdo sobre los requisitos, es decir, sobre las condiciones y

posibilidades que debe cumplir el sistema. El modelo de casos de uso sirve como acuerdo entre clientes y desarrolladores, y proporciona la entrada fundamental para el análisis, el diseño y las pruebas. [24]

2.3.4.1 Actor del Sistema

Se le llama actor a toda entidad externa al sistema que guarda una relación con éste y que le demanda una funcionalidad. Esto incluye a los operadores humanos, pero también incluye a todos los sistemas externos.[25]

Tabla 1 Actor Del Sistema

Actor	Descripción
Administrador	Es el encardado de manejar la aplicación, es decir tiene todos los permisos tanto de escritura como lectura y modificación. Su objetivo principal es el manejo de los cortafuegos

2.3.4.2 Diagrama de caso de uso del sistema

Establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. Artefacto narrativo que describe, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario.

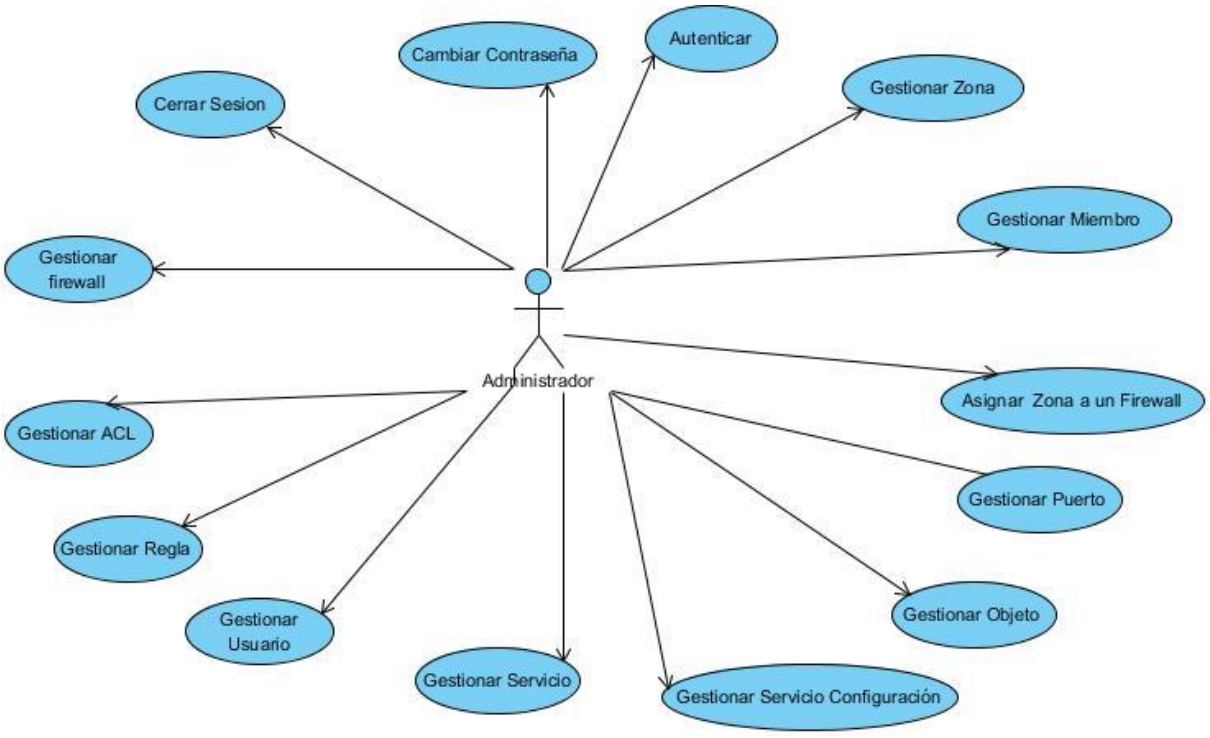


Figura 3 Caso de Uso del Sistema

2.3.4.3 Descripción de los casos de uso del sistema.

A continuación, se muestran las descripciones de los casos de uso del sistema en forma de tablas.

Tabla 2 Descripción de los casos de uso del sistema

Caso de uso del sistema	Descripción
Caso de Uso Autentificarse	Anexo B2

Caso de Uso Cerrar Sesión	Anexo B3
Caso de Uso Cambiar Contraseña	Anexo B4
Caso de Uso Gestionar Firewall	Anexo B5
Caso de Uso Gestionar Zona	Anexo B6
Caso de Uso Gestionar ACL	Anexo B7
Caso de Uso Gestionar Regla	Anexo B8
Caso de Uso Gestionar Usuario	Anexo B9
Caso de Uso Gestionar Objeto	Anexo B10
Caso de Uso Gestionar Miembros	Anexo B11
Caso de Uso Gestionar Servicio	Anexo B12
Caso de Uso Gestionar Servicio Configuración	Anexo B13

Caso de Uso Gestionar Puerto	Anexo B14
Caso de Uso Asignar Zona	Anexo B15

2.4 Diseño

2.4.1 Diagrama de Clase de Diseño Web

Caso de usos	Diagrama de clases Web
Autentificar Usuario	Anexo A1
Caso de Uso Cambiar Contraseña	Anexo A2
Caso de Uso Cerrar Sesión	Anexo A3
Caso de Uso Gestionar Firewall	Anexo A4
Caso de Uso Gestionar Zona	Anexo A5
Caso de uso Gestionar ACL	Anexo A6
Caso de uso Gestionar Regla	Anexo A7

Caso de uso Gestionar Objeto	Anexo A8
Caso de Uso Gestionar Miembros	Anexo A9
Caso de uso Gestionar Servicio	Anexo A10
Caso de Uso Gestionar Servicio Configuración	Anexo A11
Caso de Uso Asignar a un Firewall Zona.	Anexo A12
Caso de Uso Gestionar Puertos	Anexo A13
Caso de Uso Gestionar Usuarios	Anexo A14

Tabla 3 Diagrama de clases web

2.4.2 Diagrama de implementación

Los diagramas de implementación ofrecen una ilustración de la arquitectura física del hardware, del software y de los artefactos del sistema. Los diagramas de implementación pueden entenderse como lo contrario de los casos de uso, porque ilustran la forma física del sistema, en lugar de representar conceptualmente los usuarios y dispositivos que interactúan con el sistema. [26]

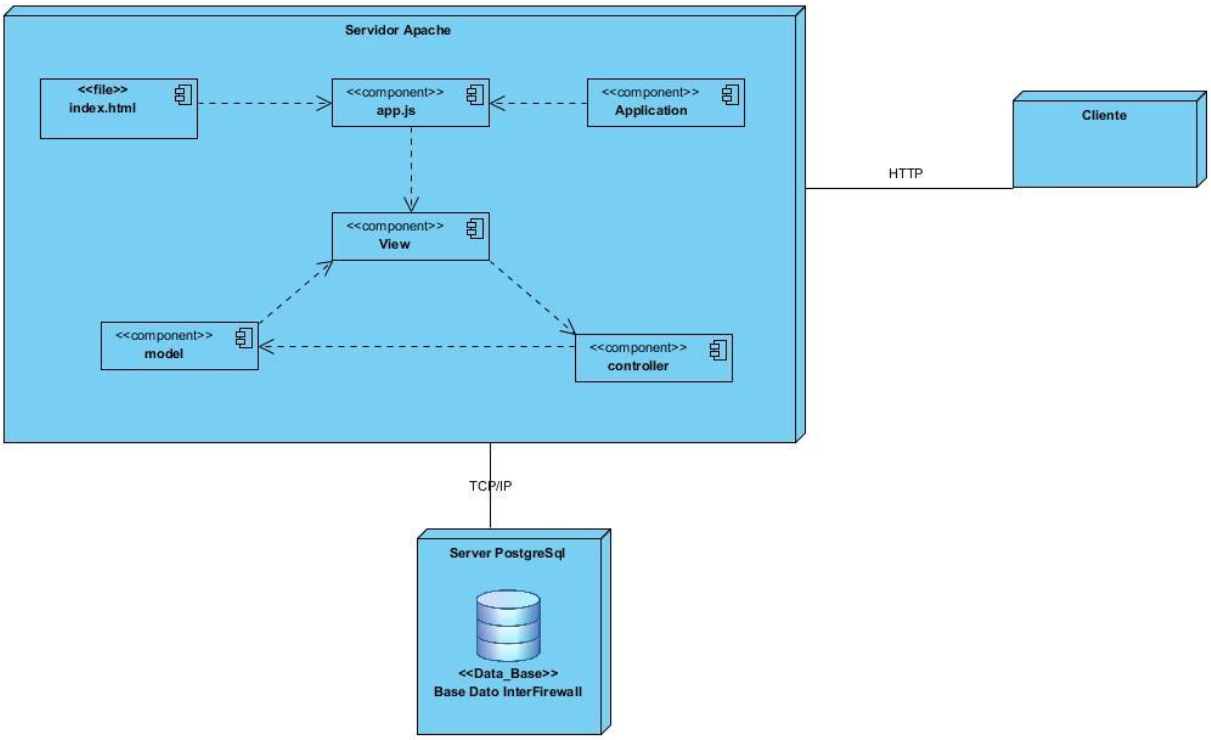


Figura 4 Diagramas de implementación

2.4.3 Principios de Diseño del Sistema (Arreglar Revisar)

Para desarrollar un sistema de software con calidad se debe prestar cuidadosa atención al diseño, que tiene como objetivo modelar el sistema para que soporte todos los requisitos, tanto funcionales como no funcionales. Además, permite al equipo de desarrollo de software evaluar la calidad del software antes de implementarlo.

2.4.4 Patrón de Diseño

Experto: es el principio básico de asignación de responsabilidades. A una clase le corresponde la está en la mayor responsabilidad de lo referente a un objeto.

Creador: asigna responsabilidades de creación de un objeto a la clase que contiene la información necesaria para crearlo.

Alta cohesión: expresa que la información que almacena una clase debe ser coherente y en la medida de lo posible relacionada con la clase.

Bajo acoplamiento: define que las clases deben depender lo menos posible una de otra, para que las modificaciones resulten simples y limpias.

Estos patrones se aplican en la definición de cada clase con el objetivo de lograr que las clases realicen las funciones relacionadas con la información que maneja para que no se sobrecarguen. Además, las clases se diseñan de forma tal que estén lo menos relacionadas entre sí para que los cambios que puedan ocurrir en una clase repercutan lo menos posibles en las otras. Se definieron las clases controladoras, las cuales son las responsables de gestionar las operaciones del sistema.

Interfaz de usuario

Para la creación de la interfaz de usuario del sistema se utilizó la propia del framework Extjs6 realizando modificaciones en el menú, la página principal y los paginados. En la interfaz de usuario se utilizan principalmente el color azul y el blanco, con fuentes blancas para lograr un contraste que facilite la lectura y sea agradable a la vista del usuario, en la zona izquierda de la pantalla se encuentran situadas todas las funcionalidades disponibles en el sistema.

La interfaz de usuario se puede dividir en las siguientes partes fundamentales:
Banner: Usualmente usado en los sitios web, muestra el nombre del sitio.
Barra de navegación: permite navegar entre las distintas partes del sitio
Contenido: En el contenido se muestra la página actual que fue accedida por

la barra de navegación.

2.4.6 Ayuda

La ayuda constituye una parte importante e imprescindible en todo sistema. En el menú principal, situado en la parte inferior derecha de la pantalla, aparece un icono de Ayuda que explicará al usuario de forma detallada cómo funciona el sistema, tratando de aclarar todas las funcionalidades que en el mismo se pueden realizar. Cada una de las opciones del sistema, así como las consideraciones que se asumen en la ejecución de ellas, está propiamente documentada para evitar cualquier tipo de confusión por parte del usuario. Cada aspecto de la ayuda está diseñado con el objetivo de expresar explícitamente cómo y en qué orden debe maniobrar el usuario.

2.4.8 Tratamiento de errores

El tratamiento de errores es uno de los aspectos más importantes a tener en cuenta durante el desarrollo del sistema. La validación de la información garantiza la corrección y precisión de todos los valores introducidos en la aplicación, además de lograr elevar la calidad de la misma. En todos los lenguajes de programación, pueden existir situaciones en las que el programa falle, estos errores se conocen como excepciones. Para capturar las excepciones se emplean las instrucciones try y catch. Entre las llaves de try se escribe el código que hará funcional el programa.

2.5 Propuestas de Reglas

Red_Usuario: se permiten los puertos 80 y 443 https para que tengan acceso a las páginas web.

Número	Acción	protocolo	Origen	destino	Puerto destino
1	permitir	Tcp	Subred_usuario	-	80
2	permitir	Tcp	Subred_usuario	-	443
3	denegar	Ip	Subred_usuario	Gestnet	
4	denegar	Ip	Subred_usuario	Servidores	
5	denegar	Ip			

Tabla 4 Red Usuario

Red administradora: los administradores a través del puerto

22/TCP open ssh8 para inspeccionar en modo comando en los servidores y operarlos, proporcionando usuario y contraseña.

3389/TCP open ms-term-serv para acceder al escritorio del servidor remoto

Número	Acción	Protocolo	Origen	Destino	Puerto destino
1	Permitir	Tcp	Red_Administrador	servidores	22

2	Permitir	Tcp	Red_Administrador	servidores	3389
3	Denegar	Ip			

Tabla 5 Red Administrador

Red Gestnet: son las máquinas q se conectan a las plantas digitales telefónicas, tiene que estar aislada de la red de usuario, porque los usuarios están conectados a internet y esta red de Gestnet pudiera infestarse de cualquier virus.

Entrada Gestnet

Número	Acción	Protocolo	Origen	Destino	Puerto destino
1	Denegar	Ip	Red_usuario	Gestnet	-

Tabla 6 Red Gestnet

Red DMZ (Zona desmilarizada): es una zona segura que se ubica entre la red interna de una organización y una red externa, generalmente en Internet.

Desde la red de usuario a la red Gestnet no se tiene acceso y en inversa tampoco, ambas si acceden a la red DMZ.

Número	Acción	Protocolo	Origen	Destino	Puerto Destino

1	Permitir	Ip	Red_usuario	DMZ	
2	Permitir	Ip	Red_Gestnet	DMZ	
3	Denegar	Ip			

Tabla 7 Red DMZ

2.6 Conclusiones

En este capítulo se realizó el modelado del dominio según lo establecido por la metodología de desarrollo de software RUP, así mismo se detallaron los artefactos de este flujo de trabajo tales como los diagramas de modelo del dominio, diagramas de clase de web y diagrama de implementación para una mejor comprensión de los procesos. Además, se determinaron los requerimientos funcionales con un total de 42, para luego realizar el diagrama de casos de uso del sistema para una visión general del software a desarrollar. Se asignaron responsabilidades al actor del sistema y se determinaron también los requerimientos no funcionales para conocer cualidades o propiedades del software. Se realizó una propuesta de reglas para la interfaz web que permite un control más seguro en el flujo de paquete que pasan por los cortafuegos.

Capítulo III

3.1 Introducción

En el presente capítulo se realiza el estudio de factibilidad del proyecto utilizando el método de estimación de Puntos de Casos de usos. Realizando una estimación del tiempo y el esfuerzo humano requerido para la realización del proyecto. Se determinan los costos asociados al desarrollo del software dependiendo del tiempo estimado por el método. También se diseñan los casos de prueba funcionales para realizar un estricto control de la calidad durante el proceso de desarrollo del software y se realiza la validación del mismo.

3.1.1 Escenario #1

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

Teniendo ubicado el intruso en la red, se dio paso a configurar el Cortafuego mediante la aplicación, modificando la política aplicada, en entrada, a la zona Usuarios, insertándole una regla para bloquear el acceso de dicha PC (ver Anexo C1).

3.1.2 Escenario #2

La empresa ETECSA cuenta con una infraestructura de red dividida en varias zonas, se limita el tráfico entre ellas, para una mejor seguridad de la red. Entre esas zonas se encuentra la zona Servidores, la zona Usuarios y la zona Administradores, se

quiere establecer una conjunto de reglas para que solo tengan acceso a los servidores los administradores, los cuales serán los únicos autorizados a configurar dichos servidores mediante pruebas de monitorización (ver Anexo C2).

A través de los escenarios de pruebas se pudo constatar que el sistema cumple con el objetivo para el cual fue implementado.

3.2 Verificación y Validación de Software

Es un conjunto de procesos de comprobación y análisis que aseguran que el software que se desarrolla está acorde a su especificación y cumple las necesidades de los clientes. [27]

Los objetivos de las actividades de verificación y validación son valorar y mejorar la calidad de los productos del trabajo generados durante el desarrollo y modificación del software. [27]

3.2.1. Caso de Prueba “Registrar Usuario”

Interfaz: Página de Principal

Descripción: Permite entrar al sistema si ya es un usuario registrado

Validaciones: La validación ocurre al hacer click sobre el botón Login

El campo Usuario no puede estar vacío.

El campo Usuario debe existir previamente en el sistema.

El campo Contraseña no puede ser nulo.

El botón login se habilita cuando el campo usuario y contraseña se llenan

Ver Anexo D1

3.2.2. Caso de Prueba “Crear Puerto”

Interfaz: Página de Crear Puerto

Descripción: Crear un puerto físico

Validaciones: La validación ocurre al hacer click sobre el botón Save

El campo Number no puede estar vacío.

