



NOVA REALTIME

**ANA PAULINA LIZARAZO
ANDRÉS PIÑA FONSECA**

**REGIONAL UNIMINUTO DE SOACHA
TECNOLOGIA EN INFORMÁTICA
SOACHA
2012**



NOVA REALTIME

**ANA PAULINA LIZARAZO
ANDRÉS PIÑA FONSECA**

Trabajo de grado para optar por el título de Tecnólogo en Informática.

**Director:
ING. PABLO FONSECA
ESPECIALISTA RAVD**

**REGIONAL UNIMINUTO DE SOACHA
TECNOLOGIA EN INFORMÁTICA
SOACHA
2012**



Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado



A nuestras familias por apoyarnos en todo momento y lugar para surgir en nuestra vida personal y profesional.



AGRADECIMIENTOS

Los autores expresan sus agradecimientos a:

A Ana Fonseca

Que por sus sabios consejos, valores y motivación. Me ha permitido desarrollarme como una persona integral y con valores, pero sobre todo por su gran amor.

A Nuestros Familiares

Los cuales siempre nos han apoyado y animado, para seguir adelante en la culminación de nuestros estudios tecnológicos y búsqueda del crecimiento personal.

A Carlos Trujillo

Por todas las enseñanzas personales y profesionales que ha transmitido a través de los años los cuales han fortalecido mis capacidades cognitivas y analíticas.

A mi Familia

Madre, esposo, hijo, hermanos y demás familiares cercanos que siempre están cerca para apoyarme, son el principal motivo para seguir con las metas propuestas en busca del conocimiento y una mejor calidad de vida.

A Diego Jaramillo

Por su apoyo incondicional, por compartir su conocimiento y las motivaciones seguir siempre hacia adelante sin importar los obstáculos. Un ejemplo digno de seguir.

Atentamente,

Lizarazo Ana Paulina
Piña Fonseca Andrés



TABLA DE CONTENIDO

	Pág.
INTRODUCCCIÓN	1
1. PROBLEMA DE LA INVESTIGACIÓN.....	2
1.1 NOMBRE DE LA APLICACIÓN	2
1.2 ANTECEDENTES.....	2
1.3 FORMULACIÓN DEL PROBLEMA	2
1.4 PLANTEAMIENTO DEL PROBLEMA.....	4
1.4.1 Descripción.	4
1.4.2 Solución.	4
1.5 JUSTIFICACIÓN.....	5
1.6 OBJETIVOS.....	6
1.6.1 Objetivo General.	6
1.6.2 Objetivos específicos.	6
1.7 BENEFICIOS QUE CONLLEVA	7
1.8 ALCANCES Y DELIMITACIONES.....	7
1.8.1 Alcances.....	7
1.8.2 Delimitaciones.....	8
1.9 FACTIBILIDAD	11
1.9.1 Factibilidad técnica.....	11
1.9.2 Factibilidad económica.....	11
1.9.3 Factibilidad operativa.	12
2. MARCO DE REFERENCIA.....	15
2.1 MARCO HISTORICO.....	15

2.2	MARCO TEORICO	20
2.2.1	Symfony.	21
2.2.2	Java.....	23
2.2.3	Mysql.....	26
2.2.4	Snmp.....	28
2.2.5	Firewall.....	34
2.2.6	Iptables.....	36
2.2.7	¿Qué es Linux?	36
2.2.8	Distribución de GNU/Linux – Ubuntu.....	39
2.2.9	Software Libre.	40
2.2.10	Html.....	40
2.2.11	Css.	41
2.2.12	Web Services.	41
3.	CONSIDERACIONES DE DISEÑO	42
3.1	CONCEPTOS BASICOS	42
3.2	REQUERIMIENTOS TÉCNICOS.....	42
3.2.1	Aplicación web.	42
3.2.2	Aplicación Mobile.	43
3.2.3	Requerimientos Generales.....	44
3.2.4	Moldeamiento de monitoreo.....	44
3.2.5	Conexión con la base de datos Mysql.....	45
3.3	REQUERIMIENTOS FUNCIONALES.....	45
3.4	REQUERIMIENTOS NO FUNCIONALES	46
3.5	ALCANCE DEL SISTEMA	48
4.	DISEÑO DEL SISTEMA.....	50
4.1	DESCRIPCIÓN GENERAL.....	50



4.2	ARQUITECTURA DEL SISTEMA.....	50
4.3	MODELADO DE CASOS DE USO	54
4.3.1	Descripción general de actores.....	56
4.3.2	Diagramas de modelado de casos de uso	56
4.3.3	Diagramas de modelado de Secuencia.....	69
4.3.4	Diagramas de modelado de clases.	86
5.	RECOMENDACIONES	101
5.1.	GENERALIDADES DE LA APLICACIÓN.....	101
5.1.1	Arquitectura de Red.	101
5.2	Optimización de servidor para Nova Realtime.....	102
5.3	Optimización de servidor para base de datos.....	102
6.	ESTIMACION DE PERTINENCIA PARA SOFTWARE DE MONITOREO DE SERVIDORES GNU/LINUX.....	103
6.1	ESTIMACION DE PERTINENCIA SIN SISTEMA DE MONITOREO	103
6.1.1	Marco de referencia para monitoreo manual de servidores	103
6.1.2	Instrumentos de aplicación.....	103
6.1.3	Reportes y gráficos.	107
6.1.4	Observaciones.	107
7.	CONCLUSIONES	108
	BIBLIOGRAFÍA.....	109



LISTA DE TABLAS

	Pág.
Tabla 1. Cronograma de actividades.	11
Tabla 2. Costos estimados para el desarrollo.	12
Tabla 3. Versiones de Symfony.	23
Tabla 4. Requerimientos no funcionales.	48
Tabla 5. Requerimientos del sistema para para modelado de casos de uso.	55

LISTA DE ILUSTRACIONES

	Pág.
Ilustración 1. Interacción entre agentes y gestores Snmp.	30
Ilustración 2. Parte de la jerarquía del espacio de nombres de los identificadores de objetos.	31
Ilustración 3. Árbol de Nombres Snmp.	32
Ilustración 4. Representación de una red LAN con Firewall.	35
Ilustración 5. Login & Logout.....	57
Ilustración 6. Caso de uso Users.	57
Ilustración 7. Caso de uso ip address.	58
Ilustración 8. Caso de uso monitor type.....	58
Ilustración 9. Caso de uso monitor ip address.	59
Ilustración 10. Caso de uso monitor user.....	60
Ilustración 11. Caso de uso monitoring.	61
Ilustración 12. Caso de uso server status.	62
Ilustración 13. Caso de uso permanent monitoring.....	63
Ilustración 14. Caso de uso reportes.	64
Ilustración 15. Caso de uso entorno mobile – configuration.	65
Ilustración 16. Caso de uso entorno mobile - login & logout.....	65
Ilustración 17. Caso de uso entorno mobile – List Servers.	66
Ilustración 18. Caso de uso entorno mobile - partitions.	67
Ilustración 19. Caso de uso entorno mobile – graphic.	68
Ilustración 20. Diagrama de secuencia - login & logout.	69
Ilustración 21. Diagrama de secuencia – users.	70
Ilustración 22. Diagrama de secuencia - ip address.	71
Ilustración 23. Diagrama de secuencia - monitor type.	72
Ilustración 24. Diagrama de secuencia - monitor ip address.	73
Ilustración 25. Diagrama de secuencia - monitor user.....	74
Ilustración 26. Diagrama de secuencia - monitoring history.....	75
Ilustración 27. Diagrama de secuencia - monitoring realtime.	76
Ilustración 28. Diagrama de secuencia - server status.	77
Ilustración 29. Diagrama de secuencia - permanent monitoring hdd.	78
Ilustración 30. Diagrama de secuencia - permanent monitoring memory.	78
Ilustración 31. Diagrama de secuencia - permanent monitoring cpu.	79
Ilustración 32. Diagrama de secuencia – reports excel.....	80
Ilustración 33. Diagrama de secuencia - reports pdf.....	81

Ilustración 34. Diagrama de secuencia aplicación mobile - Configuration.	82
Ilustración 35. Diagrama de secuencia aplicación mobile - login & logout.	82
Ilustración 36. Diagrama de secuencia aplicación mobile - list servers.	83
Ilustración 37. Diagrama de secuencia aplicación mobile - graphic RAM.	83
Ilustración 38. Diagrama de secuencia aplicación mobile - graphic cpu.	84
Ilustración 39. Diagrama de secuencia aplicación mobile - partitions.	84
Ilustración 40. Diagrama de secuencia aplicación mobile - graphic HDD.	85
Ilustración 41. Diagrama de clase de usersActions.	86
Ilustración 42. Diagrama de clase de UsersForm.	86
Ilustración 43. Diagrama de clase de BaseUsersForm.	87
Ilustración 44. Diagrama de clase de UsersTable.	87
Ilustración 45. Diagrama de clase de BaseUsers.	87
Ilustración 46. Diagrama de clase de ip_addressActions.	88
Ilustración 47. Diagrama de clase de BaseIpAddressForm.	88
Ilustración 48. Diagrama de clase de IpAddressForm.	88
Ilustración 49. Diagrama de clase de BaseIpAddress.	89
Ilustración 50. Diagrama de clase de IpAddress.	89
Ilustración 51. Diagrama de clase de IpAddressTable.	89
Ilustración 52. Diagrama de clase de monitor_typeActions.	90
Ilustración 53. Diagrama de clase de MonitorTypeForm.	90
Ilustración 54. Diagrama de clase de BaseMonitorTypeForm.	90
Ilustración 55. Diagrama de clase de BaseMonitorType.	91
Ilustración 56. Diagrama de clase de MonitorType.	91
Ilustración 57. Diagrama de clase de MonitorTypeTable.	91
Ilustración 58. Diagrama de clase de monitor_ipActions.	92
Ilustración 59. Diagrama de clase de BaseMonitorIp.	92
Ilustración 60. Diagrama de clase de BaseMonitorIpForm.	92
Ilustración 61. Diagrama de clase de BaseMonitorIp.	93
Ilustración 62. Diagrama de clase de MonitorIp.	93
Ilustración 63. Diagrama de clase de MonitorIpTable.	93
Ilustración 64. Diagrama de clase de monitor_userActions.	94
Ilustración 65. Diagrama de clase de MonitorUserForm.	94
Ilustración 66. Diagrama de clase de BaseMonitorUserForm.	94
Ilustración 67. Diagrama de clase de BaseMonitorUser.	95
Ilustración 68. Diagrama de clase de MonitorUser.	95
Ilustración 69. Diagrama de clase de MonitorUserTable.	95
Ilustración 70. Diagrama de clase de graphicActions.	96
Ilustración 71. Diagrama de clase de reportActions.	96
Ilustración 72. Diagrama de clase de apiActions.	96



Ilustración 73. Diagrama de clase de mappingTask.	97
Ilustración 74. Diagrama de clase de SnmpFormat & Map.	97
Ilustración 75. Diagrama de clase de Auxiliar.	97
Ilustración 76. Diagrama de clases de paquete utils del entorno mobile.	98
Ilustración 77. Diagrama de clases del paquete screens del entorno mobile.	99
Ilustración 78. Diagrama de clases del paquete init del entorno mobile.	100
Ilustración 79. Arquitectura de red para la ejecución de Nova Realtime.	101
Ilustración 80. Representación de datos visualizados con el comando top.	104

GLOSARIO

AGENTE: es un módulo de software de administración de red que reside en un dispositivo administrado. Un agente posee un conocimiento local de información de administración (memoria libre, número de paquetes IP recibidos, rutas, etcétera), la cual es traducida a un formato compatible

ALGORITMO: es un conjunto finito de pasos definidos, estructurados en el tiempo y formulados con base a un conjunto finito de reglas no ambiguas, que proveen un procedimiento para dar la solución o indicar la falta de esta a un problema en un tiempo determinado. [Rodolfo Quispe-Otazu, 2004]

AUTOMATIZAR: Es el uso de sistemas o elementos computarizados, electromecánicos e informáticos que permita controlar maquinarias y/o procesos industriales reemplazando el recurso humano.

CPU: Abreviatura de Central Processing Unit (unidad de proceso central), se pronuncia como letras separadas. La CPU es el cerebro del ordenador. A veces es referido simplemente como el procesador o procesador central, la CPU es donde se producen la mayoría de los cálculos.

CONCEPTUAL: Es una técnica usada para la representación gráfica del conocimiento. Como se ve, un mapa conceptual es una red de conceptos. En la red, los nodos representan los conceptos, y los enlaces las relaciones entre los conceptos.

COMPATIBILIDAD: Es la condición que hace que un programa y un sistema, arquitectura o aplicación logren comprenderse correctamente ya sea directa o indirectamente (mediante un algoritmo). El algoritmo permite que un programa u programa se ha comprendido por un sistema.

FRAMEWORK: En términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

GOTO: Manera rápida de arreglar un programa sin estructurar el código. Java no tiene ninguna sentencia goto. Sin embargo Java tiene las sentencias break y continue que cubren los casos importantes de goto.



INTERFAZ: Es la conexión entre dos ordenadores o máquinas de cualquier tipo dando una comunicación entre distintos niveles.

IPTABLES: Es un espacio de usuario del programa de aplicación que permite a un administrador del sistema para configurar las tablas proporcionadas por el kernel de Linux servidor de seguridad (implementado como diferentes Netfilter módulos) y las cadenas y reglas que almacena.

MULTIHILO: En sistemas operativos, un hilo de ejecución, hebra o subproceso es la unidad de procesamiento más pequeña que puede ser planificada por un sistema operativo.

MYSQL: Es un sistema de gestión de bases de datos (SGBD) multiusuario, multiplataforma y de código abierto.

NAT: Se Define NAT cuando se realiza traslación de una única ip pública a una única ip privada.

PRIVACIDAD: La privacidad en Internet se refiere a controlar quien puede tener acceso a la información que posee un determinado usuario que se conecta a Internet.

PLUGIN: Un complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. También se lo conoce como plug-in (del inglés "enchufable"), add-on (agregado), complemento, conector o extensión.

POOL: se denomina connection pool (agrupamiento de conexiones) al manejo de una colección de conexiones abiertas a una base de datos de manera que puedan ser reutilizadas al realizar múltiples consultas o actualizaciones.

PROTOCOLO: Es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red por medio de intercambio de mensajes.



RAM: Es una forma de equipo de almacenamiento de datos. Hoy en día, toma la forma de circuitos integrados que permiten almacenados los datos que se acceden en cualquier orden con un rendimiento peor de los casos de constante de tiempo.

SERVIDOR: Un servidor es una computadora integrada en una red, esta se encarga de proveer servicios a otras computadoras las cuales se denomina clientes.

SNAPSHOTS: Una instantánea es el estado de un sistema en un punto particular en el tiempo. El término fue acuñado como una analogía a la de la fotografía. Se puede hacer referencia a una copia real del estado de un sistema o una capacidad proporcionada por determinados sistemas.

SNMPD: Es un agente SNMP que se une a un puerto y espera las solicitudes de software de administración SNMP. Al recibir una solicitud, se procesa la solicitud.

SYMFONY (PHP): Un framework que proporciona una arquitectura, componentes y herramientas para los desarrolladores construir aplicaciones web.

TECNOLOGÍA: Es un conjunto de conocimientos técnicos, con un orden lógico y científico, permitiendo diseñar y crear bienes y servicios que facilitan la adaptación al medio ambiente y satisfacer las necesidades requeridas por los usuarios que se benefician de su uso.

THREAD: Un thread se puede considerar como la agrupación de un trozo de programa junto con el conjunto de registros del procesador que utiliza y una pila de máquina.

TIPOGRÁFICO: La tipografía es el oficio que trata el tema de las letras, números y símbolos de un texto impreso (ya sea sobre un medio físico o electromagnético), tales como su diseño, su forma, su tamaño y las relaciones visuales que se establecen entre ellos.

UBUNTU: Es un sistema operativo mantenido por Canonical y la comunidad de desarrolladores. Utiliza un núcleo Linux, y su origen está basado en Debian.



RESUMEN

El proyecto **NOVA REALTIME** tiene como fin el diseño y desarrollo de un software para monitoreo de servidores GNU/Linux. El aplicativo contara con una plataforma tipo Web y otro de tipo Mobile. Y su propósito es asistir al personal de IT o administradores de red en las tareas diarias que se emplean en el monitoreo de servidores GNU/Linux.

Basados en la metodología del proceso manual, la cual hoy en día es una de las más empleadas por administradores de red, definiendo los requerimientos para implementación, antecedentes, especificaciones e ingeniería de software, manual de usuario, descripción y comparación del proceso automatizado y manual, esto con el fin de confrontar la efectividad y precisión de la herramienta.

PALABRAS CLAVE: Aplicación web y Mobile, información en tiempo real. Monitoreo de servidores GNU/Linux.

ABSTRACT

The project **Nova Realtime** is intended the desing y developer of software for monitor server GNU/Linux. The application shall have a plataform Web and other Mobile.The purpose is to assist the IT staff and network administrators in the daily Works used in the monitoring of servers GNU/Linux.

Based on the methodology of the manual process, which today is one of the most widely used by network administrators, defining the requirements for implementation, history, specifications and software engineering, user manual, description and comparison of automated and manual process, this in order to compare the effectiveness and accuracy of the tool.

KEYWORDS: Application Web and Mobile, real time Information, monitor server GNU/Linux.



INTRODUCCIÓN

La mayor parte del tiempo de un Sysadmin (Administrador del sistema) es empleada en el monitoreo de sus servidores. En especial si se está a cargo de un gran número de servidores se da la necesidad de herramientas que faciliten y automaticen los procesos, que permitan obtener la información requerida de forma rápida y oportuna.

El medio de las redes informáticas esta dependiente de un continuo desarrollo y evolución, esto conlleva a la necesidad de buscar e implementar estrategias que permitan a los administradores automatizar los procesos de monitoreo de servidores que en la mayoría de empresas, que cuentan con una red, realizan de manera manual por medio de comandos.

En base al análisis de las necesidades básicas de los administradores de red, el monitoreo se establecerá inicialmente en dispositivos de Disco duro HDD, memoria RAM y CPU. El control automático de estos dispositivos permitirá al personal de IT hacer un análisis preciso del estado y las vulnerabilidades que representan.

Actual mente existen programas y herramientas que se desempeñan en al monitoreo de servidores, pero no logran una satisfacción total por parte del usuario final, ya sea por que consumen de mucho recurso de máquina y no operan óptimamente, otras por sus altos costos o son muy complicadas para manipular. Los administradores de red requieren una el desarrollo de una herramienta efectiva y que les permita realizar el monitoreo de sus servidores de forma rápida y confiable, que garantice la veracidad de la información.



1. PROBLEMA DE LA INVESTIGACIÓN

1.1 NOMBRE DE LA APLICACIÓN

El proyecto **NOVA REALTIME** tiene como objetivo reflejar los alcances y metas propuestas que se pueden lograr con una aplicación de este tipo. El nombre se divide en dos partes, NOVA; pretende mostrar el impacto de esta aplicación como su sede en una súper nova de este perfecto universo, REALTIME es la disposición en tiempo real para acceder a los datos, la cual ofrece la aplicación.

1.2 ANTECEDENTES

Dentro de las funciones principales en la administración de sistemas informáticos es la planificación adecuada y precisa de las tareas de gestión, lo que puede evitar sorpresas desagradables en el momento de ejecutarlas.

El administrador de IT o Analista de Sistemas debe prevenir asesorando de forma oportuna y adecuada al personal administrativo de su empresa sobre las necesidades tecnológicas en la adquisición de programas u de material informático, teniendo en cuenta los recursos que se disponga por parte de los usuarios, esto con el fin de contar con las posibilidades económicas de la empresa.

Luego de que el administrador recibe los equipos se debe realizar un plan de recolección de requerimientos, en el cual se deben tener en cuenta como mínimo la siguiente información:

Establecer el estado actual del servidor, identificar eventos críticos que comprometen los procesos de la máquina, y analizar y solucionar problemas o errores de configuración de manera eficaz e inmediata.

1.3 FORMULACIÓN DEL PROBLEMA

¿Vale la pena preguntarse porque es importante automatizar, actualizar y mejorar los recursos tecnológicos de una empresa?



En busca de reducir tiempo de ejecución de cada uno de los procesos realizados para el cumplimiento de sus actividades, se hace indispensable adquirir herramientas que ayuden a optimizar y suplir dichas necesidades.

¿Cuáles son las funciones principales del administrador de un sistema informático?

Parte de las funciones que cumple un administrador de sistemas es suplir las necesidades tecnológicas de la compañía, entre otros tenemos la administración de servidores, la administración de sus recursos y controlar el buen funcionamiento de sus sistemas informáticos.

¿Se hace realmente mesetario capacitar a todo el personal de la compañía, en busca de mantener un buen manejo y control con los servidores u equipos?

¿Cómo saber cuándo actualizar el sistema y cuáles son las necesidades actuales?

En lo que se refiere a la automatización el personal informático requiere ejecutar un plan periódico de tareas que les permita definir los recursos actuales, las problemáticas existentes y con ello recolectar información para la actualización de los equipos y herramientas. Para el administrador encargado de la automatización el desarrollo de dichas tareas u funciones se vuelve repetitivo y dispendioso, se emplea mucho más tiempo en este tipo de procedimientos, pero es una forma de prevenir errores tipográficos y conceptuales, y para el mejoramiento general de los aplicativos.

Es por esta razón que tener un sistema que se encargue de la realización de dichas tareas, contribuye a sustituir la necesidad de cada servidor que requiere automatizarse en algunas de las funciones que se realizan diariamente con los servidores.

¿Es requerido tener una estructura que cumpla con una necesidad de seguridad de la información?

Dependiendo del tipo de información tratada por el sistema, el administrador debe definir sus políticas de seguridad, tanto para los servidores disponibles, como para la red corporativa de la empresa, ya que los usuarios tienen derecho a la privacidad e integridad de sus datos.

Es necesario establecer los medios para evitar posibles ataques o fallos informáticos que afecten o incluso inhabiliten el funcionamiento normal de la



máquina o pongan en riesgo la privacidad de la información contenida por los mismos.

¿Bajo qué sistema se realizaría la administración de servidores, manera efectiva y adecuada?

En lo que se refiere a la administración se manejara con el protocolo SNMP que significa Protocolo simple de administración en red, este es un protocolo que permite al personal de IT una adecuada administración de los dispositivos en red y el oportuno diagnóstico.

Se recolectara toda la información referente a la implementación de un sistema GNU/Linux, el protocolo SNMP y demás herramientas empleadas en el desarrollo.

1.4 PLANTEAMIENTO DEL PROBLEMA

1.4.1 Descripción.

La administración de servidores y servicios que estos proveen, son una de las tareas más importantes en la administración de sistemas informáticos y a su vez esta tiende a ser compleja debido que la información del estado de cada uno de los componentes es requerida lo más actualizada y precisa posible, para poder tomar decisiones preventivas y correctivas sobre la infraestructura afectada.

En la actualidad el proceso de monitoreo a los servidores se realiza de forma manual ingresando por una consola de administración ssh y consultado el estado del hardware con el comando: `#df -h` para obtener los datos de particiones, para visualizar el estado del CPU y RAM con el comando: `#top 1`.

1.4.2 Solución.

El proyecto actual plantea una forma eficiente de solucionar este tipo de inconvenientes de acceso a la información de servidores Gnu-Linux en tiempo real por medio del protocolo SNMP, y para el manejo de configuraciones y monitoreo se tienen como herramienta un entorno web y otro de tipo Mobile, que le permita al administrador de I.T. obtener la información del estado de Memoria, CPU y Unidades de Disco en cada servidor seleccionado, generando estadísticas y



reportes de tal manera que las percepciones de las ocurrencias pasen a ser datos concretos almacenados de forma centralizada y que con base en estos datos se pueden tomar las medidas que correspondan.

Esta solución realizará este proceso automático por medio de un pool, para obtener la información necesaria de cada uno de los servidores ingresados por el entorno Web y almacenarla en DB, permitiendo generar gráficas de comportamiento históricas y de tiempo real desde el entorno Web o Mobile, los reportes se generan a partir de la información de comportamiento histórico de cada uno de los servidores.

1.5 JUSTIFICACIÓN

Teórica. En la actualidad se cuenta con varios sistemas de monitoreo SNMP de licencia Gnu entre los cuales se encuentran: Cacti, Centreon, y Nagios, los cuales son muy usados, pero su implementación requiere desarrolladores con experiencia de desarrollo en el área de php, JavaScript, Perl y python entre otros, además de la experiencia como administradores de red o I.T., lo mismo sucede cuando se quiere desarrollar nuevas funcionalidades conocidas como Plugins. Mientras que en los sistemas de licencia paga se encuentran: Hp Open View y Network Node Manager, estas aplicaciones requieren de un costo para poder hacer uso de ellas, para agregar nuevas funcionalidades es necesario contactar a la casa matriz para que ellos desarrollen dicha funcionalidad, generando sobrecostos y desfases de tiempo. La limitación general de los elementos o soluciones nombrados anteriormente no se cuenta con soporte de acceso móvil a los datos.

Metodológica. Metodología de trabajo (RAD), Rapid Application Development, será adoptada para la realización del proyecto, para poder cumplir con los tiempos y plazos acordados, ya que nos permite aglomerar (la usabilidad, rapidez y la utilización de la aplicación), esto reduce el desarrollo a máximo 90 días en módulos de (gestión de levantamiento de información, datos, proceso, aplicación, pruebas y entregable).

Práctica. Se realizará un levantamiento de información previo, donde se espera obtener el modelo de monitoreo de (CPU, HDD y Memoria RAM), junto a los datos que se desean manipular, para así mismo diseñar una base de datos que se acople a los requerimientos del modelo obtenido por el levantamiento de la



información. Para posteriormente diseñar una interfaz de usuario que se acople a todos los módulos que conformarán la aplicación en el entorno Web y Mobile, esto permitirá que la experiencia de usuario sea óptima desde el punto de vista gráfico; se generará un sistema automático tipo Pool para obtener la información, de los distintos nodos a monitorear.

Las gráficas de comportamiento estarán divididas en dos tipos: histórica la cual tomará la información que se encuentra en la base de datos para ser generada en forma de líneas; tiempo real que basará su información en una petición hacia el servidor que se quiere consultar generando gráficas en pastel y los reportes serán generados a partir de la información histórica en formato xlsx para facilitar la lectura y manipulación de los mismos.

Innovación. La Interfaz Mobile en principio estará diseñada para BlackBerry OS en su versión 6 y contara con acceso restringido por usuario y contraseña para la protección de la información, permitiendo generar reportes en tiempo real del servidor que se desee consultar.

1.6 OBJETIVOS

1.6.1 Objetivo General.

Desarrollar un sistema de acceso Web y Mobile que permita el monitoreo de servidores Gnu/Linux y sus servicios en las entidades educativas.

1.6.2 Objetivos específicos.

- Investigar a fondo la problemática en el monitoreo de servidores Gnu-Linux, la cual permita desarrollar un modelo conceptual de la misma.
- Definir de forma organizada las especificaciones que integraran la plataforma web y Mobile de monitoreo en forma de módulos, tiempos de ejecución y el alcance que tendrá la aplicación.
- Diseñar una base de datos que permita centralizar la información de todos los servidores a los que Nova Realtime atiende.



- Desarrollar y diseñar un software que cumpla con cada uno de los requerimientos establecidos.
- Examinar cada uno de los componentes del proyecto, para la verificación de su correcto funcionamiento.

1.7 BENEFICIOS QUE CONLLEVA

- **Eficiencia**

La solución informática de **Nova Realtime** ofrece acceder a la información necesaria de forma rápida, precisa y segura, permitiendo al personal IT la toma de decisiones preventivas y oportunas.

- **Escalabilidad**

El sistema de **Nova Realtime** tiene posibilidades de expansión con toda compañía que le requiera, ya sean empresas con redes pequeñas o redes más grandes.

El diseño implementado para el proyecto abre las posibilidades a futuro de ampliar el monitoreo de más dispositivos u otros servicios, se pueden incluir nuevos módulos al aplicativo, nuevos informes y funcionalidades.

- **Acceso desde cualquier locación**

Nova Realtime permite a sus usuarios acceder desde cualquier punto que cuente con acceso a internet con su aplicativo web y Mobile, así el usuario podrá acceder al servidor, editar información, insertar y generar o visualizar los diferentes reportes desde cualquier lugar.

1.8 ALCANCES Y DELIMITACIONES

1.8.1 Alcances.



Monitorear el estado de los servidores Linux desde una plataforma web y Mobile que cuenta con dos módulos, los cuales se entregaran en un 100%.

- Módulos interfaz web:

- Ip_address
- Monitor_type
- Monitor_user
- Monitor_ip
- Users
- Monitoring
- Login & Logout

- Módulos interfaz mobile:

- init
- screen
- utils

El aplicativo tiene como fin centralizar el monitoreo de los servicios tales como (HDD, RAM, CPU) de los servidores.

1.8.2 Delimitaciones.

Espacial. El proyecto dará comienzos en el municipio de Soacha, su ubicación está situada en la provincia de Soacha. Es el municipio con más poblado del departamento de Cundinamarca, su área esta suburbana con la ciudad de Bogotá.

El aplicativo comenzara distribuirse en el sector educativo, entre por colegios e instituciones educativas del municipio, ya sean estas privadas o del estado.

Cronológica. Para la llevar a cabalidad cada uno de los objetivos propuestos para el desarrollo del aplicativo se definieron 6 fases una fase inicial de investigación previa para recolectar información necesaria y las 5 restantes corresponden al desarrollo del aplicativo y edición de manuales, las cuales están representas en el siguiente cronograma.

