



Universidad  
Carlos III de Madrid

Grado en Ingeniería Informática

## TRABAJO FIN DE GRADO

---

# DISEÑO E IMPLANTACIÓN DE UN SISTEMA DE MONITORIZACIÓN BASADO EN NAGIOS

Autor: Alexandra Moreno Calabozo  
Tutor: Alejandro Calderón Mateos

Leganés, Septiembre de 2015



*“Reunirse es el principio,  
mantenerse unidos es progresar;  
trabajar unidos es triunfar.”*

Henry Ford

## **Agradecimientos**

Primeramente, agradecer todo el apoyo que me ha dado siempre mi pareja, Ramón, confiando y apoyando todos los pasos que he dado, tanto para desarrollar este proyecto, como para seguir adelante y hacer que nunca me rindiese ante cualquier adversidad.

A mi familia, por estar siempre a mi lado y confiar que podía superar todos los problemas tanto de horarios, de tiempo y de falta de horas de sueño para lograr este objetivo.

A mis amigos, por apoyarme en todo momento y animarme a que acabase este objetivo por todo lo alto, demostrando siempre todo lo que podía llegar a conseguir aunque haya tenido multitud de problemas de tiempos siempre.

A mis compañeros de trabajo, que día a día me han apoyado, no solo en este trabajo, si no durante todos los años de carrera. Muchos de ellos han sido parte importante de esta trayectoria.

A Alejandro Calderón, por ser un excepcional profesor, tutor y compañero de universidad, sabiendo en todo momento como apoyarte para que no te rindieras nunca.

A todos los compañeros de la universidad, tanto los que tuve el placer de compartir durante toda la carrera, como con los que estuve en algunos cursos. Mencionar en especial a Beatriz Martín, compañera de sufrimientos, horarios, trabajos y prácticas, durante los años que nos ha durado la carrera.

A todos los profesores de la universidad que me han ayudado a cumplir un gran objetivo y han compartido tantos conocimientos conmigo y con todos los compañeros.

Para finalizar, solo decir, gracias a todos.

*Alexandra Moreno Calabozo*

## Resumen

Esta memoria es parte del Trabajo Final de Grado realizado por Alexandra Moreno Calabozo para la Universidad Carlos III de Madrid. La idea nace durante mi trayectoria profesional en un proyecto de gestión y monitorización de servicios en un departamento de tecnologías de la información y comunicación (TIC), donde se pretende explotar una idea interesante, con objetivos claramente marcados, y con el suficiente nivel de conocimiento como para tener la oportunidad de exponer una herramienta de monitorización tan potente como Nagios.

El proyecto pretende demostrar las ventajas de usar una herramienta de monitorización, aportando usabilidad, disponibilidad, continuidad, rendimiento y normalidad, integrando esta herramienta en una red donde se pretende disponer de información instantánea de problemas en dispositivos, servidores y servicios que tienen un nivel de criticidad, siendo éstos muy importantes para facilitar un servicio a usuario y clientes, y posteriormente poder medir la satisfacción de los usuarios que acceden a ellos..

Para comenzar el proyecto se determinó un diseño de la configuración para implementar la herramienta de monitorización de Nagios [11], de manera que se siguiesen las líneas de continuidad del servicio, y se mantuviese el objetivo de minimizar los tiempos entre una interrupción de un servicio y una recuperación del mismo, para todos los facilitados por la Universidad desde el Departamento de Informática.

Nagios es una herramienta de monitorización avanzada, intuitiva y muy potente, con un número amplio de posibles chequeos definidos por defecto en la instalación del software, pero con una gran plataforma de software libre y con la posibilidad de desarrollar todos los scripts de monitorización que se deseen conociendo cualquier lenguaje de programación que reporte los datos de forma entendible para él.

Existe la posibilidad de monitorizar servidores con diferentes sistemas operativos, switches y todo tipo de dispositivos con una dirección IP en la red donde esté conectado Nagios. Además de monitorizar los chequeos básicos de PING, CPU, discos, memoria y swap, existe la posibilidad de monitorizar bases de datos, ficheros, accesos web, y más, siendo todos ellos personalizables. Todos los chequeos avanzados deberán ser desarrollados y personalizados para poder ser ejecutados en los servidores y/o clientes.

Para una mejor administración del software, se recomienda conocer varios lenguajes de programación, entre ellos, Java, HTML, PowerShell y programación de Linux/Unix (Bash/Shell), y teniendo la base de programación impartida en la Universidad, es fácilmente administrable un sistema de monitorización como Nagios.

Usando algunos de los lenguajes de programación mencionados en el anterior párrafo, es posible la monitorización de todos los servicios y procesos de cualquier servidor de una red pequeña, mediana o grande.

En los servidores con sistemas operativos de la familia Microsoft, desde Windows XP a Windows Server 2012, es posible controlar cualquier tipo de proceso y servicio, asociado o no al primero, mediante el agente de Nagios que está instalado en todos los servidores con estos sistemas. En cambio en los servidores con sistemas operativos de la familia Unix/Linux de software libre, es necesario realizar scripts para poder monitorizar todo por comandos, ya que para monitorizar un servicio o proceso, se deberán ejecutar comandos que no están definidos en ningún script precompilado en la instalación inicial.

En el caso de sistemas Windows, todos los chequeos se pueden realizar en modo activo, ya que Nagios se puede poner en contacto con el agente, y usar el modo request-response, para poder mostrar los resultados en la interfaz gráfica y éstos ser presentados en informes y tablas. En los sistemas Unix/Linux, hay varias formas de implementar chequeos, ya que pueden ser activos o pasivos, siendo los activos establecidos por el protocolo NRPE, mediante el puerto 5666 de comunicación entre el servidor de Nagios y el cliente, y los pasivos por NSCA, mediante el puerto 5667, habilitado también en ambos servidores.

Durante la presente memoria se redactan todos los puntos necesarios para poder entender y usar este software, que dispone de una fácil instalación y mucha documentación con todos los datos que se necesitan para poder ponerlo en funcionamiento en pocas horas.

Para poder comprender mejor todos los chequeos, se detallan los protocolos mencionados en el punto 8.1 [Manual de Administrador \(Nagios Core 4\)](#), donde se detallan la realización de los mismos y los tipos que se pueden establecer, según sistema operativo y protocolos de ejecución.

El protocolo NSCA se basa en el envío de la salida de la ejecución de scripts propios de Nagios o desarrollados por un administrador de la aplicación, que cumplan con el estándar de salida comprensible para Nagios. En el protocolo NRPE, desde el servidor de Nagios, se enviarán peticiones al servidor Unix/Linux, donde éste gracias al agente instalado en él, responderá con las comprobaciones oportunas que se hayan establecido en los scripts a los que se invoquen.

Nagios es recomendable instalarlo y administrarlo en un servidor con sistema operativo Linux/Unix, siendo la versión 6.4 de Centos la más adecuada, ya que dispone de completa configuración y una gran disponibilidad de acceso a servicios y procesos. Actualmente la gran comunidad de desarrolladores están mostrando pequeñas versiones de “Nagios Server” para instalar en servidor con sistema operativo Windows. Este tipo de sistema ha sido probado para determinar el alcance en el presente proyecto, pero se ha decidido realizar bajo sistema operativo Linux/Unix debido a su gran expansión y disponibilidad de documentación.

Este proyecto se ha pensado como primera puesta en marcha de una maqueta para la implantación de Nagios en el Departamento de Informática de la Universidad Carlos III de Madrid, para el control y supervisión de todos los equipos de las aulas de informática, y los servidores que mantienen actualmente el equipo de técnicos informáticos de la universidad, por lo que durante la presente memoria se van a realizar pruebas de casos críticos para mostrar el funcionamiento que tendría si se realizase dicha implantación en un entorno de producción.

La mayoría de los chequeos que se han previsto usar son chequeos ya predefinidos por Nagios, ya que los equipos de las aulas de informática, disponen de sistemas operativos Windows y/o Linux, y son completamente “monitorizables”.

Los chequeos que se mencionan, hacen referencia a la monitorización de los equipos de: la memoria libre, el uso de CPU, el espacio libre en los discos/particiones, el número de usuarios que entran al día, las conexiones que se realizan en la propia red, las conexiones realizadas a Internet, etc.

Para poder realizar este proyecto han sido necesarios multitud de conocimientos, empezando por los cursos que realicé antes de empezar la carrera, y el Grado Superior de Administración de Sistemas Informáticos, los cuales me ofrecieron muchos conocimientos para una visión global de los lenguajes de programación existentes y la iniciación a los mismos. Durante los años de carrera, todos estos lenguajes se han ido afianzando y muchas de las asignaturas cursadas, han sido base fundamental para el desarrollo, entendimiento y adecuación del proyecto y de la presente memoria.

Además de los mencionados conocimientos, es necesario el entendimiento y comprensión de la estructura de ficheros existente en un servidor con sistema operativo Linux/Unix [27], ya que Nagios instala complementos como MySQL y Apache, y es recomendable conocer su estructura y administración, para tener controlado todo el sistema y poder administrarlos en caso de tener algún problema.

En el capítulo de pruebas, gracias a todas las realizadas para el presente proyecto, se ha valorado la satisfacción de todos los técnicos y responsables con los que he tenido la oportunidad de trabajar, y se ha observado que Nagios, estando bien diseñado e implantado, es capaz de hacer más fácil el trabajo de todas las personas que en una organización o empresa, desempeñan un trabajo donde intervienen multitud de perfiles, conocimientos y ambiciones, pero siempre con la misma finalidad, la disponibilidad.

En el apartado de conclusiones, se han desarrollado un punto con los trabajos futuros, donde se comentan dos proyectos muy interesantes como son NagVis y NagiosQL, que facilitan la administración de Nagios y hacen que los propios técnicos y responsables, colaboren en la administración y mantenimiento del sistema. NagiosQL es una interfaz para interactuar con los ficheros de configuración de Nagios y así poder agregar alertas nuevas, sin tener que disponer de grandes conocimientos. NagVis, es una herramienta muy práctica

para poder ofrecer las vistas de la red y comprobar los puntos de fallo, con todos los mapas generados.

Destacar la importancia de las asignaturas donde se han aprendido lenguajes de programación, como C y HTML, ya que son las bases fundamentales para poder administrar y entender el funcionamiento de una herramienta de monitorización como Nagios, de la que casi todo el código que se necesita realizar manualmente es en estos dos lenguajes. El resto de lenguajes que se han utilizado, como por ejemplo PowerShell, han sido aprendidos durante mi vida laboral, donde este tipo de lenguajes se han convertido en herramienta imprescindible para monitorizar sistemas operativos Windows.

Después de la realización del presente TFG es necesario comentar que gracias a las asignaturas realizadas en la Universidad Carlos III de Madrid y de los cursos que he desarrollado de forma personal, se ha demostrado el conocimiento amplio, tanto de un sistema operativo Unix/Linux como del sistema de monitorización de Nagios, con todas las pruebas de los complementos para desarrollar en proyectos futuros, y las pruebas que demuestran que Nagios podría considerarse uno de los mejores sistemas de monitorización del mercado, habiendo sido investigada en profundidad su versión de software libre.

Palabras clave: Nagios, herramienta, monitorización, software libre, Windows, Unix, Linux, disponibilidad, accesibilidad, rendimiento, desarrollo, NRPE, NSCA, activo, pasivo, TIC, alertas, chequeos.

## **Abstract**

This memory is part of the Final Degree Project by Alexandra Moreno Calabozo for the Carlos III University of Madrid. The idea was born following my professional career in a services monitoring and management project within an Information and Communications Technology (ITC by its Spanish acronym) company, where the main goal is to exploit an interesting idea, with properly defined goals, and with enough level of knowledge to get a chance to expose a monitoring tool as powerful as Nagios.

The project's goal is to demonstrate all the advantages to use a monitoring tool, providing usability, availability, continuity, performance and normalcy, by integrating this tool in a network which pretends to provide instant information regarding issues or failures in devices, servers and services with a certain level of criticality, being these very important to provide a service to users and clients, and being able to measure the user satisfaction accessing those.

To begin with the project it was determined a configuration design to implement the monitoring tool called Nagios [11], following the continuity lines of the service, and maintaining the goal to minimize the times between a service disruption and its recovery, for those provided by the University from the Department of Informatics.

Nagios is an advanced monitoring tool, intuitive and very powerful, with a wide number of possible pre-defined checks within the software installation, but with a really big open source platform and the chance to develop any desired monitoring scripts just by knowing any programming language which can report data using an understandable way for it.

There is a possibility to monitor servers with multiple different operative systems, switches and any kind of devices with a reachable IP address within the network where Nagios is connected. On top of monitoring basic checks like PING, CPU, drives, memory and swap, it is possible to monitor databases, files, web access, and more being all of them customizable. Every advanced check must be developed and customized to be able to be executed in the servers and/or clients.

For a better software administration, it is recommended to have knowledge of multiple programming languages, among them, Java, HTML, PowerShell and Linux/Unix (Bash/Shell) programming, and having the base programming knowledge in the University, a monitoring system like Nagios is easily administrable.

Using some of the programming languages mentioned in the previous paragraph, it is possible to monitor all the services and processes of any server from a small, medium or big network.

In the Microsoft family servers and clients, from Windows XP to Windows Server 2012, it is possible to control any kind of process and service, associated or not to the first, by using the Nagios agent installed in every server and client with these operative systems. In the other hand, with the Unix/Linux operative system family, it is required to develop scripts in order to monitor everything using commands, as in order to monitor a service or process, it is required to execute commands which are not defined in a precompiled script within the initial installation.

In the case of Windows systems, every check can be performed in active mode, as Nagios can be in touch with its agent, and use the request-response mode, to show the results in the WebUI (graphic interface) and to be presented in reports and tables. With the Unix/Linux systems, there are multiple ways to implement checks, as they can be actives or passives, being the active checks established using the protocol NRPE, using the communication port 5666/TCP between the Nagios server and the client, and the passives using NSCA, using the port 5667/TCP enabled also in both servers.

During this current memory it will be written all necessary sections to be able to understand and use this software, which has an easy and a lot of documentation with all the required data to make it running in just a few hours.

To better understand all the checks, it have been detailed the protocols mentioned in the section 8.1 “Manual de Administrador (Nagios Core 4)”, where the checks implementation is described as the kind of checks that can be established, based on operative system and execution protocols.

The protocol NSCA it is based in the delivery of the output of the Nagios own or custom application’s administrator scripts execution, which must fulfill the standard output understandable for Nagios. In the protocol NRPE, from the Nagios server, it will be sent different requests to the Unix/Linux server, where thanks to this previously installed agent, it will reply with the opportune checks established within the invoked scripts.

It is recommended to install and administrate Nagios in a Linux/Unix server, being the CentOS version 6.4 the most optimal, as it has the complete configuration and a great access availability to services and processes. Currently the big Nagios developers’ community are providing a custom tiny version of “Nagios Server” to install it in a Windows Server. This kind of system has been tested to determine the reach of the current project, but it has been decided to use a Linux/Unix server due to its wide expansion and documentation availability.

This project has been developed as an initial laboratory start up to implement Nagios within the University Carlos III of Madrid Department of Informatics, for the control and supervision of all the devices in the informatics rooms, and the server that currently maintained by the university informatics technical teams, so during this memory there will be performed tests of critical cases to show the behavior expected if the implementation would be a production environment.

Most of the checks that have been selected are predefined within Nagios, as the devices in the informatics rooms, runs Windows or Linux and are completely “monitorable”.

The mentioned checks, reference the monitoring of: free memory, CPU usage, free space in disks/partitions, number of daily users, connections performed in the network, connections towards internet, etc.

In order to perform this project, have been required multiple knowledge, beginning with different training courses I attended before I began with the degree, and the “Grado Superior de Administracion de Sistemas Informaticos”, which provided me the “know how” and a global vision about the existing programming languages and initiation of them. During the degree years, all these languages have been strengthening and a lot of the university courses, have been fundamental for the development, understanding and customization of the project and current memory.

Besides the previously mentioned knowledge, it is required the understanding and comprehension of the existing file structure in a server running Linux/Unix [27], as Nagios installs complements like MySQL and Apache, and it is highly recommended to know its structure and administration, to have the system under control and be able to administer them in case of an issue.

Within the testing section, based in all the tests performed for the current project, it has been measured all the technical and manager parties whom I have the opportunity to work with, and it has been observed that Nagios, being properly designed and implemented, it is able to make the work easier for every person that within an organization or enterprise, develop a work where multiple profiles, knowledge and ambitions intervene, but always with the same goal, the availability.

In the conclusions section, it has been developed a subsection with future developments, where there are described two interesting projects like NagVis and NagiosQL, which facilitate the Nagios administration and achieves the collaboration between the managers and technicians with the administration and the maintenance of the system. NagiosQL is an interface to interact with the Nagios configuration files so new alerts can be added, without the need to have a deep knowledge about it. NagVis, is a really useful tool which offers a multiple network views and check different points of failure with the generated network maps.

I must highlight the importance of the university courses where programming languages have been learnt, like C and HTML, as they are the fundamental basis to be able to administer and understand the working of a monitoring tool like Nagios, which almost all the code required must be performed manually in those two languages. The remaining languages used, like PowerShell, have been learnt during my professional career, where this kind of languages have been an essential tool to monitor Windows operative systems.

After creating this “TFG” it will be good to say that thanks to the subjects performed in the University Carlos III from Madrid and the training courses that I have attended outside college, it has been demonstrated the deep knowledge

of a Linux/Unix operative system as well as the Nagios monitoring system, with all the tests of the add-ons to develop future projects, and the tests to demonstrate that Nagios could be considered one of the best monitoring systems of the market, having deeply investigated its open source version.

Key words: Nagios, tool, monitoring, open source, Windows, Unix, Linux, availability, accessibility, performance, development, NRPE, NSCA, active, passive, TIC [TCI], alerts, checks.

## Índice general

Agradecimientos.....	4
Resumen.....	5
Abstract.....	9
1. Introducción.....	22
1.1. Motivación.....	22
1.2. Objetivos.....	23
1.3. Organización del resto del documento.....	25
1.4. Motivation.....	28
1.5. Objective.....	29
1.6. Rest of the document's organization.....	30
2. Estado del arte.....	34
2.1. Introducción.....	34
2.2. Análisis de aplicaciones.....	36
2.2.1. Solución 1: Cacti.....	37
2.2.2. Solución 2: Zenoss.....	38
2.2.3. Solución 3: Pandora FMS.....	39
2.2.4. Solución 4: Zabbix.....	41
2.3. Tabla comparativa con las distintas soluciones y la propuesta.....	42
2.3.1. A nivel sistema operativo.....	42
2.3.2. A nivel aplicación.....	43
2.3.3. A nivel complementos.....	43
3. Análisis, diseño e implantación.....	46
3.1. Introducción.....	46
3.1.1. Marco regulador.....	48
3.2. Análisis.....	49
3.2.1. Chequeos activos.....	51
3.2.2. Chequeos pasivos.....	51
3.2.3. Requisitos previos.....	52
3.3. Diseño.....	53
3.3.1. Diagramas de casos de uso.....	57
3.3.1.1. Diagrama de descomposición.....	58
3.3.1.1.1. Infraestructura (Definición).....	59
3.3.1.1.2. Infraestructura (Modificación).....	65

3.3.1.1.3. Infraestructura (Eliminación).....	70
3.3.1.1.4. Notificación.....	75
3.3.1.1.5. Visualización.....	78
3.3.2. Especificación de requisitos.....	80
3.3.2.1. Requisitos funcionales.....	81
3.3.2.2. Requisitos no funcionales.....	86
3.3.3. Matriz de trazabilidad.....	88
3.3.4. Diagramas de navegación.....	89
3.3.5. Esquemas de base de datos.....	90
3.4. Implantación.....	91
4. Evaluación.....	96
4.1. Pruebas de verificación.....	96
4.2. Pruebas de rendimiento.....	102
5. Planificación y presupuesto.....	109
5.1. Planificación.....	109
5.2. Presupuesto.....	111
5.2.1. Gasto de personal.....	111
5.2.2. Gasto de material.....	112
5.2.3. Resumen de costes.....	113
6. Conclusiones y trabajos futuros.....	115
6.1. Conclusiones generales.....	115
6.1.1. Conclusiones sobre el sistema puesto en marcha.....	116
6.1.2. Conclusiones sobre la planificación usada.....	116
6.1.3. Conclusiones personales.....	117
6.2. General conclusions.....	119
6.2.1. Conclusions about the live system.....	119
6.2.2. Conclusions about the used planning.....	120
6.2.3. Personal conclusions.....	120
6.3. Trabajos futuros.....	123
6.3.1. NagVis.....	123
6.3.2. NagiosQL.....	127
6.3.3. Otros ámbitos.....	130
7. Apéndice: Glosario de términos.....	132
8. Apéndice: Manuales.....	135
8.1. Manual de Administrador (Nagios Core 4).....	135
8.1.1. Instalar Sistema Operativo.....	135
8.1.2. Instalación dependencias necesarias.....	136

8.1.3.	Usuarios y grupos .....	137
8.1.4.	Descarga e instalación de Nagios y Plugins .....	137
8.1.5.	Configuración de Nagios y Plugins .....	143
8.1.6.	Chequeos por defecto en la instalación .....	150
8.1.7.	Instalación agentes de Nagios .....	150
8.1.7.1.	Instalación agentes de Nagios Windows .....	151
8.1.7.2.	Instalación agentes de Nagios Linux/Unix .....	152
8.1.7.2.1.	Protocolo NSCA.....	152
8.1.7.2.2.	Protocolo NRPE.....	154
8.1.7.2.3.	Protocolo SNMP .....	156
8.1.8.	Definición de grupos de contactos .....	156
8.1.9.	Envío y recepción de información .....	157
8.1.10.	Caso práctico de monitorización .....	158
8.2.	Manual de Usuario .....	165
8.3.	Scripts generados .....	170
8.3.1.	Script para chequear espacio libre en Unix.....	170
8.3.2.	Script para chequear memoria libre en Unix .....	170
8.3.3.	Script para chequear servicio en Unix .....	171
8.3.4.	Script para chequear las bases de datos de Exchange .....	171
8.3.5.	Script para chequear el estado de un Ironport.....	172
8.3.6.	Script para chequear servicio Squid en Proxy .....	175
8.3.7.	Chequear el servicio DHCP en Windows.....	176
8.3.8.	Chequear un disco en Windows .....	176
8.3.9.	Chequear procesos de Antivirus en Windows.....	176
9.	Bibliografía.....	178

## Índice de tablas

Tabla 1.- Comparativa de las distintas soluciones a nivel sistema operativo ...	42
Tabla 2.- Comparativa de las distintas soluciones a nivel aplicación .....	43
Tabla 3.- Comparativa de las distintas soluciones a nivel complementos .....	43
Tabla 4.- Tipos de protocolos usados en Nagios para monitorizar .....	49
Tabla 5.- Tabla de ejemplo de un Caso de uso .....	57
Tabla 6.- CU-1: Definir grupo servidores .....	60
Tabla 7.- CU-2: Definir grupo servicios .....	60
Tabla 8.- CU-3: Definir servidor .....	60
Tabla 9.- CU-4: Definir chequeo .....	61
Tabla 10.- CU-5: Definir período .....	61
Tabla 11.- CU-6: Definir servicio .....	61
Tabla 12.- CU-7: Definir dependencia .....	62
Tabla 13.- CU-8: Definir contacto .....	62
Tabla 14.- CU-9: Definir grupo contactos .....	62
Tabla 15.- CU-10: Definir notificación .....	63
Tabla 16.- CU-11: Definir sistema escalado .....	63
Tabla 17.- CU-12: Definir vista .....	63
Tabla 18.- CU-13: Definir informe .....	64
Tabla 19.- CU-14: Modificar grupo servidores .....	66
Tabla 20.- CU-15: Modificar grupo servicios .....	66
Tabla 21.- CU-16: Modificar servidor .....	66
Tabla 22.- CU-17: Modificar chequeo .....	66
Tabla 23.- CU-18: Modificar período .....	67
Tabla 24.- CU-19: Modificar servicio .....	67
Tabla 25.- CU-20: Modificar dependencia .....	67
Tabla 26.- CU-21: Modificar contacto .....	67
Tabla 27.- CU-22: Modificar grupo contactos .....	68
Tabla 28.- CU-23: Modificar notificación .....	68
Tabla 29.- CU-24: Modificar sistema de escalado .....	68
Tabla 30.- CU-25: Modificar vista .....	68
Tabla 31.- CU-26: Modificar informe .....	69
Tabla 32.- CU-27: Eliminar grupo de servidores .....	71
Tabla 33.- CU-28: Eliminar grupo de servicios .....	71
Tabla 34.- CU-29: Eliminar servidor .....	71
Tabla 35.- CU-30: Eliminar chequeo .....	72
Tabla 36.- CU-31: Eliminar período .....	72
Tabla 37.- CU-32: Eliminar servicio .....	72
Tabla 38.- CU-33: Eliminar dependencia .....	72
Tabla 39.- CU-34: Eliminar contacto .....	73
Tabla 40.- CU-35: Eliminar grupo contactos .....	73
Tabla 41.- CU-36: Eliminar notificación .....	73
Tabla 42.- CU-37: Eliminar sistema escalado .....	73
Tabla 43.- CU-38: Eliminar vista .....	74
Tabla 44.- CU-39: Eliminar informe .....	74
Tabla 45.- CU-40: Configurar correo electrónico .....	76
Tabla 46.- CU-41: Configurar alertas .....	76

Tabla 47.- CU-42: Enviar alertas .....	76
Tabla 48.- CU-43: Recibir alertas .....	77
Tabla 49.- CU-44: Configurar informes .....	77
Tabla 50.- CU-45: Configurar vistas .....	77
Tabla 51.- CU-46: Configurar comentarios.....	77
Tabla 52.- CU-47: Visualizar alertas.....	78
Tabla 53.- CU-48: Visualizar informes.....	79
Tabla 54.- CU-49: Visualizar vistas .....	79
Tabla 55.- CU-50: Visualizar comentarios .....	79
Tabla 56.- Tabla de ejemplo de Requisitos funciones y no funcionales .....	80
Tabla 57.- RF-01: Introducir dirección IP.....	81
Tabla 58.- RF-02: Introducir datos de inicio de sesión .....	81
Tabla 59.- RF-03: Interfaz de acceso remoto a Nagios.....	81
Tabla 60.- RF-04: Acceso al servidor de Nagios como administrador.....	81
Tabla 61.- RF-05: Pantalla de estado de las alertas .....	82
Tabla 62.- RF-06: Consulta datos de históricos .....	82
Tabla 63.- RF-07: Consulta datos de históricos gráficamente.....	82
Tabla 64.- RF-08: Opcion de añadir nuevo servidor a monitorizar .....	82
Tabla 65.- RF-09: Opción de añadir nuevo servicio a monitorizar.....	83
Tabla 66.- RF-10: Opción de desactivar monitorización sobre servidor .....	83
Tabla 67.- RF-11: Opción de desactivar monitorización sobre servicio.....	83
Tabla 68.- RF-12: Opción de activar monitorización sobre servidor.....	83
Tabla 69.- RF-13: Opción de activar monitorización sobre servicio .....	84
Tabla 70.- RF-14: Opción de configurar grupos de servidores.....	84
Tabla 71.- RF-15: Opción de configurar grupos de servicios .....	84
Tabla 72.- RF-16: Opción de configurar alertas .....	84
Tabla 73.- RF-17: Opción de desactivar alertas .....	85
Tabla 74.- RF-18: Opción de activar alertas.....	85
Tabla 75.- RF-19: Opción de definir notificación .....	85
Tabla 76.- RF-20: Opción de enviar notificaciones por correo electrónico .....	85
Tabla 77.- RNF-01: Acceso a Nagios.....	86
Tabla 78.- RNF-02: Acceso de Nagios a toda la red .....	86
Tabla 79.- RNF-03: Acceso a servidores monitorizados .....	86
Tabla 80.- RNF-04: Agente de Nagios para sistemas Windows.....	86
Tabla 81.- RNF-05: Agente de Nagios para sistemas Linux/Unix .....	87
Tabla 82.- RNF-06: Agente de Nagios en servidores.....	87
Tabla 83.- RNF-07: Backup de ficheros de configuración de Nagios .....	87
Tabla 84.- RNF-08: Identificación de alertas .....	87
Tabla 85.- Tabla de ejemplo de Pruebas de verificación.....	97
Tabla 86.- PV-01: Configura el histórico de alertas de un elemento .....	98
Tabla 87.- PV-02: Consulta el histórico de alertas gráficamente.....	98
Tabla 88.- PV-03: Agregar un servidor a monitorizar .....	99
Tabla 89.- PV-04: Agregar un servicio a monitorizar .....	99
Tabla 90.- PV-05: Opción de configurar alertas .....	100
Tabla 91.- PV-06: Opción de configurar notificaciones .....	100
Tabla 92.- PV-07: Opción de configurar monitorización .....	101
Tabla 93.- PV-08: Correcto funcionamiento de Nagios .....	101
Tabla 94.- Ejemplo de alerta con Nagios por nivel de CPU alto en localhost. ....	103
Tabla 95.- Encuesta de satisfacción a Técnico 1 sobre Nagios .....	105
Tabla 96.- Encuesta de satisfacción a Técnico 2 sobre Nagios .....	105

Tabla 97.- Encuesta de satisfacción a Técnico 3 sobre Nagios .....	106
Tabla 98.- Encuesta de satisfacción a Responsable 1 sobre Nagios.....	106
Tabla 99.- Encuesta de satisfacción a Responsable 2 sobre Nagios.....	107
Tabla 94.- Coste total de personal del proyecto .....	111
Tabla 95.- Coste total de material del proyecto .....	112
Tabla 96.- Coste total proyecto .....	113
Tabla 97.- Script para chequear espacio libre en Unix.....	170
Tabla 98.- Script para chequear memoria libre en Unix .....	171
Tabla 99.- Script para chequear servicio en Unix.....	171
Tabla 100.- Script para chequear la base de datos Exchange .....	172
Tabla 101.- Script para chequear estado de un Ironport .....	174
Tabla 102.- Script para chequear servicio Squid en Proxy.....	175
Tabla 103.- Chequear DHCP en un Windows .....	176
Tabla 104.- Chequear disco en un Windows.....	176
Tabla 105.- Chequear proceso Antivirus en Windows.....	176

## Índice de ilustraciones

Ilustración 1.- Logo Nagios.....	34
Ilustración 2.- Precios de Nagios XI .....	35
Ilustración 3.- Logo Cacti.....	37
Ilustración 4.- Ejemplo gráfico Cacti .....	37
Ilustración 5.- Logo Zenoss .....	38
Ilustración 6.- Ejemplo gráfico Zenoss .....	38
Ilustración 7.- Logo Pandora FMS.....	39
Ilustración 8.- Ejemplo gráfico Pandora FMS .....	39
Ilustración 9.- Ejemplo gráfico Pandora FMS (gráficos) .....	40
Ilustración 10.- Logo Zabbix .....	41
Ilustración 11.- Ejemplo gráfico Zabbix .....	41
Ilustración 12.- Ciclo de vida del software aplicable a Nagios .....	47
Ilustración 13.- Lógica de la aplicación de Nagios para monitorizar .....	48
Ilustración 14.- Ejemplo de funcionamiento Nagios completo .....	50
Ilustración 15.- Arquitectura del sistema de Nagios en capas.....	52
Ilustración 16.- Lógica del sistema de Nagios en capas.....	53
Ilustración 17.- Diagrama de flujo de datos en Nagios .....	54
Ilustración 18.- Frontal Web "Exfoliation" para Nagios .....	55
Ilustración 19.- Frontal Web "Vautour" para Nagios .....	55
Ilustración 20.- Frontal Web "Arana" para Nagios .....	56
Ilustración 21.- Diagrama de descomposición de Nagios diseñada .....	58
Ilustración 22.- Diagrama de casos de uso de Infraestructura (Definición) .....	59
Ilustración 23.- Diagrama de casos de uso de Infraestructura (Modificación) ..	65
Ilustración 24.- Diagrama de casos de uso de Infraestructura (Eliminación)....	70
Ilustración 25.- Diagrama de casos de uso de Notificación .....	75
Ilustración 26.- Diagrama de casos de uso de Visualización.....	78
Ilustración 27.- Matriz de trazabilidad de casos de uso y requisitos.....	88
Ilustración 28.- Diagrama de navegación interna en servidor Nagios .....	89
Ilustración 29.- Estructura base de datos MySQL para Nagios .....	90
Ilustración 30.- Ficheros de configuración de Nagios en servidor Linux.....	91
Ilustración 31.- Funcionamiento de Nagios (estructura lógica del sistema).....	92
Ilustración 32.- Caso real de monitorización con Nagios.....	93
Ilustración 33.- Funcionamiento en pasos/niveles de Nagios.....	94
Ilustración 34.- Pantalla inicial de Nagios con inicio correcto en web.....	96
Ilustración 35.- Error de inicio de proceso de Nagios web .....	97
Ilustración 36.- Ejemplo de chequeos de Nagios en la web .....	102
Ilustración 37.- Ejemplo de gráfica de CPU en servidor .....	103
Ilustración 38.- Ejemplo de gráficas de servicios monitorizados en host.....	104
Ilustración 39.- Análisis de datos de encuestas.....	107
Ilustración 40.- Planificación proyecto en gráfico .....	109
Ilustración 41.- Planificación proyecto (Diagrama de Gantt) .....	110
Ilustración 42.- Coste total proyecto en gráfica .....	113
Ilustración 43.- Logo NagVis .....	123
Ilustración 44.- Ejemplo NagVis integrado con Nagios de servidores .....	124
Ilustración 45.- Ejemplo NagVis integrado (servidores y servicios) .....	124
Ilustración 46.- Ejemplo NagVis integrado con Nagios en imagen .....	125

Ilustración 47.- Ejemplo NagVis integrado (imagen detallado) .....	125
Ilustración 48.- Fotografía integrada en NagVis con chequeos de Nagios .....	126
Ilustración 49.- Logo de NagiosQL .....	127
Ilustración 50.- Instalación NagiosQL .....	127
Ilustración 51.- Acceso a NagiosQL .....	128
Ilustración 52.- Configuración principal NagiosQL .....	128
Ilustración 53.- Importación de ficheros a NagiosQL .....	128
Ilustración 54.- Consultas de monitorización en NagiosQL .....	129
Ilustración 55.- Definición hosts en NagiosQL .....	129
Ilustración 56.- Estructura ficheros Linux/Unix .....	136
Ilustración 57.- Primer inicio de la interfaz web de Nagios .....	138
Ilustración 58.- Primera vista localhost Nagios .....	140
Ilustración 59.- Vistas de iconos de gráficas en Nagios .....	141
Ilustración 60.- Previsualización de gráficas en Nagios .....	142
Ilustración 61.- Gráficas por PNP4Nagios .....	142
Ilustración 62.- Fichero de configuración de Nagios .....	143
Ilustración 63.- Ruta de vistas de Nagios .....	143
Ilustración 64.- Comprobación Nagios .....	143
Ilustración 65.- Compilación correcta de Nagios .....	144
Ilustración 66.- Compilación incorrecta de Nagios .....	144
Ilustración 67.- Autorizar acceso web a Nagios .....	145
Ilustración 68.- Acceso a la interfaz web de Nagios .....	146
Ilustración 69.- Configuración fichero de log en Nagios .....	146
Ilustración 70.- Definición de ficheros de configuración Nagios .....	147
Ilustración 71.- Arranque correcto de puerto 5667 (NSCA) .....	148
Ilustración 72.- Arranque correcto 2 de puerto 5667 (NSCA) .....	149
Ilustración 73.- Instalación NSClient++ en Windows .....	151
Ilustración 74.- Funcionamiento NSCA .....	152
Ilustración 75.- Funcionamiento NRPE .....	154
Ilustración 76.- Chequeos reales en servidor localhost .....	155
Ilustración 77.- Funcionamiento SNMP .....	156
Ilustración 78.- Recepción y envío de datos mediante Firewall .....	158
Ilustración 79.- Vista HostGroups en Nagios .....	159
Ilustración 80.- Agregar chequeos a Nagios de NSCA .....	162
Ilustración 81.- Chequeos pasivos NSCA recibidos por Nagios .....	162
Ilustración 82.- Chequeos NSCA y NRPE recibidos por Nagios .....	163
Ilustración 83.- Chequeos activos recibidos por Nagios .....	164
Ilustración 84.- Login en interfaz web de Nagios .....	165
Ilustración 85.- Pantalla inicial de interfaz web de Nagios .....	165
Ilustración 86.- Acceso a Tactical Overview .....	166
Ilustración 87.- Vista de Host .....	166
Ilustración 88.- Vista de Services .....	167
Ilustración 89.- Información accesible por chequeo .....	167
Ilustración 90.- Resumen estados en Nagios .....	168
Ilustración 91.- Alertas en un preciso instante .....	168
Ilustración 92.- Menú administración Nagios .....	168
Ilustración 93.- Availability servidores .....	169
Ilustración 94.- Trend de servicio en Nagios .....	169

# Capítulo 1: Introducción

---

## 1. Introducción

En este primer capítulo se van a explicar las motivaciones, el contexto y los objetivos planteados para poder desarrollar este proyecto, junto con las etapas que han sido necesarias para la implantación de dicho sistema.

### 1.1. Motivación

Las infraestructuras de redes, cada vez más, soportan multitud de aplicaciones y servicios estratégicos de organizaciones, siendo necesario incluir herramientas de análisis y monitoreo para garantizar la disponibilidad y accesibilidad de los servicios, tanto de forma interna como para clientes externos.

La gestión empresarial y la necesidad de prestación de servicios a pequeñas, grandes empresas y organismos, hoy en día, dependen en casi todo de equipamiento informático. El objetivo de estas empresas es, principalmente, la disponibilidad de los servicios que ofrecen, donde se ubican los servidores de éstas.

La monitorización de sistemas y servicios se ha convertido en una labor muy importante y sobre todo de carácter proactivo, con el objetivo de evitar el mayor número posible de problemas y estar prevenido de los mismos. Monitorizar es un elemento de gran utilidad ya que permiten, además de lo comentado, obtener estadísticas, informes, y datos sobre disponibilidad para la mejora continua de los sistemas, y hace que se pueda tener una gestión de problemas reactiva y proactiva.

En la actualidad, se dispone de una gran comunidad de desarrolladores a través de la cual disponemos un amplio paquete de actualizaciones. Aunque existen multitud de alternativas libres, Nagios es una de las herramientas más completas con muchas funcionalidades y un amplio número de chequeos para poder realizar a diferentes sistemas.

La herramienta de monitorización Nagios es un software libre que proporciona una gran utilidad y aporta un amplio control de servidores con diferentes sistemas operativos, Routers, Switches, dispositivos, etc. Dispone de un gran gestor de informes, estadísticas y reportes de eventos, que hacen que estemos alertados en todo momento del estado de los mismos. Para conseguir una completa monitorización, es necesario disponer de un personal operativo en las horas de servicio, ya que Nagios es una herramienta que informa de estados y siempre deberá de haber contratados diferentes técnicos que reaccionen a cualquier aviso inesperado. Siempre proactividad.

Es posible obtener información de ellos elaborando una configuración de Nagios para cada caso, por medio de agentes instalados en los servidores, o usando el protocolo SNMP (Simple Network Management Protocol) que nos permitirá obtener información completa de los sistemas monitorizados.

En este Trabajo Final de Grado (TFG) se propone diseñar e implantar la herramienta de monitorización Nagios, para demostrar la complejidad del mismo, y ser utilizado para monitorizar servidores, equipos, redes y aplicaciones, mostrando las ventajas de estar escrito en varios lenguajes de programación y de ser Software Libre, con la finalidad de ser usado para las aulas y servidores del Departamento de Informática de la Universidad Carlos III de Madrid.

Todas estas motivaciones hacen que este TFG, sea un proyecto muy interesante, debido a que en el mundo de las TIC cada día se hace más indispensable la proactividad, la mejora continua y la supervisión de todos los sistemas. Por todo lo comentado, es necesario disponer de un sistema de monitorización que permita vigilar y avisar de los sucesos y problemas que presente un servicio, para poder actuar con mayor rapidez y disponer, a su vez, de informes que nos transmitan la disponibilidad del sistema.

## **1.2. Objetivos**

El objetivo principal de este proyecto es demostrar que ayuda a mejorar la usabilidad y disponibilidad del sistema de monitorización Nagios. Además de instalar, configurar, implementar e implantar la aplicación, se mostrará su funcionamiento y las nuevas formas de controlar todos los sistemas disponibles en las TIC.

Como objetivos específicos del proyecto destacar la importancia de:

- Detectar el uso de recursos y el dimensionamiento de los mismos en los equipos monitorizados.
- Verificar la infraestructura de red de los equipos.
- Analizar la eficiencia de los sistemas de los que dispone la red.
- Mejorar la calidad de los servicios informáticos disponibles.
- Reducir el coste de los servicios usados.

Existen muchas razones por las que monitorizar automáticamente servidores, pero la principal es la económica, ya que el coste de un sistema dependerá de la disponibilidad, redundancia, escalabilidad y eficiencia del mismo [32].

Los objetivos marcados en los anteriores puntos, son determinados por las necesidades de mantener disponible el sistema de atención y recuperación de los servicios que una empresa y organismo proporciona a una serie de clientes o a otras empresas, y que dependen de ellos con multitud de acciones y/o de presupuestos.

Un ejemplo crítico de la disponibilidad de un servicio podría ser el caso de un servidor como “guernika” que da servicio a muchos alumnos de informática, para multitud de prácticas de diferentes asignaturas del departamento, y que tuviese una caída, y no se detectase hasta que un alumno o profesor comprobase que está inaccesible. Durante los años de carrera que he tenido que conectarme a “guernika” se han sufrido colapsos de sistema, que sólo han sido detectados una vez ha fallado, sin detectar el incremento del uso de memoria o de carga en la CPU.

Otro ejemplo crítico serían los equipos de las aulas de informática, que para conocer el rendimiento de los mismos y poder analizar su usabilidad y disponibilidad, sería recomendable que estuviesen monitorizados. Estos equipos durante prácticamente todo el año, tienen un uso muy elevado y una exigencia de rendimiento muy grande por parte de las clases que se imparten en ellos, y de las horas que se usan para prácticas fuera de horario lectivo, ocasionando en muchos momentos, problemas de rendimiento, ya que tienen mucha carga a lo largo del año. Una ventaja en el caso de ser monitorizados, sería enfocada a poder predecir los problemas y errores que podrían ocasionarse, antes de que puedan tener problemas de discos (espacio o rendimiento), de memoria, de CPU o de red. Estos equipos están expuestos diariamente a intrusiones de virus, descargas fraudulentas y problemas de caídas inesperadas, por lo que Nagios propocionaría una gran ventaja para la optimización de estas salas de informática.

Mejorar la calidad de los servicios, es hoy en día uno de los principales objetivos, para poder ofrecer servicios muy competitivos e intentar lograr objetivos económicos y sociales, para poder ampliar la satisfacción de los usuarios que hagan uso de los mismos.

### **1.3. Organización del resto del documento**

En esta sección vamos a detallar cuál ha sido la organización del trabajo realizado para la elaboración de esta documentación, comentando las tareas y la planificación que se ha seguido para la realización de las mismas.

En primer lugar, se realizó un análisis de las características importantes a detallar, así como de todos los puntos necesarios para mostrar la aplicación en su mayor amplitud.

Ya que dicho sistema no necesita tener un gran uso de recursos, ha sido viable instalar, configurar y administrar un software como Nagios para este proyecto.

A continuación, se expone un resumen del contenido de cada uno de los capítulos que se desarrollaran a lo largo del trabajo del proyecto.

#### **Capítulo 1: Introducción**

---

En este capítulo, se describe el contexto del proyecto, dando a conocer la motivación y los objetivos principales y específicos necesarios para su realización. En él, se describen las ideas y las soluciones que van a satisfacer las necesidades del problema que nos reportan cada día todos los sistemas informáticos.

#### **Capítulo 2: Estado del arte**

---

Este capítulo ofrece una amplia visión sobre Nagios, detallando las comparativas de este software con otros que disponen de la misma funcionalidad pero con más limitaciones. Se presentan cuatro posibles alternativas, y se comparan con Nagios para determinar que éste obtiene una mejor valoración a varios niveles.

#### **Capítulo 3: Análisis, diseño e implantación del sistema**

---

Este capítulo es el más amplio de la presente memoria, donde se detallan todos los datos de Nagios a bajo nivel.

En el subcapítulo de Análisis se detallan los datos generales del funcionamiento de Nagios, y los requisitos previos, antes de investigar más a fondo en él.

En el subcapítulo de Diseño, se explican todas las herramientas y tecnologías que se han seleccionado para construir el sistema y se exponen todos los diagramas de casos de uso, los escenarios, los diagramas de navegación, el esquema de base de datos y los requisitos funcionales y no funcionales del software.

En el subcapítulo de Implantación se exponen los pasos que se han seguido para desplegar, instalar, configurar y desarrollar el software de Nagios, adjuntando un [Manual de Administrador \(Nagios Core 4\)](#) con multitud de detalles para poder administrar este software en cualquier organización.

#### **Capítulo 4: Evaluación**

---

En el capítulo, se ha decidido dividir en dos partes las pruebas para poder realizar una evaluación más completa. Se exponen las pruebas de verificación y las de rendimiento, donde en las primeras se demuestra que todos los requisitos del software son aceptados y verificados, y en las segundas se evalúan los datos obtenidos tanto de satisfacción de los técnicos y responsables, que reciben en pocos segundos información del estado de sus servidores, como de la mejora del servicio una vez que un servidor y sus servicios están siendo monitorizados con Nagios.

#### **Capítulo 5: Planificación y presupuesto**

---

En este capítulo, se van a determinar los costes del proyecto teniendo en cuenta los datos de la planificación de tiempos que se han necesitado para desarrollar e implantar el mismo.

Además se va a mostrar un presupuesto de una instalación, configuración y mantenimiento en una empresa con dicho servicio de monitorización, donde un administrador de Nagios mantiene día a día las alertas actualizadas e investiga en nuevas formas de mejorar Nagios.

#### **Capítulo 6: Conclusiones y trabajos futuros**

---

Para finalizar se van a presentar las conclusiones finales del trabajo y las expectativas cumplidas una vez realizado todo el trabajo y haber redactado la memoria, a través de la cual se presenta todo el esfuerzo realizado y los objetivos cumplidos durante un período aproximado de 8 meses.