Ver Anexo D2

3.2.3. Caso de Prueba “Cambiar Contraseña”

Interfaz: Página de Cambio de Contraseña

Descripción: Permite a un Usuario cambiar su contraseña de entrar al sistema.

Validaciones: La validación ocurre al hacer click sobre el botón Cambiar

El campo old Password debe ser igual a la contraseña del Usuario actual.

Los campos Password y Contraseña Nueva Repetida deben ser idénticos.

Todos los campos son requeridos

Ver AnexoD3

3.2.4. Caso de Prueba “Cuando no existe conexión con el servidor”

Interfaz: Página de Principal

Descripción: Permite a un Usuario ver los datos de servidor.

Validaciones: La validación ocurre al hacer click sobre el botón Login si el servidor no esta levantado nos envía un error de Failure

Ver Anexo D4

3.2.5. Caso de Prueba “Asignar Zona”

Interfaz: Página de Asignar Zona

Descripción: Si cuando se le asigna una zona al firewall no se le inserta un Gateway

con la estructura que se debe el servidor nos lanza un error

Ver Anexo D5

3.2.6. Caso de Prueba “Servicio Configuración”

Interfaz: Página de Servicio Configuración

Descripción: Permite a un Usuario crear una Configuración a un servicio.

Validaciones: La validación comienza cuando se llenan los campos

Todos los campos son requeridos.

En el rango de puerto Port Destination tiene que ser mayor que Port source

Ver AnexoD6

3.2.7. Caso de Prueba “Member”

Interfaz: Página de Member

Descripción: Permite a un Usuario crear estaciones de red.

Validaciones: La validación comienza cuando se llenan los campos

Todos los campos son requeridos.

Los campos from y to tiene que estar en el formato de un ip (###.###.###.###),cada número menor que 255

Ver Anexo D7

3.2.9. Caso de Prueba "Firewall"

Interfaz: Página de Firewall

Descripción: Permite a un Usuario insertar un nuevo Firewall.

Validaciones: La validación comienza cuando se llenan los campos

Todos los campos son requeridos.

El campo ip tiene que estar en el formato de un ip (###.###.###.###),cada número menor que 255

Ver Anexo D8

3.2.10 Caso de prueba Adicionar reglas a la ACL

Interfaz: Pagina de ACL

Descripción : Cuando se va a adicionar una regla a la ACL

Validación :

no pueden dos reglas tener el mismo número de regla

todos los campos son requerido

ver Anexo D9

3.3 Estimación por puntos de casos de uso

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.[28]

3.3.1 Cálculo de Puntos de Casos de Uso sin Ajustar

En el presente capítulo se realiza el estudio de factibilidad del proyecto utilizando el método de estimación de Puntos de Casos de usos. Realizando una estimación del tiempo y el esfuerzo humano requerido para la realización del proyecto. Se determinan los costos asociados al desarrollo del software dependiendo del tiempo estimado por el método. También se diseñan los casos de prueba funcionales para realizar un estricto control de la calidad durante el proceso de desarrollo del software y se realiza la validación del mismo.

El primer paso es el cálculo de los puntos de casos de uso sin ajustar. Este valor se calcula a partir de la siguiente ecuación

UUCP = UAW + UUCW

Conociendo que:

UUCP: Puntos de casos de uso sin ajustar

UAW: Factor de peso de los actores sin ajustar

UUCW: Factor de peso de los casos de uso sin ajustar

Ecuación para el cálculo de los puntos de casos de uso sin ajustar:

3.3.1 Factor de Peso de los Actores sin ajustar.

El factor de pesos de los actores sin ajustar se realiza mediante un análisis de la cantidad de actores y la complejidad que posee cada uno y el factor de peso correspondiente.

Actor	*Tipo de Actor*	*Factor Peso*
Administrador	*Complejo*	*3*

Tabla 8 Clasificación de los Actores de Sistema

El actor es complejo ya que interactúa con el sistema mediante una interfaz gráfica. Multiplicando la cantidad de actores de cada tipo por el peso correspondiente se obtiene:

Tabla 3-2 Cantidad de Casos Agrupados por Clasificación

UAW = Cantidad de Actores * Factor de Peso

UAW = 1 * 3

UAW = 3

3.2.2 Factor de Peso de los Casos de Uso sin Ajustar(UUCW)

Este valor se calcula mediante un análisis de la complejidad de los casos de uso sin ajustar existentes en el sistema, esta complejidad está dada por la cantidad de transacciones que se realizan, cuando el caso de uso tiene de 1 a 3 transacciones es simple y tiene un factor de peso igual a 5, cuando tiene de 4 a 7 es medio y tiene un factor de peso igual a 10 y cuando tiene más de 8 transacciones el caso de uso es complejo y tiene un factor de peso de 15.

A continuación, se muestra una tabla con los casos de uso y le tipo

Casos de uso	Tipo
Autenticar	Simple
Cambiar Contraseña	Simple
Cerrar Sesión	Simple
Gestionar Firewall	Medio

Gestionar ACL	Medio
Gestionar Regla	Medio
Gestionar Usuario	Simple
Gestionar Servicio	Simple
Gestionar Servicio Configuración	Medio
Gestionar Objeto	Medio
Asignar Zona a un Firewall	Simple
Gestionar Firewall	Medio
Gestionar Miembro	Medio
Gestionar Zona	Medio

Tabla 9 Descripción de los casos de uso del sistema según el tipo

En la tabla de clasificación anterior se muestra que el sistema está compuesto por 15 Casos de Uso, 8 de ellos tienen clasificación media y 6 clasificación simple. Por lo que el factor de Peso de los Casos de uso sin ajustar es igual a:

UUCW= 8*10 + 6*5

UUCW= 80+30

UUCW=110

Por lo que sustituimos el valor de factor de peso delos actores sin ajustar y de los puntos de casos de uso sin ajustar se obtiene que:

UUCP = UAW + UUCW

UUCP = 110 +3

UUCP = 113

3.4 Cálculo de Puntos de Casos de Uso Ajustados

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

UCP = UUCP * TCF * EF

Donde:

UCP: Puntos de Casos de Uso ajustado

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad Técnica

EF: Factor de ambiente

3.4.1 Factor de complejidad técnica (TCF)

Factor	*Descripción*	*Peso*	*Valor Cuantificado*	*Peso * Valor cuantificado*
T1	*Sistema distribuido*	*2*	*2*	*4*
T2	*Objetivos de performance o tiempo de respuesta.*	*1*	*4*	*4*
T3	*Eficiencia del usuario final*	*1*	*4*	*4*
T4	*Procesamiento interno complejo*	*1*	*2*	*2*
T5	*El código debe ser reutilizable*	*1*	*4*	*4*

T6	*Facilidad de instalación*	*0.5*	*4*	*2*
T7	*Facilidad de uso*	*0.5*	*3*	*1.5*
T8	*Portabilidad*	*2*	*4*	*8*
T9	*Facilidad de cambio*	*1*	*3*	*3*
T10	*Concurrencia*	*1*	*2*	*2*
T11	*Incluye objetivos especiales de seguridad*	*1*	*3*	*3*
T12	*Provee acceso directo a terceras partes.*	*1*	*3*	*3*
T13	*Se requieren facilidades*	*1*	*4*	*4*

	especiales de entrenamiento a los usuarios			

Tabla 10 Factor de Complejidad Técnica

Por lo que el Factor de Complejidad Técnica resulta:

*TCF = 0.6+0.01 ** **Σ (Pesoi *Valor asignadoi)**

TCF = 0.6 +0.01 * (4+4+4+2+4+2+1.5+8+3+2+3+3+4)

TCF=0.6 + 0.01 * 44.5

TCF = 1.045

3.4.2 Factor Ambiente (EF)

El Factor ambiente se calcula atendiendo a las habilidades y el entrenamiento de la persona o grupo involucrado.

Factor	*Descripción*	*Peso*	*Valor cuantificado*	*Peso * valor cuantificado*
E1	*Familiaridad con el modelo de proyecto utilizado*	*1.5*	*4*	*6*

E2	*Experiencia en la aplicación*	*0.5*	*3*	*1.5*
E3	*Experiencia en orientación a Objetos*	*1*	*4*	*4*
E4	*Capacidad del analista líder*	*0.5*	*3*	*1.5*
E5	*Motivación*	*1*	*5*	*5*
E6	*Estabilidad de los Requerimiento*	*2*	*5*	*10*
E7	*Todo a tiempo completo*	*-1*	*1*	*-4*
E8	Dificultad del lenguaje de programación	*-1*	*1*	*-3*

Tabla 11 Factor ambiente

Σ (Pesoi * Valori) = 6+1.5+4+1.5+5+10-1-1 =6

EF = 1.4 – 0.03*26

EF = 0.62

Los puntos de caso de uso ajustado resultan

UCP = UUCP*TCF*EF

UCP = 113 * 1.045 * 0.62

UCP = 73.2127

3.5 De los Puntos de Casos de Uso a la estimación del esfuerzo:

CF: Factor de Conversión

CF = 20 Horas/Hombre

El esfuerzo por horas hombre está dado por:

E = UCP * CF

E = 73.2127* 20

E=1464.254 Horas/Hombres

Duración: Trabajando 25 días al mes y 8 horas al día como promedio, se obtiene: Duración (días) =Total de horas /hombre entre 8 horas al día =1464.254/8 = 183.03 días Duración (meses)=Total de días /25 días por mes = 183.03175 /25 = 7.32 meses ≈ 7 meses. El Proyecto se realiza en 7 meses aproximadamente

Actividad	*%Esfuerzo*	*Valor Esfuerzo*
Análisis	*10*	183.03
Diseño	*20*	*366.06*
Implementación	*40*	*732.12*
Prueba	*15*	*274.55*
Sobrecarga	*15*	*274.55*
Total	*100*	*1800.3*

Tabla 12 Estimación de Esfuerzo

3.6 Cálculo de Costo

Tomando un salario mensual de $1500.00

Costo=7 meses * $1700 mensual= $12000

3.7 Análisis de los costos y beneficios

El sistema informático como resultado de la presente investigación no implica costo alguno para la división territorial de ETECSA Cienfuegos, sin embargo, al desarrollo

de todo producto informático va asociado un costo y su justificación económica viene dado por los beneficios tangibles e intangibles que este produce.

Utilizar este nuevo software permite al personal encargado de administrar la Red de la División territorial de ETECSA Cienfuegos, gestionar la información referente a los Cortafuegoss, así como todo el flujo de información que fluye al crear estos, tal es el caso de las ACL (Lista de Control de Acceso) que posibilita mediante reglas ordenadas se deniegue o permita el flujo de datos por diferentes segmentos de red, todo esto de manera rápida y fiable. Para el desarrollo de este sistema no fue necesario invertir en medios técnicos. Estos beneficios implican un ahorro considerable del tiempo invertido en la gestión y control de esta información.

3.8 Conclusiones

La realización del cálculo de factibilidad económica para el sistema informático propuesto, estimó un tiempo de 7 meses para su desarrollo y un costo de $12000 aproximadamente. Se definieron los beneficios que trae consigo la implementación del sistema para la entidad, concluyendo de esta forma que es factible su desarrollo. Se diseñaron los casos de pruebas funcionales, permitiendo obtener una verificación dinámica del sistema, garantizando de esta forma el control de la calidad en el desarrollo del software.

Conclusiones

Tomando en consideración los objetivos planteados al inicio del trabajo, se arriban a las conclusiones siguientes:

- Se diseñó e implementó un sistema informático con funcionalidades que se adecuan a las necesidades de la institución, proporcionándole una herramienta que reduce el tiempo empleado en el proceso de gestión de los Cortafuegos que existen en la red, dando cumplimiento de esta forma a los requerimientos planteados.

- Se propusieron un conjunto de regla para la interfaz web facilitando así el trabajo de los administradores.

- El diseño de los casos de prueba funcional permitió obtener una verificación dinámica del comportamiento del sistema, garantizando de esta forma el control de la calidad en el desarrollo del software.

- Utilizando como método de estimación Puntos de Casos de Uso para determinar la factibilidad, se concluyó que fue factible desarrollar el sistema informático.

Recomendaciones

A pesar de que la investigación realizada cumplió con los objetivos trazados, se recomienda:

- Probar al máximo las funcionalidades que brinda el sistema durante un período amplio de tiempo, para comprobar de forma práctica todas sus funcionalidades y obtener los datos necesarios para su mejora.

- Generalizar el uso del sistema informático en todas las ciudades capitales.

- Continuar el estudio de los procesos asociados a la Calidad de vida subjetiva con el objetivo de añadirle nuevas funcionalidades al sistema.

Referencias Bibliografías

[1] V. M. Gomez, “El diseño de la interfaz :: Diseño de Paginas Web”, 23-may-2017. .

[2] “Cortafuegos informáticos: Qué son y para qué sirven - ComputerHoy.com”, 25-may-2017. [En línea]. Disponible en: http://computerhoy.com/noticias/internet/cortafuegos-informaticos-que-son-que-sirven-26747. [Consultado: 25-may-2017].

[3] jimdial, “¿Qué es una lista de control de acceso (ACL) de red?”, 31-may-2017. [En línea]. Disponible en: https://docs.microsoft.com/es-es/azure/virtual-network/virtual-networks-acl. [Consultado: 31-may-2017].

[4] “Cortafuegos”, 24-may-2017. [En línea]. Disponible en: https://pccito.ugr.es/cortafuegos/introduccion.html. [Consultado: 24-may-2017].

[5] “QUE ES EL ZENTYAL de fernanda samueza en Prezi”, 23-may-2017. [En línea]. Disponible en: https://prezi.com/oxocahx2zcj_/que-es-el-zentyal/. [Consultado: 23-may-2017].

[6] “Pfsense”, 03:40:21 UTC.

[7] “Que es Ipcop???? - Taringa!”, 24-may-2017. [En línea]. Disponible en: http://www.taringa.net/posts/linux/3587831/Que-es-Ipcop.html. [Consultado: 24-may-2017].

[8] DesarrolloWeb.com, “Qué es AngularJS”, *DesarrolloWeb.com*, 26-may-2017. [En línea]. Disponible en: http://www.desarrolloweb.com/articulos/que-es-angularjs-descripcion-framework-javascript-conceptos.html. [Consultado: 26-may-2017].

[9] michellebow2013, “¿Qué es y para qué sirve jQuery y HTML5?”, *QA En dispositivos moviles*, 04-may-2013. .

[10] “Node.js: ¿Qué es y para que sirve NodeJS?”, *Desarrollo web Alicante y Valencia | Marketing Online*, 30-sep-2015. .

[11]“Sencha Ext JS - EcuRed”, 25-may-2017. [En línea]. Disponible en: https://www.ecured.cu/Sencha_Ext_JS. [Consultado: 25-may-2017].

[12] “METODOLOGÍA RUP”, 23-may-2017. .

[13] “What is Unified Modeling Language”, *Lucidchart*, 11-oct-2016. [En línea]. Disponible en: https://www.lucidchart.com/pages/what-is-UML-unified-modeling-language. [Consultado: 23-may-2017].

[14] “What is Ajax (Asynchronous JavaScript and XML)? - Definition from WhatIs.com”, 23-may-2017. [En línea]. Disponible en: http://searchwindevelopment.techtarget.com/definition/Ajax. [Consultado: 23-may-2017].

[15] “Qué es JavaScript? - Tu Mentor en Programación”, 23-may-2017. [En línea]. Disponible en: http://tutoriales.tumentor.net/javascript/que-es-javascript. [Consultado: 23-may-2017].

[16] “JSON”, *Mozilla Developer Network*, 23-may-2017. [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/JSON. [Consultado: 23-may-2017].

[17] “Apache HTTP Server: ¿Qué es, cómo funciona y para qué sirve? | Blog ibrugor”. [En línea]. Disponible en: http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/. [Consultado: 04-abr-2017].

[18] DesarrolloWeb.com, “Qué es MVC”, *DesarrolloWeb.com*, 23-may-2017. [En línea]. Disponible en: http://www.desarrolloweb.com/articulos/que-es-mvc.html. [Consultado: 23-may-2017].

[19] “Visual Paradigm - EcuRed”. [En línea]. Disponible en: https://www.ecured.cu/Visual_Paradigm. [Consultado: 04-abr-2017].

[20] “WebStorm | Pyme Apps”, 25-may-2017. [En línea]. Disponible en: https://comprasoft.com/jetbrains/webstorm. [Consultado: 25-may-2017].

[21] “Modelo de Dominio | Tecnología y Synergix”. [En línea]. Disponible en: https://synergix.wordpress.com/2008/07/10/modelo-de-dominio/. [Consultado: 07-mar-2017].

[22] «Modelado de reglas de negocio: un enfoque práctico | ::everac99». [En línea]. Disponible en: https://everac99.wordpress.com/2010/04/10/modelado-de-reglas-de-negocio-un-enfoque-practico/. [Accedido: 02-jun-2017].

[23] “Requerimientos funcionales y no funcionales”, *Scribd*, 24-may-2017. [En línea]. Disponible en: https://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales. [Consultado: 24-may-2017].

[24] “Modelo de Casos de Uso del Sistema”. [En línea]. Disponible en: http://www.eumed.net/libros-gratis/2009c/585/Modelo%20de%20Casos%20de%20Uso%20del%20Sistema.htm. [Consultado: 08-mar-2017].

[25] “DEFINICION CASO DE USO, ACTORES Y ROLES”, 31-may-2017. .