Descripción	Total Días	Inicio	Fin	Recursos
Fase Investigativa	5 días	27/02/2012 09:00	02/03/2012 19:00	
Que Linux	1 día	27/02/2012 09:00	27/02/2012 19:00	Ana Paulina Lizarazo y Andrés Piña Fonseca
Que es el Protocolo SNMP	1 día	28/02/2012 09:00	28/02/2012 19:00	Ana Paulina Lizarazo y Andrés Piña Fonseca
Ventajas de un sistema de monitoreo centralizado de Servidores Linux	1 día	29/02/2012 09:00	29/02/2012 19:00	Ana Paulina Lizarazo y Andrés Piña Fonseca
Limitaciones	1 día	01/03/2012 09:00	01/03/2012 19:00	Ana Paulina Lizarazo y Andrés Piña Fonseca
Alcances	1 día	02/03/2012 09:00	02/03/2012 19:00	Ana Paulina Lizarazo y Andrés Piña Fonseca
Fase I	4 días	01/03/2012 09:00	06/03/2012 19:00	
Levantamiento de Requerimientos	2 días	01/03/2012 09:00	02/03/2012 19:00	Ana Paulina Lizarazo
Diseño Modelo Base de datos (mysqlWorkbench)	1 día	05/03/2012 09:00	05/03/2012 19:00	Ana Paulina Lizarazo, MySQL
Diseño Interfaz Web (diseño página web)	1 día	06/03/2012 09:00	06/03/2012 19:00	Andrés Piña Fonseca, HTML, Css y JavaScript
Fase II	3 días	07/03/2012 09:00	09/03/2012 19:00	
Creación task Symfony para mapeo de los servicios SNMP y almacenamiento de la información	2 días	07/03/2012 09:00	08/03/2012 19:00	Andrés Piña Fonseca, Symfony (php)[1]
Automatización de mapeo de servicios SNMP con Crontab Linux	1 día	09/03/2012 09:00	09/03/2012 19:00	Symfony (php)[1],Cron Linux[1],Ana Paulina Lizarazo
Fase III	25 días	12/03/2012 09:00	13/04/2012 19:00	
Módulo ip_address crud y adaptación de interfaz web	1 día	12/03/2012 09:00	12/03/2012 19:00	Symfony (php)[1], Mysql, Ana Paulina Lizarazo

Módulo monitor_type crud y adaptación de interfaz web	1 día	13/03/2012 09:00	13/03/2012 19:00	Andrés Piña Fonseca, Symfony (php)[2],Mysql
Módulo monitor_user crud y adaptación de interfaz web	1 día	14/03/2012 09:00	14/03/2012 19:00	Andrés Piña Fonseca, Symfony (php)[2],Mysql
Módulo monitor_ip crud y adaptación de interfaz web	1,25 días	15/03/2012 09:00	16/03/2012 11:00	Symfony (php)[2], Mysql, Ana Paulina Lizarazo
Módulo users y adaptación de interfaz web	1 día	16/03/2012 09:00	16/03/2012 19:00	Symfony (php)[2], Mysql, Ana Paulina Lizarazo
Módulo monitoring y adaptación de interfaz web	4,5 días	19/03/2012 09:00	29/03/2012 13:00	Andrés Piña Fonseca, Symfony (php)[1],Mysql
Generación de reportes	5 días	29/03/2012 15:00	05/04/2012 13:00	Ana Paulina Lizarazo
Módulo monitoring by monitor type	2 días	05/04/2012 09:00	06/04/2012 19:00	Andrés Piña Fonseca, Symfony (php)[2],Mysql
Aseguramiento aplicación Web	4 días	09/04/2012 09:00	12/04/2012 19:00	Ana Paulina Lizarazo, Symfony (php)[2],Mysql
Módulo Login & Logout y adaptación de interfaz web	1 día	13/04/2012 09:00	13/04/2012 19:00	Ana Paulina Lizarazo, Symfony (php)[1],Mysql
Fase IV	21 días	16/04/2012 09:00	14/05/2012 19:00	
Diseño Interfaz Mobile	2 días	16/04/2012 09:00	17/04/2012 19:00	Andrés Piña Fonseca, Java SDK BlackBerry
Parseo XML y conexión con el servidor	3 días	18/04/2012 09:00	20/04/2012 19:00	Andrés Piña Fonseca, Java SDK BlackBerry, Mysql, Symfony (php) WebService[1%]
WebService XML Login y Logout	3 días	23/04/2012 09:00	25/04/2012 19:00	Symfony (php) WebService, Java SDK BlackBerry, Mysql, Ana Paulina Lizarazo
Look & Feel Login	3 días	26/04/2012 09:00	30/04/2012 19:00	Andrés Piña Fonseca, Java SDK BlackBerry
WebService XML servidores asignados a usuario	3 días	01/05/2012 09:00	03/05/2012 19:00	Symfony (php) WebService, Java SDK BlackBerry, Mysql, Ana Paulina Lizarazo
Look & Feel, listar servidores asignados	2 días	04/05/2012 09:00	07/05/2012 19:00	Andrés Piña Fonseca[50%],Java SDK

				BlackBerry[50%]
WebService XML Monitoring	2 días	08/05/2012 09:00	09/05/2012 19:00	Mysql, Symfony (php) SDK BlackBerry, Ana Paulina Lizarazo
Look & Feel Monitoring	3 días	10/05/2012 09:00	14/05/2012 19:00	Andrés Piña Fonseca, Java SDK BlackBerry
Fase V	10 días	14/05/2012 09:00	25/05/2012 19:00	
Manual de instalación	5 días	14/05/2012 09:00	18/05/2012 19:00	Ana Paulina Lizarazo
Manual de usuario	5 días	21/05/2012 09:00	25/05/2012 19:00	Ana Paulina Lizarazo

Tabla 1. Cronograma de actividades.

1.9 FACTIBILIDAD

En busca del cumplimiento de los objetivos propuestos por Nova Realtime, donde su factor principal es dar forma a un sistema de monitoreo de servidores que se encargue de los procesos de supervisión, se recopilara la información necesaria para definir las posibilidades técnicas, económicas y operativas.

1.9.1 Factibilidad técnica.

Para el desarrollo de Nova Realtime tendremos dos módulos de implementación para el monitoreo uno de tipo web y otro de tipo Mobile. Tan bien se detallaran cada uno de los aspectos de seguridad que se requieren para garantizar la seguridad y privacidad de la información. La implementación estará concentrada en SO de software libreen busca de memorizar al máximo los costos generados en la presentación del sistema.

1.9.2 Factibilidad económica.

El desarrollo del aplicativo se enfocara con herramientas de uso libre, esto con el fin de reducir los gastos al máximo. Otra de las ventajas es que la adquisición de hardware para el sistema propuesto se puede implementar con requerimientos mínimos de máquina.

Ya que las herramientas que se usaron para plantear el desarrollo de la aplicación son de licencia libre o GNU, los únicos costos que se manejaran y que serán costeados por los proponentes, son atribuidos al desarrollo de la aplicación web y Mobile de la siguiente forma:

Descripción	Costo
Alquiler equipo de desarrollo	\$ 80.000
IDE Eclipse – PHP	\$ 0
IDE Eclipse – BlackBerry	\$ 0
Sistema Operativo Debían 6	\$ 0
Instalación Lamp	\$ 0
Servidores de Pruebas Virtualizados en Virtualbox	\$ 0
Papelería	\$ 50.000
Framework Symfony	\$ 0
Transporte	\$ 60.000
Total	\$ 190.000

Tabla 2. Costos estimados para el desarrollo.

1.9.3 Factibilidad operativa.

Lo que se busca es que el sistema pueda ser adquirido con una inversión mínima de inversión en quipos, el aplicativo estará al case de todo tipo de usuario y no generaría sobrecostos de licenciamiento, no hay límite para creación usuarios y estos pueden acceder desde el lugar en que estén ubicados.

- **Descripción del producto o servicio:** Nova Realtime ofrece un aplicativo de fácil manejo, con un módulo de registro de usuario controlado por dos tipos de roles, el usuario administrador que dispondrá de tolos los módulos,

el otro de tipo consultor, su acceso estará definido por su usuario administrador. El sistema contará con dos módulos, uno de acceso web y otro Mobile.

La interfaz web estará restringida por accesos de usuario a los módulos que solo le correspondan al perfil asignado, permitiendo al usuario de tipo administrador del sistemas hacer uso del CRUD(create, read, update y delete) en la aplicación, generación de gráficas, reportes, el usuario de monitoreo solo podrá generar graficas de comportamiento y reportes.

La Interfaz Mobile en principio estará diseñada para BlackBerry OS en su versión 6 y contará con acceso restringido por usuario y contraseña para la protección de la información, permitiendo generar reportes en tiempo real del servidor que se desee consultar.

- **Objetivo del Proyecto:** Desarrollar un sistema de acceso Web y Mobile que permita al personal de I.T. de las entidades educativas la administración de servidores Gnu/Linux y sus servicios.
- **Beneficios del proyecto:** Nova Realtime a diferencia de otros sistemas que ofrecen un medio Mobile de acceso a la información lo que permite estar siempre en conexión con la red, aun desde otros lugares. Otro de los beneficios es contar con la información en tiempo real, permitiendo la toma de decisiones rápidas y oportunas.
- **Alcance del proyecto:** En comienzo se busca obtener un modelo de monitoreo de (CPU, HDD y Memoria RAM), junto a los datos que se desean manipular, para así mismo diseñar una base de datos que se acople a los requerimientos del modelo obtenido por el levantamiento de la información.

Posteriormente diseñar una interfaz de usuario que se acople a todos los módulos que conformarán la aplicación en el entorno Web y Mobile, esto permitirá que la experiencia de usuario sea óptima desde el punto de vista gráfico; se generará un sistema automático tipo Pool para obtener la información, de los distintos nodos a monitorear.

- **Grupo a cargo del proyecto:** El desarrollo e investigación para Nova Realtime está liderada por Andrés Piña Fonseca estudiante de tecnología en informática. El desarrollo e implementación del aplicativo, instructivos, documentación, informes investigativos y manuales se conforma por los



estudiantes, Andrés Piña Fonseca y Ana Paulina Lizarazo estudiantes de VI semestre de tecnología en informática de la regional de Uniminuto Soacha.

- **Restricciones:** En busca de cumplir con los objetivos propuestos por Nova Realtime el monitoreo estará limitado a monitoreo de servidores GNU/Linux, en supervisión de (CPU, HDD y Memoria RAM). La implementación de software estará limitada al uso de código abierto y libre de licencias, esto con el fin de reducir al máximo los costos para el aplicativo y su distribución.

2. MARCO DE REFERENCIA

2.1 MARCO HISTORICO

Ante un fenómeno continuo de desarrollo y evolución de las redes informáticas y su amplia implementación, se buscan alternativas que contribuyan a mejorar e incrementar el uso de una red y con ello obtener herramientas u aplicativos que cumplan con las actuales necesidades de los sistemas de monitoreo, como lo son alertar sobre las caídas y fallos en los servicios, permitiendo al administrador de red tener un control continuo y eficaz sobre su infraestructura.

En la actualidad tenemos varios sistemas de monitoreo que permiten al administrador de IT tener control y administración de las redes disponibles los cuales son muy utilizados y accesibles. A continuación se dará una breve descripción de algunos de ellos.

Nagios. Es considerado como uno de los más populares, si no el más popular sistema de monitorización de red de código abierto disponible. Fue diseñado originalmente para ejecutarse en Linux, pero otras variantes de Unix son soportadas también. Nagios proporciona supervisión de los servicios de red (SMTP, POP3, HTTP, NNTP, ICMP, SNMP, FTP, SSH) y recursos de host (carga del procesador, uso de disco, los registros del sistema), entre otros. El control remoto es manejado a través de túneles SSH o SSL cifrado. Nagios tiene un diseño simple que ofrece a los usuarios la libertad para desarrollar sus chequeos de servicio sin esfuerzo propio basado en las necesidades y mediante el uso de cualquiera de las herramientas de apoyo que guste. Para detectar y diferenciar entre hosts que están abajo y los que son inalcanzables, Nagios permite definir jerarquía de la red de acogida con los hosts "padre". Cuando los servicios o los problemas de acogida se plantean, la notificación será enviada a la persona que está a cargo de la red a través del correo electrónico, SMS, etc.

Su nombre original fue Netsaint, el sistema fue creado y mantenido actualmente por Ethan Galstad, junto con un grupo de desarrolladores de software que mantienen también varios plugins.

Nagios está licenciada bajo la GNU General Public License Version 2 publicada por la Free Software Foundation.

- **Ventajas.**

- Una de las ventajas más sobresalientes de la plataforma Nagios es la capacidad de monitorear servicios de red de manera distribuida, como lo es el monitoreo de protocolos como: PING, SMTP, DHCP, POP, entre otros, los cuales permiten identificar y detectar de manera proactiva y eficaz cualquier anomalía que se presente en una red.
- Otra de las ventajas es el uso e implementación del protocolo SNMP (Simple Network Management Protocol) dentro de su plataforma, que permite capturar paquetes de administración, este aspecto le da la posibilidad de analizar el rendimiento de algún equipo y su comportamiento en determinada situación.

A continuación se mencionaran algunos de los procesos que se pueden obtener de Nagios:

- Monitoreo de servicios de red (SMTP, POP3, HTTP, NTTP, ICMP, SNMP).
- Monitoreo de los recursos de un host (carga del procesador, uso de los discos, logs del sistema) en varios sistemas operativos, incluso Microsoft Windows con el plugin NRPE_NT.
- Monitoreo remoto, a través de túneles SSL cifrados o SSH.
- Diseño simple de plugins, que permiten a los usuarios desarrollar sus propios chequeos de servicios dependiendo de sus necesidades, usando sus herramientas preferidas (Bash, C++, Perl, Ruby, Python, PHP, C#, etc.).
- Chequeo de servicios paralizados.
- Posibilidad de definir la jerarquía de la red, permitiendo distinguir entre host caídos y host inaccesibles.



- Notificaciones a los contactos cuando ocurren problemas en servicios o hosts, así como cuando son resueltos (vía email, pager, SMS, o cualquier método definido por el usuario junto con su correspondiente plugin).
- Posibilidad de definir manejadores de eventos que ejecuten al ocurrir un evento de un servicio o host para resoluciones de problemas proactivas.
- Rotación automática del archive de log.
- Soporte para implementar host de monitores redundantes.
- Interfaz web opcional, para observar el estado de la red actual, notificaciones, historial de problemas, archivos de logs, etc.

Zabbix. Es una clase de mecanismo de vigilancia tipo empresarial que está completamente equipado y tiene soporte comercial. Es capaz de monitorear y dar seguimiento de la situación de los diferentes tipos de servicios de red, servidores y otro hardware de red. Zabbix tiene grandes funcionalidades de visualización incluidas las vistas definidas por el usuario, zoom, y la cartografía. Tiene un método de comunicación versátil que permite una configuración rápida y sencilla de los diferentes tipos de notificaciones de eventos predefinidos. Zabbix cuenta con tres módulos principales: el servidor, los agentes, y el usuario. Para almacenar los datos de seguimiento, puede utilizar MySQL, PostgreSQL, Oracle o SQLite como base de datos. Sin necesidad de instalar ningún software en el host de seguimiento, Zabbix permite a los usuarios comprobar la disponibilidad y capacidad de respuesta de los servicios estándar, como SMTP o HTTP. Para supervisar las estadísticas, tales como carga de la CPU, utilización de la red y espacio en disco, un agente de Zabbix debe estar instalado en la máquina host. Zabbix incluye soporte para el monitoreo a través de SNMP, TCP y controles ICMP, IPMI y parámetros personalizados como una opción para instalar un agente en los hosts.

Zabbix fue creado por el señor Alexei Vladishev y actualmente se desarrolla y se soporta por la compañía. Zabbix SIA. Es una solución open source que permite de forma rápida y sencilla monitorizar todo tipo de servidores, aplicaciones y equipos que hacen parte de una red. Zabbix permite centralizar la información en un servidor y monitorizar múltiples hosts, además tiene una interfaz de administración vía web browser. Por otro lado, ofrece características de monitorización avanzada,



alertas y visualización que actualmente no están presentes en otros sistemas de monitorización, ni siquiera comerciales.

- Ventajas.
 - Auto-descubrimiento de servidores y dispositivos de red.
 - Distribuidos de monitoreo centralizado con administración web.
 - Servidor para Linux, Solaris, HP-UX, AIX, Free BSD, Open BSD, OS X.
 - Nativas de alto rendimiento de los agentes (software de cliente para Linux, Solaris.
 - HP-UX, AIX, Free BSD, Open BSD, OS X, Tru64/OSF1, Windows NT 4.0, Windows 2000, Windows 2003, Windows XP, Windows Vista).
 - Agente-menos de vigilancia.
 - Garantizar la autenticación de los usuarios.
 - Flexibles permisos de usuarios.

Cacti, Es una herramienta web de gráficas que está diseñada como una interfaz completa para almacenamiento de datos de RRDtool y la utilidad gráfica que permite a los usuarios monitorear y graficar la carga de la CPU, la utilización de ancho de banda de red, el tráfico de red, y mucho más. Puede ser utilizado para configurar la recopilación de datos en sí, lo que permite configuraciones particulares, a controlar sin ningún tipo de configuración manual de RRDtool. Cacti permite sondear los servicios en el período preestablecido y el gráfico de los datos resultantes. Se utiliza principalmente para representar gráficamente los datos de series temporales de parámetros tales como la carga de la CPU y la utilización de ancho de banda de red. Cacti se puede ampliar para controlar cualquier fuente a través de scripts de Shell y ejecutables. También es compatible con arquitectura de Plugins y tiene una comunidad grande y activa que se ha reunido en torno a los foros de Cacti para proporcionar scripts, plantillas y consejos sobre creación de Plugins.

RRDtool. Es el acrónimo de Round Robin Database tool, o sea que se trata de una herramienta que trabaja con una BD que maneja Planificación Round-robin. Esta técnica trabaja con una cantidad fija de datos y un puntero al elemento actual. El modo en que trabaja una base de datos utilizando Round Robin es el siguiente; se trata la BD como si fuera un círculo, sobrescribiendo los datos almacenados, una vez alcanzada la capacidad de la BD. La capacidad de la BD depende de la cantidad de información como historial que se quiera conservar.

- Ventajas
 - Medir Disponibilidad, Carga, Errores y Más con Historia.
 - Cacti puede visualizar los interfaces de enrutadores, conmutadores y su tráfico, incluyendo tráfico de errores.
 - Puede medir capacidad de disco, carga de CPU (en equipos de red y servidores), y mucho más y reaccionar con notificaciones según rangos especificado.
 - Gráficos.
 - Permite utilizar todas las funciones de rrdgraph para definir los gráficos y automatiza algunas de ellas.
 - Permite organizar la información en árboles jerárquicos.
 - Fuentes de datos.
 - Permite utilizar todas las funciones de rrdcreate y rrdupdate, incluyendo la definición de varias fuentes de datos por archivo RRD.
 - Colección de datos,
 - Las fuentes datos pueden ser actualizadas vía SNMP o mediante la definición de scripts.



- Soporte SNMP incluido utilizando php-snmp o net-snmp.
- Un componente opcional, cactid, implementa las rutinas SNMP en lenguaje C con multi-threading. Muy importante para grandes números de dispositivos.
- Plantillas.
- Permite crear plantillas para reutilizar las definiciones de gráficos, fuentes de datos y dispositivos.
- Gestión de usuarios.
- Permite definir autenticación (local o LDAP) y distintos niveles de autorización para usuarios

2.2 MARCO TEORICO

Para cumplir con los objetivos de **Nova Realtime** y obtener los resultados deseados se ha realizado un estudio de lo que es el monitoreo de servidores, los diferentes componentes que conforman los procedimientos actuales, dando a conocer las ventajas de utilizar software libre entre otros, conocer los las herramientas y programas que se implementarán para el desarrollo del aplicativo como lo son el SNMP Protocolo simple de administración de red, SYMFONY un framework proporciona una arquitectura, componentes y herramientas para los desarrolladores construir aplicaciones web, lo que es el WebService conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones, los oid(ObjectID) un identificador de objeto u OID es un identificador utilizado para nombrar a un objetos, protocolos de red implementados, métodos de desarrollo etc.

- **Software en el que está basado el desarrollo:** SYMFONY (PHP), SDK JAVA Mobile *BlackBerry*, MYSQL, SNMPD, SNMP, IPTABLES, Distribución Ubuntu 12.04 GNU-Linux, Interfaz web (HTML, CSS, JAVASCRIPT) y *WebService* XML.

2.2.1 Symfony.

Symfony es un completo framework creado y diseñado para optimizar el desarrollo de las aplicaciones web por medio de algunas de sus principales características. En comienzo, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se genere código innecesario cada vez que surge la necesidad de crear una nueva aplicación web.

En el año 2003, Fabien Potencier, creador de symfony y actual CEO de Sensio Labs, investigó acerca de las herramientas open source existentes para el desarrollo de aplicaciones web en PHP, pero ninguna de las existentes cumplió con sus expectativas. Cuando PHP 5 fue liberado, consideró que las herramientas que existían en ese momento habían madurado lo suficiente para ser integradas en un solo framework. Le llevó un año desarrollar el núcleo de symfony. Basó su trabajo en el Modelo Vista Controlador, el ORM de Propel y el ayudante para realizar plantillas de Ruby on Rails.

Características de Symfony.

Symfony fue diseñado para ajustarse a los siguientes requisitos:

- Fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Unix estándares).
- Independiente del sistema gestor de bases de datos. Su capa de abstracción y el uso de Propel, permiten cambiar con facilidad de SGBD en cualquier fase del proyecto.
- Utiliza programación orientada a objetos, de ahí que sea imprescindible PHP 5.
- Sencillo de usar en la mayoría de casos, aunque es preferible para el desarrollo de grandes aplicaciones Web que para pequeños proyectos.



- Aunque utiliza MVC (Modelo Vista Controlador), tiene su propia forma de trabajo en este punto, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador sólo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con las bibliotecas de otros fabricantes.
- Una potente línea de comandos que facilitan generación de código, lo cual contribuye a ahorrar tiempo de trabajo.

Características para el desarrollo automatizado de aplicativos web.

- Las características más comunes para el desarrollo de proyectos web están automatizadas en symfony, tales como:
- Permite la internacionalización para la traducción del texto de la interfaz, los datos y el contenido de localización.
- presentación usa templates y layouts que pueden ser construidos por diseñadores de HTML que no posean conocimientos del framework.
- Los formularios soportan la validación automática, lo cual asegura mejor calidad de los datos en las base de datos y una mejor experiencia para el usuario.
- El manejo de cache reduce el uso de banda ancha y la carga del servidor.
- La facilidad de soportar autenticación y credenciales facilita la creación de áreas restringidas y manejo de seguridad de los usuarios.

- El enrutamiento y las URLs inteligentes hacen amigable las direcciones de las páginas de la aplicación.
- Las listas son más amigables, ya que permite la paginación, clasificación y filtraje automáticos.
- Los plugins proveen un alto nivel de extensibilidad.
- La interacción con AJAX es mucho más sencilla.

Versión	Fecha de Lanzamiento	Versión PHP	Fin del Mantenimiento Para la Versión
1.0	Enero de 2007	>= 5.0	Enero de 2010
1.1	Junio de 2008	>= 5.1	Junio de 2009
1.2	Diciembre de 2008	>= 5.2	Noviembre de 2009
1.3	Noviembre de 2009	>= 5.2.4	Noviembre de 2010
1.4	Noviembre de 2009	>= 5.2.4	Noviembre de 2012
2.0	Julio de 2011	> 5.2	

Tabla 3. Versiones de Symfony.

2.2.2 Java.

Los lenguajes de programación C y Fortran que últimamente es lo más utilizado para diseñar algunos de los sistemas más complejos en lenguajes de programación estructurada y orientada a objetos, la cual ha venido creciendo hasta formar complicados procedimientos. De ahí provienen términos como

"código de espagueti" o "canguros" referentes a programas con múltiples saltos y un control de flujo difícilmente trazable.

No sólo se necesitaba un lenguaje de programación para tratar u profundizar en esta complejidad, sino un nuevo estilo para la innovación de la programación. Este cambio de paradigma de la programación estructurada a la programación orientada a objetos, así fue como comenzó hace 30 años con un lenguaje llamado Simula67.

SIMULA 67: El primer programa que se diseñó con una orientación a objetos, fue desarrollado por Kristen Nygaard y Ole-Johan Dahl en 1967 en el Centro de Computación Noruego de Oslo. Sin embargo, el primer lenguaje orientado a objetos puro es Smalltalk, cuya primera versión pública en el año 1980.

El lenguaje C++ fue un intento de tomar estos principios y emplearlos dentro de las restricciones de C. Todos los compiladores de C++ eran capaces de compilar programas de C sin clases, es decir, un lenguaje capaz de interpretar dos estilos diferentes de programación. Esta compatibilidad ("hacia atrás") que habitualmente se vende como una característica de C++ es precisamente su punto más débil. No es necesario utilizar un diseño orientado a objetos para programar en C++, razón por la que muchas veces las aplicaciones en este lenguaje no son realmente orientadas al objeto, perdiendo así los beneficios que este paradigma aporta.

Así Java utiliza convenciones casi idénticas para declaración de variables, paso de parámetros, y demás, pero sólo considera las partes de C++ que no estaban ya en C.

Punteros: Las direcciones de memoria son la característica más poderosa de C++. El inadecuado uso de los punteros provoca la mayoría de los errores de colisión de memoria, errores muy difíciles de detectar. Además, casi todos los virus que se han escrito aprovechan la capacidad de un programa para acceder a la memoria volátil (RAM) utilizando punteros. En Java, no existen punteros, evitando el acceso directo a la memoria volátil.

Variables globales: Con ellas cualquier función puede producir efectos laterales, e incluso se pueden producir fallos catastróficos cuando algún otro método cambia el estado de la variable global necesaria para la realización de otros procesos. En Java lo único global es el nombre de las clases.

Goto: Manera rápida de arreglar un programa sin estructurar el código. Java no tiene ninguna sentencia goto. Sin embargo Java tiene las sentencias *break* y *continue* que cubren los casos importantes de goto.

Asignación de memoria: La función *malloc* de C, asigna un número especificado de bytes de memoria devolviendo la dirección de ese bloque. La función *free* devuelve un bloque asignado al sistema para que lo utilice. Si se olvida de llamar a *free* para liberar un bloque de memoria, se están limitando los recursos del sistema, ralentizando progresivamente los programas. Si por el contrario se hace un *free* sobre un puntero ya liberado, puede ocurrir cualquier cosa. Más tarde C++ añadió *new* y *delete*, que se usan de forma similar, siendo todavía el programador, el responsable de liberar el espacio de memoria. Java no tiene funciones *malloc* ni *free*. Se utiliza el operador *new* para asignar un espacio de memoria a un objeto en el montículo de memoria. Con *new* no se obtiene una dirección de memoria sino un descriptor al objeto del montículo. La memoria real asignada a ese objeto se puede mover a la vez que el programa se ejecuta, pero sin tener que preocuparse de ello. Cuando no tenga ninguna referencia de ningún objeto, la memoria ocupada estará disponible para que la reutilice el resto del sistema sin tener que llamar a *free* o *delete*. A esto se le llama recogida de basura. El recolector de basura se ejecuta siempre que el sistema esté libre, o cuando una asignación solicitada no encuentre asignación suficiente.

Conversión de tipos insegura: Los moldeados de tipo (*type casting*) son un mecanismo poderoso de C y C++ que permite cambiar el tipo de un puntero. Esto requiere extrema precaución puesto que no hay nada previsto para detectar si la conversión es correcta en tiempo de ejecución. En Java se puede hacer una comprobación en tiempo de ejecución de la compatibilidad de tipos y emitir una excepción cuando falla.

Primeros proyectos en que se aplicó Java. El proyecto Green fue el primero en el que se aplicó Java, y consistía en un sistema de control completo de los aparatos electrónicos y el entorno de un hogar. Con este fin se construyó un ordenador experimental denominado *7 (*Star Seven*). El sistema presentaba una interfaz basada en la representación de la casa de forma animada y el control se llevaba a cabo mediante una pantalla sensible al tacto. En el sistema aparecía ya Duke, la actual mascota de Java.

A pesar de las expectativas que se tenían para el desarrollo del proyecto Green se encontraron dificultades no previstas. El mercado para los dispositivos electrónicos inteligentes de uso doméstico no se desarrollaba tan rápido como se tenía

anticipado. Esto comprometió desarrollo del proyecto, el cual quedo en riesgo de ser cancelado. Para fortuna del proyecto con la popularidad de World Wide Web que exploto en 1993, es así que se descubre el potencial de Java para crear contenido dinámico para las páginas Web.

Fundamentos típicos de Java. Por lo generar los sistemas java constan de diversas partes, un ambiente, el lenguaje, la interfaz de programación de aplicaciones Java (API) entre varias bibliotecas de clases.

JavaScript. JavaScript es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

SDK Java Mobile BlackBerry. Se trata de un dispositivo móvil información del perfil (MIDP) compatible con Java ME entorno para los desarrolladores que deseen mantener la portabilidad sin problemas en sus aplicaciones inalámbricas. Además, el *BlackBerry* JDE proporciona un conjunto completo de interfaces y utilidades para aprovechar algunas de las características únicas del teléfono inteligente **BlackBerry**.

2.2.3 Mysql.

Se trata de un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL es un software implementado en código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL. El lenguaje de programación que utiliza MySQL es *Structured Query Language* (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

Historia. MySQL llega a comienzos de la década del 90, Michael Widenis comenzó a usar mSQL para enlazar tablas dando uso a sus propias rutinas de bajo nivel (ISAM). Luego de las primeras pruebas realizadas, llegó a la conclusión de que mSQL no era lo bastante flexible ni rápido para lo que necesitaba, por lo



que tuvo que desarrollar nuevas funciones. Esto resulto en una interfaz SQL a su base de datos, totalmente compatible a mSQL.

El origen del nombre MySQL no se sabe con certeza de donde proviene, por un lado se dice que en sus librerías han llevado el prefijo “my” durante los diez últimos años, por otra parte, la hija de uno de los desarrolladores se llama My. Así que no está claramente definido cuál de estas dos causas han dado lugar al nombre de este conocido gestor de bases de datos.

Características. En un comienzo, MySQL tuvo dificultades pues no contaba con algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido gracias a su simplicidad, esto permitió que los elementos faltantes fueran complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

De las últimas versiones publicadas se mencionaran algunas de sus características:

- El principal objetivo de MySQL es un sistema de alta velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas

Ventajas.

- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.



- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad

Desventajas.

- Un gran porcentaje de las utilidades de MySQL no están documentadas.
- No es intuitivo, como otros programas (ACCESS).

2.2.4 Snmp.

Protocolo estándar de administración de red, el protocolo estándar de administración de red del TCP/IP que actualmente se encuentra en uso (Simple Network Management Protocol - Protocolo simple de administración de red). Protocolo que permite supervisar, analizar y comunicar información de estado entre una gran variedad de hosts, pudiendo detectar problemas y proporcionar mensajes de estado.

Forma parte de una suite de protocolos de internet definidos por la IETF. Se utiliza para monitorear los dispositivos adjuntados a una red, supervisando el desempeño de la red, en buscar y resolver problemas.

SNMP consiste de un conjunto de estándares para la administración de redes, incluyendo un protocolo de capa de aplicación, un esquema de base de datos y un conjunto de objetos de datos.



Las versiones más utilizadas son SNMPv1 y SNMPv2. La versión 3 (SNMPv3) posee cambios importantes con respecto a sus predecesoras, especialmente en la seguridad.

El protocolo Snmpv1 fue diseñado a mediados de los 80 por Case, McCloghrie, Rose, and Waldbusser, como una solución a los problemas de comunicación entre diferentes tipos de redes.