#### **Capítulo 7: Apéndice: Glosario de términos**

---

En este capítulo, se van a definir todos los términos y siglas que se empleen para redactar esta memoria, y que serán explicados en la mayor medida de lo posible, para el entendimiento de todos, siendo o no del sector de la Informática.

### **Capítulo 8: Apéndice: Manuales**

---

Para finalizar, se incluyen dos manuales, uno de administración y otro de usuario, junto con todos los scripts de pruebas implementados para comprobar que el sistema de monitorización de Nagios es capaz de monitorizar multitud de servicios y procesos, habiendo realizado los scripts personalmente en diferentes lenguajes de programación.

### **Capítulo 9: Bibliografía**

---

En este punto, se redactarán todas las páginas web y trabajos consultados para la realización del presente Trabajo Final de Carrera.

## 1.4. Motivation

The network infrastructures increasingly support multiple company applications and strategic services, being required to include analysis and monitoring tools to grant the availability and accessibility of the services both internally and for the clients.

The business management and the need of provide services to small and big companies as well as different organisms, nowadays, depends mainly in the informatics equipment. The goal for those companies is, mainly, the availability of the services offered, where the servers are installed.

Monitoring of systems and services have become a really important task mainly for its proactive character, with the goal to avoid the most possible number of issues and to be prepared for them. Monitoring is an element of great utility as it allows, on top of the previously mentioned, to obtain statistics, reports, and data about availability for the continuous improvement of the systems, making possible a problem management both reactive and proactive.

Currently, it exists a big community of developers through which we have a wide updates package. Even if there are multiple other open source choices, Nagios is the one the most comprehensive more functionalities and a wide range of checks to perform.

The monitoring tool Nagios is an open source software which provides a great utility and a wide control of servers with different operative systems, routers, switches, etc. It has a good management report module as well as statistics which make us to be alerted in each every moment about the live status of those servers. To achieve a complete monitoring state, it is required to have operators during the service hours, as Nagios is a tool that reports the states and there must be a human technician to react over the alarms.

It is possible to obtain information performing a configuration in Nagios for each case, through installed agents in the servers, or using the protocol SNMP (Simple Network Management Protocol) which will allow us to obtain complete information of the monitored systems.

Within this Final Degree Project (TFG as per its Spanish acronym) it is expected to design and implement the monitoring tool Nagios, to demonstrate the complexity of it, and to use is for monitoring servers, devices, networks and applications, showing the advantages to be written in multiple programming languages and being an open source software, with the final goal to be used in the classrooms and servers of the IT department of the University Carlos III de Madrid.

All these motivations make that this TFG, would be a really interesting project, as in the world of TIC (TCI) every day is more indispensable the proactivity, continuous improvement and the supervision of every system. As per everything already mentioned, it is required to have a monitoring system which

allows keep track and notify every event and also have reports which provide the system availability information.

## **1.5. Objective**

The main objective of the project is to demonstrate the great usability and availability of the monitoring system Nagios. Apart from installing, configuring and implementing the application, it will be shown how it Works and new ways to control all the systems available in the TIC (TCI).

As specific objectives of the project, highlight the importance of:

- Detect the use of resources and sizing of them within the monitored equipment.
- Verify the network infrastructure of the equipment
- Analyze the efficiency of the systems
- Improve the quality of the available services
- Reduce costs of the used services

There are multiple reasons to monitor automatically the servers, but the most important one is the economic, as the cost of a system will depend on its availability, redundancy, scalability and its efficiency.

The defined objectives in the previous paragraph, are determined by the needs to maintain available the system of attention and recovery of the services that a company and organism provide to its customers or other companies from which depend multiple stocks and quotes.

A critical example of the availability of a service could be the case of a server like “guernika” which provides service to a lot of users within the informatics classroom for many different department subjects, and if the server would experiment a failure and not detected until an alumni or teacher realizes that the service is unreachable. During my years in college I had to connect to “guernika” and multiple times there have been system collapses, only detected once it was failed, not being detected by the incremental use of memory or CPU usage.

Another example of critical service could be the devices within the informatics classrooms, to be able to measure their performance, usability and availability, it would be highly recommended to be monitored. These devices are used during the year, and during the regular class hours and additional practices their usage is really high, causing in multiple occasions performance issues due to hat usage. One of the advantages of being monitored, would be to predict possible problems and errors that could potentially happen in the disks (storage or performance), memory, CPU or even in the network. These devices are daily exposed to virus intrusions, fraud downloads and unexpected failures, making Nagios a great asset to optimize those informatics classrooms.

Improve the quality of the services, is becoming these days the main goal, to offer competitive services and achieve economic and social objectives, to improve the user satisfaction using those devices.

## **1.6. Rest of the document's organization**

Within this section we are going to detail which has been the project's organization to elaborate this documentation, by beginning with the tasks and planning followed in them.

First of all, it was performed an analysis of the most important characteristics to be detailed, as well as all the necessary topics to show the application in its maximum exponent.

As the mentioned system doesn't require a high use of resources, it has been possible to install, configure and administer it for this project a software like Nagios.

Coming up next, it is exposed a summary of the content of every chapter developed throughout the project.

### **Chapter 1: Introduction**

---

Within this chapter, it is described the context of the project, revealing motivation and the main and specific objectives of implementing them. In it, it is described the ideas and solutions to satisfy the software needs which report us every day all the IT systems.

### **Chapter 2: State of the Art**

---

The following chapter provides a wide vision about Nagios, detailing the comparatives between this software and other similar solutions but more limitations. Currently there are four possible alternatives, stating after comparing them that Nagios receives the best score in most of topics.

### **Chapter 3: Analysis, design and installation of the system**

---

This chapter is the most comprehensive from the memory, where all the low level details about Nagios are explained.

In the Analysis subsection, more general details about the Nagios performance are described as well as the prerequisites before going in deep into the topic.

The Design subsection shows all the tools and technologies used to build the system as well as all the use case diagrams, scenarios, browsing diagrams, database schema and the software functional and non-functional requirements.

A step-by-step guide will be described in the Installation subsection detailing how to deploy, install, configure and develop the Nagios software, adding also a comprehensive Admin guide to help with this software administration in any organization.

#### **Chapter 4: Assessment**

---

Within this chapter, the testing has been divided in two parts to be able to provide a more detailed assessment. Verification and performance testing results are described, where within the first tests is confirmed that all the software prerequisites are acknowledged and verified and the second tests shows that all the obtained data is evaluated for both operators satisfaction, who receive within seconds the servers status and the service improvement once a server and its services are being monitored by Nagios.

#### **Chapter 5: Planning and budget**

---

In this chapter, it is going to be detailed what the costs of the project are having in mind the data about the scheduling and planification used to develop and implement it.

In addition to it, it is going to be described a real case of an installation, configuration and maintenance estimate costs for a company or organization with that monitoring service, where a Nagios administrator maintains the daily operation, keeps track of the alerts and research about new methods to improve Nagios.

#### **Chapter 6: Conclusions and future projects**

---

As a last step, the final conclusions about the project are going to be presented as well as the achieved expectations once the project is finished and the memory has been redacted having spent about 11 months on it.

#### **Chapter 7: Appendix: Glossary of Terms**

---

In this chapter, terms and acronyms used throughout this memory are going to be described and explained to the greatest extent possible, for the understanding of all, whether or not they belong to the IT sector.

### **Chapter 9: Appendix: Guides**

---

Two guides are included to close the project, one administration guide and another user guide, as well as the tests scripts implemented to confirm the Nagios system is able to monitor many services and processed, being developed these scripts in different programming languages.

### **Chapter 8: Bibliography**

---

At this stage, all the websites and projects consulted to create this Final Degree Project (TFG) as all the tests and implementations performed will be summarized to achieve the goals.

## **Capítulo 2: Estado del arte**

---

## 2. Estado del arte

En este capítulo se van a realizar varios estudios de viabilidad y disponibilidad de un sistema de monitorización automatizado para sistemas informáticos, estudiando todos los niveles de arquitectura de los mismos, para poder asegurar y comprender su correcto funcionamiento.

### 2.1. Introducción

Para poder elegir un sistema de monitorización, hay tres valores técnicos que se deben valorar a la hora de elegirlo, independientemente del precio, usabilidad y disponibilidad. Los tres valores a tener en cuenta son:

- La forma de presentar los datos, los eventos, los gráficos y las alertas para poder procesarlos y darles uso, intentando que sean los más eficientes y con una simple vista, poder entender todos los datos que ahí se reflejan.
- Los tipos de conexión entre servidores y las distancias entre ellos, será un punto importante a tener en cuenta ya que variará la velocidad y rendimiento, ocasionando posibles problemas de conexión.
- Los sistemas operativos a monitorizar tendrán que ser valorados, ya que dependiendo la carga que tengan, el servicio que presten y la disponibilidad de los mismos, tendrán tipos de monitorización distintos.

Actualmente existe multitud de software de monitorización, pero ninguno de ellos tiene la suficiente potencia para competir con Nagios, tanto de funcionalidad como de coste, que serán analizados en los siguientes capítulos.



Ilustración 1.- Logo Nagios

Nagios tiene dos tipos de versiones, Nagios Core (Software Libre) y Nagios XI (compra de licencia con un coste alto al año), y como es lógico en la versión XI dispondremos de muchos más Plugins para poder monitorizar más servicios y especificaciones más técnicas.

## XI Standard Edition

100 Node: \$1,995  
200 Node: \$2,995  
Unlimited Node: \$4,995

## XI Enterprise Edition

100 Node: \$3,495  
200 Node: \$4,495  
Unlimited Node: \$6,495

Ilustración 2.- Precios de Nagios XI

Existe la versión “XI Standard Edition” donde el número de nodos a definir, determinará el precio de la licencia al año, que oscilará entre \$1,995 y \$4,995. Para la versión “XI Enterprise Edition” tendrá la misma consideración en el número de nodos, y por sus mejores características que la anterior versión, tendrá un coste de entre \$3,495 y \$6,495. Ambas versiones XI tendrán un configurador automático, proporcionando servicios que Nagios Core solo proporcionaría si un administrador de la aplicación los incluyese y los integrase en la aplicación. Para poder usar Nagios XI, simplemente se deben tener pequeños conocimientos de informática, para poder dar de alta todos los servidores y servicios asociados, en una interfaz web incorporada que hace que el alta, modificación y baja de servidores y servicios sea muy fácil e intuitiva. La ventaja de usar Nagios Core y disponer de un administrador de la aplicación, es la inclusión de nuevos servicios a monitorizar, con el desarrollo de scripts. En cambio, con la versión de Nagios XI se usarán los chequeos definidos en la aplicación y para poder incluir chequeos poco frecuentes, se tendrá que disponer de un administrador de la aplicación.

Ambas versiones monitorizan los servidores y servicios de la misma manera, pero tienen especificaciones distintas, ya que con Nagios Core se deben compilar todos los paquetes que se decidan instalar en el servidor, y Nagios XI, es proporcionado por la empresa de Nagios, directamente en una máquina virtual con el sistema que se desee (VMWare o Hyper-V, por ejemplo) para poder ser integrado directamente en el sistema informático de una empresa.

A día de hoy, son muchos los organismos que utilizan Nagios, pero que usan la versión Core, ya que al ser Software Libre es 100% personalizable, lo que tiene como ventaja que puedes generar tus propios scripts, monitorizando así lo que más interese a la organización donde esté instalado. La versión XI es mucho más interactiva gráficamente, y no es necesario disponer siempre de un administrador de Nagios ya que puede configurarse fácilmente en entorno web.

La gran variedad de software de monitorización, hace que Nagios esté observado siempre por sus competidores, pero debido a la gran comunidad de desarrolladores de la que dispone, siempre está actualizado a la última en cuanto a versiones y complementos.

A continuación, vamos a proponer varios sistemas para problemas diversos y confirmaremos que con Nagios es más rápido, fácil e intuitivo para llegar a la solución del problema.

## **2.2. Análisis de aplicaciones**

Actualmente hay multitud de aplicaciones de monitorización y muchas de ellas de Software Libre, pero se cree que no alcanzan el nivel de satisfacción necesario. Algunas de ellas, tienen su limitación de servicios monitorizados, y suponen un coste elevado para la ampliación y mantenimiento de dichas sondas.

Estas aplicaciones, además de permitir la detección de fallos, permiten determinar situaciones y evitar, en muchos casos, grandes problemas. También nos proporcionan, cada aplicación a su manera, la posibilidad de consultar datos de históricos, que nos informarán sobre los datos de monitorización anteriores, pudiendo dar a conocer una idea de las posibles alertas.

Por otro lado, disponen de monitorización por estados, que nos ayudarán a determinar la criticidad de la alerta y la actuación a realizar en cada caso. Y por último la monitorización de rendimientos o tendencias, como por ejemplo podría ser en el caso de monitorizar el tráfico de un Switchs, que ha tenido un frecuencia de un 20% de caudal y de repente pasa a tener un frecuencia constante de 70%, lo que daría a tener que analizar dicha información para poder saber el problema de dicho ancho de banda en nuestra red.

Se van a detallar de forma simplificada cuatro de las posibles soluciones más la propuesta en esta memoria del presente trabajo (Nagios), para determinar el porqué del uso de Nagios, donde se podrá observar fácilmente que las aplicaciones que se exponen son también potentes, pero tienen un cierto número de características que son necesarias para poder trabajar en una empresa, y éstas tienen sus ventajas e inconvenientes, que hacen que la valoración de las mismas influya en la elección, por muchas características, que se van a exponer a continuación.

A la hora de elegir dentro de una empresa y/u organismo una aplicación de monitorización, se deberán tener en cuenta multitud de características, de las cuales vamos a mencionar en los siguientes puntos [2], con las cuatro aplicaciones seleccionadas, para determinar si es recomendable el uso de Nagios o es preferible elegir para su uso y despliegue, algunas de las aplicaciones propuestas para su comparativa. El objetivo de nuestro proyecto es la monitorización de sistemas informáticos en la Universidad, por lo que se han valorado los puntos que se van a usar, para sacar el mayor rendimiento a la aplicación y poder observar de forma rápida y concisa los errores que pudiesen surgir.

### 2.2.1. Solución 1: Cacti



Ilustración 3.-  
Logo Cacti

Herramienta web que nos presenta en una interfaz gráfica todos los chequeos que queramos realizar en servidores y redes [20].

Principalmente, la aplicación está enfocada a la representación de los datos en gráficos para hacer que el usuario tenga mayor conocimiento de la gravedad de las diferentes alertas que aparezcan en la aplicación.

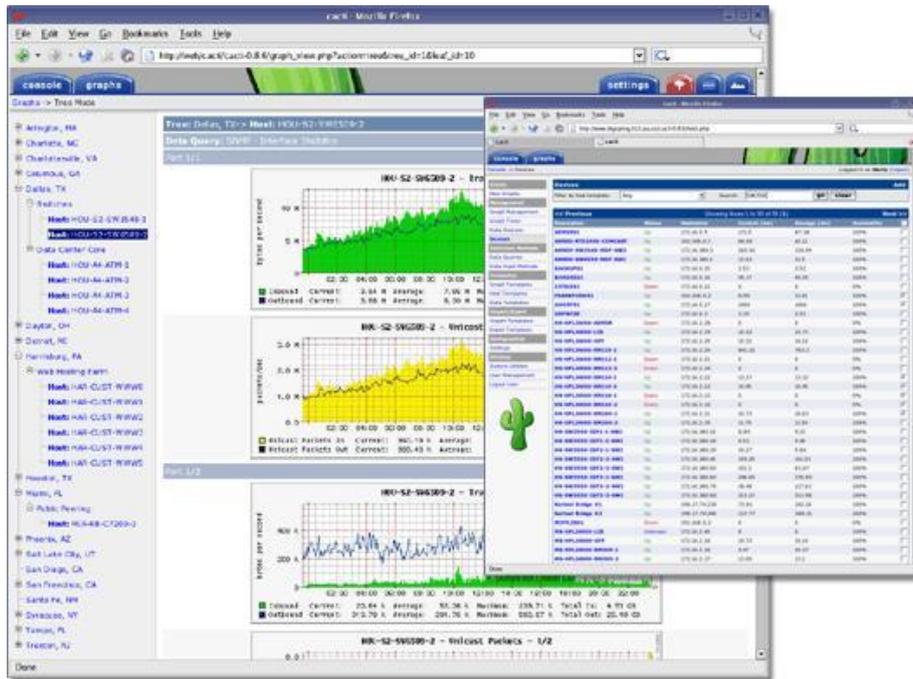


Ilustración 4.- Ejemplo gráfico Cacti

Esta herramienta también es compatible con arquitectura de Plugins y tiene una comunidad grande de desarrolladores que proporcionan multitud de scripts para el mantenimiento de dicha aplicación, pero su base de datos no está en formato MySQL, si no es una serie de ficheros de texto que hacen que las consultas tengan un alto grado de dificultad. No dispone de entorno servidor ni de protocolo SSH, lo que dificulta la posibilidad de usar determinados chequeos necesarios en algún servidor.

La negativa a usar este software es la no generación automática de informes, que hace que los datos de usabilidad y rendimiento solo sean visibles instalando distintas herramientas de complementos, pero no suficientes para obtener todos los datos de la aplicación.

## 2.2.2. Solución 2: Zenoss



Ilustración 5.- Logo Zenoss

Herramienta que dispone de una interfaz muy intuitiva posibilidad de percibir de un vistazo toda la información de un servidor [19].

La configuración que tiene dicha herramienta es especialmente difícil de manipular, y dispone de multitud de Plugins para poder monitorizar equipos en una red automáticamente utilizando SSH y SNMP. Este software no necesita usar agentes remotos, porque usando SSH puede solicitar cualquier comando a cualquier servidor para poder extraer toda la información necesaria.

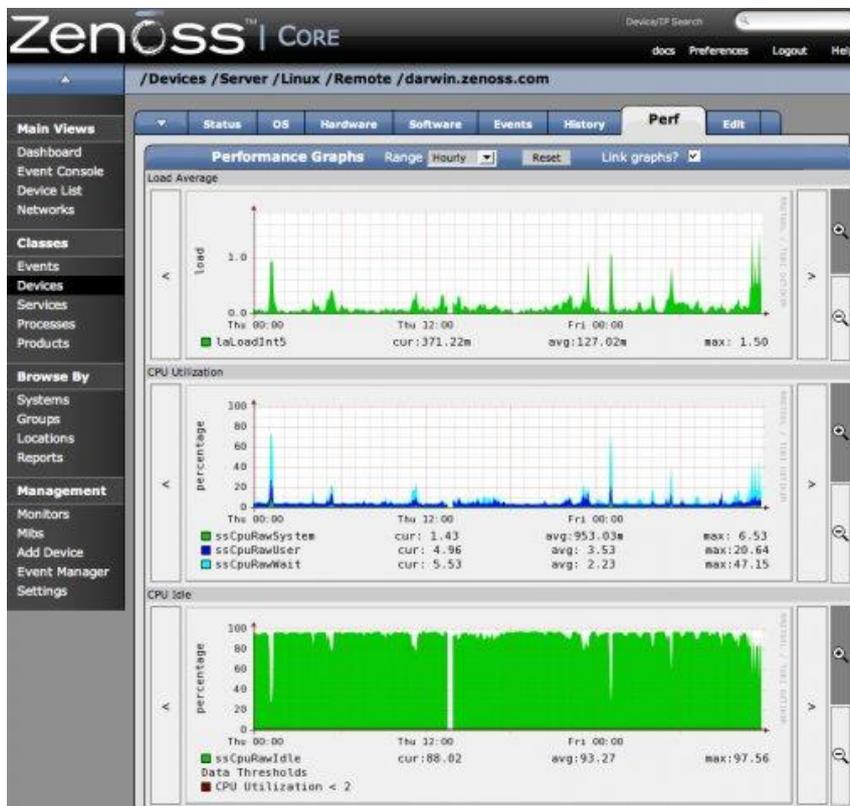


Ilustración 6.- Ejemplo gráfico Zenoss

Esta herramienta provee de muchas funcionalidades de monitorización de rendimientos y disponibilidad de servicios, con detección de elementos en la red y el sistema de alertas. Es muy completa pero ofrece multitud de problemas a la hora de configurar chequeos, ya que se deben desarrollar prácticamente todos ellos para poder usarlos en los sistemas operativos existentes.

### 2.2.3. Solución 3: Pandora FMS



Ilustración 7.-  
Logo Pandora  
FMS

Herramienta de software libre y con versión de licencia comercial, que dependerá su coste del número de agentes instalados y el número de nodos a monitorizar. Esta aplicación sirve para monitorizar y analizar todo tipo de servidores y aplicaciones, a través de una interfaz web por medio de un navegador [21]. Es una herramienta modular e intuitiva, orientada a objetos, que puede monitorizar todo tipo de servicios y parámetros con diferentes sensores mediante agentes instalados en los servidores que prestan el servicio.

Las comunicaciones entre el servidor donde se ubica la herramienta y los monitorizados puede realizar por SSH, NFS, FTP o XML, para poder mostrarlos por el entorno web del que dispone la aplicación.

Todo bajo un sistema de notificaciones, que son enviadas por correo electrónico o por SMS en el momento en el que se superen los umbrales registrados para cada chequeo.



Ilustración 8.- Ejemplo gráfico Pandora FMS

Trabaja sobre una base de datos MySQL y no requiere ningún tipo de mantenimiento, ya que se administran automáticamente todos los eventos que se registren en ella.

Al almacenar todos los datos en la base de datos es muy fácil la generación de gráficas y estadísticas en su interfaz web. Pandora FMS no es un sistema de monitorización de elementos críticos ya que no trabaja en tiempo real, ni tiene la opción de analizar logs o eventos.

Dispone de un acceso web que permite, como en todas las aplicaciones que se van a comparar, con permisos para administrarla y control sobre la herramienta, pudiendo crear diferentes usuarios con distintos permisos de acceso.

Está programado en PHP5 y no necesita ningún software adicional para su correcto funcionamiento. El entorno web está desarrollado en AJAX y HTML5, y para las gráficas es necesario disponer de Flash en el navegador, para poder visualizarlas.

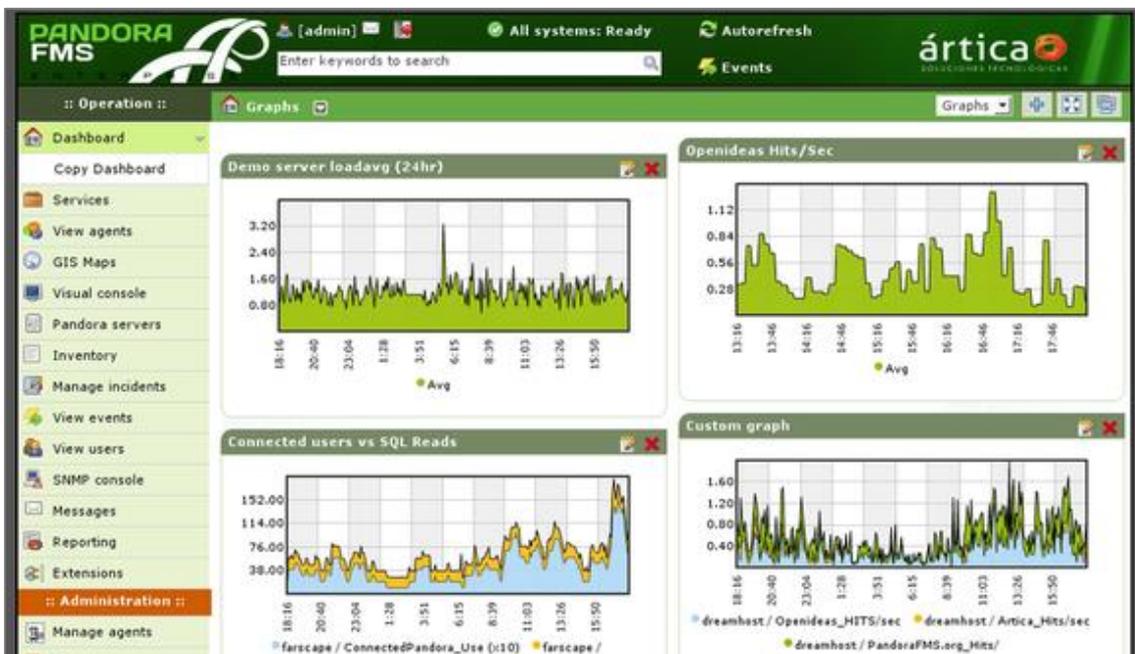


Ilustración 9.- Ejemplo gráfico Pandora FMS (gráficos)

Dispone de una plataforma similar a todas ellas, pero es necesario de disponer de varios meses para conseguir dominar la estructura de ficheros y la usabilidad de la aplicación, ya que tiene multitud de complementos para el acceso web.

## 2.2.4. Solución 4: Zabbix



Ilustración 10.- Logo Zabbix

Herramienta de monitorización capaz de recopilar datos de servidores y aplicaciones, por medio de agentes instalados en las máquinas cliente [22].

Esta herramienta tiene un sistema proactivo de acciones donde puede solucionar automáticamente errores de servidores con pequeñas tareas de ejecución.



Ilustración 11.- Ejemplo gráfico Zabbix

La herramienta tiene un sistema de gráficos muy potente que muchos software no disponen de él, pero también es un inconveniente, ya que su vista fundamentalmente es gráfica, y no aporta mucho valor instantáneo de los problemas encontrados en cada momento, sobre todo para administradores de los sistemas y aplicaciones que están monitorizados.

Dispone de simples chequeos que pueden verificar los niveles de respuesta y disponibilidad, como chequeos activos y dispone de un agente de monitorización que se instala en cualquier sistema operativo para monitorizar el resto de servicios y procesos existentes en los servidores.

## 2.3. Tabla comparativa con las distintas soluciones y la propuesta

Se ha decidido realizar una comparativa por niveles entre Nagios y las cuatro aplicaciones que se han seleccionado de similares características, ya que dependiendo de la aplicación, se puede valorar muchos criterios que tienen más o menos consideración, a la hora de conocer la complejidad, usabilidad, disponibilidad y operatividad de una aplicación software libre de monitorización.

El nivel más determinante posiblemente sea el de aplicación, ya que para poder comparar las aplicaciones, serán determinantes muchas características de los mismos para poder analizarlas.

Todas las especificaciones que se han elegido para realizar las tablas de comparativas [5], han sido seleccionadas con la finalidad de ser útiles para las necesidades de monitorización del sistema informático del Departamento de Informática de la Universidad.

### 2.3.1. A nivel sistema operativo

Primero, se evalúan a nivel sistema operativo las cinco aplicaciones que se quieren comparar, teniendo en cuenta los puntos que son de interés para cumplir con los objetivos que se han establecido.

App	Open Source	MySQL	Entorno		Protocolos	
			Web	Server	SNMP	SSH
Cacti	✓ Sí	✗ No	✓ Sí	✗ No	✓ Sí	✗ No
Zenoss	✓ Sí	✓ Sí	✓ Sí	✗ No	✓ Sí	✓ Sí
Pandora FMS	✓ Sí	✗ No	✓ Sí	✗ No	✗ No	✓ Sí
Zabbix	✓ Sí	✓ Sí	✓ Sí	✓ Sí	✗ No	✗ No
NAGIOS	✓ Sí	✓ Sí	✓ Sí	✓ Sí	✓ Sí	✓ Sí

Tabla 1.- Comparativa de las distintas soluciones a nivel sistema operativo

Como se puede observar, Nagios tiene mejores características a nivel sistema operativo, ya que se pueden configurar los entornos y los protocolos, que a diferencia de otras aplicaciones no disponen, por lo que se hace más manejable y útil la utilización de Nagios.

### 2.3.2. A nivel aplicación

Para poder valorar la parte más importante de una aplicación a la hora de determinar el funcionamiento de la misma, es necesario exponer las características que se necesitan en el departamento de Informática.

App	Gráficas	Informes SLA	Grupos lógicos	Predicción estadísticas	Monitorización distribuida	Interfaz web		
						Control	Vistas	Informes
Cacti	✓ Sí	✓ Sí	✓ Sí	✗ No	✓ Sí	✓ Sí	✓ Sí	✓ Sí
Zenoss	✓ Sí	✗ No	✓ Sí	✓ Sí	✗ No	✓ Sí	✓ Sí	✗ No
Pandora FMS	✓ Sí	✓ Sí	✓ Sí	✓ Sí	✗ No	✓ Sí	✗ No	✓ Sí
Zabbix	✓ Sí	✓ Sí	✗ No	✓ Sí	✗ No	✓ Sí	✗ No	✗ No
NAGIOS	✓ Sí	✓ Sí	✓ Sí	✓ Sí	✓ Sí	✓ Sí	✓ Sí	✓ Sí

Tabla 2.- Comparativa de las distintas soluciones a nivel aplicación

Actualmente, hay varias características importantes, como por ejemplo, la exposición de la información de los chequeos mediante gráficas, para poder visualizar en tiempo real las alertas que van apareciendo en el sistema. Otra es la característica de poder presentar informes SLA (en Español, Acuerdos de Nivel de Servicio) que nos reportarán la disponibilidad y los tiempos de respuesta de los servidores y servicios monitorizados.

Otra característica bastante importante es en la interfaz web, la disponibilidad de tener una vista de control, las vistas de disponibilidad e informes de rendimientos y alertas. Muchas aplicaciones disponen de ello, pero en el caso en el que nos centramos son pocas las que nos ofrecen las tres características necesarias para una interfaz web completa.

### 2.3.3. A nivel complementos

En este punto se van a determinar los complementos que son necesarios para la exposición completa de las alertas que pueden ser visibles en Nagios, ya sea en el entorno web, por correo electrónico o mediante logs.

App	Creación complementos	Eventos	Mapas	Logs	Logs avanzados	Recomendaciones automáticas
Cacti	Media	✗ No	✓ Sí	✓ Sí	✗ No	✓ Sí
Zenoss	Media	✓ Sí	✓ Sí	✓ Sí	✗ No	✗ No
Pandora FMS	Fácil	✓ Sí	✓ Sí	✓ Sí	✓ Sí	✗ No
Zabbix	Media	✓ Sí	✓ Sí	✓ Sí	✗ No	✗ No
NAGIOS	Fácil	✓ Sí	✓ Sí	✓ Sí	✓ Sí	✓ Sí

Tabla 3.- Comparativa de las distintas soluciones a nivel complementos

Como se puede observar en la tabla anterior, no existe ninguna aplicación, entre las analizadas, que nos facilite y nos dé solución a nuestras necesidades.

Después del resumen de cada una de las aplicaciones que son competencia directa con Nagios, se puede destacar que ninguna de las soluciones propuestas es capaz de abarcar todas las necesidades y beneficios que nos aporta Nagios.

Aunque muchas de ellas tienen entornos parecidos, similitud en cuanto al funcionamiento y a la usabilidad, cabe destacar que está demostrado que Nagios tiene un muy buen rendimiento, es fácilmente administrable por personal informático que disponga de conocimientos en los lenguajes de programación C, C++, HTML y Perl, y sea capaz de manejarse en un SO Linux para la administración del servidor Nagios.

Como conclusión de las posibles aplicaciones que dan solución a algunos de los problemas que se están valorando, hay que destacar que existen varias que dan solución a algunos de los problemas, pero no al mismo nivel al que se pretende llegar, que con Nagios se puede fácilmente.

## **Capítulo 3: Análisis, diseño e implantación del sistema**

---

### **3. Análisis, diseño e implantación**

En este capítulo se van a presentar todos los puntos importantes de este trabajo, dando a conocer el análisis, el diseño y la implantación del software libre para poder realizar su ciclo de vida completo.

De este modo, se podrán analizar todos los criterios que son necesarios para demostrar que dicho software cumple con todos los objetivos que necesitamos alcanzar para determinar la eficiencia del mismo.

#### **3.1. Introducción**

Para empezar, se estableció que el proyecto iba a constar de varias partes, para determinar el alcance de la herramienta de monitorización Nagios y una infraestructura virtual para su monitorización. En estas dos partes, se montaría Nagios en una máquina que pudiese tener completa conectividad y permisos para comunicar dicha infraestructura virtual, para así demostrar la correcta funcionalidad del sistema.

Algunas de las principales características de Nagios [6] son:

- Monitorización de los recursos (CPU, discos, memoria, procesos, servicios, etc.).
- Supervisión de servicios de red (SMTP, HTTP, POP3, PING, etc.)
- Determina una jerarquía de servidores en una red, siendo posible detectar servidores inalcanzables.
- Registra de forma automática la rotación de logs de la aplicación.
- Tiene un sistema de notificaciones altamente potente, que notifica errores cuando existen problemas o alertas, que son enviados por correo electrónico, siendo también posible su integración con diferentes sistemas de mensajería.
- Dispone de una interfaz web para la visualización del estado actual de toda la infraestructura de red monitorizada en tiempo real.
- Almacena todos los datos en una base de datos MySQL, ofreciendo la posibilidad de generar gráficas e informes de toda la red.

El ciclo de vida básico en cualquier sistema informático a desarrollar y mantener, consta de los ciclos generales [8] siguientes:

- Análisis: donde se explica todo el desarrollo del proyecto y todos los puntos que se van a analizar e implementar, para conseguir objetivos determinados.
- Diseño: punto clave en un sistema, donde se explicarán todas las partes del análisis en mayor profundidad, para determinar el alcance, la operatividad y usabilidad del sistema.

- **Desarrollo:** es un punto necesario en todos los sistemas, donde es necesario realizar implementaciones para crear una aplicación, o en el caso que afecta, generar código para su puesta en marcha.
- **Pruebas:** paso imprescindible después de una implantación y anterior a una implementación, para poder proceder a su puesta en producción. Sin pruebas no hay un sistema estable, ni tampoco se puede asegurar que el sistema tenga los resultados acordados y esperados.
- **Implementación:** esta etapa es una de las más importantes y críticas, ya que después de un desarrollo, junto con las pruebas, es el momento de poner la aplicación en marcha y demostrar en un entorno de producción que todo está funcionando correctamente.
- **Mantenimiento:** por último, pero siendo una de las más importantes, un sistema sin este servicio, no puede durar mucho tiempo, por lo que para poder asegurar un correcto funcionamiento de un sistema, ya sea nuevo o ya implantado, deberá disponer de un administrador o un usuario que mantenga la aplicación, para poder asegurar su usabilidad y rendimiento a largo plazo. En este proyecto no se analiza el coste de mantenimiento porque no está prevista su contratación real.

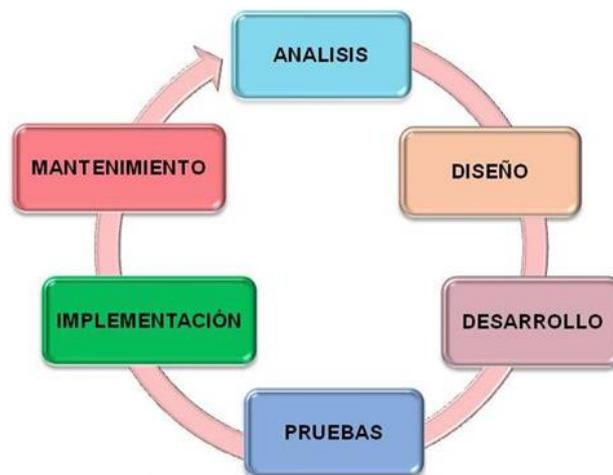


Ilustración 12.- Ciclo de vida del software aplicable a Nagios

En Nagios, un posible cuadro a seguir para desarrollar el sistema, donde determinar los puntos necesarios para entender un sistema de monitorización, sería el siguiente (véase Ilustración 13).

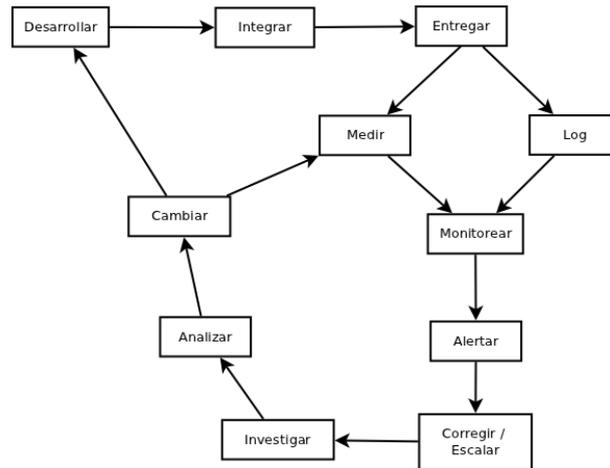


Ilustración 13.- Lógica de la aplicación de Nagios para monitorizar

Primero se desarrollarán los scripts y chequeos necesarios para la monitorización de todos los servicios específicos de algunos servidores, que no estén definidos por defecto en la instalación de Nagios (se dispondrá de varios ejemplos en el apéndice del presente documento).

A continuación, se integrarán en la aplicación y se pondrán en producción para su uso y monitorización en los sistemas en los que se deseen implantar. Estos chequeos, valorando su respuesta, se podrán medir y visualizar a través de logs, donde la monitorización por proactividad será el objetivo. Una vez recibida la alerta, se deberá informar a los contactos que sea necesario para escalar e investigar y analizar el problema, y proceder a su solución.

Nagios dispone de un potente sistema de escalado y monitorización a varios niveles que hace que se disponga de un alto nivel de detalle de información, haciendo que el período de investigación se reduzca exponencialmente y se disponga de todos los datos referentes a una alerta con todas las posibles vistas, tanto en la interfaz web, como por correo electrónico, para la llamada de atención a los contactos responsables de cierto servidor y/o servicios monitorizados.

### 3.1.1. Marco regulador

El crecimiento de la información manipulada por medios informáticos, cada día es más importante, y es necesaria la existencia de regulaciones, normativas técnicas o legales o diversas restricciones que apliquen al sistema.

Dado que Nagios es aplicable a infraestructuras de red donde ésta deberá ser segura y aislada, no será necesario el uso y mención de ninguna ley y/o normativa que competa a los datos usados para la monitorización de servidores y servicios. Todos los datos que son usados, son propios de la empresa y quedan salvados en los ficheros de configuración del servidor ubicado dentro de su red. Dicho ficheros, siempre serán propiedad de la empresa contratante.

### 3.2. Análisis

Antes de la implantación de Nagios, se ha tenido que realizar un análisis completo de la herramienta, así como su diseño, para poder determinar el nivel de monitorización al que deseábamos llegar en el Departamento de Informática de la Universidad, teniendo en cuenta todos los parámetros que eran necesarios monitorizar para poder tener un sistema estable, controlado y completo.

Nagios es un sistema de monitorización que presenta una visualización amplia y manejable de todos los servidores y servicios a monitorizar [6]. Para poder entender la organización en grupos, se van a definir los principales conceptos, para mostrar la gran capacidad de organización de la que dispone la aplicación.

En Nagios se deben definir varios conceptos, que nos servirán para poder entender en una primera idea la estructura de la herramienta [14][15].

- Grupos de máquinas (*hostgroups*): necesarios para la agrupación de servidores en diferentes vistas y así poder, por ejemplo, agruparlos por tipo de Sistema Operativo y/o categoría.
- Grupos de servicios (*servicegroups*): necesarios para poder agrupar los servicios monitorizados por tipos, como por ejemplo la agrupación de servicios Jboss o IIS.
- Máquinas (*hosts*): necesario para la definición de cada servidor monitorizado con la IP de la red que tenga fijada.
- Servicios (*services*): necesaria la definición de servicios en cada servidor para poder visualizar cambios de estado en la monitorización de dichos servicios. Como mínimo se establecerá el chequeo de PING para poder visualizar alguna información en el host.
- Chequeos activos: son aquellos que son usados por diferentes protocolos, que solicitan información a un servidor y éste le responde con los datos solicitados. Un sistema de request y response.
- Chequeos pasivos: son aquellos que usan varios protocolos definidos, donde el propio servidor cliente enviará información a Nagios.

A continuación, se exponen los protocolos más importantes existentes en Nagios, y que serán de los que se hablará durante la presente memoria.

Protocolo	Tipo	Significado	Comentario
NRPE	Activo	Nagios Remote Plugin Executor	Protocolo usado por Nagios para hablar con el agente instalado en un servidor con sistema operativo Windows y/o Linux/Unix.
NSCA	Pasivo	Nagios Service Check Acceptor	Protocolo usado para la recepción de la información generada por los servidores cliente y enviada a Nagios.
SNMP	Activo	Simple Network Management Protocol	Protocolo usado por Nagios para comunicarse con los dispositivos de red.

Tabla 4.- Tipos de protocolos usados en Nagios para monitorizar

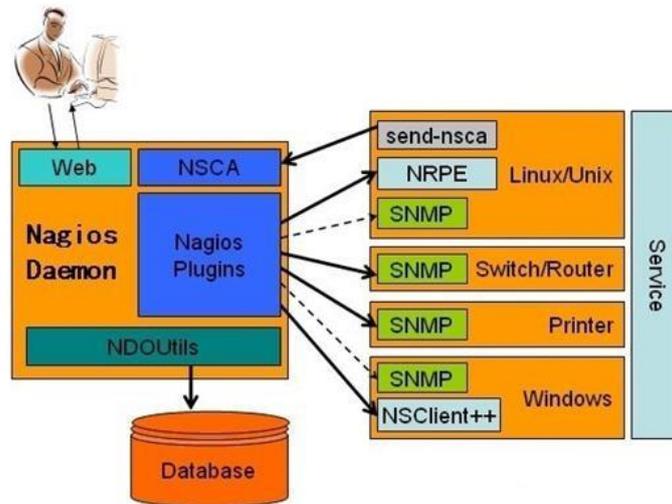


Ilustración 14.- Ejemplo de funcionamiento Nagios completo

La ilustración anterior, nos muestra el funcionamiento de la aplicación de Nagios, donde se pueden visualizar los diferentes tipos de chequeos y la respuesta de los mismos. El usuario se conecta al servidor vía web, donde puede visualizar todos los chequeos que recibe éste, ya sean por el protocolo NSCA, NRPE o SNMP. Cualquier protocolo usado para la recepción de chequeos, será visible en Nagios, teniendo en cuenta los parámetros que deben monitorizarse y los tiempos de respuesta que deben permitirse para cada caso, siempre que sea dado de alta de forma correcta, respetando la relación de Host-Hostgroup y Service-Servicegroup.

Nagios está compuesto por un núcleo que forma la parte lógica de control de negocio que tiene contenido todo el software necesario para realizar el proceso de monitorización de los servidores y sus servicios, haciendo uso de componentes propios de la aplicación y todos aquellos que sean realizados por otras personas, ya sean públicos de la página web oficial de Nagios, como desarrollados en cualquier lenguaje de programación por terceras personas [17].

Este software tiene permitida la captura de paquetes SNMP para la captación de sucesos, pero no es su principal funcionamiento ya que tiene un gran número de pequeños módulos software para realizar multitud de chequeos en dispositivos de red y servidores.

En cuanto a la monitorización activa y pasiva, hay que tener en cuenta varias ventajas e inconvenientes de cada uno de ellos [13], que se van a comentar en los puntos siguientes.

### **3.2.1. Chequeos activos**

Este tipo de chequeos envían paquetes desde Nagios a los servidores cliente, ya sea un PING o el chequeo que se necesite realizar (CPU, memoria, discos...) usando el puerto 1248 en los servidores clientes, y siendo recibido en Nagios por cualquier puerto viable del servidor.

Como gran ventaja destacar que instalando un pequeño agente de monitorización en cada servidor a monitorizar, será posible la recepción de la información de los chequeos realizados en cada uno, ya que las peticiones son de tipo request-response, obteniendo toda la información actualizada, según los parámetros que se estén queriendo visualizar y controlar.

Como única desventaja observable es que si el agente de Nagios instalado en el servidor como un servicio, se parase, no recibiríamos ningún tipo de información del servidor, y nos aparecerían en la interfaz web de Nagios, mensajes de imposibilidad de conexión con el servidor. Siempre este servicio deberá aparecer con arranque automático y activo.

Todos los chequeos para servidor Windows podrán realizarse con este tipo de chequeos, ya que el agente instalado en los servidores se establece como servicio con arranque automático en el servidor, haciendo que sea mucho más rápida la comunicación entre ambos servidores.

### **3.2.2. Chequeos pasivos**

Se basan en la recepción de información, enviada desde los clientes al servidor Nagios, disminuyendo así el tráfico de red, ya que las peticiones se realizan en un solo sentido.

Como ventaja destacar que la información que enviará el cliente siempre será más rápida, ya que se ejecutan los chequeos mediante "cron" y es el propio servidor monitorizado el que manda su información a Nagios.

La desventaja a mencionar es que los servicios no son monitorizados directamente por Nagios, si no que éste recibe la información del servidor monitorizado, y en el caso en el que éste se apague, solo veríamos el servicio de PING (chequeo activo) como CRITICAL (por ejemplo 100% de paquetes perdidos) y observando el resto de chequeos en OK, ya que la información no se envía a Nagios, porque no se ejecutan los chequeos en los servidores cliente. Tiene métricas menos exactas, ya que únicamente puede valorar los estados de OK, WARNING y CRITICAL, reflejando en gráficas sus estados en tres niveles.

Más fáciles de manejar son los chequeos activos, pero los chequeos pasivos dan la oportunidad de ampliar el número de chequeos a realizar, ya que conociendo los servicios, procesos y sobre todo, comandos con posibilidad de ser ejecutados en un servidor Linux/Unix, se podrán personalizar todos los chequeos que se desee.

### 3.2.3. Requisitos previos

Nagios es recomendable montarlo en un Sistema Operativo Linux donde pueda correr fluidamente sin necesidad de tener muchos procesos compitiendo con él, por lo que descartaremos cualquier SO de usuario como podría ser Ubuntu [1] [4]. El preferido por todos los desarrolladores de la comunidad es CentOS, actualmente en su versión estable 6.4. Es un SO ágil y con los suficientes recursos para poder administrar Nagios correctamente.

Una vez montado el servidor Linux con CentOS 6.4, se necesita tener instalado el SGBD MySQL y el servidor web Apache (véase en la ilustración 15). Las especificaciones de dicho servidor, según recomiendan los creadores de Nagios, son como mínimo 1Gb de RAM y 50GB de disco duro, y siempre se deberá incrementar de forma exponencial según el rendimiento que se le vaya a exigir, ya que a mayor número de elementos a monitorizar, mayores deberán ser las especificaciones de un servidor de monitorización con Nagios.