[26] “Diagramas de implementación UML”, 25-may-2017. [En línea]. Disponible en: https://www.altova.com/es/umodel/uml-deployment-diagrams.html. [Consultado: 25-may-2017].

[27] R. González, “Ramón González: ¿Qué es la Verificación y Validación de Software?”, *Ramón González*, de agosto de-2012. .

[28] I. Jacobson, G. Booch, y J. Rumbaugh, *El Proceso Unificado de Desarrollo de Software*. .

Bibliografía

[1] “Apache HTTP Server: ¿Qué es, cómo funciona y para qué sirve? | Blog ibrugor”, 24-may-2017. .

[2] “Arquitectura MVC con ExtJS 4 | Aprender ExtJS”, 31-may-2017. [En línea]. Disponible en: http://www.marioperez.com.mx/extjs/mvc/. [Consultado: 31-may-2017].

[3] “Cómo diseñar una política de cortafuegos”, 30-may-2017. .

[4] F. Says, “Concepto de tipos de reglas del negocio”, *elConspirador*, 16-oct-2013. .

[5] “Concepto de tipos de reglas del negocio - elConspirador”, 24-may-2017. [En línea]. Disponible en: http://www.elconspirador.com/2013/10/16/concepto-de-tipos-de-reglas-del-negocio/. [Consultado: 24-may-2017].

[6] “Cortafuegos”, 24-may-2017. [En línea]. Disponible en: https://pccito.ugr.es/cortafuegos/introduccion.html. [Consultado: 24-may-2017].

[7] “Cortafuegos informáticos: Qué son y para qué sirven - ComputerHoy.com”, 25-may-2017. [En línea]. Disponible en: http://computerhoy.com/noticias/internet/cortafuegos-informaticos-que-son-que-sirven-26747. [Consultado: 25-may-2017].

[8] “Cree una regla de dominio”, 31-may-2017. [En línea]. Disponible en: https://msdn.microsoft.com/es-es/library/hh510397.aspx. [Consultado: 31-may-2017].

[9] “DEFINICION CASO DE USO, ACTORES Y ROLES”, 31-may-2017. .

[10] “Diagramas de implementación UML”, 25-may-2017. [En línea]. Disponible en:

https://www.altova.com/es/umodel/uml-deployment-diagrams.html. [Consultado: 25-may-2017].

[11] V. M. Gomez, “El diseño de la interfaz :: Diseño de Paginas Web”, 23-may-2017. .

[12] I. Jacobson, G. Booch, y J. Rumbaugh, *El Proceso Unificado de Desarrollo de Software*. .

[13] “¿ExtJS es el mejor Framework de JavaScript? - Programadores ExtJS”, 30-may-2017. [En línea]. Disponible en: http://www.extjs.mx/extjs-es-el-mejor-framework-de-javascript. [Consultado: 30-may-2017].

[14] M. Garcia, “ExtJS-Instalando Sencha CMD”, *Recursos para formacion*, 03-ago-2015. .

[15] “Forms | Ext JS 6.0.2”, *Sencha Documentation*, 31-may-2017. [En línea]. Disponible en: https://docs.sencha.com/extjs/6.0.2/guides/components/forms.html. [Consultado: 31-may-2017].

[16] “Grids | Ext JS 6.0.2”, *Sencha Documentation*, 31-may-2017. [En línea]. Disponible en: https://docs.sencha.com/extjs/6.0.2/guides/components/grids.html. [Consultado: 31-may-2017].

[17] Center for History and New Media, “Guía rápida”. [En línea]. Disponible en: http://zotero.org/support/quick_start_guide.

[18] “JSON”, *Mozilla Developer Network*, 23-may-2017. [En línea]. Disponible en: https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_global

es/JSON. [Consultado: 23-may-2017].

[19] “Mejores prácticas en Checkpoint Firewall – All about TI : Storages. S.O´s ,Virtualización & Networking”, 30-may-2017. [En línea]. Disponible en: https://storich.wordpress.com/2015/01/05/mejores-practicas-en-checkpoint-firewall/. [Consultado: 30-may-2017].

[20] “Mejores prácticas para configurar las reglas del Firewall”, 30-may-2017. [En línea]. Disponible en: http://liacolombia.com/2012/09/mejores-practicas-para-configurar-las-reglas-del-firewall/. [Consultado: 30-may-2017].

[21] “METODOLOGÍA RUP”, 23-may-2017. .

[22] “Microsoft PowerPoint - 2.14.MVC.ppt [Modo de compatibilidad] - 2.14.MVC.pdf”, 25-may-2017. [En línea]. Disponible en: https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf. [Consultado: 25-may-2017].

[23] “Modelo de Casos de Uso del Sistema”, 24-may-2017. [En línea]. Disponible en: http://www.eumed.net/libros-gratis/2009c/585/Modelo%20de%20Casos%20de%20Uso%20del%20Sistema.htm. [Consultado: 24-may-2017].

[24] “Modelo de Dominio”, *Tecnología y Synergix*, 10-jul-2008. .

[25] “Node.js: ¿Qué es y para que sirve NodeJS?”, *Desarrollo web Alicante y Valencia | Marketing Online*, 30-sep-2015. .

[26] “Paging Grid with Memory Proxy”, 29-may-2017. [En línea]. Disponible en: https://www.sencha.com/forum/showthread.php?261248-Paging-Grid-with-Memory-Proxy. [Consultado: 29-may-2017].

[27] rubenfa, “Patrones de diseño: qué son y por qué debes usarlos”, *Genbeta Dev*, 14-jul-2014. [En línea]. Disponible en: https://www.genbetadev.com/metodologias-de-programacion/patrones-de-diseno-que-son-y-por-que-debes-usarlos. [Consultado: 29-may-2017].

[28] “Pfsense”, 03:40:21 UTC.

[29] “Programación web » WebStorm”, 25-may-2017. [En línea]. Disponible en: http://multimedia.uoc.edu/blogs/pw/es/2013/04/16/webstorm/. [Consultado: 25-may-2017].

[30] DesarrolloWeb.com, “Qué es AngularJS”, *DesarrolloWeb.com*, 26-may-2017. [En línea]. Disponible en: http://www.desarrolloweb.com/articulos/que-es-angularjs-descripcion-framework-javascript-conceptos.html. [Consultado: 26-may-2017].

[31] “Qué es AngularJS”, 26-may-2017. [En línea]. Disponible en: https://desarrolloweb.com/articulos/que-es-angularjs-descripcion-framework-javascript-conceptos.html. [Consultado: 26-may-2017].

[32] “QUE ES EL ZENTYAL de fernanda samueza en Prezi”, 23-may-2017. [En línea]. Disponible en: https://prezi.com/oxocahx2zcj_/que-es-el-zentyal/. [Consultado: 23-may-2017].

[33] “Que es Ipcop???? - Taringa!”, 24-may-2017. [En línea]. Disponible en: http://www.taringa.net/posts/linux/3587831/Que-es-Ipcop.html. [Consultado: 24-may-2017].

[34] “Qué es JavaScript? - Tu Mentor en Programación”, 23-may-2017. [En línea]. Disponible en: http://tutoriales.tumentor.net/javascript/que-es-javascript. [Consultado: 23-may-2017].

[35] DesarrolloWeb.com, “Qué es MVC”, *DesarrolloWeb.com*, 23-may-2017. [En línea]. Disponible en: http://www.desarrolloweb.com/articulos/que-es-mvc.html. [Consultado: 23-may-2017].

[36] jimdial, “¿Qué es una lista de control de acceso (ACL) de red?”, 31-may-2017. [En línea]. Disponible en: https://docs.microsoft.com/es-es/azure/virtual-network/virtual-networks-acl. [Consultado: 31-may-2017].

[37] “¿Qué es un store y cómo funciona? - Quizzpot”, 29-may-2017. [En línea]. Disponible en: https://quizzpot.com/courses/aprendiendo-ext-js-3/articles/que-es-un-store-y-como-funciona. [Consultado: 29-may-2017].

[38] michellebow2013, “¿Qué es y para qué sirve jQuery y HTML5?”, *QA En dispositivos moviles*, 04-may-2013. .

[39] R. González, “Ramón González: ¿Qué es la Verificación y Validación de Software?”, *Ramón González*, de agosto de-2012. .

[40] “Recomendaciones para la gestión eficiente de un Firewall de red”, *Informática para tu negocio*, 13-dic-2016. .

[41] “Requerimientos funcionales y no funcionales”, *Scribd*, 24-may-2017. [En línea]. Disponible en: https://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales. [Consultado: 24-may-2017].

[42] “Sencha Ext JS”, 30-may-2017. [En línea]. Disponible en: https://es.scribd.com/document/103368557/Sencha-Ext-JS. [Consultado: 30-may-2017].

[43] “Sencha Ext JS - EcuRed”, 25-may-2017. [En línea]. Disponible en: https://www.ecured.cu/Sencha_Ext_JS. [Consultado: 25-may-2017].

[44] “Sencha Ext JS Tutorial: An Introduction to Ext JS 5 | Codementor”, 23-may-2017. [En línea]. Disponible en: https://www.codementor.io/extjs/tutorial/sencha-introduction-to-extjs-5. [Consultado: 23-may-2017].

[45] “Visual Paradigm para UML”, 24-may-2017. [En línea]. Disponible en: http://www.software.com.ar/p/visual-paradigm-para-uml. [Consultado: 24-may-2017].

[46] “WebStorm | Pyme Apps”, 25-may-2017. [En línea]. Disponible en: https://comprasoft.com/jetbrains/webstorm. [Consultado: 25-may-2017].

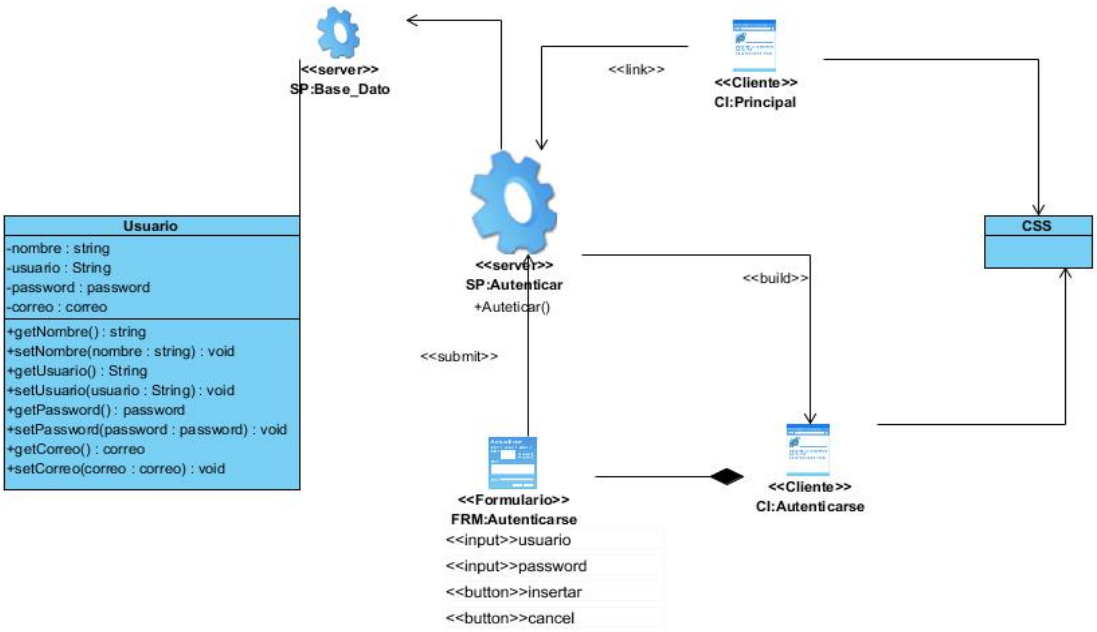
[47] “What is Ajax (Asynchronous JavaScript and XML)? - Definition from WhatIs.com”, 23-may-2017. [En línea]. Disponible en: http://searchwindevelopment.techtarget.com/definition/Ajax. [Consultado: 23-may-2017].

[48] “What is Unified Modeling Language”, *Lucidchart*, 11-oct-2016. [En línea]. Disponible en: https://www.lucidchart.com/pages/what-is-UML-unified-modeling-language. [Consultado: 23-may-2017].