En un principio, su principal meta era el lograr una solución temporal hasta la llegada de protocolos de gestión de red con mejores diseños y más completos. Pero esos administradores de red no llegaron y SNMPv1 se convirtió en la única opción para la gestión de red.

El manejo de este protocolo era simple, se basaba en el intercambio de información de red a través de mensajes (PDU's). Además de ser un protocolo fácilmente extensible a toda la red, debido a esto su uso se estandarizó entre usuarios y empresas que no querían demasiadas complicaciones en la gestión de sus sistemas informáticos dentro de una red.

No obstante este protocolo tenía deficiencias y no estaba y entre sus propósitos, el poder gestionar la inmensa cantidad de redes que cada día iban apareciendo. Para subsanar sus carencias surgió la versión 2 (SNMP v2). Las mayores innovaciones respecto a la primera versión son:

Introducción de mecanismos de seguridad, totalmente ausentes en la versión 1. Estos mecanismos protegen la privacidad de los datos, confieren autenticación a los usuarios y controlan el acceso.

Mayor detalle en la definición de las variables.

Se añaden estructuras de la tabla de datos para facilitar el manejo de los datos. El hecho de poder usar tablas hace aumentar el número de objetos capaces de gestionar, con lo que el aumento de redes dejó de ser un problema.

Realmente esta versión 2 no supuso más que un parche, es más hubo innovaciones como los mecanismos de seguridad que se quedaron en pura teoría, no se llegaron a implementar. Por estas razones, este año se ha producido la estandarización de la versión 3. Con dos ventajas principales sobre sus predecesores:

Añade algunas características de seguridad como privacidad, autenticación y autorización a la versión 2 del protocolo.

Uso de Lenguajes Orientados a Objetos (Java, C++) para la construcción de los elementos propios del protocolo (objetos). Estas técnicas confieren consistencia y llevan implícita la seguridad, por lo que ayudan a los mecanismos de seguridad.

SNMP se define como un protocolo de bajo nivel el cual proporciona dos funciones básicas: obtener la información de una variable o almacenar un valor en una variable. Todas las operaciones de SNMP están dadas por una secuencia de valores almacenados en variables.

El protocolo SNMP utiliza un servicio no orientado a la conexión (UDP) para envió de un grupo de mensajes (PDUs) el cual efectúa entre los administradores y agentes. El uso de este método es aseguramiento de las tareas de administración de red, garantizando que no se afectará al rendimiento global de la misma, ya que se evita la utilización de mecanismos de control y recuperación como los de un servicio orientado a la conexión, por ejemplo TCP.

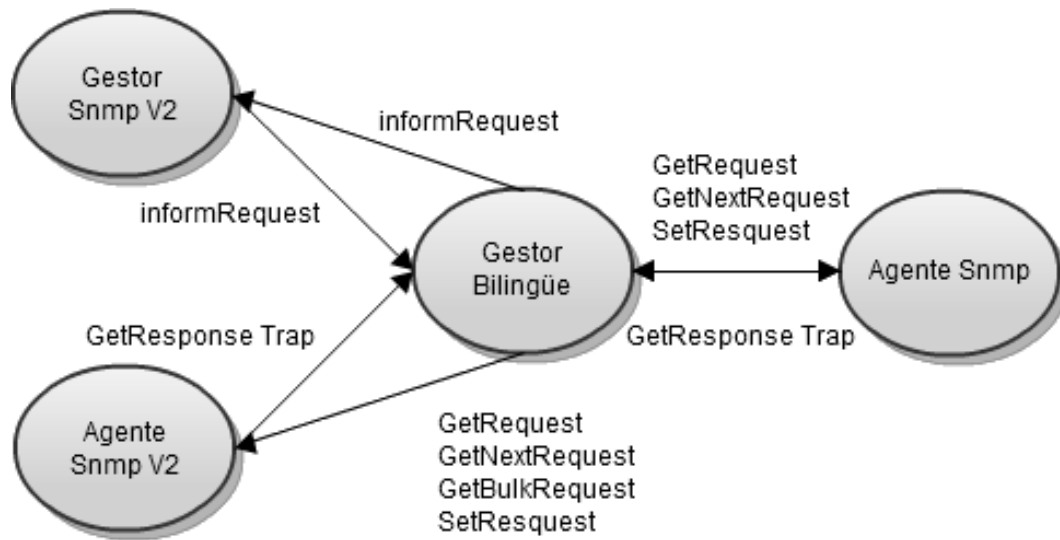


Ilustración 1. Interacción entre agentes y gestores Snmp.

Los que tiene uso más frecuente por el protocolo SNMP son:

Puerto 161 se utiliza para las transmisiones normales de comando SNMP.
Puerto 162 se utiliza para los mensajes de tipo “trap” o interrupción.

Un estándar asociado a SNMP define un conjunto de variables que son mantenidas y administradas por una entidad MIB (Management Information Base).

Estructura y representación de nombres de objetos MIB

Una Base de Información de Administración (Management Information Base, MIB) es un conjunto u colección de información que corresponde a una organizada jerárquica. Las MIB's son accedidas usando un protocolo de administración de red, en este caso, SNMP.

Un objeto administrado (algunas veces llamado objeto MIB, objeto, o MIB) es uno de cualquier número de características específicas de un dispositivo administrado. Los objetos administrados se encuentran conformados por una o más instancias de objeto, que son esencialmente variables.

Actualmente se tienen dos tipos de objetos administrados: Escalares y tabulares. Los objetos escalares definen de manera simple instancia de objeto. Los objetos tabulares definen múltiples instancias de objeto relacionadas que están agrupadas conjuntamente en tablas MIB.

El identificador de un objeto (object ID) tiene como único fin identificar el objeto administrado dentro de la jerarquía MIB. La jerarquía MIB es posible representarla como un árbol con una raíz anónima y los niveles, los cuales son asignados por diferentes organizaciones.

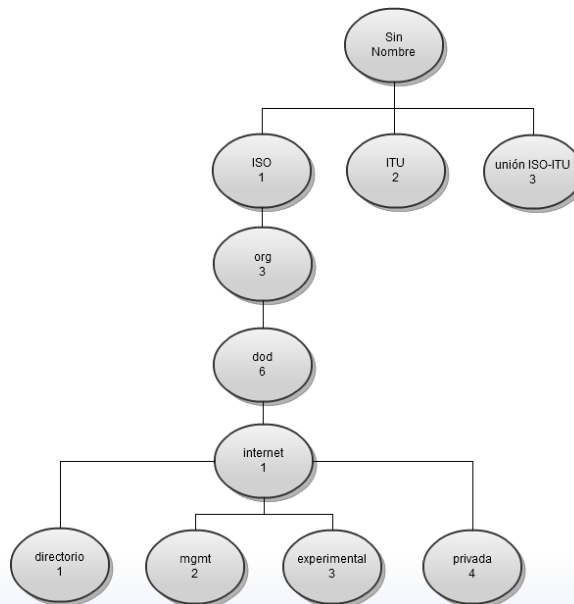


Ilustración 2. Parte de la jerarquía del espacio de nombres de los identificadores de objetos.

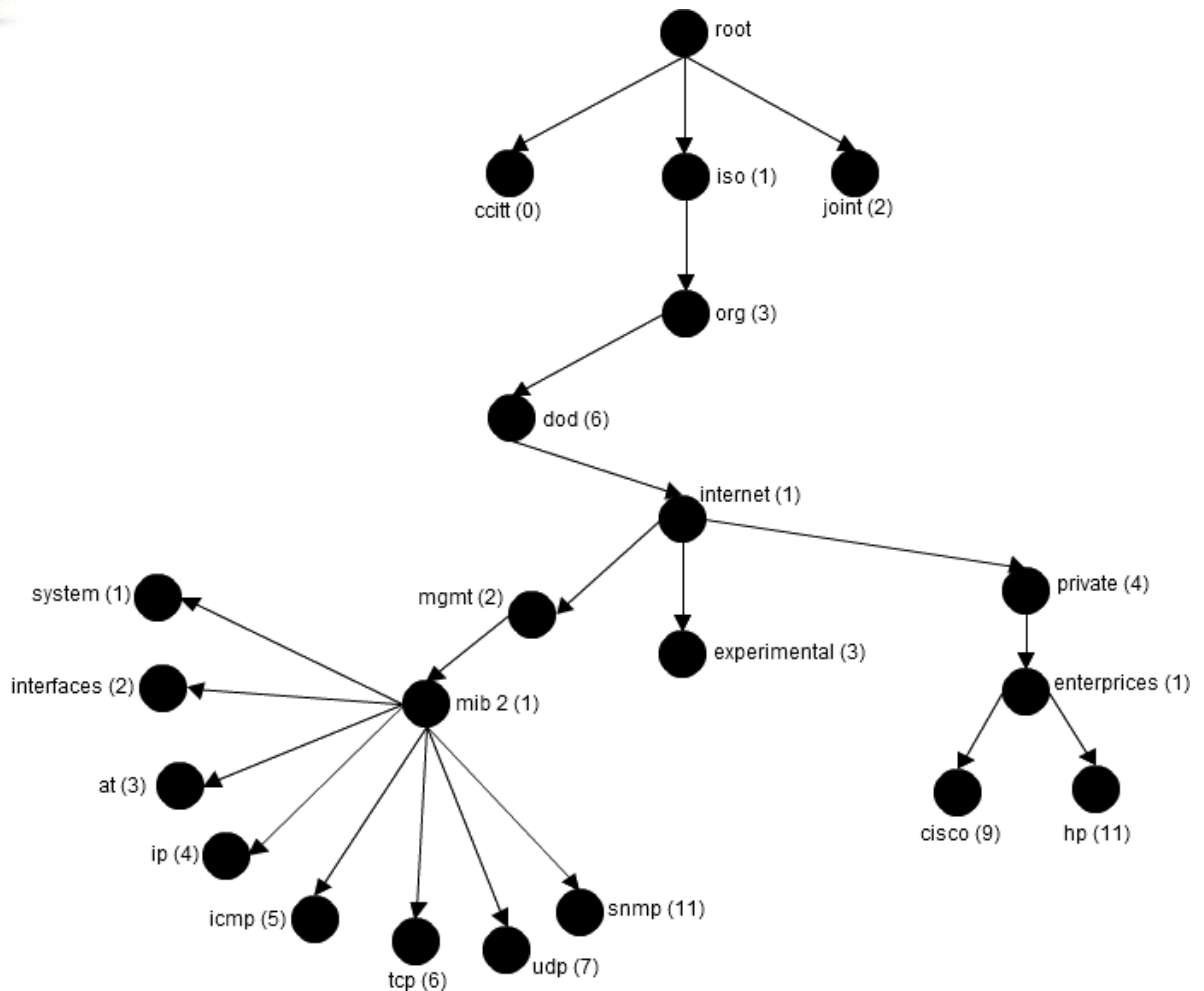


Ilustración 3. Árbol de Nombres Snmp.

Oid (object Identifiers)

Básicamente, un OID, o Identificador de Objeto, es una secuencia de números que se asignan jerárquicamente y que permite identificar objetos en la red, siendo usados con gran cantidad de protocolos.

Para qué se utilizan los OIDs

Los Identificadores de Objeto se utilizan en gran variedad de protocolos, aunque quizá los usos más comunes son los siguientes:

- Objetos y atributos que se gestionan vía SNMP.



- Clases, sintaxis y atributos en el Directorio (LDAP)
- Árboles de indexación en CIP (Common Indexing Protocol)
- Elementos dentro de una PKI (Public Key Infraestructure): Identificación unívoca de Autoridades de Certificación, Políticas de Certificación y Certification Policy Statements (CPS), atributos, nuevas extensiones, etc.

Identificadores de Objetos existentes

Una vez que se dispone de un OID asignado por alguna de las agencias registradoras existentes (como IANA, ANSI o BSI), se tiene el derecho de asignar libremente esa rama de la jerarquía según los intereses concretos de la organización en cuestión.

Existen multitud de OIDs asignados. Si quiere recorrer la raíz del árbol de OIDs lo puedes hacer en este navegador de OIDs.

¿Cómo obtener un OID?

En Octubre de 2000, la IANA asignó el siguiente OID a RedIRIS:

1.3.6.1.4.1.7547

Sobre ese OID, RedIRIS ha definido 5 ramas diferentes:

1.3.6.1.4.1.7547.1: define OIDs para las instituciones afiliadas a RedIRIS.

1.3.6.1.4.1.7547.2: define OIDs para objetos relacionados a tecnología PKI-X.509.

1.3.6.1.4.1.7547.3: define OIDs para objetos utilizados en proyectos nacionales e internacionales de indexación.

1.3.6.1.4.1.7547.4: define OIDs para objetos de directorio.

1.3.6.1.4.1.7547.5: define OIDs para objetos relacionados con SNMP.

Puede consultar los OIDs registrados en RedIRIS, y si lo desea también puede solicitar un OID.

Ventajas de un sistema de monitoreo con SNMP

- Es un sistema centralizado y estandarizado el cual permite la comunicación de todos los dispositivos en red de una forma más sencilla.
- El control y administración en forma remota de los sitios en red es una gran ventaja para el manejo de grades redes.
- Algunas UPS ya traen incorporada una salida para conexión de SNMP, otras cuentan con ranuras internas para adatar las placas para salida del protocolo, también es posible colocar un adaptador externo con la características requeridas.
- Para el intercambio de información consume muy pocos recursos en red.
- Da libertad al usuario de escoger las variables a monitorizar.

Desventajas

- Una desventaja son grandes fallos de seguridad en la trasmisión de datos en Red que podía permitir al intruso acceder a la información y hasta llegar a causar daños en las terminales. Esta deficiencia tuvo mejoras para prevención de la seguridad en la segunda versión SNMPv2.
- Actualmente posee un nivel de funcionalidad limitada.
- Actualmente la mayor cantidad de reportes que se generan son históricos.
- Su implementación es complicada.

Snmpd. Es un servicio o agente que regularmente se instala de modo predefinido en la mayoría de las distribuciones GNU/Linux, aunque no está predeterminado para ejecutarse en los servicios de arranque del sistema es una pieza fundamental para la gestión de servicios de los servidores.

2.2.5 Firewall

Un firewall es un dispositivo que filtra el tráfico entre redes, como mínimo dos. El firewall puede ser un dispositivo físico o un software sobre un sistema operativo. En general debemos verlo como una caja con DOS o más interfaces de red en la

que se establecen una reglas de filtrado con las que se decide si una conexión determinada puede establecerse o no. Incluso puede ir más allá y realizar modificaciones sobre las comunicaciones, como el NAT.

Esa sería la definición genérica, hoy en día un firewall es un hardware específico con un sistema operativo o una IOS que filtra el tráfico TCP/UDP/ICMP/..IP y decide y define si un paquete pasa, se modifica, se convierte o se descarta. Para que un firewall entre redes funcione como tal debe tener al menos dos tarjetas de red.

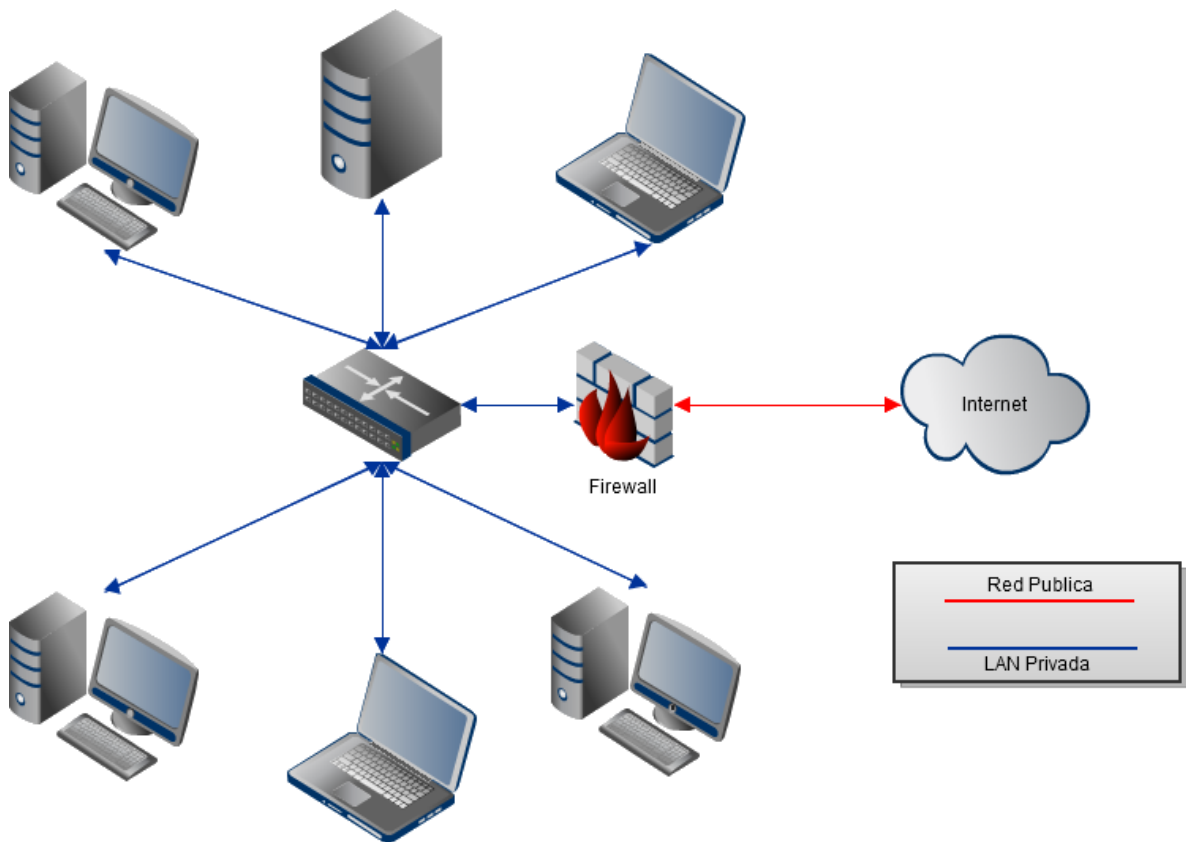


Ilustración 4. Representación de una red LAN con Firewall.

Ventajas

- Protección de información privada: Define que usuarios de la red y que información va a obtener cada uno de ellos.
- Optimización de acceso: Define de manera directa los protocolos a utilizarse



- Protección de intrusos: Protege de intrusos externos restringiendo los accesos a la red.

Desventajas

- No protege de ataques que no pasen a través del firewall.
- No protege amenazas y ataques de usuarios negligentes.
- No protege de la copia de datos importantes si se ha obtenido acceso a ellos.
- No protege de ataques de ingeniería social (ataques mediante medios legítimos, es decir cuando el atacante tiene acceso directo al sistema).

2.2.6 Iptables.

Es un espacio de usuario del programa de aplicación que permite a un administrador del sistema para configurar las tablas proporcionadas por el kernel de Linux servidor de seguridad (implementado como diferentes Netfilter módulos) y las cadenas y reglas que almacena.

El uso de Iptables ofrece una herramienta firewall con la cual se puede redirigir conexiones, hacer nat, filtrar paquetes. Su adjetivo principal es el aseguramiento del sistema y es el encargado de cerrar los puertos que no están en uso.

- **Ventajas:** protección contra intrusos y organización en los accesos.
- **Desventaja:** Dependiendo de la cantidad de las reglas puede ser muy difícil leer una estructura de muchas reglas complejas.

2.2.7 ¿Qué es Linux?

Es un sistema operativo de 32 o 64 bits para redes, similar a UNIX, con código abierto, optimizado para internet (utilizado por los piratas con mucha frecuencia) que puede funcionar en distintos dispositivos de hardware, es un sistema (así



como lo es Windows, Solaris, Mac OS X) y fue creado por Linus Torvalds en 1991 como una alternativa a los sistemas Unix de la época.

Como sistema operativo, Linux es muy eficiente y tiene un excelente diseño. Es multitarea, multiusuario, multiplataforma y multiprocesador; en las plataformas Intel corre en modo protegido; protege la memoria para que un programa no pueda hacer caer al resto del sistema; carga sólo las partes de un programa que se usan; comparte la memoria entre programas aumentando la velocidad y disminuyendo el uso de memoria; usa un sistema de memoria virtual por páginas; utiliza toda la memoria libre para cache; permite usar bibliotecas enlazadas tanto estática como dinámicamente; se distribuye con código fuente; usa hasta 64 consolas virtuales; tiene un sistema de archivos avanzado pero puede usar los de los otros sistemas y soporta redes tanto en TCP/IP como en otros protocolos.

Ventajas de Linux

- Linux es básicamente un duplicado de UNIX, lo que significa que incorpora muchas de las ventajas de este importante sistema operativo.
- En Linux pueden correr varios procesos a la vez de forma ininterrumpida como un servidor de red al tiempo que un procesador de textos, una animación, copia de archivos o revisar el correo electrónico.
- Seguridad porque es un sistema operacional diseñado con la idea de Cliente - Servidor con permisos de acceso y ejecución a cada usuario. Esto quiere decir que varios usuarios pueden utilizar una misma máquina al tiempo sin interferir en cada proceso.
- Linux es software libre, casi gratuito. Linux es popular entre programadores y desarrolladores e implica un espíritu de colaboración.
- Linux integra una implementación completa de los diferentes protocolos y estándares de red, con los que se puede conectar fácilmente a Internet y acceder a todo tipo de información disponible.
- Su filosofía y sus programas están dictados por el movimiento "Open Source" que ha venido creciendo en los últimos años y ha adquirido el suficiente fortalecimiento para hacer frente a los gigantes de la industria del software.



- Linux puede ser utilizado como una estación personal pero también como un potente servidor de red.
- Linux incorpora una gama de sistemas de interfaz gráfica (ventanas) de igual o mejor calidad que otras ofrecidas en muchos paquetes comerciales.
- Posee el apoyo de miles de programadores a nivel mundial.
- El paquete incluye el código fuente, lo que permite modificarlo de acuerdo a las necesidades del usuario.
- Utiliza varios formatos de archivo que son compatibles con casi todos los sistemas operacionales utilizados en la actualidad.

Desventajas de Linux

- Linux no cuenta con una empresa que lo respalde, por lo que no existe un verdadero soporte como el de otros sistemas operativos.
- La pendiente de aprendizaje es lenta.
- No es tan fácil de usar como otros sistemas operativos, aunque actualmente algunas distribuciones están mejorando su facilidad de uso, gracias al entorno de ventanas, sus escritorios y las aplicaciones diseñadas específicamente para él, cada día resulta más sencillo su integración y uso.
- Documentación y terminología muy técnica.
- Para usuarios corrientes, todavía no es un sistema de escritorio.
- Funciona únicamente con proveedores de hardware que accedieron a la licencia GPL y en algunas instancias no es compatible con variedad de modelos y marcas.
- Requiere consulta, lectura e investigación en lista, foros o en bibliografía dedicada al tema.
- La configuración de dispositivos de entrada y salida no es trivial.
- Muy sensible al hardware.
- Muchas distribuciones e idiomas.

- Hay que leer y entender código.

2.2.8 Distribución de GNU/Linux – Ubuntu.

Es un sistema operativo predominantemente enfocado en la facilidad de uso e instalación, la libertad de los usuarios, y los lanzamientos regulares (cada 6 meses).

El nombre procedente de un concepto africano Ubuntu, su significado es "humanidad hacia otros" o "yo soy porque nosotros somos". De este nombre también es poseedor un movimiento humanista sudafricano. Ubuntu aspira a impregnar de esa mentalidad al mundo de las computadoras. El hace de Linux un sistema operativo más accesible y fácil de usar.

Ubuntu es una distribución patrocinada por Canonical Ltda., empresa cuyo dueño es el sudafricano Mark Shuttleworth.

El 8 de julio de 2004, Mark Shuttleworth y la empresa Canonical Ltda. La implementación y creación para distribución de Ubuntu, inicialmente tuvo una financiación de 10 millones de dólares (US\$).

Su creador Mark Shuttleworth es informático, empresario y primer cosmonauta sudafricano y el segundo cosmonauta "turista" en la historia, después del vuelo del millonario estadounidense Dennis Tito.

Proyecto Ubuntu. El proyecto Ubuntu nace por iniciativa de algunos programadores integrantes de proyectos Debian, Gnome y Arch que se encontraban decepcionados con la manera de operar del proyecto Debian, la distribución Linux sin ánimo de lucro más popular del mundo y tras formar un grupo multidisciplinario, los programadores decidieron buscar el apoyo económico, este apoyo lo encontraron en Mark Shuttleworth quien disponía de la solvencia económica Y confianza e interés en el proyecto como para llevarlo a cabo. Es así como nace la empresa Canonical la cual se encarga del sostenimiento económico del proyecto bajo la comercialización de servicios y soporte técnico a otras empresas.

2.2.9 Software Libre.

Es una cuestión de libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y modificar el software. Más concretamente se refiere a los cuatro tipos de libertades para los usuarios de software:

- La libertad de usar el programa, con cualquier propósito (libertad 0).
- La libertad de estudiar el funcionamiento del programa, y adaptarlo a las necesidades (libertad 1). El acceso al código fuente es un prerrequisito para esto.
- La libertad de distribuir copias, con lo que puede ayudar a otros (libertad 2).
- La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (libertad 3). De igual forma que la libertad 1 el acceso al código fuente es un prerrequisito.

Luego de adquirir un software GNU Linux se obtienen libertades específicas para usarlo. Pero, ¿qué se entiende por Software Libre? El software libre es, ante todo, una forma ética de entender el software o los programas de un computador que incluye tanto la fabricación de los programas como su distribución y su utilización. Por lo tanto, el software libre no está orientado únicamente a la gente capaz de escribir un programa, o a la gente que posee empresas y por tanto está en condiciones de hacer negocio con los programas. Por el contrario, el software libre tiene que ver con el usuario tanto como con los desarrolladores o los comerciantes o tal vez incluso más, ya que es la libertad de los usuarios el único objetivo del movimiento del software libre.

2.2.10 Html.

HyperText Markup Language, es un lenguaje de marcado para hipertexto el cual permite la creación de páginas web por medio de tags o etiquetas para la definir la estructura del contenido y el entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen, se deben guardar con la extensión .htm o .html



Estos documentos pueden ser mostrados por los visores o "browsers" de páginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer entre otros. Html es una forma de definir, dar formato y estilo a una página.

2.2.11 Css.

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

2.2.12 Web Services.

Web services "Un Web Service es un componente de software que se comunica con otras aplicaciones codificando los mensajes y enviando estos mensaje a través de protocolos estándares de Internet tales como el Hypertext Transfer Protocol (HTTP). Intuitivamente un Web Service es similar a un sitio web que no cuenta con un interfaz de usuario y que da servicio a las aplicaciones en vez de a las personas. Un Web Service, en vez de obtener solicitudes desde el navegador y retornar páginas web como respuesta, lo que hace es recibir solicitudes a través de un mensaje formateado desde una aplicación, realiza una tarea y devuelve un mensaje de respuesta.

3. CONSIDERACIONES DE DISEÑO

3.1 CONCEPTOS BASICOS

En el diseño para el aplicativo debe acoplarse y cumplir con las expectativas previstas en comienzo, se recolectara la información necesaria para el modelado de monitoreo de (CPU, HDD y Memoria RAM), junto a los datos que se desean manipular, para así mismo diseñar una base de datos que se adate a los requerimientos del modelo obtenido por el levantamiento de la información. Para posteriormente diseñar una interfaz de usuario que se acople a todos los módulos que conformarán la aplicación en el entorno Web y Mobile, esto permitirá que la experiencia de usuario sea óptima desde el punto de vista gráfico; se generará un sistema automático tipo Pool para obtener la información, de los distintos nodos a monitorear.

3.2 REQUERIMIENTOS TÉCNICOS

3.2.1 Aplicación web.

La interfaz web debe ser totalmente accesible desde internet, su entorno debe ser dinámico y de fácil manejo.

La interfaz de usuario de contener cada uno de los módulos preestablecidos como lo son:

- Módulo de usuarios, este permitirá almacenar, listar, actualizar y eliminar cada uno de los usuarios que llegaran a manejarse en la aplicación, compuesto de datos como: nombre de usuario, nickname, password y tipo de usuario o perfil.
- Módulo de servidores o direcciones IP, este permitirá almacenar, listar, actualizar y eliminar los servidores, compuesto de datos como: dirección IP, comunidad de la configuración SNMP y el estatus de monitoreo, el cual permitirá manejar la disponibilidad del servidor en la aplicación.
- Módulo de tipos de monitoreo, este permitirá almacenar, listar, actualizar y eliminar los tipos de monitoreo que podrán ser asignados a un servidor, compuesto de datos como: el nombre del monitoreo y campos de selección para cada uno de los objetos a monitorear, memoria "cache, buffered,



shared, free, used, machine, Available Swap”, disco duro HDD “used, total y free”, cpu “idle, system y user”.

- Módulo de asignación de monitoreo a los servidores o direcciones IP, este permitirá almacenar, listar, actualizar y eliminar la asignación de un monitoreo a un servidor.
- Módulo para la asignar, listar, actualizar y eliminar a los usuarios, los diferentes monitoreos pertenecientes a un servidor.
- Módulo de monitoreo, este deberá generar graficas de cada uno de los servidores filtradas por la diferentes opciones del tipo de monitoreo y debe permitir hacer dos tipos de consulta: histórica (este grafica se generara a partir de los registros de la ultima hora almacenados en la base de datos) y en tiempo real (esta grafica se generara desde una consulta por SNMP directa hacia el servidor).
- Módulo de verificación de disponibilidad o estatus de los servidores en tiempo real, este deberá mostrar en una tabla cada uno de los datos básicos de cada objeto a monitorear de un servidor, como lo son: memoria (total, usada y libre), disco duro HDD (mostrara en cada una de las particiones el total, usado y libre) y CPU (idle, user y system).
- Módulo de reportes, este deberá generar reportes modificables en Excel e informativos en pdf, filtrados por una fecha mínima y máxima, tipo de reporte grupal o unitario en servidores, un tiempo de pool discriminado en 6 tiempos de 10 minutos que componen una hora, y podrá ser discriminado por las opciones de cada uno de los objetos a monitorear, descritas anteriormente.

3.2.2 Aplicación Mobile.

Entorno Mobile, Esta deberá contar con una validación de usuario y contraseña, si el usuario no es validado por la plataforma, esta no deberá permitir el acceso al uso de la herramienta. Pero si el usuario es validado correctamente, este deberá redirigirlo a una pantalla donde se listaran los diferentes servidores que el usuario tiene asignados y el tipo de vista, con las siguientes opciones: cpu, memoria y disco duro. Al hacer click en un botón generar, deberá redirigirlo a una pantalla donde se debe visualizar una gráfica de comportamiento del servidor y una tabla con los datos de la última hora según la vista seleccionada, el usuario debe tener la opción de recargar el gráfico y la tabla de datos. Desde las pantallas de



gráficos el usuario no debe tener la opción de salir de la aplicación hasta que vuelva a la pantalla donde se ejecuta la generación las gráficas.