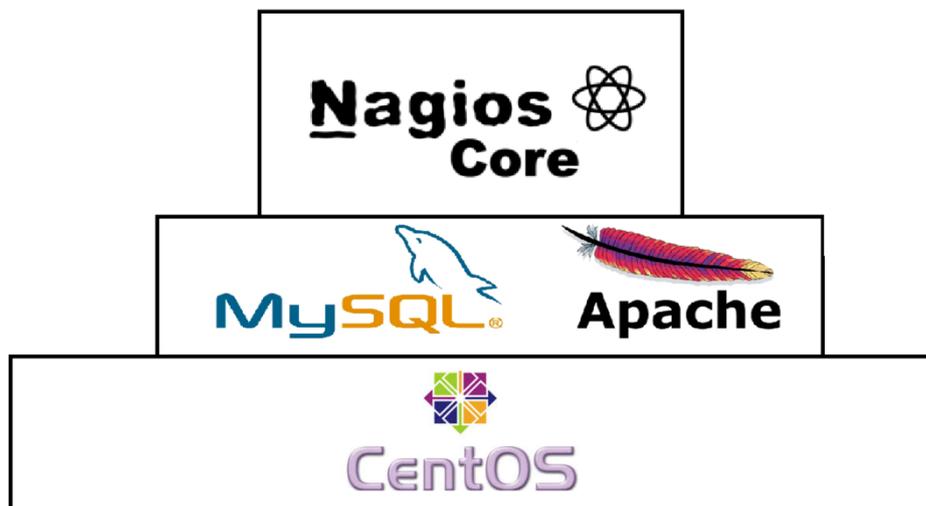


Ilustración 15.- Arquitectura del sistema de Nagios en capas

Una vez montado todo el SO con los requisitos que se han comentado, Nagios será instalado con los paquetes que vamos a definir en los siguientes apartados y que estarán claramente detallados en el [Manual de Administrador](#) [1] [3] [16] que se adjunta en el apéndice de la presente memoria.

### 3.3. Diseño

En este apartado se van a detallar todos los puntos claves para el diseño de la aplicación, donde se expondrá el que se ha decidido desarrollar para que Nagios funcione de la forma más completa y ágil posible en el Departamento de Informática de la Universidad.

Además de hablar de la interfaz de Nagios, se expondrá el diseño que se ha elegido para la base de datos, a la que se podrá acceder haciendo posible la usabilidad de la misma.

Se ha decidido usar un sistema operativo Linux con CentOS 6.4, donde se ha instalado un servidor apache y una base de datos MySQL. Para poder desplegar Nagios, es necesario disponer de estas dos herramientas antes de su instalación, debido a que son complementos necesarios (dependencias) para su acceso, instalación y utilización.

A continuación, se muestra en una imagen, una lógica del despliegue a nivel de sistema, de datos y de interfaz (véase Ilustración 16).

- En el acceso a datos deberemos disponer de un disco duro para almacenar la información, una base de datos MySQL donde almacenar los datos referentes a Nagios, y acceso a la red para poder monitorizar todos los servidores y servicios disponibles en la red donde está prestando Nagios el servicio.
- En la lógica del sistema tendremos el Núcleo de Nagios, donde están compilados y desplegados todos los chequeos o extensiones, ya sean oficiales o de terceros, para poder monitorizar todo lo que sea necesario en la red.
- En la interfaz se dispondrá de un acceso web mediante el servicio de apache, que publicará una web intuitiva y completa de Nagios, que será accesible desde cualquier navegador desde un equipo de la red.

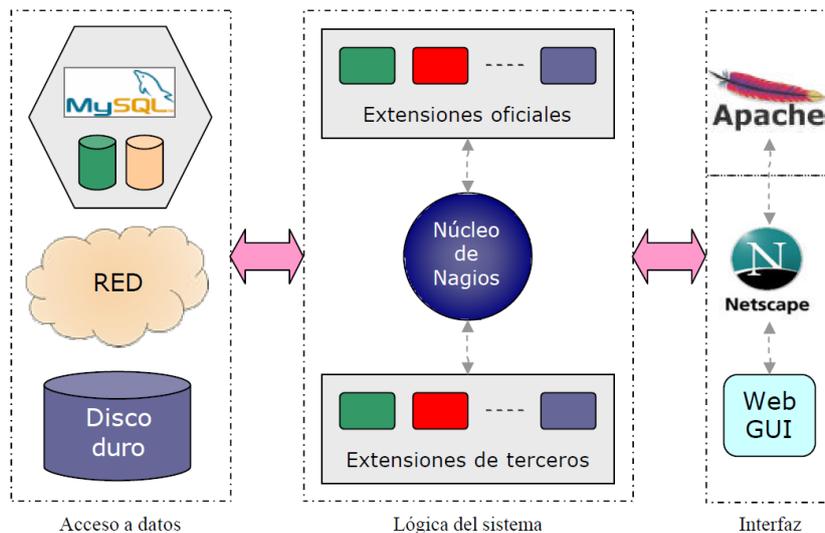


Ilustración 16.- Lógica del sistema de Nagios en capas

En este caso, para nuestro proyecto, Nagios es accesible por medio de la URL siguiente:

<http://192.168.1.5/nagios>

En el momento en el que por medio de nuestro navegador, nos conectamos a la interfaz web de Nagios [7], es de valorar la operativa que por detrás realiza el software para poder mostrar la información actualizada y completa (véase Ilustración 17).

Nagios, por medio de sus Plugins, monitoriza de forma activa o pasiva servidores, bases de datos, switch, etc., y es accesible el estado de los mismos por medio de un navegador y las notificaciones serán enviadas por correo electrónico. Siempre seguirá esta operativa, para monitorizar y avisar de los problemas y alertas que surgan en la red.

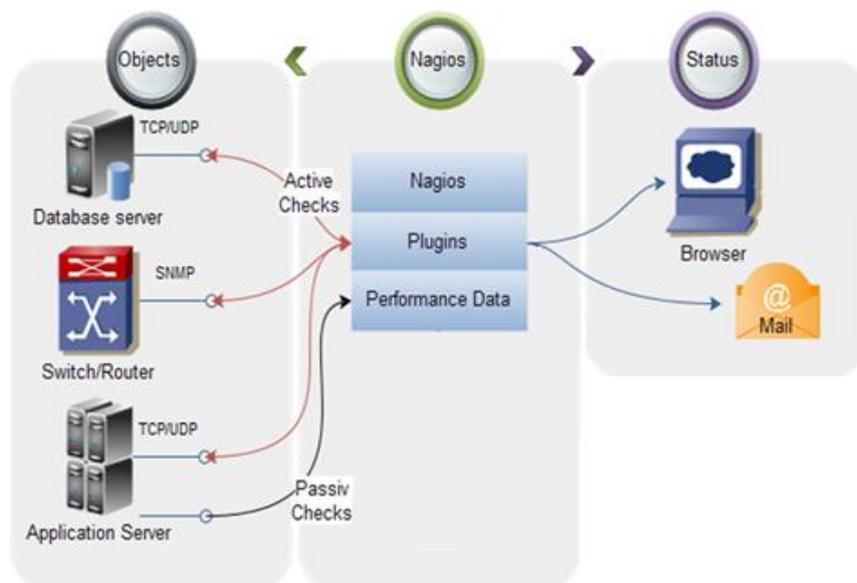


Ilustración 17.- Diagrama de flujo de datos en Nagios

Para poder exponer el diseño completo de la aplicación de Nagios, es necesario desglosar la misma en los diferentes casos de uso, requisitos funcionales y no funcionales, y esquemas de bases de datos.

Junto con los siguientes datos y el manual detallado de administrador que ha sido redactado, se podrá disponer de todos los datos para el despliegue, implantación, configuración y correcto funcionamiento de Nagios, en cualquier red que se quiera implantar.

En el primer inicio de Nagios, el diseño gráfico inicial es el predefinido, y dispone de varios para que puedan cambiarse y modificarse según la imagen que se quiera dar del mismo. Para la versión 4, son tres los temas disponibles llamados “Exfoliation” (véase Ilustración 18), “Vautour” (véase Ilustración 19) y “Arana” (véase Ilustración 20) [7].

The screenshot shows the Nagios Tactical Monitoring Overview page. It includes a left sidebar with navigation menus for General, Current Status, Reports, and System. The main content area displays 'Tactical Monitoring Overview' with system information, 'Monitoring Performance' statistics, 'Network Outages' (0), 'Network Health' (Host and Service Health), 'Hosts' status (0 Down, 0 Unreachable, 13 Up, 0 Pending), 'Services' status (0 Critical, 1 Warning, 0 Unknown, 45 Ok, 0 Pending), and 'Monitoring Features' for Flap Detection, Notifications, Event Handlers, Active Checks, and Passive Checks, all showing 'All Services Enabled' or 'All Hosts Enabled'.

Ilustración 18.- Frontal Web "Exfoliation" para Nagios

El disponible por defecto es el primero de ellos, y se podrán modificar simplemente sustituyen los ficheros gráficos existentes en "/usr/local/nagios/share", por los descargados.

The screenshot shows the Vautour Nagios web interface. It features a sidebar with 'Monitoring', 'Reporting', and 'Configuration' sections. The main content area displays 'Current Network Status' (Last Updated: Fri May 29 00:43:44 CEST 2009), 'Host Status Totals' (Up: 1, Down: 0, Unreachable: 0, Pending: 0), and 'Service Status Totals' (Ok: 2, Warning: 0, Unknown: 0, Critical: 0, Pending: 0). A table shows 'Service Status Details For All Hosts' for 'localhost' with services 'load' and 'ping' both in 'OK' status.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	load	OK	05-29-2009 00:42:21	0d 1h 13m 23s	1/3	OK - load average: 0.27, 0.09, 0.03
localhost	ping	OK	05-29-2009 00:41:51	0d 1h 28m 53s	1/3	PING OK - Packet loss = 0%, RTA = 0.14 ms

Ilustración 19.- Frontal Web "Vautour" para Nagios

El tema "Vautour" tiene un formato atractivo y rápido, donde en el menú izquierdo, como en todos los diseños de Nagios, se dispondrá de acceso a todos los servicios disponibles con un diseño diferente (véase Ilustración 19).

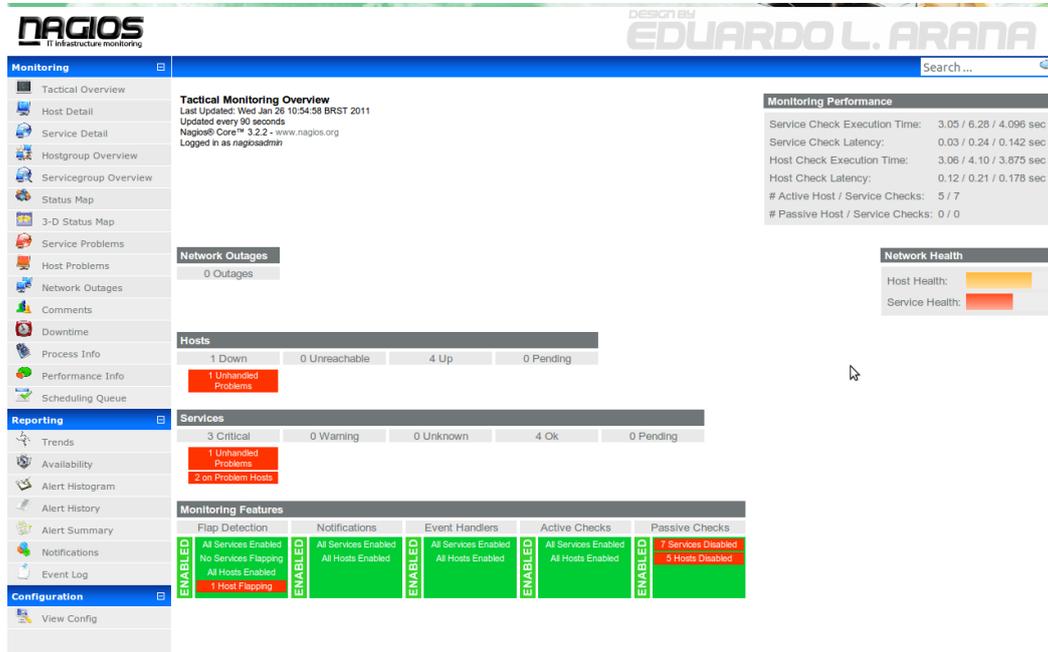


Ilustración 20.- Frontal Web "Arana" para Nagios

El tema de "Arana" es más joven, atractivo y dinámico, pero al disponer de colores e imágenes, hace que éste sea más pesado a la hora de realizar las recargas automáticas de la página web que se esté visualización (véase Ilustración 20).

Estos tres temas son los desarrollados por la propia empresa de Nagios, pero existen multitud de ellos desarrollados por la comunidad, siendo estos tres temas también totalmente personalizables en los scripts existentes en las rutas de /usr/local/nagios/share.

### 3.3.1. Diagramas de casos de uso

En este apartado se van a definir los casos de uso que van a indicar las necesidades del sistema de monitorización Nagios, acorde con el análisis realizado anteriormente para la aplicación.

Se mostrarán todos los casos de uso de este sistema de monitorización que se encargará de supervisar el correcto funcionamiento tanto de los sistema de comunicaciones de la red, como de los servidores y servicios que componen la infraestructura, para informar a uno o varios administradores en el momento en el que se detecte algún problema que pueda ocasionar algún fallo más grande, desencadenando otros problemas.

La tabla que va a ser usada para exponer los casos de uso, es la siguiente:

Identificador: CU xxx	
Nombre	
Actores	
Objetivo	
Precondiciones	
Postcondiciones	
Escenario básico	

Tabla 5.- Tabla de ejemplo de un Caso de uso

Dónde:

- Nombre: es el identificador del caso de uso.
- Actores: son los que pueden hacer uso de dicho caso.
- Objetivo: representa al significado del mismo.
- Precondiciones: las que son necesarias para que dicho caso se complete satisfactoriamente.
- Postcondiciones: las que se deben comprobar una vez realizados todos los pasos que contempla el caso.
- Escenario básico: los pasos que se deben realizar para conseguir la opción que se necesitan para realizar el caso.

### 3.3.1.1. Diagrama de descomposición

Para poder visualizarse correctamente todos los casos de uso existentes en este proyecto para la implantación del sistema de monitorización Nagios, se ha decidido subdividir la aplicación en varios módulos, los cuales se exponen en el siguiente diagrama de descomposición (véase Ilustración 21).

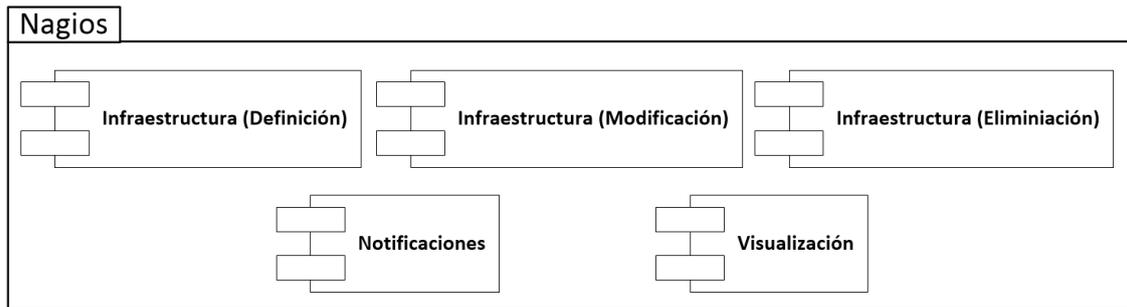


Ilustración 21.- Diagrama de descomposición de Nagios diseñada

Es necesario mencionar que en el desarrollo de los siguientes diagramas y casos de uso, se ha reflejado como precondition el acceso al servidor de Nagios y/o al servidor a monitorizar, siendo ésta necesaria para el administrador en ambos casos, para no tener problemas en la instalación y configuración de los agentes correspondientes, en el caso de los hubiese.

A su vez, comentar que los ficheros con extensión “\*.cfg” que se mencionan en los casos de uso creados, se definen con un nombre determinado por el Administrador de Nagios, que será el que más convenga en cada caso. Los expuestos en los casos de uso son los nombres estándar que Nagios recomienda para su rápida identificación y para la mejor administración y mantenimiento de la aplicación.

### 3.3.1.1.1. Infraestructura (Definición)

En este punto, se va a proceder a describir los casos de uso correspondientes a la configuración de la infraestructura de Nagios, en la parte de definición de elementos desde la consola de administración del servidor contenedor del sistema de monitorización, para la posterior visualización en el entorno web de la aplicación (véase Ilustración 22).

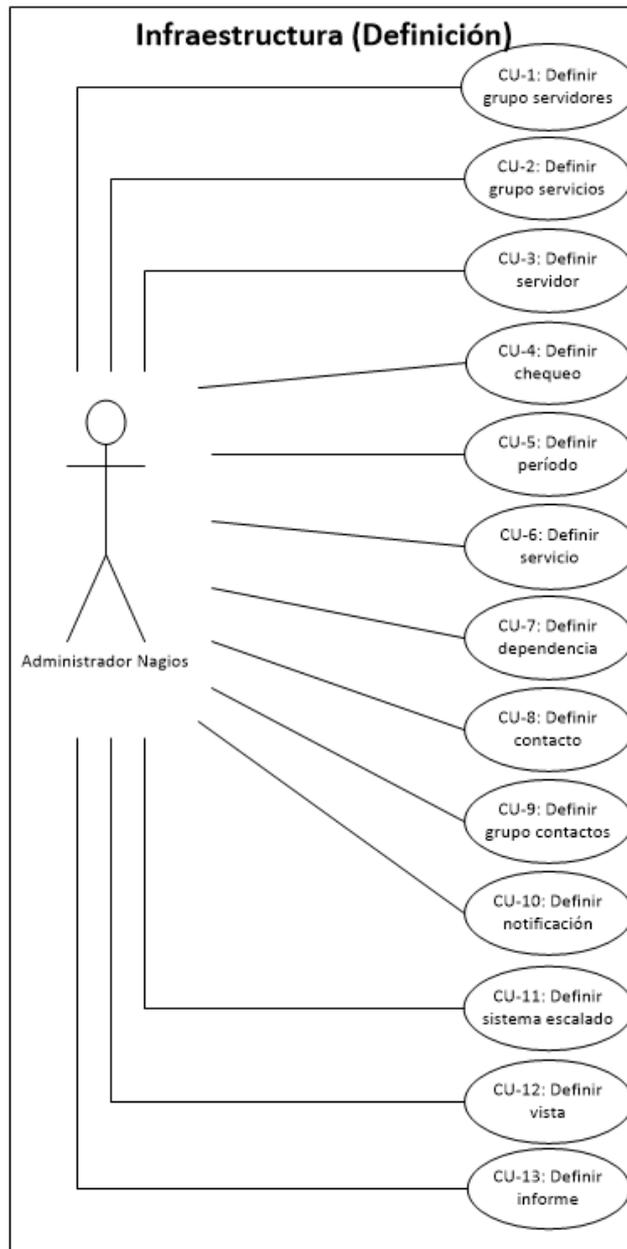


Ilustración 22.- Diagrama de casos de uso de Infraestructura (Definición)

A continuación, se definen once casos de uso, descritos gráficamente en el anterior diagrama, donde se van a especificar todos los datos necesarios para cumplir con ellos [1].

Identificador: CU-1	
Nombre	Definir un grupo de servidores
Actores	Administrador de Nagios
Objetivo	Monitorizar nuevo grupo de host
Precondiciones	Acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
Postcondiciones	Agregar host para poder visualizarse en Nagios.
Escenario básico	<ol style="list-style-type: none"> <li>1. El AdminNagios crea el fichero hostgroups.cfg en /usr/local/nagios/etc, donde registra el nombre del grupo de servidores a definir.</li> </ol>

Tabla 6.- CU-1: Definir grupo servidores

Identificador: CU-2	
Nombre	Definir un grupo de servicios
Actores	Administrador de Nagios
Objetivo	Monitorizar nuevo grupo de servicios
Precondiciones	Acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
Postcondiciones	Agregar servicios para poder visualizarse en Nagios.
Escenario básico	<ol style="list-style-type: none"> <li>1. El AdminNagios crea el fichero servicegroups.cfg en /usr/local/nagios/etc, donde registra el nombre del grupo de servicios a definir.</li> </ol>

Tabla 7.- CU-2: Definir grupo servicios

Identificador: CU-3	
Nombre	Definir un servidor
Actores	Administrador de Nagios
Objetivo	Monitorizar nuevo host
Precondiciones	Permisos de acceso e instalación del agente de Nagios en el host, acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
Postcondiciones	El host solo aparecerá monitorizado si se le define un grupo de servidores (hostgroup) y un servicio.
Escenario básico	<ol style="list-style-type: none"> <li>1. El AdminNagios accede al servidor.</li> <li>2. El AdminNagios copiamos el instalador del agente de Nagios en el host.</li> <li>3. El AdminNagios instala el agente de Nagios.</li> <li>4. El AdminNagios accede a Nagios.</li> <li>5. El AdminNagios crea el host en el archivo de configuración de Nagios llamado hosts.cfg, ubicado en /usr/local/nagios/etc.</li> <li>6. El AdminNagios asigna el host creado a un hostgroup.</li> </ol>

Tabla 8.- CU-3: Definir servidor

<b>Identificador: CU-4</b>	
<b>Nombre</b>	Definir un chequeo
<b>Actores</b>	Administrador de Nagios
<b>Objetivo</b>	Disponer de más chequeos para monitorizar cosas menos comunes.
<b>Precondiciones</b>	Acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
<b>Postcondiciones</b>	Estará creado un nuevo chequeo.
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El AdminNagios crea el fichero del chequeo, sin importar el lenguaje de programación, pero si teniendo en cuenta la configuración de salida que debe leer Nagios para poder entenderlo. La ruta de ubicación de los chequeos es /usr/local/nagios/libexec, donde están los que Nagios instala por defecto y todos los que vayamos generando deberemos almacenarlo en esa ubicación.</li> </ol>

Tabla 9.- CU-4: Definir chequeo

<b>Identificador: CU-5</b>	
<b>Nombre</b>	Definir un periodo
<b>Actores</b>	Administrador de Nagios
<b>Objetivo</b>	Disponer de tiempos de monitorización.
<b>Precondiciones</b>	Acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
<b>Postcondiciones</b>	Asignación de diferentes tiempos de monitorización.
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El AdminNagios crea el fichero timeperiods.cfg en /usr/local/nagios/etc, donde registra periodo de notificación según la nomenclatura.</li> </ol>

Tabla 10.- CU-5: Definir período

<b>Identificador: CU-6</b>	
<b>Nombre</b>	Definir un servicio
<b>Actores</b>	Administrador de Nagios
<b>Objetivo</b>	Monitorizar nuevo servicio en servidor.
<b>Precondiciones</b>	Acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
<b>Postcondiciones</b>	El servicio nuevo está monitorizado.
<b>Escenario básico</b>	<ol style="list-style-type: none"> <li>1. El AdminNagios crea el fichero services.cfg en /usr/local/nagios/etc, donde registra el nombre del servicio a definir para un servidor ya creado.</li> </ol>

Tabla 11.- CU-6: Definir servicio

Identificador: CU-7	
Nombre	Definir una dependencia
Actores	Administrador de Nagios
Objetivo	Monitorizar host por medio de dependencias
Precondiciones	Acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
Postcondiciones	La nueva dependencia estará creada.
Escenario básico	1. El AdminNagios crea el fichero dependency.cfg en /usr/local/nagios/etc, donde registra, según nomenclatura, los periodos de notificación dependiendo de las alertas de otros servicios en otros servidores.

Tabla 12.- CU-7: Definir dependencia

Identificador: CU-8	
Nombre	Definir un contacto
Actores	Administrador de Nagios
Objetivo	Creación de contacto para definir grupo de contactos.
Precondiciones	Acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
Postcondiciones	El nuevo contacto estará creado.
Escenario básico	1. El AdminNagios creará un fichero contacts.cfg en /usr/local/nagios/etc, donde registra, según nomenclatura, las direcciones de correo del dominio del servidor de correo registrado en el servidor de Nagios (sendmail).

Tabla 13.- CU-8: Definir contacto

Identificador: CU-9	
Nombre	Definir un grupo de contactos
Actores	Administrador de Nagios
Objetivo	Creación de grupo de contactos para enviar notificaciones de alertas.
Precondiciones	Acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
Postcondiciones	El nuevo grupo de contactos estará preparado para asociarlo a una notificación de alertas de un servidor y/o servicio.
Escenario básico	1. El AdminNagios creará un fichero contactsgroup.cfg en /usr/local/nagios/etc, donde registrará, según nomenclatura, nombres de grupos con los contactos que sean necesarios.

Tabla 14.- CU-9: Definir grupo contactos

Identificador: CU-10	
Nombre	Definir una notificación
Actores	Administrador de Nagios
Objetivo	Crear notificaciones para alertas de servicios y servidores.
Precondiciones	Acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
Postcondiciones	La nueva notificación de alerta estará creada.
Escenario básico	<ol style="list-style-type: none"><li>1. El AdminNagios debe generar dicha configuración en el fichero de services.cfg, donde se definió el servicio para el servidor, que siguiendo nomenclatura, completará el campo contact_groups.</li></ol>

Tabla 15.- CU-10: Definir notificación

Identificador: CU-11	
Nombre	Definir un sistema de escalados
Actores	Administrador de Nagios
Objetivo	Crear niveles de escalado para la mejor resolución de alertas.
Precondiciones	Acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
Postcondiciones	El nuevo sistema de escalado estará creado.
Escenario básico	<ol style="list-style-type: none"><li>1. El AdminNagios creará un fichero escalations.cfg en /usr/local/nagios/etc, donde registrará, según nomenclatura, nombres de grupos con los contactos que sean necesarios.</li></ol>

Tabla 16.- CU-11: Definir sistema escalado

Identificador: CU-12	
Nombre	Definir una vista
Actores	Administrador de Nagios
Objetivo	Crear vistas específicas para visualizar mejor las alertas.
Precondiciones	Acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
Postcondiciones	Las vistas estarán disponibles para los actores que necesiten hacer uso de ellas.
Escenario básico	<ol style="list-style-type: none"><li>1. El AdminNagios creará un fichero, preferiblemente en HTML para la mejor visualización del contenido en Nagios, en /usr/local/nagios/share.</li><li>2. El AdminNagios dentro del fichero creado, deberá desarrollar el código, según los ejemplos que el software proporciona, realizando llamadas a las vistas ya definidas en Nagios, con el lenguaje de programación PHP.</li></ol>

Tabla 17.- CU-12: Definir vista

<b>Identificador: CU-13</b>	
<b>Nombre</b>	Definir un informe
<b>Actores</b>	Administrador de Nagios
<b>Objetivo</b>	Crear informes para la visualización de los datos de monitorización que interesen en cada momento.
<b>Precondiciones</b>	Acceso al servidor de Nagios y definición de los nuevos ficheros en /usr/local/nagios/etc/nagios.cfg.
<b>Postcondiciones</b>	Los informes estarán disponibles para los actores que necesiten hacer uso de ellos.
<b>Escenario básico</b>	<ol style="list-style-type: none"><li>1. El AdminNagios creará un fichero, preferiblemente en HTML para la mejor visualización del contenido en Nagios, en /usr/local/nagios/etc.</li><li>2. El AdminNagios dentro del fichero creado, deberá desarrollar el código, según los ejemplos que el software proporciona, realizando llamadas a las vistas ya definidas en Nagios, con el lenguaje de programación PHP.</li></ol>

Tabla 18.- CU-13: Definir informe

### 3.3.1.1.2. Infraestructura (Modificación)

En este punto, se va a proceder a describir los casos de uso correspondientes a la configuración de la infraestructura de Nagios, en la parte de modificación de elementos desde la consola de administración del servidor contenedor del sistema de monitorización, para la posterior visualización en el entorno web de la aplicación (véase Ilustración 23).

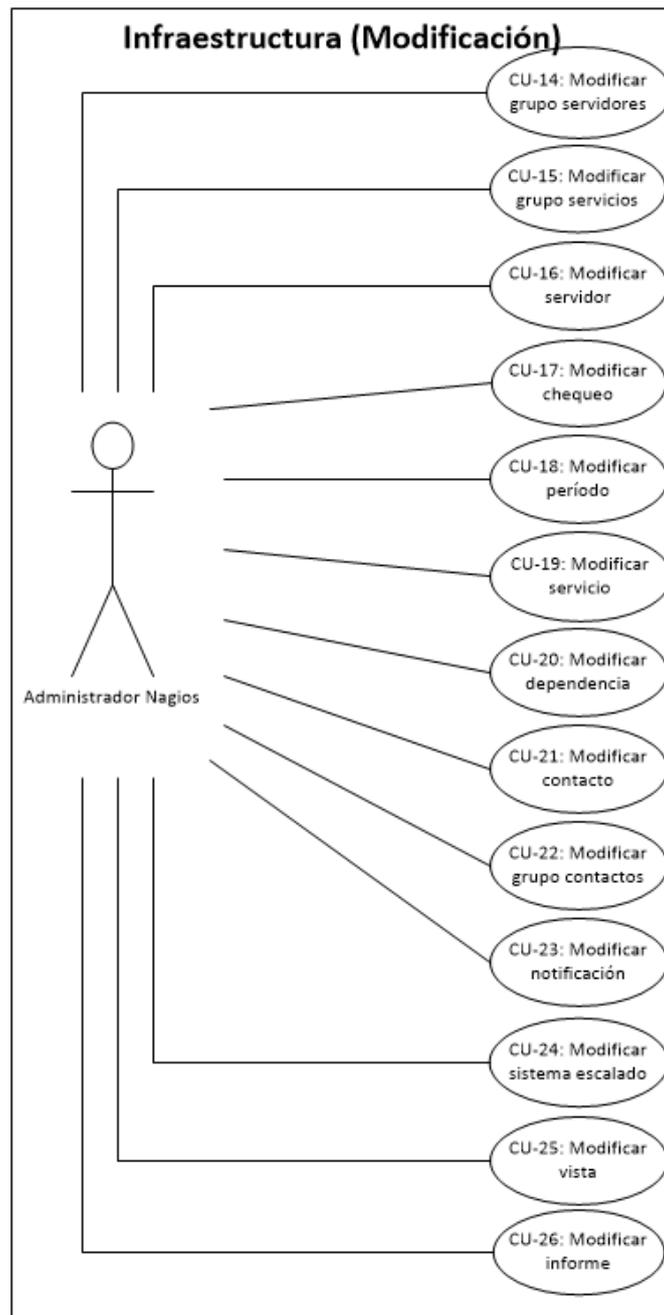


Ilustración 23.- Diagrama de casos de uso de Infraestructura (Modificación)

Identificador: CU-14	
Nombre	Modificar grupo de servidores
Actores	Administrador de Nagios
Objetivo	Modificar grupo de servidores
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Deberá ser modificado el nombre del grupo de servidores en todos los servidores definidos en el fichero host.cfg, que hagan referencia al mismo.
Escenario básico	1. El AdminNagios modificará en el fichero hostgroups.cfg el nombre del grupo.

Tabla 19.- CU-14: Modificar grupo servidores

Identificador: CU-15	
Nombre	Modificar grupo de servicios
Actores	Administrador de Nagios
Objetivo	Modificar grupo de servicios
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Deberá ser modificado el nombre del grupo de servicios en todos los servidores definidos en el fichero services.cfg, que hagan referencia al mismo.
Escenario básico	1. El AdminNagios modificará en el fichero servicegroups.cfg el nombre del grupo.

Tabla 20.- CU-15: Modificar grupo servicios

Identificador: CU-16	
Nombre	Modificar servidor
Actores	Administrador de Nagios
Objetivo	Modificar características de un servidor
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Deberá ser modificado el nombre del servidor en todas las definiciones donde esté definido el mismo, como por ejemplo en services.cfg.
Escenario básico	1. El AdminNagios modificará en el fichero hosts.cfg el nombre del servidor.

Tabla 21.- CU-16: Modificar servidor

Identificador: CU-17	
Nombre	Modificar chequeo
Actores	Administrador de Nagios
Objetivo	Modificar un chequeo definido en /usr/local/nagios/libexec.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Deberá ser comprobado que ese chequeo ahora sigue devolviendo la información que interesa en los servidores donde se esté comprobando.
Escenario básico	1. El AdminNagios modificará en el fichero correspondiente al chequeo seleccionado para realizar la acción de modificación.

Tabla 22.- CU-17: Modificar chequeo

Identificador: CU-18	
Nombre	Modificar período
Actores	Administrador de Nagios
Objetivo	Modificar un período de notificación definido en el fichero timeperiods.cfg.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Deberá ser comprobado que el cambio de período de notificación, afecta solo a la monitorización de servicios que interesa modificar.
Escenario básico	1. El AdminNagios modificará en el fichero timeperiods.cfg los tiempos de notificación para un tiempo establecido.

Tabla 23.- CU-18: Modificar período

Identificador: CU-19	
Nombre	Modificar servicio
Actores	Administrador de Nagios
Objetivo	Modificación de un servicio definido para un servidor o un grupo de servidores.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Deberá ser comprobado que el cambio de nombre o estructura de un servicio, afecta solo a la monitorización de servicios que interesa modificar.
Escenario básico	1. El AdminNagios modificará en el fichero services.cfg el servicio que interese para un servidor o un grupo de servidores.

Tabla 24.- CU-19: Modificar servicio

Identificador: CU-20	
Nombre	Modificar dependencia
Actores	Administrador de Nagios
Objetivo	Modificación de una dependencia para asignar alertas de un o unos servicios a una condición.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	-
Escenario básico	1. El AdminNagios modificará en el fichero dependency.cfg el servicio que interese modificar para la notificación de alertas, para un servidor o un grupo de servidores

Tabla 25.- CU-20: Modificar dependencia

Identificador: CU-21	
Nombre	Modificar contacto
Actores	Administrador de Nagios
Objetivo	Modificación de un contacto.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Si se modificara el nombre del contacto, es necesario usar CU-22, para modificar el nombre del contacto dentro del grupo de contactos donde esté asociado.
Escenario básico	1. El AdminNagios modificará en el fichero contacts.cfg el contacto que interese modificar.

Tabla 26.- CU-21: Modificar contacto

Identificador: CU-22	
Nombre	Modificar grupo de contactos
Actores	Administrador de Nagios
Objetivo	Modificación de un grupo de contactos.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Si se modificara el nombre de un grupo de contactos, será necesario modificarlo en todos los ficheros donde la condición de contacts_group aparezca el grupo modificado.
Escenario básico	1. El AdminNagios modificará en el fichero contactsgroup.cfg el grupo de contactos que interese modificar.

Tabla 27.- CU-22: Modificar grupo contactos

Identificador: CU-23	
Nombre	Modificar notificación
Actores	Administrador de Nagios
Objetivo	Modificación de una notificación para un servicio.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	-
Escenario básico	1. El AdminNagios modificará en el fichero services.cfg los grupos de contactos donde un servicio reporta una notificación de alerta.

Tabla 28.- CU-23: Modificar notificación

Identificador: CU-24	
Nombre	Modificar sistema de escalado
Actores	Administrador de Nagios
Objetivo	Modificar el escalado de alertas con referencia a un servidor o servicio determinado.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	-
Escenario básico	1. El AdminNagios modificará en el fichero escalations.cfg los grupos de escalados que estén definidos para poder escalar correctamente alertas dependientes de monitorización.

Tabla 29.- CU-24: Modificar sistema de escalado

Identificador: CU-25	
Nombre	Modificar vista
Actores	Administrador de Nagios
Objetivo	Modificar la vista de los servidores y servicios en Nagios.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	-
Escenario básico	1. El AdminNagios modificará el fichero dtree_data.js para la llamada a las diferentes entradas en la vista principal de Nagios.

Tabla 30.- CU-25: Modificar vista

<b>Identificador: CU-26</b>	
<b>Nombre</b>	Modificar informe
<b>Actores</b>	Administrador de Nagios
<b>Objetivo</b>	Modificar la vista de un informe en Nagios.
<b>Precondiciones</b>	Acceso al servidor de Nagios.
<b>Postcondiciones</b>	-
<b>Escenario básico</b>	1. El AdminNagios modificará en /usr/local/nagios/share, los fichero escritos en HTML para la configuración de los informes que hay predefinidos, y así poder generar los propios.

Tabla 31.- CU-26: Modificar informe

### 3.3.1.1.3. Infraestructura (Eliminación)

En este punto, se van a redactar los casos de uso correspondientes a la configuración de la infraestructura de Nagios, en la parte de eliminación de elementos desde la consola de administración del servidor contenedor del sistema de monitorización, para la posterior visualización en el entorno web de la aplicación (véase Ilustración 24).

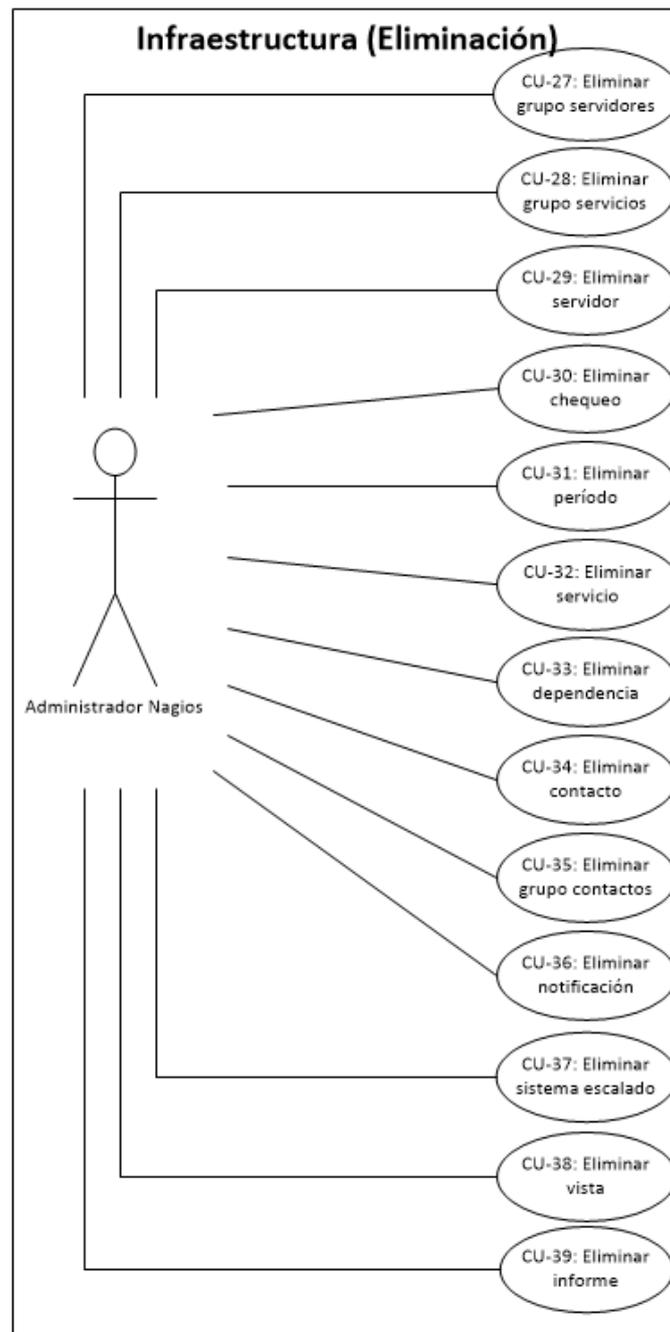


Ilustración 24.- Diagrama de casos de uso de Infraestructura (Eliminación)

Este punto tiene una alta criticidad ya que Nagios tiene un alto grado de dependencias entre nombres de servidores y servicios, y de dependencias y escalados, entre otros, por lo que será necesaria la atención máxima para no perder datos ni ocasionar errores en Nagios.

Identificador: CU-27	
Nombre	Eliminar grupo de servidores
Actores	Administrador de Nagios
Objetivo	Eliminación de un grupo de servidores de Nagios.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Observar que ningún servidor tiene asociado ese nombre de grupo.
Escenario básico	<ol style="list-style-type: none"><li>1. El AdminNagios elimina el grupo de servidores creado en el fichero hostgroups.cfg.</li><li>2. El AdminNagios elimina de las definiciones de servidores, la del grupo de servidores, en el caso de estar en alguno de ellos.</li></ol>

Tabla 32.- CU-27: Eliminar grupo de servidores

Identificador: CU-28	
Nombre	Eliminar grupo de servicios
Actores	Administrador de Nagios
Objetivo	Eliminar un grupo de servicios de Nagios.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Observar que ningún servicio tiene asociado ese nombre de grupo.
Escenario básico	<ol style="list-style-type: none"><li>1. El AdminNagios elimina el grupo de servicios creado en el fichero servicegroups.cfg.</li><li>2. El AdminNagios elimina de las definiciones de servicios, la del grupo de servicios, en el caso de estar en alguno de ellos.</li></ol>

Tabla 33.- CU-28: Eliminar grupo de servicios

Identificador: CU-29	
Nombre	Eliminar servidor
Actores	Administrador de Nagios
Objetivo	Eliminar un servidor definido en Nagios.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Observar que ningún servicio tiene asociado ese servidor.
Escenario básico	<ol style="list-style-type: none"><li>1. El AdminNagios elimina el servidor creado en el fichero hosts.cfg.</li><li>2. El AdminNagios elimina de las definiciones del servidor, y todos los servicios, dependencias y grupos, asociados a él.</li></ol>

Tabla 34.- CU-29: Eliminar servidor

Identificador: CU-30	
Nombre	Eliminar chequeo
Actores	Administrador de Nagios
Objetivo	Eliminar un chequeo de la configuración de Nagios.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Observar que ningún servicio tiene asociado ese chequeo.
Escenario básico	1. El AdminNagios eliminará un chequeo teniendo en cuenta los servicios que tengan al mismo asociado. Simplemente con eliminar el fichero de /usr/local/nagios/libexec, será eliminado el mismo.

Tabla 35.- CU-30: Eliminar chequeo

Identificador: CU-31	
Nombre	Eliminar período
Actores	Administrador de Nagios
Objetivo	Eliminar período de notificación para alertas generadas en Nagios.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Observar que ningún servicio tiene asociado ese período.
Escenario básico	1. El AdminNagios eliminará un período de notificación, en el fichero timeperiods.cfg, y simplemente deberá eliminar la definición del mismo en los ficheros services.cfg donde esté reflejado.

Tabla 36.- CU-31: Eliminar período

Identificador: CU-32	
Nombre	Eliminar servicio
Actores	Administrador de Nagios
Objetivo	Eliminar un servicio asociado a un servidor.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Observar que dicho servicio no tiene asociado algún otro campo en Nagios, como dependencias o contactos.
Escenario básico	1. El AdminNagios eliminará un servicio asociado a un servidor, en el fichero services.cfg, y siempre se tendrá en cuenta si está o no definido en algún otro fichero.

Tabla 37.- CU-32: Eliminar servicio

Identificador: CU-33	
Nombre	Eliminar dependencia
Actores	Administrador de Nagios
Objetivo	Eliminar una dependencia entre servidores y/o servicios.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	-
Escenario básico	1. El AdminNagios eliminará en el fichero dependency.cfg el registro realizado para dicha dependencia.

Tabla 38.- CU-33: Eliminar dependencia

Identificador: CU-34	
Nombre	Eliminar contacto
Actores	Administrador de Nagios
Objetivo	Eliminación de un contacto definido en Nagios.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Deberá ser eliminado, si estuviese registrado también, en un grupo de contactos.
Escenario básico	<ol style="list-style-type: none"> <li>1. El AdminNagios eliminará el contacto concreto del fichero contacts.cfg, en la ruta /usr/local/nagios/etc.</li> <li>2. El AdminNagios deberá observar si este contacto está definido en un grupo de contactos para proceder a su eliminación también. (CU-35).</li> </ol>

Tabla 39.- CU-34: Eliminar contacto

Identificador: CU-35	
Nombre	Eliminar grupo contactos
Actores	Administrador de Nagios
Objetivo	Eliminación de un grupo de contactos compuesto por contactos.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Deberá ser eliminado o modificado el nombre del grupo en los servicios que tenga asociados.
Escenario básico	<ol style="list-style-type: none"> <li>1. El AdminNagios eliminará un grupo de contactos, en el fichero contactsgroup.cfg, en la ruta /usr/local/nagios/etc, teniendo en cuenta que este grupo puede estar definido en uno o varios servicios como contacto de notificación.</li> </ol>

Tabla 40.- CU-35: Eliminar grupo contactos

Identificador: CU-36	
Nombre	Eliminar notificación
Actores	Administrador de Nagios
Objetivo	Eliminación de una notificación de alertas para servidores y servicios donde esté definido.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Deberá supervisar que esta eliminación no provoque que solo él reciba la misma.
Escenario básico	<ol style="list-style-type: none"> <li>1. El AdminNagios eliminará la notificación de una alerta en Nagios, para que no avise de un problema en un servicio y/o servidor.</li> </ol>

Tabla 41.- CU-36: Eliminar notificación

Identificador: CU-37	
Nombre	Eliminar sistema escalado
Actores	Administrador de Nagios
Objetivo	Eliminar un sistema de escalado de notificaciones.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	-
Escenario básico	<ol style="list-style-type: none"> <li>1. El AdminNagios eliminará una definición de escalado en el fichero escalations.cfg en /usr/local/nagios/etc.</li> </ol>

Tabla 42.- CU-37: Eliminar sistema escalado

<b>Identificador: CU-38</b>	
<b>Nombre</b>	Eliminar vista
<b>Actores</b>	Administrador de Nagios
<b>Objetivo</b>	Eliminará una vista de la interfaz web del servidor Nagios.
<b>Precondiciones</b>	Acceso al servidor de Nagios.
<b>Postcondiciones</b>	Se deberá comprobar que ya no es accesible, y que no está definida en el árbol principal de acceso de Nagios.
<b>Escenario básico</b>	1. El AdminNagios eliminará una vista creada manualmente en /usr/local/nagios/share, y quitará la accesibilidad del árbol principal de acceso.

Tabla 43.- CU-38: Eliminar vista

<b>Identificador: CU-39</b>	
<b>Nombre</b>	Eliminar informe
<b>Actores</b>	Administrador de Nagios
<b>Objetivo</b>	Eliminación de los informes personalizados.
<b>Precondiciones</b>	Acceso al servidor de Nagios.
<b>Postcondiciones</b>	-
<b>Escenario básico</b>	1. El AdminNagios eliminará los informes que han sido creados con algún fin, normalmente para Responsables y/o técnicos.

Tabla 44.- CU-39: Eliminar informe

### 3.3.1.1.4. Notificación

A continuación, se van a redactar los casos de uso correspondientes a la configuración de las notificaciones de Nagios, desde la consola de administración del servidor contenedor del sistema de monitorización, para la posterior administración correcta de todas las alertas generadas en la aplicación (véase Ilustración 25).