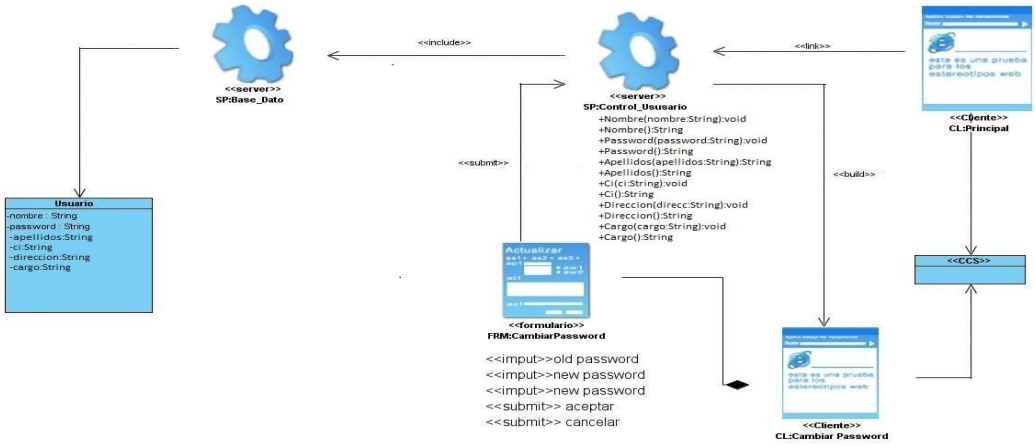
Anexos

Anexos A Diagramas de clases Web

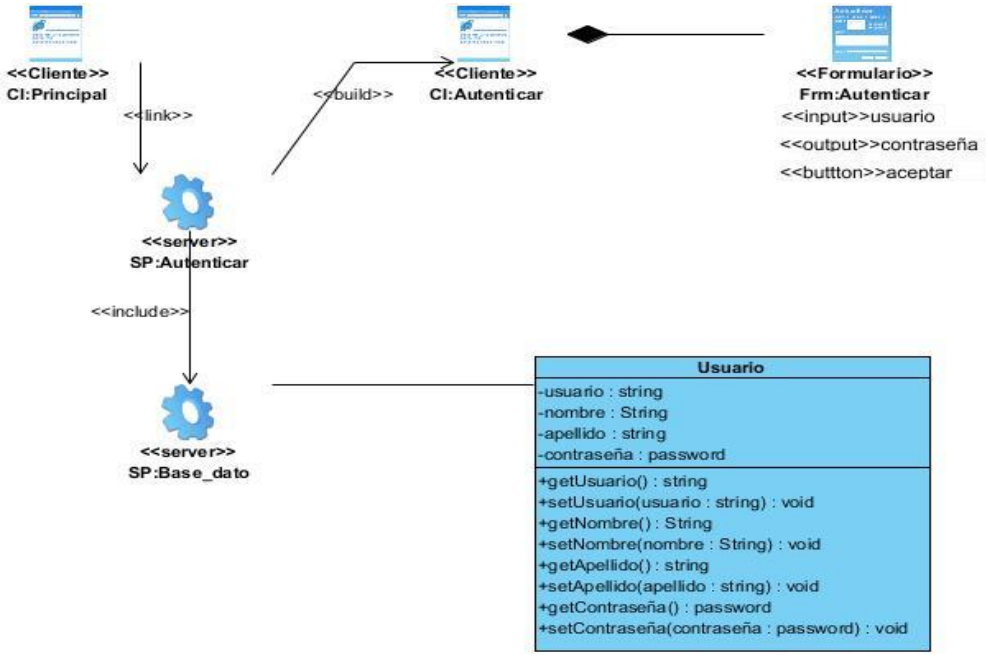
Anexo A1 Autentificar Usuario



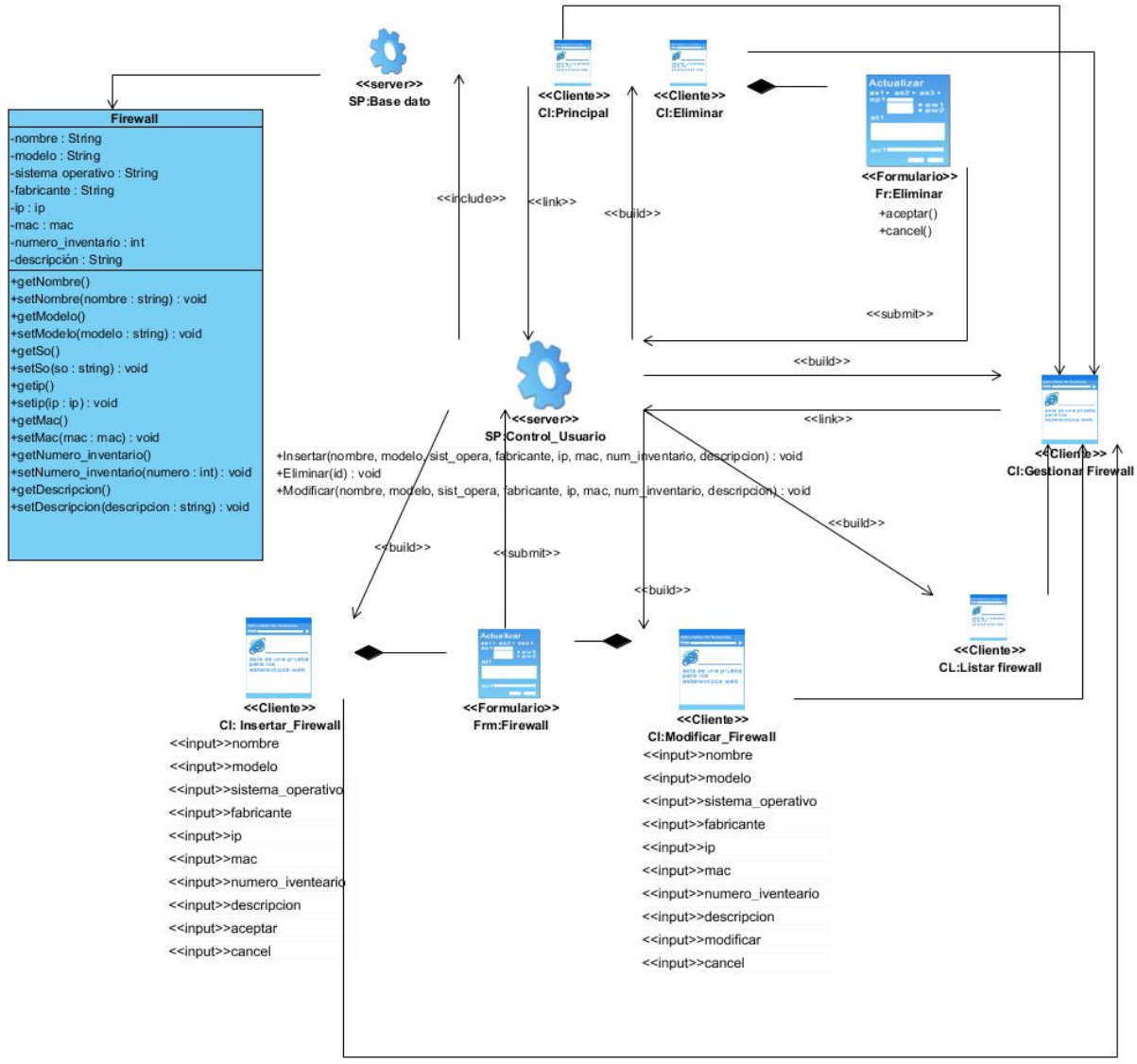
Anexo A2 Caso de Uso Cambiar Contraseña



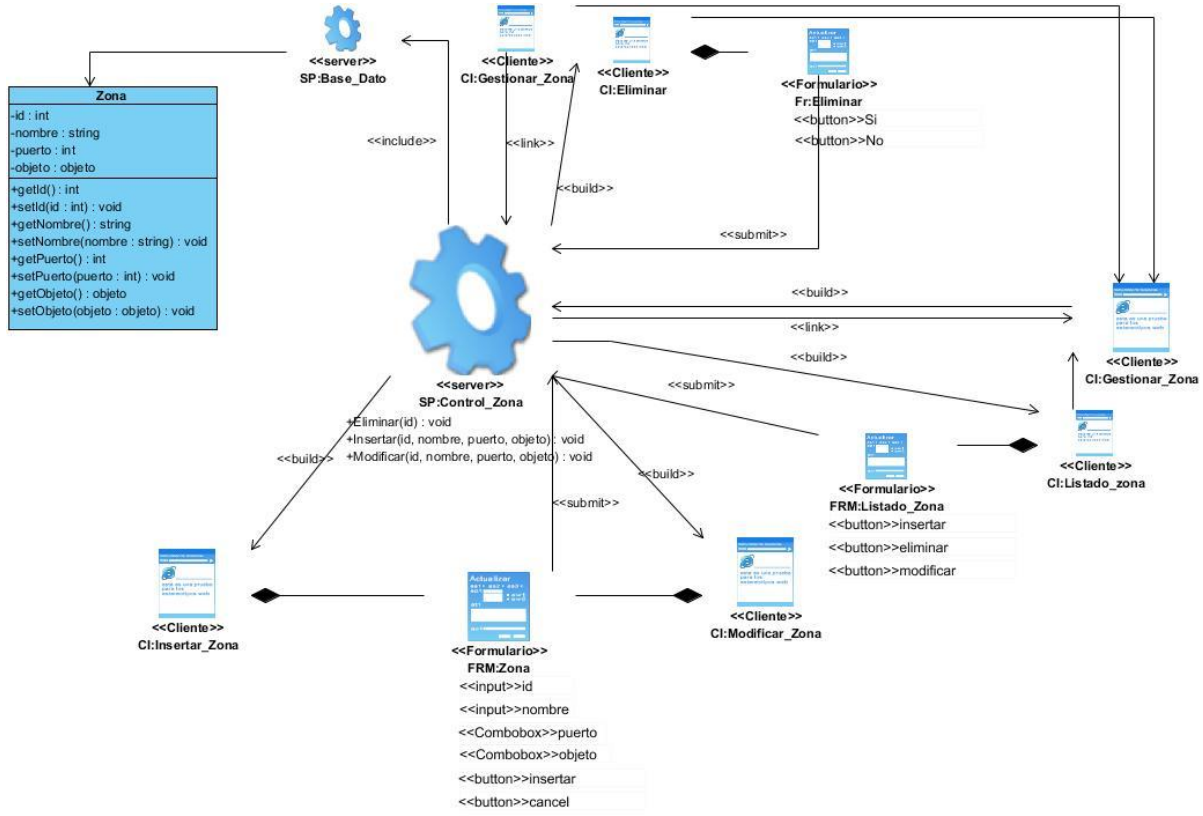
Anexo A3 Caso de Uso Cerrar Sesión



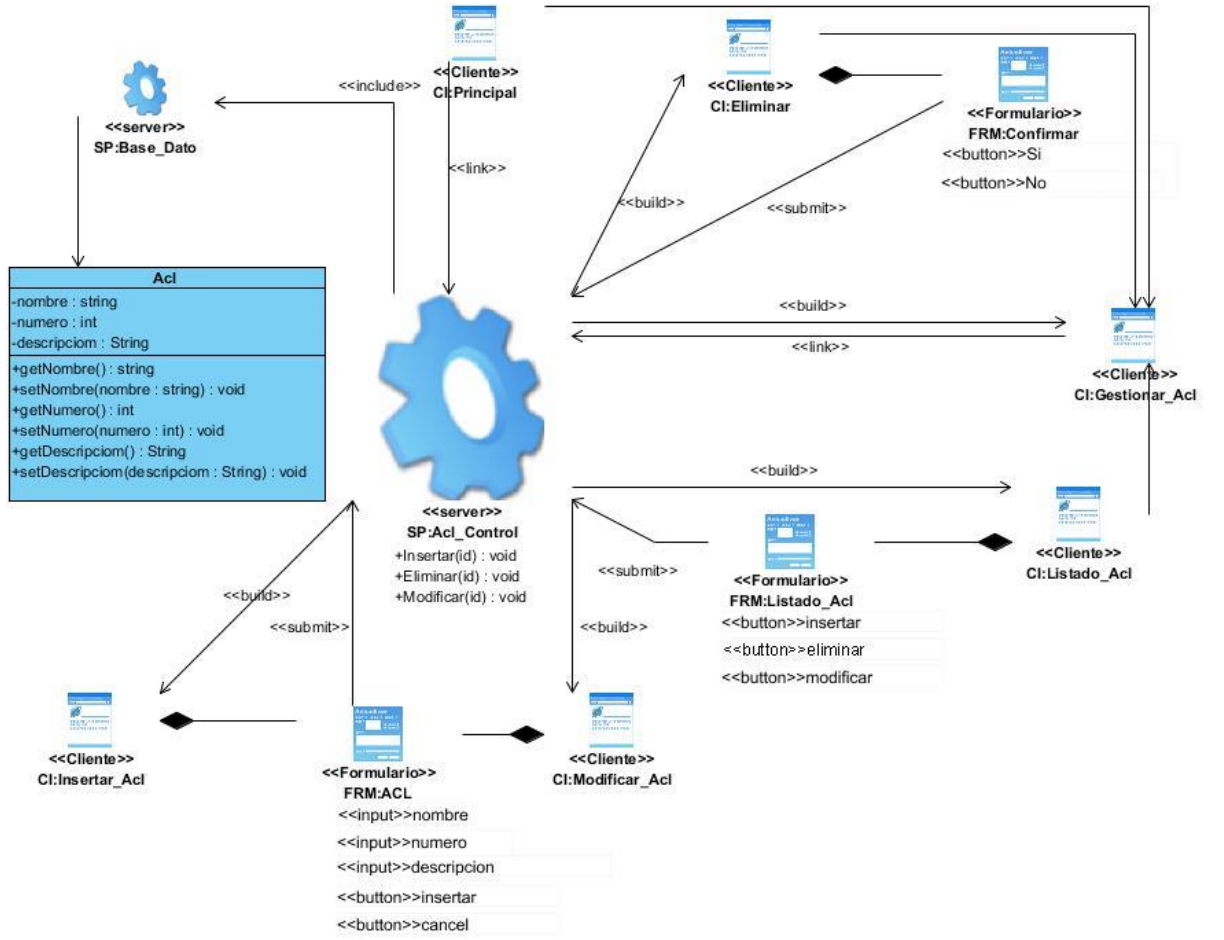
Anexo A4 Caso de Uso Gestionar Firewall



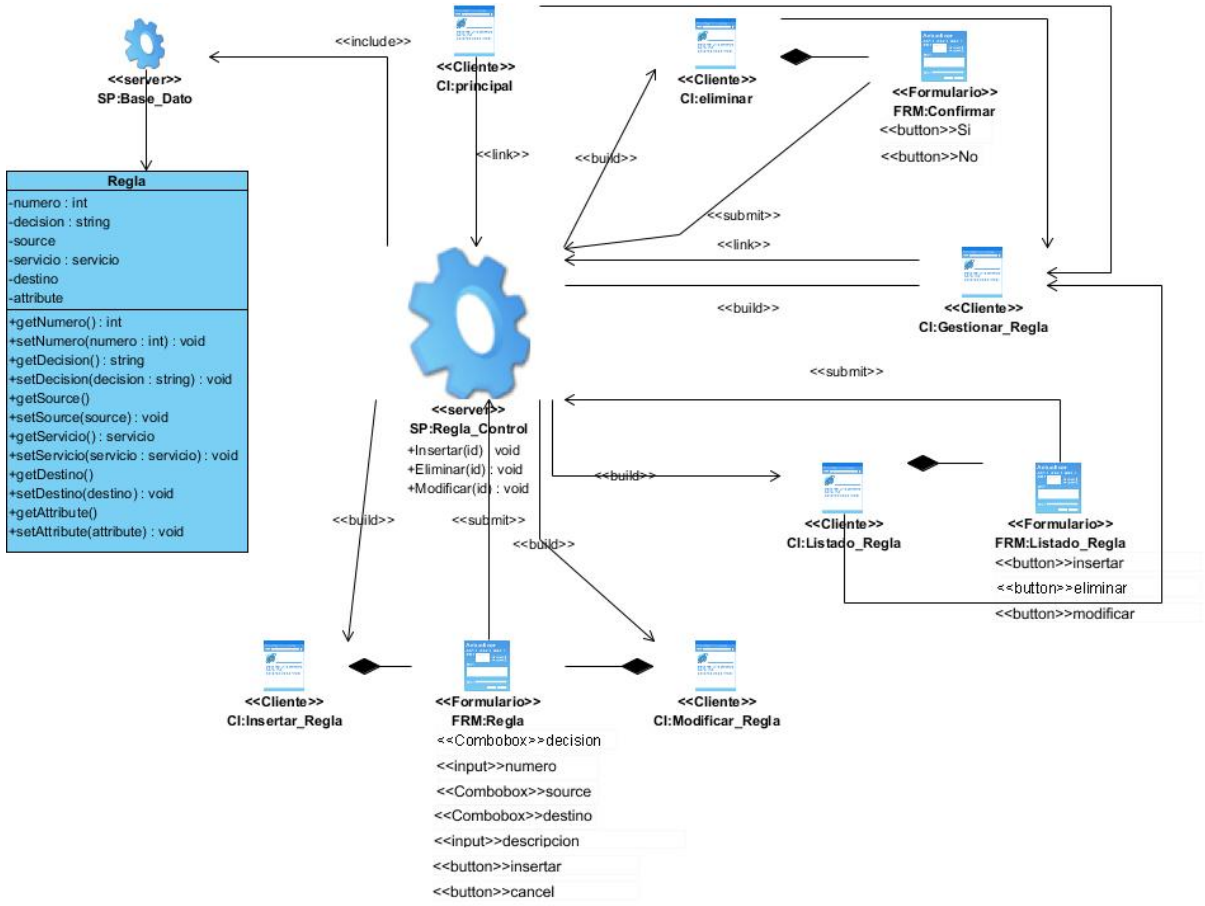
Anexo A5 Caso de Uso Gestionar Zona



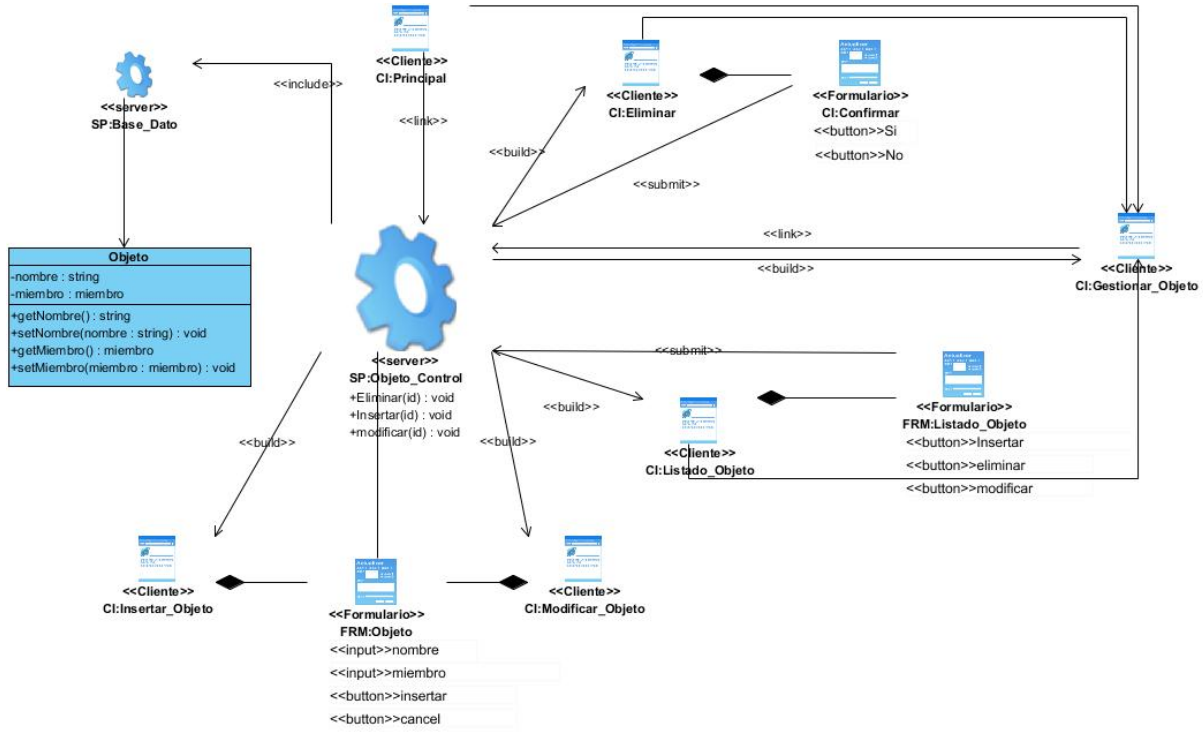