3.2.3 Requerimientos Generales.

Para el desarrollo de los aplicativos se emplearan herramientas como:

- Symfony un framework PHP una herramienta de código abierto cuyo objetivo principal es acelerar la creación y el mantenimiento de los aplicativos.
- Java un lenguaje de programación orientado a objetos, el más utilizado para el desarrollo del aplicativos web y que cuenta con las herramientas necesarias para este proyecto.
- SDK JAVA Mobile BlackBerry la herramienta que nos facilitara la configuración e implementación de las diferentes funcionalidades que poseen los últimos modelos de hardware.
- Web (HTML, CSS, JAVASCRIPT) la integración de estas herramientas permite cumplir con las expectativas definidas para el entorno Web.
- WebService XML cuenta con un conjunto de protocolos que permitirán el intercambio de información en la red.

3.2.4 Moldeamiento de monitoreo.

El diseño del aplicativo debe acoplarse a la información que se desea manipular en cada uno de los dispositivos preestablecidos en el documento base, (CPU, HDD y Memoria RAM), se debe definir un formato para representación de la información contenida por cada uno de los servicios.

El proyecto inicialmente se centrara en el monitoreo de sistemas que operen bajo licencia Gnu-Linux, y sus servicios.



3.2.5 Conexión con la base de datos Mysql.

El aplicativo debe contar con acceso permanente a la información contenida por la base de datos, debe cumplir con estándares de seguridad que garanticen un buen manejo y manipulación de la información obtenida

Adicional mente la el servidor de base de datos debe actualizar constantemente contacto con los servidores de monitoreo y actualizar la información periódicamente. Se realizara un proceso automático de pool, esto con el fin de obtener la información necesaria de cada uno de los servidores ingresados en el aplicativo Web.

3.3 REQUERIMIENTOS FUNCIONALES

- El software debe acoplarse y limitarse a los parámetros definidos en el documento base. Esto con el objetivo de cumplir a cabalidad con los objetivos de Nova Realtime.
- El diseño de Nova Realtime debe ser total mente dinámico, entendible y de fácil manejo para el personal de IT, contara con los módulos necesarios, acordes al perfil del usuario registrado.
- Se debe recolectar la información necesaria para el modelo de monitoreo de (CPU, HDD y Memoria RAM), junto a los datos que se desean manipular, para así mismo diseñar una base de datos que se acople a los requerimientos del modelo obtenido por el levantamiento de la información.
- El aplicativo Web y Mobile que permita manejar: usuarios, servidores, tipos de monitoreo, asignación de servidores a usuarios de monitoreo, asignación de tipos de monitoreo a servidores, reportes y graficas de comportamiento histórico o tiempo real de cada máquina, a su vez desde la interfaz Mobile monitorear en tiempo real el estado de la máquina que se desee.
- El aplicativo estará conformado por un módulo de pool el cual tomara las direcciones IP habilitadas para monitoreo de cada uno de los Servidores GNU-Linux, almacenara información obtenida en tablas de la misma base de datos.
- Nova Realtime debe garantizar el acceso desde cualquier locación que cuente con acceso a internet, esto permitirá al usuario estar siempre en

línea con el servidor de monitoreo y supervisar sus servidores y servicios desde cualquier ubicación, ya sea por medio de la web o Mobile.

- El aplicativo debe contar con un manual de usuario donde se describa de forma detallada las opciones y funciones de cada módulo.

3.4 REQUERIMIENTOS NO FUNCIONALES

Nombre	Atributos	Unidad operacional	Metas propuestas	Resultados
Económico	Mano de obra requerida para manejo del aplicativo	Tiempo estimado para capacitación	2 horas de capacitación al personal administrativo	Capacitación mínima estimada 1 hora retroalimentación para otros usuarios.
	Costeo para desarrollo e implementación	Gastos de alquiler para equipo de desarrollo	Los gastos serán asumidos por los desarrolladores, se busca que el desarrollo se realice en los tiempos dispuestos, esto con el fin evitar costos mayores innecesarios.	Se culmina el desarrollo total del aplicativo con el presupuesto disponible.
	Desarrollo Web	IDE Eclipse – PHP	Implementación de herramientas de licenciamiento libre, que no generen costos adicionales.	Interfaz Web accesible para el usuario final.

	Desarrollo Mobile	IDE Eclipse – BlackBerry	Implementación de herramientas de licenciamiento libre, que no generen costos adicionales.	Interfaz Mobile accesible para el usuario final. Adaptable a las últimas tecnologías de tipo Mobile.
	Simulación de monitoreo	Servidores de Pruebas Vitalizados en Virtualbox	Implementación de herramienta que nos permitirá hacer el monte de varios servidores, sin generar costos.	Montaje de servidores dispuestos para demostración.
Eficiencia	Tiempo de respuesta a las peticiones del servidor Nova	Tempo estimado para respuesta de las peticiones	Tiempo estimado de 5 a 10 segundos, según proceso en ejecución.	Reducción tiempos de respuestas a 3 segundos.
	Tiempo de respuesta en la generación de reportes	Tempo estimado para respuesta para generación de reportes y gráficos.	Tiempo estimado 10 segundos, según grafica o reporte y dependiendo del medio solicitante.	Reducción tiempo respuestas a 5 segundos.
	Interacción de interfaz gráfica y cliente	Operatividad de usuario con el aplicativo.	Claridad de procesos, facilidad de manejo y comprensión de reportes y gráficos.	Satisfacción de usuario final con la interfaz y representación de la información.
	Margen de error	Tiempo real de respuesta y exactitud de la información.	Manejar margen mínimo de error con la información obtenida a nivel automático, y la manual.	Resultado de probabilidad de error 0% entre datos procesados y tomas manuales

Escalabilidad	Parámetros para monitoreo de servidores Gnu-Linux	Implementación de monitoreo de servidores Gnu-Linux y sus servicios	Soporte y aplicación solo para servidores bajo licencia Linux.	monitoreo limitado a brindar soporte en servidores Gnu-Linux y sus servicios
Seguridad	Roles de usuario y perfiles definidos	Prueba satisfactorias de creación de usuarios, asignación de servidores y tipos de monitoreo	Definición de dos tipos de usuarios Administrador y Consultar, el acceso a los módulos y servidores disponible se definirán por un usuario administrador.	Asignación de usuario acorde con las necesidades del administrador.
	Acceso seguro al aplicativo	Pruebas de acceso seguro en aplicativos.	Módulo de registro de usuario, ingreso y cierre de sección correctos.	Autenticación y salida del servidor de forma adecuada.

Tabla 4. Requerimientos no funcionales.

3.5 ALCANCE DEL SISTEMA

El aplicativo contara un acceso de tipo Web y otro de medio Mobile, tendrá disponibilidad desde cualquier punto que cuente con una conexión con internet.

El ingreso para el aplicativo contara con un módulo de autenticación de usuario, esto permitira garantizar que la información solo estará disponible para personal registrado.

Según perfil asignado por el administrador de sistemas este podrá hacer uso del CRUD (create, read, update y delete) en la aplicación, generación de gráficas y reportes y el usuario solo podrá generar graficas de comportamiento y reportes.

Construcción de gráficas y reporte se adecuara dependerá del tipo de reporte a generar. Para los reportes y gráficos de Realtime el aplicativo tendrá continuo contacto con los servidores y realizara una actualización constante de la información. Para los de tipo histórico el usuario podrá definir la consulta por lazos de tiempo, servidores que se quiere mostrar en el reporte y los tipos de monitoreo de cada servidor registrado.



El acceso de tipo Mobile es el elemento más importante con que cuenta Nova Realtime, pues en los sistemas actuales que ofrecen este tipo de servicios no cuentan con la integración de esta tecnología en sus paquetes de servicio.

El aplicativo inicialmente enfocara el modelo de monitoreo de (CPU, HDD y Memoria RAM), junto a los datos que se desean manipular, esto con el fin de cumplir con los objetivos plateados para el desarrollo.

4. DISEÑO DEL SISTEMA

4.1 DESCRIPCIÓN GENERAL

El software que cuenta con un aplicativo web y un de tipo Mobile, permitiendo al usuario acceder desde cualquier punto con internet.

El software contara con por un módulo de pool el cual tomara las direcciones IP habilitadas para monitoreo de cada uno de los Servidores GNU-Linux.

Se generaran reportes con la información obtenida modelado de (CPU, HDD y Memoria RAM), los reportes serán de dos tipos modificables y no modificables, según requerimientos del administrador.

El almacenamiento de información para la base de datos, se establecerá en un tiempo de pool discriminado en 6 tiempos de 10 minutos que componen una hora, y podrá ser discriminado por las opciones de cada uno de los objetos a monitorear.

4.2 ARQUITECTURA DEL SISTEMA

4.2.1 Requerimientos de Hardware

- Procesador: Intel Pentium Processor T4500 (1M Cache, 2.30 GHz, 800 MHz FSB)
- Memoria RAM: 2 Gigabytes
- Disco Duro: 250 Gigabytes
- Conectividad: router linksys wrt54g2

4.2.2 Requerimientos de Software

- Oracle VM VirtualBox, Versión 4.1.14
- Ubuntu Server, Versión 12.04 LTS X86
- Apache 2
- Php 5.3
- Mysql, Versión 5.5.22
- Open SSH 5.9p1
- Symfony 1.4.17



- Mysql Workbench 5.2 CE
- Balsamiq Mockups, Versión 1.6.10
- Eclipse SDK, Versión 3.7.2 (PDT 1.3.2) y Versión 3.7.2 (PDT 1.3.2) BlackBerry Java SDK 5.0.0.25
- Mozilla Firefox 12.0
- Firebug, Versión 1.9.2
- Putty, Versión 0.62
- BlackBerry Smartphone Simulator 6.0.0
- Adobe Fireworks CS5, Versión 11.0.0.484
- Windows 7 Enterprise
- Enterprise Architec, Versión 7.5.845
- Notepad++, Versión 6.1

4.2.3 Guía de instalación

Para la instalación de Nova Realtime, se debe tener como pre-requisito instalado lo siguiente:

- Instalado la versión de Linux Gnu destinada a ser el server de Nova Realtime
- mysql-server
- mysql-client
- apache2
- php 5.3
- librerías: libapache2-mod-php5 , php5-mysql, php5-snmp y php-pear
- habilitar el puertos: 80, 161 y 162
- symfony-1.4.17

Instalación de Nova Realtime:

- a) Crear base de datos para Nova Realtime:

```
mysql> create database nova_realtime;
```

- b) Generar usuario mysql para la base de datos:

```
mysql> GRANT ALL PRIVILEGES ON nova_realtime.* TO 'user_aplication'@'localhost' IDENTIFIED BY '123456' WITH GRANT OPTION;
```



- c) Copiar el archivo `install_db.sql` al servidor e instalar las tablas en la base de datos usando el usuario que se creó:

```
# mysql -u user_aplication -p nova_realtime < install_db.sql
```

- d) Agregar todo el folder de la aplicación Nova Realtime “`www.nova.co`” en el folder `/var/www`

- e) Habiendo agregado el folder se procede a agregar el usuario de la base de datos a la aplicación, para ello hay que editar el archivo `databases.yml` que se encuentra en el folder `config` de la carpeta raíz de la aplicación y su contenido debe quedar de la siguiente forma:

```
all:
  doctrine:
    class: sfDoctrineDatabase
    param:
      dsn:  mysql:host=localhost;dbname=nova
      username: user_aplication
      password: 123456
```

- f) Crear y configurar el virtualhost en apache, esto se hace con la ayuda de algún editor de texto.

```
vim /etc/apache2/sites-available/www.nova.co
```

Y el contenido del archivo debe ser el siguiente:

```
<VirtualHost *:80>
    ServerName www.nova.co
    DirectoryIndex index.php
    DocumentRoot /var/www/www.nova.co/web
    Alias /sf /usr/share/php/data/symfony/web/sf
    <Directory "/usr/share/php/data/symfony/web/sf">
```



```
AllowOverride All
```

```
Allow from All
```

```
</Directory>
```

```
<Directory "/var/www/www.nova.co/web">
```

```
AllowOverride All
```

```
Allow from All
```

```
</Directory>
```

```
</VirtualHost>
```

Ahora guardar el contenido del archivo y se ejecuta el comando “**a2ensite www.nova.co**” crea un enlace en **sites-enable** su función es activar el virtualhost que se acabó de crear, se ejecuta también el comando “**a2enmod rewrite**” el cual permite habilitar los módulos del apache que se encuentren en “/etc/apache2/mods-available/” y “/etc/apache2/mods-enabled/”.

g) Agregar el dominio al archivo hosts

Abrir el archivo hosts con un editor de texto

```
vim /etc/hosts
```

y agregar la siguiente línea

```
127.0.0.1    www.nova.co
```

Guardar y reiniciar el servicio con el comando “**service apache2 restart**”

El ejemplo de instalación anterior fue ejecutado en la distribución **Linux Ubuntu Server basada en Debian GNU/Linux**.

4.3 MODELADO DE CASOS DE USO

Requerimientos del sistema		
Modulo	Descripción	Prioridad
Login	Autenticación e ingreso (usuario y password)	Alta
	cierre de sesión	
User	Creación de un nuevo usuario que se compone de (nombre del usuario, nickname, password y el tipo de usuario)	
	Formulario búsqueda o filtro de usuarios existentes.	
	Herramienta de edición o eliminación.	
Ip Address	Creación de un nuevo servidor o dirección que se compone de (dirección ip de la máquina, comunidad SNMP y el estado de monitoreo).	Alta
	Formulario de búsqueda o filtro de direcciones Ip listadas.	
	Herramienta de edición o eliminación.	
Monitor Type	Creación de un nuevo tipo de monitoreo, que se compone de (nombre para el monitoreo, elementos de memoria, elementos del disco duro y elementos de Cpu)	Alta
	Formulario de búsqueda o filtro de tipos de monitoreo	
	Herramienta de edición o eliminación.	
Monitor user	Asignación de monitoreo de un servidor o dirección ip a usuarios, que se compone de (usuario, servidor o dirección ip y tipo de monitoreo)	Alta
	Formulario de búsqueda o filtro de las asignaciones de tipo de monitoreo de los servidores o direcciones ip a usuarios.	
	Herramienta de edición o eliminación.	

Monitoring	Selección del tipo de monitoreo de un servidor o dirección ip y el tipo de grafica a seleccionar	Alta
	Selecciona el tipo de monitoreo history las gráficas serán generadas a partir de la base de datos	
	se selecciona monitoreo realtime se realizara una consulta directa hacia el servidor que se seleccionó y se graficaran estos datos	
	Generación de grafica de la memoria de la máquina	
	Generación de gráfica del disco duro de la maquina	
	Generación de grafica de la cpu	
Server status	Listado de direcciones ip de los servidores que en su opción de estatus de monitoreo se encuentra habilitado.	Normal
	Listado de las particiones con actualización automática de cada servidor con el total y el usado de la partición	
	Link para la generación de gráficos en tiempo real permanente de las particiones del disco duro	
	Listado de los tres estados en los que se puede encontrar la RAM "Total, en uso y libre" con actualización automática	
	Para la generación de grafico en tiempo real permanente de los estados de la memoria RAM	
	Listado de los tres estado en los que se puede encontrar la CPU "Usuarios, sistema y libre"	
	Link para la generación de grafico en tiempo real permanente de los estados de la CPU.	

Tabla 5. Requerimientos del sistema para para modelado de casos de uso.



4.3.1 Descripción general de actores.

El aplicativo está dirigido a los al personal de IT u administradores de red, Nova Realtime busca ser una herramienta que agilice y facilite las tereas de monitoreo de servidores Gnu-Linux. No se requiere tener un nivel profesional de alto nivel para el manejo del a aplicativo basta con que maneje conocimientos básicos en redes e informática; por lo tanto lo pueden manejar técnicos, tecnólogos o ingenieros de sistemas.

Los usuarios podrán acceder al aplicativo desde cualquier lugar con acceso a internet, se requiere estar registrado con una cuenta de usuario y unos permisos asignados, dicha cuenta debe ser solicitada al administrador o director encargado de liderar el departamento de informática.

El aplicativo está diseñado para manejar dos tipos de roles, el administrador y el consultor, es posible restringir, habilitar y asignar al usuario los servidores que puede incluir en el monitoreo y los servicios a monitorear.

4.3.2 Diagramas de modelado de casos de uso

El modelo de casos de uso refleja al detalle cado uno de los procesos que conforman el sistema. Se realizara una descripción de cada uno de los diagramas representados.

4.3.2.1 Aplicación Web.

- **Login & Logout**

Descripción: Permite a los usuarios ingresar y salir de la aplicación.

Actores Principales: *Administrator y Consultant.*

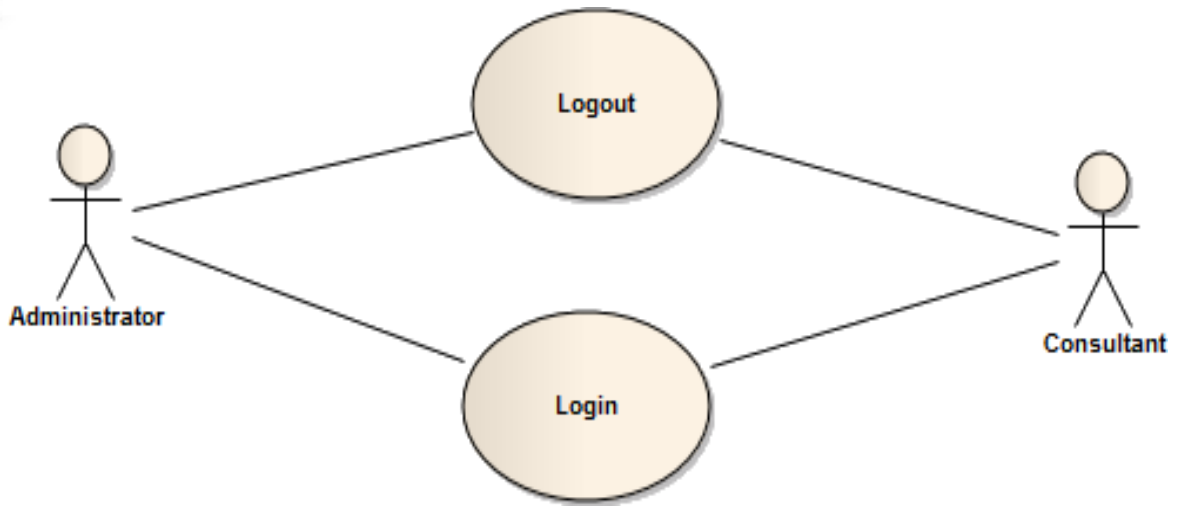


Ilustración 5. Login & Logout

- **Users**

Descripción: Permite crear, editar, borrar y listar los usuarios que estarán asociados en la aplicación.

Actor Principal: *Administrator*.

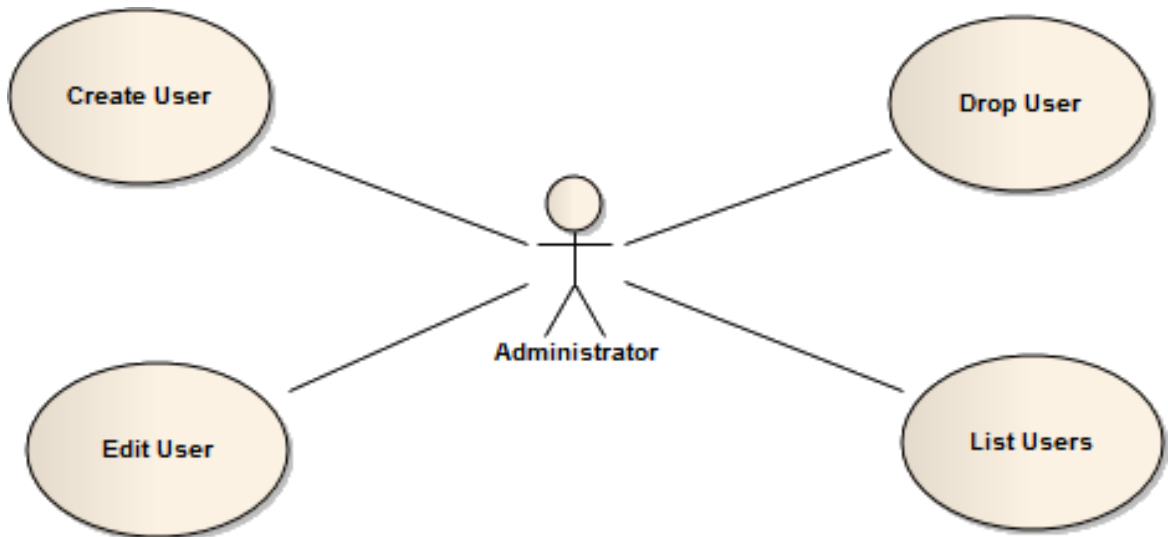


Ilustración 6. Caso de uso Users.

- **Ip Address**

Descripción: Permite crear, editar, borrar y listar los servidores que serán monitoreados desde Nova Realtime.

Actor Principal: Administrator

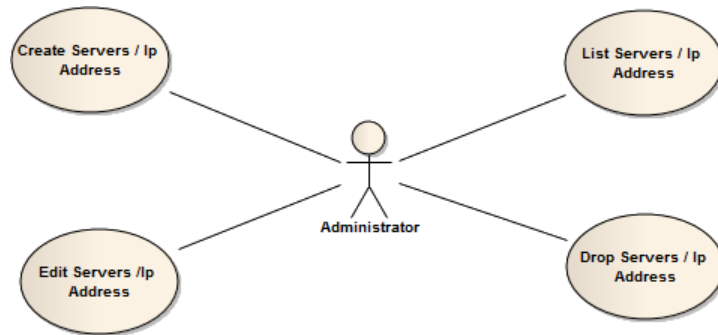


Ilustración 7. Caso de uso ip address.

- **Monitor Type**

Descripción: Permite crear, editar, borrar y listar los tipos de monitoreo que se asociaran a los servidores.

Actor Principal: *Administrator*.

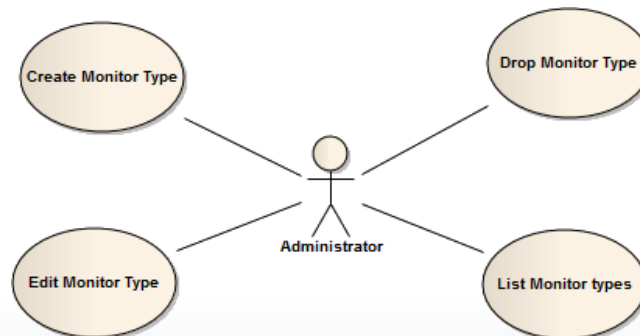


Ilustración 8. Caso de uso monitor type.

- **Monitor Ip Address**

Descripción: Permite agregar, editar, borrar y listar tipos de monitoreo asignados a los servidores o direcciones ip.

Actor Principal: Administrator.

Precondiciones:

- Tener tipos de monitoreo creados
- Tener servidores o direcciones ip creados.

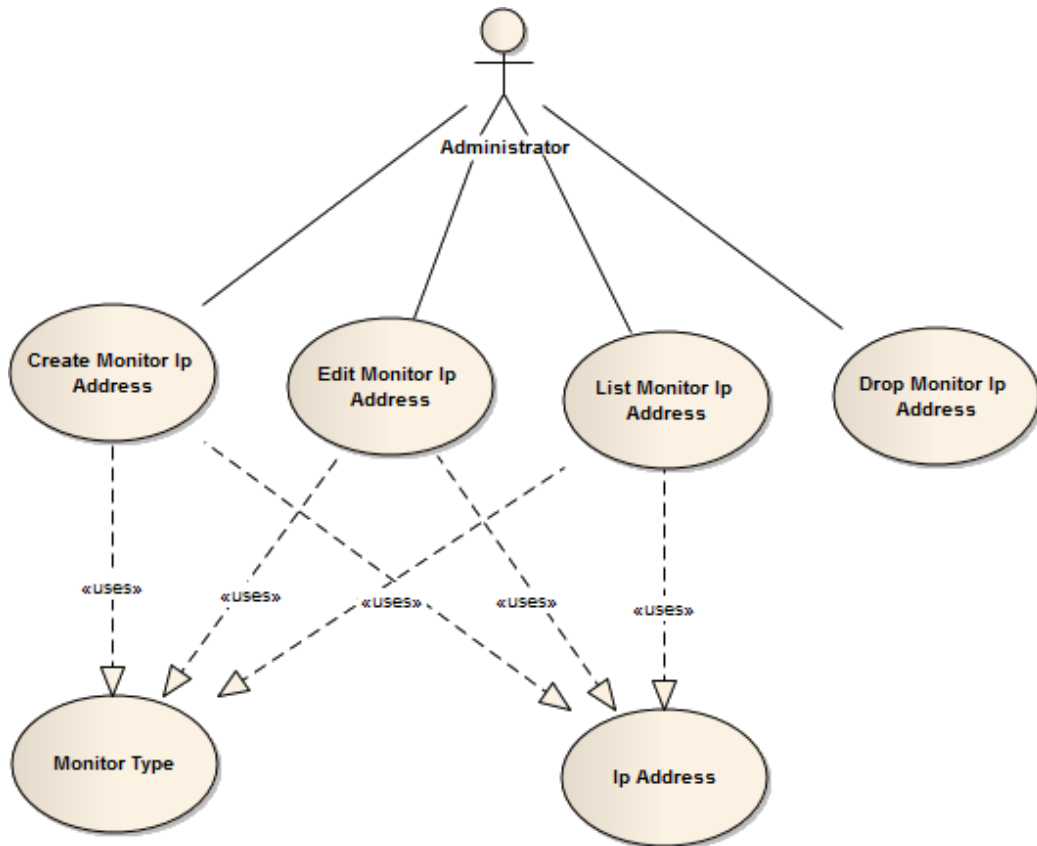


Ilustración 9. Caso de uso monitor ip address.

- **Monitor User**

Descripción: Permite crear, editar, borrar y listar los monitoreos de los servidores asignados a los usuarios.

Actor Principal: Administrator.

Precondiciones:

- Tener usuarios creados.
- Tener tipos de monitoreo adicionados a los servidores o direcciones ip.

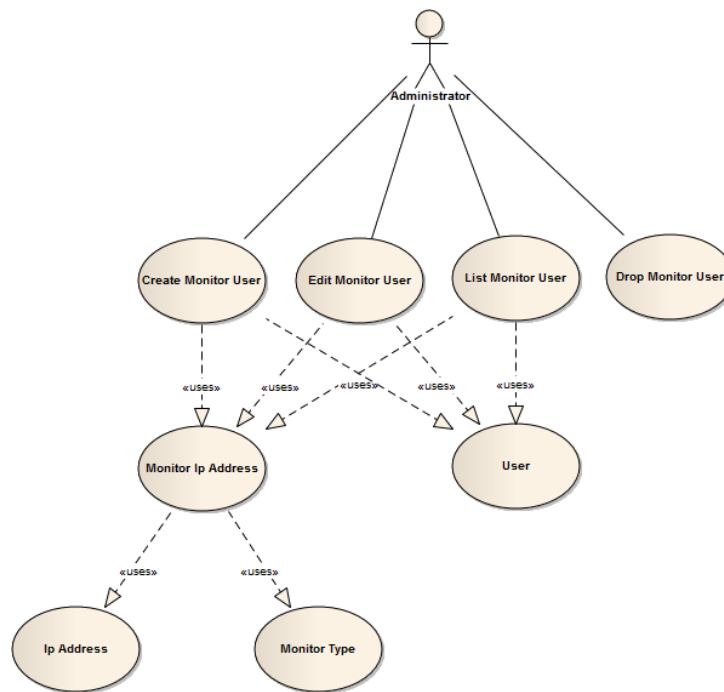


Ilustración 10. Caso de uso monitor user.

- **Monitoring**

Descripción: Permite realizar monitoreos en forma gráfica de la Memoria, Hdd y Cpu de los servidores o direcciones Ip asignados al usuario.

Actores Principales: Administrator y Consultant.

Precondiciones:

- Tener los monitoreos de servidores o direcciones ip asignados a los usuarios.

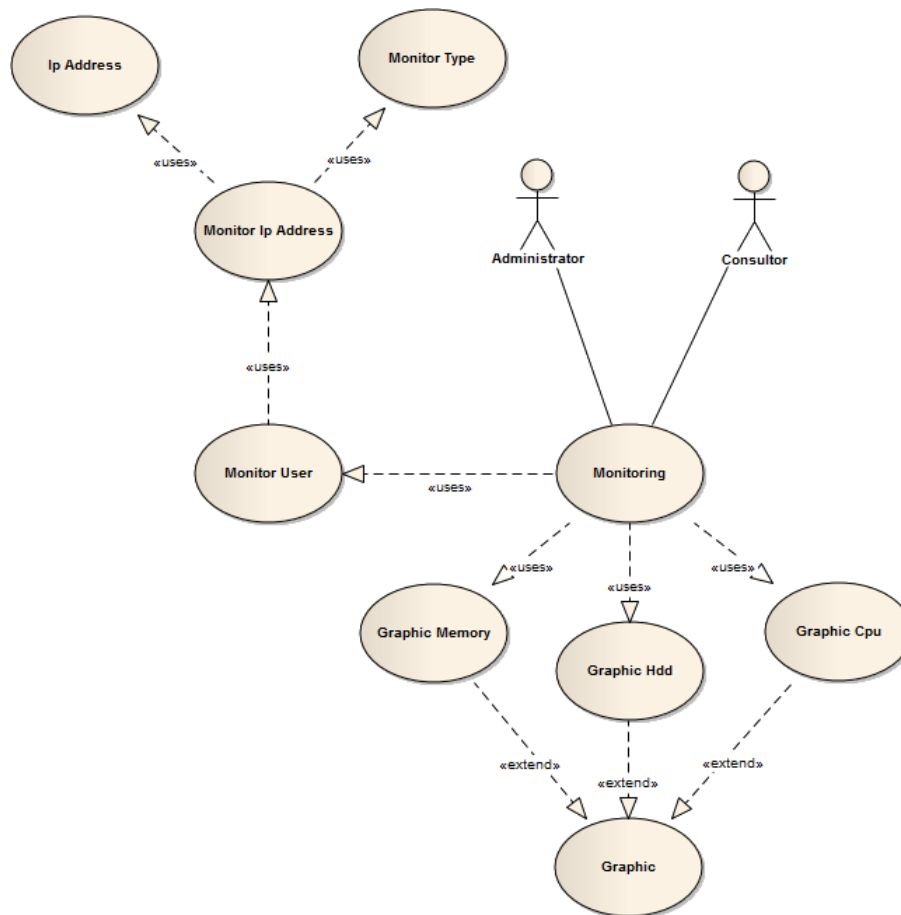


Ilustración 11. Caso de uso monitoring.