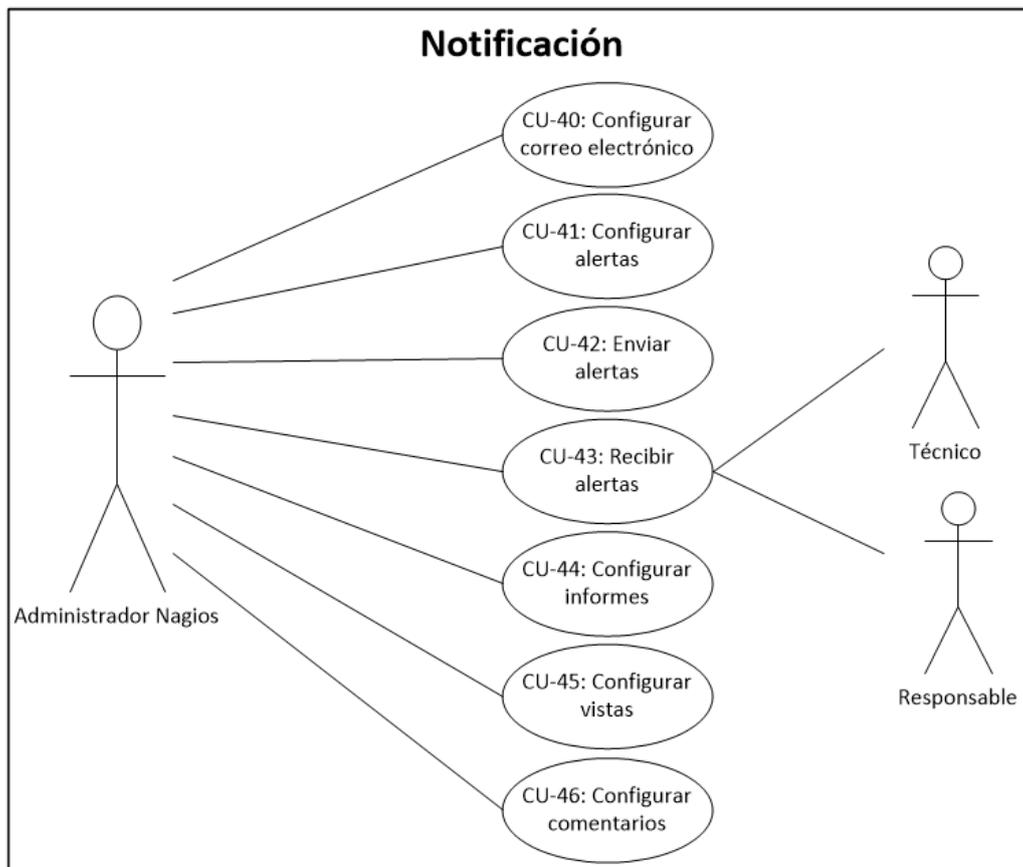


Ilustración 25.- Diagrama de casos de uso de Notificación

Identificador: CU-40	
Nombre	Configurar correo electrónico
Actores	Administrador de Nagios
Objetivo	Configuración del correo electrónico en el servidor Nagios para el envío de notificaciones a los técnicos.
Precondiciones	Acceso al servidor de Nagios y tener instalado el servicio sendmail para el envío de correo electrónico desde el servidor.
Postcondiciones	Comprobar que se envía correctamente correo electrónico desde el servidor de Nagios a los buzones de correo definidos.
Escenario básico	<ol style="list-style-type: none"><li>1. El AdminNagios configura el correo electrónico en los ficheros de configuración del servicio sendmail.cf, ubicado en /etc/mail.</li><li>2. El AdminNagios deberá buscar una línea llamada "#Smart" relay host (may be null)" y registraremos el nombre del servidor de correo a continuación de las letras DS, por ejemplo DScoreo.dominio.es.</li></ol>

Tabla 45.- CU-40: Configurar correo electrónico

Identificador: CU-41	
Nombre	Configurar alertas
Actores	Administrador de Nagios
Objetivo	Configuración de alertas de servidores y servicios.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Observar todas las alertas se ejecutan en los tiempos registrados en los ficheros de monitorización.
Escenario básico	<ol style="list-style-type: none"><li>1. El AdminNagios deberá configurar los períodos de alertas de los servicios para luego configurar los tipos de notificación y posteriormente los periodos de monitorización.</li></ol>

Tabla 46.- CU-41: Configurar alertas

Identificador: CU-42	
Nombre	Enviar alertas
Actores	Administrador de Nagios
Objetivo	Envío de alertas de problemas en la monitorización.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Observar que ningún servidor tiene asociado ese nombre de grupo.
Escenario básico	<ol style="list-style-type: none"><li>1. El AdminNagios programará el envío de alertas y se enviarán con la configuración realizada para cada chequeo de un servicio de cada servidor.</li></ol>

Tabla 47.- CU-42: Enviar alertas

Identificador: CU-43	
Nombre	Recibir alertas
Actores	Administrador de Nagios, Técnico y Responsable
Objetivo	Recepción de todas las alertas según recepcionista.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Observar que los contactos que tienen que recibir las alertas, las han recibido correctamente según la configuración realizada.
Escenario básico	1. El AdminNagios elimina el grupo de servidores creado en el fichero hostgroups.cfg.

Tabla 48.- CU-43: Recibir alertas

Identificador: CU-44	
Nombre	Configurar informes
Actores	Administrador de Nagios
Objetivo	Configuración de informes para la visibilidad de los datos en la interfaz web de Nagios.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Observar que los nuevos informes son accesibles.
Escenario básico	1. El AdminNagios define tantos ficheros HTML como informes sean necesarios, incluyéndolos en el acceso principal de Nagios para que sean accesibles a todos los usuarios, o personalizados por sesión.

Tabla 49.- CU-44: Configurar informes

Identificador: CU-45	
Nombre	Configurar vistas
Actores	Administrador de Nagios
Objetivo	Configuración de las vistas posibles en Nagios en su interfaz web gráfica.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Observar que las nuevas vistas son accesibles.
Escenario básico	1. El AdminNagios puede modificar los ficheros contenidos en /usr/local/nagios/share, para cambiar las vistas de todas las páginas webs contenidas tanto en la interfaz web principal, como en todas las subvistas.

Tabla 50.- CU-45: Configurar vistas

Identificador: CU-46	
Nombre	Configurar comentarios
Actores	Administrador de Nagios
Objetivo	Configuración de los comentarios que se crean y eliminan en Nagios, tanto por medio de la interfaz web como por consola.
Precondiciones	Acceso al servidor de Nagios.
Postcondiciones	Observar que los comentarios son accesibles, y con posibilidad de creación y eliminación.
Escenario básico	1. El AdminNagios configura el acceso a los comentarios de los servicios y servidores contenidos en Nagios para la aclaración de diferentes alertas y notificaciones.

Tabla 51.- CU-46: Configurar comentarios

### 3.3.1.1.5. Visualización

Durante este punto, se van a redactar los casos de uso correspondientes a las vistas de los datos de Nagios, desde la interfaz web, para la rápida asociación y apreciación de los datos y los resultados obtenidos tanto en el mismo segundo como en períodos de tiempos que se deseen examinar (véase Ilustración 26).

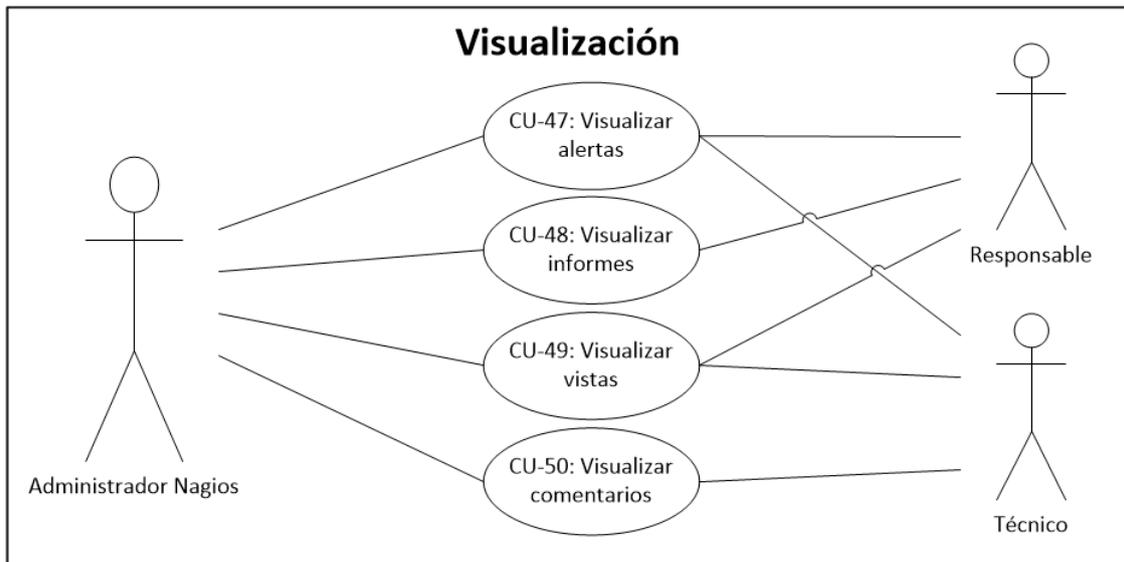


Ilustración 26.- Diagrama de casos de uso de Visualización

Los casos de uso que se van a definir a continuación, son los cuatro generales para la parte de visualización, donde se encuentran visibles los casos en los que van a poder actuar los otros dos actores existentes en cualquier organización.

Identificador: CU-47	
Nombre	Visualizar alertas
Actores	Administrador de Nagios, Técnico y Responsable
Objetivo	Visualización de las alertas en la interfaz gráfica de Nagios.
Precondiciones	Acceso web al servidor de Nagios.
Postcondiciones	Observar que se visualiza todo correctamente.
Escenario básico	<ol style="list-style-type: none"> <li>1. El AdminNagios visualiza las alertas que Nagios refleja en su interfaz web y en consola.</li> <li>2. El Técnico visualiza las alertas que Nagios refleja en su interfaz web.</li> <li>3. El Responsable visualiza las alertas que Nagios refleja en su interfaz web.</li> </ol>

Tabla 52.- CU-47: Visualizar alertas

<b>Identificador: CU-48</b>	
<b>Nombre</b>	Visualizar informes
<b>Actores</b>	Administrador de Nagios y Responsable
<b>Objetivo</b>	Visualización de informes en la interfaz web de rendimientos e histórico de alertas en Nagios.
<b>Precondiciones</b>	Acceso web al servidor de Nagios.
<b>Postcondiciones</b>	Observar que se visualiza todo correctamente.
<b>Escenario básico</b>	<ol style="list-style-type: none"><li>1. El AdminNagios visualiza los informes que Nagios refleja en su interfaz web.</li><li>2. El Responsable visualiza los informes que Nagios refleja en su interfaz web.</li></ol>

Tabla 53.- CU-48: Visualizar informes

<b>Identificador: CU-49</b>	
<b>Nombre</b>	Visualizar vistas
<b>Actores</b>	Administrador de Nagios, Técnico y Responsable
<b>Objetivo</b>	Visualizar las diferentes vistas en la interfaz web con gráficos y tablas de alertas en Nagios.
<b>Precondiciones</b>	Acceso web al servidor de Nagios.
<b>Postcondiciones</b>	Observar que se visualiza todo correctamente.
<b>Escenario básico</b>	<ol style="list-style-type: none"><li>1. El AdminNagios visualiza las vistas que Nagios refleja en su interfaz web.</li><li>2. El Técnico visualiza las vistas que Nagios refleja en su interfaz web.</li><li>3. El Responsable visualiza las vistas que Nagios refleja en su interfaz web.</li></ol>

Tabla 54.- CU-49: Visualizar vistas

<b>Identificador: CU-50</b>	
<b>Nombre</b>	Visualizar comentarios
<b>Actores</b>	Administrador de Nagios y Técnico
<b>Objetivo</b>	Visualizar los diferentes comentarios en la interfaz web en tablas en Nagios.
<b>Precondiciones</b>	Acceso web al servidor de Nagios.
<b>Postcondiciones</b>	Observar que se visualiza todo correctamente.
<b>Escenario básico</b>	<ol style="list-style-type: none"><li>1. El AdminNagios visualiza los comentarios que Nagios refleja en su interfaz web.</li><li>2. El Técnico visualiza los comentarios que Nagios refleja en su interfaz web.</li></ol>

Tabla 55.- CU-50: Visualizar comentarios

### 3.3.2. Especificación de requisitos

Una vez que se han definido los casos de uso y los escenarios, se pretenden analizar las especificaciones de requisitos para estudiar en una profundidad mucho mayor.

La especificación de requisitos que se va a realizar, está dividida en dos partes, donde se analizan los requisitos funcionales y no funcionales de la aplicación. Se definen en una tabla específica, donde los campos que han sido completados, se dan a conocer con la mayor definición y exactitud para su fácil entendimiento.

Identificador: RF-XXX / RNF-XXX	
Título	
Tipo	Funciona/No funcional
Prioridad	Baja/Media/Alta
Necesidad	Imprescindible/Opcional
Estabilidad	Baja/Media/Alta
Descripción	

Tabla 56.- Tabla de ejemplo de Requisitos funciones y no funcionales

Dónde:

- Título: es la descripción abreviada del requisito.
- Tipo: determina si es funcional o no funcional.
- Prioridad: determina la urgencia con la que se necesita el mismo.
- Necesidad: determina la suficiencia del requisito en el sistema.
- Estabilidad: determina la seguridad que tiene dicho requisito.
- Descripción: es el comentario ampliado del título.

### 3.3.2.1.Requisitos funcionales

En este punto del TFG se va proceder a hablar de los requisitos que debe cumplir el proyecto.

Identificador: RF-01	
Título	Introducir dirección IP.
Tipo	Funcional.
Prioridad	Alta.
Necesidad	Imprescindible.
Estabilidad	Alta.
Descripción	Se deberá disponer de la dirección IP para poder monitorizarla y conectarse a ella.

Tabla 57.- RF-01: Introducir dirección IP

Identificador: RF-02	
Título	Introducir datos de inicio de sesión.
Tipo	Funcional.
Prioridad	Alta.
Necesidad	Imprescindible.
Estabilidad	Alta.
Descripción	Se deberá disponer de los datos de acceso a los servidores para poder ser monitorizados.

Tabla 58.- RF-02: Introducir datos de inicio de sesión

Identificador: RF-03	
Título	Interfaz de acceso remoto a Nagios.
Tipo	Funcional.
Prioridad	Alta.
Necesidad	Imprescindible.
Estabilidad	Alta.
Descripción	Se deberá disponer de acceso remoto a Nagios para poder desarrollar toda la infraestructura del sistema de monitorización.

Tabla 59.- RF-03: Interfaz de acceso remoto a Nagios

Identificador: RF-04	
Título	Acceso al servidor de Nagios como administrador.
Tipo	Funcional.
Prioridad	Alta.
Necesidad	Imprescindible.
Estabilidad	Alta.
Descripción	Se deberá disponer del pleno acceso al servidor de Nagios para poder administrarlo remotamente desde un interfaz gráfico o de consola.

Tabla 60.- RF-04: Acceso al servidor de Nagios como administrador

<b>Identificador: RF-05</b>	
<b>Título</b>	Pantalla de estado de las alertas.
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá proporcionar la interfaz gráfica para poder visualizar todos los estados de las alertas que surjan durante los periodos de tiempo establecidos.

Tabla 61.- RF-05: Pantalla de estado de las alertas

<b>Identificador: RF-06</b>	
<b>Título</b>	Consultar datos de históricos
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Media.
<b>Necesidad</b>	Opcional.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá tener opción a consultar elementos de históricos monitorizados para tener todos los elementos controlados en todo momento.

Tabla 62.- RF-06: Consulta datos de históricos

<b>Identificador: RF-07</b>	
<b>Título</b>	Consultar datos de históricos gráficamente
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Media.
<b>Necesidad</b>	Opcional.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá tener opción a consultar elementos de históricos monitorizados gráficamente para tener todos los elementos controlados en todo momento.

Tabla 63.- RF-07: Consulta datos de históricos gráficamente

<b>Identificador: RF-08</b>	
<b>Título</b>	Opción de añadir nuevo servidor a monitorizar
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá disponer de la dirección IP o el nombre DNS del servidor para poder monitorizar el servidor. Se configurará el servidor para poder monitorizarlo lo más completo posible.

Tabla 64.- RF-08: Opcion de añadir nuevo servidor a monitorizar

<b>Identificador: RF-09</b>	
<b>Título</b>	Opción de añadir nuevo servicio a monitorizar
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá disponer de un servidor monitorizado para agregarle nuevos servicios. Existiendo el servidor, se deberá tener acceso al mismo para poder instalar el agente de Nagios y proceder a definir chequeos nuevos.

Tabla 65.- RF-09: Opción de añadir nuevo servicio a monitorizar

<b>Identificador: RF-10</b>	
<b>Título</b>	Opción de desactivar monitorización sobre servidor
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Opcional.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá disponer de una opción para poder desactivar el proceso de monitorización sobre un servidor en concreto, en el caso en el que sea necesaria su desactivación y no su eliminación.

Tabla 66.- RF-10: Opción de desactivar monitorización sobre servidor

<b>Identificador: RF-11</b>	
<b>Título</b>	Opción de desactivar monitorización sobre servicio
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá disponer de servicios monitorizados en un servidor. Existiendo el servicio, se deberá disponer de una opción para poder desactivar el proceso de monitorización sobre un servicio en concreto, en el caso en el que sea necesaria su desactivación y no su eliminación.

Tabla 67.- RF-11: Opción de desactivar monitorización sobre servicio

<b>Identificador: RF-12</b>	
<b>Título</b>	Opción de activar monitorización sobre servidor
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá disponer de una opción para poder activar el proceso de monitorización sobre un servidor en concreto, en el caso en el que sea necesaria su activación.

Tabla 68.- RF-12: Opción de activar monitorización sobre servidor

<b>Identificador: RF-13</b>	
<b>Título</b>	Opción de activar monitorización sobre servicio
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá disponer de servicios monitorizados en un servidor. Existiendo el servicio, se deberá disponer de una opción para poder activar el proceso de monitorización sobre un servicio en concreto, en el caso en el que sea necesaria su activación.

Tabla 69.- RF-13: Opción de activar monitorización sobre servicio

<b>Identificador: RF-14</b>	
<b>Título</b>	Opción de configurar grupos de servidores
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá tener la opción de crear/modificar/eliminar los grupos de servidores para poder ser visualizados de forma correcta en el entorno web de Nagios.

Tabla 70.- RF-14: Opción de configurar grupos de servidores

<b>Identificador: RF-15</b>	
<b>Título</b>	Opción de configurar grupos de servicios
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá tener la opción de crear/modificar/eliminar los grupos de servidores para poder ser visualizados de forma correcta en el entorno web de Nagios.

Tabla 71.- RF-15: Opción de configurar grupos de servicios

<b>Identificador: RF-16</b>	
<b>Título</b>	Opción de configurar alertas
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá tener la opción de crear/modificar/eliminar las alertas definidas para los grupos definidos en Nagios para su control.

Tabla 72.- RF-16: Opción de configurar alertas

<b>Identificador: RF-17</b>	
<b>Título</b>	Opción de desactivar alertas
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá disponer de una opción para poder desactivar el las alertas sobre un servidor o servicio, en el caso en el que sea necesaria su desactivación y no su eliminación.

Tabla 73.- RF-17: Opción de desactivar alertas

<b>Identificador: RF-18</b>	
<b>Título</b>	Opción de activar alertas
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá disponer de una opción para poder activar el las alertas sobre un servidor o servicio, en el caso en el que sea necesaria su activación.

Tabla 74.- RF-18: Opción de activar alertas

<b>Identificador: RF-19</b>	
<b>Título</b>	Opción de definir notificación
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá disponer de una opción para poder definir notificaciones para un servidor existente en el sistema de monitorización, y poder notificar todas las alertas generadas sobre él mismo.

Tabla 75.- RF-19: Opción de definir notificación

<b>Identificador: RF-20</b>	
<b>Título</b>	Opción de enviar notificaciones por correo electrónico
<b>Tipo</b>	Funcional.
<b>Prioridad</b>	Alta.
<b>Necesidad</b>	Imprescindible.
<b>Estabilidad</b>	Alta.
<b>Descripción</b>	Se deberá disponer de una opción para poder enviar correo electrónico a través de un servidor de correo, donde Nagios será el servidor remitente y usará un servidor como Relay para el reenvío de los mismos en una red.

Tabla 76.- RF-20: Opción de enviar notificaciones por correo electrónico

### 3.3.2.2.Requisitos no funcionales

A continuación, se van a definir los requisitos no funcionales necesarios para el correo funcionamiento del sistema de monitorización Nagios, para

Identificador: RNF-01	
Título	Acceso a Nagios
Tipo	No funcional.
Prioridad	Alta.
Necesidad	Imprescindible.
Estabilidad	Alta.
Descripción	Se deberá disponer de acceso al servidor de Nagios donde se almacena el sistema de monitorización.

Tabla 77.- RNF-01: Acceso a Nagios

Identificador: RNF-02	
Título	Acceso de Nagios a toda la red
Tipo	No funcional.
Prioridad	Alta.
Necesidad	Imprescindible.
Estabilidad	Alta.
Descripción	Se deberá disponer de acceso a toda la red desde el servidor de Nagios, para poder monitorizar todos los servidores y servicios incluidos en ella.

Tabla 78.- RNF-02: Acceso de Nagios a toda la red

Identificador: RNF-03	
Título	Acceso a servidores monitorizados
Tipo	No funcional.
Prioridad	Alta.
Necesidad	Imprescindible.
Estabilidad	Alta.
Descripción	Se deberá disponer de acceso a los servidores que se quieran monitorizar, ya que se necesitará instalar el agente de Nagios para dicho fin.

Tabla 79.- RNF-03: Acceso a servidores monitorizados

Identificador: RNF-04	
Título	Agente de Nagios para sistemas Windows
Tipo	No funcional.
Prioridad	Alta.
Necesidad	Imprescindible.
Estabilidad	Alta.
Descripción	Se deberá disponer de la aplicación NSClient++ necesaria para instalar el agente de Nagios en los servidores con Windows, indistintamente de la versión, y el acceso remoto a los mismos, para poder instalar el agente y así enviar los datos del estado de los servidores y servicios a Nagios.

Tabla 80.- RNF-04: Agente de Nagios para sistemas Windows

Identificador: RNF-05	
Título	Agente de Nagios para sistemas Linux/Unix
Tipo	No funcional.
Prioridad	Alta.
Necesidad	Imprescindible.
Estabilidad	Alta.
Descripción	Se deberá disponer de los scripts compilados NSCA necesarios para poder ejecutar el agente de Nagios en los servidores con Unix, teniendo en cuenta la versión ya que existen varios tipos de compilación, y el acceso remoto a los mismos, para poder ejecutar el agente (por medio de cron) y así enviar los datos del estado de los servidores y servicios a Nagios.

Tabla 81.- RNF-05: Agente de Nagios para sistemas Linux/Unix

Identificador: RNF-06	
Título	Agente de Nagios en servidores
Tipo	No funcional.
Prioridad	Alta.
Necesidad	Imprescindible.
Estabilidad	Alta.
Descripción	Se deberá disponer un agente de Nagios en cada servidor monitorizado para que la información del estado de los servicios sea reportada a Nagios.

Tabla 82.- RNF-06: Agente de Nagios en servidores

Identificador: RNF-07	
Título	Backup de ficheros de configuración Nagios
Tipo	No funcional.
Prioridad	Alta.
Necesidad	Imprescindible.
Estabilidad	Alta.
Descripción	Se deberá disponer de un backup de todos los fichero de configuración ubicados en /usr/local/nagios, para poder disponer de un backup actualizado cada vez que se realiza un cambio en el servidor.

Tabla 83.- RNF-07: Backup de ficheros de configuración de Nagios

Identificador: RNF-08	
Título	Identificación alertas
Tipo	No funcional.
Prioridad	Alta.
Necesidad	Imprescindible.
Estabilidad	Alta.
Descripción	Se deberá informar de los tipos de alertas que Nagios puede notificar. Existen 4 estados CRITICAL, WARNING, OK y UNKNOWN, que serán reportados en rojo, amarillo, verde y gris, respectivamente.

Tabla 84.- RNF-08: Identificación de alertas

### 3.3.3. Matriz de trazabilidad

A continuación, se muestra la matriz de trazabilidad entre requisitos y casos de uso para el este proyecto (véase Ilustración 27).

REQ/ CU	RF-1	RF-2	RF-3	RF-4	RF-5	RF-6	RF-7	RF-8	RF-9	RF-10	RF-11	RF-12	RF-13	RF-14	RF-15	RF-16	RF-17	RF-18	RF-19	RF-20	RNF-1	RNF-2	RNF-3	RNF-4	RNF-5	RNF-6	RNF-7	RNF-8
CU-1														X							X							
CU-2																						X						
CU-3	X							X														X	X	X	X	X	X	
CU-4																						X						
CU-5																						X						
CU-6	X								X						X							X				X		
CU-7																						X						
CU-8																						X						
CU-9																						X						
CU-10												X	X		X	X	X				X	X		X				
CU-11											X											X						
CU-12																						X					X	
CU-13																						X						
CU-14				X										X								X						
CU-15															X							X						
CU-16								X	X													X						X
CU-17																	X					X						
CU-18							X															X						
CU-19					X																	X						
CU-20			X						X													X						
CU-21																			X			X			X	X		
CU-22																				X		X						
CU-23												X	X							X	X							
CU-24	X	X																				X					X	
CU-25						X																X					X	
CU-26																						X						
CU-27															X							X						
CU-28								X														X						
CU-29																						X						
CU-30											X																	X
CU-31																X												
CU-32		X										X											X	X				
CU-33														X														
CU-34														X														
CU-35								X														X						
CU-36		X										X										X						
CU-37									X													X						
CU-38																						X				X		
CU-39									X													X						
CU-40														X	X							X						
CU-41							X							X	X							X		X				
CU-42		X																				X						
CU-43							X															X						
CU-44											X	X																
CU-45																X												
CU-46				X	X																X							
CU-47							X															X						
CU-48		X					X											X										
CU-49																						X						
CU-50																					X							

Ilustración 27.- Matriz de trazabilidad de casos de uco y requisitos

### 3.3.4. Diagramas de navegación

A continuación, se va a desarrollar un conjunto de diagramas de navegación de la aplicación, donde se mostrará el funcionamiento de Nagios a nivel sistema, para el entendimiento del paso de los datos entre los diferentes complementos instalados en el servidor de monitorización.

El primer diagrama que se va a presentar, es uno del funcionamiento que Nagios proporciona al más bajo nivel. Como se muestra “Nagios” recibe información de los “Dispositivos a Monitorear” y son enviados al servidor de monitorización, que “Recopila, Almacena y Formatea los datos” que se presentan en la interfaz WEB, y son procesados por el módulo “PNP4NAGIOS” para poder representar los datos en “Gráficas y tablas de medición” (véase Ilustración 28).

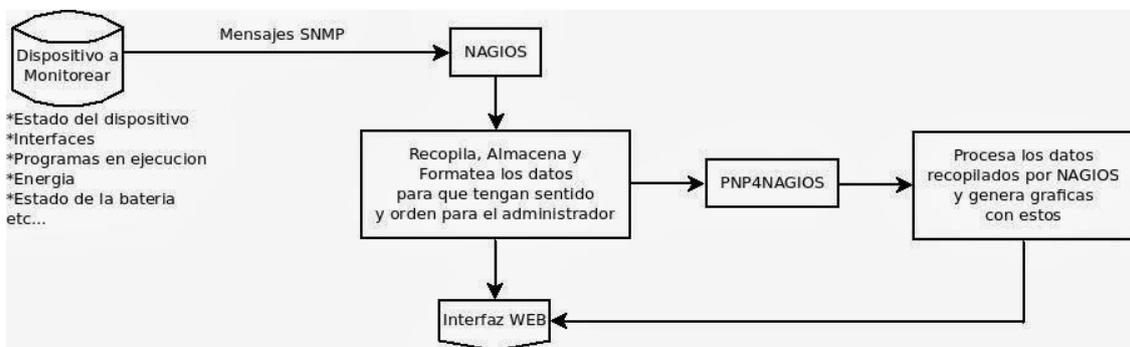


Ilustración 28.- Diagrama de navegación interna en servidor Nagios

Se comentan todos los pasos que realiza el sistema de monitorización Nagios para poder comunicar todos los procesos existentes y necesario que tiene instalados el servidor.

Dentro del servidor de monitorización se tratará toda la información de los chequeos que sean recibidos, para poder exponerlos en las diferentes vistas que el administrador y usuario podrán visualizar, en cualquiera de las pantallas que se definirán en el [Manual de Administrador \(Nagios Core 4\)](#) y [Manual de Usuario](#), adjuntos en el apéndice de la presente memoria.

Todos los chequeos, serán recepcionados por Nagios, siempre que éstos estén dados de alta en la aplicación, de tal forma que sea posible su visualización, tanto gráficamente como en informes y tablas.

### 3.3.5. Esquemas de base de datos

La base de datos de Nagios se instalará en la ejecución del paquete de instalación de Nagios, siendo MySQL.

Nagios usa una serie de módulos, definidos por NDOUtils, que instala también en el paquete de instalación del software de monitorización. Todos estos paquetes sirven para administrar la base de datos, ya que se instalan automáticamente, y hacen que sea mucho más fácil su administración y mantenimiento (véase Ilustración 29).

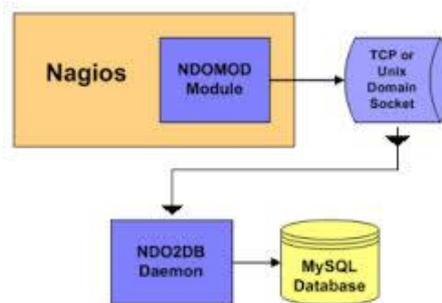


Ilustración 29.- Estructura base de datos MySQL para Nagios

NDOUtils [23] [25] es una herramienta complementaria, que sirve para leer el log de Nagios y hacer que la base de datos MySQL sea accesible. En dicha base de datos se tienen almacenados: comandos, servidores, servicios, notificaciones, contactos, el sistema de escalado, períodos de notificación, etc.

Haciendo uso de NDOUtils junto con Nagios, permite almacenar la configuración, respuestas de servidores, test de conectividad a servicios y resultados obtenidos de todos los chequeos en la base de datos. Es altamente recomendable disponer este complemento bien configurado, ya que observaremos en el punto de “Trabajos futuros” que será necesario tener accesibles los datos en todo momento para ser presentados en todas las herramientas complementarias que se instalen en el servidor de Nagios, para poder mostrar los datos de diferentes formas, que como comentaremos en dicho punto, es necesaria para mostrar datos en herramientas como NagiosQL o NagVis.

### 3.4. Implantación

En el presente capítulo se van a describir todos los pasos necesarios para la implantación y correcto funcionamiento del sistema de monitorización Nagios.

Para poder desplegar la aplicación de Nagios será necesario desde un principio conocer todos los ficheros que son imprescindibles para su instalación y configuración.

En el apéndice se ha adjuntado un [Manual de Administración](#), donde se detalla toda la configuración necesaria para desplegar Nagios, y se comentan todos los ficheros que se deben modificar para poder acceder a la interfaz web de Nagios y poder administrar el mismo conectándose por ssh (puerto 22) al servidor donde se encuentra instalado.

Una vista de los ficheros de configuración (véase Ilustración 30) donde se exponen todos los que se despliegan desde un inicio en la instalación, y que son necesarios para construir el sistema.

Los ficheros importantes en los que se debe tocar con mucha precaución, son los que aparecen recuadrados en la parte blanca superior, los cuales competen a los que definen todas las llamadas a ficheros binarios y ejecutables para que aplicación preste el correcto servicio.

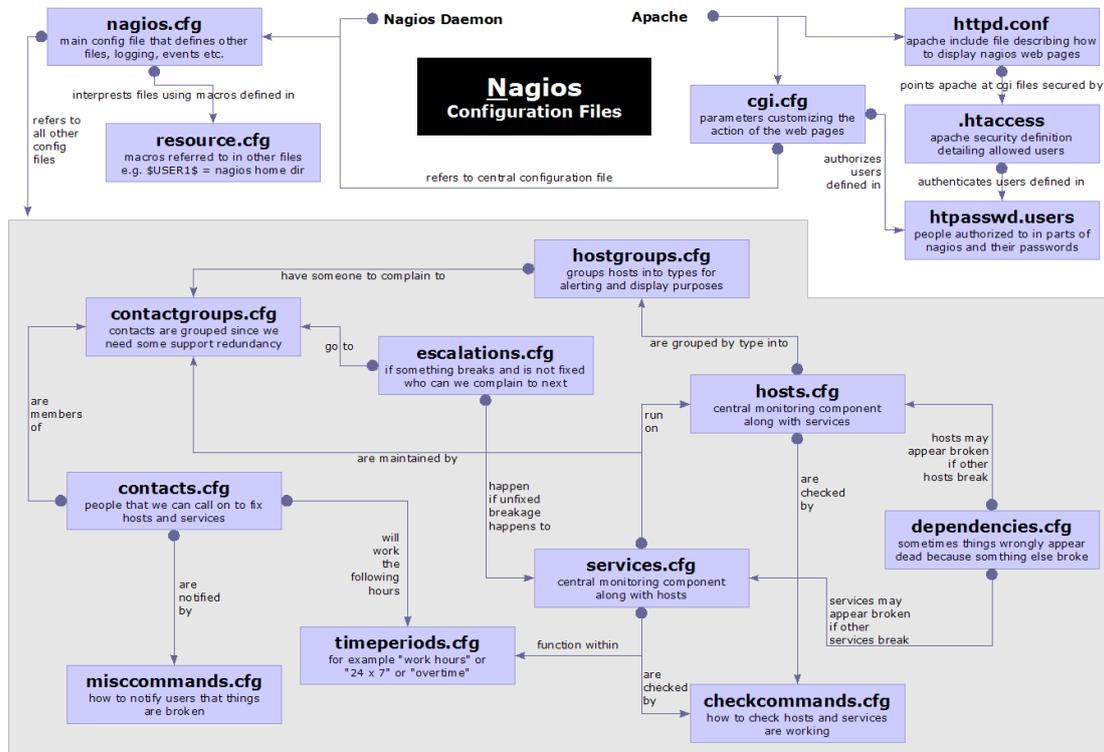


Ilustración 30.- Ficheros de configuración de Nagios en servidor Linux

Dentro del recuadro gris de la parte inferior de la ilustración anterior, son todos aquellos que el administrador de Nagios deberá modificar para poder visualizar los servidores y servicios como se desee en la interfaz web, las dependencias ente ellos, los contactos a notificar durante las alertas que se reciban, los comandos que se deseen ejecutar para cada uno de ellos, y los períodos de notificación para los diferentes contactos.

La estructura lógica del sistema de Nagios, está compuesto por un complejo número de eventos que son necesarios para su ejecución y recepción de chequeos (véase Ilustración 31). En la imagen se pueden percibir el gran número de operaciones que son necesarias para que toda la información en Nagios esté completamente estable, que ésta se registre correctamente y sea posible su visualización desde la interfaz web con todos los datos necesarios.

Como se ha explicado en puntos anteriores y se detallará en el manual de administrador, será necesario programar manualmente prácticamente toda la parte lógica de la aplicación (capas grises de la imagen siguiente), para poder visualizar toda la información tanto dentro del servidor como en la parte gráfica disponible.

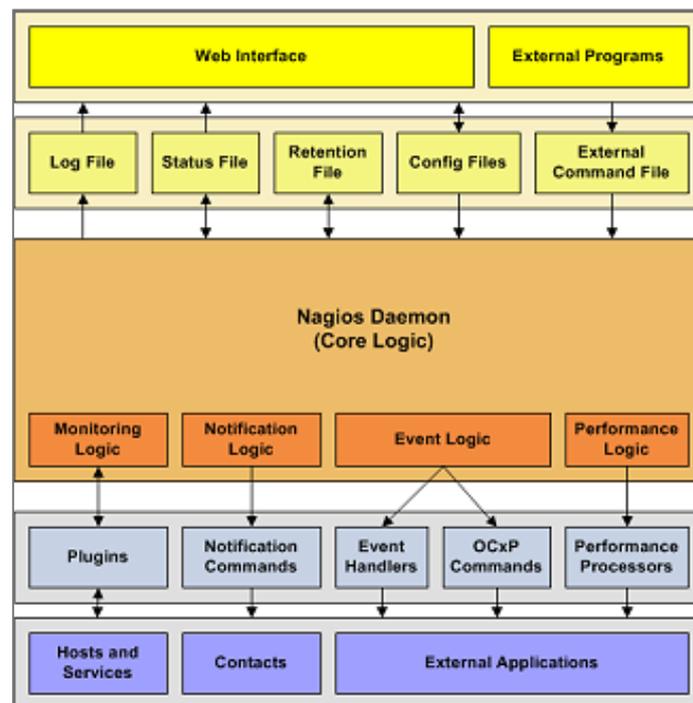


Ilustración 31.- Funcionamiento de Nagios (estructura lógica del sistema)

En el proceso de configuración del servidor de Nagios es necesario disponer de un servidor capaz de almacenar y mantener toda la infraestructura que necesita (véase Ilustración 31), donde se puede observar el Demonio de Nagios y todos los complementos y accesos que son necesarios para su correcto funcionamiento. Una vez que se haya realizado la instalación completa del sistema, se deberá disponer de esta organización, y se mantendrá durante todo el período de vida del mismo.

Un caso real de una red de sistemas informáticos monitorizados con Nagios, podría ser la siguiente, donde el “Servidor de Monitorización Nagios” dispone de acceso completo a toda la red donde esté ubicado el servidor, para que pueda tener conectividad a todos los servicios del mismo (véase Ilustración 32).

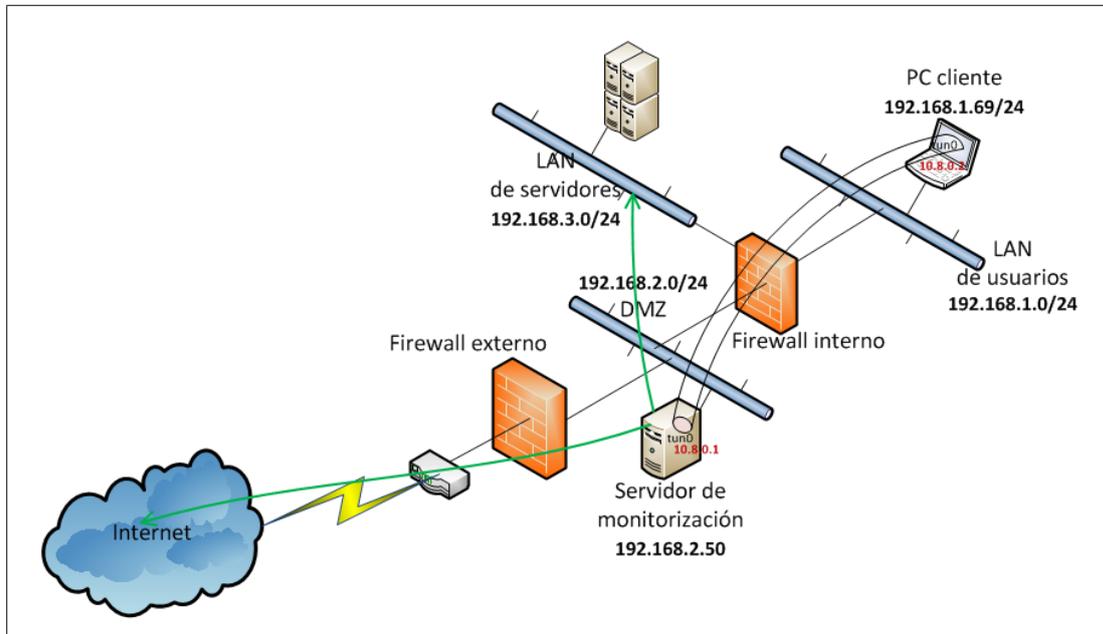


Ilustración 32.- Caso real de monitorización con Nagios

En la ilustración anterior, se puede observar cómo se coloca Nagios entre los dos Firewalls de seguridad de una organización, para poder tener acceso a todos los elementos, tanto dentro como fuera de la red. Para este tipo de casos, es necesaria la autorización de las conexiones de Nagios a los servidores, y la de los servidores a Nagios, ya que si no los paquetes de envío y recepción de información serán “cortados” por los firewalls de la red.

Dentro de una DMZ es necesario establecer una configuración de seguridad de la red, para poder acceder a los servicios a monitorizar, y no obtener errores de pérdida de paquetes y/o problemas de comunicación. Se recomienda la instalación de Nagios en la zona segura para impedir intentos de accesos indeseados a la red.

En el apéndice del documento se adjunta un [Manual de Administrador \(Nagios Core 4\)](#) donde se detallan todas las recomendaciones de instalación, configuración y seguridad que son recomendadas, para la mejor administración y funcionamiento del mismo.

Para definir el correcto funcionamiento de Nagios, se ha generado un diagrama (véase Ilustración 33), de todos los pasos que realiza Nagios para entender y mostrar todos sus datos, donde se puede visualizar la actuación del sistema en caso de alerta y en caso de estar todo correcto.

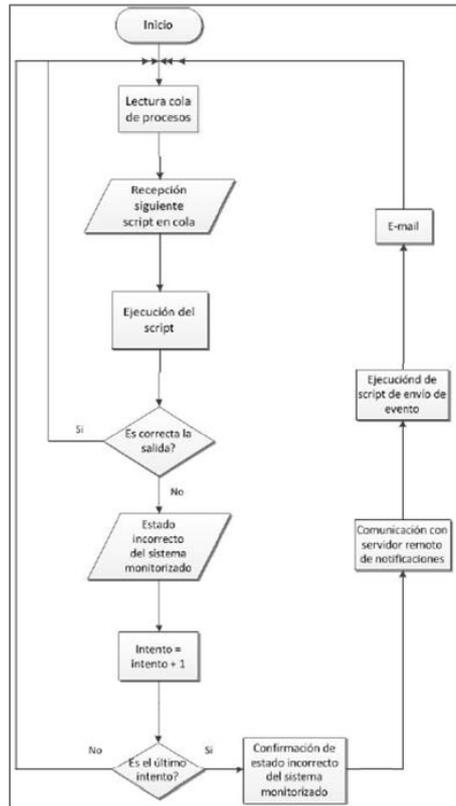


Ilustración 33.- Funcionamiento en pasos/niveles de Nagios

Como se puede observar en la ilustración anterior, comienza el proceso de monitorización, por ejemplo en la monitorización de un servicio de un servidor con sistema operativo Windows, donde se leerá la cola de proceso que realizará la ejecución del script, y dependiendo de la salida que tenga el chequeo, se volverá a ejecutar el proceso de forma normal, o se procederá a realizar una serie de intentos que se determinan por el tipo de chequeo, hasta que llegue al último (normalmente son máximo tres), donde se procederá a comunicar, en nuestro caso por correo electrónico, el estado incorrecto del servicio que se está monitorizando en el servidor Windows.

## **Capítulo 4: Evaluación**

---

## 4. Evaluación

En este capítulo se van a realizar en dos partes, dos diferentes pruebas de evaluación, para dar a conocer el rendimiento y verificar el correcto funcionamiento del sistema de monitorización Nagios en una red donde se desean monitorizar servidores y servicios con una cierta criticidad y necesidad de su funcionamiento estable y constante.

Se van a evaluar las condiciones y especificaciones que se consideran necesarias para cumplir con los objetivos marcados, teniendo en cuenta multitud de circunstancias, tanto negativas como positivas, a diferentes niveles en una arquitectura por capas.

### 4.1. Pruebas de verificación

En este punto se va a verificar que el sistema cumple con las condiciones marcadas por el objetivo de monitorizar servidores y servicios de forma constante y estable. Se va a comprobar que tanto las características como la funcionalidad especificadas en los requisitos del proyecto, cumplen completamente su desarrollo.

Para comprobar que funciona correctamente Nagios, y que la instalación del mismo ha sido correcta, nos deberá aparecer la pantalla de inicio (véase Ilustración 34).



Ilustración 34.- Pantalla inicial de Nagios con inicio correcto en web

Si nos saliese otra pantalla (véase Ilustración 35), se deberá revisar la instalación de Nagios y comprobar si el servicio del mismo está corriendo en el servidor.

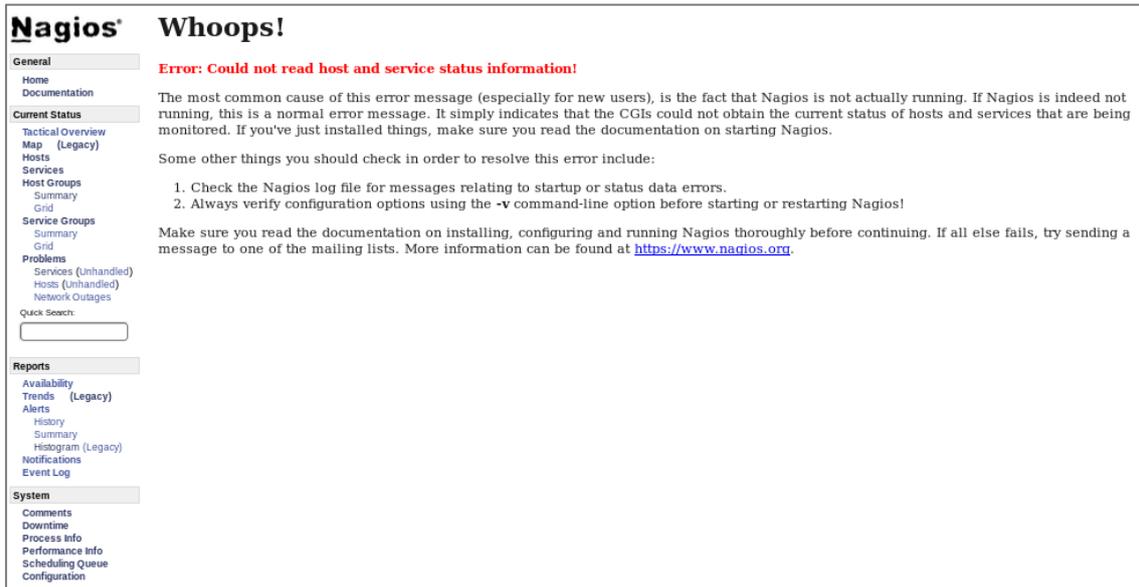


Ilustración 35.- Error de inicio de proceso de Nagios web

La tabla que se va a usar en este punto, va a ser la siguiente:

Identificador: PV-XX	
Título	
Descripción	
Requisitos relacionados	
Procedimiento	
Criterio de aceptación	

Tabla 85.- Tabla de ejemplo de Pruebas de verificación

A continuación se definen los procedimientos que se van a realizar para verificar que se han cumplido los requisitos establecidos, relacionándolos con los requisitos establecidos en los puntos anteriores, y teniendo en cuenta todos los datos que han sido presentados del funcionamiento real de un sistema de monitorización Nagios, en una red.