Anexo A6 Caso de uso Gestionar ACL



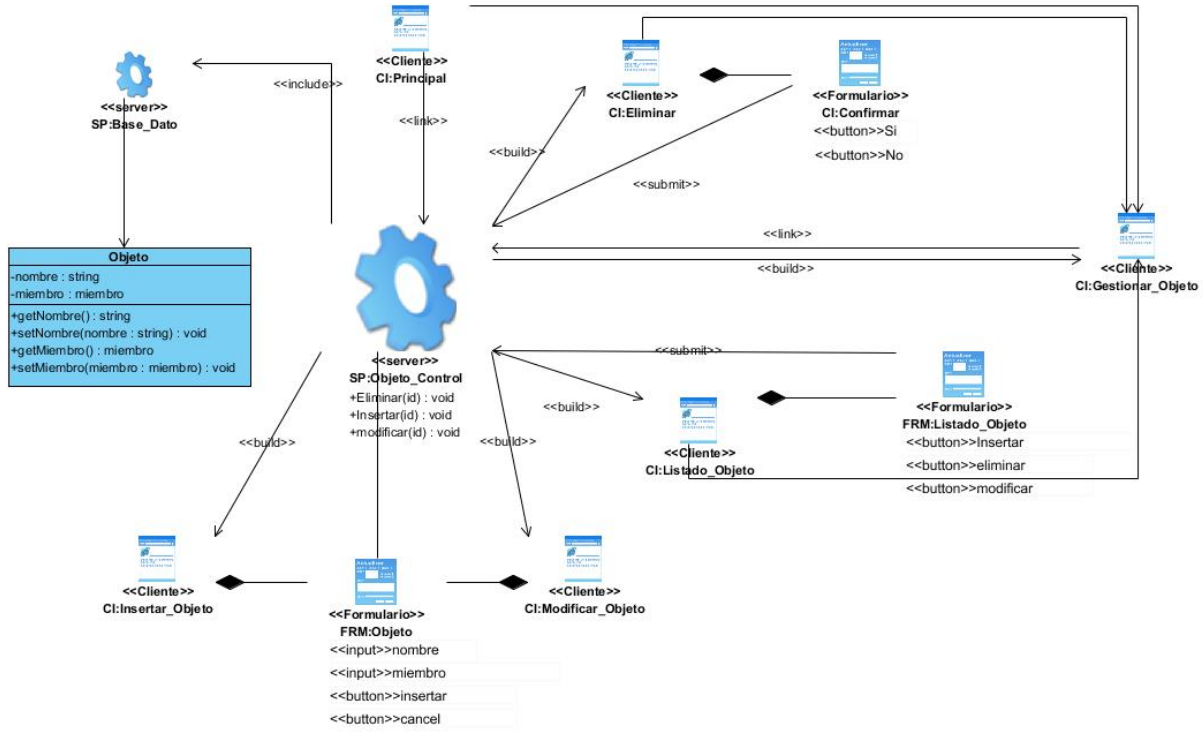
Anexo A7 Caso de uso Gestionar Regla



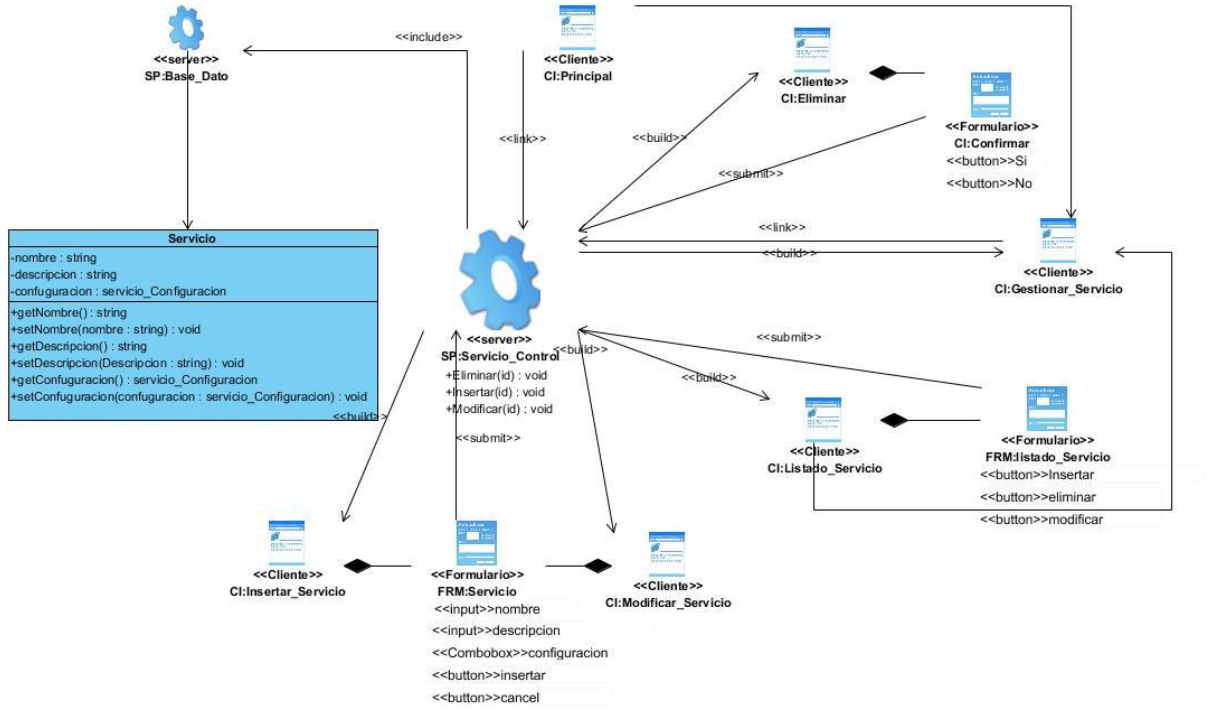
Anexo A8 Caso de uso Gestionar Objeto



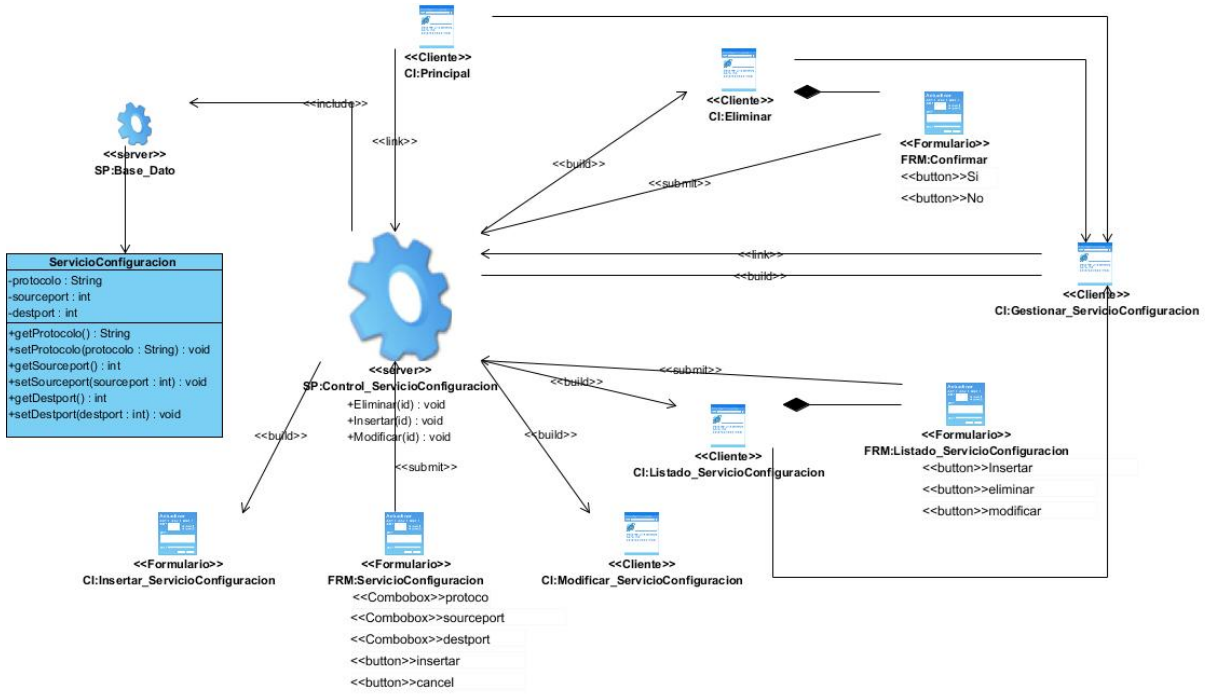
Anexo A9 Caso de Uso Gestionar Miembros



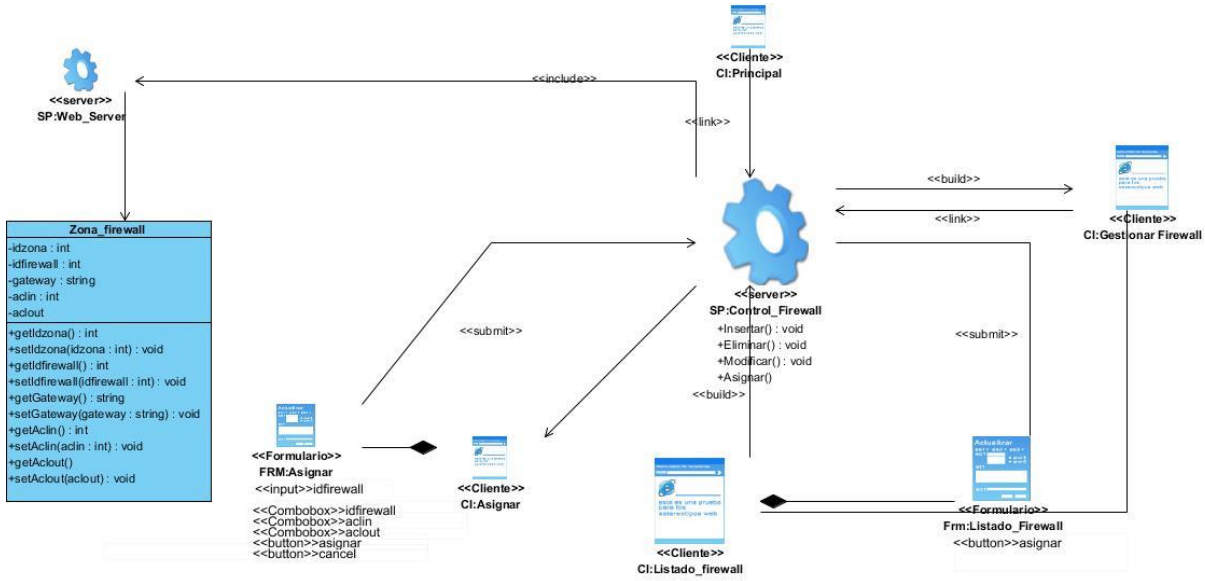
Anexo A10 Caso de uso Gestionar Servicio



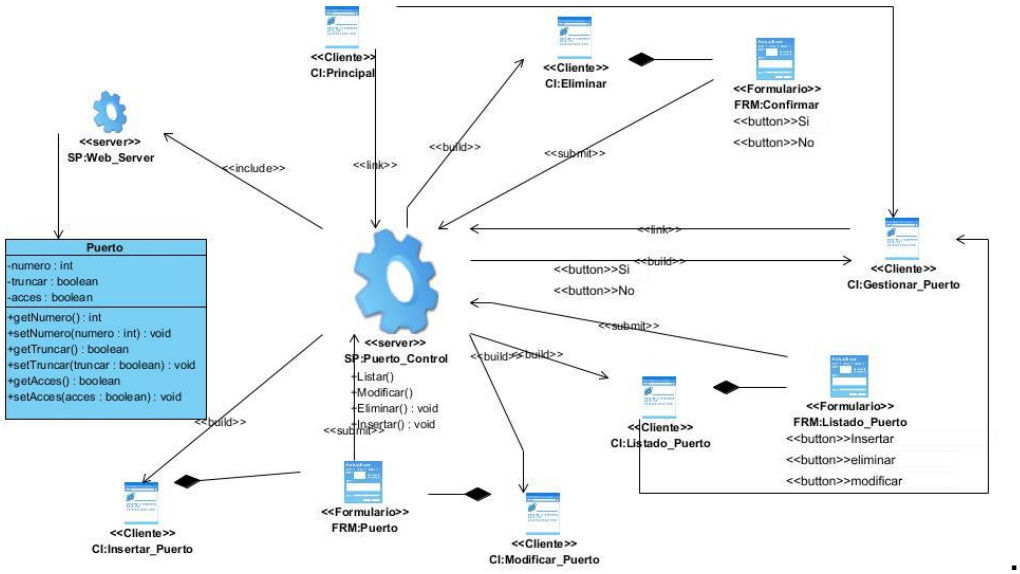
Anexo A11 Caso de Uso Gestionar Servicio Configuración



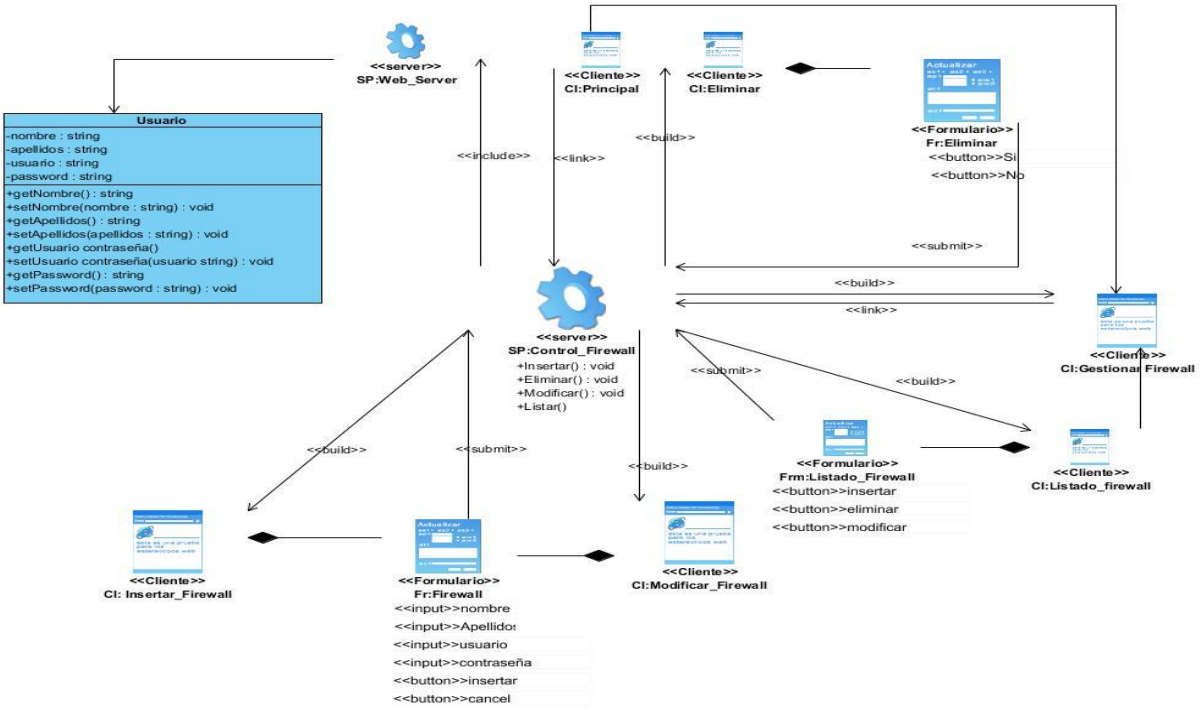
Anexo A12 Caso de Uso Asignar a un Firewall Zona.