- **Server Status**

Descripción: Permite ver los datos de la memoria, Cpu y Hdd de los servidores que el usuario tenga asignados con actualización permanente.

Actores Principales: Administrator y Consultant.

Precondiciones:

- Tener los monitoreos de servidores o direcciones ip asignados a los usuarios.
- Disponibilidad del servidor para ser monitoreado.

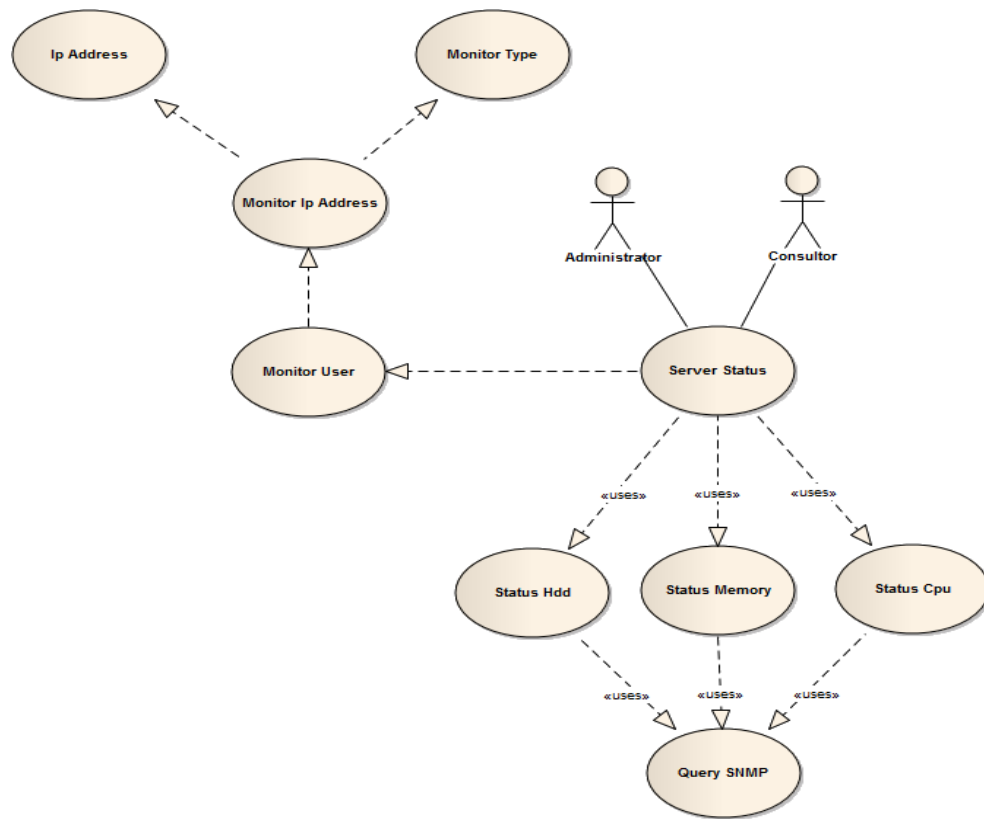


Ilustración 12. Caso de uso server status.

- **Permanent Monitoring**

Descripción: Permite al usuario hacer un monitoreo específico de la memoria, Cpu o hdd del servidor que se desee, en forma de gráfico que se actualiza constantemente.

Actores Principales: Administrator y Consultant.

Precondiciones: Disponibilidad del servidor para ser monitoreado.

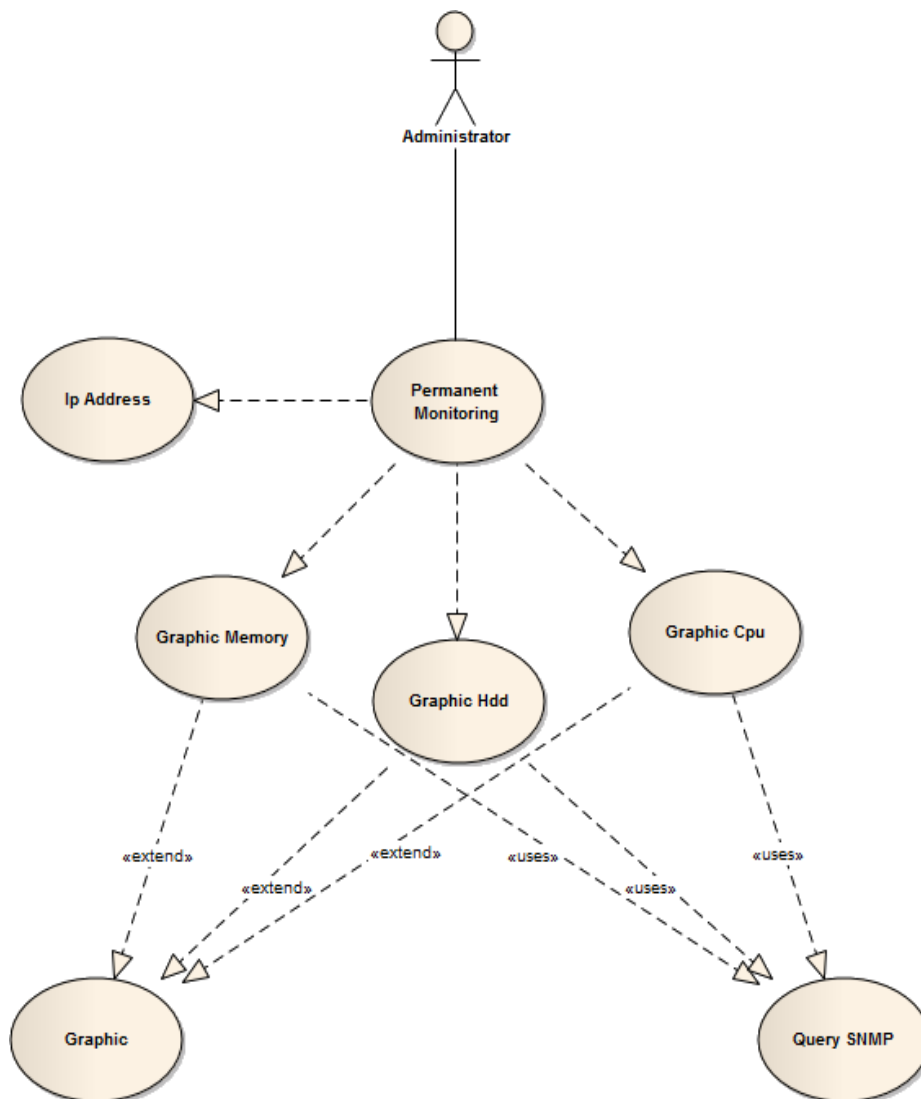


Ilustración 13. Caso de uso permanent monitoring.

- **Reports**

Descripción: Permite al usuario generar reportes en Xlsx o Pdf de la información de los servidores.

Actores Principales: Administrator y Consultant.

Precondiciones:

- Tener la información pre almacenada en una base de datos.

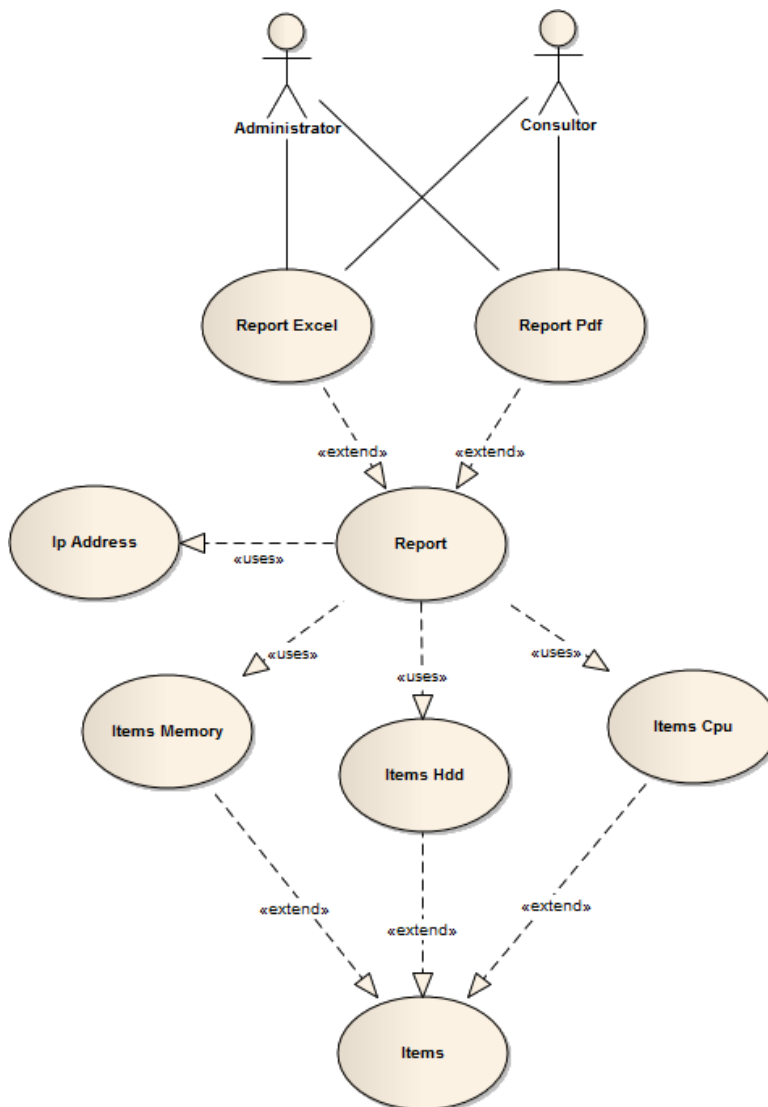


Ilustración 14. Caso de uso reportes.

4.3.2.2 Aplicación Mobile.

- **Configuration**

Descripción: Permite al usuario configurar la dirección ip del servidor de Nova Realtime.

Actor Principal: Administrator.

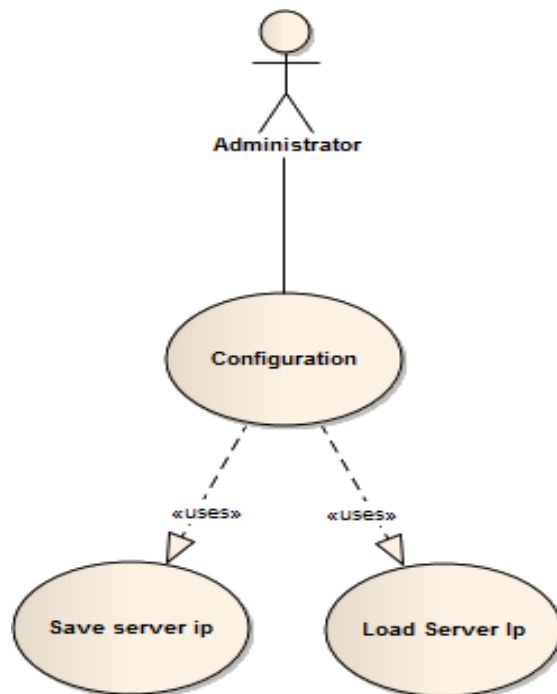


Ilustración 15. Caso de uso entorno mobile – configuration.

- **Login & Logout**

Descripción: Permite a los usuario ingresar y salir de la aplicación.

Actor Principal: Administrator.

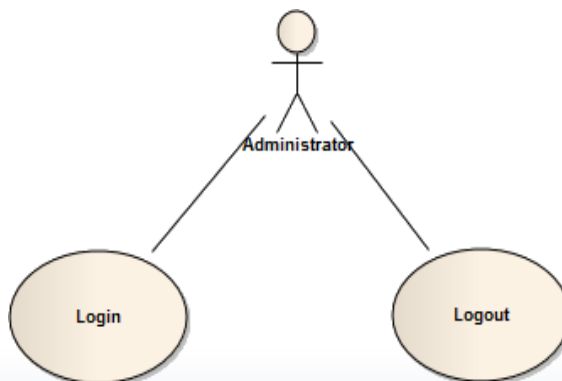


Ilustración 16. Caso de uso entorno mobile - login & logout.

- **List Servers**

Descripción: Permite al usuario listar los servidores que tenga asignados y los tipos de vistas en los que se puede generar el grafico de comportamiento.

Actor Principal: Administrator.

Precondiciones:

- Asignación de los tipos de monitoreo de un servidor o dirección ip al usuario.

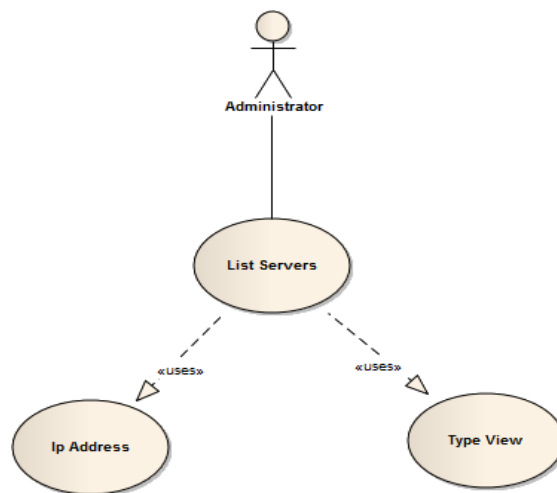


Ilustración 17. Caso de uso entorno mobile – List Servers.

- **Partitions**

Descripción: Permite listar las particiones de un servidor cuando en “List Servers” se escoge el tipo de vista Hdd.

Autor Principal: Administrator.

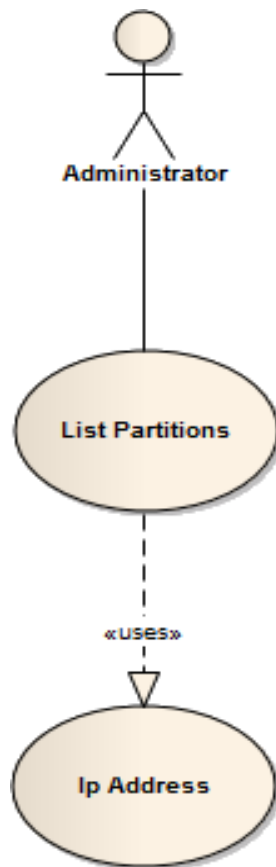


Ilustración 18. Caso de uso entorno mobile - partitions.

- **Graphic**

Descripción: permite al usuario generar los gráficos del servidor según el tipo de vista seleccionado.

Actor Principal: Administrator.

Precondiciones:

- Asignación de los tipos de monitoreo de un servidor o dirección ip al usuario.
- Haber seleccionado un tipo de vista para discriminar el tipo de grafico a generar "Memoria, Cpu y Hdd".

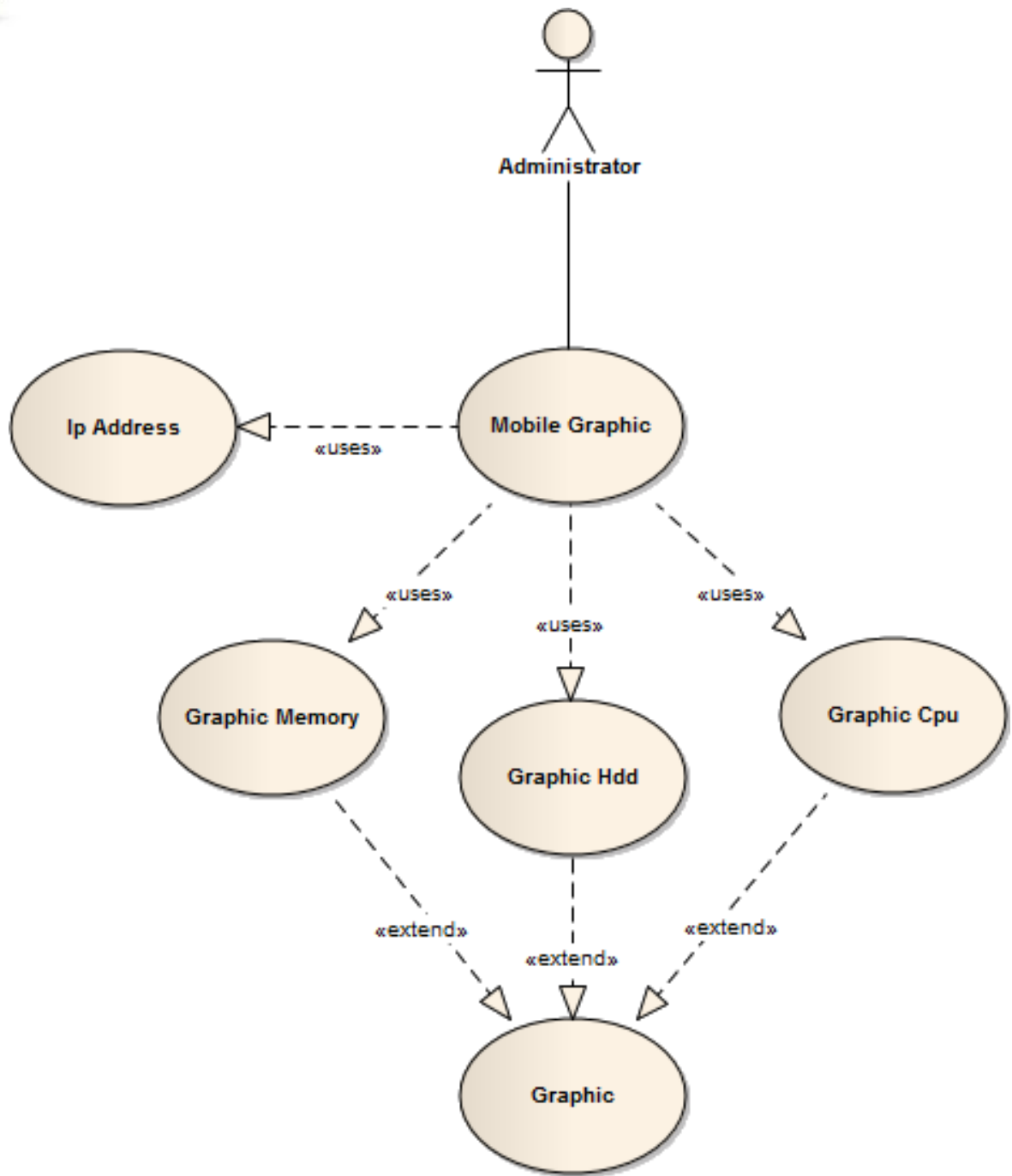


Ilustración 19. Caso de uso entorno mobile – graphic.

4.3.3 Diagramas de modelado de Secuencia.

4.3.3.1 Aplicación Web.

- Diagrama de Secuencia: Login & Logout.

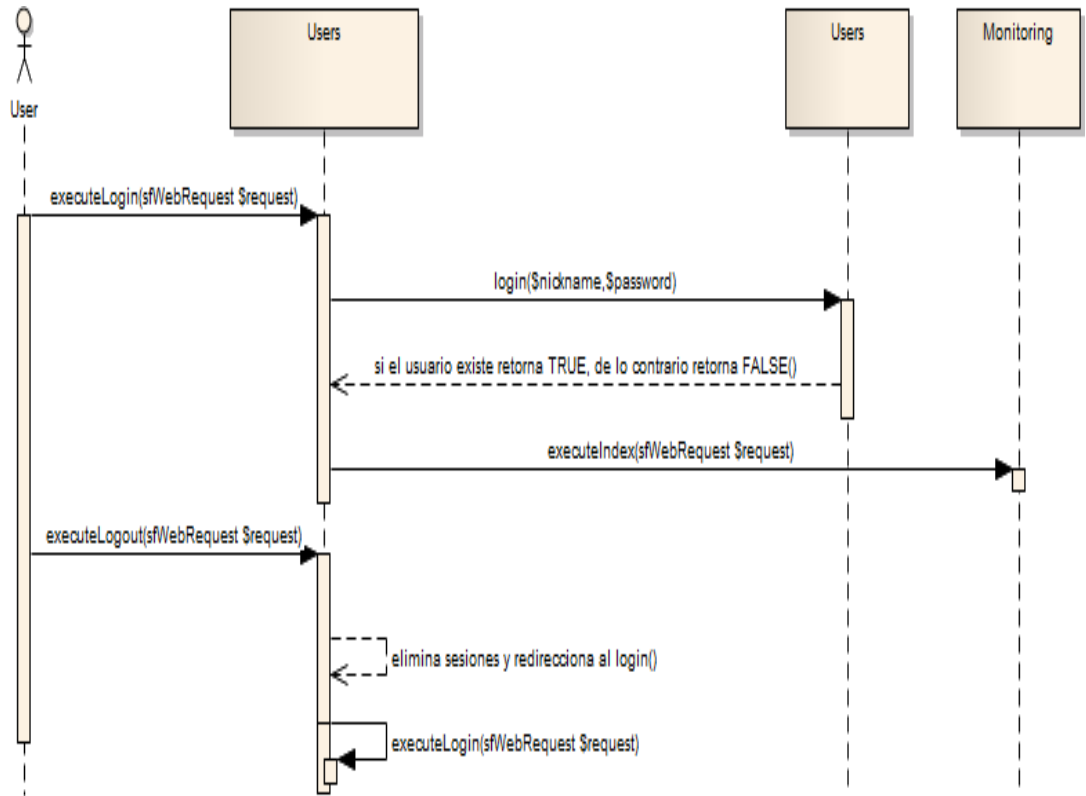


Ilustración 20. Diagrama de secuencia - login & logout.

- Diagrama de Secuencia: Users.

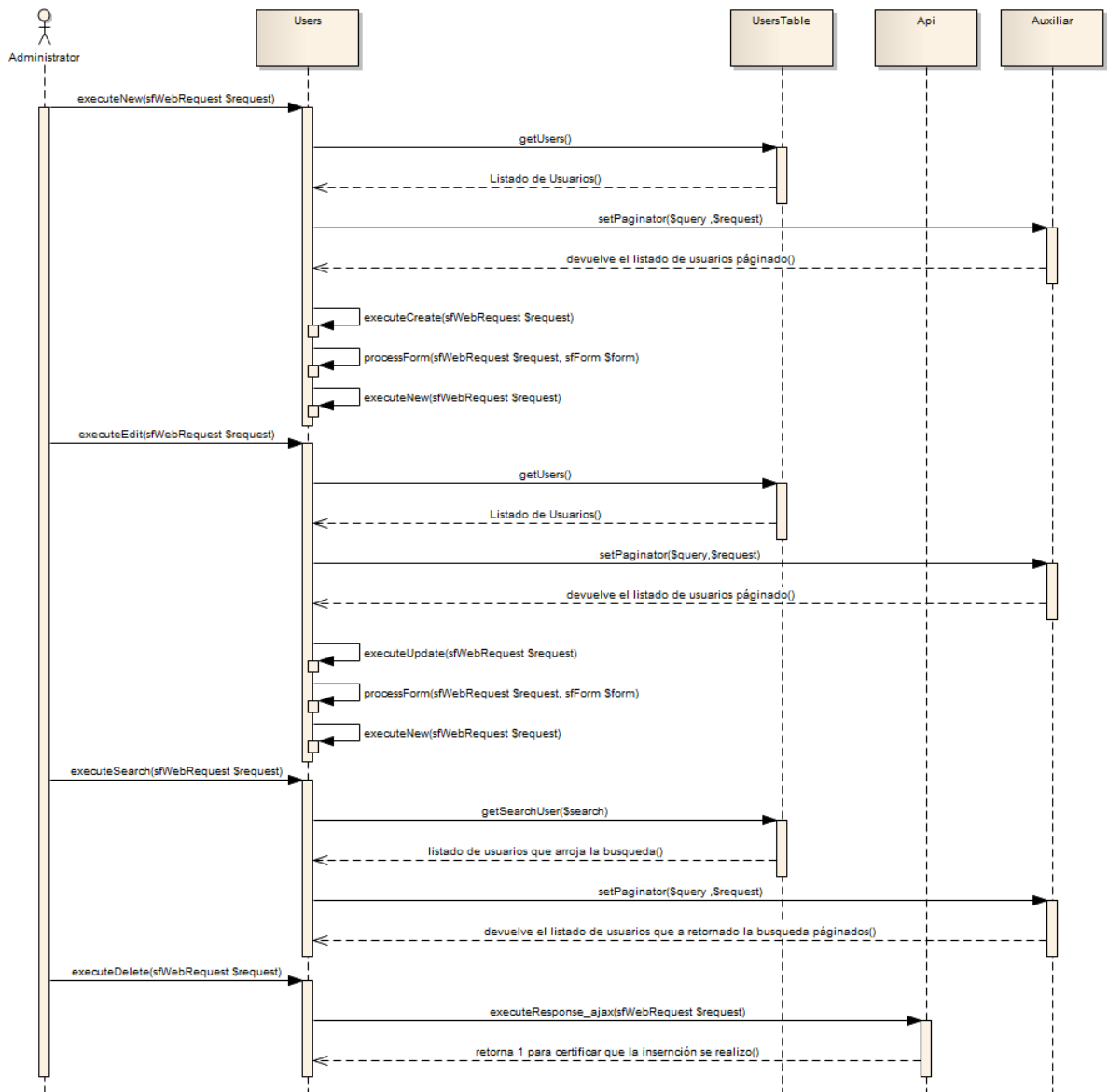


Ilustración 21. Diagrama de secuencia – users.

- Diagrama de Secuencia: Ip Address.

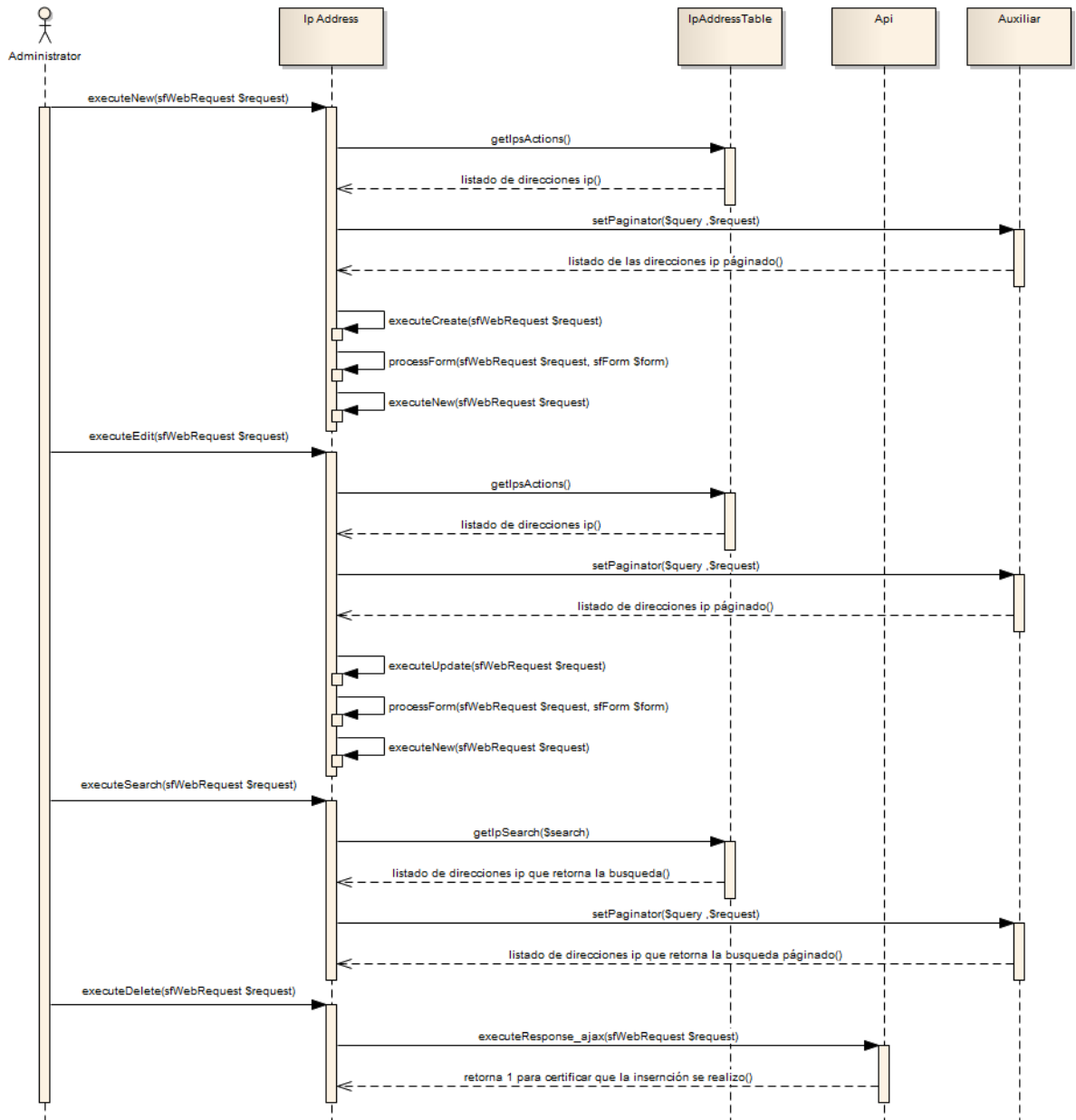


Ilustración 22. Diagrama de secuencia - ip address.

- Diagrama de Secuencia: Monitor Type.

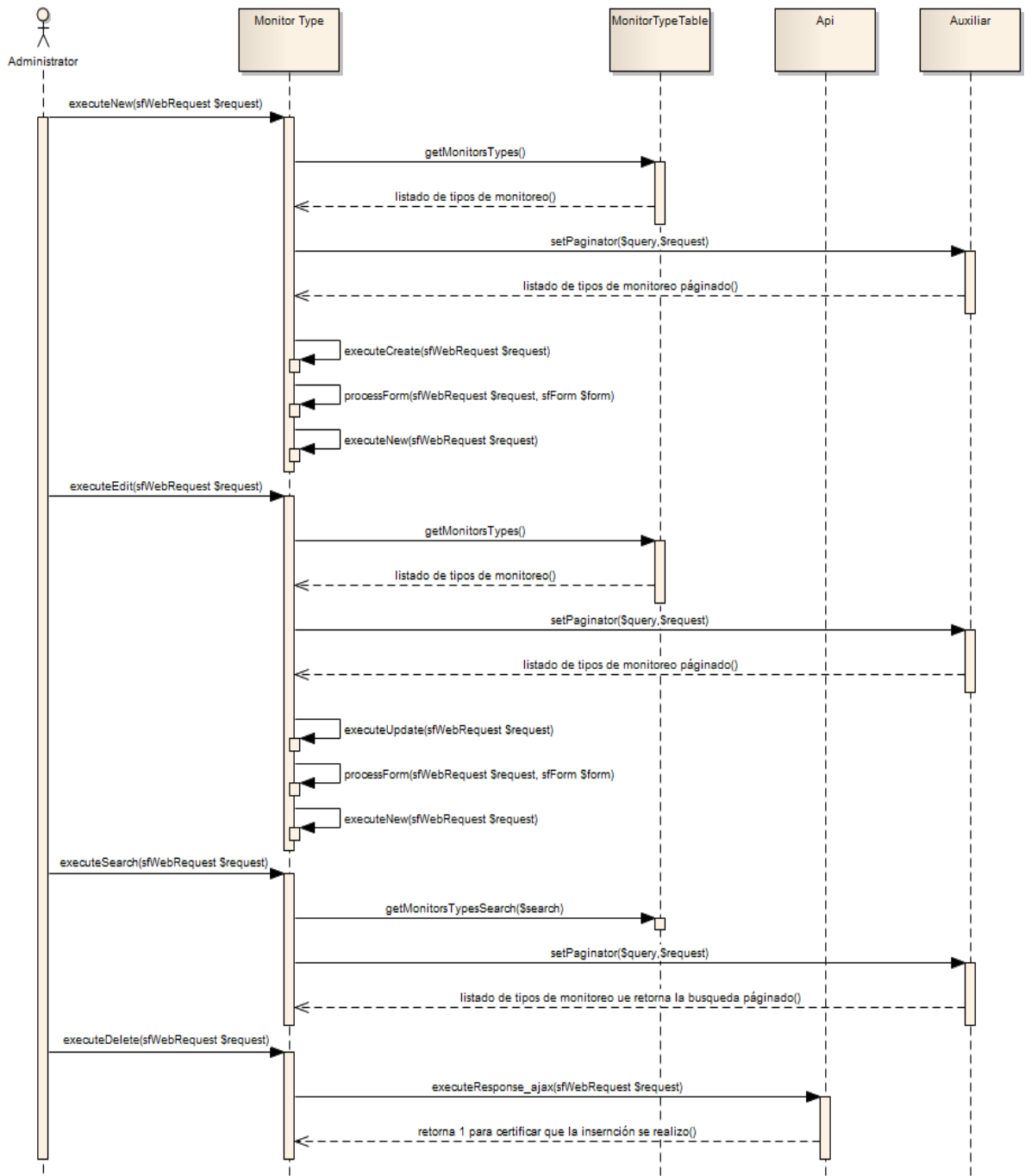


Ilustración 23. Diagrama de secuencia - monitor type.

- Diagrama de Secuencia: Monitor Ip Address.

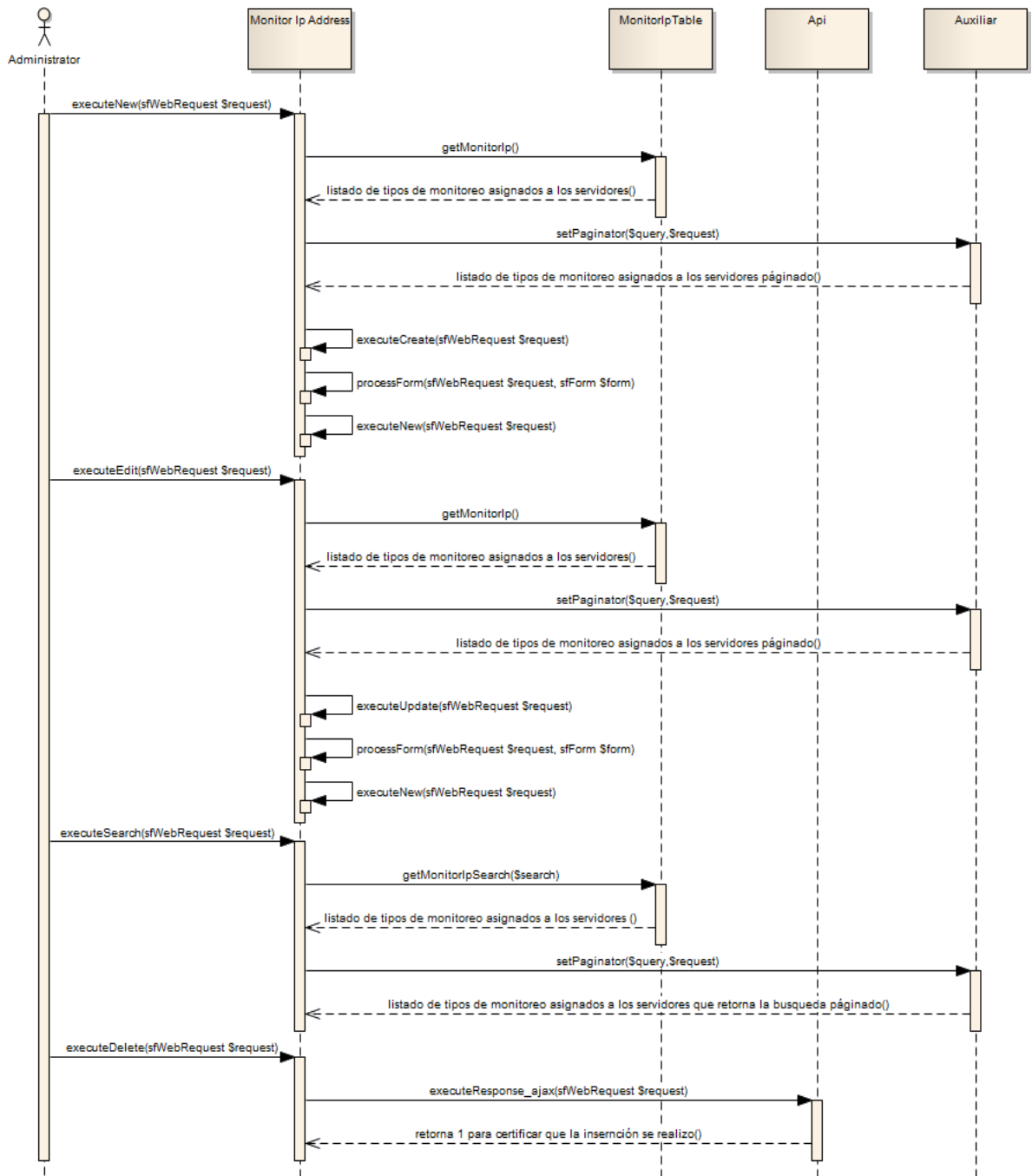


Ilustración 24. Diagrama de secuencia - monitor ip address.

- Diagrama de Secuencia: Monitor User.

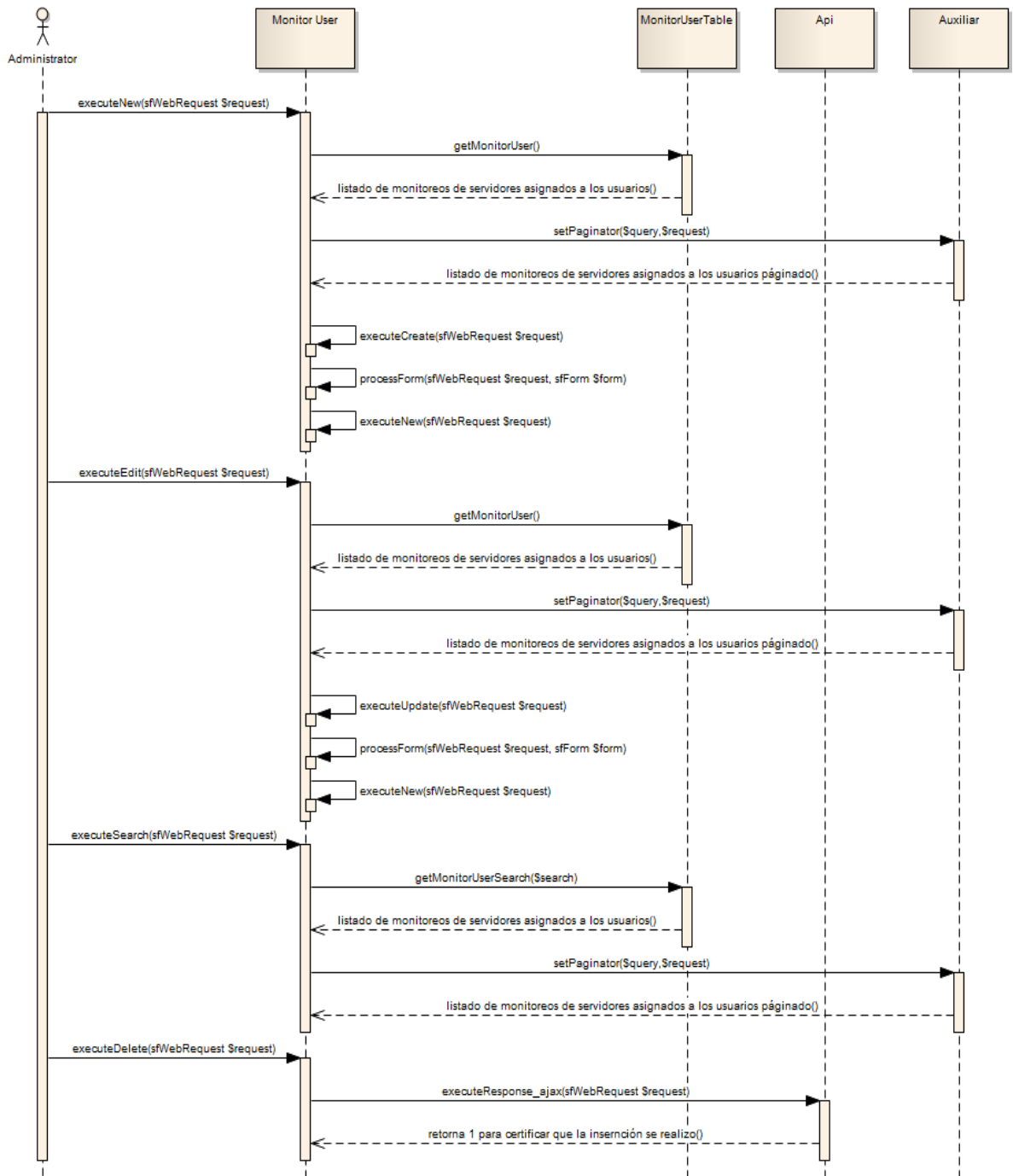


Ilustración 25. Diagrama de secuencia - monitor user.

- Diagrama de Secuencia: Monitoring – History.

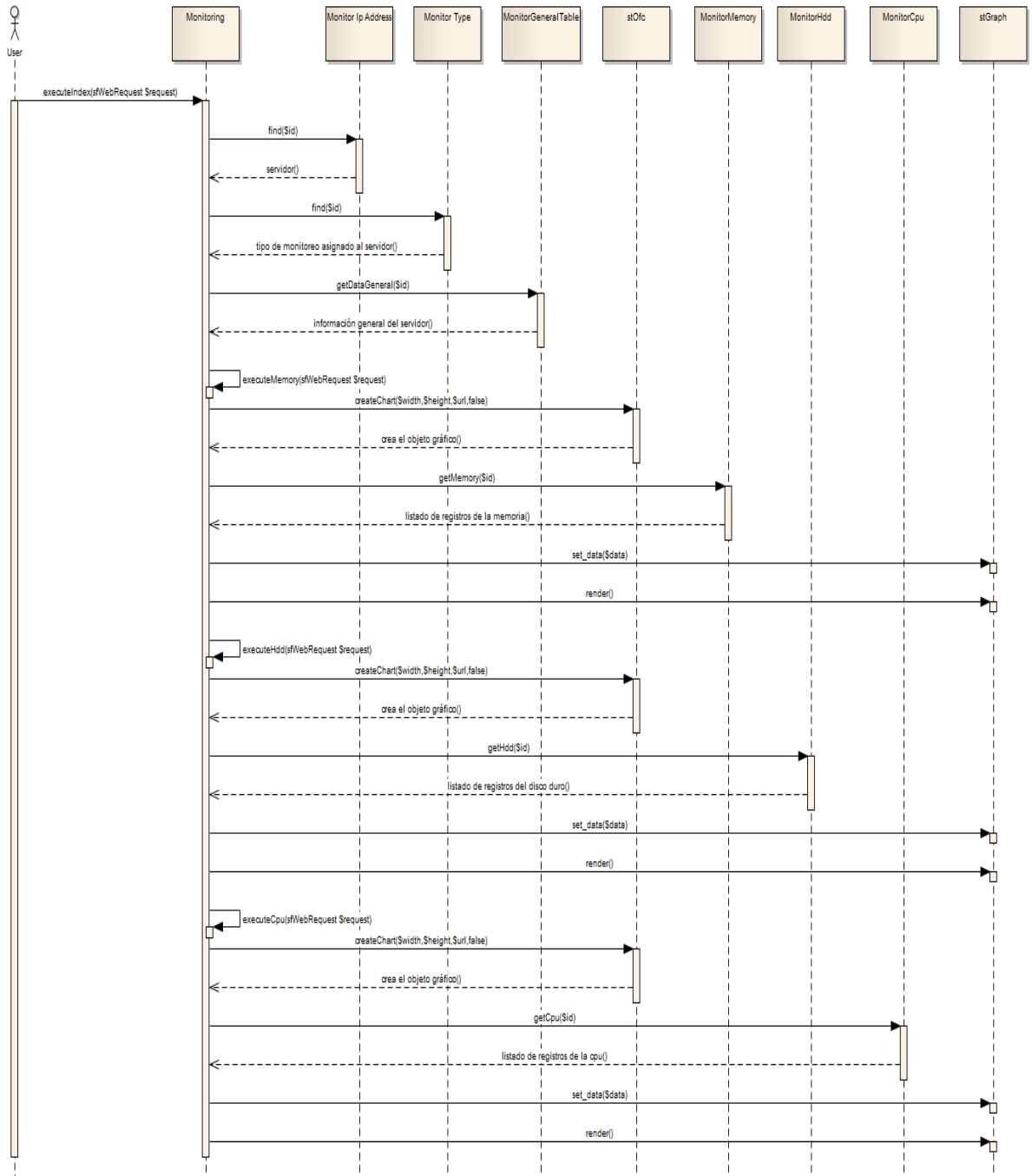


Ilustración 26. Diagrama de secuencia - monitoring history.

- Diagrama de Secuencia: Monitoring – Realtime.

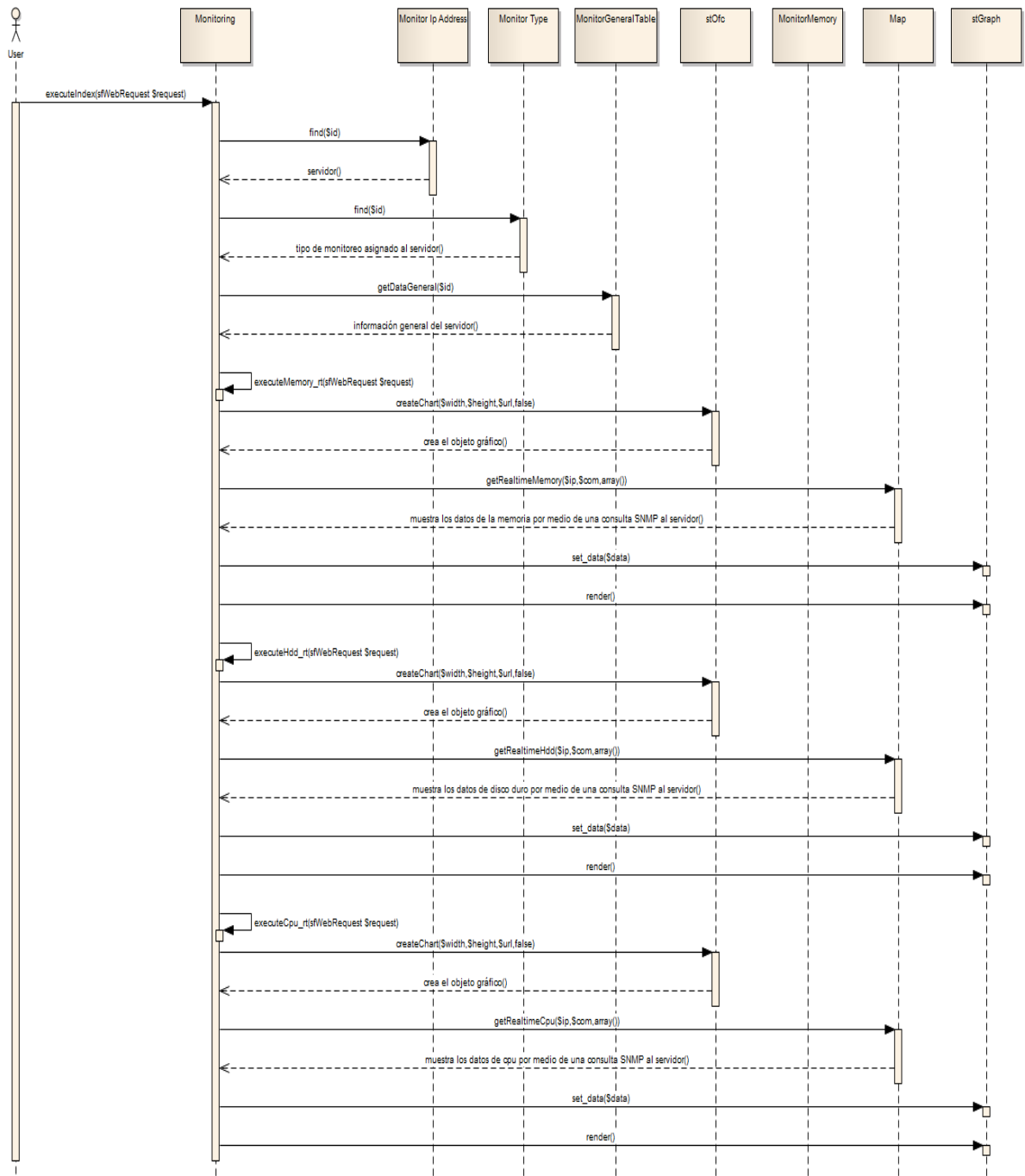


Ilustración 27. Diagrama de secuencia - monitoring realtime.

- Diagrama de Secuencia: Server Status.

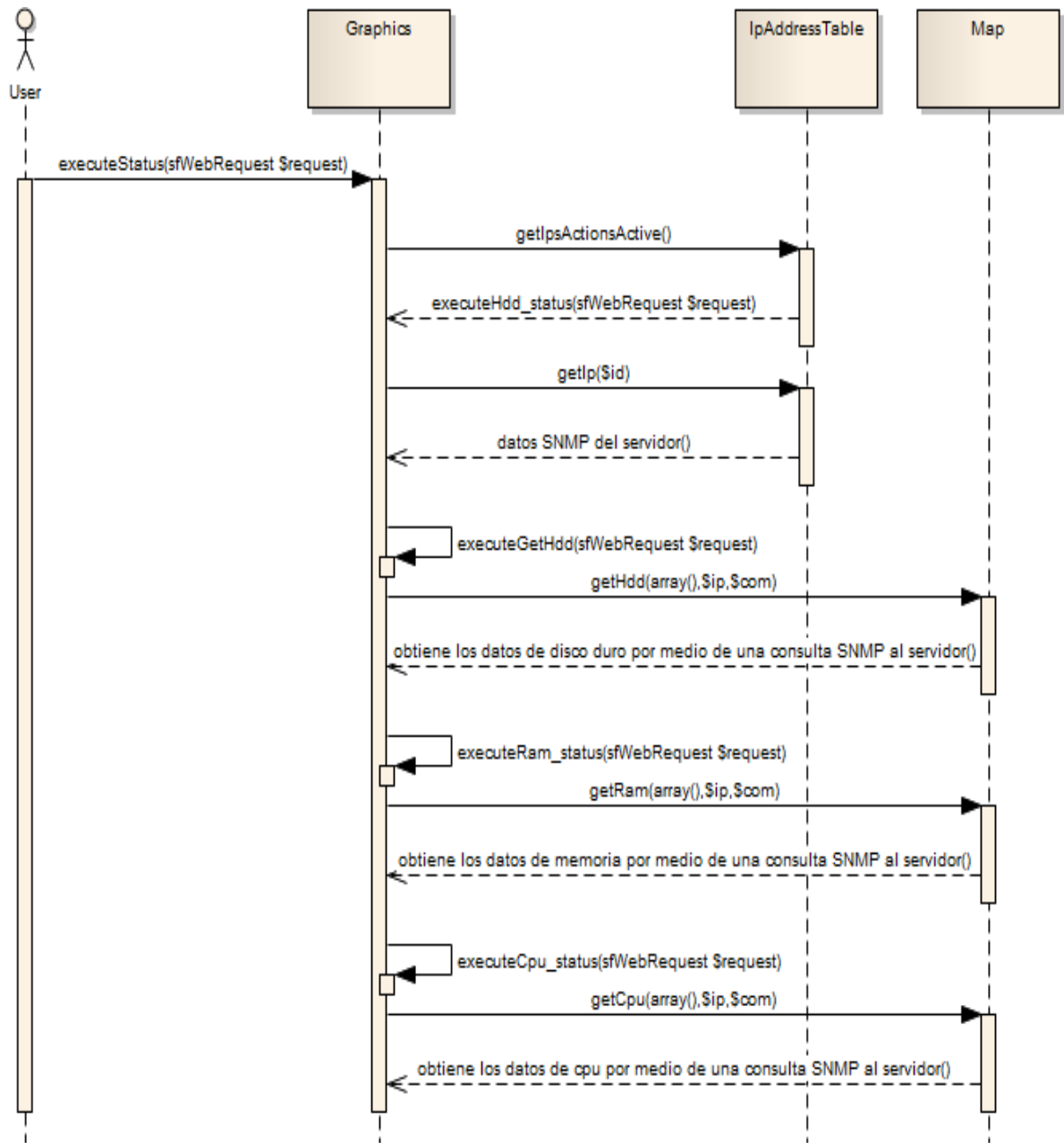


Ilustración 28. Diagrama de secuencia - server status.

- Diagrama de Secuencia: Permanent Monitoring – Hdd.

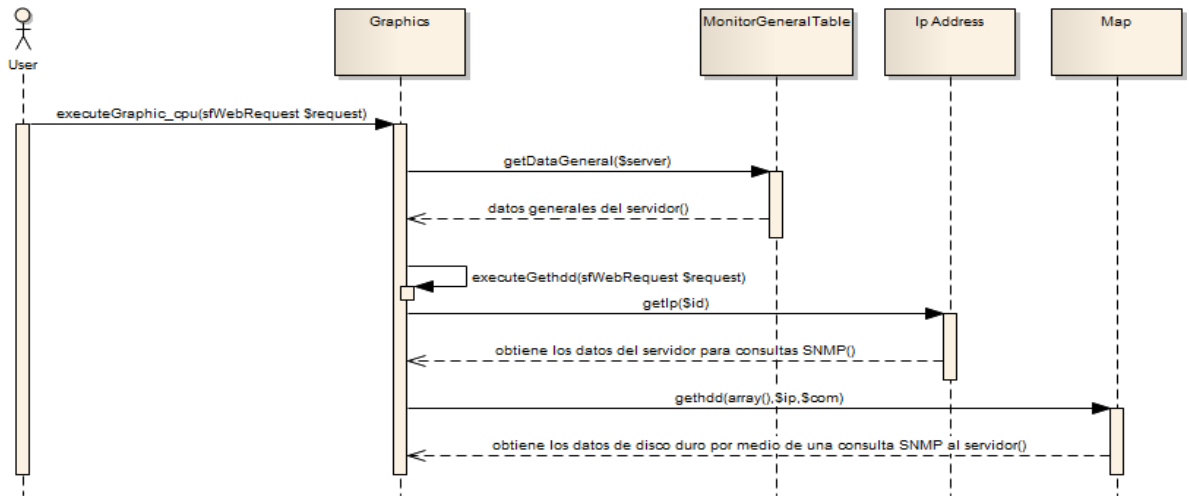


Ilustración 29. Diagrama de secuencia - permanent monitoring hdd.

- Diagrama de Secuencia: Permanent Monitoring – Memory.

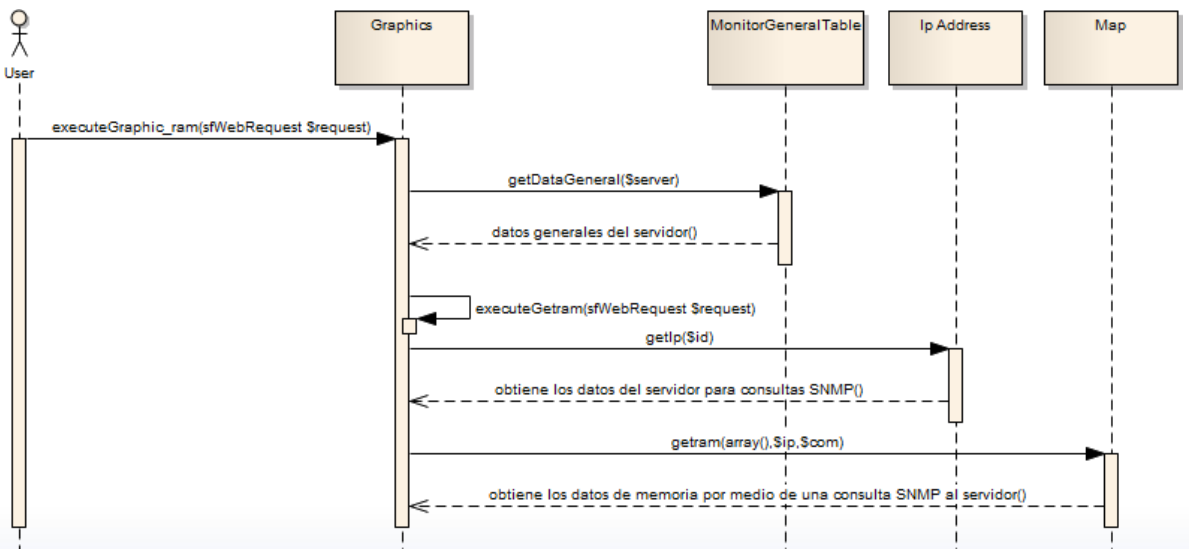


Ilustración 30. Diagrama de secuencia - permanent monitoring memory.

- Diagrama de Secuencia: Permanent Monitoring – Cpu.

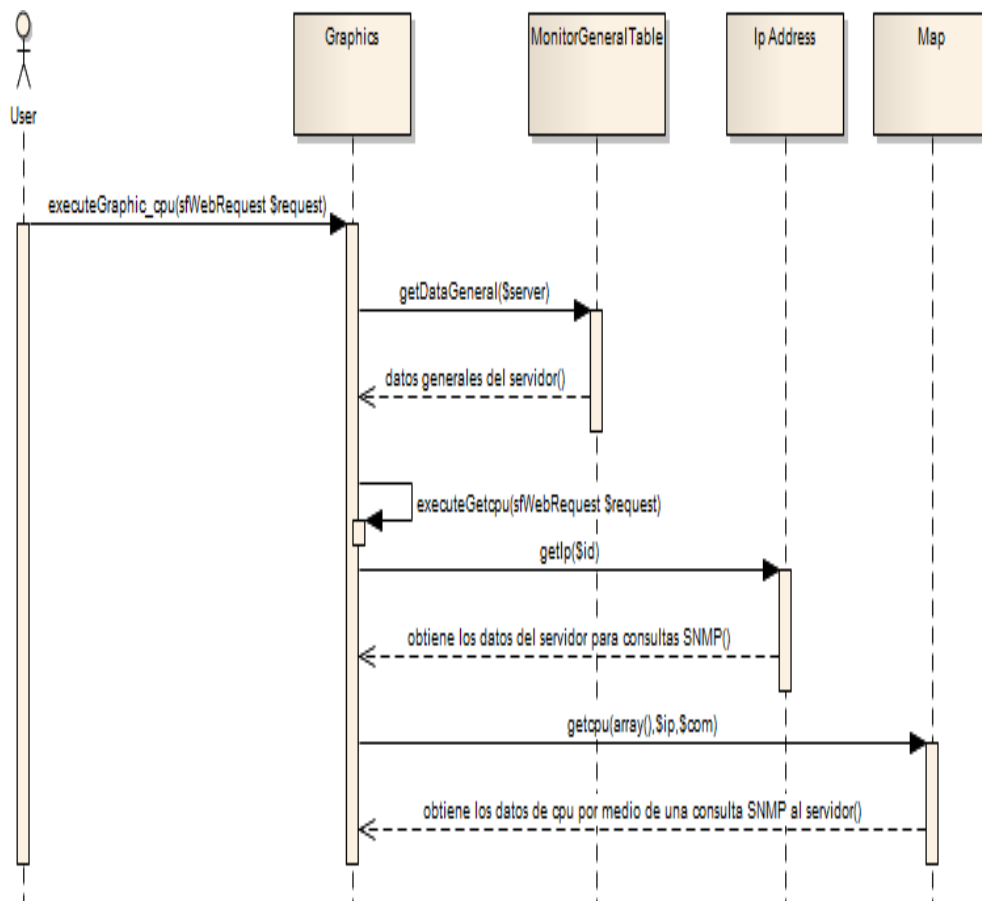


Ilustración 31. Diagrama de secuencia - permanent monitoring cpu.

- Diagrama de Secuencia: Reports – Excel.

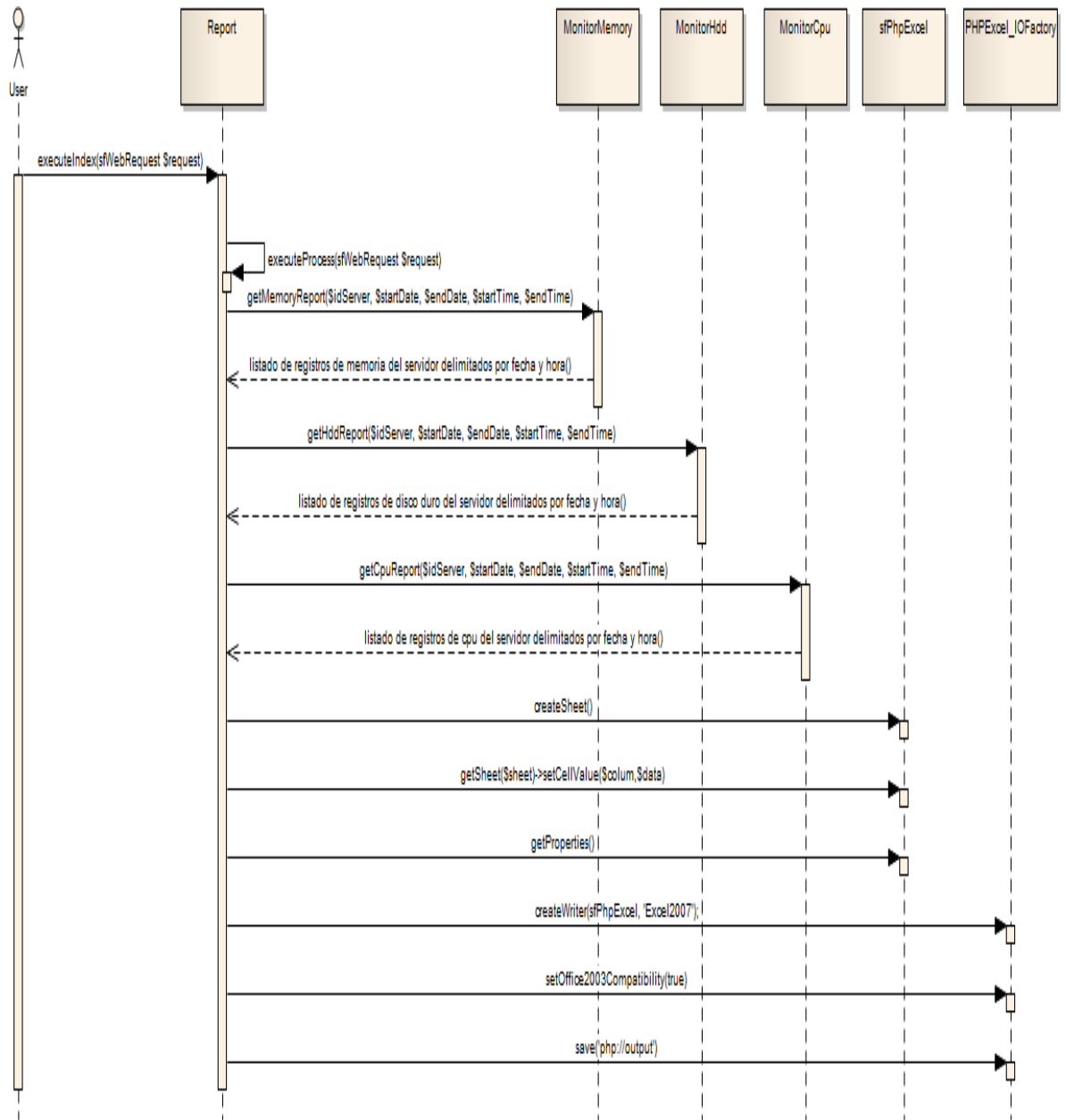


Ilustración 32. Diagrama de secuencia – reports excel.

- Diagrama de Secuencia: Reports – Pdf.

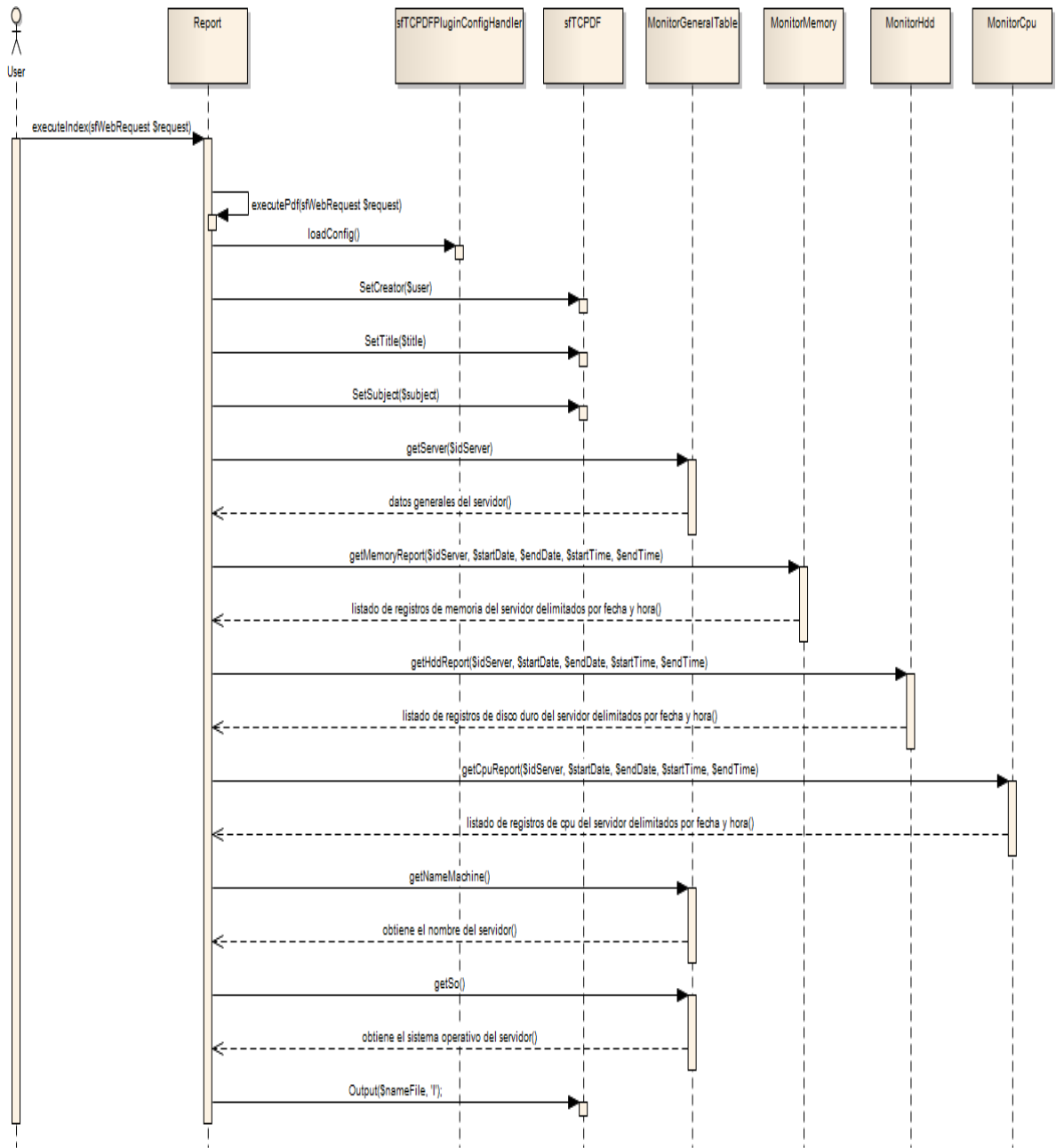


Ilustración 33. Diagrama de secuencia - reports pdf.

4.3.3.2 Aplicación Mobile.

- Diagrama de Secuencia: Configuration.

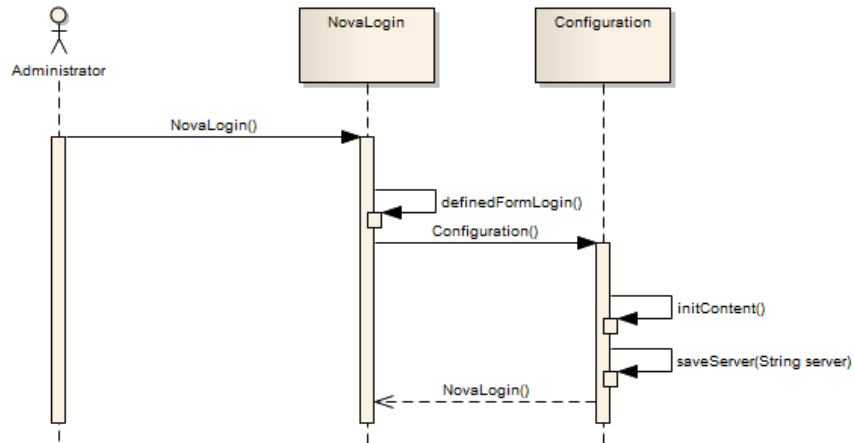


Ilustración 34. Diagrama de secuencia aplicación mobile - Configuration.

- Diagrama de Secuencia: Login & Logout.

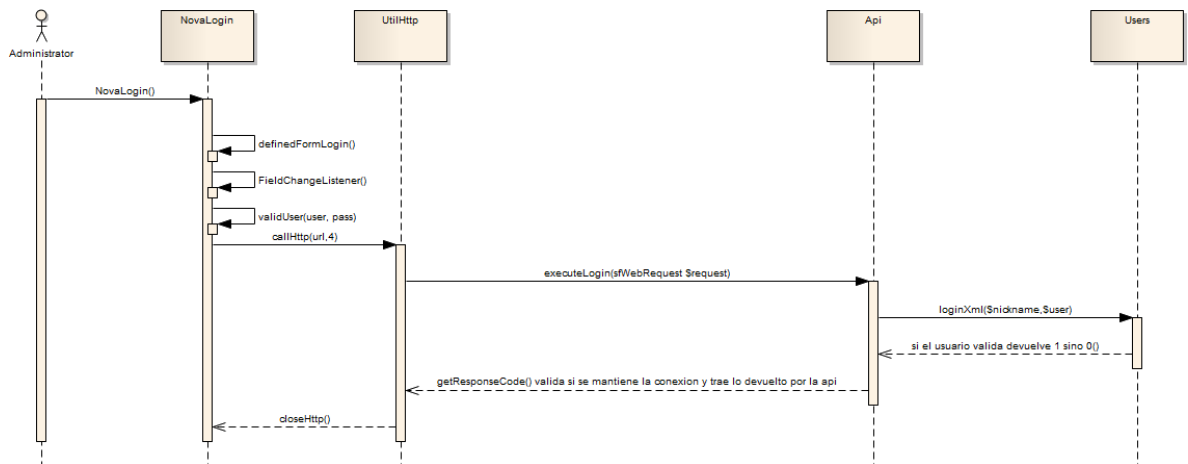


Ilustración 35. Diagrama de secuencia aplicación mobile - login & logout.

- Diagrama de Secuencia: List Servers.

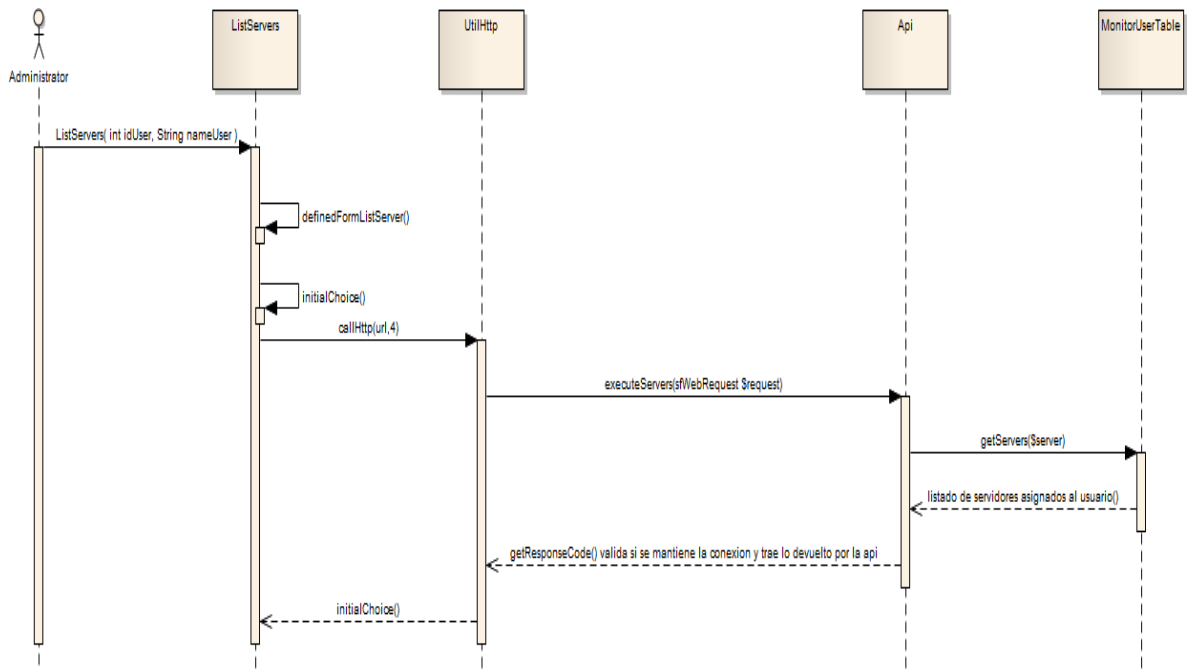


Ilustración 36. Diagrama de secuencia aplicación mobile - list servers.

- Diagrama de Secuencia: Graphic RAM

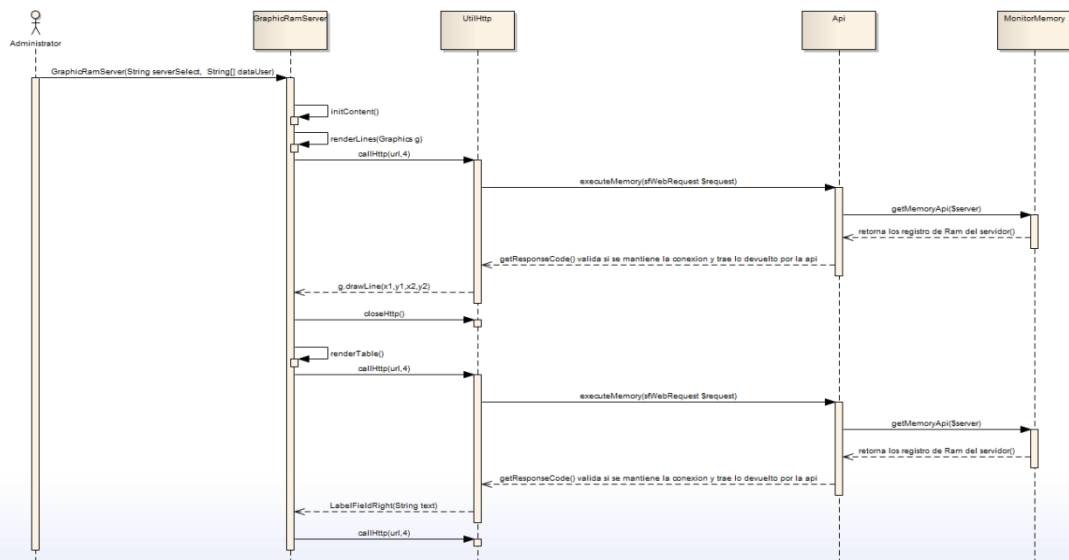


Ilustración 37. Diagrama de secuencia aplicación mobile - graphic RAM.

- Diagrama de Secuencia: Graphic CPU.

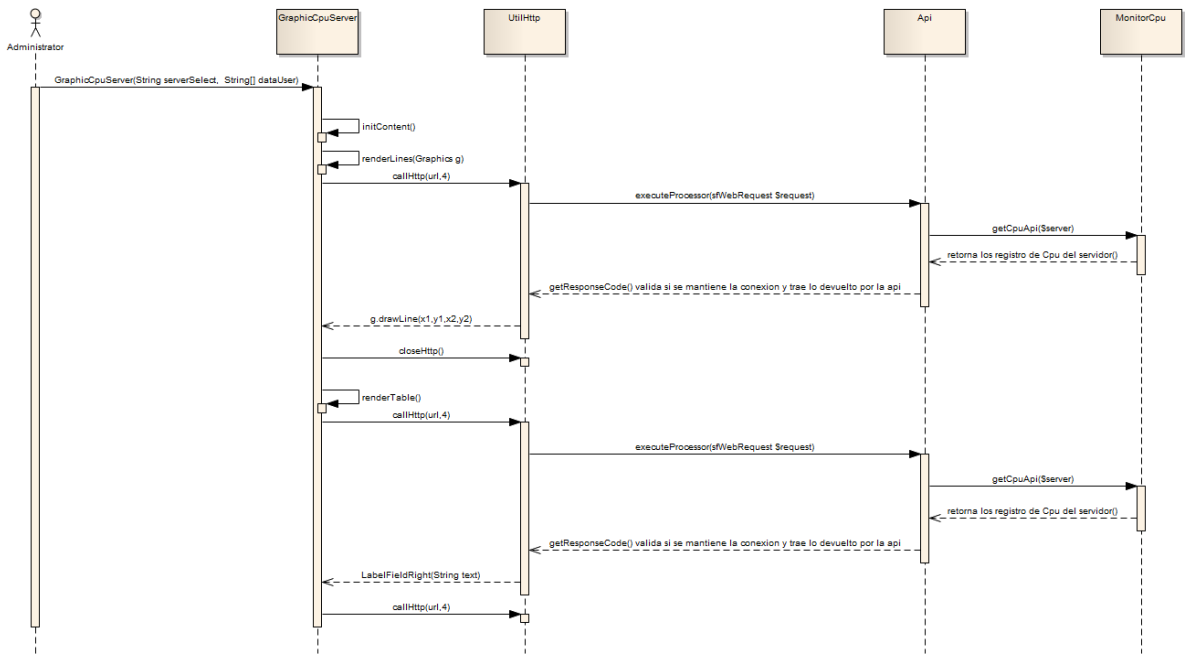


Ilustración 38. Diagrama de secuencia aplicación mobile - graphic cpu.

- Diagrama de Secuencia: Partitions.

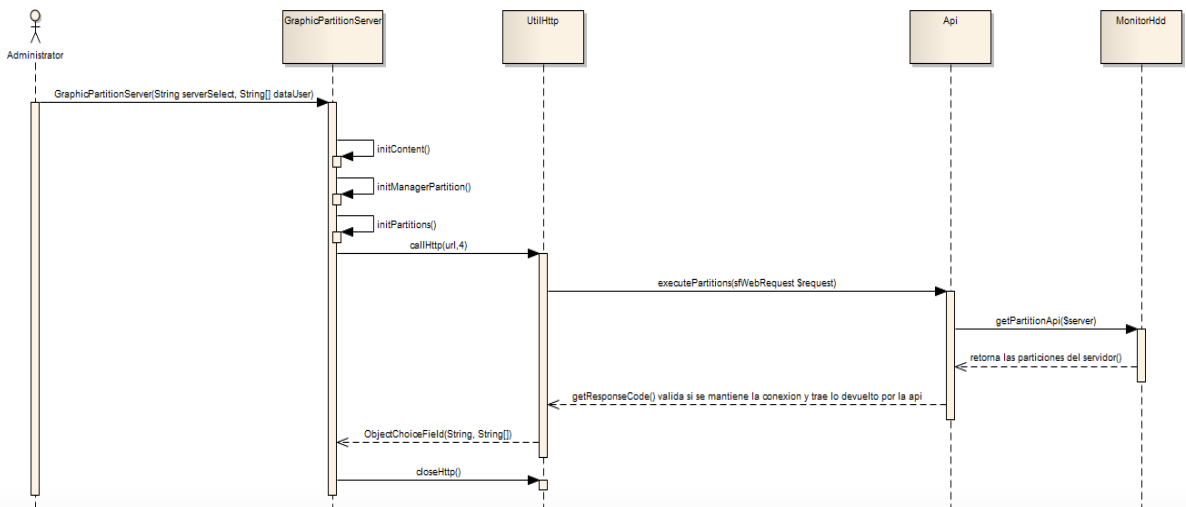


Ilustración 39. Diagrama de secuencia aplicación mobile - partitions.

- Diagrama de Secuencia: Graphic HDD.

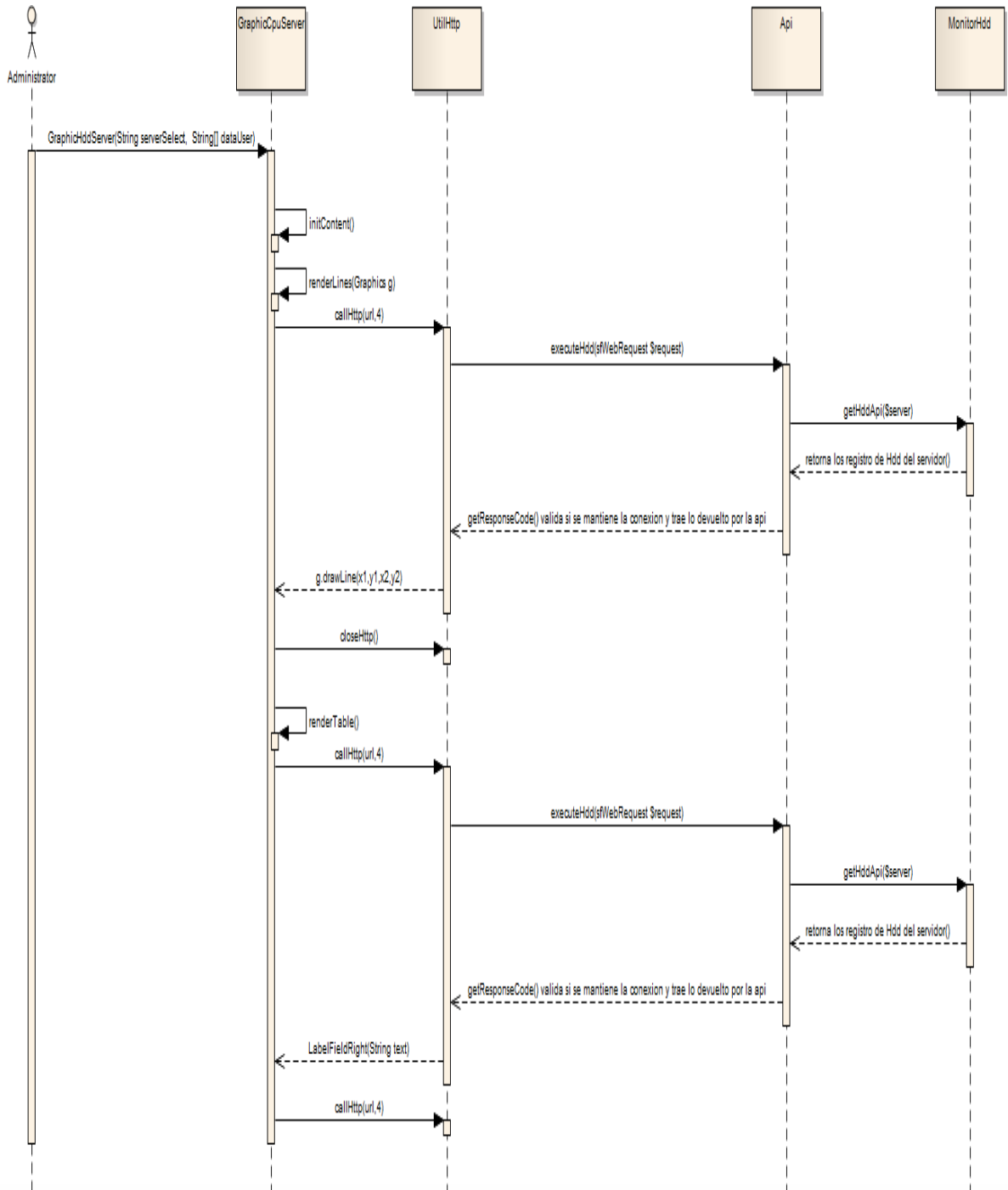


Ilustración 40. Diagrama de secuencia aplicación mobile - graphic HDD.

4.3.4 Diagramas de modelado de clases.

4.3.4.1 Aplicación web.

- Clase usersActions.

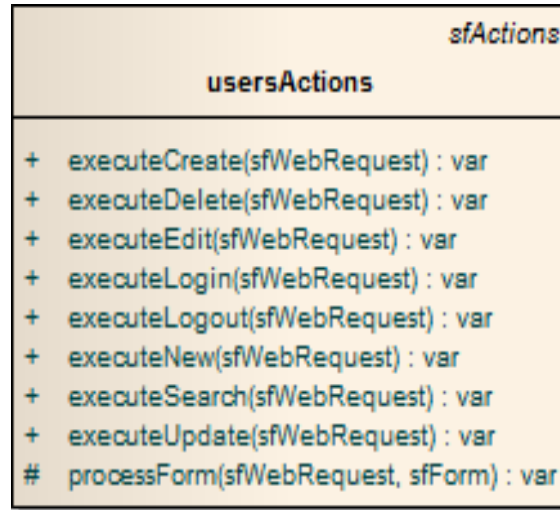


Ilustración 41. Diagrama de clase de usersActions.

- Clase UsersForm del formulario.

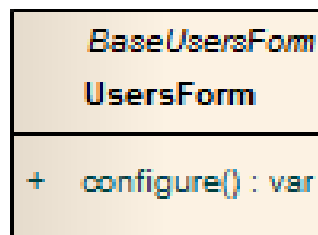


Ilustración 42. Diagrama de clase de UsersForm.

- Clase BaseUsersForm del formulario.

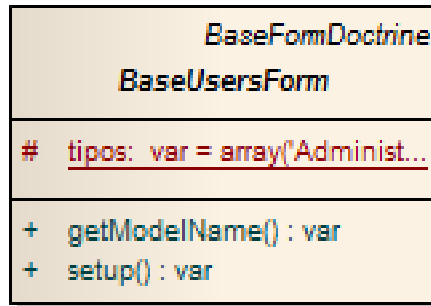


Ilustración 43. Diagrama de clase de BaseUsersForm.

- Clase UsersTable del Modelo.

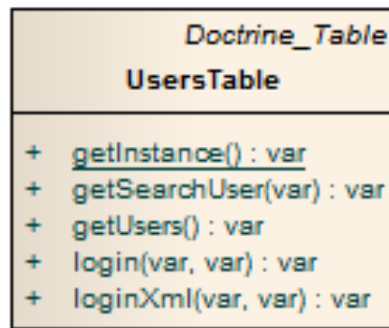


Ilustración 44. Diagrama de clase de UsersTable.

- Clase BaseUsers del Modelo.

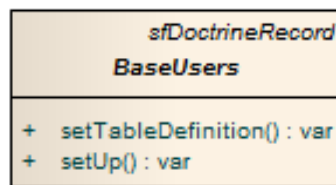


Ilustración 45. Diagrama de clase de BaseUsers.

- Clase ip_addressActions del módulo ip_address.



Ilustración 46. Diagrama de clase de ip_addressActions.

- Clase BaseIpAddressForm del formulario.

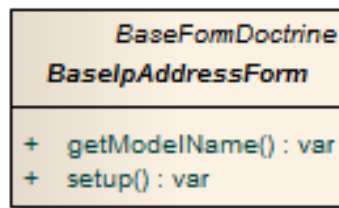


Ilustración 47. Diagrama de clase de BaseIpAddressForm.

- Clase IpAddressForm del formulario.



Ilustración 48. Diagrama de clase de IpAddressForm.

- Clase BaseIpAddress del modelo.

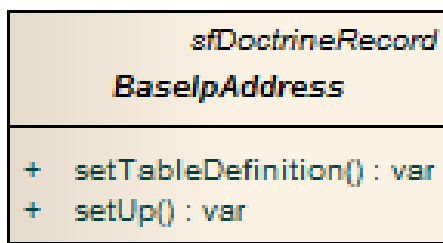


Ilustración 49. Diagrama de clase de BaseIpAddress.

- Clase IpAddress del modelo.

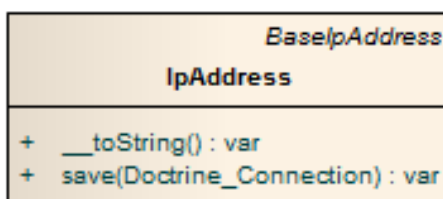


Ilustración 50. Diagrama de clase de IpAddress.

- Clase IpAddressTable del modelo.

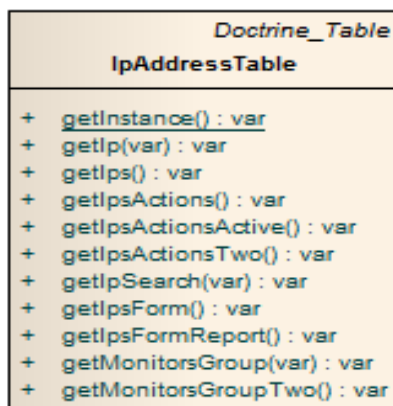


Ilustración 51. Diagrama de clase de IpAddressTable.

- Clase monitor_typeActions del módulo monitor Type.

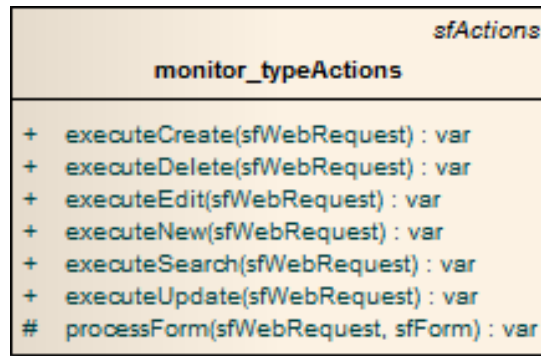


Ilustración 52. Diagrama de clase de monitor_typeActions.

- Clase MonitorTypeForm del formulario.

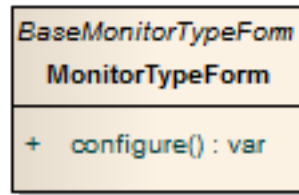


Ilustración 53. Diagrama de clase de MonitorTypeForm.

- Clase BaseMonitorTypeForm del formulario.

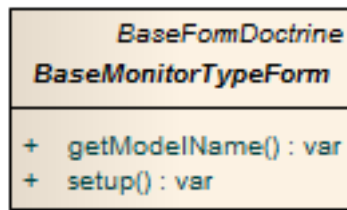


Ilustración 54. Diagrama de clase de BaseMonitorTypeForm.

- Clase BaseMonitorType del modelo.

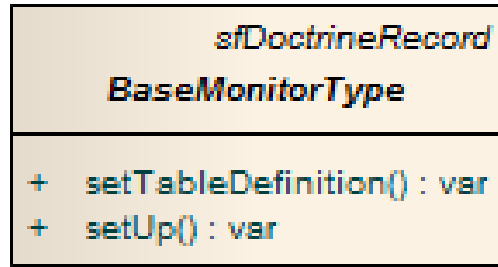


Ilustración 55. Diagrama de clase de BaseMonitorType.

- Clase MonitorType del modelo.

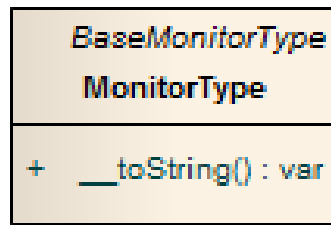


Ilustración 56. Diagrama de clase de MonitorType.

- Clase MonitorTypeTable del modelo.

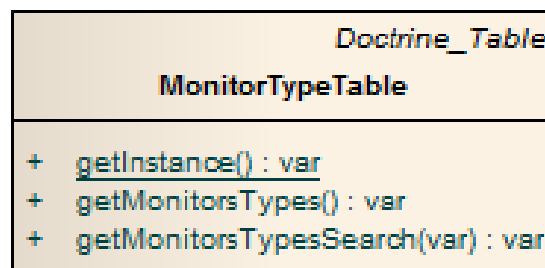


Ilustración 57. Diagrama de clase de MonitorTypeTable.

- Clase monitor_ipActions del módulo monitor_ip.

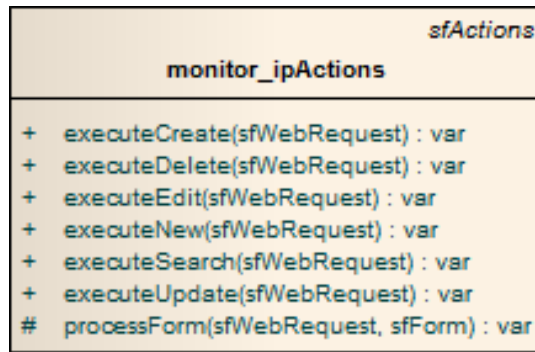


Ilustración 58. Diagrama de clase de monitor_ipActions.

- Clase BaseMonitorIp del Formulario.

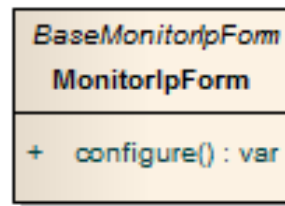


Ilustración 59. Diagrama de clase de BaseMonitorIp.

- Clase BaseMonitorIpForm del formulario.

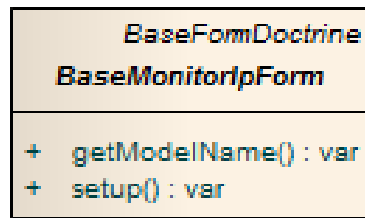


Ilustración 60. Diagrama de clase de BaseMonitorIpForm.

- Clase BaseMonitorIp del modelo.

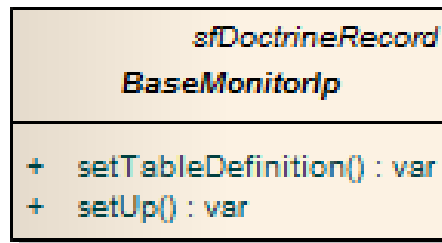


Ilustración 61. Diagrama de clase de BaseMonitorIp.

- Clase BaseMonitorIp del modelo.

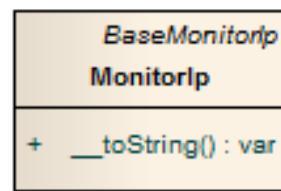


Ilustración 62. Diagrama de clase de MonitorIp.

- Clase MonitorIpTable del modelo.

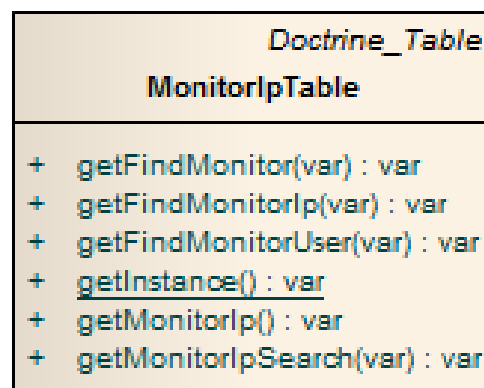


Ilustración 63. Diagrama de clase de MonitorIpTable.

- Clase monitor_userActions del módulo monitor_user.

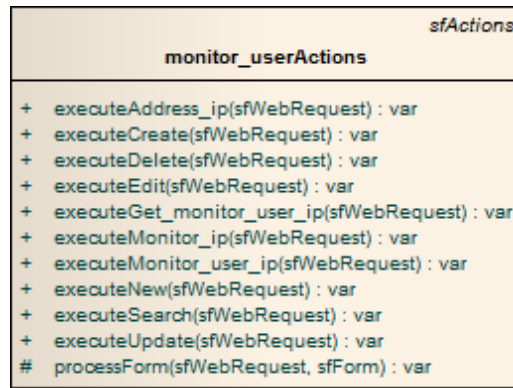


Ilustración 64. Diagrama de clase de monitor_userActions.

- Clase MonitorUserForm del formulario.

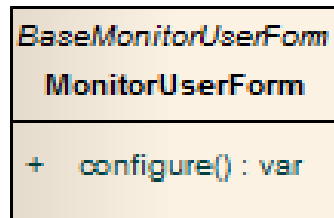


Ilustración 65. Diagrama de clase de MonitorUserForm.

- Clase BaseMonitorUserForm del formulario.

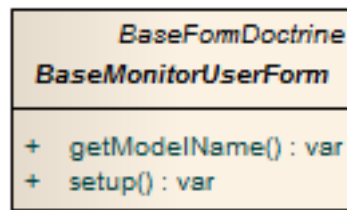


Ilustración 66. Diagrama de clase de BaseMonitorUserForm.

- Clase BaseMonitorUser del formulario.

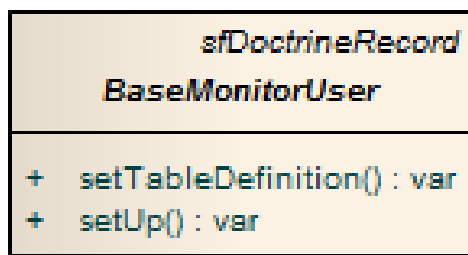


Ilustración 67. Diagrama de clase de BaseMonitorUser.

- Clase MonitorUser del modelo.

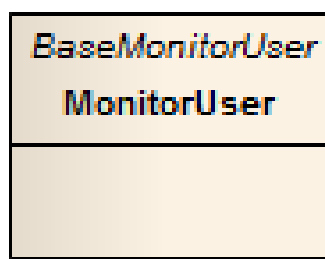


Ilustración 68. Diagrama de clase de MonitorUser.

- Clase MonitorUserTable del modelo.

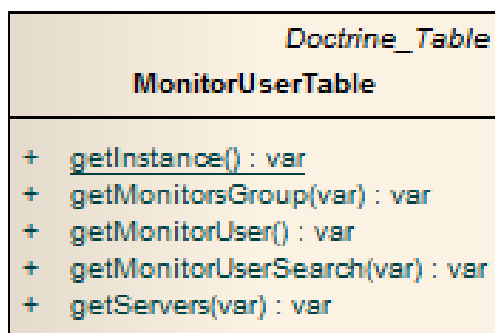


Ilustración 69. Diagrama de clase de MonitorUserTable.

- Clase graphicActions del módulo graphics.



Ilustración 70. Diagrama de clase de graphicActions.

- Clase reportActions del módulo Reports.



Ilustración 71. Diagrama de clase de reportActions.

- Clase apiActions del módulo Api.



Ilustración 72. Diagrama de clase de apiActions.

- Clase mappingTask.

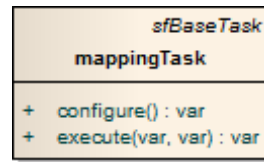


Ilustración 73. Diagrama de clase de mappingTask.

- Diagrama de clases: SnmpFormat & Map.

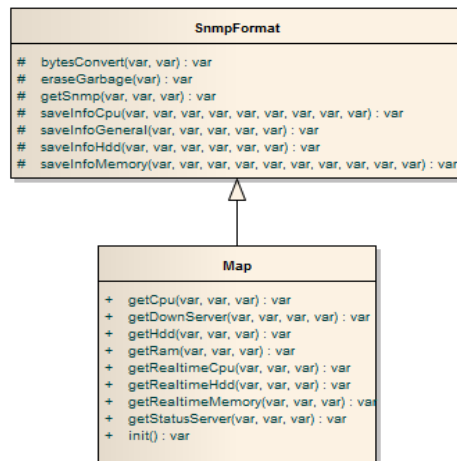


Ilustración 74. Diagrama de clase de SnmpFormat & Map.

- Clase Auxiliar.

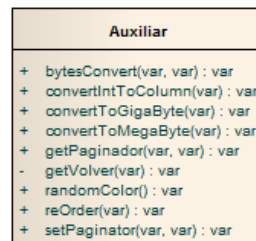


Ilustración 75. Diagrama de clase de Auxiliar.

4.3.4.2 Aplicación Mobile.

- Clase del Paquete utils.

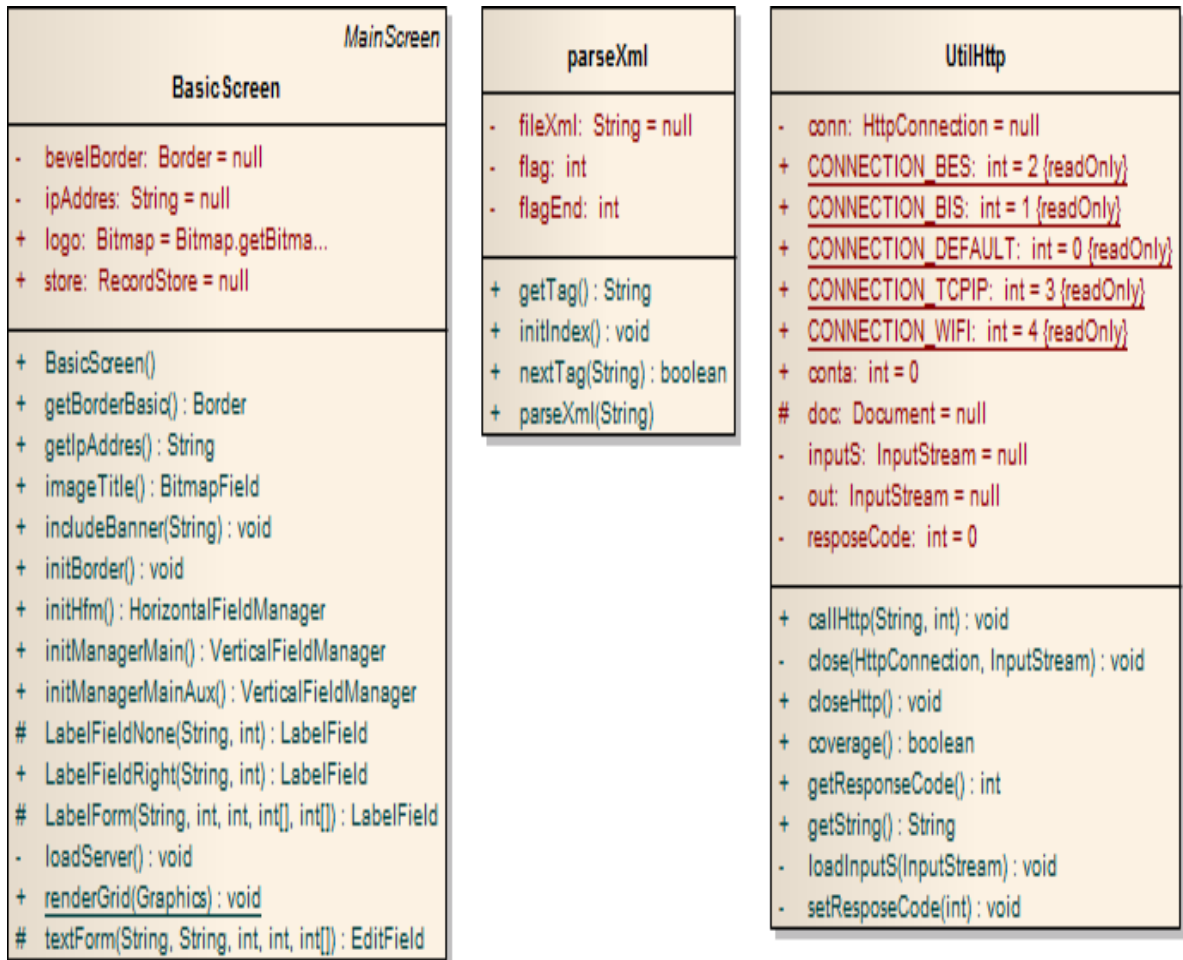


Ilustración 76. Diagrama de clases de paquete utils del entorno mobile.

- Clases del Paquete screens.

<i>BasicScreen</i> <i>FieldChangeListener</i>
Configuration
- contentImg: BitmapField = null - lblConfig: LabelField = null - managerMain: VerticalFieldManager = null - saveBtn: ButtonField = null - txtServer: EditField = null
+ Configuration() + fieldChanged(Field, int) : void - initContent() : void + saveServer(String) : void

<i>BasicScreen</i> <i>KeyListener</i>
GraphicCpu Server
- dataUser: String ([]) = new String[2] - httpUtil: UtilHttp = null - managerGraphic: VerticalFieldManager = null - managerMain: VerticalFieldManager = null - serverSelect: String = null
+ GraphicCpuServer(String, String[]) - initContent() : void - initManagerGraphic() : VerticalFieldManager + keyChar(char, int, int) : boolean + keyDown(int, int) : boolean + keyRepeat(int, int) : boolean + keyStatus(int, int) : boolean + keyUp(int, int) : boolean - renderLines(Graphics) : void - renderTable() : void

<i>BasicScreen</i> <i>KeyListener</i>
GraphicHdd Server
- dataUser: String ([]) = new String[2] - httpUtil: UtilHttp = null - managerGraphic: VerticalFieldManager = null - managerMain: VerticalFieldManager = null - partitionSelect: String = null - serverSelect: String = null
+ GraphicHddServer(String, String, String[]) - initContent() : void - initManagerGraphic() : VerticalFieldManager + keyChar(char, int, int) : boolean + keyDown(int, int) : boolean + keyRepeat(int, int) : boolean + keyStatus(int, int) : boolean + keyUp(int, int) : boolean - renderLines(Graphics) : void - renderTable() : void

<i>BasicScreen</i> <i>KeyListener</i>
GraphicPartition Server
+ actions: FieldChangeListener = new FieldChange... - dataUser: String ([]) = new String[2] - httpUtil: UtilHttp = null - itemRefresh: MenuItem = new MenuItem("R... - managerMain: VerticalFieldManager = null - partitions: ObjectChoiceField = null - serverBtn: ButtonField = null - serverSelect: String = null
+ GraphicPartitionServer(String, String[]) - initContent() : void - initManagerPartition() : void - initPartitions() : String[] + keyChar(char, int, int) : boolean + keyDown(int, int) : boolean + keyRepeat(int, int) : boolean + keyStatus(int, int) : boolean + keyUp(int, int) : boolean + onClose() : boolean

<i>BasicScreen</i> <i>KeyListener</i>
GraphicRam Server
- dataUser: String ([]) = new String[2] + httpUtil: UtilHttp = null - managerGraphic: VerticalFieldManager = null - managerMain: VerticalFieldManager = null + serverSelect: String = null
+ GraphicRamServer(String, String[]) - initContent() : void - initManagerGraphic() : VerticalFieldManager + keyChar(char, int, int) : boolean + keyDown(int, int) : boolean + keyRepeat(int, int) : boolean + keyStatus(int, int) : boolean + keyUp(int, int) : boolean - renderLines(Graphics) : void - renderTable() : void

<i>BasicScreen</i> <i>FieldChangeListener</i>
ListServers
- banner: LabelField = null + contentImg: BitmapField = null - dataUser: String ([]) = new String[2] - httpUtil: UtilHttp = null - idUser: int = 0 - listServers: String ([]) = null - manager: VerticalFieldManager = null + serverBtn: ButtonField = null - servers: ObjectChoiceField = null - types: ObjectChoiceField = null ~ typeView: String ([]) = { "Cpu", "Memor... {readOnly}
- definedFormListServer() : void + fieldChanged(Field, int) : void - initialChoice() : void + ListServers(int, String) + onClose() : boolean

<i>BasicScreen</i> <i>FieldChangeListener</i>
NovaLogin
- config: MenuItem = new MenuItem("C... - contentImg: BitmapField = null - httpUtil: UtilHttp = null - managerMain: VerticalFieldManager = null - passLbl: LabelField = null - sendBtn: ButtonField = null - userField: EditField = null - userLbl: LabelField = null - userPass: PasswordEditField = null
+ fieldChanged(Field, int) : void - initContent() : void + NovaLogin() + onClose() : boolean - validUser(String, String) : void

Ilustración 77. Diagrama de clases del paquete screens del entorno mobile.



- Diagrama de Clases: Paquete init.

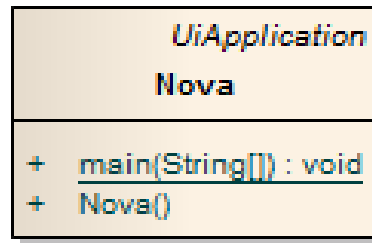


Ilustración 78. Diagrama de clases del paquete init del entorno mobile.

5. RECOMENDACIONES

5.1. GENERALIDADES DE LA APLICACIÓN

5.1.1 Arquitectura de Red.

Para una óptima arquitectura de red se recomienda tener en cuenta la siguiente arquitectura de red:

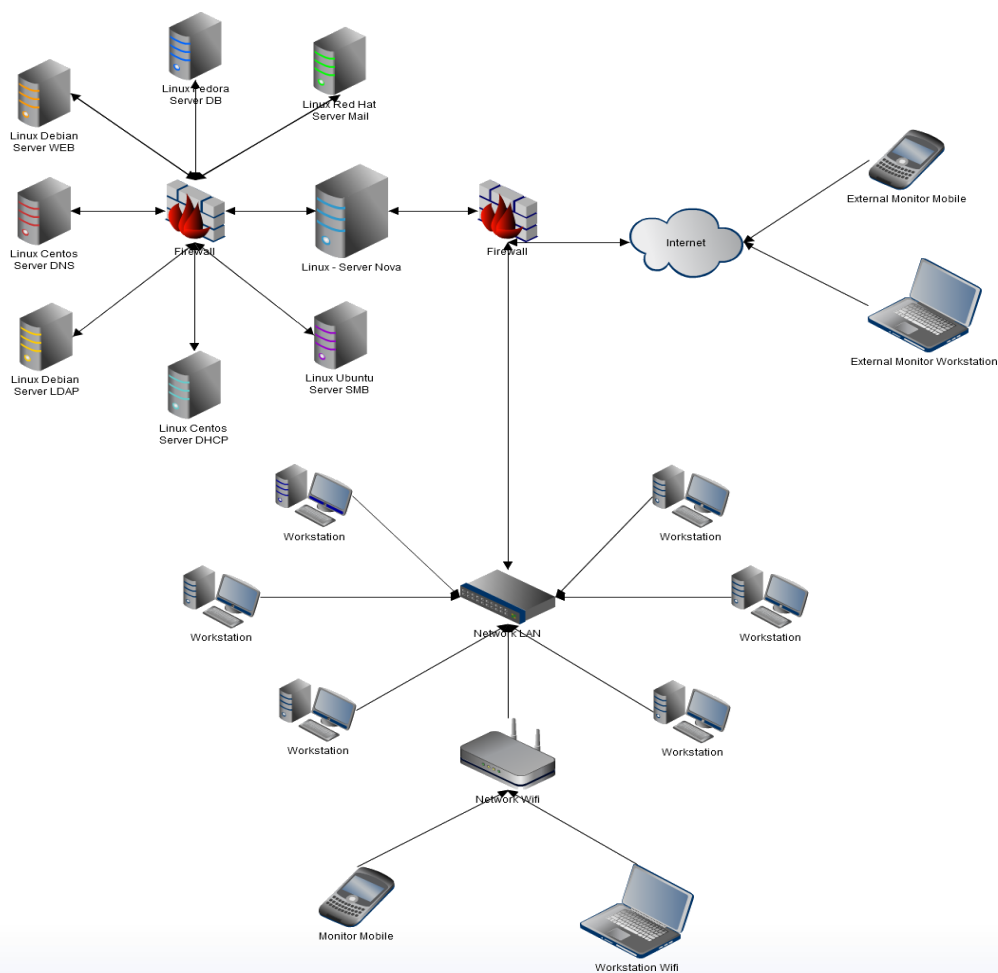


Ilustración 79. Arquitectura de red para la ejecución de Nova Realtime.



5.2 Optimización de servidor para Nova Realtime.

Para tener un servidor con un rendimiento óptimo se recomienda instalar el acelerador apc para php, ya que permite tener cacheada la aplicación pre compilándola en bytecode.

5.3 Optimización de servidor para base de datos.

Para mejorar consultas de búsquedas en la base de datos se recomienda emplear índices que se caractericen por los campos principales de búsqueda.

No se debe crear más índices de los necesarios, ya que esto puede ser contraproducente para el mejoramiento de las consultas.

Procure que el tipo de dato anisado para cada campo nuevo, sea el apropiado y que los demás campos de otras tablas que asocien la misma información mantengan el mismo formato.

Trate le mantener las tablas solo con la información necesaria, mantenga definidas sus llaves primarias y evitar llenar los campos de valores nulos esto genera más lentitud en las consultas.

6. ESTIMACION DE PERTINENCIA PARA SOFTWARE DE MONITOREO DE SERVIDORES GNU/LINUX

6.1 ESTIMACION DE PERTINENCIA SIN SISTEMA DE MONITOREO

Los sistemas GNU/Linux y entornos Unix cuentan con una serie de utilidades o comandos que nos permiten realizar el monitoreo de forma manual, en este tutorial mostraremos el uso de los comandos que nos permitirán el monitoreo de Hdd, Cpu y memoria, esto con el fin de evaluar el procedimiento manual de monitoreo.

6.1.1 Marco de referencia para monitoreo manual de servidores

En inicio veremos la utilización de comandos Linux para monitoreo de servidores y dispositivos, evaluar cuál sería la metodología para la elaboración de reportes y manejo de la información, también evaluaremos una de la uno de los sistemas existentes.

6.1.2 Instrumentos de aplicación.

Para obtener la información de estado de servidor ingresaremos por una consola de administración ssh y se consultara el estado del hardware, con el uso de comandos, esto nos permitirá extraer información necesaria para luego armar los reportes e informes de los dispositivos evaluados en un archivo de Excel.

- **Modo de ingreso Ssh**, Para acceder a los servidores de la red y realizar el monitoreo de cada uno de ellos utilizaremos el protocolo Ssh, este nos permitirá comunicarnos entre los sistemas utilizado la arquitectura cliente/servidor, nos proporciona una conexión remota y seguridad, tanto para el sistema cliente como para el sistema remoto.

Para este caso emplearemos la herramienta PUTTY que nos proporcionará el acceso ssh y podremos conectarnos con los servidores, crear sesiones, crear usuarios y definir permisos de acceso usuarios.

- **Comandos que proporcionan información**

- Comando **df** (disk free) se puede ver cuánto espacio tenemos disponible en nuestras unidades, mientras que con **du** se hace la operación inversa (cuanto tenemos ocupado en la ruta actual). Con **fsck** podemos realizar chequeos para comprobar el estado de una partición.

```
Filesystem            1K-blocks      Used Available Use%
Mounted on
/dev/mapper/VolGroup00-LogVol100
  o 150263916 14440324 128067408 11% /
/dev/sda1              101086         10896     84971 12% /boot
tmpfs                  253336          0       253336  0%
/dev/shm
```

En la salida anterior:

```
/dev/mapper/VolGroup00-LogVol100
  → Especifica el sistema de archivos.
/dev/sda1
  → Especifica el sistema de archivos.
tmpfs
  → Especifica el sistema de archivos.
```

- **Comando Top**, Es una herramienta que permite obtener una visión dinámica en tiempo real. El comando muestra un listado de los procesos que se están ejecutando, también proporciona información de uso de memoria y procesador.

```
top - 13:31:12 up 3:42, 2 users, load average: 1.58, 1.39, 1.11
Tasks: 140 total, 3 running, 134 sleeping, 2 stopped, 1 zombie
Cpu(s): 33.3%us, 11.3%sy, 0.0%ni, 54.7%id, 0.0%wa, 0.3%hi, 0.3%si, 0.0%st
Mem: 2065636k total, 1190052k used, 875584k free, 55020k buffers
Swap: 1903660k total, 0k used, 1903660k free, 601680k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 3474 root        20   0   191m 122m 10m  S   44   6.1   22:55.00 wish8.5
 3383 root        20   0   475m 200m 37m  R   17  10.0   44:03.88 firefox-bin
 1887 root        20   0   391m  59m 18m  S   13   2.9   15:12.75 Xorg
 3410 root        20   0   145m  45m 20m  S   11   2.2   21:34.25 rhythmbox
```

Ilustración 80. Representación de datos visualizados con el comando top.



Podemos observar en la Ilustración 80 algo de la información que nos proporciona el comando top.

- PID
 - USUARIO
 - %CPU
 - %MEMORIA
 - NOMBRE
- **El comando uptime:** Este comando indica:
- La hora actual
 - El tiempo que el sistema está en marcha
 - El número de usuarios conectados
 - La carga promedio del sistema para los últimos 1, 5 y 15 minutos

```
#uptime
```

```
10:21:06 up 5 days, 19:11, 2 users, load average: 0.08, 0.03, 0.00
```

- Comando **Free** muestra información de memoria relativa, el resultado obtenido es estático, para visualizar información de manera dinámica, podemos incluir el comando **watch**.

```
#watch -n 1 -d free
```

```
Every 1,0s: free  Wed Sep 23 10:22:47 2009
```

	Total	used	free	shared	buffers
cached					

```
Mem:3632736 1049556 2583180 0 359644 255072
```

-/+ buffers/cache: 434840 3197896

Swap:1991920 0 1991920

La opción `-n` indica el tiempo de refrescamiento de la información y `-d` para resaltar la evolución.

- El comando **Vmstat**, proporciona información sobre uso de memoria virtual por los threads del núcleo, el disco y la actividad del procesador.

```
#vmstat
procs -----memory----- ---swap-- -----io-----
--system-- ----cpu----
r  b   swpd   free   buff  cache   si   so   bi   bo
in   cs us  sy id wa
1  0     0 2583032 359664 255088     0   0     1   14
63   22  1  0 99  0
```

- El Comando **lostat**, Muestra estadísticas sobre la lectura/escritura de los dispositivos.

```
#iostat
Linux 2.6.15-23-386 (owl02.owliancet.local)
23.09.2009
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           ■  0,53    0,03    0,04    0,07    0,00   99,33

Device:            tps   Blk_read/s   Blk_wrtn/s
Blk_read  Blk_wrtn
sda              1,53           1,21          28,79
604661   14429814
```

- **du**: muestra el espacio en disco usado por los directorios y sub-directorios, por defecto, partiendo del directorio en uso `'.'`. Si se pasa el nombre de un directorio como parámetro, el análisis se hará sobre éste.
- **df**: muestra la cantidad de espacio libre en los diferentes dispositivos montados o pasados como parámetro.



- **ps:** muestra una lista de los procesos que se están ejecutando y el nombre del usuario actual.
- **who:** muestra la lista de usuarios conectados.

6.1.3 Reportes y gráficos.

El acceso remoto y utilización de comandos para monitoreo, solo proporciona un medio de interacción con los servidores para visualizar su estado y comportamiento. La información será proporcional a la constancia con que el administrador de red o consultor verifique los estados de cada IP o Servidor.

Para la elaboración de reportes y graficas estadísticas, se requiere adquirir software o herramientas adicionales, esto con el fin de extraer la información a manipular. Otra de las posibilidades diseñar los cuadros, gráficos y reportes manualmente en documentos Excel y Word.

6.1.4 Observaciones.

La implementación de comandos para monitoreo es efectivo, en lo que refiere a consultar, analizar y proporciona la información de estada. El monitoreo manual es el más empleado por los operadores de sistemas pero este método se hace realmente costoso en lo que se refiere a esfuerzo (horas-hombre), los procesos se convierten en tareas repetitivas.

El almacenamiento de información, la elaboración de reportes, la generación de gráficos y demás medios que son primordiales para el análisis de información, depende de emplear más tiempo y recursos.

En consecuencia de que el almacenamiento de datos y la generación de reportes toma mucho tiempo y más recursos, la labor de administrador se hace complicada. La información obtenida no será contundente a la hora de tomar decisiones o prevenir mayores inconvenientes con el sistema.

7. CONCLUSIONES

Luego de analizar y evaluar los inconvenientes que implican realizar al monitoreo de servidores en forma manual, se hace evidente que este método no cumple a cabalidad con el objetivo principal, el cual consiste en alertar y prevenir de forma oportuna los inconvenientes que se presentan con cada equipo.

El monitoreo no solo implica realizar continuamente una revisión de cada Ip o dispositivo que se encuentra integrada en la red, también conlleva la generación de reportes y estadísticas que permitan al administrador estudiar y analizar el comportamiento, ya sea de información histórica o en tiempo real.

De los sistemas que actualmente encontramos en el mercado y que operan como gestores de monitoreo y administración de red, no llegan a ser realmente útiles, son complicados de manejar, generan sobrecostos para mantenimientos y modificaciones, además de consumir mucho recurso de máquina, esto hace inefectiva las herramientas actuales.

Los administradores y personal de IT siempre buscan estar involucrados con los avances tecnológicos. El hecho de contar con una plataforma Mobile como la que brinda Nova Realtime, que le permita estar en contacto con sus servidores se hace una herramienta indispensable para el monitoreo además es un elemento que no se encuentra en ninguno de los sistemas existentes.

BIBLIOGRAFÍA

ABREHAM. SILBERSCHATZ. Fundamentos de Bases de Datos. 4 ed. Madrid: Mac Graw Hill, 2002.

BRUCE. ECKEL. Piensa en Java. 2 ed. Madrid: Pearson, 2002.

DAITEL, Hardey M y Daitel, Paul J, Cuarta Edición Como programar en C ++ y Java, Pearson Educación, México 2004.

DOUGLAS. COMER. Redes Globales de Información con Internet y TCP/IP. 3 ed. México: Printice-Hall, 1995.

FABIE. POTIENCIER. The Symfony Reference Book. Francia: Sensio labs, 2009.

TANENBAUM. ANDREW. Redes de Computadoras. 4 ed. Ámsterdam: Pearson, 2003.

FRANCOIS ZANINOTTO, FABIEN POTIENCER. Symfony 1.2, La Guía Definitiva. Internet: (http://www.librosweb.es/symfony_1_2/).

NET-SNMP. Tutorials. Internet: (<http://www.net-snmp.org/wiki/index.php/Tutorials>).

ROGER. PERSSMAN. Ingeniería del Software. 5 ed. Madrid: Mac Graw Hill, 2002.

RESEARCH IN MOTION. BlackBerry 5 API. Internet: (<http://www.blackberry.com/developers/docs/5.0.0api/index.html>).

SENSIO. LABS. Symfony API. Internet: (http://www.symfony-project.org/api/1_4/).

Linux. Comandos para monitorear el sistema. Internet: (<http://es.kioskea.net/faq/3435-linux-comandos-para-monitorear-el-sistema>)