<b>Identificador: PV-01</b>	
<b>Título</b>	Consultar el histórico de alertas de un elemento.
<b>Descripción</b>	Se comprueba que Nagios muestra el estado de las alertas y la respuesta de los servicios monitorizados por él mismo.
<b>Requisitos relacionados</b>	RF-02, RF-05, RF-06
<b>Procedimiento</b>	<ol style="list-style-type: none"> <li>1. Acceso al servidor de Nagios mediante un navegador, introduciendo la URL "http://nagios/nagios" e identificarse con el usuario y la contraseña definidos.</li> <li>2. Seleccionar en el grupo de "Reports", la pestaña de "Alerts" y a continuación pulsar "History".</li> <li>3. Aparecerá un menú llamado "Type" donde se podrá elegir por "Host" o "Service". <ol style="list-style-type: none"> <li>a. Si pulsa en "Host" aparecerá un desplegable para elegir el servidor que queramos visualizar.</li> <li>b. Si pulsamos en "Service" aparecerá un desplegable con todos los servidores y servicios relacionados a éstos.</li> </ol> </li> <li>4. Una vez seleccionada la opción necesaria, se podrán observar en una tabla y un pequeño gráficos los datos de monitorización de un servidor o servicio concreto.</li> </ol>
<b>Criterio de aceptación</b>	Se muestra una tabla y una pequeña gráfica con el histórico de alertas que ha generado un servidor o servicio monitorizado concreto.

Tabla 86.- PV-01: Configura el histórico de alertas de un elemento

<b>Identificador: PV-02</b>	
<b>Título</b>	Consultar el histórico de alertas de un elemento gráficamente
<b>Descripción</b>	Se comprueba que Nagios muestra el estado de las alertas y la respuesta de los servicios monitorizados por él mismo.
<b>Requisitos relacionados</b>	RF-02, RF-07
<b>Procedimiento</b>	<ol style="list-style-type: none"> <li>1. Acceso al servidor de Nagios mediante un navegador, introduciendo la URL "http://nagios/nagios" e identificarse con el usuario y la contraseña definidos.</li> <li>2. Seleccionar en el grupo de "Reports", la pestaña de "Availability".</li> <li>3. Aparecerá un menú llamado "Type" donde se podrá elegir por "Host" o "Service". <ol style="list-style-type: none"> <li>a. Si pulsa en "Host" aparecerá un desplegable para elegir el servidor que queramos visualizar.</li> <li>b. Si pulsamos en "Service" aparecerá un desplegable con todos los servidores y servicios relacionados a éstos.</li> </ol> </li> <li>4. Una vez seleccionada la opción necesaria, se podrá configurar el tipo de gráfica que se desee por fechas y períodos.</li> </ol>
<b>Criterio de aceptación</b>	Se muestra una gráfica con el histórico de alertas que ha generado un servidor o servicio monitorizado concreto.

Tabla 87.- PV-02: Consulta el histórico de alertas gráficamente

<b>Identificador: PV-03</b>	
<b>Título</b>	Agregar un servidor a monitorizar
<b>Descripción</b>	Se comprueba que Nagios muestra el nuevo servidor a monitorizar con los servicios que se hayan definido para él.
<b>Requisitos relacionados</b>	RF-03, RF-04, RF-01, RF-08, RF-14, RNF-01, RNF-02, RNF-03, RNF-04, RNF-05, RNF-06
<b>Procedimiento</b>	<ol style="list-style-type: none"> <li>1. Acceso al servidor de Nagios mediante el puerto 22 del protocolo SSH, para acceder a los ficheros de configuración en /usr/local/nagios/etc.</li> <li>2. Abrir en modo escritura el fichero hosts.cfg, y creamos el servidor con su hostgroup, su nombre, su alias, y su dirección IP.</li> <li>3. Como mínimo deberemos agregar un servicio para poder visualizarlo en la interfaz web, por lo que creamos el servicio de PING.</li> <li>4. Una vez guardada toda la información, reiniciamos el servicio de Nagios para recompilar el código (service nagios reload/restart).</li> </ol>
<b>Criterio de aceptación</b>	Se muestra el nuevo servidor monitorizado dentro de la pantalla principal de "Hostgroup", con un chequeo de PING para hacerlo visible en la interfaz web de Nagios.

Tabla 88.- PV-03: Agregar un servidor a monitorizar

<b>Identificador: PV-04</b>	
<b>Título</b>	Agregar un servicio a monitorizar
<b>Descripción</b>	Se comprueba que Nagios muestra el nuevo servicio a monitorizar en un servidor.
<b>Requisitos relacionados</b>	RF-03, RF-04, RF-09, RF-15, RNF-01, RNF-02, RNF-03, RNF-04, RNF-05, RNF-06
<b>Procedimiento</b>	<ol style="list-style-type: none"> <li>1. Acceso al servidor de Nagios mediante el puerto 22 del protocolo SSH, para acceder a los ficheros de configuración en /usr/local/nagios/etc.</li> <li>2. Abrir en modo escritura el fichero services.cfg, y creamos el servicio con su servicegroup, su nombre, su nombre del servicio, su chequeo, su grupo de contactos y su tipo de monitorización.</li> <li>3. Una vez guardada toda la información, reiniciamos el servicio de Nagios para recompilar el código (service nagios reload/restart).</li> </ol>
<b>Criterio de aceptación</b>	Se muestra el nuevo servicio monitorizado dentro de la pantalla principal de "Hostgroup" dentro del servidor o en la pantalla de "Servicegroup" de forma asociada al servidor donde lo hemos definido.

Tabla 89.- PV-04: Agregar un servicio a monitorizar

<b>Identificador: PV-05</b>	
<b>Título</b>	Opción de configurar alertas
<b>Descripción</b>	Se comprueba que en Nagios se pueden activar y desactivar alertas para servidores y servicios.
<b>Requisitos relacionados</b>	RF-03, RF-04, RF-16, RF-17, RF-18, RNF-01, RNF-08
<b>Procedimiento</b>	<ol style="list-style-type: none"> <li>1. Acceso al servidor de Nagios mediante el puerto 22 del protocolo SSH, para acceder a los ficheros de configuración en /usr/local/nagios/etc.</li> <li>2. Dentro del fichero services.cfg se definirán los tipos de alertas y el envío al grupo de contactos seleccionado.</li> <li>3. Una vez guardada toda la información, reiniciamos el servicio de Nagios para recompilar el código (service nagios reload/restart).</li> </ol>
<b>Criterio de aceptación</b>	Se muestra una alerta nueva, dentro de un servidor o servicio ya existente en Nagios

Tabla 90.- PV-05: Opción de configurar alertas

<b>Identificador: PV-06</b>	
<b>Título</b>	Opción de configurar notificaciones
<b>Descripción</b>	Se comprueba que en Nagios se pueden activar y desactivar notificaciones para servidores o servicios monitorizados.
<b>Requisitos relacionados</b>	RF-03, RF-19, RF-20, RNF-01
<b>Procedimiento</b>	<ol style="list-style-type: none"> <li>1. Acceso al servidor de Nagios mediante la interfaz gráfica, para poder seleccionar los servidores donde queramos detener la notificación de alertas.</li> <li>2. Se busca el servidor concreto, y si se quiere quitar la notificación a:             <ol style="list-style-type: none"> <li>a. Un servidor, pulsamos en el nombre del "host", y en la parte derecha de la pantalla aparece un amplio menú de opciones, donde se selecciona en "Disable notifications for this host".</li> <li>b. Un servicio, pulsamos en el nombre del "service" de dicho "host", y en el menú que aparecerá en la parte derecha, se selecciona "Disable notifications for this service".</li> <li>c. Todos los servicios, pulsamos en el nombre del "host" y en el menú de la parte derecha, se selecciona "Disable notifications for all services on this host".</li> </ol> </li> <li>3. Una vez seleccionada alguna de las tres posibles opciones, aparece una ventana con un botón informando de la acción que se va a realizar, se debe pulsar en "Done" y ya aparecerá en Nagios con un símbolo que indica el estado de Notificación deshabilitada.</li> <li>4. Para volver a habilitar las notificaciones en un servidor, servicio o servicios de un servidor, habrá que realizar el mismo paso, y en lugar de usar opciones de "Disable", será la misma opción pero con el texto "Enable".</li> </ol>
<b>Criterio de aceptación</b>	Se muestra un servidor, un servicio o todos los servicios de un servidor con las notificaciones deshabilitadas.

Tabla 91.- PV-06: Opción de configurar notificaciones

Identificador: PV-07	
<b>Título</b>	Opción de configurar monitorización
<b>Descripción</b>	Se comprueba que en Nagios se pueden activar y desactivar períodos de monitorización para servidores y servicios.
<b>Requisitos relacionados</b>	RF-03, RF-06, RF-10, RF-11, RF-12, RF-13, RNF-03
<b>Procedimiento</b>	<ol style="list-style-type: none"> <li>1. Acceso al servidor de Nagios mediante el puerto 22 del protocolo SSH, para acceder a los ficheros de configuración en /usr/local/nagios/etc.</li> <li>2. Abrir en modo escritura al fichero hosts.cfg o services.cfg, dependiendo el cambio de configuración que se desee realizar.</li> <li>3. Una vez guardada toda la información, reiniciamos el servicio de Nagios para recompilar el código (service nagios reload/restart).</li> </ol>
<b>Criterio de aceptación</b>	Se muestra una configuración de monitorización a un servidor, servicio o conjunto de servicio.

Tabla 92.- PV-07: Opción de configurar monitorización

Identificador: PV-08	
<b>Título</b>	Correcto funcionamiento Nagios
<b>Descripción</b>	Se comprueba que en Nagios se mantiene estable y funciona correctamente, teniendo en cuenta todos los parámetros de configuración descritos anteriormente.
<b>Requisitos relacionados</b>	RF-01, RF-03, RF-04, RNF-01, RNF-02, RNF-03, RNF-07
<b>Procedimiento</b>	<ol style="list-style-type: none"> <li>1. Acceso al servidor de Nagios mediante el puerto 22 del protocolo SSH, para acceder a los ficheros de configuración en /usr/local/nagios/etc.</li> <li>2. Evaluar el correcto estado y configuración de los fichero contenidos en /usr/local/nagios/etc, principalmente.</li> <li>3. Siempre que se realice una modificación, se realizará un backup de los ficheros contenidos en /usr/local/nagios/, para evitar una posible pérdida de datos en el caso en el que el servidor de Nagios tenga problemas o deje de funcionar.</li> </ol>
<b>Criterio de aceptación</b>	Siempre se mantendrá el servidor de monitorización de Nagios contralado y con la información almacenada en backup, con toda la información relativa a la configuración, a los informes, las vistas, etc., para que en el caso de pérdida parcial o total del funcionamiento del servidor, se pueda restaurar una copia de todos los ficheros, de forma rápida y sin pérdida total del servicio.

Tabla 93.- PV-08: Correcto funcionamiento de Nagios

## 4.2. Pruebas de rendimiento

En este punto, se han realizado una serie de pruebas de rendimiento, comparando datos obtenidos de servidores y servicios, antes y después de ser monitorizados, para poder determinar el impacto que tiene la ejecución del agente de Nagios en ellos.

A su vez, ha sido valorado el impacto que tiene en los técnicos, la notificación de alertas por correo electrónico, mejorando en un porcentaje muy alto la detección de problemas y anticipándose a los futuros incidentes dentro de una organización.

En la siguiente imagen (véase Ilustración 36), se muestran 34 chequeos realizados en varios servidores, donde se observan todos en estado OK, ya que están dentro de los valores establecidos por parámetros en los chequeos para cada servidor.

The screenshot shows the Nagios web interface. At the top, there are three summary boxes: 'Current Network Status' (Last Updated: Tue Apr 2 11:45:20 CEST 2013), 'Host Status Totals' (Up: 6, Down: 0, Unreachable: 0, Pending: 0), and 'Service Status Totals' (Ok: 34, Warning: 0, Unknown: 0, Critical: 0, Pending: 0). Below these is a table titled 'Service Status Details For All Hosts' with a 'Limit Results' dropdown set to 100. The table has the following columns: Host, Service, Status, Last Check, Duration, Attempt, and Status Information. The table lists 34 services, all with a status of 'OK'. For example, the 'backup' service is a PING check that is OK, with a last check time of 02-04-2013 10:56:29 and a duration of 0d 0h 48m 51s. Other services include cpu, drive\_c, memuse, nsclient, nsclient version, pagefile use, uptime, users, wmi, coredown, coreup, corevirtual, and jason.

Host	Service	Status	Last Check	Duration	Attempt	Status Information	
backup	PING	OK	02-04-2013 10:56:29	0d 0h 48m 51s	1/4	PING OK - Packet loss = 0%, RTA = 1.43 ms	
	cpu	OK	02-04-2013 10:56:30	0d 2h 50m 43s	1/3	CPU Load 18% (5 min average)	
	drive_c	OK	02-04-2013 10:56:30	0d 16h 30m 1s	1/3	c - total: 15.83 Gb - used: 7.18 Gb (46%) - free 8.45 Gb (54%)	
	drive_d	OK	02-04-2013 10:56:30	0d 16h 30m 1s	1/3	d - total: 18.28 Gb - used: 9.38 Gb (51%) - free 8.90 Gb (49%)	
	memuse	OK	02-04-2013 10:56:30	0d 16h 30m 1s	1/3	Memory usage: total:3949.03 Mb - used: 762.66 Mb (19%) - free: 3186.38 Mb (81%)	
	nrpe	OK	02-04-2013 10:56:30	0d 2h 50m 43s	1/3	TCP OK - 0.013 second response time on port 5686	
	nsclient	OK	02-04-2013 10:56:29	0d 2h 50m 43s	1/3	TCP OK - 0.009 second response time on port 12489	
	nsclient version	OK	02-04-2013 10:56:30	0d 2h 50m 43s	1/3	NSClient++ 0.3.0.1 RC 2007-11-28	
	pagefile use	OK	02-04-2013 10:56:30	0d 0h 48m 50s	1/3	10	
	uptime	OK	02-04-2013 10:56:30	0d 16h 30m 1s	1/3	System Uptime - 10 day(s) 20 hour(s) 10 minute(s)	
	users	OK	02-04-2013 10:56:30	0d 0h 48m 50s	1/3	2	
	wmi	OK	02-04-2013 10:56:30	0d 0h 48m 50s	1/3	WMI/MGMT: Started	
	coredown	PING	OK	02-04-2013 10:56:29	0d 0h 48m 51s	1/4	PING OK - Packet loss = 0%, RTA = 1.60 ms
	coreup	PING	OK	02-04-2013 10:56:30	0d 0h 48m 50s	1/4	PING OK - Packet loss = 0%, RTA = 4.77 ms
	corevirtual	PING	OK	02-04-2013 10:56:30	0d 0h 48m 50s	1/4	PING OK - Packet loss = 0%, RTA = 3.61 ms
	jason	PING	OK	02-04-2013 10:56:30	0d 0h 48m 50s	1/4	PING OK - Packet loss = 0%, RTA = 2.31 ms
	cpu	OK	02-04-2013 10:56:30	0d 2h 50m 43s	1/3	CPU Load 0% (5 min average)	
	drive_c	OK	02-04-2013 10:56:30	0d 2h 50m 43s	1/3	c - total: 39.90 Gb - used: 15.92 Gb (40%) - free 23.98 Gb (60%)	
	memuse	OK	02-04-2013 10:56:29	0d 2h 50m 43s	1/3	Memory usage: total:8189.37 Mb - used: 975.69 Mb (12%) - free: 7213.68 Mb (88%)	
	nrpe	OK	02-04-2013 10:56:30	0d 2h 50m 43s	1/3	TCP OK - 0.013 second response time on port 5686	
	nsclient	OK	02-04-2013 10:56:30	0d 2h 50m 43s	1/3	TCP OK - 0.006 second response time on port 12489	
	nsclient version	OK	02-04-2013 10:56:30	0d 2h 50m 43s	1/3	NSClient++ 0.4.1.90 2013-02-04	
	pagefile use	OK	02-04-2013 10:56:30	0d 2h 50m 43s	1/3	0	
	uptime	OK	02-04-2013 10:56:30	0d 2h 50m 43s	1/3	System Uptime - 53 day(s) 21 hour(s) 28 minute(s)	

Ilustración 36.- Ejemplo de chequeos de Nagios en la web

La aplicación está programada para que avise de los problemas detectados por correo electrónico, proporcionando toda la información que se visualiza en Nagios, y dando a conocer de un vistazo, todos los datos de la generación de dicha alerta.

Si el chequeo de monitorización “CPU” del servidor “backup” estuviese en estado CRITICAL (color rojo), los técnicos que estén puestos como contactos en la notificación de alertas, recibirían el siguiente correo electrónico (véase Tabla 94).

-----Mensaje original-----  
De: nagios [mailto:nagios@nagios]  
Enviado el: xxx  
Para: Alertas Nagios  
Asunto: \*\* CRITICAL en backup/CPU \*\*

\*\*\*\*\* Nagios \*\*\*\*\*

Tipo de Notificacion: PROBLEM

Servicio: CPU

**Host: localhost**

Alias: Servidor local Nagios

IP: 192.168.1.5

Estado: CRITICAL

Fecha: xxx

-----  
**CPU CRITICAL, 0% idle**  
-----

Contacto: xxx

Tabla 94.- Ejemplo de alerta con Nagios por nivel de CPU alto en localhost

A su vez, Nagios también proporcionará, en este caso, una vista del consumo de la CPU (véase Ilustración 37), que nos informará de los posibles problemas que tenga el servidor, ya que podría estar mal dimensionado y tener asignados poco recursos, y de ahí el número elevado de alertas.

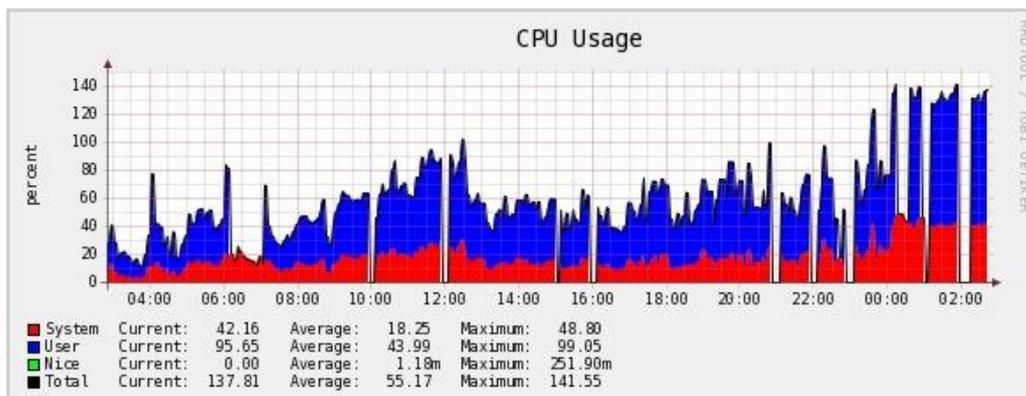


Ilustración 37.- Ejemplo de gráfica de CPU en servidor

Al igual que se pueden visualizar las gráficas individualmente por chequeos, se dispone de una vista amplia, que los técnicos usan en muchas ocasiones, dando información completa de la disponibilidad de todos los servicios que ofrece un servidor (véase Ilustración 38).

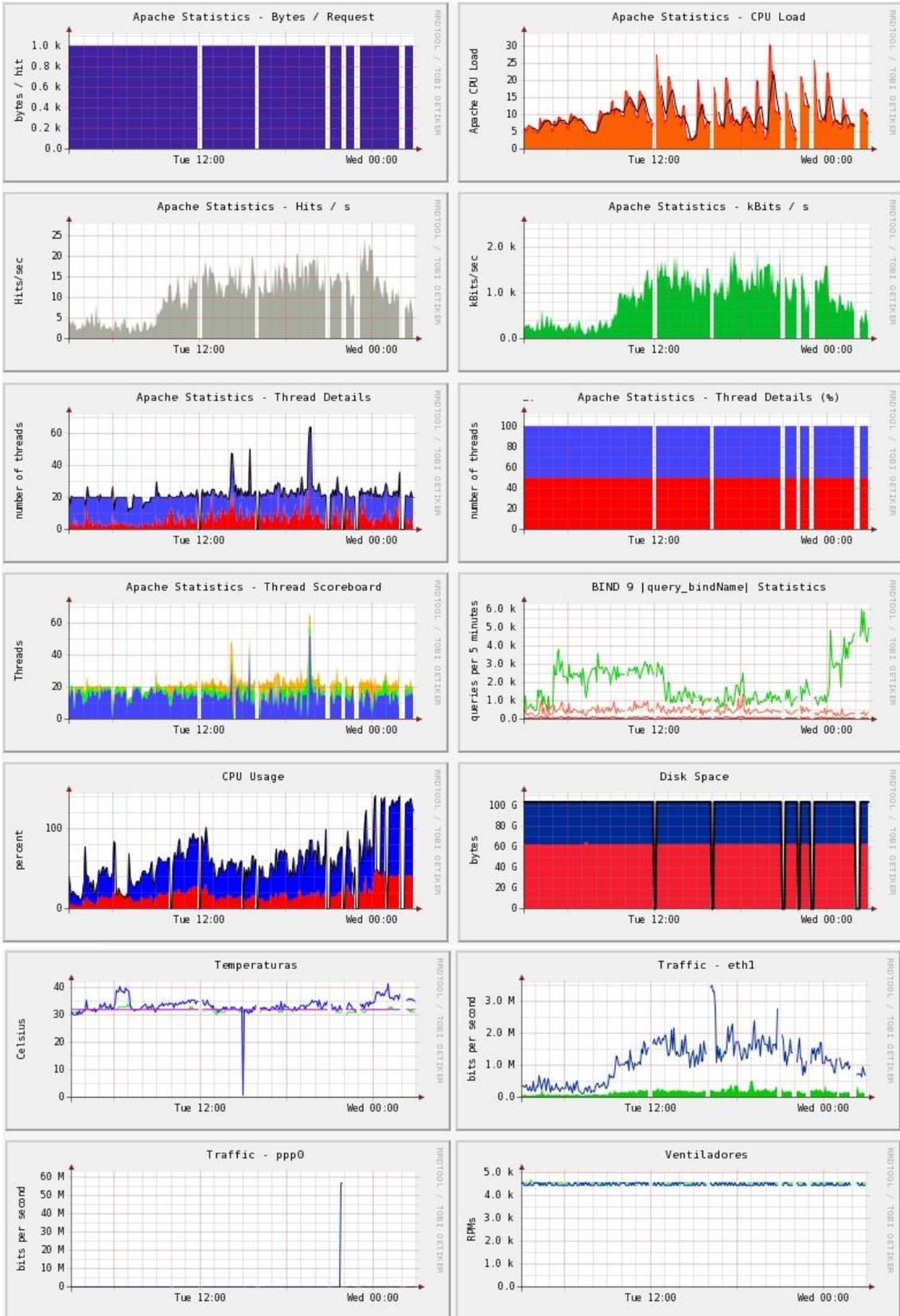


Ilustración 38.- Ejemplo de gráficas de servicios monitorizados en host

Para finalizar este subcapítulo de rendimiento, se ha realizado una breve encuesta de satisfacción a tres técnicos y a dos responsables de sistemas, para poder evaluarla fácilmente, y saber si Nagios les proporciona usabilidad, disponibilidad y el rendimiento que necesitan.

El objetivo de la presente encuesta es recibir una primera impresión general de las personas con las que trabajo y usan dicha herramienta, con una serie de preguntas, que van a determinar una visión de la funcionalidad y uso que les proporciona a los técnicos y responsables.

### Técnico 1:

Nº	Preguntas	SI	NO
1	¿Conocía Nagios antes de ser implantado en su organización?		X
2	¿Cree que Nagios mejora la disponibilidad de los sistemas en su organización?	X	
3	¿Cree que Nagios mejora su rendimiento personal?	X	
4	¿Cree que Nagios dispone de los chequeos necesarios para su uso?	X	
5	¿Cree que Nagios facilita el trabajo diario?	X	
6	¿Cree que podría administrar todos sus sistemas sin Nagios?		X
7	¿Cree que Nagios informa correctamente de las alertas?	X	
8	¿Cree que Nagios es un sistema proactivo?	X	
9	¿Cree que Nagios puede mostrarle los informes que necesite?	X	
10	¿Cree que Nagios necesita de un Administrador del software para su alto rendimiento? Detalle a continuación: <i>Sí, es necesario para ampliar el número de chequeos y personalizarlo.</i>	X	

Tabla 95.- Encuesta de satisfacción a Técnico 1 sobre Nagios

### Técnico: 2

Nº	Preguntas	SI	NO
1	¿Conocía Nagios antes de ser implantado en su organización?	X	
2	¿Cree que Nagios mejora la disponibilidad de los sistemas en su organización?	X	
3	¿Cree que Nagios mejora su rendimiento personal?	X	
4	¿Cree que Nagios dispone de los chequeos necesarios para su uso?	X	
5	¿Cree que Nagios facilita el trabajo diario?	X	
6	¿Cree que podría administrar todos sus sistemas sin Nagios?		X
7	¿Cree que Nagios informa correctamente de las alertas?	X	
8	¿Cree que Nagios es un sistema proactivo?	X	
9	¿Cree que Nagios puede mostrarle los informes que necesite?	X	
10	¿Cree que Nagios necesita de un Administrador del software para su alto rendimiento? Detalle a continuación: <i>Para poder completar la monitorización en servidores de diferentes sistemas operativos, es necesario contar con una administrador que amplie los chequeos.</i>	X	

Tabla 96.- Encuesta de satisfacción a Técnico 2 sobre Nagios

**Técnico: 3**

Nº	Preguntas	SI	NO
1	¿Conocía Nagios antes de ser implantado en su organización?		X
2	¿Cree que Nagios mejora la disponibilidad de los sistemas en su organización?	X	
3	¿Cree que Nagios mejora su rendimiento personal?	X	
4	¿Cree que Nagios dispone de los chequeos necesarios para su uso?	X	
5	¿Cree que Nagios facilita el trabajo diario?	X	
6	¿Cree que podría administrar todos sus sistemas sin Nagios?		X
7	¿Cree que Nagios informa correctamente de las alertas?	X	
8	¿Cree que Nagios es un sistema proactivo?	X	
9	¿Cree que Nagios puede mostrarle los informes que necesite?	X	
10	¿Cree que Nagios necesita de un Administrador del software para su alto rendimiento? Detalle a continuación: <i>Es necesario, ya que sin un administrador, el número de chequeos solo serían los que aparecen por defecto y en muchos sistemas no son suficientes.</i>	X	

Tabla 97.- Encuesta de satisfacción a Técnico 3 sobre Nagios

**Responsable: 1**

Nº	Preguntas	SI	NO
1	¿Conocía Nagios antes de ser implantado en su organización?		X
2	¿Cree que Nagios mejora la disponibilidad de los sistemas en su organización?	X	
3	¿Cree que Nagios mejora su rendimiento personal?	X	
4	¿Cree que Nagios dispone de los chequeos necesarios para su uso?	X	
5	¿Cree que Nagios facilita el trabajo diario?	X	
6	¿Cree que podría administrar todos sus sistemas sin Nagios?		X
7	¿Cree que Nagios informa correctamente de las alertas?	X	
8	¿Cree que Nagios es un sistema proactivo?	X	
9	¿Cree que Nagios puede mostrarle los informes que necesite?	X	
10	¿Cree que Nagios necesita de un Administrador del software para su alto rendimiento? Detalle a continuación: <i>Sí, ya que es una herramienta que sin administración proactiva, no dispone de más información que la básica, proporcionada por medio de una instalación sencilla del sistema.</i>	X	

Tabla 98.- Encuesta de satisfacción a Responsable 1 sobre Nagios

**Responsable: 2**

Nº	Preguntas	SI	NO
1	¿Conocía Nagios antes de ser implantado en su organización?	X	
2	¿Cree que Nagios mejora la disponibilidad de los sistemas en su organización?	X	
3	¿Cree que Nagios mejora su rendimiento personal?	X	
4	¿Cree que Nagios dispone de los chequeos necesarios para su uso?	X	
5	¿Cree que Nagios facilita el trabajo diario?	X	
6	¿Cree que podría administrar todos sus sistemas sin Nagios?		X
7	¿Cree que Nagios informa correctamente de las alertas?	X	
8	¿Cree que Nagios es un sistema proactivo?	X	
9	¿Cree que Nagios puede mostrarle los informes que necesite?	X	
10	¿Cree que Nagios necesita de un Administrador del software para su alto rendimiento? Detalle a continuación: <i>Observando el alto rendimiento de mis sistemas, será necesario disponer de un administrador de Nagios para poder ampliar los chequeos necesarios para conseguir la alta disponibilidad y la proactividad.</i>	X	

Tabla 99.- Encuesta de satisfacción a Responsable 2 sobre Nagios

El resumen que nos aparece, se va a representar en una gráfica para determinar los pesos de cada una de las respuestas que nos han facilitado los técnicos y responsables que han trabajado con Nagios (véase Ilustración 39).

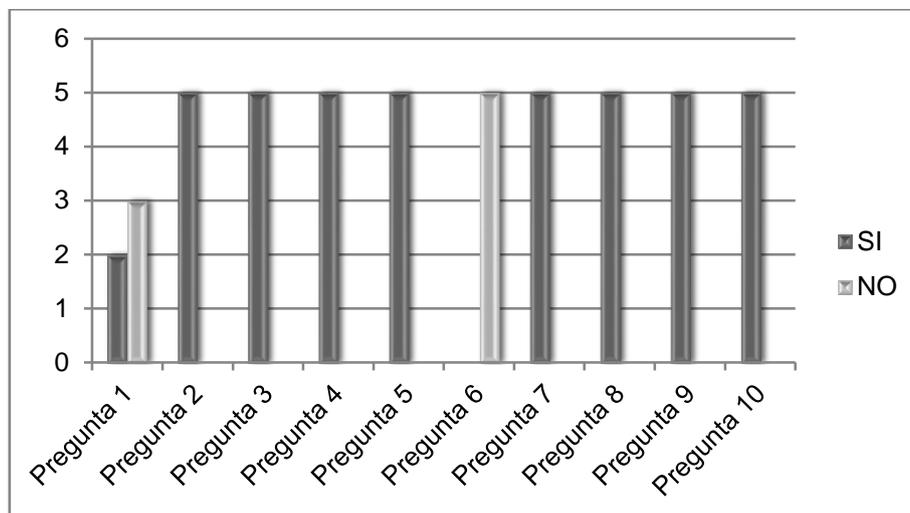


Ilustración 39.- Análisis de datos de encuestas

Observando estos datos se entiende que actualmente la gente que ha trabajado con Nagios, necesita dicha herramienta para poder realizar mejor su trabajo diario y para prever los fallos, anticipándose a errores mayores o caídas generalizadas de algún sistema informático.

# **Capítulo 5: Planificación y presupuesto**

---

## 5. Planificación y presupuesto

En este capítulo, se exponen los datos de los tiempos y presupuesto que se ha invertido en la realización del proyecto. Todas las fechas han sido establecidas para conseguir cumplir con los tiempos del proyecto que fueron marcados meses antes de su entrega y defensa.

En los siguientes apartados, se describen la planificación y el presupuesto final para el desarrollo del proyecto, teniendo en cuenta las fases de los capítulos que están incluidas en este proyecto.

### 5.1. Planificación

A continuación, se van a mostrar los tiempos de planificación establecidos para la realización del presente proyecto, con una gráfica de porcentajes y un diagrama de Gantt realizado con Microsoft Project 2013, con todas las fechas de la realización de las diferentes fases que comprenden el proyecto (véase Ilustración 41).

En la siguiente ilustración se muestra un gráfico con los porcentajes que han comprendido el desarrollo del presente proyecto, siendo las etapas de Diseño, Análisis e Implantación, las que mayor tiempo han abarcado (véase Ilustración 40).

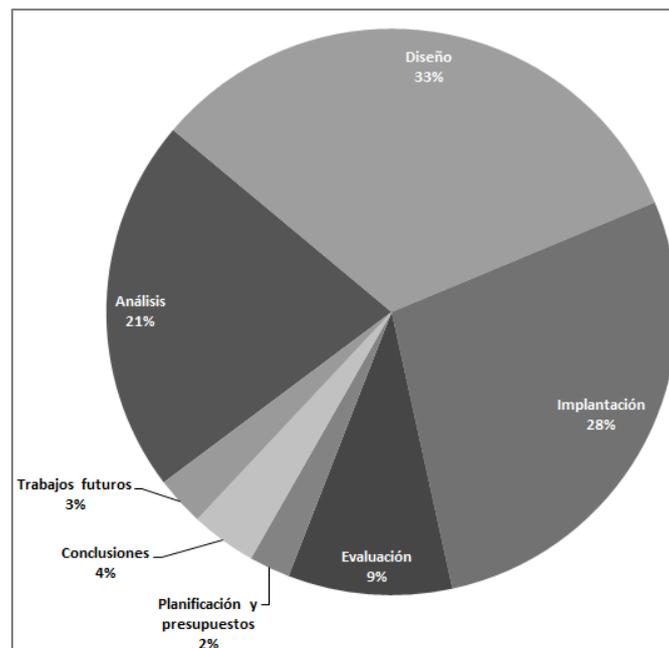


Ilustración 40.- Planificación proyecto en gráfico

Trabajo Fin de Grado  
Alexandra Moreno Calabozo

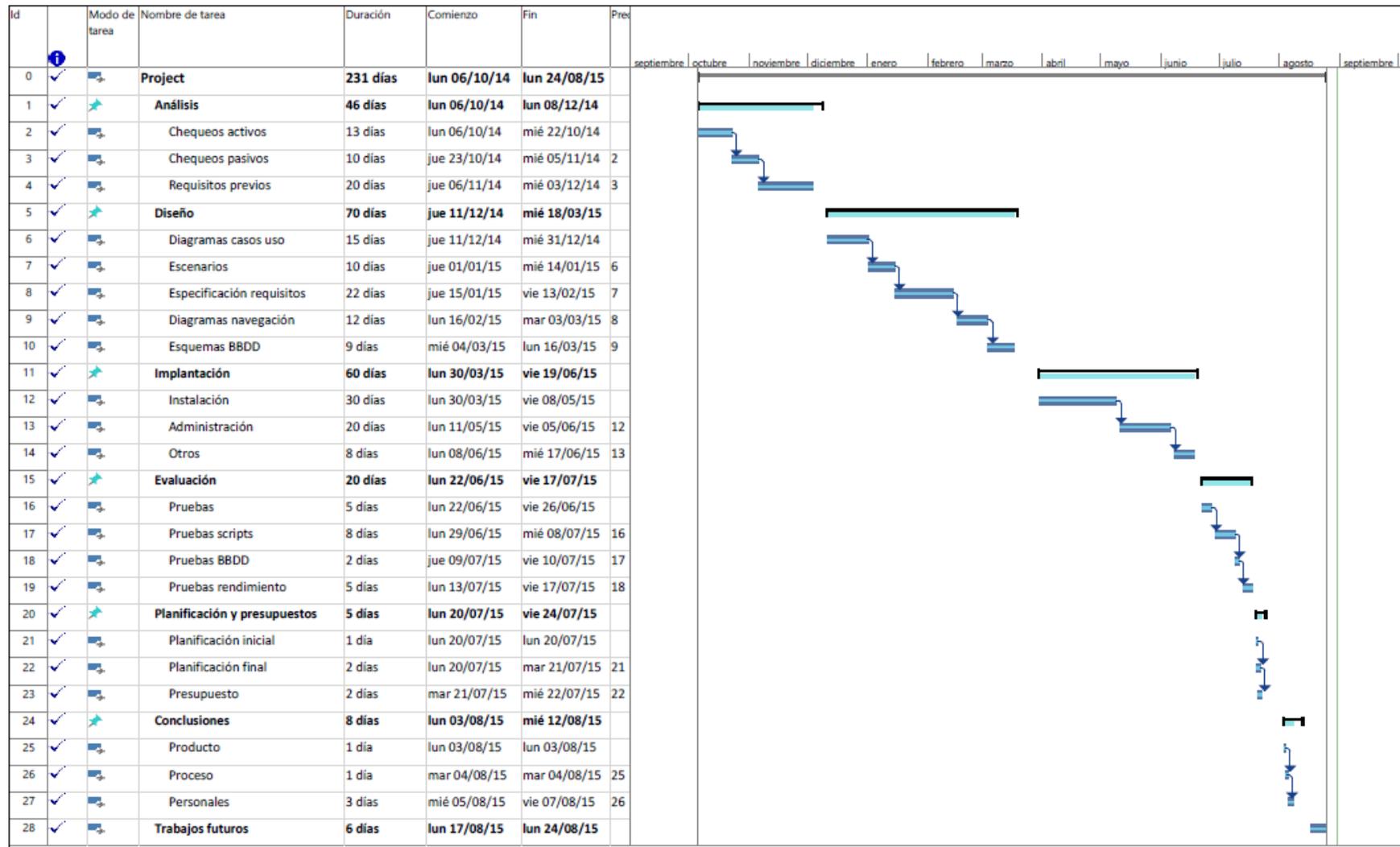


Ilustración 41.- Planificación proyecto (Diagrama de Gantt)

## 5.2. Presupuesto

A continuación, se van a exponer los costes derivados de la realización del proyecto, desglosando los cálculos de personal y material, para poder determinar un coste total durante el período de duración que ha conllevado el presente proyecto.

### 5.2.1. Gasto de personal

Para poder desarrollar este proyecto, se deben tener en cuenta varios aspectos de personal, ya que los encargados del mismo serán dos roles, representados normalmente por dos personas, que en este caso será el Tutor y el alumno, reportando el primero horas a la “Jefatura de proyecto”, y el alumno reportándolas entre los distintos roles del proyecto. Los roles que se han establecido son cuatro: Jefe de Proyecto, Diseñador, Analista, Programador y Técnico.

Estos cinco roles serán necesarios para poder diseñar e implantar Nagios, teniendo en cuenta que como mínimo deberán ser dos personas físicas. El desglose de los precios que tendrá el personal, serán los siguientes:

Costes totales de personal			
Roles	Horas	Coste por hora	Coste total
Jefe de Proyecto	100	30	3.000 €
Diseñador	80	40	3.200 €
Analista	200	30	6.000 €
Programador	600	25	15.000 €
Técnico	400	20	8.000 €
<b>TOTAL</b>			<b>36.200 €</b>

Tabla 100.- Coste total de personal del proyecto

Los coste por hora de cada uno de los roles, han sido estudiados a raíz de los datos del Convenio de Oficinas y Despachos del que se rigen casi todas las empresas del sector de las telecomunicaciones en España, para el cálculo de los sueldos.

### 5.2.2. Gasto de material

Para poder evaluar el gasto en material del proyecto, hemos establecido unos períodos de amortización y los coeficientes lineales máximos que marca la Ley española.

Para poder calcular el coste total de los 8 meses de proyecto, aproximadamente, es necesario realizar la siguiente fórmula:

$$\text{Coste total} = \text{Precio Unidad} \times \text{Coeficiente lineal} / 12 \text{ meses} \times 8 \text{ meses de proyecto}$$

Siguiendo la fórmula anterior, podremos determinar el coste del material que hemos usado en el proyecto.

Costes totales de material					
Material	Precio por unidad	Coeficiente lineal máximo	Período de amortización	Duración del proyecto	Coste total
Servidor Intel Core i7 3,1Ghz	600	20%	3 años	8 meses	80,00 €
Software Nagios	0	-	-	8 meses	0,00 €
Equipo Desarrollador	1000	20%	3 años	8 meses	133,33 €
Microsoft Windows 8.1	200	33%	3 años	8 meses	26,67 €
Microsoft Office 2010	100	33%	3 años	8 meses	13,33 €
<b>TOTAL</b>					<b>253,33 €</b>

Total 1.140,00 €

Remanente 760,00 €

Tabla 101.- Coste total de material del proyecto

En total, el gasto obtenido en el material del proyecto supone un total de 253,33€.

### 5.2.3. Resumen de costes

Para finalizar la parte de costes y presupuestos del proyecto realizado, se realiza la suma de los dos anteriores apartados, donde sumamos los costes del personal y el material.

Coste total	
Personal	36.200,00 €
Material	253,33 €
<b>TOTAL</b>	<b>36.453,33€</b>

Tabla 102.- Coste total proyecto

Dentro de los costes de cualquier proyecto, es necesaria la división entre los costes de personal y de material, siendo los primeros los que mayor número de horas y de coste necesitan (véase Ilustración 42). En el caso del coste del material se presupone la durabilidad de los elementos para poder acotar el coste de dicho material durante el período que necesitamos calcularlo.

Como en todo proyecto de implantación de un sistema, el valor casi total del proyecto, proviene del personal necesario para la realización de dicha puesta en marcha, con todos los perfiles precisos para la misma.

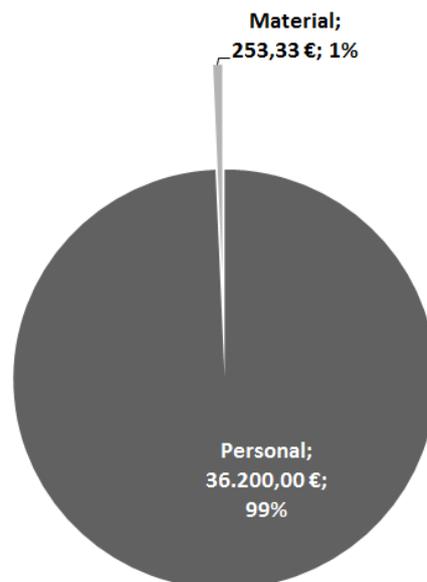


Ilustración 42.- Coste total proyecto en gráfica

## **Capítulo 6: Conclusiones y trabajos futuros**

---

## **6. Conclusiones y trabajos futuros**

Para finalizar el desarrollo del proyecto es necesario comentar el cumplimiento exitoso de todos los objetivos que se propusieron al comienzo, y detallar los conocimientos adquiridos y desarrollados durante la carrera, para llegar a diseñar e implantar este proyecto.

Además, se van a proponer futuros trabajos a raíz de todas las ideas y objetivos propuestos en este proyecto, con una amplia descripción de todas las posibilidades que presenta el haber elegido este proyecto, con un amplio abanico de posibilidades futuras.

### **6.1. Conclusiones generales**

Para desarrollar mi proyecto me he basado en la necesidad de mejora continua de los sistemas de información, y en la alta disponibilidad, operatividad, eficiencia y satisfacción de los usuarios. La gran cantidad de sistemas disponibles en la actualidad, y la dependencia de los sistemas informáticos, hace que sea necesario cumplir con unos niveles de servicio, y ésta es una de las principales razones por las que nace este proyecto.

Nagios es un software libre desarrollado por una gran empresa, con el mismo nombre, que nace como sistema de monitorización de redes en 1999. Éste es capaz de monitorizar en alto nivel de detalle servidores, dispositivos y servicios, con el objetivo de ser proactivos y poder visualizar en tiempo real todos los fallos de comunicaciones o de disponibilidad.

Proporciona una gran versatilidad para consultar cualquier dato de interés de un sistema, ya que tiene la facilidad de usar script propios o desarrollados por la comunidad de desarrolladores o por cualquier administrador de la aplicación con conocimiento en cualquier lenguaje de programación. Usa un sistema de alertas con notificaciones, para poder enviar por correo electrónico y de este modo poder escalar cualquier problema en cualquier servidor a técnicos o responsables.

Después de desarrollar el proyecto, es necesario comentar que Nagios es una herramienta completa, interesante y práctica. Ya que en el presente trabajo quería realizar un buen diseño e implantación, he tenido la suerte de poner investigar a fondo todas las posibilidades y ventajas que tiene para una pequeña, mediana o grande empresa, el uso de este sistema de monitorización, que hace que esté controlado toda la parte de negocio informático a un nivel muy alto de detalle, donde los errores que surgen son detectados desde el primer segundo de fallo, pudiendo agilizar su solución y haciendo que el trabajo del día a día sea mucho más cómodo y ameno para todos los responsables y técnicos de las aplicaciones y/o servicios de los que disponga su negocio.

### **6.1.1. Conclusiones sobre el sistema puesto en marcha**

Al principio de la presente memoria se han presentado los objetivos que se establecieron y a continuación, se van a evaluar todo ellos, para comprobar que han sido cumplidos íntegramente.

Se ha demostrado en los puntos anteriores, que uno de los objetivos principales es la detección y proactividad en la visión de alertas inesperadas, para el cambio y modificación de las características de los equipos monitorizados.

Uno de los puntos que se han recalcado en anteriores capítulos ha sido el análisis de la eficiencia por medio de los informes SLA, y con éstos la reducción de costes de los servicios usados, por la monitorización constante y activa que se realizada sobre todos los dispositivos y servidores, para la mejora de la calidad del servicio y de todos los componentes y dispositivos dados de alta en la aplicación, ya sea con uno o varios servicios monitorizados.

Por último, Nagios podrá detectar y verificar la infraestructura de los servidores en red que dispongan de comunicación con él, y de éste modo poder notificar cualquier cambio de estado de lo reflejado en los parámetros de configuración de los chequeos que hayan sido definidos para un servidor determinado.

Nagios, en su versión de software libre, dispone de un amplio número de objetivos, y se está demostrando que para poder monitorizar no hace falta pagar, ya que teniendo un buen administrador de Nagios, todo es monitorizable y accesible, para hacerlo visible y controlable en por él y todos los técnicos de soporte asociados a un servicio.

### **6.1.2. Conclusiones sobre la planificación usada**

Durante el desarrollo del presente proyecto se han encontrado algunos problemas a la hora de instalar Nagios, ya que la configuración en un entorno de pruebas, con varias máquinas dentro de una red local, no tiene el mismo nivel de pruebas que en una red amplia de administración completa.