Anexo A13 Caso de Uso Gestionar Puertos



Anexo A14 Caso de Uso Gestionar Usuarios

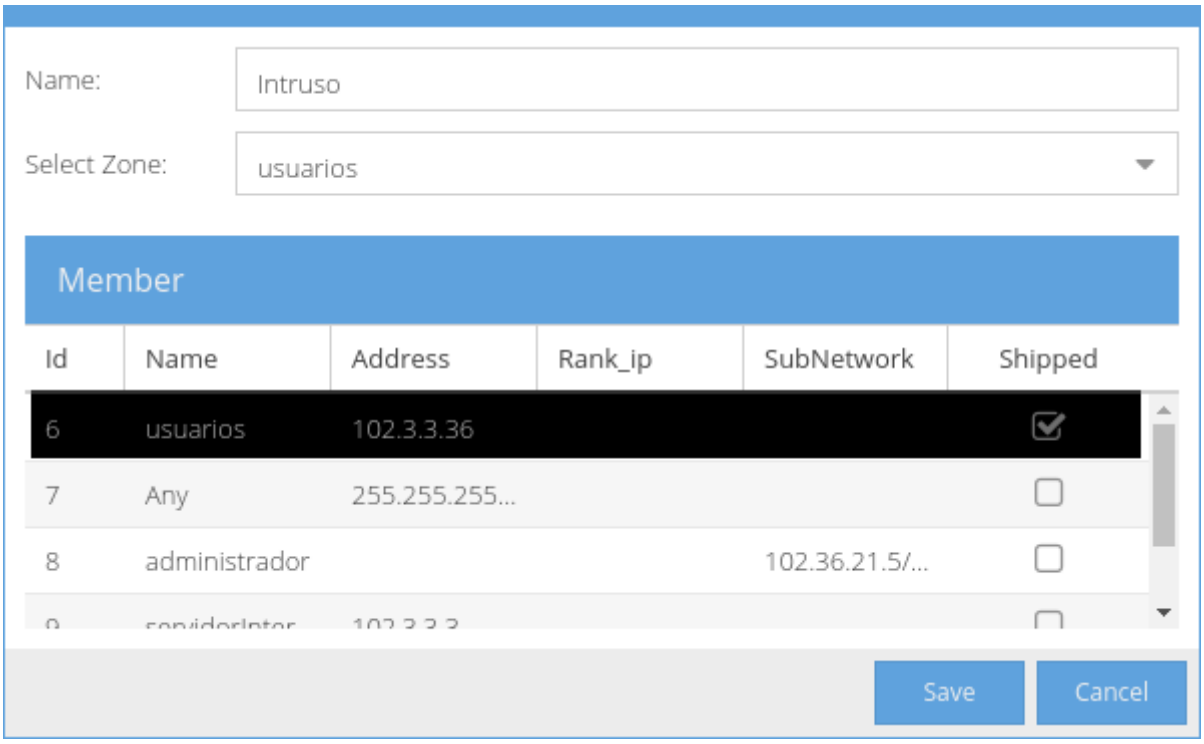


Anexo C Escenario de Pruebas

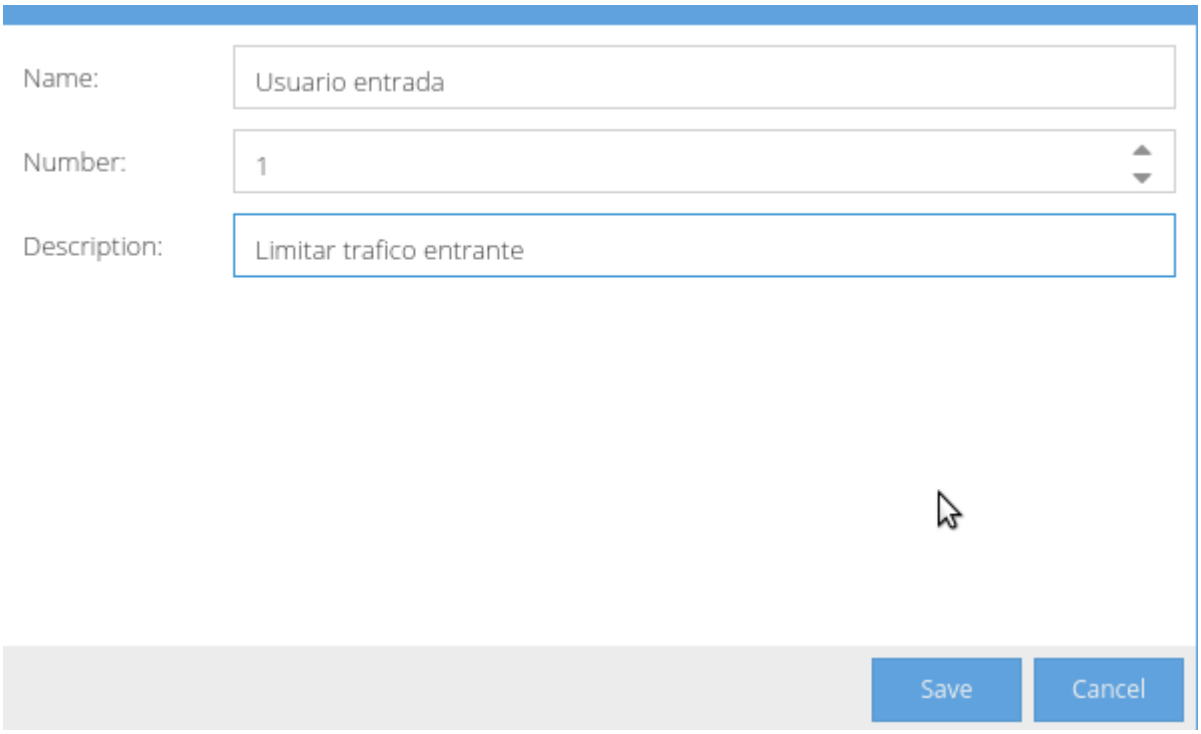
Anexo C1

Se crea la zona usuarios asignándole el puerto 2 de firewall

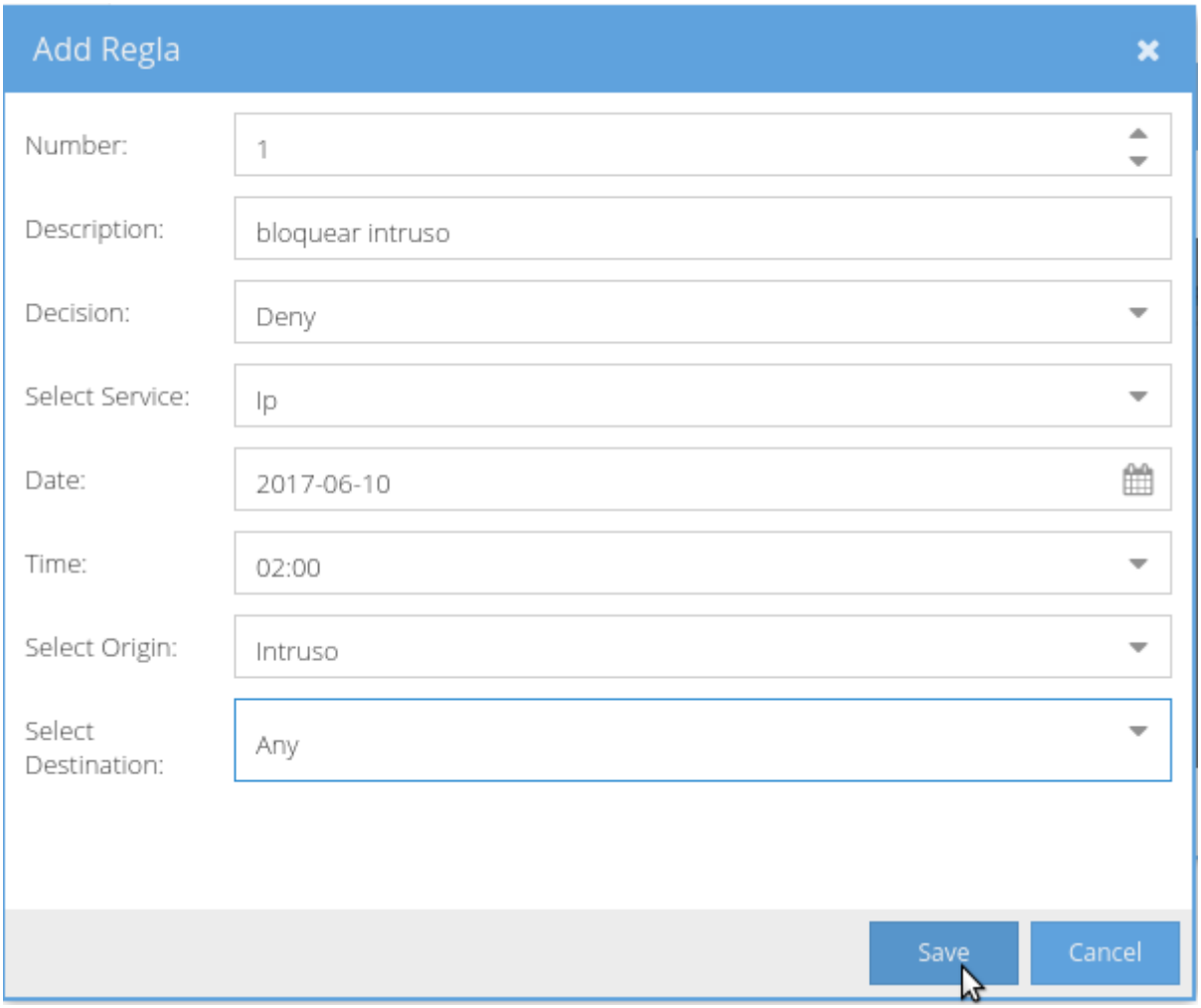
Se crea un objeto llamado intruso



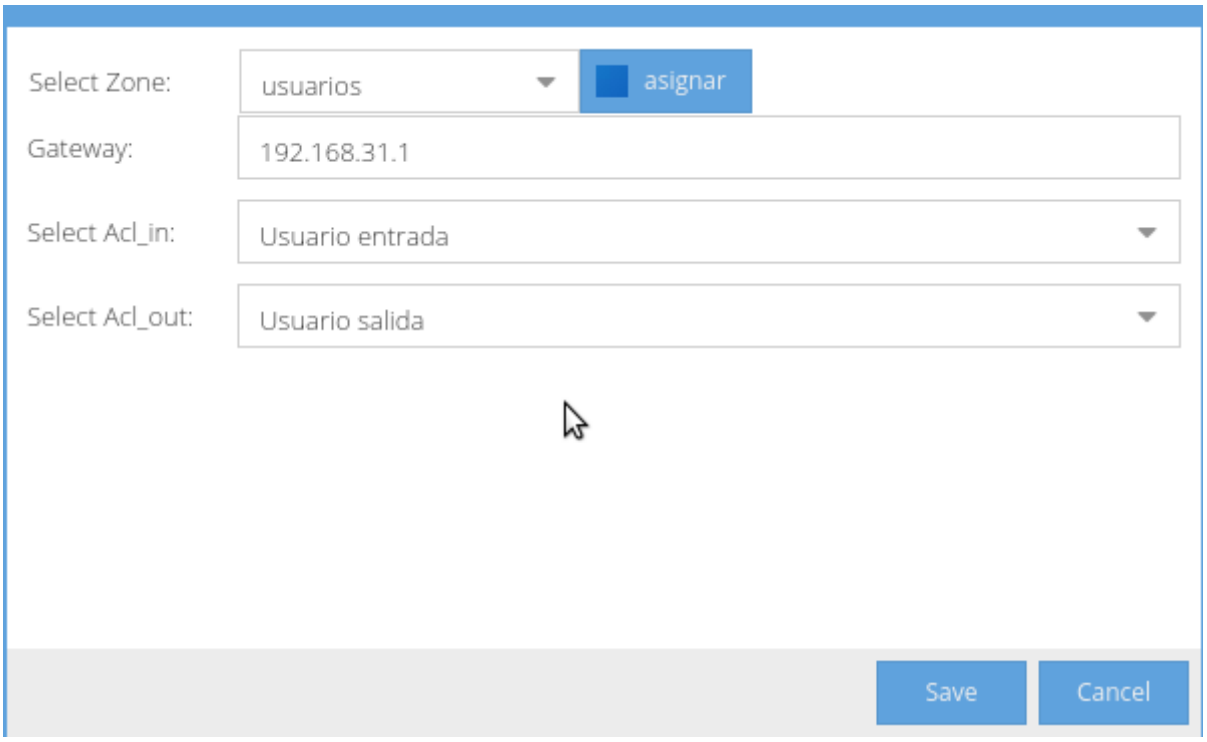
Creamos la Acl usuario en entrada es donde aplicamos las reglas



A la regla anterior le aplicamos la siguiente regla

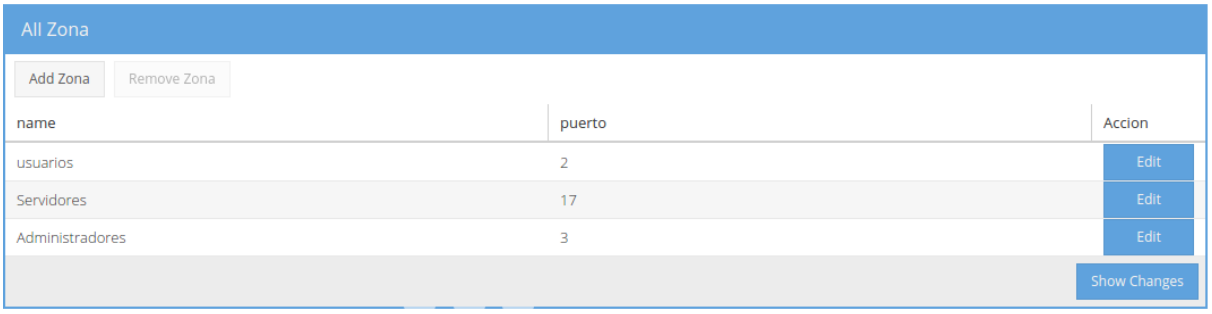


Terminamos asignando le las ACL IN Y EN OUT al Firewall

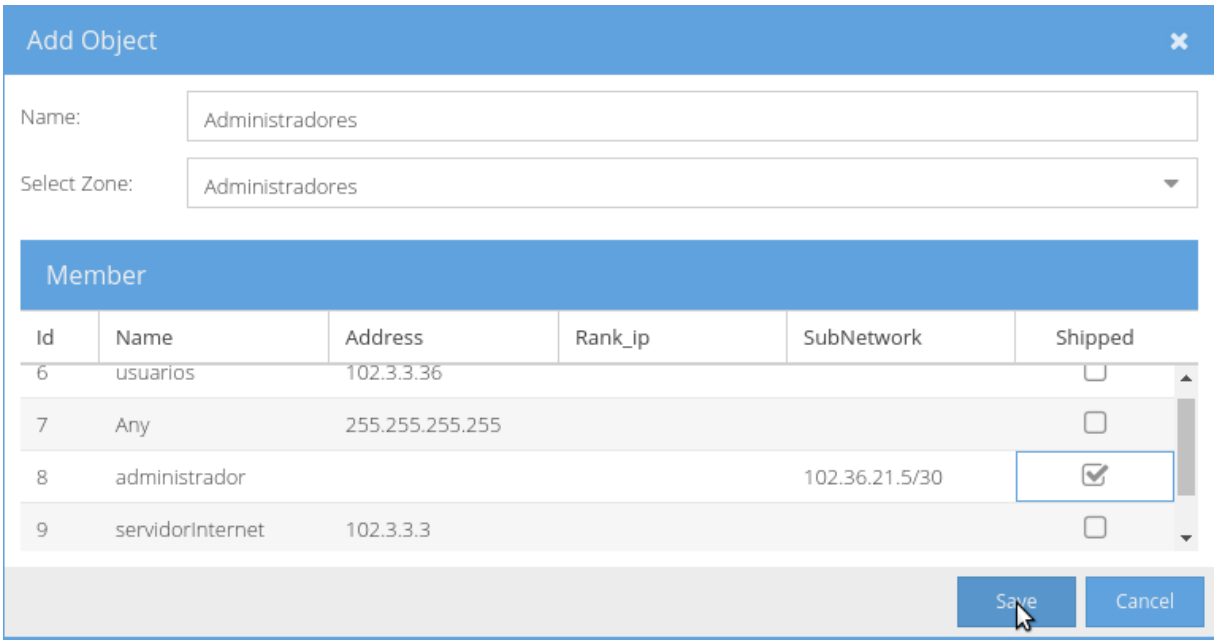


Anexo C2

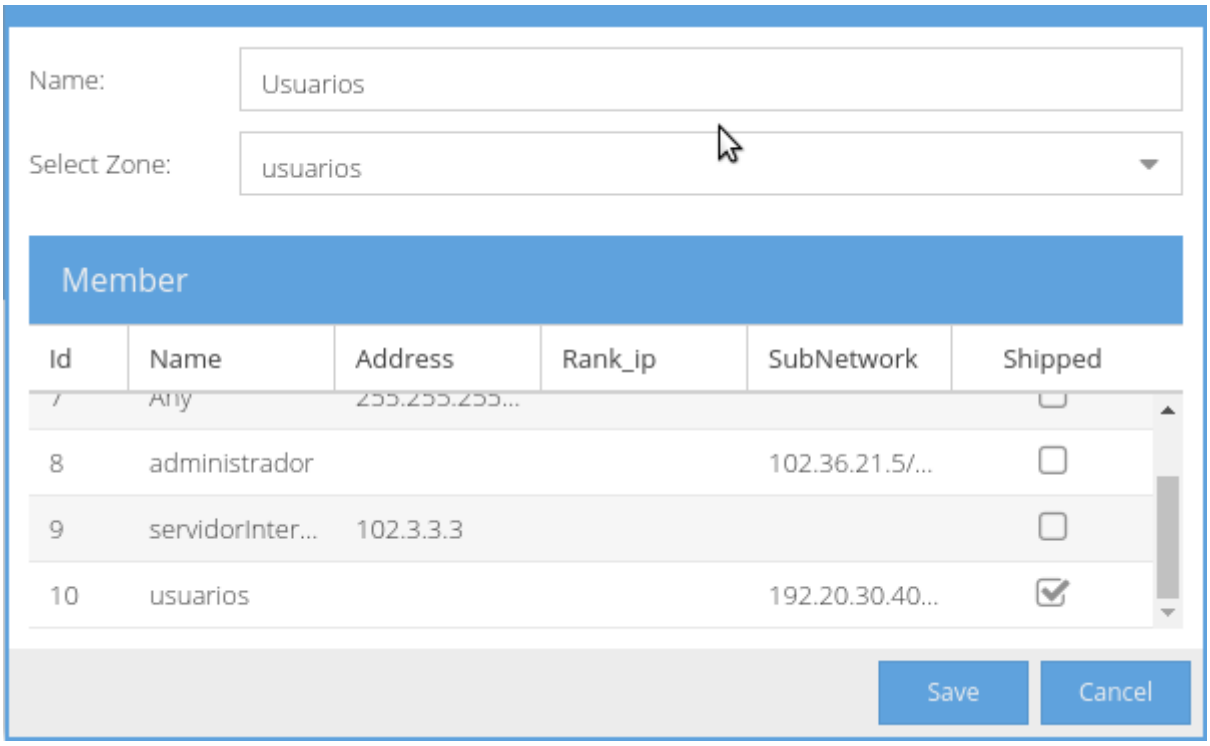
Se Crean las tres zona usuarios, servidores, Administradores



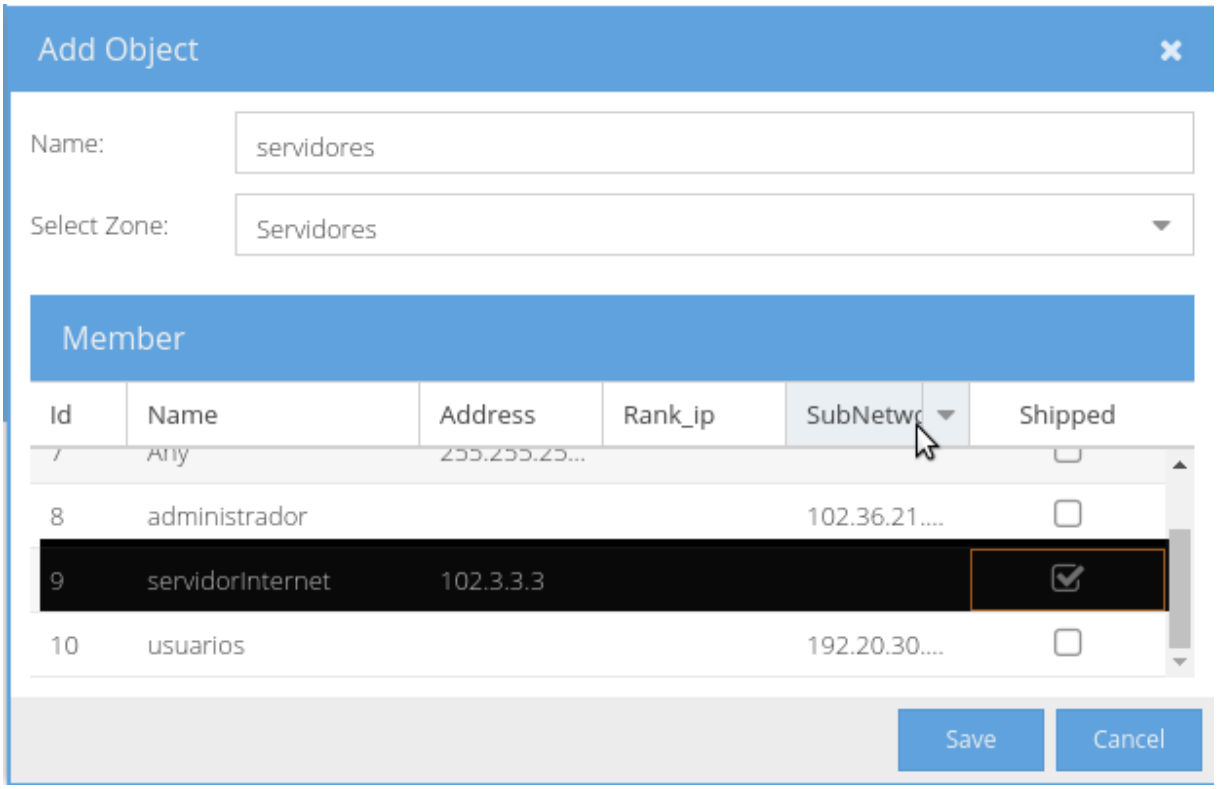
Se crea el objeto Administradores



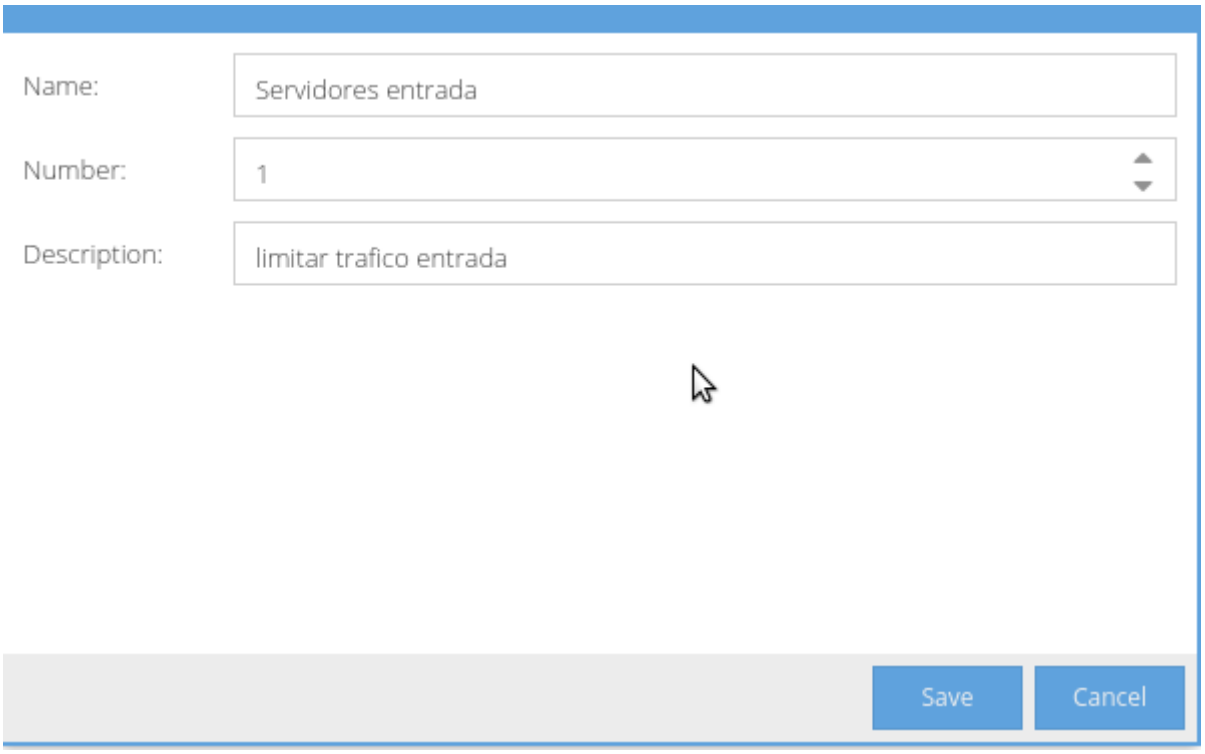
Se crea el objeto usuarios



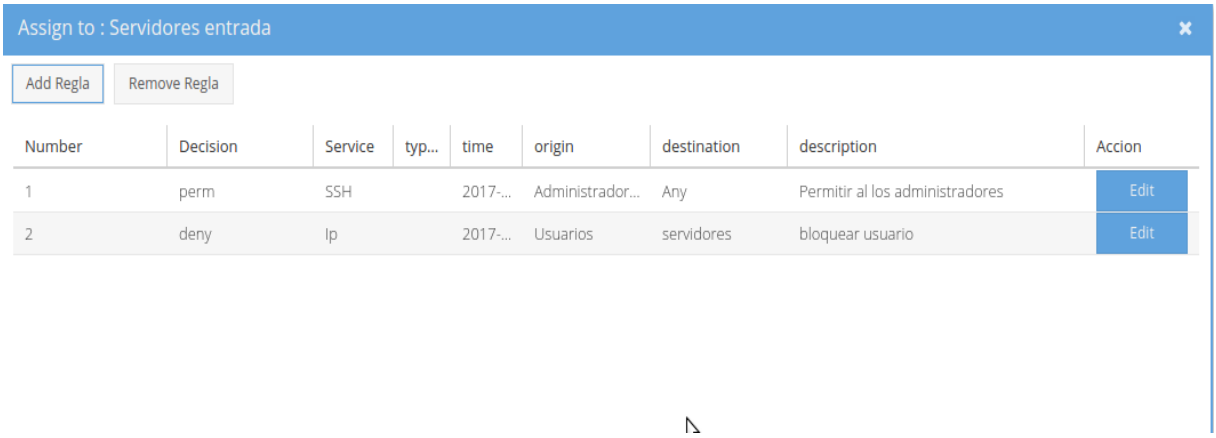
Se crea el objeto servodores



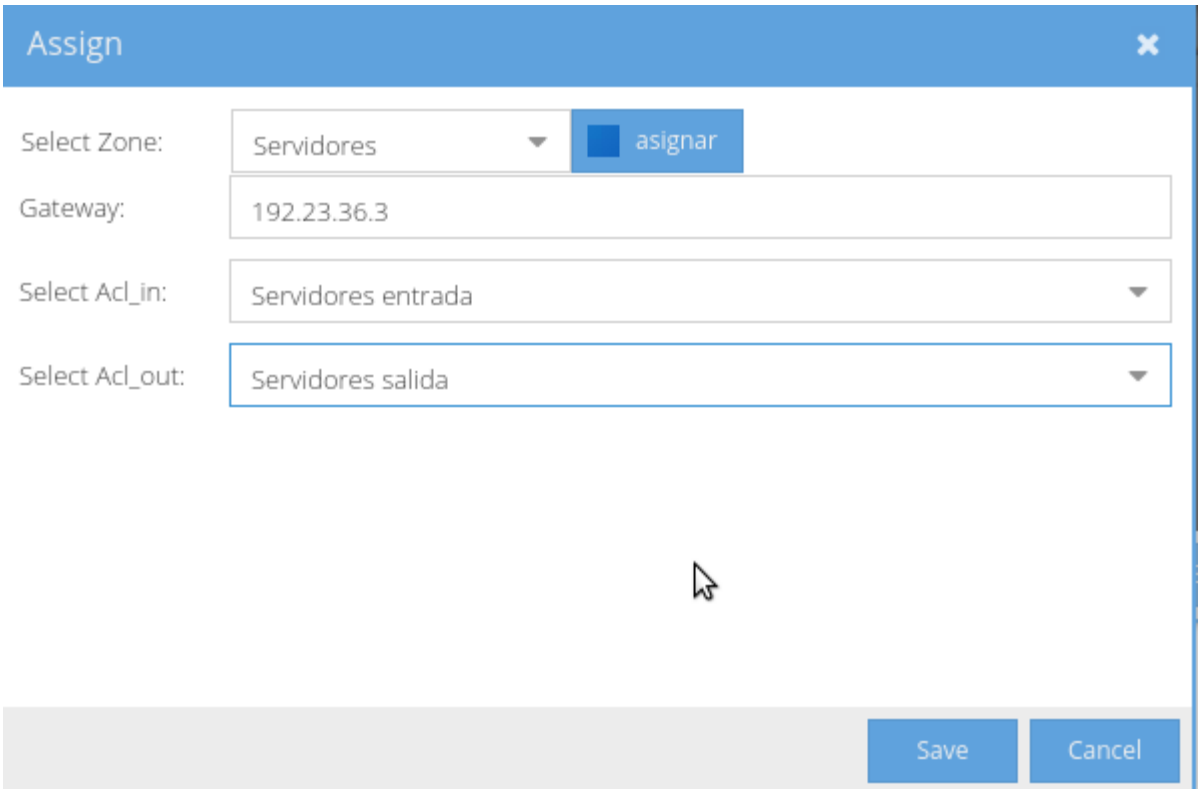
Creamos la Acl en entrada al servidor



La ACL anterior le añadimos las siguientes reglas



Terminamos asignando las ACL en entrada y en salida al firewall



Anexo B Descripción de Caso de uso del Sistema

Anexo B2 Caso de Uso Autentificarse

Nombre del Caso de Uso	Autentificarse
Actor	Administrador
Propósito	Autentificarse para poder acceder a las acciones del subsistema.

Resumen	El administrador inicia el caso de uso cuando desea acceder a las funcionalidades del sistema. Para esto debe enviar sus datos usuario y contraseña, si estos son correctos podrá hacer uso del sistema, de lo contrario no tendrá permiso para entrar al sistema.
Referencia	R1
Precondiciones	El administrador tiene que estar registrado en la Base de datos con sus permisos correspondientes.
Post-condiciones	El usuario podrá acceder a las funcionalidades del sistema .
Requisitos Especiales	

Anexo B3 Caso de Uso Cerrar Sesión

Nombre del Caso de Uso	Cerrar Sesión
Actor	Administrador(inicia)

Resumen	El administrador inicia el caso de uso cuando selecciona salir del sistema.
Referencia	R2
Precondiciones	El administrador tiene que estar Autenticado en el sistema
Post-condiciones	El usuario sale del sistema ,no puede acceder a las funcionalidades del sistema
Requisitos Especiales	

Anexo B4 Caso de Uso Cambiar Contraseña

Nombre del Caso de Uso	Cambiar Contraseña
Actor	Administrador(inicia)
Resumen	El administrador inicia el caso de uso cuando desea por cuestiones de seguridad cambiar su contraseña de entrada al sistema, debe introducir los datos pedidos usuario y contraseña actual y luego tiene opción de poner la

	nueva, si los datos actuales son correctos cambia la contraseña, sino lanza un error.
Referencia	R3
Precondiciones	El Administrador debe estar previamente autentificado en el sistema.
Post-condiciones	Tiene mejor seguridad el sistema, los datos son guardados correctamente.
Requisitos Especiales	

Anexo B5 Caso de Uso Gestionar Firewall

Nombre del Caso de Uso	Gestionar Firewall
Actor	Administrador(inicia)
Propósito	Configurar un nuevo firewall
Resumen	El caso de uso comienza cuando el administrador desea tomar el control de un nuevo firewall lo primera acción que el realiza es insertarlo al sistema, lo cual

	aquí tendrá las posibilidades de listar todos los firewall existentes, modificarlos y eliminarlos .
Referencia	R4,R5,R6,R28
Precondiciones	El Administrador debe estar previamente autentificado en el sistema.
Post-condiciones	Tener un firewall bien configurado para no tener tráfico de paquetes maliciosos en la red
Requisitos Especiales	

Anexo B6 Caso de Uso Gestionar Zona

Nombre del Caso de Uso	Gestionar Zona
Actor	Administrador(inicia)
Propósito	Llevar un control de las Zonas
Resumen	El caso de uso comienza cuando el administrador desea controlar una zona y

	la inserta al sistema, lo cual aquí tendrá las posibilidades de listar todas las Zonas existentes luego podrá , modificarlos y eliminarlos, sino existe crear uno nuevo
Referencia	R7,R8,R9,R29
Precondiciones	El Administrador debe estar previamente autentificado en el sistema. Deben existir puertos.
Post-condiciones	Controlar el fluido de paquetes en todas las zonas
Requisitos Especiales	

Anexo B7 Caso de Uso Gestionar ACL

Nombre del Caso de Uso	Gestionar ACL
Actor	Administrador(inicia)
Propósito	Llevar un control del tráfico de paquete

Resumen	El caso de uso comienza cuando el administrador necesita brindar seguridad a la red, en dependencia de lo que desee, tendrá acceso a todas las ACL (Action Control List) y podrá modificarlas a su conveniencia, si le hiciera falta añadir o eliminar alguna existente, cada una de las ACL tiene su propias lista control acceso de reglas estas podrán ser configuradas como añadir, modificar y eliminar. Su orden es muy importante.
Referencia	R22,R23,R24,R30
Precondiciones	El Administrador debe estar previamente autentificado en el sistema.
Post-condiciones	Luego de ser creada la ACL debe ser añadida a una zona, esta le brindara la seguridad requerida por el administrador

Anexo B8 Caso de Uso Gestionar Regla

Nombre del Caso de Uso	Gestionar Regla

Actor	Administrador(inicia)
Propósito	Delimita el control del tráfico de paquete
Resumen	El caso de uso comienza luego de que el administrador creo una ACL, comienza al insertar nuevas reglas en la lista de control de acceso estas deben estar ordenadas por prioridad. El administrador tendrá como funciones eliminarlas, modificarlas y añadirlas.
Referencia	R25,R26,R27,R33
Precondiciones	El Administrador debe estar previamente autentificado en el sistema. Y debe existir una ACL(Action Control List) a la cual se les va aplicar las reglas.
Post-condiciones	La ACL o lista de control de acceso puede ser ejecutada.