Para probar Nagios completamente, se han definido varias máquinas virtuales en un ESX, dentro de una red local, con direccionamiento IP 192.168.1.0/8, donde se han realizado todas las pruebas.

Todos los scripts que se han incluido han sido desarrollados por mí, pero para otro proyecto externo, ya que dicho scripts no pueden ser probados en una arquitectura creada para pruebas, y con la finalidad de documentar las características de diseño e implantación de una herramienta de monitorización, tan potente como Nagios.

### **6.1.3. Conclusiones personales**

Durante el desarrollo del presente Trabajo Final de Carrera, han sido muchas las asignaturas a las que tengo que agradecer los conocimientos de los que dispongo.

A continuación se van a presentar las asignaturas que han sido necesarias para presentar dicho trabajo con un pequeño resumen y los objetivos que se han alzado gracias a las mismas.

- Estructura de computadores:
  - Resumen: en esta asignatura se presentan y describen los principales componentes de un computador y el funcionamiento básico de los mismos.
  - Objetivos: gracias a esta asignatura se ha demostrado un amplio conocimiento en la estructura, organización, funcionamiento e interconexión entre los sistemas informático, los fundamentos de programación y la resolución de problemas.
  
- Sistemas Operativos:
  - Resumen: en esta asignatura se amplían los conocimientos de sistemas operativos, con sus características, funcionalidades y estructura, enseñando técnicas de programación y métodos de gestión de procesos y recursos.
  - Objetivos: gracias a esta asignatura se ha podido ampliar el número de chequeos en Nagios, ya que para poder monitorizar ciertos procesos, debes conocer métodos y llamadas al sistema que se desconocen en un principio.
  
- Arquitectura de Computadores:
  - Resumen: en esta asignatura se enseñan modelos de programación y rendimiento de sistemas, que hacen relacionarlo completamente con mi TFG.
  - Objetivos: gracias a los conocimientos impartidos en esta asignatura es muy fácil e intuitivo comprobar el funcionamiento de una memoria o un procesador en un servidor monitorizado.
  
- Diseño de Sistemas Operativos:
  - Resumen: en esta asignatura se enseñaron conocimiento sobre el uso y programación de ordenadores, sistemas operativos, bases de datos y diferentes programas informáticos, diseñando e implementando aplicaciones basadas en sus servicios.
  - Objetivos: gracias a esta asignatura adquirí un amplio nivel de conocimiento, que me enseñaron a conocer en profundidad sistemas operativos y bases de datos, para poder manipular su contenido y administrarlo.

- Interfaces de usuario:
  - Resumen: en esta asignatura se consiguió aprender a diseñar interfaces para hacerlas más intuitivas y útiles para los usuarios por medio de diferentes estructuras y métodos de navegación.
  - Objetivos: gracias a esta asignatura se aprendió a comprender mejor una interfaz de usuario que fuera atractiva, útil y usable para los usuarios que la fueran a usar.
  
- Sistemas en tiempo real:
  - Resumen: en esta asignatura se consiguió entender el funcionamiento de un sistema en tiempo real y determinar las calidades que un servicio necesita para poder realizar arquitecturas más completas y funcionales.
  - Objetivos: gracias a esta asignatura se ha podido entender todo el nivel de calidad de un sistema en tiempo real, así como la funcionalidad de los mismos.

Para poder llevar a cabo este TFG se han necesitado también otros conocimientos anteriores y posteriores al Grado universitario.

Dos años antes de empezar la carrera, realicé un Grado Superior de Formación Profesional en Administración de Sistemas Informático, que me dio una amplia base para poder afrontar la carrera y ahora este TFG.

Cuando acabé FP II, comencé a trabajar en una empresa de informática y telecomunicaciones, dedicada al Sector Informático en la Administración Pública, y que gracias a los proyectos en los que he podido trabajar, me han dado pie a alcanzar un alto nivel de estudios y conocimientos.

Desde hace seis años soy administradora de Nagios en un organismo público perteneciente al Ministerio del Interior, y gracias a estos conocimientos, he conseguido certificarme en las dos titulaciones oficiales de Nagios. He aprendido un amplio abanico de conocimientos, ya que gracias a la aplicación de Nagios, deben de tenerse conocimientos de multitud de sistemas operativos, servicios, aplicaciones, etc., para poder administrar un software tan potente.

## **6.2. General conclusions**

To execute my project I took the need of the information systems continuous improvement requirement as well as the high availability, operability, efficiency and users satisfaction. The big amount of available devices currently, and the dependency of the informatics systems, requires to achieve a certain level of service being this one of the main reasons why this project has born.

Nagios is an open source software created by a big company with the same name, born as a network monitoring system in 1999. It is capable of monitoring with high level of detail servers, devices and services, with the goal of being proactive and allow us to have a real-time understanding about all the communications or availability failures.

This software's versatility allows it to request any interesting data from a system using its own or custom scripts developed by the open development community or any application administrator with knowledge on any programming language. It uses an alert system with events to be able to show and escalate any issue in any server to the operators or managers.

After finishing up this project, is mandatory to mention that Nagios is a really complex, interesting and useful tool. As my goal with this project was to provide a good design and implementation, I had the luck to be able to research in detail all the possibilities and advantages that this software can provide to a small, medium or big enterprise. The use of this monitoring system allows having every IT device controlled in a high detailed manner, where the errors and failures are detected in the very same second as they appear, providing a quick response and speeding its solution making the daily work more easy and comfortable for all the managers and application and/or services operators in the company.

### **6.2.1. Conclusions about the live system**

At the beginning of this current memory it has been presented the established objectives and now, all of them are going to be evaluated, to confirm that they have been properly achieved.

It has been demonstrated within the previous sections, that one of the main goals is the detection and proactivity in the reporting of unexpected alerts as well as for the change and modification of the monitored system's characteristics.

One of the topics that has been stressed in the previous sections has been the efficiency analysis by using SLA reports, and with them the cost decrease of the used services, by the continuous and active monitoring performed over every devices and server, to improve the service quality and all the components and

devices signed up within the application, being with just one service or multiple services monitored.

Finally, Nagios will be able to detect and verify the reachable networking server's infrastructure, and with this method being able to notify any status change over the previously configured parameters in the defined checks for one specific server.

Nagios, on its open source version, provides a wide number of solutions demonstrating that to be able to monitor systems it is not mandatory to expend money for the software, as having a proper Nagios administrator, everything is able to be monitored and accessible to make it visible and manageable for him.

### **6.2.2. Conclusions about the used planning**

Throughout the development of this project it has been found some issues while installing Nagios, as configuring it in a lab environment, with multiple machines within the same local network, it doesn't have the same level of tests as it would have in a wide network fully managed.

To be able to completely tests Nagios, it has been defined some virtual machines in an ESX, all of them inside the same local network, with a network IP address of 192.168.1.0/8 where all the tests have been performed.

Every script included have been developed by myself for another external project, as those scripts can't be tested in a testing infrastructure, with the final goal of document the implementation and design characteristics of a monitoring tool as powerful as Nagios.

### **6.2.3. Personal conclusions**

During the current's Final Degree Project (TFG) development, there are multiple subjects I have to thank for the knowledge that I acquired by studying them.

I will present some of the subjects that have been required to be able to present this TFG with a small summary as well as its targets raised thanks to them.

- Computer structure:
  - Summary: Within this subject, it has been presented and described the main components of a computer as well as its basic operation
  - Objective: Thanks to this subject it has been proved a wide knowledge about the structure, organization, fundamentals or programming and troubleshooting.

- Operative Systems:
  - Summary: In this subject the skills about operative systems has been increased, with their characteristics, functionalities and structure, showing programming technics and process and resources management methods.
  - Objective: Thanks to this subject it was possible to increase the number of checks within Nagios, as to be able to monitor certain processes, you must know different methods and system calls that were unknown at the start of the project.
  
- Computer architecture:
  - Summary: In this subject it is taught different programming models and system performance which completely matches with this TFG.
  - Objective: Thanks to the skills taught in this course, it has been very easy and intuitive test the operation of a memory or a processor in a monitored server.
  
- Operative System Design:
  - Summary: In this subject it is taught skills about the use and programming of computers, operative systems, databases and multiple different software, designing and implementing applications based on their services.
  - Objective: Thanks to this subject I acquired great skills, which allowed me to know in deep different operative systems and databases, so I could manipulate its content and have a good level to be able to administer both.
  
- User interfaces:
  - Summary: In this subject we learnt to design interfaces to be more intuitive and useful for the users through different structures and browsing methods.
  - Objectives: Thanks to this subject I learnt to better understand a user interface that was attractive, useful and helpful for the users who were using it.
  
- Real-time systems:
  - Summary: In this subject it was achieved to understand how a real-time system works and determine the qualities that a service needs to be able to create the most functional and complex architectures.
  - Objectives: Thanks to this subject I could understand the quality level of a real-time system as well as its functionality.

To be able to perform this TFG it was required some other skills acquired prior and post this university degree.

Two years before this degree, I coursed the “Grado Superior de Formación Profesional de Administracion de Sistemas Informaticos” which gave me a good baseline to meet the degree and now this TFG.

Once I finished FP2, I began working in an IT Company, focused on the IT Sector within the Public Administration, and thanks to the projects I participated, it allowed me to achieve a high level of studies and skills.

I have been a Nagios administrator for six years in a public organism belonging to the Ministry of Interior, and thanks to these skills, I was able to get the two Nagios official certifications, and learn some skills as thanks to the Nagios application, you must have skills about multiple operative systems, services, applications, etc. to be able to administer such a powerful software.

### 6.3. Trabajos futuros

A continuación, se van a describir en el mayor grado de detalle, varias aplicaciones que se pueden integrar con Nagios, y hace que ésta sea mucho más atractiva y práctica, ofreciéndonos un amplio abanico de opciones y de vistas, para los distintos perfiles dentro de una organización. Estas dos aplicaciones que se van a detallar son NagVis y NagiosQL, que harán más fácil el trabajo diario con Nagios.

#### 6.3.1. NagVis

NagVis es uno de los complementos estrella de Nagios [9]. NagVis es una aplicación que permite crear de forma sencilla e intuitiva páginas webs personalizadas en Nagios, teniendo la opción de agregar imágenes, iconos y símbolos, para representar servidores y servicios [28].



Ilustración 43.- Logo NagVis

Como siempre es necesario disponer de unos requisitos mínimos para poder instalarlo y desplegarlo, y que en este caso nos valdrán todos los ya instalados para Nagios. Para poder descargar NagVis por medio del comando “yum” será necesario tener el repositorio “EPEL” definido entre la lista de descarga de aplicaciones, para así poder descargar la última versión del mismo.

Para poder descargarlo y desplegarlo, será necesaria la ejecución de los siguientes comandos:

```
[root@nagios ~]# yum install php-mbstring php-pdo graphviz rsync -y
[root@nagios ~]# cd /tmp/
[root@nagios ~]# wget
http://downloads.sourceforge.net/project/nagvis/NagVis%201.7/nagvis-
1.7.9.tar.gz
[root@nagios ~]# tar zxfv nagvis-1.7.9.tar.gz
[root@nagios ~]# cd nagvis-1.7.9
[root@nagios ~]# ./install.sh
```

Durante la instalación, se deberán tener en cuenta un detalle importante cuando nos solicite confirmación de la instalación. Se deberá prestar atención a la ruta de instalación, ya que por defecto selecciona “/usr/lib/nagios”, y si el sistema instalado es de 64bits, la ruta será “/usr/lib64/nagios”.

Una vez instalado y recibida la confirmación de que NagVis ha sido instalado correctamente, procedemos a reiniciar el servicio de “Apache” y el de Nagios.

```
[root@nagios ~]# service httpd restart
[root@nagios ~]# service nagios restart
```



Al igual que pueden ser desarrollados pequeños documentos Visio con los servidores y servicios, se pueden usar imágenes ya existentes para situar elementos monitorizados, como por ejemplo un mapa de Europa, y situar los servidores de las sedes de una empresa.

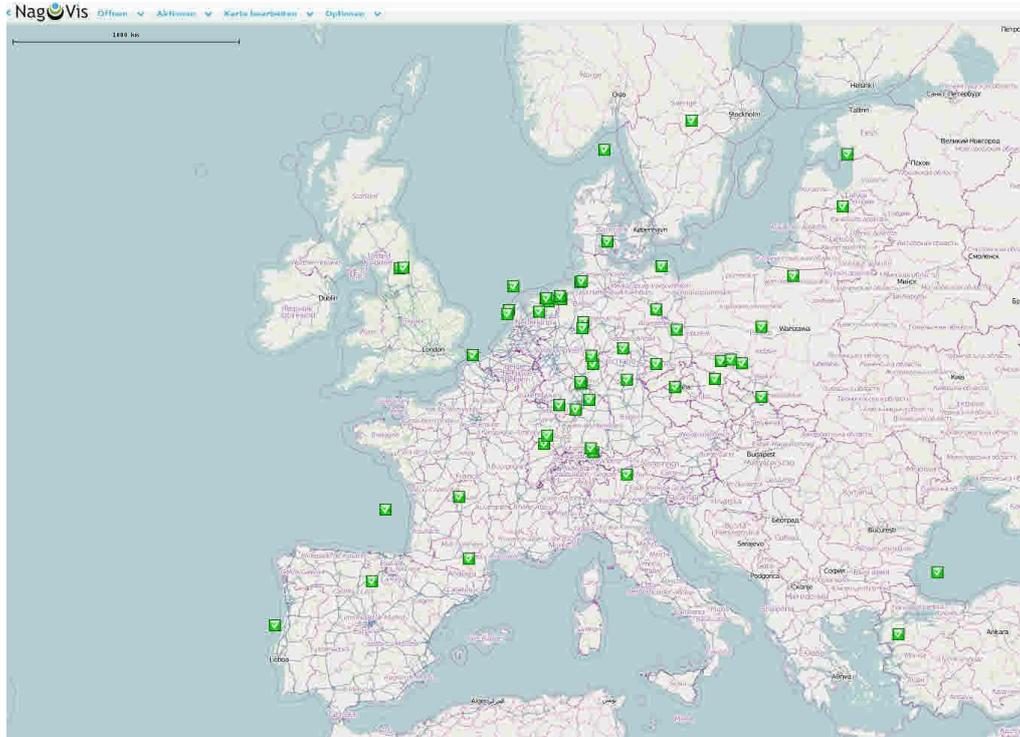


Ilustración 46.- Ejemplo NagVis integrado con Nagios en imagen

En el caso del mapa anterior (véase Ilustración 46), se pueden visualizar todos los puntos, que podrían representar sedes, servidores o servicios en concreto.



Ilustración 47.- Ejemplo NagVis integrado (imagen detallado)

En el anterior mapa (véase Ilustración 47) se muestran los datos detallados de un cierto icono del mapa, donde se puede visualizar la alerta por la que dicho icono está en color amarillo (WARNING).

También es posible incluir diseños con un nivel inferior de “originalidad”, pero con el diseño claro y conciso de cómo se organizan los servidores, por ejemplo en un CPD.



Ilustración 48.- Fotografía integrada en NagVis con chequeos de Nagios

En la Universidad, NagVis sería una herramienta muy práctica para poder tener documentados todos los equipos y servidores, y sobre todo, ubicaciones de los mismos, ya que los equipos de las aulas de informática, muchas veces se necesita el mapa de donde se encuentran y en que aula.

### 6.3.2. NagiosQL

NagiosQL es una herramienta de interfaz web que permite interactuar con la configuración establecida en los ficheros de configuración de Nagios para poder administrar éste último desde la interfaz [30], de forma muy práctica e intuitiva.



Ilustración 49.- Logo de NagiosQL

Para poder instalarla es necesario disponer de un servidor apache, MySQL y Nagios Core instalado, además de las dependencias que ya se instalan al inicio de la instalación de éste último, como son PHP y PECL [29].

Una vez descargado el paquete de NagiosQL, simplemente se deberán ejecutar los siguientes comandos vía consola.

```
[root@nagios ~]# wget
http://sourceforge.net/projects/nagiosql/files/nagiosql/NagiosQL%203.2.0/nagios
sql_320.tar.gz/download
[root@nagios ~]# mv download nagiosql_320.tar.gz
[root@nagios ~]# tar zxvf nagiosql_320.tar.gz
[root@nagios ~]# cd nagiosql32
[root@nagios ~]# ./install.sh
[root@nagios ~]# chown nagios.apache /usr/local/nagios/var/rw/nagios.cmd
```

NOTA: Se recomienda la lectura de la bibliografía 29 y 30, para la completa instalación de complementos de NagiosQL para que no falle ninguna ejecución.

La instalación se realiza por consola en el servidor, pero la configuración se realiza por medio de la interfaz gráfica desde un navegador (véase Ilustración 50) por la url:

<http://nagios/nagiosql/install/index.html>



Ilustración 50.- Instalación NagiosQL

Después de seguir la configuración de NagiosQL, nos aparece una ventana de login, para poder acceder a configurar los ficheros de enlace con Nagios Core.



Ilustración 51.- Acceso a NagiosQL

Una vez que hayamos accedido, disponemos de un menú en la interfaz, en la zona izquierda, donde nos proporcionará todas las opciones para su configuración.



Ilustración 52.- Configuración principal NagiosQL

Primero, deberemos acceder a Tools, y clicar en Data Import, para que NagiosQL pueda reconocer los ficheros de Nagios Core.

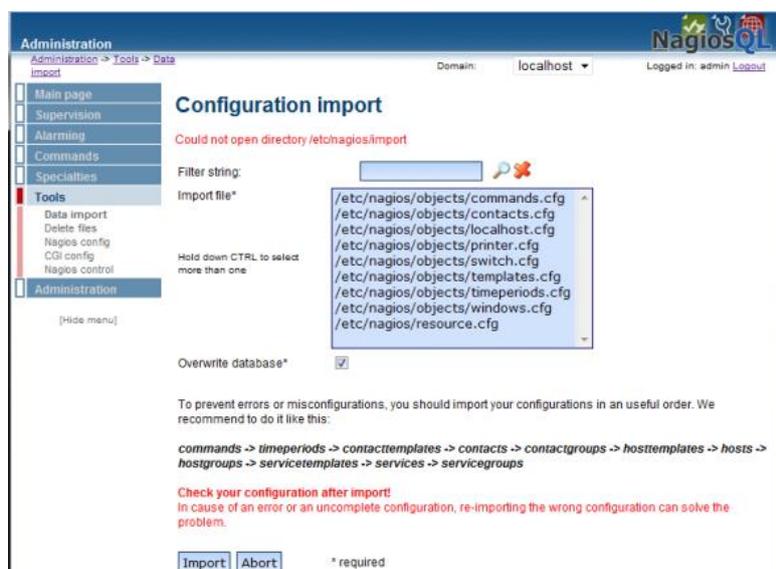


Ilustración 53.- Importación de ficheros a NagiosQL

Una vez importado, podremos acceder a toda la documentación existente en Nagios, para poder modificar o mejorar la configuración del mismo en una interfaz gráfica muy intuitiva y complementa personalizable.

Grupo	Activo	Inactivo
Hosts	4	0
Servicios	21	0
Grupo de Host	4	0
Servicios de Grupos	0	0
Plantillas de Host	5	0
Plantillas de servicios	2	0

Ilustración 54.- Consultas de monitorización en NagiosQL

Será posible la definición de hosts, services, hostgroups, servicegroups, contact, y personalización de alarmas y plantillas de envío de alertas.

00111100

Configuración común | **Comprobar Opciones** | Opciones Alarmas | Ajustes adicionales

**Comprobar Opciones**

Estado Inicial:  o  d  u ?

Máximos intentos de comprobación\*:  ?

Comprobación activa activada:  on  off  saltar  null ?

Comprobación pasiva activada:  on  off  saltar  null ?

Comprobar Periodo\*:  ?

Refresco de comprobación:  on  off  saltar  null ?

Gestor de eventos:  ?

Umbral bajo de oscilación:  % ?

Detección de oscilación activada:  on  off  saltar  null ?

Mantener información de estado:  on  off  saltar  null ?

Datos sobre el rendimiento de los procesos:  on  off  saltar  null ?

Intervalo de Reintento:  min ?

Intervalo de Comprobación:  min ?

Umbral de refresco:  sec ?

Concentración en el equipo:  on  off  saltar  null ?

Gestor de eventos activado:  on  off  saltar  null ?

Umbral alto de oscilación:  % ?

Opciones de la detección de estabilidad:  o  d  u ?

Mantiene el resto de la información:  on  off  saltar  null ?

\* Requerido

Ilustración 55.- Definición hosts en NagiosQL

Todas las modificaciones que se realicen en la interfaz web de NagiosQL se realizarán en los ficheros de configuración de Nagios en las rutas de /usr/local/nagios.

En el caso para el que estamos exponiendo el proyecto en el departamento de Informática de la Universidad, sería muy práctica la instalación de dicha herramienta para que todo el personal que trabaja en él, pudiese dar de alta elementos, sin tener que modificar en modo consola los mismos, ya que para ello solo debería existir un administrador de monitorización, para que Nagios siempre mantuviese la consistencia y estuviese organizado y correctamente administrado.

Para finalizar, comentar que esta herramienta complementaria a Nagios, es una de las que más rápidamente se configura, debido a su pantalla de configuración, ya que con solo registrar la dirección IP del servidor de Nagios, podrá reconocer todos los ficheros de configuración, y si son por defecto, solo habrá que seguir el asistente para finalizar su instalación y configuración.

### **6.3.3. Otros ámbitos**

Nagios puede ser usado en multitud de empresas y/u organismos de distintos sectores. Un caso muy práctico podría ser el uso del mismo en una fábrica, donde se disponga de maquinaria que esté conectada a una red. Podrá ser monitorizada la disponibilidad de una máquina y de todos los procesos que puedan ser “monitorizables”, ya sean servicios o puertos que deban estar disponibles.

En este trabajo nos hemos centrado en la monitorización de servicios, dispositivos y servicios de una infraestructura informática con necesidades de prestar una serie de servicios con unos tiempos de respuesta y un alto grado de disponibilidad del Departamento de Informática de la Universidad Carlos III de Madrid, pero este tipo de características también se pueden prestar en empresas, como se ha mencionado en el párrafo anterior, que disponga de herramientas con una dirección IP, para poder ser controlado su funcionamiento.

Hoy en día es muy habitual disponer de sistemas informáticos en la mayoría de las empresas, siendo necesario disponer de sistemas de control y monitorización, que nos estén informando en tiempo real de los estados de nuestros servidores, y tenerlos controlados para evitar posibles errores, y ocasionando gran pérdida de tiempo, dinero y posiblemente información.

Por el conocimiento del que dispongo y de la experiencia que he ido adquiriendo, hay multitud de dispositivos monitorizables con un alto grado de confidencialidad y seguridad ante datos de carácter sensible, como es el caso de sistemas de control telemático para un sistema de alejamiento entre personas. Se han conocido casos de monitorización con Nagios de radares en carreteras de servicio privado, y casos de sistemas de médicos ha sido utilizado para la supervisión y mantenimiento de sistemas de respiración asistida y casos de análisis moleculares por medio de servidores de precisión con la supervisión de un sistema de monitorización con Nagios.

# Apéndice

---

## 7. Apéndice: Glosario de términos

**TFG:** Trabajo Final de Grado.

**TIC:** Tecnologías de la Información y la Comunicación.

**Chequeo pasivo:** Nagios envía un chequeo y espera respuesta para actualizar la información del estado de dicha alerta.

**Chequeo activo:** el servidor a monitorizar ejecuta scripts por cron, enviando información del estado de sus alertas a Nagios.

**Chequeo:** scripts que se ejecutan para comprobar el estado de un servidor o servicio.

**Disponibilidad:** factor que se asegura de que los servicios monitorizados estén disponibles durante el período de tiempo acordado.

**NAGIOS;** sistema de monitorización automatizado con multitud de posibilidades de monitorizar servidores de diferentes sistemas operativos y servicios dentro de servidores y dispositivos. [11]

**NAGVIS:** aplicación web integrable en Nagios, que facilita la vista de las alertas de Nagios en diferentes imágenes y mapas de red, desarrollados por un administrador que disponga de conocimiento de la infraestructura. [9]

**SO:** Sistema Operativo.

**UNIX:** sistema operativo portable, multiusuario y multitarea. Los más importantes son: Solaris, AIX, HP-UX y Mac OS X. También existen: Red Hat, Suse Linux, Debian y \*BSD.

**LINUX:** similar a Unix denominado Linux por estar bajo los términos de GPL, (Licencia Pública General), siendo de software libre.

**Cron:** administrador de procesos en segundo plano para ejecutar ficheros con un determinado formato.

**Demonio:** nomenclatura usada para definir un servicio en Unix/Linux.

**HTML:** *HyperText Markup Language* o lenguaje de marcas de hipertexto, es una estándar para elaborar páginas web.

**Xinetd:** *Extended Internet Daemon* o Demonio Extendido de Internet, que es un servicio usado por servidores Unix/Linux para administrar las conexiones provenientes desde Internet.

**Dirección IP:** número correspondiente al nivel de red que identifica a un dispositivo (normalmente un servidor o Switch), de forma única.

**Java:** lenguaje de programación para aplicaciones software que puede ser usado en multitud de plataformas.

**Log:** registro de datos de una aplicación o servidor para su posterior visualización.

**Perl:** lenguaje de programación, con características similares al lenguaje C, interpretado por "Shell" (sh).

**SSH:** (Secure Shell) protocolo de acceso a equipos Linux/Unix remotos por medio del puerto 22.

**SSL:** (Secure Sockets Layer) protocolo criptográfico que proporciona una comunicación segura por una red determinada.

**Script:** documento de texto con permisos de ejecución que genera unos datos de salida que al enviarlos a Nagios son entendidos por éste y presentados en su interfaz.

**Shell:** interfaz que es usada para poder manipular todos los ficheros existentes en un servidor dentro de su sistema operativo.

**SNMP:** (Protocolo Simple de Administración de Red) protocolo de red que permite el intercambio de datos de administración entre diferentes dispositivos de red.

**NSCA:** protocolo de comunicación para la recepción de chequeos remotos de un servidor Linux/Unix, para actualizar periódicamente el estado en Nagios.

**NRPE:** plugin para ejecutar comandos remotos en servidor Linux/Unix. Instalando nrpe en el cliente y en el servidor, se enviarán chequeos activos, para la recepción de datos a Nagios.

**Tar:** formato de compresión de ficheros.

**UML:** Lenguaje de modelado

**AdminNagios:** administrador de la aplicación de Nagios con permisos de administración sobre el servidor y la propia aplicación.

**Software:** soporte lógico de un sistema informático.

**Hardware:** soporte físico de un sistema informático.

**LAN:** red de área local, donde es posible la interconexión de varios servidores, ordenadores, dispositivos y periféricos.

**ESXi:** es una plataforma de virtualización, a modo de hipervisor para todos los servidores incluidos en dicho servidor físico.

**Firewall:** es una parte fundamental de una red o sistema diseñada para el bloqueo de accesos no deseados, permitiendo los accesos autorizados.

**DMZ:** zona perimetral segura ubicada entre una red interna de una organización y una externa, con el fin de hacer seguras las comunicaciones.

**CU:** caso de uso, donde se describen todos los pasos que se deben realizar para llevar a cabo algún proceso.

**RF:** requisito funcional. Función del sistema que describe un comportamiento deseado.

**RNF:** requisito no funcional. Función del sistema que describe un comportamiento que es necesario.

**PV:** pruebas de verificación. Paso necesario para la comprobación del correcto funcionamiento de un sistema, ya que sin pruebas no se puede confirmar que un sistema es estable.

**SLA:** *Server Level Agreement* o Acuerdos de Nivel de Servicio según nomenclatura ITIL, indica los tiempos de respuesta establecidos en los servicios contratados.

**ESX:** hipervisor de sistemas virtuales para la generación de máquinas en una misma plataforma.

**YUM:** comando de instalación de paquetes rpm en CentOS y Fedora.

**Hostgroups:** grupos de máquinas para la vista de Nagios web.

**Servicegroups:** grupos de servicios para la vista de Nagios web.

**Hosts:** definición de servidores en Nagios.

**Services:** definición de servicios asociados a servidores en Nagios.

**SGBD:** Sistema Gestor de Bases de Datos, necesario para la administración de todos los datos almacenados en cualquier tipo de base de datos.

**PHP:** (*Hypertext Preprocessor*) lenguaje de programación usado mayormente para la programación de aplicaciones web.

**PECL:** (*PHP Extension Community Library*) repositorio para extensiones de código en PHP, administrado por una gran comunidad de desarrolladores de este lenguaje.

**Socket:** constituyen los canales de comunicación para que dos máquinas intervengan un flujo de datos con los mismos parámetros de comunicaciones, y entendibles por ambas.

## **8. Apéndice: Manuales**

A continuación, se van a detallar los manuales de administrador y usuario para la administración y el uso de ambos perfiles. Además se incluyen multitud de scripts realizados para las pruebas de los chequeos recibidos por Nagios, para incluir más script en el software y de este modo poder monitorizar más elementos y crear más tipos de chequeos.

### **8.1. Manual de Administrador (Nagios Core 4)**

Se va proceder a redactar en varios puntos, todos los pasos necesarios para poder instalar de forma correcta el sistema de monitorización Nagios, con las recomendaciones que realiza la propia empresa suministradora del software. Se detallan todos los pasos a nivel configuración, sobre todo en el comienzo, para que no se generen problemas en los primeros pasos [16].

Además de comentar los pasos a seguir para instalar los paquetes necesarios y la configuración ineludible, se presentan varios casos prácticos implementados en los servidores creados para el estudio e investigación del presente proyecto, con diferentes sistemas operativos y por lo tanto, diferentes tipos de chequeos y de comandos a usar en cada uno de ellos.

#### **8.1.1. Instalar Sistema Operativo**

Para poder realizar una instalación y configuración limpia de Nagios, después de muchas pruebas, desde Nagios® recomiendan instalar su software en el Sistema Operativo libre CentOS, que realiza una instalación mínima y dispone de todos los paquetes necesarios para el funcionamiento correcto de Nagios.

CentOS tiene una instalación básica, en la que de forma gráfica, en un par de menús de fácil configuración tendrá visible el escritorio de acceso a su servidor. Para poder administrar de forma correcta el sistema, es necesario tener un conocimiento mínimo de la ubicación de los ficheros que se van a manejar en el servidor, ya que sin el conocimiento de las rutas existentes, tendremos problemas a la hora de administrar el sistema y de realizar búsquedas y mejoras futuras en él.

Las rutas de ficheros que deben conocerse para la correcta administración del sistema, son los que se determinan en la siguiente imagen (véase Ilustración 56).

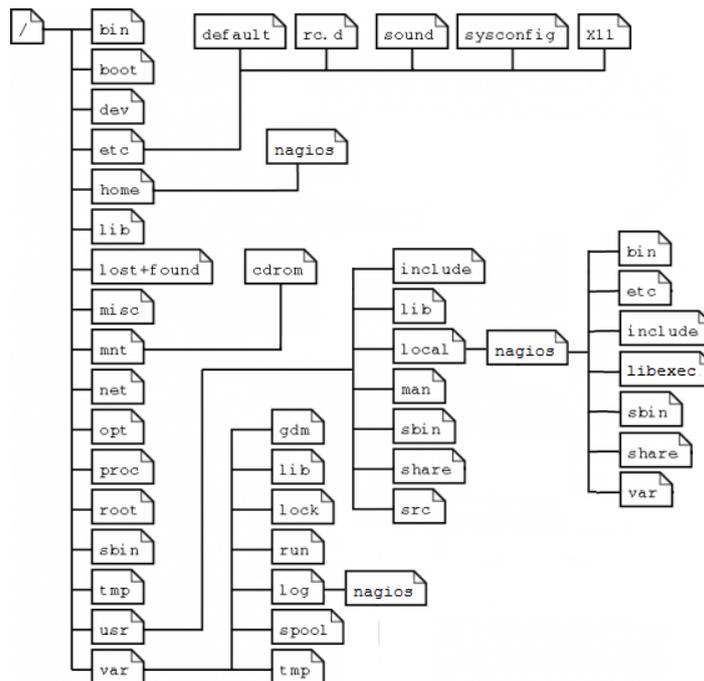


Ilustración 56.- Estructura ficheros Linux/Unix

En este árbol, se observan todos los ficheros que están disponibles en la entidad raíz del sistema, además de las rutas de ficheros de Nagios, que serán detallados en puntos posteriores, donde se observarán los ficheros importantes y que es necesario conocer, tanto de la estructura `/usr/local/nagios`, como de los ficheros existentes en `/var/log/nagios`.

Los ficheros contenidos en `/usr/local/nagios` serán los ficheros más importantes para nuestra administración del sistema Nagios, ya que dentro de cada uno de esos directorios incluidos, estarán contenidos los ficheros de configuración y ejecución de Nagios.

NOTA: Siempre se deberá disponer del usuario "root" para poder instalar y configurar los paquetes de instalación y sus dependencias para no tener problema a la hora de modificar y escribir en ubicaciones del sistema.

### 8.1.2. Instalación dependencias necesarias

Para poder proceder a instalar el software de Nagios se ha seguido el manual de Nagios Core 4 Documentation, de la web oficial de Nagios [6], donde aparecen todos los pasos de instalación y configuración para un correcto funcionamiento del mismo.

Primero, comenzar con la instalación los paquetes de dependencias que son los siguientes:

```
[root@nagios ~]# yum groupinstall "Development Tools"
[root@nagios ~]# yum install wget
```

```
[root@nagios ~]# yum localinstall perl-Crypt-DES-2.05-9.el6.x86_64.rpm perl-Crypt-DES-2.05-9.el6.x86_64.rpm fping-3.10-1.el6.rf.x86_64.rpm
[root@nagios ~]# yum install -y wget httpd php gcc glibc glibc-common gd gd-devel make net-snmp net-snmp-perl openldap openldap-devel mysql-devel mysql-libs net-snmp-devel postgresql-libs postgresql-devel dos2unix
```

Antes de empezar a instalar Nagios en CentOS es necesario instalar otros paquetes y es posible que ya tengamos instalado algunos de estos, ya que muchos de ellos son básicos para el funcionamiento de los sistemas de hoy en día.

### 8.1.3. Usuarios y grupos

Antes de instalar, debemos configurar el sistema operativo, para la administración correcta del software. Para poder realizar una instalación correcta de Nagios, deberemos crear un nuevo usuario llamado “nagios” y la cuenta de grupo “nagcmd”, así como asignarle la contraseña correspondiente a la cuenta. También, agregaremos el usuario “nagios” y el usuario “apache” al grupo “nagcmd”, para que ambos estén autorizados a trabajar con la aplicación de Nagios.

```
[root@nagios ~]# useradd nagios
[root@nagios ~]# groupadd nagcmd
[root@nagios ~]# usermod -a -G nagcmd nagios
[root@nagios ~]# passwd nagios
[root@nagios ~]# usermod -a -G nagcmd apache
```

### 8.1.4. Descarga e instalación de Nagios y Plugins

A continuación, procedemos a instalar descargar el software de Nagios Core 4, siendo recomendando desde la URL siguiente, ya que es la última versión y la oficial:

```
[root@nagios ~]# wget
https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.1.1.tar.gz
```

Una vez creado el usuario de nagios en el servidor y el grupo asociado, procedemos a descomprimir e instalar el paquete de Nagios que antes hemos descargado:

```
[root@nagios ~]# tar -xzvf nagios-4.0.7.tar.gz
[root@nagios ~]# cd nagios-4.0.7
[root@nagios ~]# ./configure --with-command-group=nagcmd
[root@nagios ~]# make all
[root@nagios ~]# make install
[root@nagios ~]# make install-init
[root@nagios ~]# make install-commandmode
[root@nagios ~]# make install-config
[root@nagios ~]# make install-webconf
```

Copiamos estos archivos del paquete de Nagios y que usaremos más adelante:

```
[root@nagios ~]# cp -R contrib/eventhandlers/ /usr/local/nagios/libexec/  
[root@nagios ~]# chown -R nagios:nagios  
/usr/local/nagios/libexec/eventhandlers
```

Durante esta instalación, únicamente nos solicitará si queremos cambiar la configuración predefinida por Nagios, en cuanto a rutas de ficheros de instalación y ejecución de scripts.

Nagios recomienda dejar la instalación por defecto para no tener problemas de inconsistencia de datos, por lo que todos estarán almacenados en /usr/local/nagios/.

```
[root@nagios ~]# service nagios start  
[root@nagios ~]# service httpd start
```

Y para poder acceder deberemos crear un usuario para habilitar el acceso a la interfaz web, que en este caso será “nagiosadmin”.

```
[root@nagios etc]# htpasswd -c /usr/local/nagios/etc/htpasswd.users  
nagiosadmin  
New password:  
Re-type new password:  
Adding password for user nagiosadmin
```

Una vez instalado y configurado por defecto, podremos acceder a <http://localhost/nagios>, donde nos mostrará la interfaz web de Nagios (véase Ilustración 57), con el servidor localhost (propio servidor de Nagios) monitorizado.



Ilustración 57.- Primer inicio de la interfaz web de Nagios

Para asegurar la disponibilidad completa de Nagios, aunque se reinicie de forma inesperada, deberemos asegurar el menor impacto de indisponibilidad del mismo, por lo que necesitamos agregar los servicios de nagios y apache con el comando chkconfig al inicio del sistema, para que éste arranque automáticamente con el servidor.

```
[root@nagios ~]# chkconfig --add nagios
[root@nagios ~]# chkconfig --level 35 nagios on
[root@nagios ~]# chkconfig --add httpd
[root@nagios ~]# chkconfig --level 35 httpd on
```

Para que la nueva configuración esté funcionando, deberemos reiniciar el servicio de Nagios y Apache:

```
[root@nagios ~]# service nagios restart
[root@nagios ~]# service httpd restart
```

A continuación, para poder monitorizar servidores y servicios, es necesario instalar los Plugins de Nagios, y así poder completar el sistema de monitorización.

Descargamos los Plugins de la página web oficial:

```
[root@nagios ~]# wget http://nagiosplugins.org/download/nagios-plugins-2.0.2.tar.gz
```

Una vez descargados se procede a su instalación, siguiendo los siguientes pasos:

```
[root@nagios ~]# tar -xzvf nagios-plugins-2.0.2.tar.gz
[root@nagios ~]# cd nagios-plugins-2.0.2
[root@nagios ~]# ./configure --with-nagios-user=nagios --with-nagios-group=nagios
[root@nagios ~]# make
[root@nagios ~]# make install
```

Cambiamos los permisos de la carpeta donde hemos instalado los plugins, para que los pueda utilizar nagios este paso lo tendremos que hacer cada vez que instalemos un plugin nuevo.

```
[root@nagios ~]# chown nagios:nagios /usr/local/nagios/libexec
[root@nagios ~]# chmod -R 777 /usr/local/nagios/libexec
```

En la ruta `/usr/local/nagios/libexec` estarán todos los Plugins descargados y listos para su uso en cualquier monitorización de cualquier servidor.

Para poder acceder a la información de estados e históricos de un servidor y/o servicio en Nagios, se deberá tener instalado un módulo de Nagios llamado LiveStatus, que permitirá acceder a toda esta información y mostrarlos en tiempo real, mediante sockets, y se podrá consultar en todo momento en la interfaz web de Nagios. Para descargarlo se debe acceder a:

```
[root@nagios ~]# wget http://mathias-kettner.de/download/mk-livestatus-1.2.5i2pl.tar.gz
```

Y para su descompresión e instalación, se deben realizar los siguientes pasos:

```
[root@nagios ~]# tar xfvz mk-livestatus-1.2.5i2p1.tar.gz
[root@nagios ~]# cd mk-livestatus-1.2.5i2p1
[root@nagios ~]# ./configure --with-nagios4
[root@nagios ~]# make
[root@nagios ~]# make install
```

Cuando finaliza la instalación de este módulo, deberá ser registrado en el fichero nagios.cfg, agregando la siguiente información:

```
event_broker_options=-1
```

```
broker_module=/usr/local/lib/mk-livestatus/livestatus.o /usr/local/nagios/var/rw/live
```

Para poder trabajar con este módulo, será necesario tener un gestor de servicios llamado Xinetd, para que actúe de pasarela entre el socket creado por el módulo LiveStatus y el puerto TCP.

```
[root@nagios ~]# yum install xinetd
[root@nagios ~]# chkconfig xinetd on
```

Para poder gestionar dicho servicio, se deberá crear dentro del directorio /etc/xinetd.d un nuevo servicio "/etc/xinetd.d/livestatus", y copiará el fichero generado en "/usr/local/lib/mk-livestatus/livestatus.o" en la nueva ruta mencionada. Una vez realizado este paso, es necesario iniciar el servicio para comprobar su correcto funcionamiento.

```
[root@nagios ~]# service xinetd start
```

Ahora Nagios se mostrará de la siguiente manera (véase ilustración 58), con la monitorización del propio servidor, que es la que nos aparece por defecto en el primer acceso a Nagios.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	08-27-2014 11:43:08	57d 19h 25m 51s	1/4	OK - load average: 0.01, 0.02, 0.05
	Current Users	OK	08-27-2014 11:39:30	57d 19h 25m 13s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	08-27-2014 11:42:08	35d 22h 33m 35s	1/4	HTTP OK: HTTP/1.1 200 OK - 453 bytes in 0.001 second response time
	PING	OK	08-27-2014 11:39:55	57d 19h 23m 58s	1/4	PING OK - Packet loss = 0%, RTA = 0.04 ms
	Root Partition	OK	08-27-2014 11:41:34	57d 19h 23m 21s	1/4	DISK OK - free space: / 26587 MB (89% inode=98%):
	SSH	OK	08-27-2014 11:42:48	57d 19h 22m 43s	1/4	SSH OK - OpenSSH_6.0p1 Debian-4+deb7u1 (protocol 2.0)
	Swap Usage	OK	08-27-2014 11:41:51	57d 19h 22m 8s	1/4	SWAP OK - 100% free (1020 MB out of 1020 MB)
	Total Processes	OK	08-27-2014 11:43:53	57d 19h 21m 28s	1/4	PROCS OK: 48 processes with STATE = RSZDT

Ilustración 58.- Primera vista localhost Nagios

Para poder generar gráficas desde la vista principal de Nagios, será necesario instalar PHP4Nagios, que es un complemento extra para este fin, con la opción de generar gráficas determinando el período en el que se quieren visualizar. Antes de instalar PNP4Nagios, es necesario instalar RRDtool, para la lectura de información en Nagios.

```
[root@nagios ~]# BUILD_DIR=/tmp/rrdbuild
[root@nagios ~]# INSTALL_DIR=/opt/rrdtool-1.4.3
[root@nagios ~]# mkdir -p $BUILD_DIR
[root@nagios ~]# cd $BUILD_DIR
[root@nagios ~]# wget http://oss.oetiker.ch/rrdtool/pub/rrdtool-1.4.3.tar.gz
[root@nagios ~]# tar -xvzf rrdtool-1.4.3.tar.gz
[root@nagios ~]# cd rrdtool-1.4.3
[root@nagios ~]# yum install cairo-devel libxml2-devel pango-devel pango
libpng-devel freetype freetype-devel libart_lgpl-devel
[root@nagios ~]# ./configure --prefix=$INSTALL_DIR && make && make install
```

La instalación PNP4Nagios que se deberá realizar, es la siguiente [31]:

```
[root@nagios ~]# wget
http://sourceforge.net/projects/pnp4nagios/files/latest/download
[root@nagios ~]# mv download pnp4nagios-0.6.25.tar.gz
[root@nagios ~]# tar -xvzf pnp4nagios-0.6.25.tar.gz
[root@nagios ~]# cd pnp4nagios-0.6.25
[root@nagios ~]# ./configure && make && make install
[root@nagios ~]# service httpd restart
[root@nagios ~]# service nagios restart
```

Después de esta instalación, podremos ver el acceso a las gráficas desde la vista de hostgroups y dentro de cada servidor (véase Ilustraciones 57 y 58).



Ilustración 59.- Vistas de iconos de gráficas en Nagios

Situándonos por encima de los iconos, podrá visualizarse una pequeña previsualización de las gráficas de estados del servicio que deseemos (véase Ilustración 58).

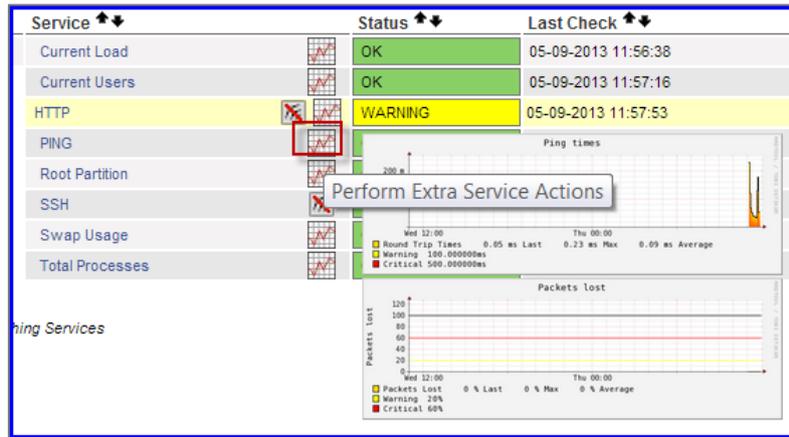


Ilustración 60.- Previsualización de gráficas en Nagios

Pinchando en el icono, se podrá visualizar toda la información relativa a dicho chequeo con los periodos de 4 horas, 25 horas, una semana, un mes o un año, según deseemos visualizar la información (véase Ilustración 61).

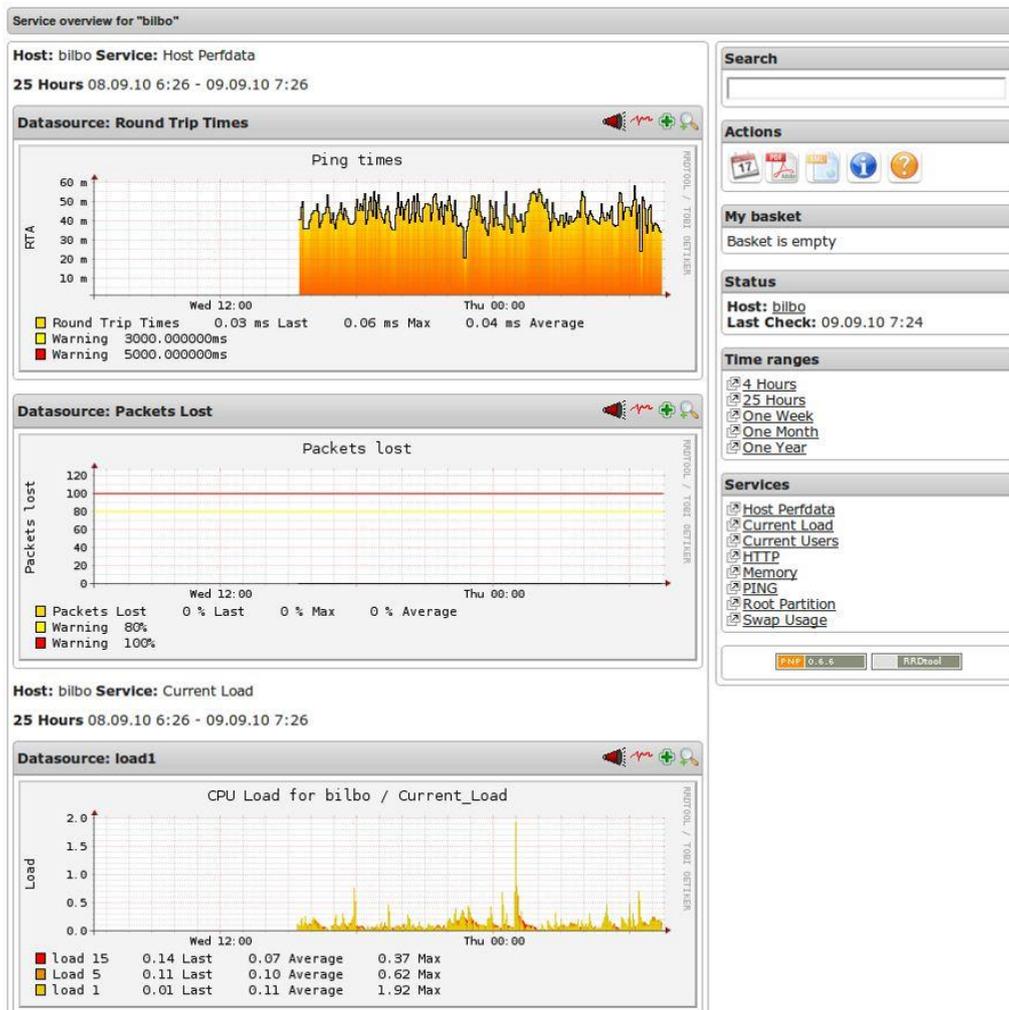


Ilustración 61.- Gráficas por PNP4Nagios

### 8.1.5. Configuración de Nagios y Plugins

Para comenzar con la configuración de Nagios, se van a definir en primer lugar la estructura de ficheros, en la cual definiremos los servidores y servicios a monitorizar. Donde primeramente, le daremos permisos al usuario de Nagios, tanto hacerle propietario como tener todos los permisos, para crear, modificar y eliminar todos los ficheros contenidos en la ruta /usr/local/nagios/etc.

```
[root@nagios ~]# chown nagios:nagios /usr/local/nagios/etc/ -R
```

Una vez dados los permisos, deberemos definir cuál es el fichero que contiene toda la información principal de Nagios. Este fichero está ubicado en /usr/local/nagios/etc/ y se llama cgi.cfg.

```
# MAIN CONFIGURATION FILE
# This tells the CGIs where to find your main configuration file.
# The CGIs will read the main and host config files for any other
# data they might need.

main_config_file=/usr/local/nagios/etc/nagios.cfg
```

Ilustración 62.- Fichero de configuración de Nagios

Definimos también la ruta donde se ubicarán los ficheros de las vistas y la configuración visual de la interfaz web de Nagios.

```
# PHYSICAL HTML PATH
# This is the path where the HTML files for Nagios reside. This
# value is used to locate the logo images needed by the statusmap
# and statuswrl CGIs.

physical_html_path=/usr/local/nagios/share
```

Ilustración 63.- Ruta de vistas de Nagios

Por defecto, Nagios tiene un auto-chequeo donde se podría detectar si hay fallos en la configuración tras cualquier modificación, y está definido en el fichero cgi.cfg:

```
# PROCESS CHECK COMMAND
# This is the full path and filename of the program used to check
# the status of the Nagios process. It is used only by the CGIs
# and is completely optional. However, if you don't use it, you'll
# see warning messages in the CGIs about the Nagios process
# not running and you won't be able to execute any commands from
# the web interface. The program should follow the same rules
# as plugins: the return codes are the same as for the plugins,
# it should have timeout protection, it should output something
# to STDOUT, etc.
#
# Note: If you are using the check_nagios plugin here, the first
# argument should be the physical path to the status log, the
# second argument is the number of minutes that the status log
# contents should be "fresher" than, and the third argument is the
# string that should be matched from the output of the 'ps'
# command in order to locate the running Nagios process. That
# process string is going to vary depending on how you start
# Nagios. Run the 'ps' command manually to see what the command
# line entry for the Nagios process looks like.
#
#process_check_command=/usr/local/nagios/libexec/check_nagios /usr/local/nagios/var/status.log 5 '/usr/local/nagios/bin/nagios'
#nagios_check_command=/usr/local/nagios/libexec/check_nagios /usr/local/nagios/var/status.log 5 '/usr/local/nagios/bin/nagios'
```

Ilustración 64.- Comprobación Nagios

Para comprobarlo, únicamente se debe ejecutar el comando siguiente:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

Si no aparece ningún error, nos mostrará una pantalla como la que aparece en la siguiente imagen (véase Ilustración 65).

```
[root@nagios ~]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.0.8
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-12-2014
License: GPL

Website: http://www.nagios.org
Reading configuration data...
  Read main config file okay...
  Read object config files okay...

Running pre-flight check on configuration data...

Checking objects...
  Checked 4 services.
  Checked 5 hosts.
  Checked 2 host groups.
  Checked 0 service groups.
  Checked 3 contacts.
  Checked 2 contact groups.
  Checked 26 commands.
  Checked 5 time periods.
  Checked 0 host escalations.
  Checked 0 service escalations.
Checking for circular paths...
  Checked 5 hosts
  Checked 0 service dependencies
  Checked 0 host dependencies
  Checked 5 timeperiods
Checking global event handlers...
Checking obsessive compulsive processor commands...
Checking misc settings...

Total Warnings: 0
Total Errors: 0

Things look okay - No serious problems were detected during the pre-flight check
```

Ilustración 65.- Compilación correcta de Nagios

Si se ejecuta dicho comando y en el código existe algún error, mostrará una pantalla similar a la siguiente (véase Ilustración 66).

```
[root@nagios~]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
Nagios Core 4.0.8
Copyright (c) 2009-present Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 08-12-2014
License: GPL

Website: http://www.nagios.org
Reading configuration data...
  Read main config file okay...
Error: Unexpected token or statement in file '/usr/local/nagios/etc/hosts/hosts.cfg' on line 20.
  Error processing object config files!

***> One or more problems was encountered while processing the config files...

Check your configuration file(s) to ensure that they contain valid
directives and data defintions.  If you are upgrading from a previous
version of Nagios, you should be aware that some variables/definitions
may have been removed or modified in this version.  Make sure to read
the HTML documentation regarding the config files, as well as the
'Whats New' section to find out what has changed.
```

Ilustración 66.- Compilación incorrecta de Nagios

En la anterior imagen nos indica que en la línea 20 hay un error en el fichero hosts.cfg, y por lo tanto, hasta que no modifique dicha línea, no podemos compilar de nuevo e impedirá el correcto funcionamiento de Nagios.

Este tipo de ejecución se realiza para poder comprobar que toda la configuración está correcta antes de reiniciar el servicio.

```
[root@nagios ~]# service nagios reload/restart
```

En el caso de no haber realizado la comprobación, pero si haber reiniciado el servicio, Nagios se parará hasta quitar dichos errores, y tendrás que ejecutar dicho comando para saber dónde está fallando la compilación del código modificado.

Para poder dar permisos a los usuarios de Nagios a acceder y trabajar con él desde la interfaz web, se deberá ejecutar lo siguiente para poder crear un usuario de acceso a la aplicación:

```
[root@nagios etc]# htpasswd /usr/local/nagios/etc/htpasswd.users xxxxxx
New password:
Re-type new password:
Adding password for user xxxxxx
```

NOTA: Se tiene que cambiar el contexto de dichos directorios, y vamos a realizar esto de manera recursiva para que afecte a todos los archivos que se encuentran en su interior. Para esto vamos a utilizar el comando chcon, y con la ruta de la instalación de Nagios.

```
[root@nagios ~]# chcon -R -t httpd_sys_content_t /usr/local/nagios
```

Reiniciamos el servicio de Apache para que la nueva configuración se actualice en el servicio que está arrancado en la máquina.

```
[root@nagios ~]# service httpd restart
```

Deberá escribirse en el fichero cgi.cfg en las líneas donde se escriben los usuarios que tienen acceso a la interfaz web.

```
# GLOBAL HOST/SERVICE VIEW ACCESS
# These two options are comma-delimited lists of all usernames that
# can view information for all hosts and services that are being
# monitored. By default, users can only view information
# for hosts or services that they are contacts for (unless you
# you choose to not use authorization). You may use an asterisk (*)
# to authorize any user who has authenticated to the web server.

authorized_for_all_services=nagiosadmin
authorized_for_all_hosts=nagiosadmin
```

Ilustración 67.- Autorizar acceso web a Nagios

En este momento, se podrá acceder a <http://localhost/nagios> con el usuario nagiosadmin y la contraseña que se haya definido (véase Ilustración 68).

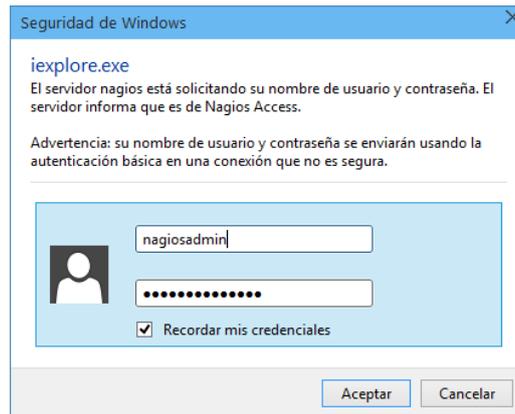


Ilustración 68.- Acceso a la interfaz web de Nagios

Para poder crear usuarios con todos los privilegios para acceder a la interfaz de Nagios, se crearán de la siguiente manera.

```
[root@nagios ~]# htpasswd -c /usr/local/nagios/etc/htpasswd.users "usuario"
```

Para la creación de usuarios de sólo lectura, se realizará de la siguiente manera.

```
[root@nagios ~]# htpasswd -m /usr/local/nagios/etc/htpasswd.users "usuario"
```

Es posible realizar la autorización con diferentes perfiles en el fichero `cgi.cfg`, en la ruta `/usr/local/nagios/etc`, donde una vez dados de alta con el comando `htpasswd`, podremos autorizar la entrada a la interfaz web. Los tipos de acceso que se autorizan, son los siguientes:

```
authorized_for_system_information
authorized_for_system_commands
authorized_for_configuration_information
authorized_for_all_hosts
authorized_for_all_host_commands
authorized_for_all_services
authorized_for_all_service_commands
```

Durante la configuración de Nagios, es necesario mencionar en varias ocasiones la importancia del fichero "nagios.cfg", en el cual se definirán todos los ficheros que queremos que se lean para presentarse en la interfaz web.

Se registrarán muchas líneas de configuración en él, comenzando por definir el fichero de log de la aplicación (véase Ilustración 69).

```
# LOG FILE
# This is the main log file where service and host events are logged
# for historical purposes. This should be the first option specified
# in the config file!!!

log_file=/usr/local/nagios/var/nagios.log
```

Ilustración 69.- Configuración fichero de log en Nagios

En las líneas sucesivas de dicho fichero, será donde se definirá todos ficheros de configuración de definición de servidores, servicios, períodos, chequeos, etc. Si un fichero de hosts, no se define aquí, no será visible en la compilación de Nagios y no se mostrará en la aplicación web.

```
## Timeperiods
cfg_file=/usr/local/nagios/etc/timeperiods.cfg
## Commands
cfg_file=/usr/local/nagios/etc/commands.cfg
## Contacts
cfg_file=/usr/local/nagios/etc/contacts.cfg
cfg_file=/usr/local/nagios/etc/contactgroups.cfg
cfg_file=/usr/local/nagios/etc/contacttemplates.cfg
## Hosts
cfg_file=/usr/local/nagios/etc/hosts/hosts.cfg
cfg_file=/usr/local/nagios/etc/hosttemplates.cfg
cfg_file=/usr/local/nagios/etc/hostgroups/hostgroups.cfg
cfg_file=/usr/local/nagios/etc/hostextinfo.cfg
cfg_file=/usr/local/nagios/etc/hostescalations.cfg
cfg_file=/usr/local/nagios/etc/hostdependencies.cfg
## Services
cfg_file=/usr/local/nagios/etc/services/services.cfg
cfg_file=/usr/local/nagios/etc/servicetemplates.cfg
cfg_file=/usr/local/nagios/etc/servicegroups.cfg
cfg_file=/usr/local/nagios/etc/serviceextinfo.cfg
```

Ilustración 70.- Definición de ficheros de configuración Nagios

Nagios actualmente cuenta con varios protocolos de monitorización, siendo los más conocidos y extendidos NSCA y NRPE. En ambos casos, el servidor o equipo que queramos monitorizar, deberá tener instalado el plugins que elijamos para que los datos los reciba Nagios.

Al igual que realizamos la descarga de Nagios y sus plugins generales, descargamos el plugin de NSCA desde la web oficial de Nagios.

```
[root@nagios ~]# wget
http://prdownloads.sourceforge.net/sourceforge/nagios/nsca-2.9.1.tar.gz
[root@nagios ~]# gunzip nsca-2.9.1.tar.gz
[root@nagios ~]# tar -xzvf nsca-2.9.1.tar
[root@nagios ~]# cd nsca-2.9.1
[root@nagios ~]# ./configure --with-nsca-user=nagios --with-nsca-grp=nagcmd
[root@nagios ~]# make all
```

Ahora se copian los archivos de ejemplo en el directorio de Nagios “/usr/local/nagios/etc” y les damos los permisos necesarios:

```
[root@nagios ~]# cp sample-config/nsca.cfg sample-config/send_nsca.cfg
/usr/local/nagios/etc/

[root@nagios ~]# cp src/send_nsca src/nsca /usr/local/nagios/bin/

[root@nagios ~]# chown nagios:nagios /usr/local/nagios/bin/nsca
/usr/local/nagios/bin/send_nsca

[root@nagios ~]# chown nagios:nagcmd /usr/local/nagios/etc/nsca.cfg
/usr/local/nagios/etc/send_nsca.cfg

[root@nagios ~]# chmod g+r /usr/local/nagios/etc/nsca.cfg
```

Para que Nagios pueda recibir todos los chequeos por el servicio NSCA, se debe modificar el archivo de configuración nsca.cfg en el caso en el que se quiera poner algún método de cifrado en los paquetes enviados y recibidos por

Nagios, y proporcionar a todos los ficheros de los permisos del usuario "nagios".

```
[root@nagios ~]# vi /usr/local/nagios/etc/nsca.cfg
[root@nagios ~]# chown nagios:nagios /usr/local/nagios/bin/nsca
/usr/local/nagios/bin/send_nsca

[root@nagios ~]# chown nagios.nagcmd /usr/local/nagios/etc/nsca.cfg
/usr/local/nagios/etc/send_nsca.cfg
```

Se podrá verificar que NSCA funciona correctamente, ejecutando el siguiente comando:

```
[root@nagios ~]# /usr/local/nagios/bin/nsca -c /usr/local/nagios/etc/nsca.cfg
```

Confirmamos que el proceso esta ejecutándose si disponemos de la siguiente salida (véase Ilustración 71):

```
[root@nagios ~]# ps -ef | grep -v grep | grep -i nsca
[root@nagios ~]# netstat -planet | grep 5667
```

```
[root@nagios ~]# netstat -planet | grep 5667
tcp        0      0 0.0.0.0:5667          0.0.0.0:*            LISTEN    0      412812868  24904/xinetd
```

Ilustración 71.- Arranque correcto de puerto 5667 (NSCA)

Para poder poner en funcionamiento el servicio NSCA, deberemos agregarlo al fichero services donde reservaremos el puerto dedicado a la recepción de dichas conexiones, donde Nagios recibirá los datos de los servidores cliente.

```
[root@nagios ~]# vi /etc/services
```

Y agregamos la siguiente línea:

```
nsca      5667/tcp  # NSCA
```

Copiamos el archivo de ejemplo nsca.xinetd dentro del directorio /etc/xinetd.d/nsca:

```
[root@nagios ~]# cp sample-config/nsca.xinetd /etc/xinetd.d/nsca
```

Matamos el proceso NSCA ya que debemos pararlo completamente para poder iniciarlo con la nueva configuración.

```
[root@nagios ~]# rm /var/run/nsca.pid
```

Configuramos el archivo /etc/xinetd.d/nsca:

```
[root@nagios ~]# vi /etc/xinetd.d/nsca
```

```
only_from      = "ipaddress_NagiosServer"

/etc/rc.d/init.d/xinetd restart
```

Comprobamos que el proceso NSCA está corriendo y el puerto está escuchando (véase Ilustración 72):

```
[root@nagios ~]# netstat -planet | grep 5667
```

```
[root@nagios ~]# netstat -planet | grep 5667
tcp        0      0 0.0.0.0:5667          0.0.0.0:*           LISTEN     0      412812868      24904/xinetd
```

Ilustración 72.- Arranque correcto 2 de puerto 5667 (NSCA)

Deberemos también poner el servicio NSCA en el arranque del sistema:

```
[root@nagios ~]# chkconfig nsca on
[root@nagios ~]# service xinetd restart
```

A continuación, se va a implementar la configuración de correo electrónico del servidor de Nagios, para el envío de alertas por este medio.

```
[root@nagios ~]# yum install sendmail sendmail-cf
```

Una vez instalado, los ficheros de configuración se encontrarán en /etc/mail/, y en dicha ruta tendremos un fichero llamado sendmail.cf.

```
[root@nagios ~]# vi /etc/mail/sendmail.cf
```

Dentro de este fichero, deberemos encontrar una línea llamada DS, donde deberemos rellenarla de la siguiente manera.

```
# "Smart" relay host (may be null)
DScorreo.dominio.es
```

NOTA: El contacto de Nagios llamado "nagios.dominio" deberá ser creado en el servidor de correo electrónico de la Universidad, en nuestro caso para que dicho servidor funcione de relay, y deje a Nagios que lo use para entregar los correos electrónicos a los contactos definidos dentro de nuestra red.

### 8.1.6. Chequeos por defecto en la instalación

Durante el proceso de instalación de los plugins de Nagios, se generan 67 chequeos automáticos, que podrán ser utilizados para solicitar la información a un servidor remoto del estado de sus servicios o procesos.

Después de instalarlos se copian en `/usr/local/nagios/libexec`, y son los siguientes:

<code>check_apt</code>	<code>check_ircd</code>	<code>check_oracle</code>	<code>check_users</code>
<code>check_breeze</code>	<code>check_jabber</code>	<code>check_overcr</code>	<code>check_wave</code>
<code>check_by_ssh</code>	<code>check_ldap</code>	<code>check_pgsql</code>	<code>eventhandlers</code>
<code>check_clamd</code>	<code>check_ldap</code>	<code>check_ping</code>	<code>negate</code>
<code>check_cluster</code>	<code>check_load</code>	<code>check_pop</code>	<code>urlize</code>
<code>check_dhcp</code>	<code>check_log</code>	<code>check_procs</code>	<code>utils.pm</code>
<code>check_dig</code>	<code>check_mailq</code>	<code>check_real</code>	<code>utils.sh</code>
<code>check_disk</code>	<code>check_mrtg</code>	<code>check_rpc</code>	
<code>check_disk_smb</code>	<code>check_mrtgtraf</code>	<code>check_sensors</code>	
<code>check_dns</code>	<code>check_mysql</code>	<code>check_simap</code>	
<code>check_dummy</code>	<code>check_mysql_query</code>	<code>check_smtp</code>	
<code>check_file_age</code>	<code>check_nagios</code>	<code>check_spop</code>	
<code>check_flexlm</code>	<code>check_nntp</code>	<code>check_ssh</code>	
<code>check_ftp</code>	<code>check_nntp</code>	<code>check_ssmt</code>	
<code>check_http</code>	<code>check_nrpe</code>	<code>check_swap</code>	
<code>check_icmp</code>	<code>check_nt</code>	<code>check_tcp</code>	
<code>check_ide_smart</code>	<code>check_ntp</code>	<code>check_time</code>	
<code>check_ifoperstatus</code>	<code>check_ntp_peer</code>	<code>check_udp</code>	
<code>check_ifstatus</code>	<code>check_ntp_time</code>	<code>check_ups</code>	
<code>check_imap</code>	<code>check_nwstat</code>	<code>check_uptime</code>	

### 8.1.7. Instalación agentes de Nagios

Para poder monitorizar servicios en servidores con Nagios, es necesaria la instalación de un agente, que dependiendo del sistema operativo, tendrá un formato u otro.

En los siguientes puntos se van a detallar los pasos que se deben realizar para instalar el agente de Nagios en un sistema operativo Windows y en otro Linux/Unix, para poder incluir sus servicios monitorizados en Nagios.

### 8.1.7.1. Instalación agentes de Nagios Windows

Para poder monitorizar todos los chequeos posibles en un servidor Windows, deberemos instalar el software llamado NSClient++ [3] [4], y los descargaremos desde la web oficial: <http://www.nsclient.org/download/>.

En nuestro caso probamos la versión NSCP-0.4.1.90-Win32. Al instalar elegimos “Custom Setup” y nos selecciona todos los componentes para instalar. Configuramos el parámetro del servidor de Nagios para que se pueda conectar a dicho servidor.

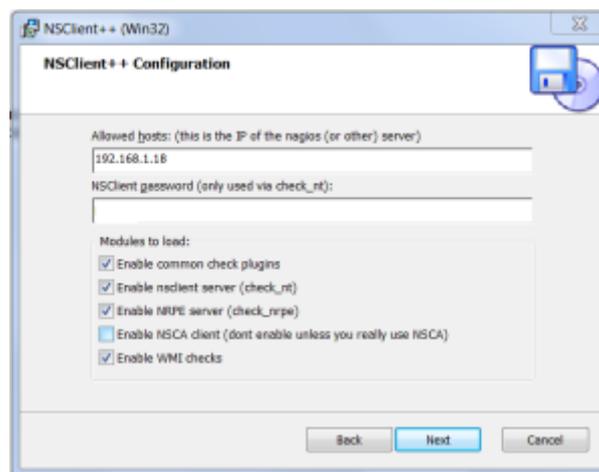


Ilustración 73.- Instalación NSClient++ en Windows

El instalador nos crea un servicio llamado “NSClient++”, registrándolo como servicio de arranque automático e inicia un proceso llamado nsc. NSClient funciona por defecto en los puertos 5666 (check\_nt) y 12489 (nrpe).

```
C:\Users\xandra> netstat -an | find "5666"
TCP 0.0.0.0:5666 0.0.0.0:0 LISTENING
TCP [::]:5666 [::]:0 LISTENING
C:\Users\xandra> netstat -an | find "12489"
TCP 0.0.0.0:12489 0.0.0.0:0 LISTENING
TCP [::]:12489 [::]:0 LISTENING
```

Para no tener problemas de conectividad, es necesaria tener una regla en el Firewall de Windows, para que pueda comunicarse con el servidor de Nagios sin errores, o la opción de deshabilitarlo (aunque no se recomienda).

Para poder comprobar que el agente de Nagios en dicho servidor Windows está funcionando correctamente, comprobaremos desde el servidor de Nagios, si está accesible la información del mismo.

```
[root@nagios ~]# cd /usr/local/nagios/libexec
[root@nagios ~]# check_nt -H 192.168.1.33 -v UPTIME -p 1248
System Uptime - 0 day(s) 2 hour(s) 30 minute(s)
```

```
[root@nagios ~]# check_nt -H 192.168.1.33 -v CPULOAD -l 60,90,95 -p 1248
CPU Load 2% (60 min average) | '60 min avg Load'=2%;90;95;0;100

[root@nagios ~]# check_nt -H 192.168.1.33 -v USEDDISKSPACE -l c -w 80 -c 90 -p 1248
c:\ - total: 488,28 Gb - used: 226,80 Gb (46%) - free 261,48 Gb (54%)

[root@nagios ~]# check_nt -H 192.168.1.33 -v SERVICESTATE -l iis_admin -p 1248
OK: All services are in their appropriate state.
```

En el siguiente punto 9.1.7, se definirá como dar de alta este tipo de chequeos en Nagios para ser visibles en la interfaz web.

### 8.1.7.2. Instalación agentes de Nagios Linux/Unix

Para monitorizar servidor Unix/Linux por medio de los agentes de Nagios, hay que valorar si conviene más realizar chequeos activos (NRPE) o pasivos (NSCA) [11], ya que dependiendo del tipo de servicio que preste dicho servidor, será más crítica su monitorización y por tanto, será recomendable que se monitorice con chequeos activos.

#### 8.1.7.2.1. Protocolo NSCA

Primeramente, es necesario acceder al servidor cliente, que queremos monitorizar. Una vez dentro, copiaremos una carpeta llamada agente\_nagios en el directorio “home” del usuario Nagios, creado en todos los servidores remotos para su correcta configuración, donde están los siguientes ficheros:

```
[root@cliente ~]# ls /home/nagios/agente_nagios/
Scripts  send_nsca.cfg  src  submit_check_result_via_nsca  tmp
```

Con el complemento NSCA es posible enviar chequeos pasivos desde servidores remotos a Nagios, siendo este método eficaz en sistemas distribuidos y redundantes.

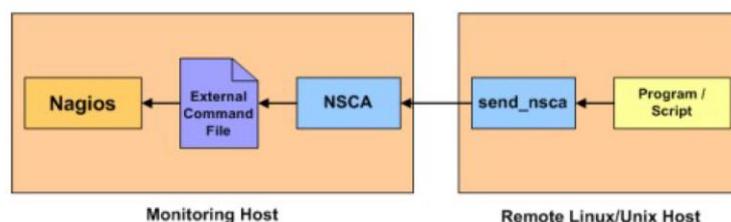


Ilustración 74.- Funcionamiento NSCA

El fichero principal para la ejecución de los chequeos por NSCA, es el llamado "submit\_check\_result\_via\_nsca". Este fichero debe ser copiado de la ruta del propio servidor de Nagios, en /usr/local/nagios/etc/send\_nsca.cfg, y renombrado si se desea.

La estructura del fichero es la siguiente:

```
#!/bin/sh

# SUBMIT_CHECK_RESULT_VIA_NSCA
# Arguments:
# $1 = host_name (Short name of host that the service is
# associated with)
# $2 = svc_description (Description of the service)
# $3 = return_code (An integer that determines the state
# of the service check, 0=OK, 1=WARNING, 2=CRITICAL,
# 3=UNKNOWN).
# $4 = plugin_output (A text string that should be used
# as the plugin output for the service check)s
#
#
printfcmd="/usr/bin/printf"

NscaBin="/usr/local/nagios/libexec/send_nsca"
NscaCfg="/usr/local/nagios/etc/send_nsca.cfg"
NagiosHost="nagioshost"

# Fire the data off to the NSCA daemon using the send_nsca script
$printfcmd "%s\t%s\t%s\t%s\n" "$1" "$2" "$3" "$4" | $NscaBin -H $NagiosHost -c $NscaCfg

# EOF
```

Para poder tener los datos correctamente registrados en el servidor a monitorizar, se deberán cambiar las siguientes líneas anteriores marcadas en rojo, a la configuración recomendada:

```
NscaBin="/home/nagios/agente_nagios/send_nsca"
NscaCfg="/home/nagios/agente_nagios/send_nsca.cfg"
NagiosHost="192.168.1.5"
```

Una vez configurado, deberemos acceder al directorio Scripts, donde crearemos los chequeos, estarán todos los necesarios para ejecutar comando en dicho servidor y que sean enviados a Nagios.

Es necesario recordar que los chequeos pasivos se ejecutan en el propio servidor a monitorizar y es él el que envía los datos a Nagios. Para poder ejecutar estos scripts, se deberán definir en el "cron" de la máquina, de la siguiente manera:

```
[root@cliente ~]# crontab -l
05,20,35,50 * * * * /home/nagios/agente_nagios/Scripts/Comprueba discos 80 90
00,15,30,45 * * * * /home/nagios/agente_nagios/Scripts/Comprueba memoria 5 2
03,18,33,48 * * * * /home/nagios/agente_nagios/Scripts/Comprueba CPU 5 3
09,19,39,49 * * * * /home/nagios/agente_nagios/Scripts/Comprueba swap 20 10
```

El archivo de crontab tiene la siguiente estructura de izquierda a derecha:

- Minutos (rango de 0-59)
- Horas (0-23)
- Día del mes (1-31)
- Mes (1-12)
- Día de la semana (0-6 siendo 0=Domingo)
- Path completo al script o programa que queremos ejecutar

Los campos con \* quiere decir que se ejecuta a cualquier posible valor de ese campo, por lo que nuestros scripts se ejecutarán cada 15 minutos.

### 8.1.7.2.2. Protocolo NRPE

NRPE es un complemento que nos permite ejecutar plugins en los servidores con sistema operativo Linux/Unix de forma activa.

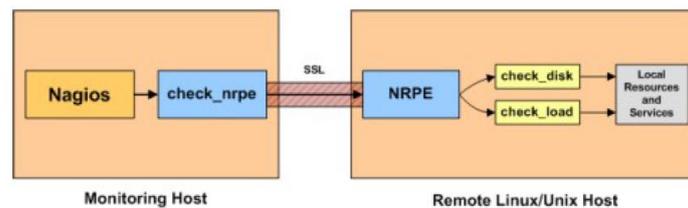


Ilustración 75.- Funcionamiento NRPE

Para poder monitorizar de forma más rápida los servidores Linux/Unix, se puede usar también el protocolo NRPE, en lugar de NSCA explicado en el punto anterior.

Una de las grandes diferencias que tienen estos dos protocolos es que NSCA únicamente tenemos que crear un directorio en el servidor monitorizado y en él ejecutaremos por medio de crontab los chequeos que queremos.

En NRPE deberemos realizar una pequeña instalación en el cliente, para que disponga el servidor de todos los chequeos que están creados en Nagios y de esta manera poder ejecutar cualquier comando. La diferencia más importante es que NRPE ejecuta chequeos en modo activo, funcionando de manera similar a la ejecución de NSClient en los servidores con sistema operativo Windows.

Para poder poner en funcionamiento NRPE en un servidor cliente Linux/Unix a monitorizar, primeramente descargaremos el paquete de instalación desde el propio servidor.

```
[root@cliente ~]# wget http://downloads.sourceforge.net/project/nagios/nrpe-2.x/nrpe-2.15/nrpe-2.15.tar.gz?r=http%3A%2F%2Fsourceforge.net%2Fprojects%2Fnagios%2Ffiles%2Fnrpe-2.x%2Fnrpe-2.15%2F&ts=1441226054&use_mirror=vorboss
[root@cliente ~]# tar -xzvf nrpe-2.15.tar.gz
```

Agregamos el puerto a los servicios:

```
[root@cliente ~]# vi /etc/services
nrpe 5666/tcp      #NRPE para Nagios
```

Habilitamos por NRPE al servidor de Nagios y al servidor local.

```
[root@cliente ~]# vi /etc/hosts.allow
nrpe: 192.168.1.5
nrpe: 127.0.0.1
```

Ahora habilitamos el servicio y damos permisos a todos los Plugins para su ejecución.

```
[root@cliente ~]# chkconfig nrpe on
[root@cliente ~]# service nrpe restart
[root@cliente ~]# chmod 777 /etc/nagios
[root@cliente ~]# chmod 777 /usr/lib64/nagios/Plugins
[root@cliente ~]# service nrpe restart
```

Para poder monitorizar chequeos por el protocolo NRPE, deberá realizar la instalación anterior comentada, tanto en el servidor de Nagios como en el servidor cliente Unix/Linux que se vaya a monitorizar con este protocolo.

Una vez instalación, se detallaran en los próximos puntos, como se realiza la definición de los mismos en Nagios para su monitorización completa, por medio de chequeos activos, lanzados desde el servidor de Nagios, y siendo enviada y recibida la respuesta por el puerto 5666 de cliente y servidor.

NOTA: Es necesario disponer de un usuario llamado "nagios" y tener permisos con él para reiniciar servicio NRPE en el servidor a monitorizar.

A continuación, se van a mostrar la definición de los diferentes chequeos según la instalación del agente que se haya montado en cada servidor.

Se han adjuntado durante el presente documento las pruebas del correcto funcionamiento tanto de scripts por NRPE como por NSCA. Los scripts que se han probado para comprobarlo, se visualizan en la interfaz web de Nagios de la siguiente manera (véase Ilustración 76).

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Acceso_Web	OK	09-08-2015 13:23:03	1d 19h 17m 49s	1/3	HTTP OK: HTTP/1.1 200 OK - 11694 bytes in 0,004 second response time
	PING	OK	09-08-2015 13:19:44	0d 1h 10m 5s	1/3	PING OK - Packet loss = 0%, RTA = 0.52 ms
	apache	?	09-08-2015 13:15:00	4d 2h 37m 44s	1/1	Servicio Apache OK (3)
	cpu	?	09-08-2015 13:15:04	3d 2h 59m 28s	1/1	CPU OK, 100% idle
	jboss	?	09-08-2015 13:15:00	1d 0h 45m 46s	1/1	Proceso Jboss OK
	memoria	?	09-08-2015 13:18:00	2d 4h 31m 1s	1/1	OK - 2944 MB (50%) Free Memory
	nntp	?	09-08-2015 13:18:00	8d 4h 10m 59s	1/1	Servicio NTP OK (1)
	ocupacion_discos	?	09-08-2015 13:05:00	1d 21h 16m 12s	1/1	DISK OK
	ssh	OK	09-08-2015 13:24:03	2d 16h 20m 34s	1/3	TCP OK - 0,000 second response time on 172.28.130.42 port 22

Ilustración 76.- Chequeos reales en servidor localhost

Los servicios que aparecen con un símbolo de “?”, son aquellos que se realizan en modo pasivo, ya que Nagios con dicho símbolo está informando que él está esperando a recibir información en los períodos de tiempo establecidos. Los otros tres ejemplos de la ilustración anterior (PING, Acceso\_Web y SSH), son chequeos activos, ya que en este caso se realizan solicitudes de estados a puertos de comunicación del servidor cliente.

### 8.1.7.2.3. Protocolo SNMP

El protocolo SNMP es usado para la monitorización de Routers, switches y dispositivos de red, con la finalidad de comprobar en ellos el correcto funcionamiento de sus componentes y conexiones.

El funcionamiento real de los chequeos (véase Ilustración 77) se produce mediante chequeos activos, donde Nagios invocando al chequeo (Check\_snmp) realizada la llamada usando este protocolo para comprobar su estado.

Este protocolo no necesita instalación del agente de Nagios, ya que la configuración de los dispositivos, debe ser realizada por los administradores de los mismos, para dar acceso a Nagios y de este modo poder ser monitorizados.

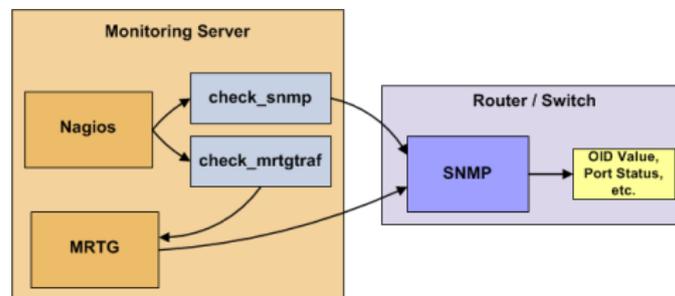


Ilustración 77.- Funcionamiento SNMP

### 8.1.8. Definición de grupos de contactos

Para poder definir los contactos a los que Nagios va a notificar las alertas de monitorización, es necesario tener definidos primero éstos para agregarlos a grupos de contactos, que serán lo que tengan el envío de las alertas asociados, aunque éstos estén compuestos por un único contacto.

Deberemos acceder al fichero contacts.cfg en la ruta /usr/local/nagios/etc, y veremos que por defecto se crea el contacto “nagiosadmin” [18].

```
[root@nagios ~]# vi /usr/local/nagios/etc/contacts.cfg
```

```
define contact{
    contact_name      nagiosadmin
    use                generic-contact
    alias             Nagios Admin
    email             admin.nagios@alexandra.com
}
```

Para poder definir otro, simplemente deberemos agregar una definición igual que la anterior con los datos que queramos.

```
define contact{
    contact_name      nagios
    use                generic-contact
    alias             Nagios consultor
    email             consultor.nagios@alexandra.com
}
```

Ahora deberemos crear los contact-groups que necesitemos para poder notificar las alertas.

El que observamos en el fichero contactgroups.cfg, en la ruta /usr/local/nagios/etc, es el grupo por defecto de notificación de alertas, que es responsabilidad del administrador de Nagios.

```
define contactgroup{
    contactgroup_name  admins
    alias              Nagios Administrators
    members            nagiosadmin
}
```

Agregaremos uno nuevo con el contacto nuevo creado.

```
define contactgroup{
    contactgroup_name  consultor
    alias              Nagios consultor
    members            nagios
}
```

Ahora solo faltará agregar dichos contactgroups a los servidores y servicios asociados, y se enviarán las notificaciones de alertas a ellos. Se explicará en el [Caso práctico de monitorización](#) explicado más adelante.

### 8.1.9. Envío y recepción de información

Para poder enviar y recibir los chequeos solicitados desde el servidor de Nagios a los servidores remotos, es necesario tener en consideración un esquema de red, para poder comprobar que todos los puertos necesarios están habilitados y es posible interactuar mediante ellos (véase Ilustración 78).

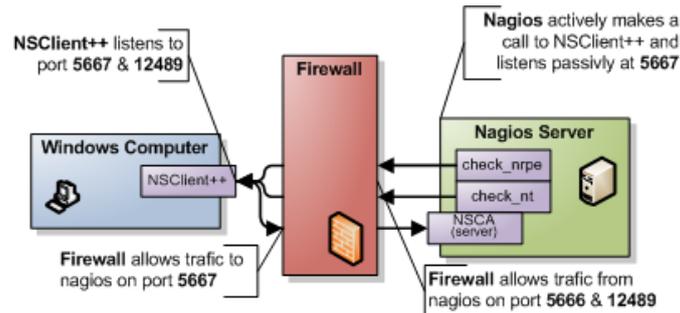


Ilustración 78.- Recepción y envío de datos mediante Firewall

Por último, para que Nagios pueda entender la información recibida por medio de los puertos comentados para el protocolo NSCA, es necesario mantener una estructura, tanto para el envío como para la recepción de documentación.

```
# Cadena de envío a Nagios:
# /home/nagios/agente_nagios/submit_check_result_via_nsca:
#
# Script de envío.
# `uname -n`: Nombre de la maquina,
# $1: Nombre chequeo, coincidente con Nombre del servicio en Nagios.
# $estado: Variable Global de Nagios.
# "$InfoGral": Variable GLOBAL que ofrece información del Resultado del chequeo.
```

```
/home/nagios/agente_nagios/submit_check_result_via_nsca `uname -n` $1 $estado $InfoGral
```

### 8.1.10. Caso práctico de monitorización

Nagios dispone de una organización a bajo de nivel con la posibilidad de definir grupos de servidores y de servicios.

A nivel de definición de grupos en Nagios existen: host-groups y service-groups.

Los llamados host-groups, nos servirán para definir aquellos grupos de máquinas que a nivel de servicio prestado, tengan alguna unión y/o similitud. Por ejemplo, en la mayoría de los Nagios instalados existe un host-group llamado Windows y otro llamado Unix. Podría ponerse cualquier nombre, pero es interesante poder tener esos grupos con la asociación real de servidores.

Los llamados service-groups, nos servirán para definir aquellos grupos de servicios que estén relacionados a nivel de servicio. Por ejemplo, en una de las aulas correspondientes al Departamento de Informática, que sean usadas para programación, se podrá realizar un service-group con nombre programación, para poder determinar todos los equipos que dispongan de servicios de java o tomcat, por ejemplo.

Con este tipo de organización, podremos disponer de diversas vistas de grupos, donde la página inicial de Nagios por defecto es de todos los host-groups definidos en Nagios, donde se podrán visionar

The screenshot shows two Nagios host group views. The first view is for 'Test 1 (-test1)' and shows a table with columns: Host, Status, Services, and Actions. The row for 'localhost1' shows a status of 'UP' (in a green box) and 'No matching services'. The second view is for 'Linux Servers (linux-servers)' and shows a table with columns: Host, Status, Services, and Actions. The row for 'localhost' shows a status of 'UP' (in a green box) and '3 OK' (in a green box).

Ilustración 79.- Vista HostGroups en Nagios

La definición de los host-groups, se realizará en el fichero llamado nagios.cfg.

```
[root@nagios ~]# vi /usr/local/nagios/etc/nagios.cfg
```

Donde definiremos los host-groups en único fichero, y cada uno de ellos en un directorio, si se desea para más comodidad.

```
cfg_file=/usr/local/nagios/etc/hostgroups.cfg
cfg_dir=/usr/local/nagios/etc/test1
cfg_dir=/usr/local/nagios/etc/linux-servers
cfg_file=/usr/local/nagios/etc/servicegroups.cfg
```

Dentro de cada uno de esos directorios, se deberá tener un fichero como mínimo donde se defina un host y un servicio asociado.

En hostgroups.cfg definiremos lo siguiente:

```
#####
#
# HOST GROUPS
#
#####

define hostgroup{
    hostgroup_name linux-servers
    alias          linux-servers
}

define hostgroup{
    hostgroup_name test1
    alias          linux-servers
}
```

En servicegroups.cfg definiremos lo siguiente:

```
#####  
#  
# SERVICE GROUPS  
#  
#####  
  
define servicegroup{  
    servicegroup_name linux-servers  
    alias linux-servers  
}  
  
define servicegroup{  
    servicegroup_name windows-servers  
    alias windows-servers  
}
```

NOTA: El hostgroup y el servicegroup no tienen por qué tener el mismo nombre.

Accedemos al directorio linux-servers, para poder generar los ficheros necesarios.

```
[root@nagios ~]# cd /usr/local/nagios/etc/linux-servers  
[root@nagios ~]# touch hosts_linux-servers.cfg  
[root@nagios ~]# touch services_linux-servers.cfg  
[root@nagios ~]# ls  
hosts_linux-servers.cfg  services_linux-servers.cfg
```

```
[root@nagios ~]# cd /usr/local/nagios/etc/test1  
[root@nagios ~]# touch hosts_test1.cfg  
[root@nagios ~]# touch services_test1.cfg  
[root@nagios ~]# ls  
hosts_test1.cfg  services_test1.cfg
```

Ya que se han creado dos hostgroups para realizar las pruebas, uno para servidores Windows y uno para servidores Linux/Unix, se van a mostrar una configuración de cada tipo de sistema operativo [10].

Comenzamos con el hostgroup “Linux-servers”, y procedemos a configurar el fichero de hosts (hosts\_linux-servers.cfg):

```
#####  
# HOSTS LINUX SERVERS  
#####  
  
define host{  
    name Linux-servers-host  
    use generic-host  
    hostgroups linux-servers  
    contact_groups Monitorizacion  
    register 0  
}  
  
#####  
  
define host{  
    use Linux-servers-host  
    host_name localhost  
    alias Servidor local  
    address 192.168.1.5  
}
```

Procedemos a configurar el fichero de services (services\_linux-servers.cfg), definiendo el chequeo activo de PING, disponible en la configuración estándar, y varios chequeos pasivos mencionados en el punto 9.1.6.2.1 del Protocolo NSCA, para explicar definición que se realiza sobre el servidor de Nagios.

```
#####  
# SERVICES - LINUX SERVERS  
#####  
  
# Definicion global para todos los servicios de este cliente.  
define service{  
    name                Linux-servers-service  
    use                 generic-host  
    notification_options w,u,C,r  
    servicegroups      linux-servers  
    contact_groups     Monitorizacion  
    register            0  
}  
  
define service{  
    name                Linux-servers-remote-service  
    use                 passive_checkservice  
    notification_options w,u,C,r  
    servicegroups      linux-servers  
    contact_groups     Monitorizacion  
    register            0  
}  
  
#####  
  
# Chequeo ACTIVO  
define service{  
    use                 Linux-servers-service  
    host_name           localhost  
    service_description PING  
    contact_groups     Monitorizacion  
    servicegroups      linux-servers  
    check_command       check_host_alive  
}  
  
# Chequeos pasivos NSCA  
define service{  
    use                 Linux-servers-remote-service  
    host_name           localhost  
    service_description memoria  
    contact_groups     Monitorizacion  
    servicegroups      linux-servers  
}  
  
define service{  
    use                 Linux-servers-remote-service  
    host_name           localhost  
    service_description CPU  
    contact_groups     Monitorizacion  
    servicegroups      linux-servers  
}  
  
define service{  
    use                 Linux-servers-remote-service  
    host_name           localhost  
    service_description discos  
    contact_groups     Monitorizacion  
    servicegroups      linux-servers  
}  
  
define service{  
    use                 Linux-servers-remote-service  
    host_name           localhost  
    service_description swap  
    contact_groups     Monitorizacion  
    servicegroups      linux-servers  
}
```

```
# Chequeos por NRPE
define service{
    use                Linux-servers-service
    host_name          localhost
    contact_groups     Monitorizacion
    service_description Discos
    servicegroups      linux-servers
    check_command      check_nrpe!check_disk
}

define service{
    use                Linux-servers-service
    host_name          localhost
    contact_groups     Monitorizacion
    service_description Memoria
    servicegroups      linux-servers
    check_command      check_nrpe!check_mem
}

define service{
    use                Linux-servers-service
    host_name          localhost
    contact_groups     Monitorizacion
    service_description Swap
    servicegroups      linux-servers
    check_command      check_nrpe!check_swap
}
```

Una vez que los chequeos estén definidos, procederemos a reiniciar el servicio de Nagios:

```
[root@nagios ~]# service nagios restart
```

Y después de reiniciar, nos aparecerán los chequeos de la siguiente manera en la interfaz web de Nagios:

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	cpu	PENDING	N/A	0d 0h 3m 23s+	1/3	Service check scheduled for Wed Aug 17 17:26:40 CEST
	discos	PENDING	N/A	0d 0h 3m 23s+	1/3	Service check scheduled for Wed Aug 17 17:25:11 CEST
	memoria	PENDING	N/A	0d 0h 3m 23s+	1/3	Service check scheduled for Wed Aug 17 17:26:40 CEST
	PING	OK	08-17-2011 17:20:12	0d 1h 25m 0s	1/4	ECG OK - Paquetes perdidos = 0%, RTA = 0.05 ms
	swap	PENDING	N/A	0d 0h 3m 23s+	1/3	Service check scheduled for Wed Aug 17 17:25:11 CEST

Ilustración 80.- Agregar chequeos a Nagios de NSCA

Una vez que el servidor a monitorizar envía a Nagios toda la información, según la programación establecida en el “cron”, se verán actualizados todos los datos en Nagios de la siguiente manera:

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	PING	OK	08-27-2015 12:53:37	28d 2h 42m 53s	1/3	PING OK - Packet loss = 0%, RTA = 0.07 ms
	cpu	OK	08-27-2015 12:52:17	0d 0h 3m 47s	1/1	CPU OK, 95% idle
	memoria	OK	08-27-2015 11:30:03	371d 19h 18m 57s	1/1	OK - 1733 MB (85%) Free Memory
	discos	OK	08-27-2015 12:35:05	183d 18h 57m 44s	1/1	DISK OK
	swap	OK	08-27-2015 12:49:01	454d 2h 15m 1s	1/1	OK - 1983 MB (100%) Free Memory Swap

Ilustración 81.- Chequeos pasivos NSCA recibidos por Nagios

Si monitorizásemos el servidor “localhost” con NRPE también, podríamos ver la siguiente información (véase Ilustración 82):

Host **	Service **	Status **	Last Check **	Duration **	Attempt **	Status Information
localhost	PING	OK	08-27-2015 12:53:37	28d 2h 42m 53s	1/3	PING OK - Packet loss = 0%, RTA = 0.07 ms
	cpu	?	08-27-2015 12:52:17	0d 0h 3m 47s	1/1	CPU OK, 95% idle
	memoria	?	08-27-2015 11:30:03	371d 19h 18m 57s	1/1	OK - 1733 MB (85%) Free Memory
	discos	?	08-27-2015 12:35:05	183d 18h 57m 44s	1/1	DISK OK
	swap	?	08-27-2015 12:49:01	454d 2h 15m 1s	1/1	OK - 1983 MB (100%) Free Memory Swap
	CPU	OK	08-27-2015 12:49:31	454d 2h 15m 1s	1/1	CPU OK, 95% idle
	Memoria	OK	08-27-2015 12:49:27	454d 2h 15m 1s	1/1	OK - 1733 MB (85%) Free Memory
	Discos	OK	08-27-2015 12:49:14	454d 2h 15m 1s	1/1	DISK OK
	Swap	OK	08-27-2015 12:49:08	454d 2h 15m 1s	1/1	OK - 1983 MB (100%) Free Memory Swap

Ilustración 82.- Chequeos NSCA y NRPE recibidos por Nagios

Ahora, se van a definir los chequeos activos para el hostgroups llamado “test1” donde existen un servidor con sistema operativo Windows, llamado localhost1. Dentro de la ruta /usr/local/nagios/etc/test1, definiremos el fichero hosts\_test1.cfg:

```
#####
# HOSTS TEST1
#####

define host{
    name                test1-host
    use                  generic-host
    hostgroups           test1
    contact_groups      Monitorizacion
    register             0
}

#####

define host{
    use                  test1-host
    host_name            localhost1
    alias                Servidor localhost1 de test1
    address              192.168.1.10
}

```

A continuación, para definir servicios sobre el servidor localhost1, tendremos que definir los servicios dentro del servidor services\_test1.cfg.

```
#####
# SERVICES TEST1
#####

# Definicion global para todos los servicios de este cliente.
define service{
    name                test1-service
    use                  generic-service
    notification_options w,u,c,r
    servicegroups       windows-servers
    contact_groups      Monitorizacion
    register            0
}

#####

define service{
    use                  test1-service
    host_name            localhost1
    service_description PING
    contact_groups      Monitorizacion
}

```

```

        servicegroups      windows-servers
        check_command      check_host_alive
    }

define service{
    use                    test1-service
    host_name              localhost1
    service_description    Disco_C
    contact_groups        Monitorizacion
    servicegroups          windows-servers
    check_command          check_nt_disk!C!90!95
}

define service{
    use                    test1-service
    host_name              localhost1
    service_description    CPU
    contact_groups        Monitorizacion
    servicegroups          windows-servers
    check_command          check_nt_cpuload!5,80,90!
}

define service{
    use                    test1-service
    host_name              localhost1
    service_description    Memoria
    contact_groups        Monitorizacion
    servicegroups          windows-servers
    check_command          check_nt_memuse!80!90
}
    
```

Una vez modificados dichos ficheros, procederemos a reiniciar el servicio de Nagios. En el momento en el que Nagios solicita al servidor la información, ésta es actualizada automáticamente en la interfaz web [12].

Host **	Service **	Status **	Last Check **	Duration **	Attempt **	Status Information
localhost1	CPU	OK	08-27-2015 12:46:38	56d 19h 57m 32s	1/3	CPU Load 0% (5 min average)
	Disco_C	OK	08-27-2015 12:46:16	0d 12h 42m 33s	1/3	C: - total: 8,01 Gb - used: 5,64 Gb (70%) - free 2,36 Gb (30%)
	Memoria	OK	08-27-2015 12:46:57	321d 21h 53m 22s	1/3	Memory usage: total:3945,52 MB - used: 490,84 MB (12%) - free: 3454,69 MB (88%)
	PING	OK	08-27-2015 12:47:33	22d 0h 44m 23s	1/3	PING OK - Packet loss = 0%, RTA = 0.50 ms

**Ilustración 83.- Chequeos activos recibidos por Nagios**

## 8.2. Manual de Usuario

Para poder acceder a Nagios, el usuario deberá poner `http://nagios/nagios`, y con el usuario de acceso que se le haya definido, tendrá el perfil generado.

La vista principal a la que accederá, es la misma que hemos comentado anteriormente, que corresponde a la del menú en la parte izquierda de la interfaz.

Lo primero que aparecerá será una pantalla de login, donde pondremos el usuario y la contraseña facilitada para acceder.

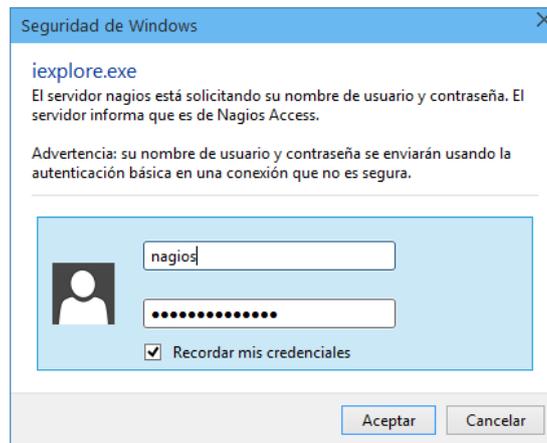


Ilustración 84.- Login en interfaz web de Nagios

A continuación, se accederá a la pantalla inicial de Nagios (véase Ilustración 85).



Ilustración 85.- Pantalla inicial de interfaz web de Nagios

El usuario podrá acceder a todo el menú izquierdo, con opciones de acceso y/o modificación, determinados en el fichero cgi.cfg en cada uno de los tipos de perfil. A continuación, se van a mostrar todos los accesos que aparecen en dicho menú, para mostrar la información que se visualiza en cada caso.

**Nagios®** Tactical Monitoring Overview

Updated every 90 seconds  
Nagios® Core™ 4.0.11 - www.nagios.org  
Logged in as nagiosadmin

**Monitoring Performance**

Service Check Execution Time:	0.00 / 4.00 / 0.505 sec
Service Check Latency:	0.00 / 0.00 / 0.000 sec
Host Check Execution Time:	4.00 / 4.00 / 4.003 sec
Host Check Latency:	0.00 / 0.00 / 0.000 sec
# Active Host / Service Checks:	1 / 8
# Passive Host / Service Checks:	0 / 0

**Network Health**

Host Health: ██████████

Service Health: ██████████

**Network Outages**

0 Outages

**Hosts**

0 Down    0 Unreachable    1 Up    0 Pending

**Services**

0 Critical    1 Warning    0 Unknown    7 Ok    0 Pending

1 Unhandled Problems

**Monitoring Features**

Flap Detection	Notifications	Event Handlers	Active Checks	Passive Checks
✓ All Services Enabled	✓ 2 Services Disabled	✓ All Services Enabled	✓ All Services Enabled	✓ All Services Enabled
No Services Flapping	All Hosts Enabled	All Hosts Enabled	All Hosts Enabled	All Hosts Enabled
All Hosts Enabled				
No Hosts Flapping				

Ilustración 86.- Acceso a Tactical Overview

La siguiente ilustración nos muestra la vista de los hosts monitorizados en Nagios. (Véase Ilustración 87)

**Nagios®** Current Network Status

Updated every 90 seconds  
Nagios® Core™ 4.0.11 - www.nagios.org  
Logged in as nagiosadmin

**Host Status Totals**

Up	Down	Unreachable	Pending
1	0	0	0

**Service Status Totals**

Ok	Warning	Unknown	Critical	Pending
7	1	0	0	0

**Host Status Details For All Host Groups**

Limit Results: 100

Host	Status	Last Check	Duration	Status Information
localhost	UP	10-17-2013 03:43:39	2d 17h 23m 40s	PING OK - Packet loss = 0%, RTA = 0.02 ms

Results 1 - 1 of 1 Matching Hosts

Ilustración 87.- Vista de Host

En la siguiente vista, se podrán visualizar los servicios monitorizados, para comprobar el estado de los mismos. (Véase Ilustración 88).

The screenshot shows the Nagios web interface. On the left is a navigation menu with 'Services' highlighted. The main content area is titled 'Service Status Details For All Hosts'. It includes summary statistics for Host Status Totals and Service Status Totals. Below these is a table of service details for the 'localhost' host.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
localhost	Current Load	OK	10-17-2013 03:43:39	2d 17h 22m 46s	1/4	OK - load average: 0.10, 0.16, 0.10
	Current Users	OK	10-17-2013 03:43:39	2d 17h 22m 3s	1/4	USERS OK - 2 users currently logged in
	HTTP	WARNING	10-17-2013 03:43:39	2d 17h 26m 22s	4/4	HTTP WARNING: HTTP/1.1 403 Forbidden - 5237 bytes in 0.001 second response time
	PING	OK	10-17-2013 03:43:39	2d 17h 26m 5s	1/4	PING OK - Packet loss = 0%, RTA = 0.02 ms
	Root Partition	OK	10-17-2013 03:43:39	2d 17h 25m 35s	1/4	DISK OK - free space: / 8565 MB (89% inode+95%):
	SSH	OK	10-17-2013 03:43:39	2d 17h 24m 57s	1/4	SSH OK - OpenSSH_5.3 (protocol 2.0)
	Swap Usage	OK	10-17-2013 03:44:15	2d 17h 24m 20s	1/4	SWAP OK - 98% free (248 MB out of 256 MB)
	Total Processes	OK	10-17-2013 03:44:31	2d 17h 23m 42s	1/4	PROCS OK: 14 processes with STATE = RSZDT

Ilustración 88.- Vista de Services

En la siguiente vista, se podrán visualizar los parámetros y datos registrados en un chequeo de un servidor. (Véase Ilustración 89).

The screenshot shows the 'Nagios Process Information' page. It displays the current status of the Nagios process and a list of process commands that can be executed. The 'Process Information' section lists various parameters and their values.

Parameter	Value
Program Version:	4.0.1
Program Start Time:	10-17-2014 03:42:32
Total Running Time:	0d 0h 8m 20s
Last Log File Rotation:	N/A
Nagios PID	32031
Notifications Enabled?	YES
Service Checks Being Executed?	YES
Passive Service Checks Being Accepted?	YES
Host Checks Being Executed?	YES
Passive Host Checks Being Accepted?	YES
Event Handlers Enabled?	Yes
Obsessing Over Services?	No
Obsessing Over Hosts?	No
Flap Detection Enabled?	Yes
Performance Data Being Processed?	No

The 'Process Commands' section includes the following actions:

- Shutdown the Nagios process
- Restart the Nagios process
- Disable notifications
- Stop executing service checks
- Stop accepting passive service checks
- Stop executing host checks
- Stop accepting passive host checks
- Disable event handlers
- Start obsessing over services
- Start obsessing over hosts
- Disable flap detection
- Enable performance data

Ilustración 89.- Información accesible por chequeo

Para poder comprobar en todo momento el número de chequeos que están en alerta, es recomendable usar el campo de “All Problems” (véase Ilustración 90).



Ilustración 90.- Resumen estados en Nagios

Al pulsar en “All Problems” podremos disponer de una vista rápida, como la siguiente, que ordenándola por “Duration” ascendente, siempre nos informará de las últimas novedades de alertas.

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Router_2	PING	WARNING	09-07-2015 13:12:36	0d 0h 0m 17s	1/10	PING WARNING - Packet loss = 0%, RTA = 141.94 ms
Router_1	PING	WARNING	09-07-2015 13:11:48	0d 0h 1m 5s	1/10	PING WARNING - Packet loss = 0%, RTA = 61.99 ms
localhost3	JBoss	CRITICAL	09-07-2015 13:05:38	0d 0h 7m 15s	1/3	CRITICAL - Socket timeout after 30 seconds
	Disco_E	CRITICAL	09-07-2015 13:05:28	0d 0h 7m 25s	1/3	CRITICAL - Socket timeout after 30 seconds
	Memoria	CRITICAL	09-07-2015 13:05:27	0d 0h 7m 26s	1/3	CRITICAL - Socket timeout after 30 seconds
	MySQL	CRITICAL	09-07-2015 13:05:22	0d 0h 7m 31s	1/3	CRITICAL - Socket timeout after 30 seconds
	PING	CRITICAL	09-07-2015 13:10:32	0d 0h 8m 20s	3/3	PING CRITICAL - Packet loss = 100%
Switch	Disco_C	CRITICAL	09-07-2015 13:05:02	0d 0h 8m 51s	2/3	CRITICAL - Socket timeout after 30 seconds
	PING	CRITICAL	09-07-2015 13:10:57	0d 0h 45m 54s	10/10	PING CRITICAL - Packet loss = 100%

Ilustración 91.- Alertas en un preciso instante

Los usuarios con ciertos privilegios de modificación, podrán acceder a la información de servicios y servidores avanzada, así como agregar comentarios al servidor o servicios de estados.

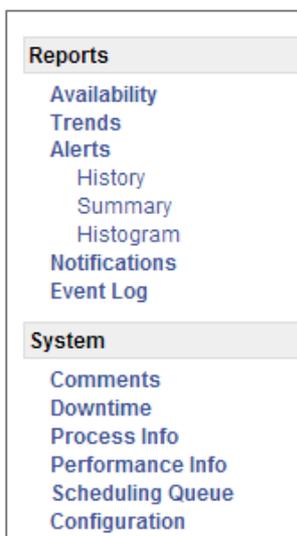


Ilustración 92.- Menú administración Nagios

En el menú que observamos a la izquierda de la pantalla, además de las vistas de hostgroups, servicegroups, hosts y services, dispondremos por defecto de la vista de “Reports” y “System”, para el administrador, los usuarios no deben tener acceso a “System”, ya que mucha configuración de Nagios puede ser manipulada desde este submenú.

Desde la vista de “Reports” podremos acceder a las gráficas de disponibilidad, donde cada una de ellas nos reportará un tipo de informe.

En el caso de “Availability” nos mostrará una gráfica del estilo de la siguiente imagen. (Véase Ilustración 91).

**Service State Breakdowns:**

Host	Service	% Time OK	% Time Warning	% Time Unknown	% Time Critical	% Time Undetermined
	DNS	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	FTP	98.927% (98.927%)	0.000% (0.000%)	0.000% (0.000%)	1.073% (1.073%)	0.000%
	HTTP	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	IMAP	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	PING	96.876% (96.876%)	3.124% (3.124%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	SMTP	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	/ Disk Usage	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	/dev1/html	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	/boot Disk Usage	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	/dev/sda S.M.A.R.T.	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	/home Disk Usage	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	/store Disk Usage	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	/tmp Disk Usage	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	Backups: Home Dirs	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	Backups: Mondo Rescue	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
	Backups: MySQL	0.000% (0.000%)	0.000% (0.000%)	0.000% (0.000%)	100.000% (100.000%)	0.000%

Ilustración 93.- Availability servidores

Como se puede observar en la ilustración anterior, nos muestra rendimientos y porcentajes de disponibilidad de los servicios monitorizados en los diferentes servidores.

Si clicamos en “Trends”, nos mostrará una serie de desplegables para que elijamos el servidor y servicio del que queremos ver una disponibilidad, mostrando una gráfica del estilo de la siguiente imagen. (Véase Ilustración 92).

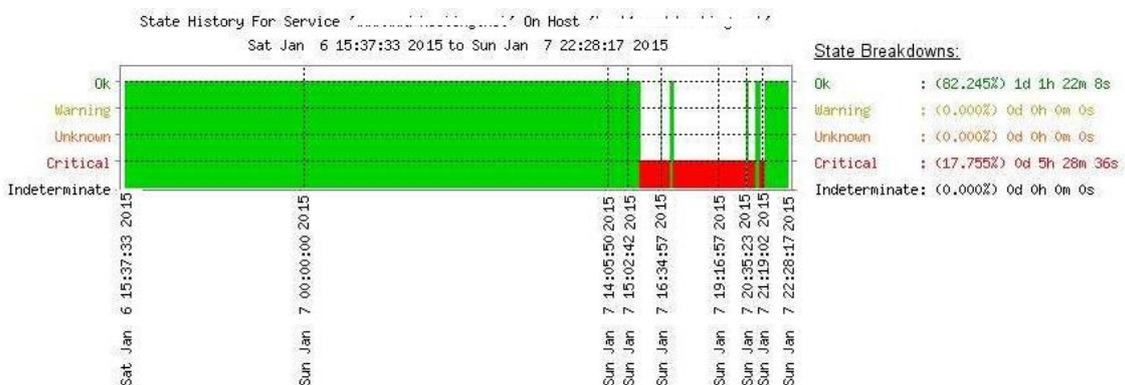


Ilustración 94.- Trend de servicio en Nagios

En la opción de “Notifications” se visualiza una lista de las últimas notificaciones mostradas en la interfaz web, y refleja las que han sido avisadas a los contactos dados de alta en Nagios.

Y por último, mostrará en “Event Log” toda la información que Nagios ha solicitado y recibido de los diferentes servidores monitorizados, que será el log que se almacenará completo en la máquina en /usr/local/nagios/var/nagios.log.

En la vista de “System”, se podrán agregar y modificar comentarios en los chequeos existentes, y se podrá parar el agente de Nagios y todos sus procesos en el servidor de monitorización.

### 8.3. Scripts generados

A continuación, se van a adjuntar una serie de scripts y chequeos que se han desarrollado para la monitorización de elementos que no están definidos en la configuración por defecto de Nagios. Todos ellos están disponibles para su descarga y uso en la web de Nagios, donde todos los desarrolladores compartimos todas las aportaciones posibles [7].

Estos scripts han sido realizados en Shell, Perl, y PowerShell, para poder ser ejecutados en servidor Unix y Windows.

#### 8.3.1. Script para chequear espacio libre en Unix

```
#####  
#  
#!/bin/ksh  
#  
# 02/06/2015 A. Moreno Creación Script.  
#  
#####  
#  
Warning=$2  
Critical=$3  
#  
InfoGral="/home/nagios/agente_nagios/Scripts/check_disk $Warning $Critical"  
echo "$InfoGral" | awk '{print $2}' | while read state  
do  
    case $state in  
        "CRITICAL")  
            estado=2  
            /home/nagios/agente_nagios/submit_check_result_via_nsca "Nagios" $1 $estado "$InfoGral"  
            ;;  
        "WARNING")  
            estado=1  
            /home/nagios/agente_nagios/submit_check_result_via_nsca "Nagios" $1 $estado "$InfoGral"  
            ;;  
        "OK")  
            estado=0  
            /home/nagios/agente_nagios/submit_check_result_via_nsca "Nagios" $1 $estado "$InfoGral"  
            ;;  
        esac  
    done  
#  
#####  
;;
```

Tabla 103.- Script para chequear espacio libre en Unix

#### 8.3.2. Script para chequear memoria libre en Unix

```
# Script to check real memory usage  
#  
# ##### Script Modifications #####  
# -----  
#!/bin/bash  
calc=/tmp/memcalc  
percent_free=/tmp/mempercent  
critical=""  
warning=""  
STATE_OK=0  
STATE_WARNING=1  
STATE_CRITICAL=2  
STATE_UNKNOWN=3  
# Total memory available  
total=`free -m | head -2 | tail -1 | gawk '{print $2}'`  
# Total memory used  
used=`free -m | head -3 | tail -1 | gawk '{print $3}'`  
# Calc total minus used  
free=`free -m | head -3 | tail -1 | gawk '{print $4}'`  
echo "5" > $calc # decimal accuracy
```

```
echo "k" >> $calc # commit
echo "100" >> $calc # multiply
echo "$free" >> $calc # division integer
echo "$total" >> $calc # division integer
echo "/" >> $calc # division sign
echo "*" >> $calc # multiplication sign
echo "p" >> $calc # print
percent=`/usr/bin/dc $calc|/bin/sed 's/^\.0./'|usr/bin/tr "." " "|usr/bin/gawk {'print $1'}`

if [[ "$percent" -le $critical ]]
then
    echo "CRITICAL - $free MB ($percent%) Free Memory"
    exit 2
fi
if [[ "$percent" -le $warning ]]
then
    echo "WARNING - $free MB ($percent%) Free Memory"
    exit 1
fi
if [[ "$percent" -gt $warning ]]
then
    echo "OK - $free MB ($percent%) Free Memory"
    exit 0
fi
```

Tabla 104.- Script para chequear memoria libre en Unix

### 8.3.3. Script para chequear servicio en Unix

```
conexiones_nsca)
#
#####
#####
#
conex=`ps -u nagios -o pid|wc -l`
conex_multi=`netstat -an | grep nagios`

if [ $conex -ge 60 ]
then
    estado=2
    /home/nagios/agente_nagios/submit_check_result_via_nsca "Nagios" $1 $estado "$InfoGral"
else
    estado=0
    /home/nagios/agente_nagios/submit_check_result_via_nsca "Nagios" $1 $estado "$InfoGral"
fi

#
#####
#####
#
;;
```

Tabla 105.- Script para chequear servicio en Unix

### 8.3.4. Script para chequear las bases de datos de Exchange

Este script está en el servidor de correo de Exchange, y es llamado por el agente de Nagios como chequeo activo desde el servidor de Nagios, para su ejecución, programado en el lenguaje PowerShell.

```
# Test Mailbox Database Health (Exchange 2007)
#
# To execute from within NSClient++
#
#[NRPE Handlers]
#check_mailbox_health=cmd /c echo C:\Agente_Nagios\Mbhealth2007.ps1 | PowerShell.exe -Command -
#
Add-PSSnapin Microsoft.Exchange.Management.PowerShell.Admin
$NagiosStatus = "0"
$NagiosDescription = ""

Get-MailboxDatabase -Status | ForEach-Object {
    if ($_.Mounted -ne "True") {
        $NagiosStatus = "2"
        if ($NagiosDescription -ne "") {
```

```
        $NagiosDescription = $NagiosDescription + ", "
    }
    $NagiosDescription = $NagiosDescription + $_.Name + ":Dismounted"
}
else
{
    $I=0
}
}
if ($NagiosStatus -eq "2") {
    Write-Host "CRITICAL: " $NagiosDescription
} else {
    Write-Host "OK: All Mailbox Databases are mounted."
}
exit $NagiosStatus
```

Tabla 106.- Script para chequear la base de datos Exchange

### 8.3.5. Script para chequear el estado de un Ironport

Chequeo que se ejecuta en el servidor de Nagios, lanzando peticiones al servidor de Ironport para comprobar su estado por medio del report en XML que genera.

```
#!/bin/sh
# Plugin to check Ironport Appliances
# input parameters
HOSTNAME=$1
USER=$2
PASSWORD=$3
ARGS=$4
PAR_WARN=$5
PAR_CRIT=$6

STATUSFILE=/tmp/status.`echo $RANDOM$RANDOM|cut -c1-4`

# Nagios return codes
STATE_OK=0
STATE_WARNING=1
STATE_CRITICAL=2
STATE_UNKNOWN=3
STATE_DEPENDENT=4

PROGNAME=`basename $0`

wget --http-user=$USER --http-password=$PASSWORD --no-check-certificate --no-proxy https://$HOSTNAME/xml/status --output-document=$STATUSFILE >> /dev/null 2>&1

case "$ARGS" in
status)
    IPORTPAR=`grep "system status" $STATUSFILE | cut -d\" -f 2`
    if [ $IPOINTPAR = online ]; then
        exitstatus=${STATE_OK}
        echo "STATUS OK: $IPOINTPAR" "| RESULT=1;"
    else
        exitstatus=${STATE_CRITICAL}
        echo "STATUS CRITICAL: $IPOINTPAR" "| RESULT=0;"
    fi
    ;;
cpu)
    IPORTPAR=`grep total_utilization $STATUSFILE | cut -d\" -f 4`
    if [ $IPOINTPAR -ge $PAR_WARN ]; then
        if [ $IPOINTPAR -ge $PAR_CRIT ]; then
            exitstatus=${STATE_CRITICAL}
            echo "CPU CRITICAL: $IPOINTPAR%" "| RESULT=$IPOINTPAR%;$PAR_WARN;$PAR_CRIT;"
        else
            exitstatus=${STATE_WARNING}
            echo "CPU WARNING: $IPOINTPAR%" "| RESULT=$IPOINTPAR%;$PAR_WARN;$PAR_CRIT;"
        fi
    else
        exitstatus=${STATE_OK}
        echo "CPU OK: $IPOINTPAR%" "| RESULT=$IPOINTPAR%;$PAR_WARN;$PAR_CRIT;"
    fi
    ;;
ram)
    IPORTPAR=`grep ram_utilization $STATUSFILE | cut -d\" -f 4`
    if [ $IPOINTPAR -ge $PAR_WARN ]; then
        if [ $IPOINTPAR -ge $PAR_CRIT ]; then
            exitstatus=${STATE_CRITICAL}
            echo "RAM CRITICAL: $IPOINTPAR%" "| RESULT=$IPOINTPAR%;$PAR_WARN;$PAR_CRIT;"
        else
            exitstatus=${STATE_WARNING}
            echo "RAM WARNING: $IPOINTPAR%" "| RESULT=$IPOINTPAR%;$PAR_WARN;$PAR_CRIT;"
        fi
    fi
    ;;
```

Trabajo Fin de Grado  
Alexandra Moreno Calabozo

```

    else
        fi
        exitstatus=${STATE_OK}
        echo "RAM OK: $IPORTPAR%" "| RESULT=$IPORTPAR%;$PAR_WARN;$PAR_CRIT;"
    fi
    ;;
msgxhour)
    IPORTPAR=`grep -A 3 "rate name=\"inj_msgs\" $STATUSFILE | grep last_15_min | cut -d\" -f 2 `
    if [ $IPORTPAR -ge $PAR_WARN ]; then
        if [ $IPORTPAR -ge $PAR_CRIT ]; then
            exitstatus=${STATE_CRITICAL}
            echo "MAIL RATE CRITICAL: $IPORTPAR msgs/hr" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        else
            exitstatus=${STATE_WARNING}
            echo "MAIL RATE WARNING: $IPORTPAR msgs/hr" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        fi
    else
        exitstatus=${STATE_OK}
        echo "MAIL RATE OK: $IPORTPAR msgs/hr" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
    fi
    ;;
conn_in)
    IPORTPAR=`grep conn_in $STATUSFILE | cut -d\" -f 4 `
    if [ $IPORTPAR -ge $PAR_WARN ]; then
        if [ $IPORTPAR -ge $PAR_CRIT ]; then
            exitstatus=${STATE_CRITICAL}
            echo "Incomming Connections CRITICAL: $IPORTPAR connections" "|
RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        else
            exitstatus=${STATE_WARNING}
            echo "Incomming Connections WARNING: $IPORTPAR connections" "|
RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        fi
    else
        exitstatus=${STATE_OK}
        echo "Incomming Connections OK: $IPORTPAR connections" "|
RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
    fi
    ;;
conn_out)
    IPORTPAR=`grep conn_out $STATUSFILE | cut -d\" -f 4 `
    if [ $IPORTPAR -ge $PAR_WARN ]; then
        if [ $IPORTPAR -ge $PAR_CRIT ]; then
            exitstatus=${STATE_CRITICAL}
            echo "Outgoing Connections CRITICAL: $IPORTPAR connections" "|
RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        else
            exitstatus=${STATE_WARNING}
            echo "Outgoing Connections WARNING: $IPORTPAR connections" "|
RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        fi
    else
        exitstatus=${STATE_OK}
        echo "Outgoing Connections OK: $IPORTPAR connections" "|
RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
    fi
    ;;
queue)
    IPORTPAR1=`grep \"attempted_recips $STATUSFILE | cut -d\" -f 4 `
    IPORTPAR2=`grep unattempted_recips $STATUSFILE | cut -d\" -f 4 `
    IPORTPAR=`expr $IPORTPAR1 + $IPORTPAR2`
    if [ $IPORTPAR -ge $PAR_WARN ]; then
        if [ $IPORTPAR -ge $PAR_CRIT ]; then
            exitstatus=${STATE_CRITICAL}
            echo "QUEUE CRITICAL: $IPORTPAR msgs" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        else
            exitstatus=${STATE_WARNING}
            echo "QUEUE WARNING: $IPORTPAR msgs" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        fi
    else
        exitstatus=${STATE_OK}
        echo "QUEUE OK: $IPORTPAR msgs" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
    fi
    ;;
workqueue)
    IPORTPAR=`grep msgs_in_work_queue $STATUSFILE | cut -d\" -f 4 `
    if [ $IPORTPAR -ge $PAR_WARN ]; then
        if [ $IPORTPAR -ge $PAR_CRIT ]; then
            exitstatus=${STATE_CRITICAL}
            echo "WORKQUEUE CRITICAL: $IPORTPAR msgs" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        else
            exitstatus=${STATE_WARNING}
            echo "WORKQUEUE WARNING: $IPORTPAR msgs" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        fi
    else
        exitstatus=${STATE_OK}
        echo "WORKQUEUE OK: $IPORTPAR msgs" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
    fi
    ;;
resourceconservation)
    IPORTPAR=`grep resource_conservation $STATUSFILE | cut -d\" -f 4 `
    if [ $IPORTPAR -ge $PAR_WARN ]; then
        if [ $IPORTPAR -ge $PAR_CRIT ]; then
            exitstatus=${STATE_CRITICAL}
            echo "RESOURCECONSERVATION CRITICAL: $IPORTPAR" "| RESULT=$IPORTPAR;"
        else
            exitstatus=${STATE_WARNING}
            echo "RESOURCECONSERVATION WARNING: $IPORTPAR" "| RESULT=$IPORTPAR;"
        fi
    else
        exitstatus=${STATE_OK}
    fi
    ;;
else
    fi
    ;;

```

Trabajo Fin de Grado  
Alexandra Moreno Calabozo

```
        exitstatus=${STATE_OK}
        echo "RESOURCECONSERVATION OK: $IPORTPAR " "| RESULT=$IPORTPAR;"
    fi
    ;;
disk_util)
    IPORTPAR=`grep disk_utilization $STATUSFILE | cut -d\" -f 4`
    if [ $IPORTPAR -ge $PAR_WARN ]; then
        if [ $IPORTPAR -ge $PAR_CRIT ]; then
            exitstatus=${STATE_CRITICAL}
            echo "Disk Utilization CRITICAL: $IPORTPAR " "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        else
            exitstatus=${STATE_WARNING}
            echo "Disk Utilization WARNING: $IPORTPAR " "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        fi
    else
        exitstatus=${STATE_OK}
        echo "Disk Utilization OK: $IPORTPAR" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
    fi
    ;;
msgs_in_quarantine)
    IPORTPAR=`grep msgs_in_quarantine $STATUSFILE | cut -d\" -f 4`
    if [ $IPORTPAR -ge $PAR_WARN ]; then
        if [ $IPORTPAR -ge $PAR_CRIT ]; then
            exitstatus=${STATE_CRITICAL}
            echo "MSG In Quarantine CRITICAL: $IPORTPAR" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        else
            exitstatus=${STATE_WARNING}
            echo "MSG In Quarantine WARNING: $IPORTPAR" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        fi
    else
        exitstatus=${STATE_OK}
        echo "MSG In Quarantine OK: $IPORTPAR" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
    fi
    ;;
quarantine_utilization)
    IPORTPAR=`grep quarantine_utilization $STATUSFILE | cut -d\" -f 4`
    if [ $IPORTPAR -ge $PAR_WARN ]; then
        if [ $IPORTPAR -ge $PAR_CRIT ]; then
            exitstatus=${STATE_CRITICAL}
            echo "MSG In Quarantine CRITICAL: $IPORTPAR" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        else
            exitstatus=${STATE_WARNING}
            echo "MSG In Quarantine WARNING: $IPORTPAR" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
        fi
    else
        exitstatus=${STATE_OK}
        echo "MSG In Quarantine OK: $IPORTPAR" "| RESULT=$IPORTPAR;$PAR_WARN;$PAR_CRIT;"
    fi
    ;;
queuedisk_usage)
    IPORTPAR1=`grep kbytes_free $STATUSFILE | cut -d\" -f 4`
    IPORTPAR2=`grep kbytes_used $STATUSFILE | cut -d\" -f 4`
    IPORTPAR3=$(( $IPORTPAR1 / 100 ))
    IPORTPAR=$(( $IPORTPAR2 / $IPORTPAR3 ))
    if [ $IPORTPAR -ge $PAR_WARN ]; then
        if [ $IPORTPAR -ge $PAR_CRIT ]; then
            exitstatus=${STATE_CRITICAL}
            echo "QueueDisk Usage CRITICAL: $IPORTPAR%" "| RESULT=$IPORTPAR%;$PAR_WARN;$PAR_CRIT;"
        else
            exitstatus=${STATE_WARNING}
            echo "QueueDisk Usage WARNING: $IPORTPAR%" "| RESULT=$IPORTPAR%;$PAR_WARN;$PAR_CRIT;"
        fi
    else
        exitstatus=${STATE_OK}
        echo "QueueDisk Usage OK: $IPORTPAR%" "| RESULT=$IPORTPAR%;$PAR_WARN;$PAR_CRIT;"
    fi
    ;;
*)
    print_usage
    exit $STATE_OK
esac

rm -rf $STATUSFILE

exit $exitstatus
```

Tabla 107.- Script para chequear estado de un Ironport

### 8.3.6. Script para chequear servicio Squid en Proxy

Ejecución de chequeo en modo activo desde el servidor de Nagios, solicitando al proxy la información de estado de su servicio de Squid.

```
#!/usr/bin/perl -w
#
# check_squid - Nagios check plugin for testing a Squid proxy

require 5.004;
use POSIX;
use strict;
use Getopt::Long;
use vars qw($opt_V $opt_h $opt_t $opt_u $opt_n $opt_s
            $opt_p $opt_l $opt_o $opt_m $opt_e);

use vars qw($PROGRAMNAME);
use lib "/usr/lib/nagios/plugins";
use utils qw($TIMEOUT $ERRORS &print_revision &support &usage);
use LWP::UserAgent;
use HTTP::Request::Common qw(POST GET);
use HTTP::Headers;

my ($url, $urluser, $urlpass, $proxy, $proxyport,
    $proxyuser, $proxypass, $expectstatus, $res, $req);

$PROGRAMNAME = "check_squid_lw.pl";

Getopt::Long::Configure('bundling');
GetOptions("v" => \$opt_V, "version" => \$opt_V,
    "h" => \$opt_h, "help" => \$opt_h,
    "t=s" => \$opt_t, "timeout=i" => \$opt_t,
    "u=s" => \$opt_u, "url=s" => \$opt_u,
    "n=s" => \$opt_n, "urluser=s" => \$opt_n,
    "s=s" => \$opt_s, "urlpass=s" => \$opt_s,
    "p=s" => \$opt_p, "proxy=s" => \$opt_p,
    "l=s" => \$opt_l, "proxyport=s" => \$opt_l,
    "o=s" => \$opt_o, "proxyuser=s" => \$opt_o,
    "m=s" => \$opt_m, "proxypass=s" => \$opt_m,
    "e=i" => \$opt_e, "status=i" => \$opt_e);

if(defined($opt_n)) { $urluser = $opt_n; }
if(defined($opt_s)) { $urlpass = $opt_s; }
if(defined($opt_o)) { $proxyuser = $opt_o; }
if(defined($opt_m)) { $proxypass = $opt_m; }

my $ua = new LWP::UserAgent;
my $h = HTTP::Headers->new();

if ($proxy)
{
    $ua->proxy(['http', 'ftp'], "http://$proxy:$proxyport");
    if ($proxyuser)
    {
        $h->proxy_authorization_basic($proxyuser,$proxypass);
    }
}
if ($urluser)
{
    $h->authorization_basic($urluser, $urlpass);
}

$req = HTTP::Request->new('GET', $url, $h);
$res = $ua->request($req);

if ($res->status_line =~ /^$expectstatus/)
{
    print "OK - Status: ".$res->status_line."\n";
    exit $ERRORS{"OK"};
}
else
{
    print "CRITICAL - Status: ".$res->status_line." (but expected $expectstatus...)\n";
    exit $ERRORS{"CRITICAL"};
}
}
```

Tabla 108.- Script para chequear servicio Squid en Proxy

### 8.3.7. Chequear el servicio DHCP en Windows

Ejecución de chequeo en modo activo desde el servidor de Nagios, hacia los servidores indicados.

```
define service{
  use                               servidores-service
  host_name                         localhost, localhost2
  service_description              DHCP
  contact_groups                   Monitorizacion
  check_command                    check_nt_service!"DHCPServer"
}
```

Tabla 109.- Chequear DHCP en un Windows

### 8.3.8. Chequear un disco en Windows

Ejecución de chequeo en modo activo desde el servidor de Nagios, hacia los servidores indicados.

```
define service{
  use                               servidores-service
  host_name                         localhost1
  service_description              Disco_D
  contact_groups                   Monitorizacion
  check_command                    check_nt_disk!D!90!95
}
```

Tabla 110.- Chequear disco en un Windows

### 8.3.9. Chequear procesos de Antivirus en Windows

Ejecución de chequeo en modo activo desde el servidor de Nagios, hacia los servidores indicados.

```
define service{
  use                               servidores-service
  host_name                         localhost1
  service_description              Symantec_Endpoint_Protection
  contact_groups                   Monitorizacion
  check_command                    check_nt_service!"SepMasterservice"
}

define service{
  use                               servidores -service
  host_name                         localhost1
  service_description              Symantec_Management_Client
  contact_groups                   Monitorizacion
  check_command                    check_nt_service!"SmcService"
}
```

Tabla 111.- Chequear proceso Antivirus en Windows

# **Bibliografía**

---

## 9. Bibliografía

A continuación, se van a detallar todas las páginas web, antiguos TFGs y documentos encontrados en Internet, que han servido de referencia para la realización de la presente documentación y el desarrollo completo del TFG.

- [1] Cayuqueo, S. (2015). *CAPACITACIÓN EN MONITREO DE RED*. 1st ed. [ebook] Available at: <http://cayu.com.ar/files/manuales-nagios.pdf>
- [2] Doc.ubuntu-es.org, (2015). *Monitorización/Comparativa - doc.ubuntu-es*. [online] Available at: <http://doc.ubuntu-es.org/Monitorizaci%C3%B3n/Comparativa>
- [3] Docs.nscient.org, (2015). *NSClient++ Core Tutorial — NSClient++ 0.4.3 documentation*. [online] Available at: <http://docs.nscient.org/tutorial/core/index.html>
- [4] El Despistado., (2013). *NSCLIENT++. Instalación y configuración básica. - El Despistado..* [online] Available at: <http://www.eldespistado.com/nsclient-instalacion-configuracion-basica/>
- [5] Es.wikipedia.org, (2015). *Comparación de sistemas de monitorización de redes*. [online] Available at: [https://es.wikipedia.org/wiki/Anexo:Comparaci%C3%B3n\\_de\\_sistemas\\_de\\_monitorizaci%C3%B3n\\_de\\_redes](https://es.wikipedia.org/wiki/Anexo:Comparaci%C3%B3n_de_sistemas_de_monitorizaci%C3%B3n_de_redes)
- [6] Nagios, (2015). *Nagios Documentation - Nagios*. [online] Available at: <https://www.nagios.org/documentation/>
- [7] Exchange.nagios.org, (2015). *Nagios Exchange*. [online] Available at: <https://exchange.nagios.org/>
- [8] Informatica.iesvalledeljerteplasencia.es, (2015). *Monitorización de servidores – Nagios | Ciclo Formativo de Grado Superior Administración de Sistemas Informáticos en Red*. [online] Available at: <http://informatica.iesvalledeljerteplasencia.es/wordpress/monitorizacion-de-servidores-nagios/>
- [9] Lhotse21.gfz-potsdam.de, (2015). *NagVis 1.7 Documentation*. [online] Available at: [http://lhotse21.gfz-potsdam.de/nagvis/docs/en\\_US/index.html](http://lhotse21.gfz-potsdam.de/nagvis/docs/en_US/index.html)
- [10] Linux-party.com, (2015). *Noticias de Linux y tecnología - LinuxParty*. [online] Available at: <http://www.linux-party.com/index.php/106-nagios/>
- [11] Nagios, (2015). *Nagios Core. Nagios Open Source Project..* [online] Available at: <http://www.nagios.org>

- [12] Nagios.sourceforge.net, (2015). *Custom Object Variables*. [online] Available at: [http://nagios.sourceforge.net/docs/3\\_0/customobjectvars.html](http://nagios.sourceforge.net/docs/3_0/customobjectvars.html)
- [13] Nagios.sourceforge.net, (2015). *External Commands*. [online] Available at: [http://nagios.sourceforge.net/docs/3\\_0/extcommands.html](http://nagios.sourceforge.net/docs/3_0/extcommands.html)
- [14] Nagios.sourceforge.net, (2015). *Nagios Core Documentation - Table Of Contents*. [online] Available at: <http://nagios.sourceforge.net/docs/nagioscore/4/en/toc.html>
- [15] Support.nagios.com, (2015). *Nagios Support Wiki - Main Page*. [online] Available at: [http://support.nagios.com/wiki/index.php/Main\\_Page](http://support.nagios.com/wiki/index.php/Main_Page)
- [16] Tropiezos en la red, (2014). *Instalar y configurar NAGIOS para la monitorización de equipos en la red..* [online] Available at: <http://www.tropiezosenlarede.com/instalar-y-configurar-nagios-para-la-monitorizacion-de-equipos-en-la-red>
- [17] Universidad Oberta de Catalunya, (2015). *TFC Implantación de Sistema de Monitorización Nagios e Integración con Twitter*. [online] Cataluña: María Isabel March Hermo. Available at: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/40561/3/dperverTFC0115memoria.pdf>
- [18] Díaz, O. (2013). *Crear contactos y grupos en Nagios*. [online] Un byte tras otro. Available at: <https://byteafterbyte.wordpress.com/2013/06/28/crear-contactos-y-grupos-en-nagios/>
- [19] Zenoss, (2015). *Zenoss*. [online] Available at: <http://www.zenoss.com/>
- [20] Cacti.net, (2015). *Cacti® - The Complete RRDTool-based Graphing Solution*. [online] Available at: <http://cacti.net/>
- [21] Wiki.pandorafms.com, (2015). *Pandora FMS Wiki*. [online] Available at: [http://wiki.pandorafms.com/index.php?title=Main\\_Pag](http://wiki.pandorafms.com/index.php?title=Main_Pag)
- [22] LLC, Z. (2015). *Zabbix :: The Enterprise-Class Open Source Network Monitoring Solution*. [online] Zabbix.com. Available at: <http://www.zabbix.com/>
- [23] Galstad, E. (2007). *NDOUtils. 1st ed.* [ebook] Available at: <https://assets.nagios.com/downloads/nagioscore/docs/ndoutils/NDOUtils.pdf>
- [24] Galstad, E. (2007). *NRPE. 1st ed.* [ebook] Available at: <https://assets.nagios.com/downloads/nagioscore/docs/nrpe/NRPE.pdf>
- [25] Galstad, E. (2015). *NDOUtils - Nagios Exchange*. [online] Exchange.nagios.org. Available at: <https://exchange.nagios.org/directory/Addons/Database-Backends/NDOUtils/details>

- [26] Galstad, E. (2015). *NRPE - Nagios Remote Plugin Executor - Nagios Exchange*. [online] Exchange.nagios.org. Available at: <https://exchange.nagios.org/directory/Addons/Monitoring-Agents/NRPE--2D-Nagios-Remote-Plugin-Executor/details>
- [27] Blackshell.usebox.net, (2015). *Comandos UNIX*. [online] Available at: [http://blackshell.usebox.net/pub/shell/taller\\_sh/x137.html](http://blackshell.usebox.net/pub/shell/taller_sh/x137.html)
- [28] Michelsen, L. (2015). *Welcome to NagVis Home! - NagVis.org*. [online] Nagvis.org. Available at: <http://www.nagvis.org/>
- [29] Homann, R. (2015). *Documentation | NagiosQL*. [online] Nagiosql.org. Available at: <http://www.nagiosql.org/documentation.html>
- [30] La Comunidad de Nagios en Chile, (2013). *Configuración de NagiosQL*. [online] Available at: <http://www.nagios-cl.org/configuracion-de-nagiosql>
- [31] Docs.pnp4nagios.org, (2015). *es:pnp-0.6:install [PNP4Nagios Docs]*. [online] Available at: <http://docs.pnp4nagios.org/es/pnp-0.6/install>
- [32] Cayu.com.ar, (2015). *manuales:nagios* [Cayu - Wiki de Sergio Cayuqueo]. [online] Available at: <http://cayu.com.ar/wiki/doku.php?id=manuales:nagios>

**Alexandra Moreno Calabozo**  
Entregado: 24 de septiembre de 2015  
Defensa: 8 de octubre de 2015