Anexo B9 Caso de Uso Gestionar Usuario

Nombre del Caso de Uso	Gestionar Usuario

Actor	Administrador(inicia)
Propósito	Añadir otro administrador
Resumen	El caso de uso comienza cuando llega un nuevo administrador al centro debe insertado en el sistema, si desea modificar uno existente o si un administrador abandona el centro eliminarlo por completo
Referencia	R36,R37,R38,R42
Precondiciones	El Administrador debe estar previamente autentificado en el sistema.
Post-condiciones	Tener un control sobre los administradores que trabaja con el sistema.
Requisitos Especiales	

Anexo B10 Caso de Uso Gestionar Objeto

Nombre del Caso de Uso	Gestionar Objetos

Actor	Administrador(inicia)
Propósito	Poder agrupar en un conjunto las estaciones de red que van hacer objetos de control.
Resumen	Cuando el administrador va a insertar una regla y no desea poner ni una ip ni un rango de fuente crea un objeto ya que este podrá tener varios miembros que son vario segmentos de red como rango y otros con ip o una subred, en si es más abarcador
Referencia	R10,R11,R12,R31
Precondiciones	El Administrador debe estar previamente autentificado en el sistema. Deben existir miembros creados.
Post-condiciones	Filtra por los Objetos Seleccionados
Requisitos Especiales	

Anexo B11 Caso de Uso Gestionar Miembros

Nombre del Caso de Uso	Gestionar Miembros
Actor	Administrador(inicia)
Propósito	Determinar qué es lo que se va a controlar
Resumen	Este caso de uso es muy importante ya que es donde el administrador crea los segmentos de red que se van a controlar .
Referencia	R13,R14,R15,R32
Precondiciones	El Administrador debe estar previamente autentificado en el sistema.
Post-condiciones	
Requisitos Especiales	

Anexo B12 Caso de Uso Gestionar Servicio

Nombre del Caso de Uso	Gestionar Servicio

Actor	Administrador(inicia)
Propósito	Poder agrupar en un conjunto de puerto tanto de fuente como destino para un protocolo
Resumen	El caso de uso comienza cuando el administrador desea darle un servicio a una regla, los servicios tiene la función de ser añadido, eliminados o modificados estos brindan mediante su configuración el protocolo que va usar por los puertos que va a traficar de una fuentes a destino.
Referencia	R16,R17,R18
Precondiciones	El Administrador debe estar previamente autentificado en el sistema.
Post-condiciones	
Requisitos Especiales	

Anexo B13 Caso de Uso Gestionar Servicio Configuración

Nombre del Caso de Uso	Gestionar Servicio Configuración
Actor	Administrador(inicia)
Propósito	Poder configurar el servicio en cuanto el protocolo que usara los puertos q vas a usar.
Resumen	El caso de uso comienza cuando el administrador va a definirle un protocolo a un servicio ya creado y al mismo le define un puerto fuente y otro destino.
Referencia	R19,R20,R21
Precondiciones	El Administrador debe estar previamente autentificado en el sistema, tiene que existir un servicio el cual se va a configurar
Post-condiciones	
Requisitos Especiales	

Anexo B14 Caso de Uso Gestionar Puerto

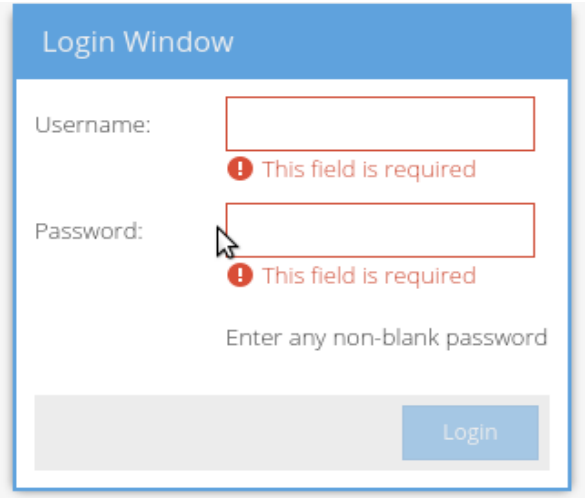
Nombre del Caso de Uso	Gestionar Puerto
Actor	Administrador(inicia)
Propósito	Crear puertos para luego al crear una zona añadírselos.
Resumen	El caso de uso comienza cuando el administrador desea insertar nuevos puertos en el sistemas, tendrá permiso en a las funciones de Insertar, modificar, eliminar,
Referencia	R39,R40,R41,R35
Precondiciones	El Administrador debe estar previamente autentificado en el sistema, tiene que existir un servicio el cual se va a configurar
Post-condiciones	Insertarlos en una Zona
Requisitos Especiales	

Anexo B15 Caso de Uso Asignar Zona

Nombre del Caso de Uso	Asignar Zona
Actor	Administrador(inicia)
Propósito	Asignar a un firewall una Zona con una ACL en entrada y una en salida
Resumen	El caso de uso comienza cuando el administrador tiene una zona ya creada y un firewall, desea asignarle al firewall una Zona con la ACL en entrada y en salida
Referencia	R34
Precondiciones	El Administrador debe estar previamente autentificado en el sistema, debe existir un firewall, zona y ACL insertada en el sistema
Post-condiciones	
Requisitos Especiales	

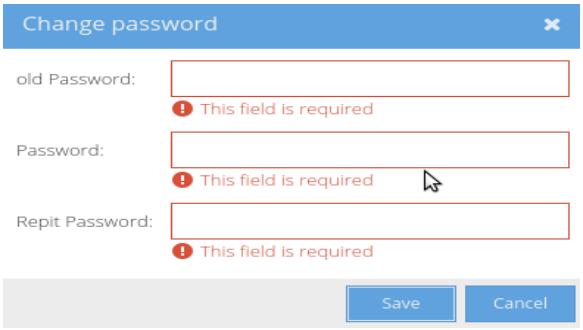
Anexo D Pruebas Funcionales

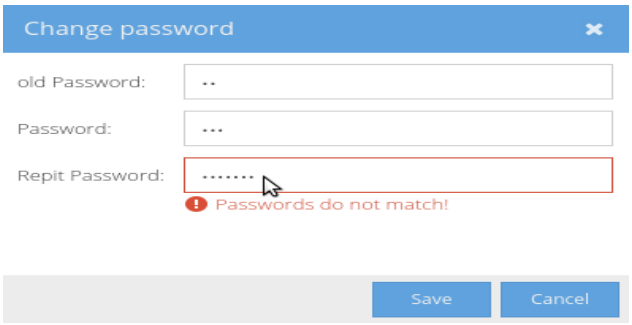
Anexo D1



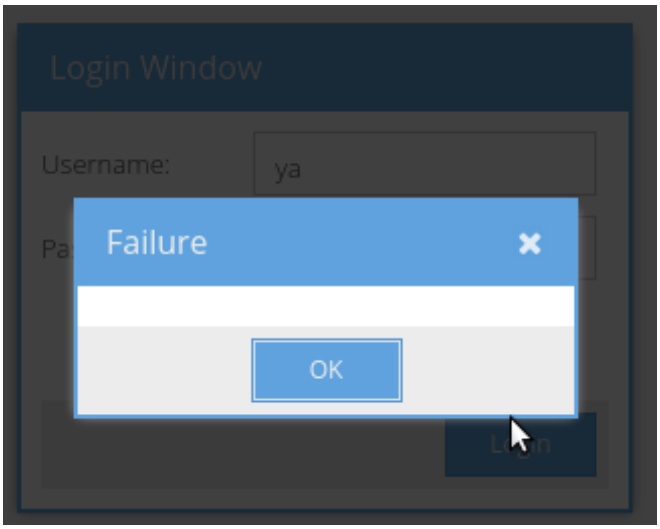
Anexo D2

Anexo D3

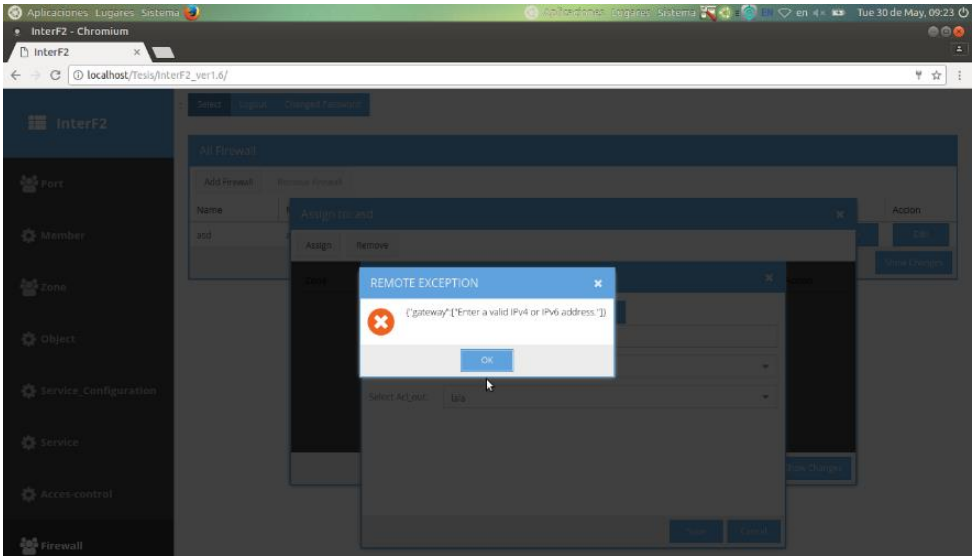

Change password
old Password:
This field is required
Password:
This field is required
Repit Password:
This field is required
Save
Cancel


Change password
old Password:
Password:
Repit Password:
Passwords do not match!
Save
Cancel

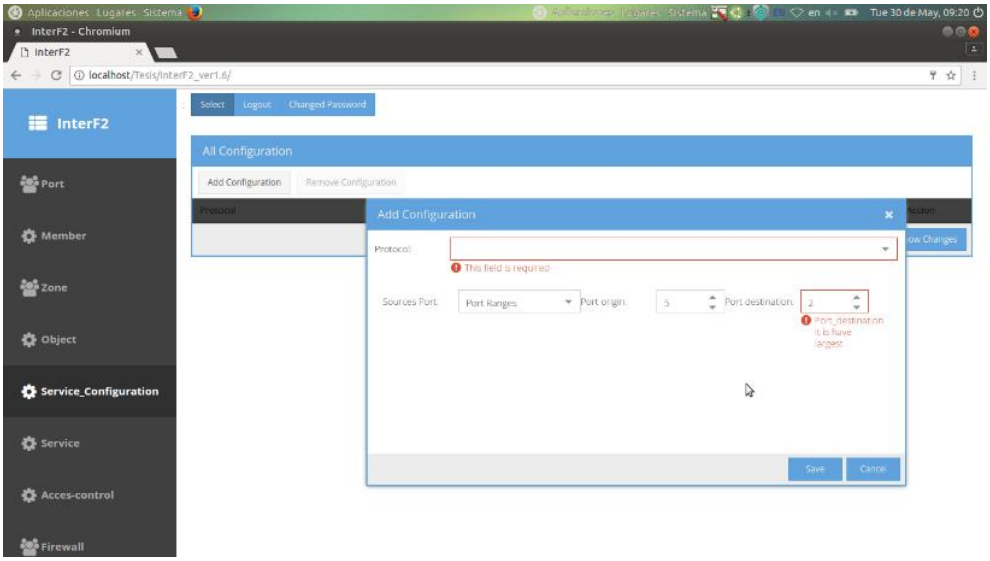
Anexo D4



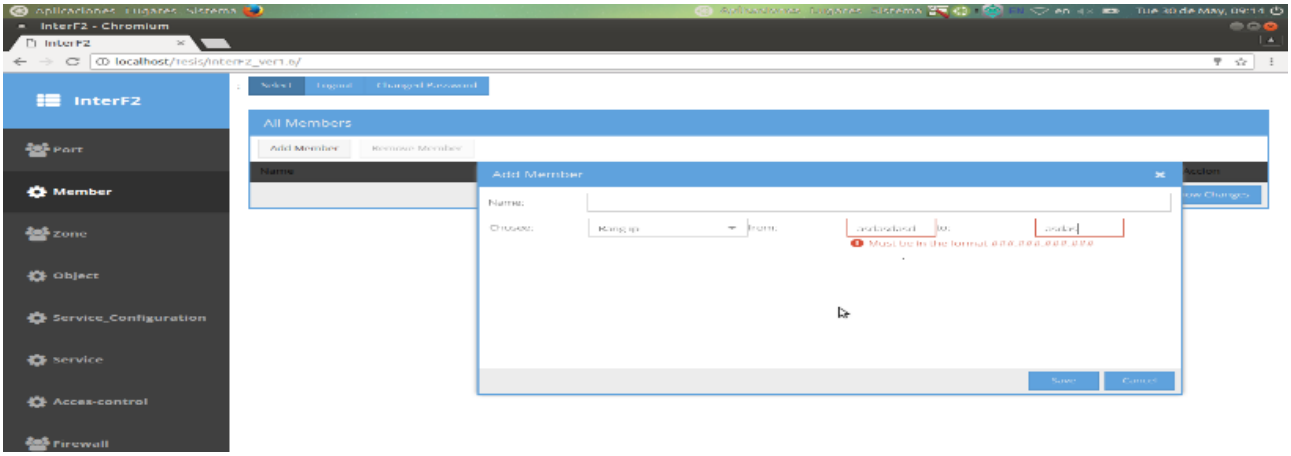
Anexo D5



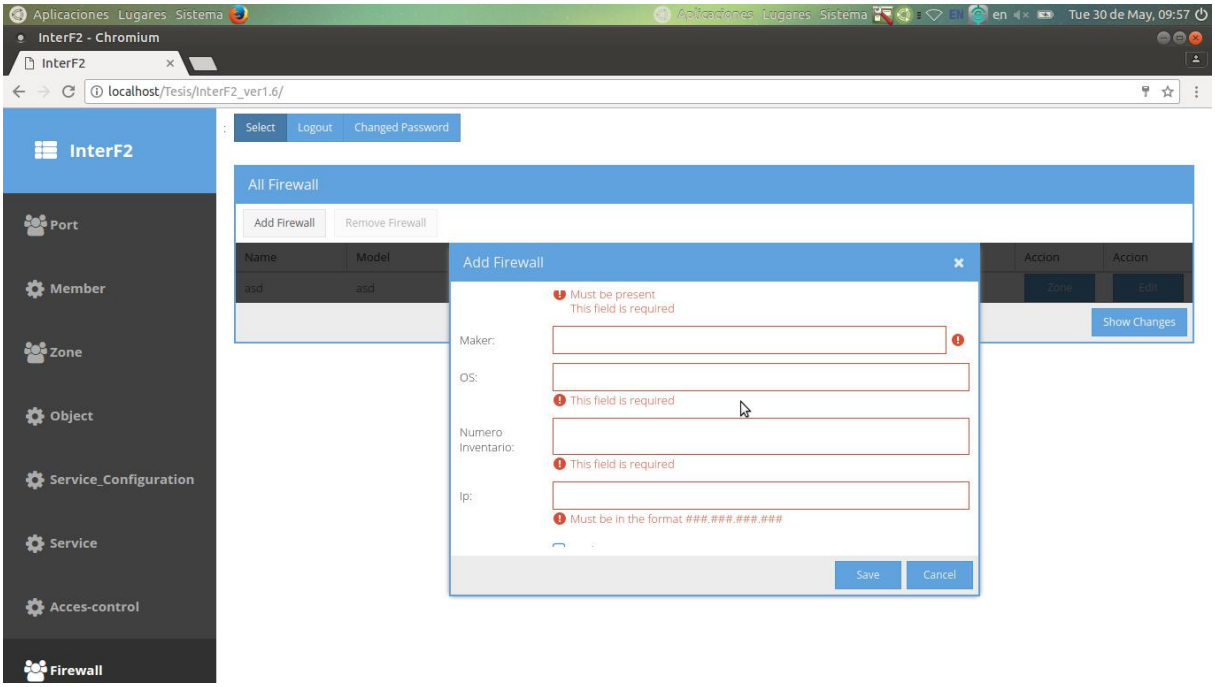
Anexo D6



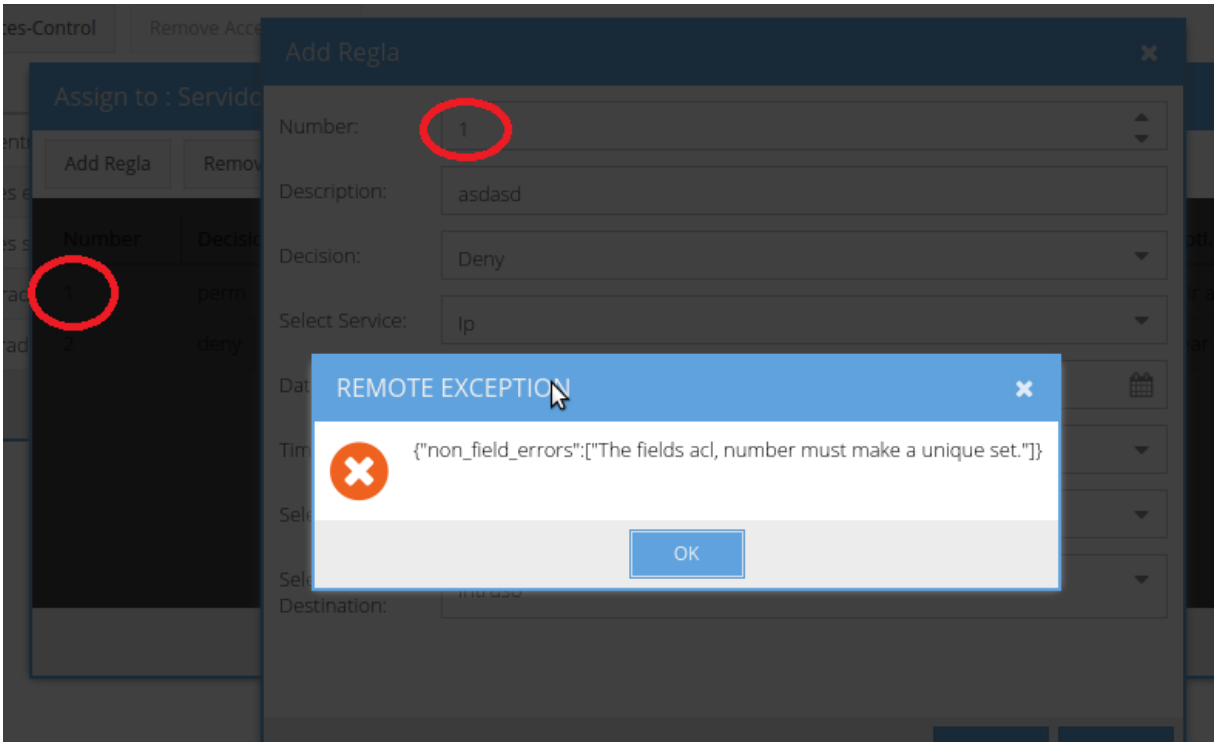
Anexo D7



Anexo D8


Aplicaciones Lugares Sistema
Tue 30 de May, 09:57
InterF2 - Chromium
InterF2
localhost/Tesis/InterF2_ver1.6/
InterF2
Select
Logout
Changed Password
Port
Member
Zone
Object
Service_Configuration
Service
Acces-control
Firewall
All Firewall
Add Firewall
Remove Firewall
Name
Model
Accion
Accion
asd
asd
Zone
Edit
Show Changes
Add Firewall
Must be present
This field is required
Maker:
OS:
This field is required
Numero Inventario:
This field is required
Ip:
Must be in the format ###.###.###.###
Save
Cancel

Anexo D9


Add Regla
Number:
1
Description:
asdasd
Decision:
Deny
Select Service:
Ip
REMOTE EXCEPTION
{"non_field_errors":["The fields acl, number must make a unique set."]}
OK
Destination: