Universitat Autònoma
de Barcelona

SISTEMA DE MONITORIZACIÓN DE SERVICIOS EN LINUX

Memòria del projecte

d'Enginyeria Tècnica en

Informàtica de *Sistemes*

realitzat per

Carlos Ramos Gallardo

i dirigit per

Marc Talló Sendra

Escola Universitària d'Informàtica

Sabadell, *Juny* de *2010*

El/la sotasignant, *Marc Talló Sendra*,

professor/a de l'Escola Universitària d'Informàtica de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en *Carlos Ramos Gallardo*

I per a que consti firma la present.
Sabadell, *Juny* de *2010*

Signat: *Marc Talló Sendra*

Resumen

En la actualidad, se puede considerar casi obligatorio el uso de herramientas de control de los recursos informáticos por parte de las empresas del sector de las tecnologías de la información. Esto se debe a que disponer de un sistema de monitorización centralizado que permita gestionar las incidencias en las infraestructuras informáticas implica un considerable ahorro de tiempo, personal y, por lo tanto, dinero. Además, mejora la calidad del servicio que se ofrece y proporciona seguridad a los usurarios, clientes y responsables informáticos que utilizan o gestionan éstos sistemas.

El objetivo de este proyecto es implementar un sistema de monitorización de servicios en Linux que ofrezca ciertas funcionalidades. Se estudian diferentes herramientas de monitorización Open Source competentes el sector. Se utiliza *Nagios* como sistema de monitorización principal adaptándolo a las necesidades del proyecto.

El sistema permite controlar desde un entorno Web y de manera visual el estado de los monitores configurados, y realizar tareas relacionadas con los mismos, como la generación de informes, visualización de logs o deshabilitar monitorizaciones y alertas. Por otro lado, notifica, mediante SMS y correo electrónico, de las alertas que surjan en los servidores y servicios gestionados. Envía alertas cuando un servicio falla, no responde o no está operativo y, posteriormente, notifica cuando el estado del monitor vuelve a ser el correcto. También permite recuperar servicios que se han venido abajo de manera automática y sin necesidad de la intervención humana. Otro aspecto importante para el seguimiento del rendimiento de los servicios es la generación de gráficas de los monitores, que también lo ofrece el sistema de monitorización implementado.

Un aspecto que me ha motivado a la elegir este proyecto ha sido el haber utilizado de manera profesional, en el mundo laboral, herramientas de monitorización, con lo que he podido ver de cerca las ventajas y facilidades que proporcionan respecto a no tener ningún tipo de control de incidencias en las plataformas de los clientes.

Contenido

1. Introducción

En éste primer capítulo de la memoria se describe la presentación inicial del proyecto. Seguidamente se plantean los objetivos generales. A continuación se expone el estado del arte. Posteriormente se expresan las motivaciones que han generado el interés y la elección de éste proyecto. Por último, se define la estructura de la memoria, los capítulos que contiene y el formato utilizado.

1.1. Presentación

Desde el departamento de ciencias de la computación de la Universidad Autónoma de Barcelona (UAB) se presenta éste proyecto que tiene como título "Sistemas de monitorización de servicios en Linux".

El trabajo a realizar consiste en el estudio e implementación de una herramienta de monitorización de servicios sobre un servidor Linux. El servidor en cuestión debe prestar diferentes servicios: Web, correo electrónico, bases de datos (**BBDD**) o Domine Name System (**DNS**). La aplicación a implementar controlará que los servicios se ejecuten de manera satisfactoria en el servidor. Si, por el contrario, un servicio se viene abajo o no se ejecuta correctamente, el sistema debe enviar alertas configurables a móvil (**SMS**) y correo electrónico (e-mail) para notificar los problemas específicos en los servicios del servidor.

Hoy en día es muy habitual que las empresas dispongan de servicios externalizados de alojamiento de servidores y servicios (Hosting). Esto implica que los administradores informáticos responsables de gestionar los servicios no dispongan físicamente de los servidores a los que acudir presencialmente si surge cualquier tipo de incidencia. Además, ya sean servidores internos en la propia empresa o externos, muchas veces se hace difícil detectar un problema a nivel de servicio y es el cliente o el usuario final el encargado de alertar de la incidencia. Pero, sin duda alguna, el factor económico pesa sobre todos los demás: el acuerdo de nivel de servicio o Service Level Agreement (**SLA**) pactado con el cliente en relación a la calidad de servicio hace muy necesario el disponer de un sistema de control y notificación de incidencias con el que poder actuar de manera rápida y dentro del margen de tiempo estipulado. No cumplir éstos contratos puede implicar grandes pérdidas de dinero e incluso el hecho de perder definitivamente al cliente.

1.2. Objetivos

Los objetivos generales que se han determinado son los siguientes:

- Instalar un servidor Linux y configurar determinados servicios, como los ya mencionados anteriormente, creando un entorno de trabajo adecuado para la posterior implementación del sistema de monitorización.
- Estudiar las herramientas existentes en el sector analizando sus características y funcionalidades.
- Implementar un sistema de monitorización de servicios, en una máquina que actúe como cliente, apto para el entorno Linux en el que se trabaja y que permita gestionar los monitores y notificar las incidencias que puedan surgir en los mismos, vía SMS y e-mail.
- Realizar las pruebas pertinentes para analizar y valorar los resultados de la solución implementada en el entorno de trabajo.

El proyecto se aplica directamente a la necesidad existente de controlar los servicios de servidores en empresas con recursos informáticos, ya sea a nivel de servidores internos en sus propiedades físicas, o bien, alojados en servidores externos y gestionados por otras entidades.

La herramienta a implementar permite solucionar el problema que tienen las empresas con el control de sus recursos informáticos dando la posibilidad de notificar las incidencias e incluso auto solucionarlas, en la medida de lo posible, o escalarlas a los responsables informáticos correspondientes. Todo ello desde una gestión remota a los servidores y cumpliendo con los SLA establecidos con los clientes.

En el segundo capítulo de ésta memoria: "Estudio de viabilidad. Objetivos" se describen los objetivos específicos que se deben satisfacer para el éxito del proyecto.

1.3. Estado del arte

Como ya se ha introducido anteriormente, la monitorización de servidores, servicios y redes permite un control sobre la administración de los sistemas que resulta ser la base para la calidad del servicio que se ofrece.

En el sector informático y, más específicamente, en el ámbito de la consultoría de sistemas y **TI** (tecnologías de la información) existen diferentes herramientas competentes de control y monitorización remota de los recursos.

Un ejemplo de software privado es *HP Sitescope*. Es una herramienta comercial del fabricante Mercury. En el año 2006 Hewlett Packard (HP) compró el software Mercury Interactive. Éste software permite la monitorización de servicios, parámetros de servidor o redes y comunicaciones. No necesita de la instalación de software en los servidores monitorizados. Permite una gran posibilidad de configuraciones de los monitores. También ofrece la posibilidad de configurar alertas a móviles (SMS) y a correos electrónicos.

Un ejemplo open source (software desarrollado y distribuido libremente) con licencia **GNU GPL** (General Public License) es *Nagios*. Es una herramienta de monitorización de redes, hardware y software. Permite la posibilidad de programación de Plugins (complementos) para monitorizaciones a medida. Estos complementos dan la posibilidad de ampliar o modificar las funciones por parte de los desarrolladores y depurar problemas de software. Éste software también ofrece un sistema de alertas SMS y correo electrónico.

Otra herramienta open source a destacar es *PandoraFMS*. Permite controlar el hardware, software, sistemas operativos o interfaces de red de la infraestructura en la que trabaje. Tiene la capacidad de generar informes y estadísticas, controlar SLA's y medir rendimientos a muchos niveles: conexiones, sesiones, servicios, etc. También permite configurar notificaciones e incidencias vía SMS o e-mail.

Existen otras aplicaciones más orientadas a la gestión de la red y las comunicaciones. Como *Cacti*, una herramienta que ofrece estadísticas y gráficas a éste nivel trabajando con el **RRDTool** (Round Robin Database tool) muy eficiente en el control de datos temporales y seriales.

En el segundo capítulo de ésta memoria: "Estudio de viabilidad. Estado del arte" se realiza un análisis comparativo de totas las herramientas estudiadas, tanto a nivel de funcionamiento, como de capacidad, lenguajes de programación utilizados en el desarrollo e implementación, etc.

1.4. Motivaciones

La experiencia laboral previa en el sector de la consultoría de sistemas es el motivo por el cual se ha conocido de cerca la necesidad de utilizar éste tipo de herramientas para gestionar las diferentes plataformas y servidores de los clientes.

Cuando el volumen de servidores y, por tanto, de servicios, se contabiliza en centenares resulta casi obligatorio disponer de una gestión centralizada de los mismos para poder actuar eficientemente ante cualquier incidencia que pueda surgir.

Como se ha introducido en el anterior apartado, existe una gran cantidad de herramientas que permiten gestionar la infraestructura informática de una empresa, desde la comunicación con los servidores y los servicios que ofrecen hasta las interfaces y dispositivos de red, redes virtuales, túneles de comunicaciones, etc. Monitorizar todos estos recursos, poder generar gráficas y estadísticas, analizar rendimientos o notificar incidencias son muchas de las posibilidades de estos software. Para llevar a cabo estas tareas, se utilizan diferentes protocolos, métodos y lenguajes de programación.

La motivación principal para el desarrollo del proyecto es analizar, desde el punto de vista crítico y objetivo de un ingeniero, las herramientas que existen para, posteriormente, implementarlas en un entorno de trabajo creado para este fin.

Conociendo las necesidades reales del sector y las herramientas trabajadas e implementadas en el entorno mencionado, se pretende medir el funcionamiento y la capacidad de las mismas realizando las pruebas prácticas pertinentes.

1.5. Estructura de la memoria

Para el correcto entendimiento de la memoria se describen los siguientes capítulos que la componen:

- Estudio de viabilidad: se determina la viabilidad del proyecto en base a los objetivos determinados, el estudio actual del sector y sus necesidades, las especificaciones (tanto funcionales y no funcionales como técnicas), la planificación (tanto a nivel de horas invertidas en los diferentes perfiles de trabajo como a nivel económico) y los y riesgos existentes.

- Fundamentos teóricos: se explican todos los conceptos necesarios para entender las herramientas, protocolos y tecnologías utilizadas en el transcurso del proyecto.
- Análisis: se describe el diseño y funcionamiento de las herramientas analizadas y del entorno de pruebas implementado.
- Implementación: se describen las fases de implantación de las herramientas en el entorno de pruebas desarrollado.
- Pruebas: se realizan las pruebas necesarias para medir la capacidad y funcionamiento de las herramientas implementadas en el entorno de pruebas y sus funcionalidades.
- Conclusiones: se describen las conclusiones finales después de la realización del proyecto y las desviaciones surgidas durante el transcurso del mismo.
- Bibliografía: se especifican las fuentes de información utilizadas durante el desarrollo del proyecto.
- Apéndice: se detalla un índice de ilustraciones, los códigos fuente implementados y comentados y un glosario con la explicación de términos y abreviaturas utilizados en el transcurso del proyecto.

1.5.1. Formatos del texto

A continuación se especifica el formato utilizado en la memoria:

- Fuente: Verdana.
- Tipografía:
  - Título 1:
    - Fuente: Negrita.
    - Tamaño de la fuente: 14.
  - Título 2:
    - Fuente: Negrita.
    - Tamaño de la fuente: 12.
  - Título 3:
    - Fuente: Negrita.
    - Tamaño de la fuente: 11.
  - Texto:
    - Fuente: Normal.
    - Tamaño de la fuente: 10.

- Tablas:
  - Título:
    - Fuente: Negrita.
    - Tamaño de la fuente: 8.
  - Campos de título:
    - Fuente: Negrita.
    - Relleno del sombreado: Gris claro.
    - Tamaño de la fuente: 8.
  - Otros campos:
    - Fuente: Normal.
    - Relleno del sombreado: Sin relleno.
    - Tamaño de la fuente: 8.
- Ilustraciones:
  - Título:
    - Fuente: Negrita.
    - Tamaño de la fuente: 8.
- Formato del texto: justificado.
- Interlineado: 1,5.

1.5.2. Nomenclaturas

Los nombres de las empresas, entidades, marcas o software mencionados en la memoria se expresan en cursiva. Ejemplo: *Nagios*.

Las palabras y abreviaciones que se incluyen en el glosario para su mejor entendimiento se determinan en negrita. Ejemplo: **GNU GPL**.

La mención concreta a algún apartado de la memoria se detalla entre paréntesis. Ejemplo: "Estudio de viabilidad. Objetivos".

2. Estudio de viabilidad

Una vez introducidos los objetivos generales del proyecto y el contexto en el que se sitúa es el momento de especificar, más concretamente, el alcance del mismo y todos los análisis necesarios para determinar su viabilidad.

2.1. Introducción

Éste capítulo describe los aspectos fundamentales del proyecto determinando, primeramente, los objetivos concretos a los que se quiere llegar, el estado del arte, las especificaciones, los perfiles del sistema, los recursos necesarios, la planificación del trabajo, los riesgos y valoraciones y, finalmente, las conclusiones.

2.2. Objetivos

Los objetivos específicos que se deben cumplir para satisfacer las necesidades del proyecto son los siguientes:

- Instalación de un servidor con plataforma Linux: estudiar y seleccionar una distribución Linux para instalarla en una máquina que actúe como servidor en el hardware del que se dispone para el entorno de pruebas.
- Configuración de los servicios en el servidor: seleccionar, instalar y configurar diferentes servicios de propósito general en el servidor.
- Instalación de un cliente con plataforma Windows.
- Instalación del software necesario en el cliente: configurar el cliente Windows con las aplicaciones necesarias.
- Configuración de la red local (**LAN**) en el entorno de pruebas para la comunicación entre el cliente y el servidor.
- Realización de un estudio comparativo de las herramientas de monitorización open source: estudiar y seleccionar las herramientas de monitorización a implantar en el entorno de pruebas.
- Instalación de las herramientas de monitorización: instalar y configurar las herramientas previamente seleccionadas.
- Realización de las pruebas pertinentes para comparar los resultados obtenidos en relación a los resultados esperados.

2.3. Estado del arte

En los sistemas de monitorización de recursos informáticos se pueden encontrar diferentes soluciones, tanto privadas como de distribución gratuita, que ofrecen gran variedad de posibilidades. Las necesidades de las empresas que las requieren son las que dictan la utilización de unas determinadas herramientas u otras.

2.3.1. Recursos que permiten monitorización

En la infraestructura informática de una empresa existen diferentes recursos que pueden llegar a monitorizarse con las herramientas adecuadas:

- Hardware:
  - Estado y rendimiento de dispositivos de red o periféricos.
  - Temperaturas y voltajes de los elementos de servidores, dispositivos o armarios de comunicaciones.
  - Consumo de memoria **RAM**, caché, procesadores o espacio en discos duros.
- Software:
  - Sistemas operativos.
  - Servicios que se ejecutan en una máquina.
  - Rendimiento de aplicaciones y bases de datos.
  - Sistemas virtuales.

2.3.2. Servicios de propósito general

Este proyecto, como su propio título indica, está enfocado a la monitorización de servicios. Los servidores corporativos con plataformas Linux pueden ofrecer diferentes servicios, siempre en función de las necesidades y los recursos de las empresas. Los más comunes son:

- Páginas Web.
- Acceso remoto por línea de comandos.
- **FTP**.
- Bases de datos.
- Aplicaciones.
- Antivirus.
- Antispam.

- Datos y ficheros.
- Correo electrónico.
- DNS.
- Proxies.
- Noticias.
- Foros.
- Chats.
- Repositorios de información.

2.3.3. Estudio comparativo de las herramientas de monitorización

Es necesario realizar un estudio comparativo acorde para analizar las posibilidades que ofrecen las diferentes soluciones. No obstante, solo se estudian aplicaciones open source con licencias que permiten la distribución gratuita del software. De ésta manera se minimiza el coste del proyecto considerándose, por supuesto, que al utilizar un servidor de plataforma Linux la competencia de herramientas open source es alta y existen muchas posibilidades robustas y estables, adecuadas para el desarrollo de este proyecto, sin necesidad de utilizar software privado.

La siguiente tabla muestra las características de las 10 herramientas seleccionadas, que son las que se han considerado más competentes después del estudio realizado:

	Ámbito monitorización	**Código**	**Comunicación/Configuración**	**Software requerido**	**Interfaz**	**Plantillas**	**Alertas**
Cacti	Networking Gráficas	PHP	SNMP Scritps RRDTool	Cron MySQL Apache PHP SNMP RRDTool	Admin. Usuarios	Config. monitores	

	Ámbito monitorización	Código	Comunicación/Configuración	Software requerido	Interfaz	Plantillas	Alertas
God	General	Ruby on Rails Open Source	Scripts				E-mail
JFFNMS	Networking	PHP	SNMP RRDTool	Cron Apache PHP SNMP MySQL RRDTool Nmap	Admin. Usuarios Monitores		E-mail
Monit	General	C	Scripts	Apache	Admin. Monitores		E-mail
Munin	Networking Gráficas	Perl	SNMP Scripts Plugins RRDTool	Munin-node (en el cliente) Perl Apache o similar	Admin. Usuarios Monitores		E-mail
Nagios	General	C	SNMP Scripts Plugins	Apache	Admin. Monitores	Configuraciones varias Monitores	E-mail SMS
Osmius	General	Java JavaScript C++	Scripts Eventos	Tomcat MySQL JDK	Admin. Usuarios Monitores	Configuraciones varias	E-mail SMS

	Ámbito monitorización	Código	Comunicación/Configuración	Software requerido	Interfaz	Plantillas	Alertas
				ACE			
PandoraFMS	General Gráficas	Perl PHP	SNMP Scripts Tentacle SSH FTP Plugins	Apache MySQL Perl SNMP Nmap WMI	Admin. Usuarios Monitores	Configuraciones varias Monitores	E-mail SMS
Zabbix	General	C PHP JavaScript	SNMP SSH IPMI Eventos	Apache PHP MySQL/PostgreSQL/SQLite/Oracle	Admin. Usuarios Monitores	Configuraciones varias Monitores	E-mail SMS
Zenoss	General	Phyton	SNMP Twisted RRDTool Eventos Plugins	Zope Python SNMP MySQL	Admin. Usuarios Monitores	Configuraciones varias Monitores	E-mail SMS

Tabla 1: Comparativa de herramientas de monitorización open source

Como se puede observar en la tabla, existen herramientas que están enfocadas a los dispositivos de red y comunicaciones. Otras, en cambio, son de propósito general. Por lo tanto, se plantea la posibilidad de utilizar una combinación de varias herramientas para obtener un resultado más óptimo y completo.

El lenguaje de programación difiere en varias de las herramientas analizadas ya que algunas están desarrolladas en **C**, **C++**, **PHP**, Java, Perl o Phyton. Los plugins o complementos que se pueden incluir en algunas de ellas no tienen porqué estar desarrollados en el lenguaje propio de la herramienta.

También existen diferentes tipos de métodos y protocolos a nivel de configuración y comunicación entre los diferentes módulos o funcionalidades de las herramientas. Predomina el protocolo **SNMP** y las configuraciones mediante scripts y también se pueden encontrar varias soluciones que utilizan el método RRDTool enfocado la generación de gráficas y estadísticas.

Casi todas las herramientas estudiadas necesitan software o paquetes adiciones para funcionar correctamente. Es el caso de un servidor Web para el entorno visual de gestión y monitorización y las BBDD en las que gestionan y almacenan la información.

Algunas herramientas permiten la administración de los monitores sobre un entorno Web controlando los usuarios que acceden mediante la gestión de grupos y permisos.

El uso de plantillas para la configuración del sistema y de los monitores también se considera un factor a tener en cuenta ya que no lo soportan todas las herramientas.

Es necesario que las soluciones implementen un sistema de notificación mediante SMS y correo electrónico. Algunas de las herramientas incluso soportan algún método más, como la mensajería instantánea.

Que algunos apartados de la tabla estén vacíos es debido a que la documentación oficial de la solución correspondiente no detalla esa información. No significa, de forma trivial, que la herramienta no disponga de esa característica.

2.4. Especificaciones

En este apartado se describen las diferentes especificaciones funcionales, no funcionales y técnicas.

2.4.1. Funcionales

Las especificaciones funcionales son las que la solución implantada debe satisfacer a nivel de funcionalidad. Se determinan las siguientes:

- Monitorizar los servicios del servidor desde un entorno Web: controlar los servicios mediante el estado visual de los monitores configurados.
- Notificar las alertas o errores en los monitores: enviar avisos mediante mensajes a móvil y correo electrónico de los monitores cuando el estado de los mismos indique que existe un error.
- Auto recuperar servicios no operativos: levantar automáticamente los servicios que se han venido abajo cuando el estado de los monitores así lo indique.
- Generar gráficas y estadísticas de los monitores: controlar gráficamente el estado y rendimiento de los servicios controlados por los monitores.

2.4.2. No funcionales

Las especificaciones no funcionales son las que dan un valor añadido al resultado final de la solución implementada. Se detallan las siguientes:

- El sistema debe solicitar un usuario y contraseña para acceder al entorno de monitorización.
- El sistema debe satisfacer una gestión completa del entorno monitorizado, siempre sujeta a los permisos del usuario que accede.
- La gestión de los monitores y todas las configuraciones relacionadas debe ser intuitiva y de fácil uso para los usuarios.
- El sistema debe ser estable y seguro para poder implementarse en cualquier entorno corporativo que lo requiera.
- El sistema debe ser escalable, siendo apto para un mayor número de servidores, servicios y clientes y sin sufrir ninguna pérdida de eficiencia.
- El acceso y gestión en el entorno de monitorización debe ser rápido y efectivo.

2.4.3. Técnicas

Las especificaciones técnicas son las necesidades que deben cumplirse a éste nivel. Se estipulan las siguientes:

- Se requiere de un servidor o conjunto de servidores, ofreciendo determinados servicios, para ser monitorizados.
- Se necesitan una o varias máquinas que actúen como cliente en el sistema de monitorización.
- Los clientes deben disponer de navegador Web y el sistema debe soportar diferentes versiones y modelos.
- Se requiere de un dispositivo de red que ofrezca una conexión LAN y también acceso a Internet.
- El entorno Web debe ser accesible desde la red local y desde Internet.

2.5. Perfiles

En este apartado se describen los perfiles de usuarios que pueden utilizar el sistema implementado:

- Administrador: gestión y acceso total al sistema de monitorización.
- Usuarios: el acceso y posibilidades en el entorno de monitorización Web dependerá de los permisos de los que dispongan y los grupos en los que estén incluidos.

2.6. Recursos

A continuación se detallan los recursos de hardware y software disponibles y necesarios para el entorno de pruebas.

2.6.1. Hardware

El hardware consta de una máquina que funciona como servidor, otra que actúa como cliente y un router que ofrece la conexión LAN y la de Internet.

- Servidor:
  - Procesador: AMD Athlon XP 1800+ a 1,5 GHz.
  - Arquitectura: i686.

  - Memoria RAM: 768 MB.
  - Disco Duro: 40 GB.
- Cliente:
  - Procesador: Intel Core 2 Duo a 2 GHz.
  - Arquitectura: x86-64.
  - Memoria RAM: 2 GHz.
  - Disco Duro: 250 GB.
- Dispositivo de red local e Internet:
  - Dispositivo: Router.
  - Marca: *Comtrend*.
  - Modelo: HG536+.
  - Tipo: **ADSL+2**.
- Dispositivo móvil:
  - Dispositivo: Teléfono móvil.
  - Marca: Nokia.
  - Modelo: N70.
  - Tipo: **GSM** y **UMTS**.

2.6.2. Software del servidor

Existen muchas distribuciones Linux para servidores de propósito general. No se contemplan las distribuciones que exigen una licencia comercial, como en el caso de *Red Hat* o *Novell Suse*, puesto que no se necesita para la realización de éste proyecto y además se minimiza el coste final del mismo.

Se destacan las siguientes distribuciones gratuitas (versiones estables):

- *CentOS 4.8.*
- *Debian 5.0.4.*
- *Fedora 12.*
- *Mandriva 2010.*
- *OpenSUSE 11.2.*
- *Ubuntu 9.10 Server.*

A continuación se describen las características comunes entre ellas:

- Modelo de desarrollo: Software libre de código abierto.
- Licencia: GNU GPL (Licencia Pública General de GNU).

- Soporte en plataformas: i686 y x86-64 (entre otras).
- Tipo de núcleo: monolítico.

Se selecciona la distribución *Ubuntu 9.10 Server* por las siguientes razones:

- Está patrocinada por *Canonical Ltd.* que se financia a partir de la venta de soporte técnico y se mantiene de manera libre y gratuita, permitiendo a la comunidad de usuarios participar y mejorar el sistema.
- Está basada *en Debian* y posee una versión específica para servidor.
- Dispone de la instalación y gestión de paquetes Dpkg con la que se está más familiarizado que con el sistema **RPM** (que utilizan *Fedora*, *Mandriva* o *Suse*).
- Requiere de los requisitos mínimos de hardware siguientes:
    - Procesador: a 700 MHz.
    - Memoria RAM: 384 MB.
    - Disco Duro: 8 GB.

Cabe destacar que cualquiera de las otras distribuciones Linux estudiadas es apta para el desarrollo del proyecto.

Se determina la instalación y configuración de los siguientes servicios de propósito general que debe ofrecer el servidor:

- Servidor de acceso remoto:
    - *Open SSH*: utiliza el protocolo **SSH**. Posee una licencia **BSD** y dispone de técnicas de cifrado, a diferencia del Telnet*,* y permite acceder de manera completa y segura al servidor utilizando un intérprete de comandos.
- Servidores Web:
    - *Apache*: posee una licencia de Apache (GPL compatible).
    - *Apache Tomcat*: posee una licencia de Apache (GPL compatible) y, ambos, son ampliamente utilizados en sector informático y en plataformas Linux.
- Servidor de BBDD:
    - *MySQL*: posee una licencia GPL.
    - *PostgreSQL* con licencia BSD.
- Servidor de DNS:

- ***Bind*** *(Berkeley Internet Name Domain):* posee una licencia BSD y se ha convertido en un estándar por la gran aceptación otorgada por parte de los usuarios en plataformas Unix y Linux.

- Servidor de E-mail:
  - *Postfix*: es software libre y fue desarrollado por IBM. Otra alternativa válida podría ser *Sendmail*. Ambos son muy utilizados en plataformas Linux.
- Servidor de antivirus:
  - *ClamAV*: posee una licencia GPL.
  - *Amavis*: posee una licencia GPL y, ambos, son compatibles con el servidor *Postfix.*
- Servidor de antispam:
  - *SpamAssassin*: posee una licencia de Apache (GPL compatible) y es compatible con el servidor *Postfix*.
- Servidor de archivos:
  - *Samba*: posee una licencia GPL y permite la compartición de directorios y validación de usuarios entre diferentes plataformas Linux y Windows.

Para la implementación del sistema de notificaciones SMS es necesaria la siguiente herramienta:

- *Gnokii*: posee una licencia GPL y permite utilizar funciones como el envío de SMS de un teléfono móvil, compatible con la herramienta, desde la línea de comandos.

2.6.3. Software del cliente

El sistema operativo de la máquina cliente es el siguiente:

- Sistema operativo: *Windows 7 Ultimate*.
- Plataforma: Windows x86 (32 bits).

Se determina la utilización de un sistema operativo de plataforma *Windows* en el cliente, pese al incremento del coste que supone, ya que se considera importante realizar el proyecto trabajando con diferentes plataformas y permitiendo analizar, también, las incompatibilidades o problemas que pudieran surgir en base a este hecho.

El software necesario en la máquina cliente para el desarrollo del proyecto es el siguiente:

- Cliente SSH:
  - *PuTTY*: posee una licencia **MIT**, muy similar a la BSD y se considera software libre. Es un cliente SSH (entre otros) y permite el acceso remoto, utilizando éste mismo protocolo, a la máquina servidor.
  - *WinSCP*: posee una licencia GNU GPL. Es un cliente **SFTP** gráfico que utiliza el protocolo SSH.
- Navegador Web:
  - *Microsoft Internet Explorer*: incluido en el propio sistema operativo *Windows*. Otra alternativa podría ser *Mozilla Firefox* que es open source y gratuito (entre otros).
- Herramientas Ofimáticas:
  - *Microsoft Office 2007*: software privado de *Microsoft* para el desarrollo de la memoria.
  - *Microsoft Project 2007*: software privado de *Microsoft* para la planificación del proyecto.

2.6.4. Recursos humanos

Se definen diferentes perfiles para el desarrollo del proyecto en todas sus fases:

- Analista de sistemas: encargado de estudiar el problema y buscar las soluciones.
- Analista de software: encargado de desarrollar e implementar las soluciones.
- Técnico de sistemas: encargado de configurar e implantar el sistema.
- Documentador técnico: encargado de realizar la documentación escrita.

2.7. Planificación

En este apartado se realiza una planificación del tiempo de desarrollo del proyecto y se estiman los costes derivados.

2.7.1. Modelo de desarrollo

Para el desarrollo del proyecto se sigue el método de ciclo de vida clásico, que consta de las siguientes fases:

a) Investigación preliminar: selección del proyecto y negociación con el responsable del mismo para llevarlo a cabo.
b) Determinación de los requerimientos del sistema: analizar el problema, estudiar el entorno, plantear las decisiones y planificar el trabajo.
c) Diseño del sistema: especificar los detalles para que se cumplan con éxito todos los requerimientos analizados.
d) Desarrollo del software: desarrollar la herramienta en base a los requerimientos y el diseño previos.
e) Pruebas de software: testear el software desarrollado depurando y perfeccionando el producto final.
f) Implantación y evaluación: cuando el producto se determina apto se realiza la implantación en el entorno y la evaluación de los resultados.
g) Mantenimiento: una vez implantado el producto en el entorno real se realiza el correspondiente mantenimiento del software.

A continuación se muestra la representación gráfica de las principales fases anteriormente descritas:

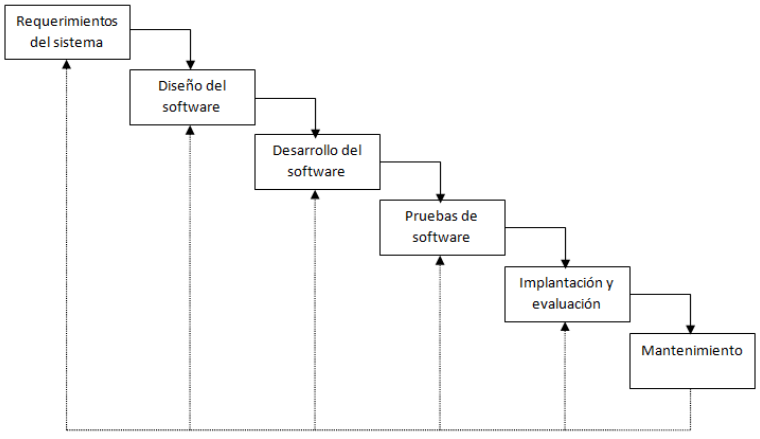


Ilustración 1: Fases del proyecto

2.7.2. Planificación temporal

Para controlar el tiempo invertido en cada fase del proyecto es necesario realizar una planificación acorde a las necesidades del proyecto.

En la siguiente tabla se muestran las diferentes tareas en las que se distribuye el proyecto y el perfil responsable de cada tarea y la duración estimada:

Nº	Tarea	Perfil	Duración (en horas)
1	Estudio previo del problema	Analista de sistemas	10
2	Estudio de viabilidad	Analista de sistemas	20
3	Instalación del servidor	Técnico de sistemas	6
4	Instalación y configuración de los servicios del servidor	Técnico de sistemas	20
5	Instalación y configuración del cliente	Técnico de sistemas	4
6	Configuración de la LAN e Internet	Técnico de sistemas	4
7	Análisis y diseño del sistema	Analista de software	50
8	Aprendizaje de las tecnologías y lenguajes necesarios	Analista de software	84
9	Instalación de las herramientas de monitorización	Técnico de sistemas	10
10	Configuración de las herramientas de monitorización	Técnico de sistemas	20
11	Implementación y desarrollo de las funcionalidades	Analista de software	80
12	Pruebas de software y verificación final	Técnico de sistemas	10
13	Elaboración de la documentación	Documentador técnico	40
TOTAL			**358**

Tabla 2: Tareas, perfil y duración del proyecto

El diagrama de Gantt correspondiente a las tareas anteriormente descritas es el siguiente:

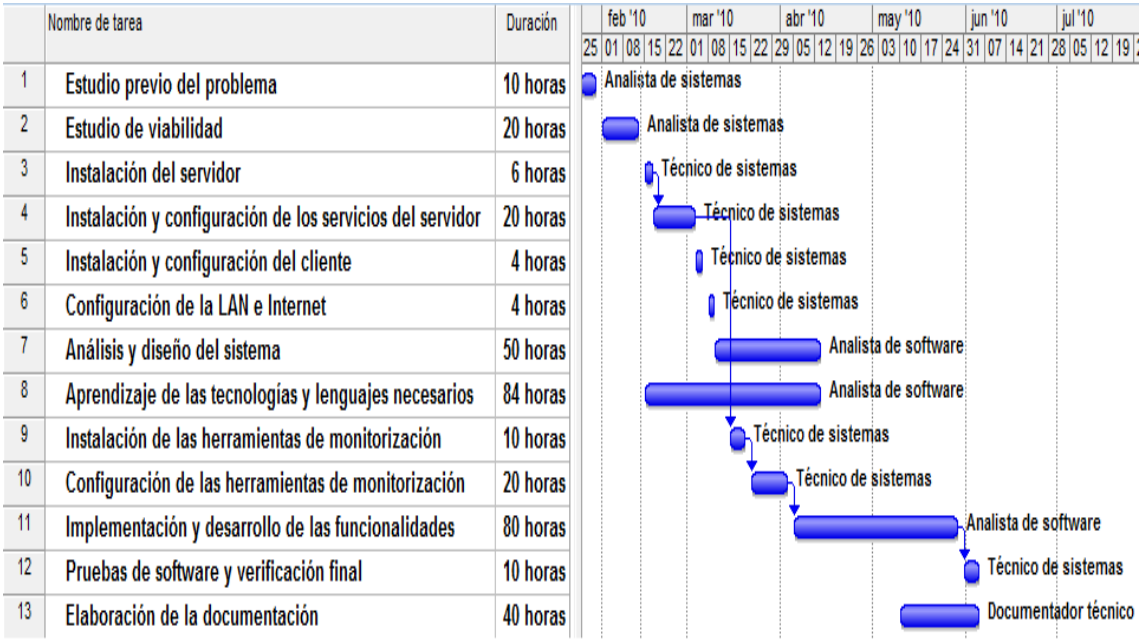


Ilustración 2: Diagrama de Gantt

2.7.3. Planificación de costes

En este apartado se estiman los gastos económicos contemplando el sueldo de los diferentes responsables, los costes del hardware utilizado y los del software necesario.

En la tabla siguiente se muestra el gasto del proyecto en función del sueldo de los diferentes responsables:

Perfil	Coste/hora	Horas	Coste (en euros)
Analista de sistemas	20	30	600
Analista de software	20	214	4280
Técnico de sistemas	12	74	888
Documentar técnico	6	40	240
TOTAL			**6008**

Tabla 2: Coste de los responsables del proyecto

A continuación se muestran los gastos relacionados con el hardware:

Material	Coste (en euros)
Ordenador servidor	600
Ordenador cliente	1000
Router	30
TOTAL	**1630**

Tabla 3: Gastos de hardware

En la tabla siguiente se detallan los gastos del software:

Material	Coste (en euros)
Sistema operativo servidor	0
Sistema operativo cliente	319
Software adicional servidor (servidores/servicios)	0
Herramientas ofimáticas	1430
TOTAL	**1749**

Tabla 4:Gastos de software

Y, finalmente, se especifican los gastos totales:

Material	Coste (en euros)
Personal	6008
Hardware	1630
Software	1749
Consumo de electricidad	100
Internet (ADSL)	120
TOTAL	**9607**

Tabla 5: Gastos totales

2.8. Riesgos

Es importante tener en cuenta los riesgos que existen a diferentes niveles tanto técnicos como de costes y, por supuesto, de tiempo:

- Conocimientos técnicos: limitaciones en el desarrollo e implementación de las funcionalidades de las herramientas.
- Factor económico: trabajar en plataformas privadas como Windows implica costes asociados. También el uso de servicios y complementos que no sean de distribución gratuita repercuten en el coste final del producto.
- Factor tiempo: retrasarse en cualquiera de las fases del proyecto puede afectar en el desarrollo del mismo hasta el punto de no conseguir ajustarse a las diferentes fechas de presentación de documentos, verificaciones y en el resultado del producto final.

2.8.1. Alternativas

Para reducir el riesgo ante los factores planteados se generan las siguientes alternativas asociadas:

- Posibilidad de cambiar de lenguajes de programación y tecnologías si existieran serias dificultades para avanzar en los seleccionados previamente. Esto es posible si la planificación y estructura del proyecto son las adecuadas, puesto que los objetivos a satisfacer no limitan el lenguaje de programación ni las tecnologías a utilizar.
- A nivel de costes, se va a trabajar en una plataforma Linux. Los servicios a implantar en éste entorno son de código abierto y licencias de libre distribución, para minimizar, en la medida de lo posible, en coste final del producto.
- En relación al tiempo del que se dispone, realizar una correcta planificación y distribución de las tareas y recursos se vuelve un factor básico para el éxito del proyecto.

2.9. Valoración

Después de haber finalizado el estudio de viabilidad se consideran las siguientes valoraciones:

- Planificar el trabajo en fases, estudiando los recursos necesarios y los costes de tiempo y dinero asociados es fundamental para encaminar correctamente el proyecto desde el primer día.
- Realizar un estudio de las soluciones actuales del mercado es básico para entender cómo solucionar los problemas que se plantean en el contexto adecuado.
- Estudiar las tecnologías y lenguajes de programación necesarios para el desarrollo e implementación de las funcionalidades del sistema es necesario para llevar a cabo un desarrollo óptimo minimizando los problemas técnicos y de conocimiento.

2.10. Conclusiones

Y, finalmente, después del estudio realizado se detallan las siguientes conclusiones:

- Los sistemas de monitorización ofrecen:
  - Control centralizado y completo del estado y rendimiento de los servidores.
  - Reducción de las tareas de resolución y gestión de las incidencias.
  - Notificación de alertas e incidencias automática.
  - Reducción de personal humano trabajando en la infraestructura informática.
  - Aumenta la seguridad y la calidad del servicio.
- Utilizar software open source y de distribución gratuita minimiza sustancialmente los costes del producto final y mejora la calidad de las herramientas utilizadas e implementadas en proyectos futuros sin las limitaciones de los software privados.
- Se puede considerar que el proyecto es viable considerando las mejoras que se obtienen con el producto final en relación a los costes técnicos y económicos invertidos.

3. Fundamentos teóricos

Antes de empezar con el análisis del sistema se deben explicar todos los conceptos necesarios para entender las tecnologías, herramientas y lenguajes de programación que se utilizan durante todo el transcurso del proyecto.

3.1. Introducción

En este capítulo de la memoria se definen, más concretamente, los servicios que se pretenden monitorizar y las herramientas de monitorización utilizadas, analizando las alternativas respecto a las soluciones estudiadas. También se especifican los servidores Web y bases de datos implementados. Finalmente se detallan los lenguajes de programación utilizados.

3.2. Servicios a monitorizar

A continuación se explican las características de los servicios que ofrece el servidor Linux y que servirán como base para la posterior implementación de las herramientas de monitorización.

3.2.1. Servidor de acceso remoto

El servidor de acceso remoto es *Open SSH*:

- Utilidad: permite acceder por línea de comandos al servidor de manera segura utilizando el propio protocolo SSH. Utiliza técnicas para cifrar todos los datos que son transmitidos durante la sesión.
- Puerto: por defecto utiliza el 22 **TCP**.
- Demonio: sshd ubicado en "/usr/sbin/".
- Fichero de configuración: sshf_config ubicado en "/etc/ssh/".

No existen alternativas competentes a la utilización del protocolo SSH en las conexiones remotas. Sí se pueden encontrar otras herramientas que lo implementan como:

- *FreeNX* es una versión gratuita y open source de la empresa *NoMachine*. También utiliza el protocolo SSH y está enfocado a conexiones con entorno gráfico (**X11**) y escritorio remoto.

- *Neatx* es la alternativa open source al servidor comercial NX server de *NoMachine*.
- *VNC server* para realizar conexiones remotas de forma gráfica. Existen diferentes clientes VNC comerciales y libres pero para que la conexión sea segura se debe implementar la utilización del protocolo SSH en el servidor.
- Telnet es un protocolo que permite la conexión remota al servidor por terminal pero no implementa seguridad ni cifrado en sus conexiones.

Se utiza *OpenSSH* ya que:

- Es software libre y open source.
- Permite que las comunicaciones sean cifradas y, por tanto, seguras.
- Fácil de instalar y configurar.
- No es necesario implementar conexiones para entornos gráficos o escritorios remotos, como *VNC* o *NX server*, puesto que el servidor no tiene interface gráfica ni la requiere para el desarrollo del proyecto.

3.2.2. Servidores Web

Los servidores Web son *Apache*:

- Utilidad: permite alojar páginas en entornos Web soportando **HTTP** y **HTTPS** (seguro).
- Puertos: por defecto utiliza los puertos 80 TCP (HTTP) y 443 TCP (**SSL**).
- Demonio: apache2 ubicado en "/usr/sbin/".
- Ficheros de configuración: httpd.conf (configuración global) y ports.conf (puertos) ubicados en "/etc/apache2/".

Y *Apache Tomcat*:

- Utilidad: permite alojar páginas en entornos Web soportando servlets y **JSP** (Java Server Pages).
- Puertos: por defecto utiliza los puertos 8080 TCP (HTTP), 8443 TCP (HTTPS) y 8009 TCP (conector Jakarta AJP).
- Demonio: se gestiona por los scripts startup.sh (arranque) y shutdown.sh (parada) ubicados en "/usr/share/tomcat6/bin/".
- Fichero de configuración: server.xml ubicado en "/etc/tomcat6/".

Existen otras alternativas destacables como:

- *Miscrosoft Internet Information Services (**IIS**)* es software propietario de Microsoft para plataformas Windows.

Se utilizan *Apache* y *Tomcat* ya que:

- Son software libres y open source, por el contrario del *IIS*, que es software privado y se descarta por este motivo.
- Son ampliamente utilizados en el sector.
- Instalación básica fácil.
- Permiten implementar muchos complementos y funcionalidades.
- Son compatibles con otros software y tecnologías como Java, JSP, PHP, *MySQL*, etc.

3.2.3. Servidores de BBDD

Los servidores de bases de datos son *MySQL*:

- Utilidad: permite la gestión de BBDD relacionales, multihilo, multiusuario y con soporte multiplataforma. Utiliza el lenguaje estándar **SQL**.
- Puerto: por defecto utiliza el puerto 3306.
- Demonio: mysqld ubicado en "/usr/sbin/".
- Fichero de configuración: my.conf ubicado en "/etc/mysql".

Y *PostgreSQL*:

- Utilidad: permite la gestión de BBDD relacionales y utiliza el lenguaje SQL.
- Puerto: por defecto utiliza el puerto 5432 TCP.
- Demonio: postgres ubicado en "/usr/lib/postgresql/8.4/bin/".
- Ficheros de configuración: postgresql.conf (configuración global) y pg_hba.conf (conexiones) ubicados en "/etc/postgresql/8.4/main/".

Existen alternativas como:

- *Firebird* es menos pesada que *MySQL* o *PostgreSQL* y permite diseñar BBDD portables.
- *SQLite* es menos pesada que *MySQL* o *PostgreSQL.*
- *Apache Derby* está limitado a un entorno de máquina virtual en Java.
- *Microsoft SQL Server* es software privado de Microsoft para plataformas Windows.

- *Oracle* es software privado y se considera uno de los gestores de BBDD más completos.

Se utilizan MySQL y PostgreSQL ya que:

- Son software con licencia de distribución gratuita.
- Son los más extendidos en el sector, sin considerar los software privados como *Oracle* o *Miscrosoft SQL server*, que descartan por éste motivo.
- Poseen una documentación completa.
- Están capacitados para entornos de trabajo grandes, por el contrario de *SQLite* o *Firebird*.
- Tienen soporte multiplataforma, por el contrario de *Apache Derby*, que está limitado a entornos Java.

3.2.4. Servidor de DNS

El servidor de DNS es *Bind*:

- Utilidad: permite resolver las direcciones lógicas (**IP**) en nombres de dominios y viceversa.
- Puertos: por defecto utiliza el puerto 53 TCP (transferencias) y 53 **UDP** (consultas).
- Demonio: named ubicado en "/usr/sbin/".
- Fichero de configuración: named.conf ubicado en "/etc/bind".

Existen alternativas como:

- *DjbDNS* que combina la función de servidor y la de caché en un solo programa.
- *PowerDNS* que utiliza *MySQL* en vez de ficheros de datos.
- DNS de plataformas Windows que necesita un sistema Windows para implementarse.

Se utiliza *Bind* ya que:

- Es el servidor de DNS más utilizado en el sector para plataformas Linux, por el contrario de *DjbDNS*, *PowerDNS* u otros.
- Es open source y de distribución gratuita, por el contrario del servidor de DNS de Microsoft.
- Posee una documentación muy completa.

3.2.5. Servidor de correo

El servidor de correo es *Postfix*:

- Utilidad: permite el enrutamiento y envío de e-mails utilizando el protocolo **SMTP** (Simple Mail Transfer Protocol).
- Puertos: por defecto utiliza el puerto 25.
- Demonio: master ubicado en "/usr/lib/postfix/".
- Fichero de configuración: main.cf ubicado en "/etc/postfix/".

Existen alternativas como:

- *Sendmail* se considera el más completo y, a la vez, complejo de configurar.
- *Exim* se considera sencillo y fácil de utilizar.
- *Microsoft Exchange Server* es software privado de Microsoft para plataformas Windows.

Se utiliza *Postfix* ya que:

- Es open source y software libre, por el contrario de *Microsoft Exchange Server.*
- Es fácil de configurar, por el contrario de *Sendmail*.
- Es ampliamente utilizado en el sector, junto con *Sendmail*, y por contra de *Exim* u otros.

3.2.6. Servidor de antivirus

El servidor de antivirus es *ClamAV*:

- Utilidad: sistema antivirus de ficheros y correo electrónico.
- Puerto: no requiere de puertos propios.
- Demonios: freshclam ubicado en "/usr/bin/" y clamd ubicado en "/usr/sbin/".
- Ficheros de configuración: clamd.conf (global) y freshclam.conf (base de datos) ubicados en "/etc/clamav".

Se utiliza, también, *Amavis*:

- Utilidad: Permite la comunicación entre el servidor de correo y los filtrados de virus y spam.
- Puerto: por defecto utiliza el 10024 TCP.

- Demonio: amavisd.
- Ficheros de configuración: diferentes ficheros ubicados en "/etc/amavis/conf.d".

Se utilizan *ClamAV* y *Amavis* ya que:

- Poseen completa compatibilidad con *Postfix* siendo, además, open source y con licencia GPL y, por lo tanto, no se estudian otras alternativas.
- Tienen un gran soporte en desarrollo, actualizaciones y documentación por parte de usuarios y colaboradores del proyecto *ClamAV*.

3.2.7. Servidor de antispam

El servidor de antispam es *SpamAssassin*:

- Utilidad: sistema de filtro de spam en correos electrónicos.
- Puerto: por defecto utiliza el 783 TCP.
- Demonio: spamd ubicado en "/usr/sbin/spamd".
- Fichero de configuración: 65_debian.cf ubicado en "/etc/spamassassin".

Se utiliza *SpamAssassin* ya que:

- Posee completa compatibilidad con *Postfix* siendo, además, open source y con licencia GPL compatible (de Apache) y, por lo tanto, no se estudian otras alternativas.
- Es ampliamente utilizado en el sector y compatible con otras herramientas a nivel de seguridad como *ClamAV* y *Amavis*.

3.2.8. Servidor de archivos

El servidor de archivos es *Samba*:

- Utilidad: permite compartir archivos y dispositivos entre plataformas Linux y Windows.
- Puertos: por defecto utiliza los puertos 139 y 445 (TCP) y 137 (UDP).
- Demonios: smbd y nmbd ubicados en "/usr/sbin/".
- Fichero de configuración: smb.conf ubicado en "/etc/samba".

Se utiliza *Samba* ya que:

- No existen alternativas competentes open source y de distribución gratuita que cumplan éstas funcionalidades.

3.3. Herramientas de monitorización

Una vez detallados los servicios a monitorizar, se describen las herramientas de monitorización utilizadas.

3.3.1. Monitorización de servicios

La herramienta para monitorizar servicios es *Nagios*.

Características:

- Tiene una arquitectura cliente-servidor mediante chequeos a los recursos (con agentes) y servicios (sin agentes).
- Permite el seguimiento de los servicios de red (SMTP, POP3, HTTP, NNTP, PING, etc).
- Controla los recursos de host (carga del procesador, espacio ocupado en, etc).
- Permite diseñar plugins, o complementos, para desarrollar chequeos personalizados. No están limitados a ningún lenguaje de programación específico.
- Permite controles de servicio paralelizado.
- Posibilita definir la jerarquía y dependencias entre los objetos.
- Notifica de alertas cuando se producen problemas y cuando se resuelven vía e-mail, sms, o el método definido por el usuario.
- Posibilita definir manejadores de eventos que se ejecuten durante determinados períodos de servicio para auto resolver problemas.
- Contempla la rotación automática de archivos de registro en logs.
- Permite asignación de roles a los usuarios.
- Utiliza plantillas en los ficheros de configuración del sistema y monitores.
- Incluye la interfaz de red, opcional, para ver el estado actual de la monitorización, hosts, servicios, notificaciones, alertas, históricos o gráficas.

Requerimientos principales:

- Servidor Web, preferiblemente *Apache*, para la interfaz de monitorización.

- Compilador **GCC** y librerías **GD**.

Demonio:

- Nagios3 ubicado en "/usr/sbin/".

Ficheros de configuración:

- Nagios3.conf ubicado en "/etc/nagios3/". Es el fichero principal de configuración.
- Command.cfg ubicado en "/etc/nagios3/conf.d". Se definen los comandos de chequeo.
- Contacts.cfg ubicado en "/etc/nagios3/conf.d". Se definen los contactos.
- Timeperiods.cfg ubicado en "/etc/nagios3/conf.d". Se definen los tiempos de ejecución.
- Templates.cfg ubicado en "/etc/nagios3/conf.d". Se definen las plantillas.

Alternativas:

- Se descartan las herramientas *God* o *Monit* ya que no son soluciones completas para entornos corporativos medianamente grandes al no disponer de todas las funcionalidades requeridas, como interfaces web o notificaciones a SMS.
- *Osmius*, *PandoraFMS*, *Zabbix* y *Zenoss* son, cualquiera de ellas, herramientas completas y válidas para integrarlas en el sistema de monitorización.

Se utiliza *Nagios* ya que:

- Es una solución madura, puesto que se empieza a desarrollar en 1999, a diferencia de soluciones más jóvenes como *Osmius* (2008), *Zenoss* (2006), *PandoraFMS* (2005).
- Es ampliamente utilizada en el sector.
- Tiene abundante documentación a todos los niveles, instalación, configuración y plugins.
- Aunque una instalación básica de *Nagios* no completaría todas las funcionalidades requeridas, su principal punto a favor es la modularidad. Añadiendo e implementado plugins se puede llegar a obtener una monitorización completa y en la medida de las necesidades.

- Existen herramientas que tienen control total vía interface web para desarrollar todo el sistema de monitorización en éste entorno, como por ejemplo, *Zabbix*. No obstante, se considera preferible utilizar *Nagios*, configurando el sistema desde su propia estructura interna, puesto que se requiere para entender realmente cómo funciona la solución y como se comunica con todos los elementos. Además, existen plugins en *Nagios* que permiten añadir una gestión web completa al sistema, previamente instalado.

3.3.2. Monitorización de gráficas

Para la generación de gráficas sobre los rendimientos de los servicios se planteaba, inicialmente, la combinación de un sistema de monitorización de servicios y otro, encargado de mostrar gráficamente los resultados, además de soportar diferentes implementaciones y configuraciones.

Como soluciones, de las estudiadas en éste ámbito, se destacan:

- *Cacti*
- *JFFNMS*
- *Munin*

Las 3 son herramientas actualizadas y completas que permiten almacenar datos controlados por los monitores y generar gráficas personalizables para analizar rendimientos o evoluciones de servicios, servidores, puertos, sensores, etc. No obstante, carecen de plena funcionalidad referente a la notificación de alertas.

La pieza fundamental que los une es la RRDTool:

- Es una herramienta que utiliza el método Round Robin, un algoritmo complejo muy utilizado en el ámbito de los sistemas operativos y las redes.
- Permite almacenar y representar datos en intervalos temporales.
- Implementa bases de datos circulares, con la que las bases de datos no aumentan de tamaño con el tiempo, puesto que siempre trabaja con el mismo número de datos.

Pero el sistema de monitorización *Nagios,* como ya se ha mencionado anteriormente, posibilita la implementación de plugins para ampliar las

funcionalidades en la medida de las necesidades. Y, finalmente, para monitorizar gráficas de rendimientos se utiliza *PNP4Nagios*.

Características:

- Es un complemento para *Nagios*.
- Utiliza RRDTool.
- Analiza los datos de rendimiento obtenidos por los plugins de *Nagios* y los almacena automáticamente en bases de datos RDD.
- Posee una instalación fácil y un mantenimiento mínimo.

Requerimientos principales:

- RRDTool.
- *Nagios*.
- Servidor web *Apache*.

Demonio:

- npcd ubicado en "/usr/local/pnp4nagios/bin/".

Fichero de configuración:

- npcd.cfg ubicado en "/usr/local/pnp4nagios/etc/".

Se utiliza *PNP4Nagios* ya que:

- Pese a que existen herramientas potentes a nivel de reportes y gráficas, como *Cacti*, *JFFNMS* o *Munin*, se ha considerado utilizar *PNP4Nagios* ya que es un complemento desarrollado para *Nagios* que ofrece mejor compatibilidad en un mismo entorno y cumple con las funcionalidades requeridas igual que las otras herramientas estudiadas a éste nivel.

3.4. Servidor Web

El servidor web utilizado para el entorno de monitorización es *Apache*. Ya se han detallado las características del mismo y las alternativas existentes en el sector (en "Fundamentos teóricos. Servicios a monitorizar. Servidores Web"), puesto que también es un servicio a monitorizar en el propio sistema.

En el sistema de monitorización, *Apache* permite:

- Atender a peticiones que usan los protocolos HTTP y HTTPS en el entorno de web de *Nagios* y *PNP4Nagios*.
- Gestionar sus páginas tanto con contenidos estáticos como dinámicos.
- Soportar las tecnologías web HTML y **CGI** (método para generar contenido dinámico) del frontal web de *Nagios*.
- Soportar la tecnología PHP del frontal web de *PNP4Nagios*.

La estructura de directorios que tiene se basa en los siguientes archivos:

- Fichero de configuración principal: apache2.conf en "/etc/apache2/".
- Fichero de configuración de la comunicación: ports.conf en "/etc/apache2/".
- Ficheros de configuración de los sitios web:
  - Apache2.conf en "/etc/nagios3/" o nagios3.conf en "/etc/apache2/conf.d/".
  - Pnp4nagios.conf en "/etc/apache2/conf.d/".
- Localización de los sitios web:
  - "/usr/share/nagios3/htdocs/" para *Nagios*.
  - "/usr/local/pnp4nagios/share/" para *PNP4Nagios*.
- **URL** de los sitios web:
  - "http://<servidor_ip>/nagios3/".
  - "http://<servidor_ip>/pnp4nagios/".

  Donde <servidor_ip> es la dirección lógica IP del servidor Linux.

3.5. Servidor de bases de datos

El servidor de BBDD utilizado para el entorno de monitorización es *MySQL.* Ya se han detallado las características del mismo y las alternativas existentes en el sector (en "Fundamentos teóricos. Servicios a monitorizar. Servidores de BBDD") puesto que también es un servicio a monitorizar en el propio sistema.

En el sistema de monitorización, *MySQL* permite:

- Almacenar toda la información de los ficheros de configuración y eventos de *Nagios* en una base de datos SQL.
- Disponer de la información almacenada para que otros plugins de *Nagios* puedan acceder a ella de manera eficiente y segura.

Inicialmente *Nagios* no opera con bases de datos. Para implementar ésta funcionalidad se utiliza *NDOutils.* Es un complemento de *Nagios* que consta de 4 componentes:

- NDOMOD (Event Broker Module): Exporta los datos generados por *Nagios*.
- LOG2NDO: Exporta ficheros de logs.
- FILE2SOCK: Envía el contenido de los ficheros vía socket TCP o unix.
- NDO2DB: Recoge la salida de NDOMOD y LOG2NDO y los almacena en la base de datos *MySQL*.

3.6. Lenguajes de programación

Existen diferentes lenguajes de programación utilizados en el desarrollo e implementación del sistema de monitorización.

3.6.1. Lenguajes Web

El lenguaje de programación Web predominante en el entorno de monitorización es el HTML.

Sus características son:

- Significa HyperText Markup Language (lenguaje de marcas hipertextuales).
- Desarrollado por el World Wide Web Consortium.
- Utiliza etiquetas, marcas de hipertexto e hipervínculos.

- Permite implementar scripts y otros lenguajes, como el PHP, utilizando de base el HTML.
- Los ficheros HTML tienen extensión .htm o .html.

Como inconvenientes, se pueden destacar:

- Es un lenguaje estático.
- Es soportado por todos los navegadores, aunque puede visualizar resultados diferentes en la muestra de la página web según el explorador y versión.
- Es un lenguaje limitado por las posibilidades de las etiquetas.

Como segundo lenguaje de programación Web utilizado está el PHP.

Sus características son:

- Significa PHP Hypertext Pre-processor.
- Es un lenguaje dinámico y orientado a objetos.
- Desarrollado por PHP Group.
- Es multiplataforma.
- No requiere de compilación para su ejecución.
- Puede trabajar sobre el lenguaje HTML.
- Los ficheros PHP tienen la extensión .php.

Inconvenientes:

- Requiere de un servidor Web que implemente las librerías PHP.
- Con un número grande de peticiones puede resultar ineficiente.
- La estructura y comprensión del código, si existen muchas líneas o se implementan varios lenguajes, puede resultar difícil.

3.6.2. Lenguajes de BBDD

El lenguaje de bases de datos es el SQL.

Sus características son:

- Significa Structured Query Language (lenguaje de consulta estructurado).
- Es un lenguaje declarativo de bases de datos relacionales.
- Permite gran variedad de operaciones con los datos.

- Es un estándar por su amplio uso en el sector y se ha ido adaptando con los años para cumplir con los estándares requeridos.

Inconvenientes:

- El orden de la sentencia afecta significativamente a la eficiencia en el resultado.
- En ocasiones requiere la implementación de índices para acelerar las consultas, aunque repercuten en la actualización de los datos.

3.6.3. Lenguajes de aplicación

El lenguaje en el que se ha desarrollado la aplicación de monitorización implementada, internamente, es el C.

Sus características son:

- Desarrollado por los laboratorios Bell.
- Dispone de estructuras de alto nivel pero permite, también, un control a bajo nivel, pudiendo acceder directamente a memoria o a periféricos.
- Permite gran variedad de funcionalidades e implementación de librerías.

Inconvenientes:

- No dispone de una gestión nativa de memoria.
- No soporta programación orientada a objetos.
- No permite encapsulación.
- No dispone de soporte nativo para programación multihilo.

4. Análisis

Una vez explicados todos los conceptos necesarios para el desarrollo y entendimiento del proyecto es el momento de especificar el análisis y diseño del sistema de monitorización.

4.1. Introducción

En este capítulo se detalla el análisis y diseño del sistema. Se verifica el cumplimiento de los requerimientos funcionales y no funcionales. Se detallan los casos de uso. Seguidamente, se especifica el diseño general y completo del sistema, de la base de datos y de la interfaz Web.

4.2. Requerimientos funcionales

Se verifican los requerimientos funcionales estipulados:

- Monitorizar los servicios del servidor desde un entorno Web: controlar los servicios mediante el estado visual de los monitores configurados.

Se utiliza la interfaz Web de *Nagios*, como muestra la siguiente captura:

Host	Service	Status	Last Check	Duration	Attempt	Status Information
NeKoTrAyA	PING	OK	22-03-2010 15:01:52	4d 22h 44m 6s	1/4	ECO OK - Paquetes perdidos = 0%, RTA = 6.53 ms
PDA	PING	OK	22-03-2010 15:01:52	0d 1h 8m 17s	1/4	ECO OK - Paquetes perdidos = 0%, RTA = 74.92 ms
cliente	PING	OK	22-03-2010 15:01:52	4d 22h 44m 35s	1/4	ECO OK - Paquetes perdidos = 0%, RTA = 14.34 ms
router	PING	OK	22-03-2010 15:01:52	4d 22h 44m 34s	1/4	ECO OK - Paquetes perdidos = 0%, RTA = 8.30 ms
servidor	APT	CRITICAL	22-03-2010 15:01:52	0d 0h 30m 4s	4/4	CRITICAL - Plugin timed out after 10 seconds
	APT-DISTUPGRADE	CRITICAL	22-03-2010 15:01:52	0d 0h 30m 4s	4/4	CRITICAL - Plugin timed out after 10 seconds
	Apache	OK	22-03-2010 15:01:52	0d 0h 31m 37s	1/4	HTTP OK: HTTP/1.1 200 OK - 452 bytes in 0,001 second response time
	Apache Tomcat	OK	22-03-2010 15:01:53	0d 0h 31m 38s	1/4	HTTP OK: HTTP/1.1 200 OK - 2134 bytes in 0,055 second response time
	ClamAV - BBDD Virus	WARNING	22-03-2010 15:01:52	0d 0h 6m 42s	4/4	ClamAV WARNING: daily.cvd 10606 out of date by 1 revision
	Current Load	CRITICAL	22-03-2010 15:01:52	0d 0h 44m 51s	4/4	CRITICAL - carga media: 3.72, 4.03, 4.14
	Current Users	OK	22-03-2010 15:02:05	18d 20h 14m 46s	1/4	USERS OK - 1 users currently logged in
	DNS - www.google.com	OK	22-03-2010 15:02:05	17d 19h 31m 13s	1/4	DNS ACEPTAR: 0,080 segundos de tiempo de respuesta. www.google.c
	Disk Space	OK	22-03-2010 15:02:05	18d 20h 12m 46s	1/4	DISK OK
	MySQL	OK	22-03-2010 15:02:05	4d 23h 49m 5s	1/4	Uptime: 15667 Threads: 2 Questions: 97339 Slow queries: 0 Opens: 233
	PING	OK	22-03-2010 15:01:52	4d 23h 44m 54s	1/4	ECO OK - Paquetes perdidos = 0%, RTA = 0.10 ms
	Postfix - Cola	OK	22-03-2010 15:01:52	17d 19h 27m 12s	1/4	OK: mailq reports queue is empty
	Postfix - SMTP	OK	22-03-2010 15:01:52	4d 23h 48m 14s	1/4	SMTP OK - 0,002 sec. response time
	PostgreSQL	OK	22-03-2010 15:01:52	4d 23h 46m 1s	1/4	OK - base de datos postgres (0 seg.)
	SSH	OK	22-03-2010 15:02:05	4d 23h 45m 48s	1/4	SSH OK - OpenSSH_5.1p1 Debian-6ubuntu2 (protocolo 2.0)
	Samba	OK	22-03-2010 15:01:52	4d 23h 44m 39s	1/4	192.168.1.10 servidor<00>
	Total Processes	OK	22-03-2010 15:01:52	18d 20h 12m 16s	1/4	PROCS ACEPTAR: 127 procesos

Ilustración 3: Ejemplo de monitorización de servicios en Nagios

Se puede observar la columna "Status" que muestra el estado de los monitores de cada uno de los servicios de la columna "Service".

- Notificar las alertas o errores en los monitores: enviar avisos mediante mensajes a móvil y correo electrónico de los monitores cuando el estado de los mismos indique que existe un error.

En el sistema de monitorización se ha implementado la notificación de alertas a móvil y sms y, se puede visualizar en la interfaz Web de *Nagios*, como muestra la captura siguiente:

Host	Service	Type	Time	Contact	Notification Command	Information
servidor	ClamAV - BBDD Virus	WARNING	22-03-2010 14:57:52	root	notify-service-by-email	ClamAV WARNING: daily.cvd 10606 out of date by 1 revision
servidor	ClamAV - BBDD Virus	WARNING	22-03-2010 14:57:52	root	notify-service-by-sms	ClamAV WARNING: daily.cvd 10606 out of date by 1 revision
servidor	APT	CRITICAL	22-03-2010 14:33:13	root	notify-service-by-email	CRITICAL - Plugin timed out after 10 seconds
servidor	APT	CRITICAL	22-03-2010 14:33:13	root	notify-service-by-sms	CRITICAL - Plugin timed out after 10 seconds
servidor	APT-DISTUPGRADE	CRITICAL	22-03-2010 14:33:13	root	notify-service-by-email	CRITICAL - Plugin timed out after 10 seconds
servidor	APT-DISTUPGRADE	CRITICAL	22-03-2010 14:33:13	root	notify-service-by-sms	CRITICAL - Plugin timed out after 10 seconds
servidor	APT	WARNING	22-03-2010 14:32:23	root	notify-service-by-email	APT WARNING: 18 packages available for upgrade (0 critical updates).
servidor	APT	WARNING	22-03-2010 14:32:23	root	notify-service-by-sms	APT WARNING: 18 packages available for upgrade (0 critical updates).
servidor	APT-DISTUPGRADE	WARNING	22-03-2010 14:32:22	root	notify-service-by-email	APT WARNING: 24 packages available for dist-upgrade (0 critical updates).
servidor	APT-DISTUPGRADE	WARNING	22-03-2010 14:32:22	root	notify-service-by-sms	APT WARNING: 24 packages available for dist-upgrade (0 critical updates).
servidor	Apache Tomcat	OK	22-03-2010 14:31:37	root	notify-service-by-email	HTTP OK: HTTP/1.1 200 OK - 2134 bytes in 0,843 second response time
servidor	Apache Tomcat	OK	22-03-2010 14:31:37	root	notify-service-by-sms	HTTP OK: HTTP/1.1 200 OK - 2134 bytes in 0,843 second response time
servidor	Apache	OK	22-03-2010 14:31:15	root	notify-service-by-email	HTTP OK: HTTP/1.1 200 OK - 452 bytes in 0,043 second response time
servidor	Apache	OK	22-03-2010 14:31:15	root	notify-service-by-sms	HTTP OK: HTTP/1.1 200 OK - 452 bytes in 0,043 second response time
servidor	Apache Tomcat	CRITICAL	22-03-2010 14:30:37	root	notify-service-by-email	ConexiÃ³n rechazada
servidor	Apache Tomcat	CRITICAL	22-03-2010 14:30:37	root	notify-service-by-sms	ConexiÃ³n rechazada
servidor	Apache	CRITICAL	22-03-2010 14:30:35	root	notify-service-by-email	ConexiÃ³n rechazada
servidor	Apache	CRITICAL	22-03-2010 14:30:35	root	notify-service-by-sms	ConexiÃ³n rechazada
servidor	Current Load	CRITICAL	22-03-2010 14:18:04	root	notify-service-by-email	CRITICAL - carga media: 4.46, 4.26, 4.02
servidor	Current Load	CRITICAL	22-03-2010 14:18:04	root	notify-service-by-sms	CRITICAL - carga media: 4.46, 4.26, 4.02
servidor	Apache	OK	22-03-2010 14:17:35	root	notify-service-by-email	HTTP OK: HTTP/1.1 200 OK - 452 bytes in 0,015 second response time
servidor	Apache	OK	22-03-2010 14:17:35	root	notify-service-by-sms	HTTP OK: HTTP/1.1 200 OK - 452 bytes in 0,015 second response time
servidor	Current Load	WARNING	22-03-2010 14:17:14	root	notify-service-by-email	WARNING - carga media: 4.22, 4.23, 4.00
servidor	Current Load	WARNING	22-03-2010 14:17:14	root	notify-service-by-sms	WARNING - carga media: 4.22, 4.23, 4.00
servidor	Current Load	CRITICAL	22-03-2010 14:17:06	root	notify-service-by-email	CRITICAL - carga media: 4.49, 4.27, 4.01
servidor	Current Load	CRITICAL	22-03-2010 14:17:06	root	notify-service-by-sms	CRITICAL - carga media: 4.49, 4.27, 4.01
servidor	Current Load	WARNING	22-03-2010 13:55:44	root	notify-service-by-email	WARNING - carga media: 4.26, 4.11, 3.18
servidor	Current Load	WARNING	22-03-2010 13:55:44	root	notify-service-by-sms	WARNING - carga media: 4.26, 4.11, 3.18
servidor	Apache	CRITICAL	22-03-2010 13:44:49	root	notify-service-by-email	ConexiÃ³n rechazada
servidor	Apache	CRITICAL	22-03-2010 13:44:49	root	notify-service-by-sms	ConexiÃ³n rechazada
servidor	APT-DISTUPGRADE	CRITICAL	22-03-2010 13:44:14	root	notify-service-by-email	CRITICAL - Plugin timed out after 10 seconds
servidor	APT-DISTUPGRADE	CRITICAL	22-03-2010 13:44:14	root	notify-service-by-sms	CRITICAL - Plugin timed out after 10 seconds
servidor	APT	CRITICAL	22-03-2010 13:44:13	root	notify-service-by-email	CRITICAL - Plugin timed out after 10 seconds
servidor	APT	CRITICAL	22-03-2010 13:44:13	root	notify-service-by-sms	CRITICAL - Plugin timed out after 10 seconds
servidor	Apache Tomcat	OK	22-03-2010 13:43:52	root	notify-service-by-email	HTTP OK: HTTP/1.1 200 OK - 2134 bytes in 0,859 second response time
servidor	Apache Tomcat	OK	22-03-2010 13:43:52	root	notify-service-by-sms	HTTP OK: HTTP/1.1 200 OK - 2134 bytes in 0,859 second response time
servidor	Apache Tomcat	CRITICAL	22-03-2010 13:43:11	root	notify-service-by-email	ConexiÃ³n rechazada
servidor	Apache Tomcat	CRITICAL	22-03-2010 13:43:11	root	notify-service-by-sms	ConexiÃ³n rechazada
servidor	Apache	OK	22-03-2010 13:18:00	root	notify-service-by-email	HTTP OK: HTTP/1.1 200 OK - 452 bytes in 0,002 second response time
servidor	Apache	OK	22-03-2010 13:18:00	root	notify-service-by-sms	HTTP OK: HTTP/1.1 200 OK - 452 bytes in 0,002 second response time
servidor	Apache Tomcat	OK	22-03-2010 13:00:45	root	notify-service-by-email	HTTP OK: HTTP/1.1 200 OK - 2134 bytes in 0,295 second response time
servidor	Apache Tomcat	OK	22-03-2010 13:00:45	root	notify-service-by-sms	HTTP OK: HTTP/1.1 200 OK - 2134 bytes in 0,295 second response time
servidor	Apache	CRITICAL	22-03-2010 12:59:25	root	notify-service-by-email	ConexiÃ³n rechazada
servidor	Apache	CRITICAL	22-03-2010 12:59:25	root	notify-service-by-sms	ConexiÃ³n rechazada

Ilustración 4: Ejemplo de notificaciones de Nagios

La columna "Notification Command" muestra las notificaciones generadas relacionadas con los estados de cada monitor, indicados en la columna "Type".

- Auto recuperar servicios no operativos: levantar automáticamente los servicios que se han venido abajo cuando el estado de los monitores así lo indique.

En el sistema de monitorización se ha implementado la auto recuperación de servicios y se puede visualizar en la interfaz Web de *Nagios* como muestra la siguiente captura, donde se ha subrayado la información que, posteriormente, se explica:

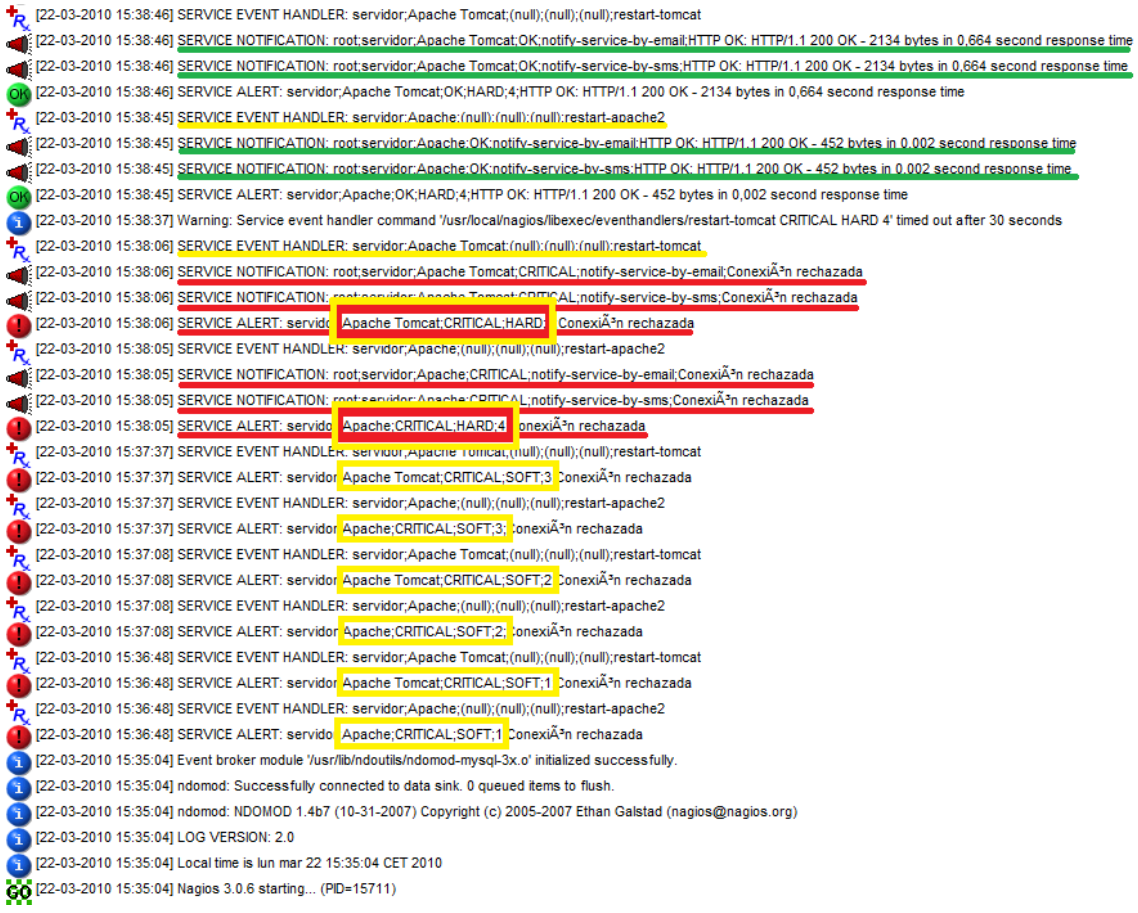


Ilustración 5: Ejemplo del log de eventos de Nagios

Se observa que los monitores de *Apache* y *Apache Tomcat* comienzan a fallar. Empieza con los valores "CRITICAL", "SOFT" y "1". En el fallo nº 4 le llegan las variables "CRITICAL" y "HARD" y el servicio "EVENT HANDLER" ejecuta el script que reinicia los servicios *Apache* y *Apache Tomcat*. Finalmente se observa que las alertas de servicio "SERVICE ALERT" dan valores correctos y las notificaciones de servicio "SERVICE NOTIFICATION" indican "OK".

- Generar gráficas y estadísticas de los monitores: controlar gráficamente el estado y rendimiento de los servicios controlados por los monitores.

Se utiliza *PNP4Nagios* integrado la propia interfaz Web de *Nagios* como se observa en la siguiente captura:

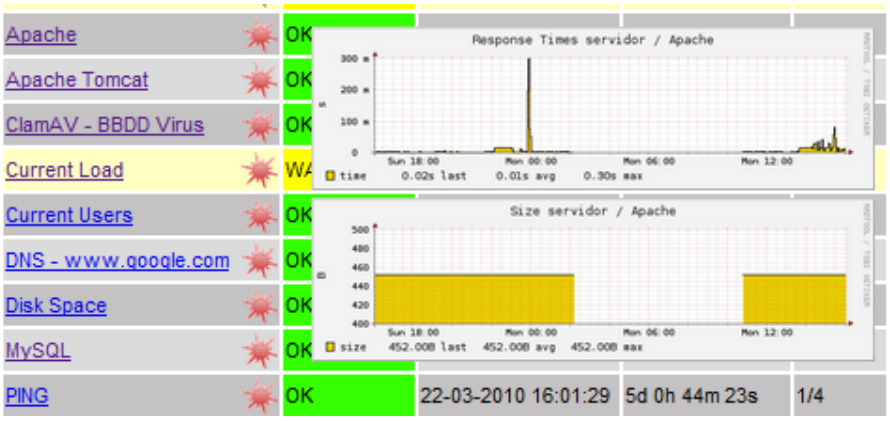


Ilustración 6: Ejemplo de Pop-Up de PNP4Nagios integrado en Nagios

También se puede acceder directamente a la interfaz Web de *PNP4Nagios*:

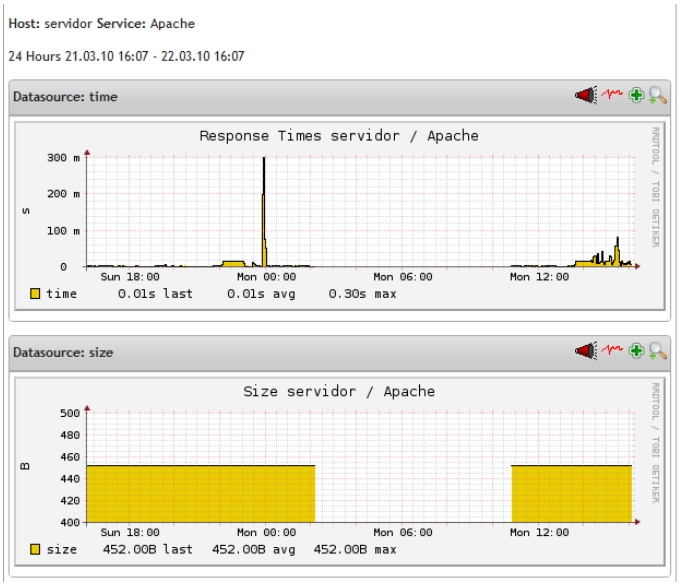


Ilustración 7: Ejemplo de gráficas en PNP4Nagios

4.3. Requerimientos no funcionales

Se verifican los requerimientos no funcionales estipulados:

- El sistema debe solicitar un usuario y contraseña para acceder al entorno de monitorización.
- El sistema debe satisfacer una gestión completa del entorno monitorizado, siempre sujeta a los permisos del usuario que accede.

La herramienta *Nagios* solicita usuario y contraseña para acceder a la interfaz Web y permite definir diferentes permisos a cada usuario. La siguiente captura muestra la pantalla de login:

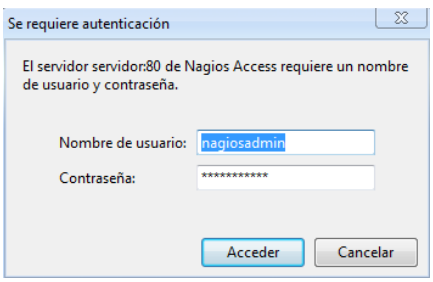


Ilustración 8: Login de Nagios

- La gestión de los monitores y todas las configuraciones relacionadas debe ser intuitiva y de fácil uso para los usuarios.
- El sistema debe ser estable y seguro para poder implementarse en cualquier entorno corporativo que lo requiera.
- El sistema debe ser escalable, siendo apto para un mayor número de servidores, servicios y clientes y sin sufrir ninguna pérdida de eficiencia.
- El acceso y gestión en el entorno de monitorización debe ser rápido y efectivo.

4.4. Casos de uso

La siguiente imagen muestra el diagrama de casos de uso del usuario administrador:

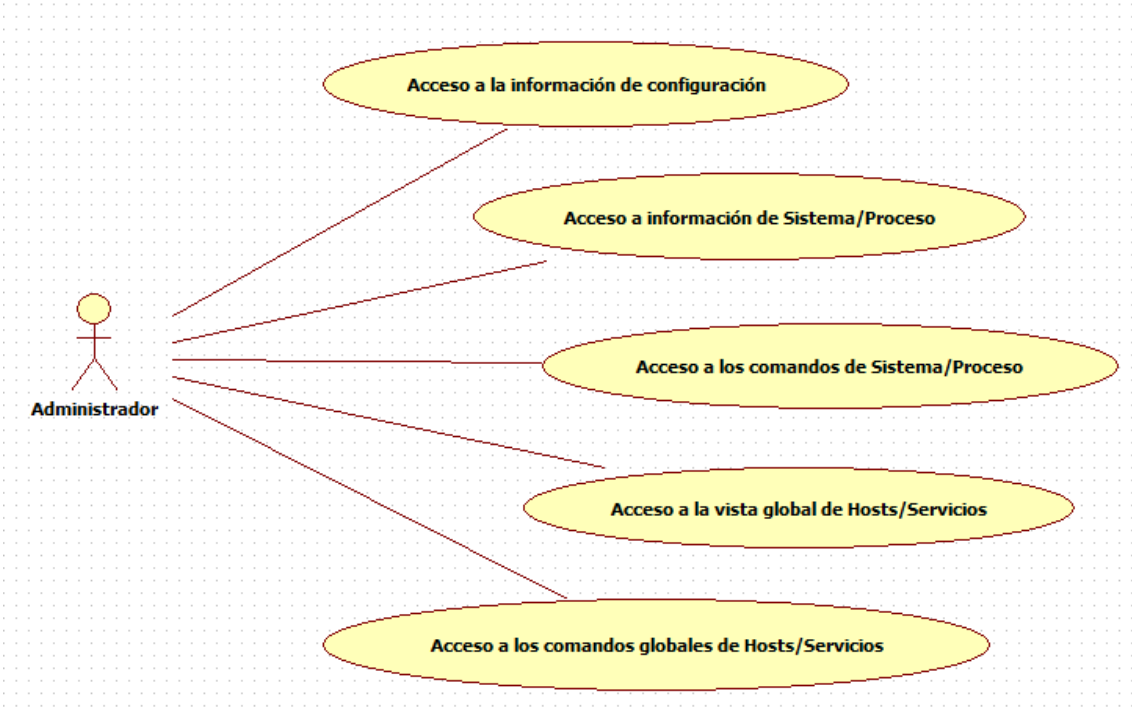


Ilustración 9: Diagrama de casos de uso del Administrador

El usuario "Administrador" tiene acceso total a todas las funcionalidades de las que dispone la interfaz Web de Nagios. Son las siguientes:

- Acceso a la información de configuración.
- Acceso a información de Sistema/Proceso.
- Acceso a los comandos de Sistema/Proceso.
- Acceso a la vista global de Hosts/Servicios.
- Acceso a los comandos globales de Hosts/Servicios.

En la siguiente imagen se muestra el diagrama correspondiente a un usuario que solo tiene permisos para acceder a la visión global de los hosts y servicios configurados en el sistema, pero sin posibilidad de acceder a la información de configuración, sistema, procesos y comandos.

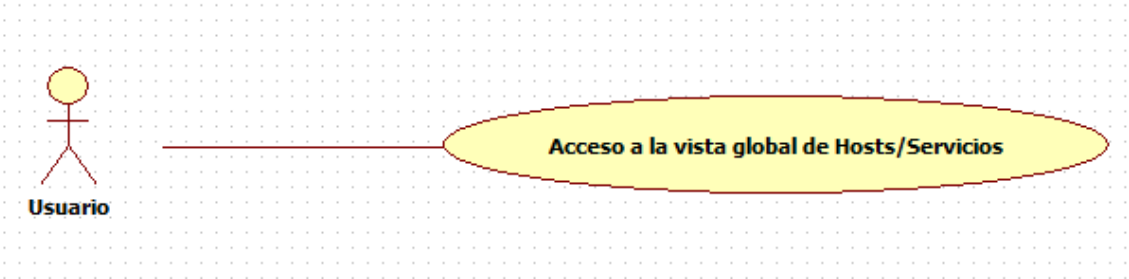


Ilustración 10: Diagrama de casos de uso de un Usuario

4.5. Módulos del sistema

El siguiente diagrama muestra el funcionamiento de *Nagios* y lo que representa cada módulo indicado:

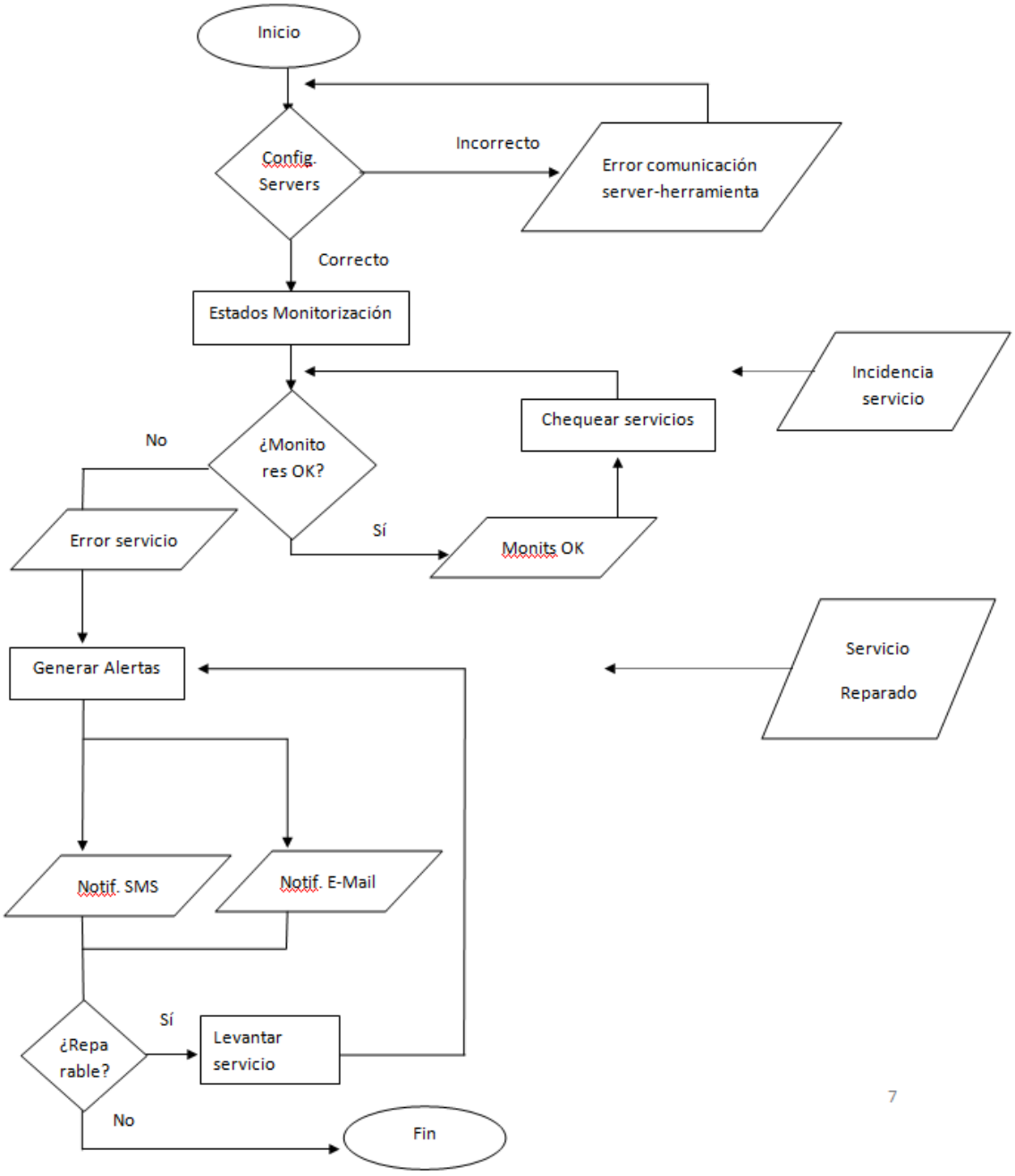


Ilustración 11: Diseño de los módulos del sistema

Existen diferentes módulos del sistema encargados de cumplir ciertas funciones:

- Estados de monitorización: representa el estado visual de los monitores controlados por el sistema. En el sistema de monitorización se aplica al frontal Web de *Nagios*.
- Chequear servicios: es el encargado de controlar el estado de los servicios controlados por los monitores. En el sistema de monitorización se aplica a los comandos y parámetros de chequeo definidos para cada servicio y a todo el conjunto de *Nagios-Plugins*.
- Generar alertas: cuando la monitorización no es correcta se notifican las incidencias vía SMS y correo electrónico. Cuando la monitorización que genera alarmas se vuelve la correcta, al solucionarse la incidencia, se notifica también de ello. En el sistema de monitorización se aplica a comandos y parámetros definidos en los ficheros de configuración. Para las alertas por correo electrónico se requiere de una configuración específica en el servidor de correo *Postfix* y una cuenta externa de correo creada para éste fin. Para las alertas vía SMS se requiere de una configuración específica de la herramienta *Gnokii* y el dispositivo móvil GSM utilizado.
- Levantar servicio: es el encargado de levantar servicios de monitores que tienen configurado el reinicio del servicio que controlan de forma automática cuando el estado del monitor así lo indica. En el sistema de monitorización se aplica al manejador de eventos y a los comandos y parámetros relacionados con el mismo.

Todas las configuraciones e implementaciones de los módulos y funcionalidades del sistema de monitorización se detallan en el siguiente capítulo de esta memoria: "Implementación".

4.6. Diseño del sistema

La siguiente imagen muestra la estructura general del sistema a nivel de red, servidor, componentes y servicios:

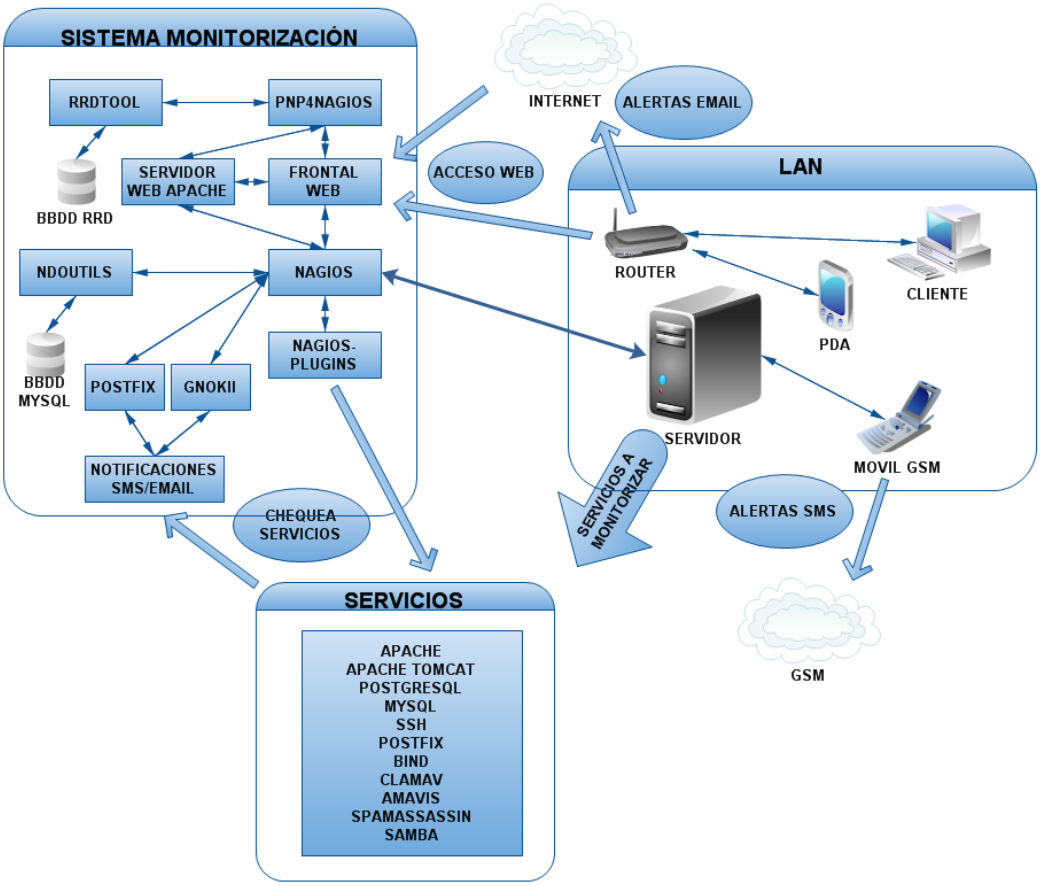


Ilustración 12: Diseño del sistema de monitorización, la LAN y los servicios a monitorizar.

A continuación se explican y amplían todos los elementos de la anterior imagen separados en tres bloques.

4.6.1. Entorno en la red de área local (LAN)

La LAN consta de:

- Un ordenador cliente conectado a la red por Ethernet. También se monitoriza su estado en la red.
- Una PDA conectada vía Wi-Fi. También se monitoriza su estado en la red.
- Un móvil GSM para la notificación de alertas vía SMS.

- El servidor que incorpora el sistema de monitorización y los servicios monitorizados.
- Un router que permite la red LAN y la conexión a internet, necesaria para la notificación de alertas vía email y para los servicios que ofrecen acceso desde Internet o el propio frontal Web de *Nagios*.

4.6.2. El sistema de monitorización

El sistema de monitorización consta de:

- La herramienta de monitorización *Nagios*.

Dispone de los siguientes ficheros de configuración:

- Fichero de configuración principal (Main Config File): contiene las directivas en las que se basa el demonio de *Nagios*.
- Fichero de recursos (Resource File): contiene macros y rutas definidas.
- Ficheros de definición de objetos (Object Definition Files): son los ficheros donde se definen todos los servicios, grupos, contactos, comandos, etc implementados en el sistema.
- Fichero de configuración CGI (CGI Config File): contiene las directivas referentes al CGI (Common Gateway Interface) para la programación dinámica a nivel de Web.

La siguiente imagen muestra la relación entre éstos elementos y el demonio de *Nagios*:

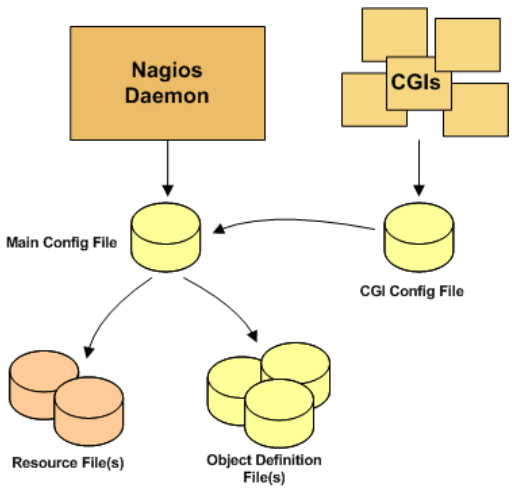


Ilustración 13: Configuración de Nagios

El fichero de configuración CGI lee la información relacionada del fichero de configuración principal. Los ficheros de recursos y definición de objetos son leídos por el fichero de configuración principal.

- El complemento *Nagios-Plugins*.

Los plugins son scripts o ejecutables que se lanzan desde un comando para chequear el estado de un host o servicio. *Nagios* ejecuta el plugin según el estado del monitor que tenga configurado y realiza las tareas de notificación de alertas y gestión de eventos relacionadas.

Como se puede observar en la siguiente imagen, los plugins actúan como una capa de abstracción entre la monitorización lógica del sistema y los hosts y servicios reales monitorizados:

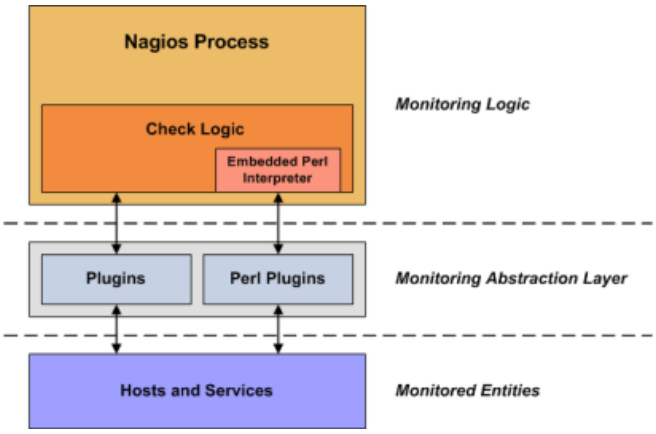


Ilustración 14: Plugins de Nagios

- La utilidad *NDOUtils*.

La siguiente imagen muestra el esquema general de NDOUtils:

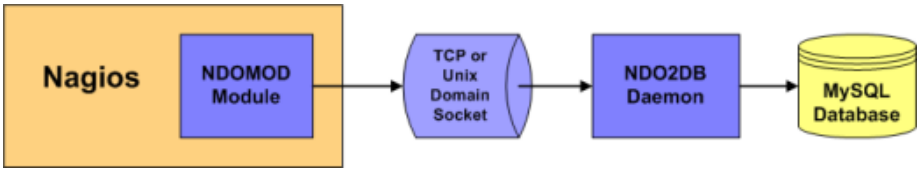


Ilustración 15: NDOUtils en Nagios

A partir de un socket TCP o Unix, el módulo "NDOMOD" le pasa la información al demonio "NDO2DB", que la almacena en una base de datos *MySQL*.

- El servidor de BBDD *MySQL*.

En el fichero de configuración de *Nagios* "NAGIOS.CFG" se define el módulo "NDOMOD" encargado de traspasar los datos del sistema utilizando el demonio "NDO2DB". Para ello utiliza los ficheros de configuración de *NDOUTILS* "NDO2DB.CFG", que contiene las credenciales para la conexión a la base de datos *MySQL* y "NDOMOD.CFG" que contiene información sobre el buffer, socket y puerto para establecer dicha conexión.

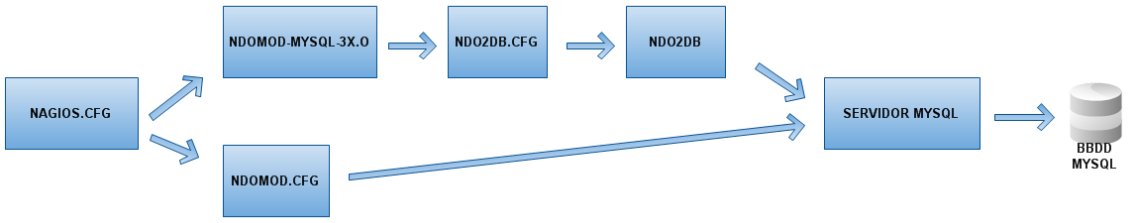


Ilustración 16: Servidor MySQL configurado con NDOUtils

- El complemento *PNP4Nagios* que implementa la *RRDTool*.

El fichero de configuración principal de *Nagios* y el fichero de comandos contienen los parámetros y definiciones necesarias. El sistema trabaja con el fichero "process_perfdata.pl" que se comunica con la *RRDTool* para generar los ficheros .xml y .rrd correspondientes a cada servicio y que confeccionan la base de datos RRD que implementa la herramienta, como se observa en la siguiente imagen:

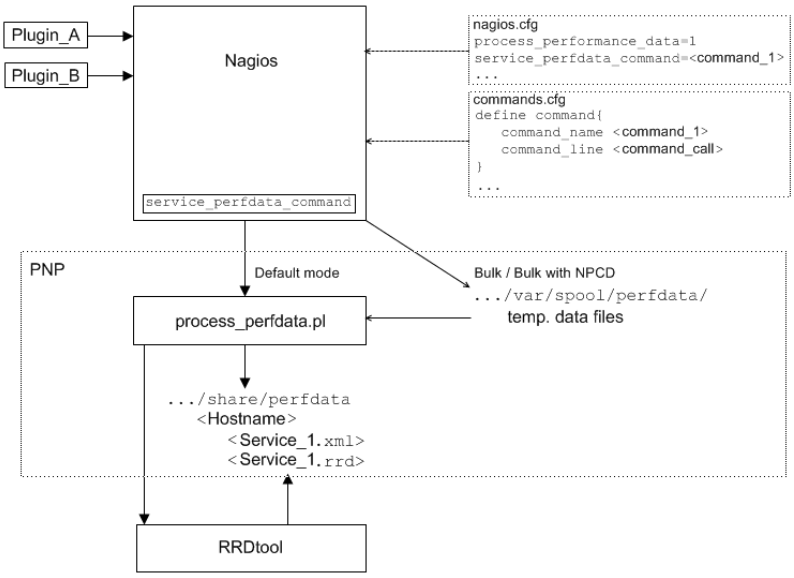


Ilustración 17: PNP4Nagios y RRDTool en Nagios

- El servidor Web *Apache.*

El servidor *Apache* implementa la tecnología PHP necesaria para los frontales Web del sistema de monitorización. Dispone de un directorio "CONF.D" que contiene los ficheros "PNP4NAGIOS.CONF" y "NAGIOS3.CONF" y que apuntan, respectivamente, a los propios contenidos Web de *PNP4Nagios* y *Nagios*. Desde cualquier dispositivo con navegador Web de la red LAN es posible acceder a ambos frontales: "http://servidor/PNP4Nagios/" y "http://servidor/Nagios3/" donde "servidor" es el nombre del propio servidor Linux, substituible por la IP privada correspondiente al mismo. Para el acceso externo a la red LAN "servidor" tiene que substituirse por la IP pública asignada por el proveedor de Internet. La siguiente imagen muestra la estructura explicada:

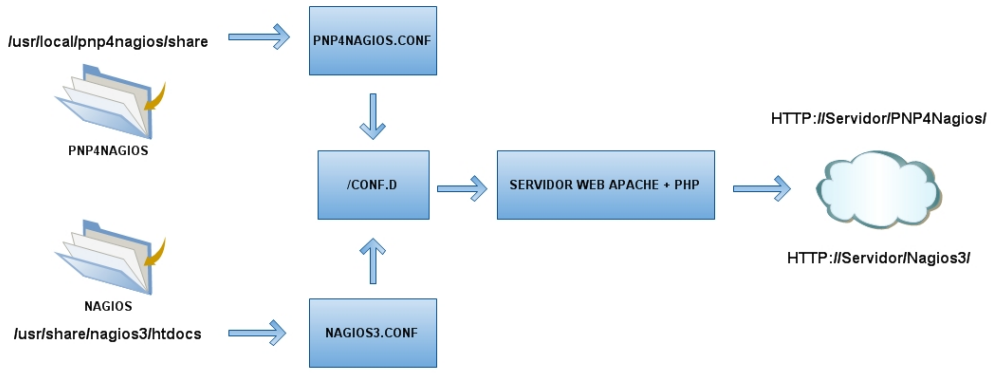


Ilustración 18: Servidor Apache configurado para Nagios y PNP4Nagios

- El servidor de correo *Postfix*.

El servidor *Postfix* dispone de los ficheros "MAIN.CF", donde se especifica el servidor SMTP, "GENERIC", donde se relaciona la cuenta de correo local del servidor con la cuenta externa utilizada para el envío de notificaciones y "SASL_PASSWD", donde se definen las credenciales de conexión a dicha cuenta. En el sistema *Nagios* se definen los comandos y parámetros necesarios para las notificaciones vía email en el fichero "COMANDOS.CFG" y se estipulan las direcciones de correo electrónico de los receptores de las alertas en el fichero "CONTACTOS.CFG". La siguiente imagen muestra el esquema de éstos ficheros en relación al servidor de correo *Postfix*:

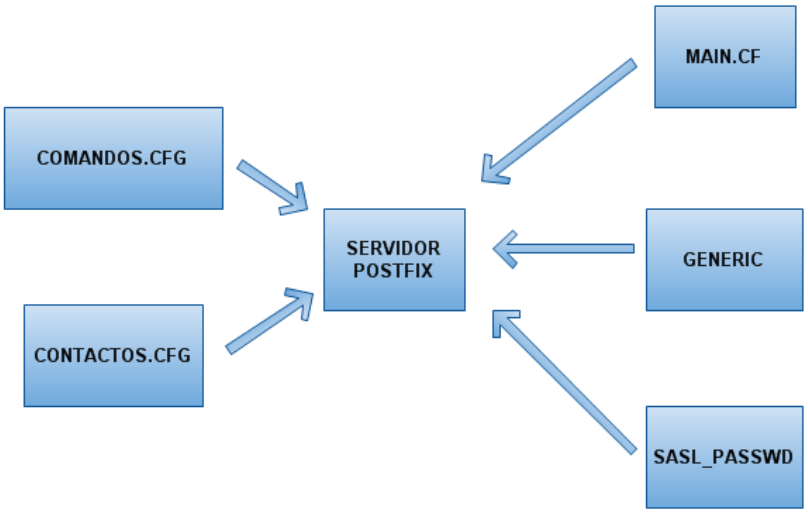


Ilustración 19: Servidor de correo Postfix configurado para Nagios

- La herramienta *Gnokii*.

La herramienta *Gnokii* dispone del fichero "GNOKIIRC" donde se especifican los datos de conexión al dispositivo GSM como el puerto y el modelo. En el sistema *Nagios* se definen los comandos y parámetros necesarios para las notificaciones vía SMS en el fichero "COMANDOS.CFG" y se estipulan los números de teléfono de los receptores de las alertas en el fichero "CONTACOS.CFG". La siguiente imagen muestra el esquema de éstos ficheros en relación la herramienta *Gnokii*:

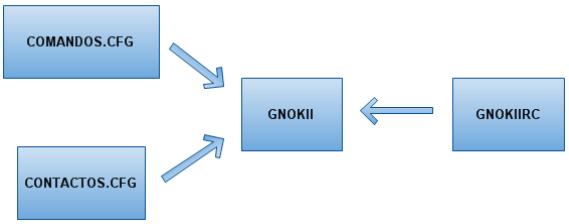


Ilustración 20: Gnokii configurado para Nagios

La siguiente imagen muestra el esquema general de procesos y ficheros del sistema de monitorización:

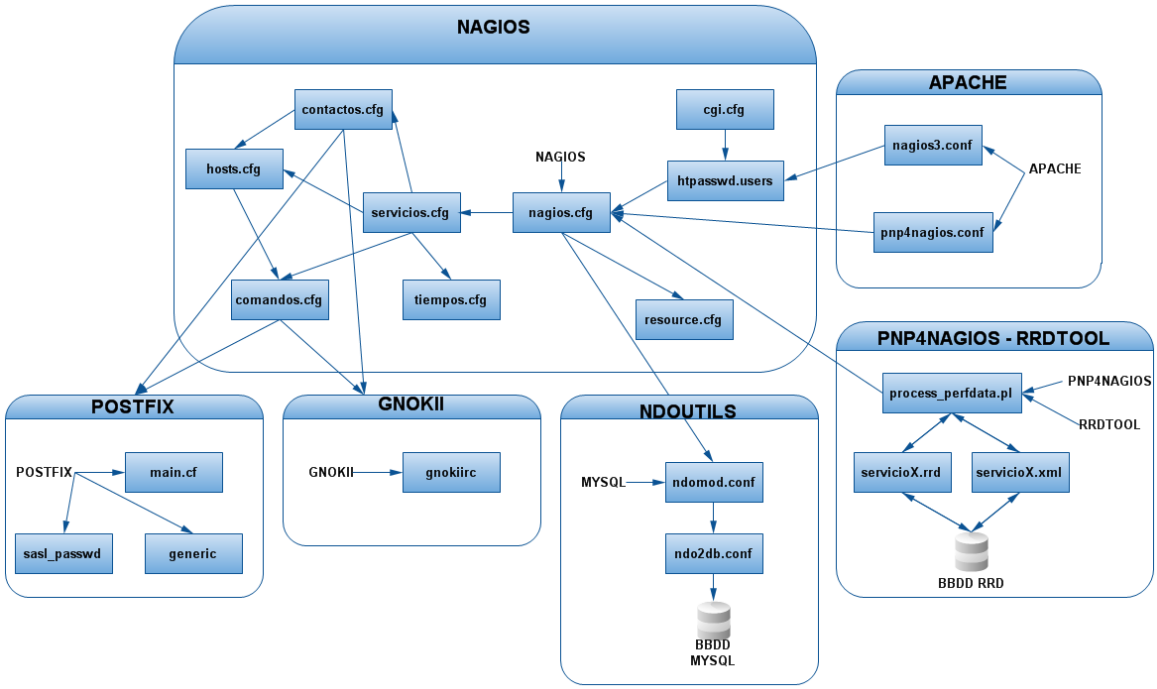


Ilustración 21: Estructura de procesos y ficheros del sistema

Referente al sistema *Nagios* se trabaja, concretamente, con los ficheros siguientes:

- NAGIOS.CFG: configuración principal.
- CGI.CFG: directivas CGI.
- HTPASSWD.USERS: usuarios del entorno Web de *Nagios.*
- RESOURCE.CFG: configuración de los recursos.
- SERVICIOS.CFG: configuración de los servicios monitorizados.
- HOSTS.CFG: configuración de las máquinas monitorizadas.
- CONTACTOS.CFG: configuración de los contactos.
- COMANDOS.CFG: configuración de los comandos y notificaciones.
- TIEMPOS.CFG: configuración de los intervalos de tiempo de chequeos, notificaciones y eventos.

El sistema *Nagios* se comunica tanto con *Postfix* como con *Gnokii* utilizando los ficheros "CONTACTOS.CFG" y "COMANDOS.CFG".

Para la comunicación con *NDOUTILS* y *PNP4Nagios* se utiliza el fichero de configuración principal "NAGIOS.CFG".

Para la comunicación con *Apache* se utiliza el fichero de configuración principal "NAGIOS.CFG" y "HTPASSWD.USERS".

4.6.3. Los servicios monitorizados

Por un lado están los servicios instalados en el servidor con el fin de ser monitorizados. Además, el sistema de monitorización también controla la comunicación con el router, con el ordenador cliente y con la PDA y monitoriza el estado en la red de cada uno.

La siguiente tabla muestra los servicios monitorizados y, entre ellos, los que se requieren en la propia arquitectura del sistema (se marcan con una "X"). También se detallan las máquinas y dispositivos a los que hacen referencia:

Servicios	Solo monitorización	Propios de la arquitectura	Dispositivos
Apache		X	Servidor
Apache Tomcat	X		
PostgreSQL	X		
MySQL		X	
SSH	X		
Postfix		X	
Bind	X		
ClamAV	X		
Amavis	X		
Spamassassin	X		
Samba	X		
Comunicación Servidor	X		
Comunicación Cliente	X		Cliente
Comunicación PDA	X		PDA
Comunicación Router	X		Router

Tabla 6: Relación de los servicios y dispositivos monitorizados

4.7. Diseño de la base de datos

El almacenamiento de los datos en MySQL lo realiza, como ya se ha explicado, la utilidad NDOUtils. No obstante, en éste apartado, se describen las tablas más importantes y sus relaciones. No se entra en la descripción detallada de cada campo y tabla puesto que lo que se pretende es entender su estructura general por si requiere su uso en otros plugins de *Nagios* que necesiten trabajar con datos almacenados en una base de datos relacional, como es el caso de *NagVis*, un complemento que permite generar mapas de red del sistema monitorizado.

4.7.1. Tablas principales

Las tablas principales de la base de datos MySQL son las siguientes:

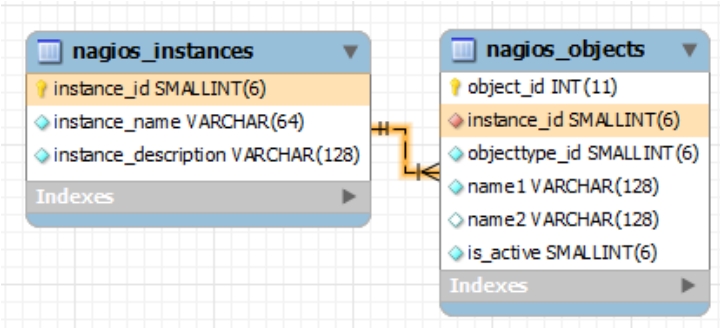


Ilustración 22: Tablas principales

- Nagios_instances
- Nagios_objects

La tabla "nagios_instances" contiene información sobre el número de instancias que existen en el sistema. La tabla "nagios_ objects" almacena la información de todos los elementos del sistema, tratados como objetos.

Están relacionadas de manera que "instance_id" de "nagios_instances" es la clave foránea de "nagios_ objects".

4.7.2. Tabla de debug

La tabla que almacena datos relacionados con el debug del sistema es la siguiente:

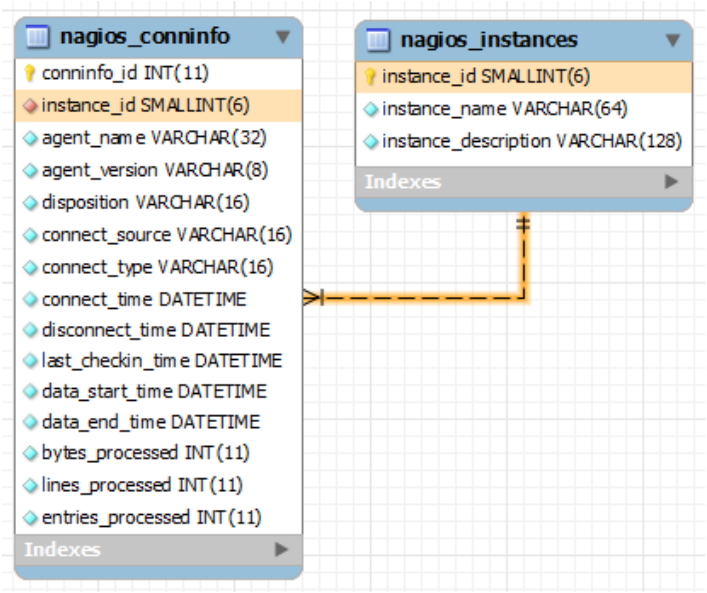


Ilustración 23: Tabla de debug

- Nagios_conninfo

La tabla "nagios_conninfo" almacena información sobre el debug del demonio "NDO2DB" y los agentes como "NDOMOD" referentes, ambos, a NDOUTILS.

Están relacionadas de manera que "instance_id" de "nagios_instances" es la clave foránea de "nagios_conninfo".

4.7.3. Tablas de históricos

Las tablas que almacenan los datos relacionados con los históricos del sistema son las siguientes:

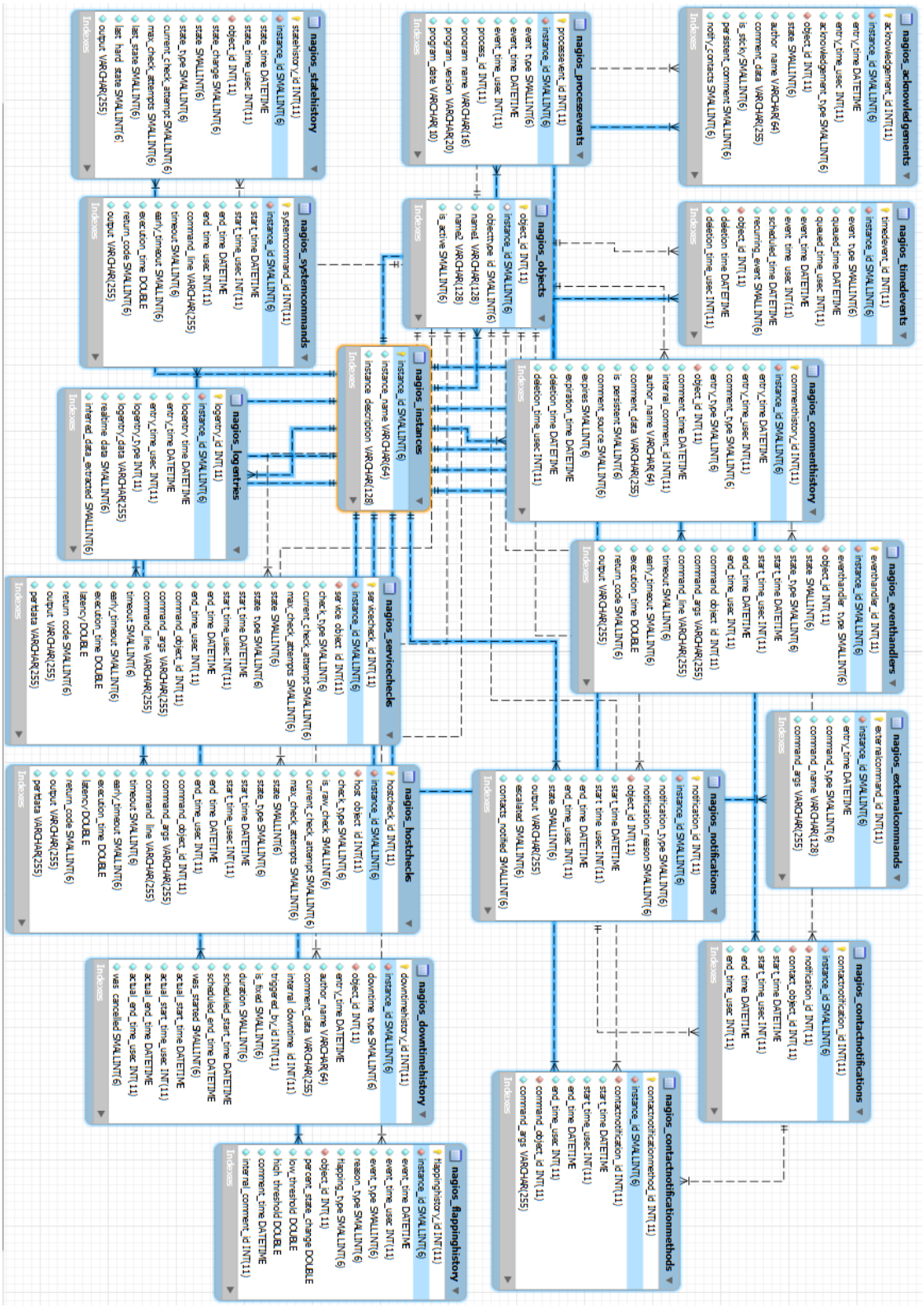


Ilustración 24: Tablas de históricos

- Nagios_acknowledgements
- Nagios_commenthistory
- Nagios_contactnotifications
- Nagios_downtimehistory
- Nagios_eventhandlers
- Nagios_externalcommands
- Nagios_flappinghistory
- Nagios_hostchecks
- Nagios_logentries
- Nagios_notifications
- Nagios_processevents
- Nagios_servicechecks
- Nagios_statehistory
- Nagios_systemcommands
- Nagios_timedevents

Están relacionadas de manera que "instance_id" de "nagios_instances" es la clave foránea del resto de tablas mostradas. Varias de las tablas también contienen el campo "object_id" (u otro) que es clave foránea de "object_id" de la tabla "nagios_objects".

4.7.4. Tablas de estados

Las tablas que almacenan los datos relacionados con los estados del sistema son las siguientes:

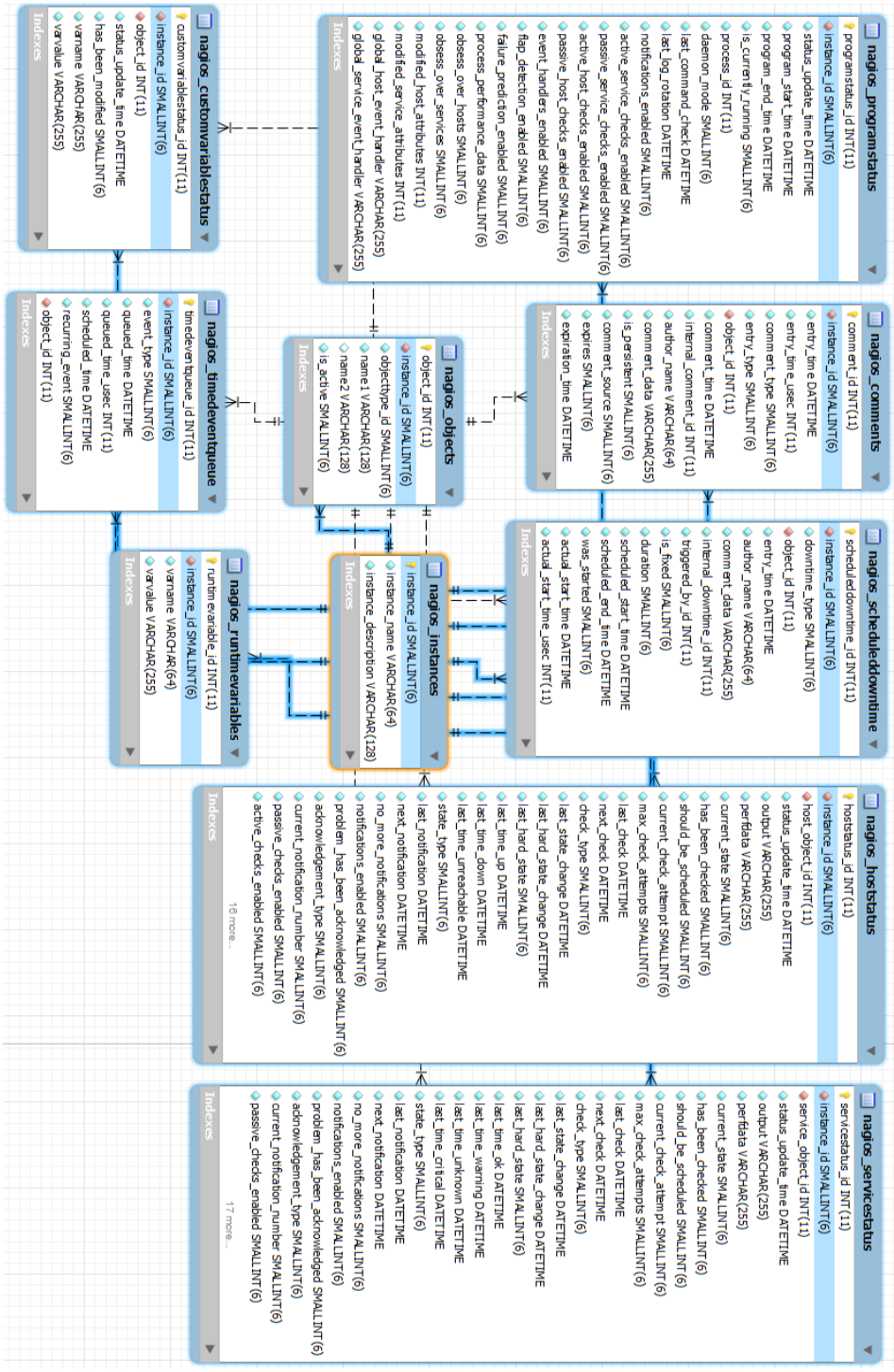


Ilustración 25: Tablas de estados

- Nagios_comments
- Nagios_customvariablestatus
- Nagios_hoststatus
- Nagios_programstatus
- Nagios_runtimevariables
- Nagios_scheduleddowntime
- Nagios_servicestatus
- Nagios_timedeventqueue

Están relacionadas de manera que "instance_id" de "nagios_instances" es la clave foránea del resto de tablas mostradas. Varias de las tablas también contienen el campo "object_id" (u otro) que es clave foránea de "object_id" de la tabla "nagios_objects".

4.7.5. Tablas de configuración

Existen diferentes tablas que utiliza el sistema para almacenar su configuración. Dichas tablas no alteran el funcionamiento del sistema. Tan solo contienen la última configuración utilizada. Por este motivo y por la cantidad elevada de tablas y la poca claridad en la que se mostraría el mapa relacional de las mismas, se expresa solamente el listado de ellas:

- Nagios_commands
- Nagios_configfiles
- Nagios_configfilevariables
- Nagios_contact_addresses
- Nagios_contact_notificationcommands
- Nagios_contactgroup_members
- Nagios_contactgroups
- Nagios_contactnotificationmethods
- Nagios_contacts
- Nagios_customobjectvariables
- Nagios_host_contactgroups
- Nagios_host_parenthosts
- Nagios_hostdependencies
- Nagios_hostescalation_contactgroups

- Nagios_hostescalations
- Nagios_hostgroup_members
- Nagios_hostgroups
- Nagios_hosts
- Nagios_service_contactgroups
- Nagios_servicedependencies
- Nagios_serviceescalation_contactgroups
- Nagios_serviceescalations
- Nagios_servicegroup_members
- Nagios_servicegroups
- Nagios_services
- Nagios_timeperod_timeranges
- Nagios_timeperiods

4.8. Diseño de la interfaz Web

Aunque en la interfaz Web de *Nagios* se han implementado las gráficas de *PNP4Nagios*, incrustadas como Pop-Ups, se detallan ambos frontales por separado.

4.8.1. Nagios

La interfaz Web de *Nagios* utiliza la siguiente estructura de ficheros:

El directorio físico de la Web es:

- /usr/share/nagios3/htdocs
    - Los directorios y ficheros que contiene son:
        - /contexthelp (contenido de ayuda)
        - /docs (documentación)
        - /images (imágenes)
        - /media (multimedia)
        - /ssi (personalización del CGI)
        - Index.html (fichero índice .html)
        - Main.html (fichero contenido .html)
        - Robots.txt (fichero meta)
        - Side.html (fichero contenido lateral .html)

La visualización del frontal en un navegador se rige por la siguiente forma:

Sidebar	Content

Tabla 7: Estructura frontal Web Nagios

Y un ejemplo del aspecto real:

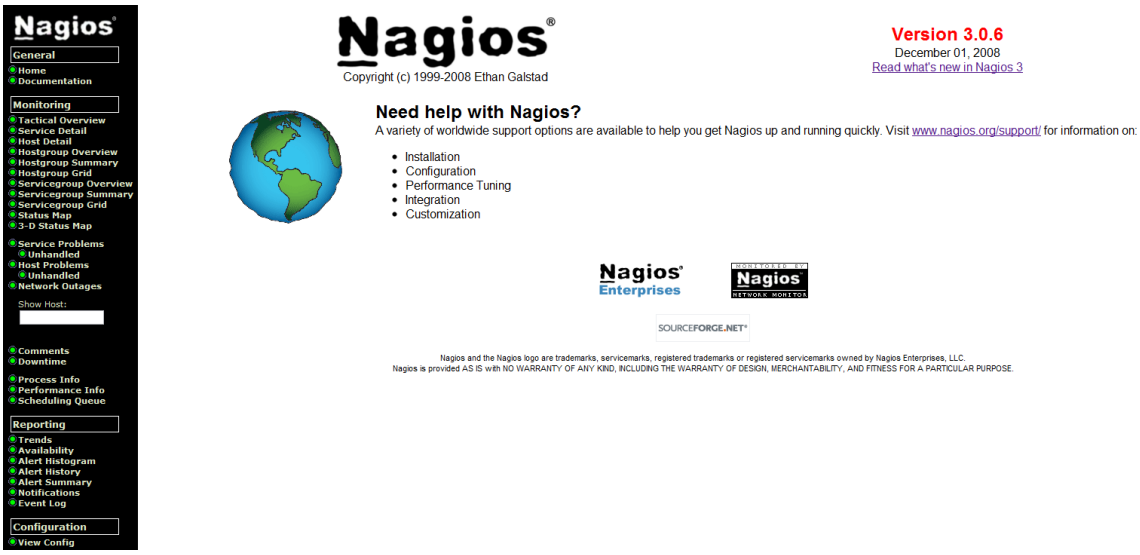


Ilustración 26: Página de inicio de Nagios

Funciones:

- Solicita loguin de acceso.
- En la parte izquierda se encuentra la columna lateral con todas las posibilidades que ofrece el sistema:
  - General:
    - Home: mostrar página de inicio del entorno Web.
    - Documentation: mostrar documentación.
  - Monitoring:
    - Tactical Overview: vista general.
    - Service Detail: detalle de los servicios.

- Host Detail: detalle de los hosts.
- Hostgroup Overview: vista de grupos de hosts.
- Hostgroup Summary: resumen de grupos de hosts.
- Hostgroup Grid: parrilla de grupos de hosts.
- Servicegroup Overview: vista de grupos de servicios.
- Servicegroup Summary: resumen de grupos de servicios.
- Servicegroup Grid: parrilla de grupos de servicios.
- Status Map: muestra mapa de la red monitorizada.
- 3-D Status Map: descargar mapa como archivo .cgi.
- Service Problems: problemas en servicios.
  - Unhandled: no gestionado.
- Host Problems: problemas en hosts.
  - Unhandled: no gestionado.
- Network Outages: muestra bloqueos o cortes.
- Show Host: encontrar un host realizando una búsqueda.
- Comments: muestra los comentarios.
- Downtime: muestra el scripts de tiempo configurados.
- Process Info: muestra información de proceso.
- Performance Info: muestra información de métricas y estadíscas en relación a los parámetros configurados.
- Scheduling Queue: muestra planificación programada de chequeos de monitores.

- Reporting:
  - Trends: configurar informes.
  - Availability: muestra el tiempo de actividad o inactvidad.
  - Alert Histogram: muestra información de un host o servicio representada gráficamente con un histograma.
  - Alert History: muestra el historial de alertas.
  - Alert Summary: generar informes de alertas.
  - Notifications: muestra las notificaciones.
  - Event Log: muestra el historial de eventos.
- Configuration
  - View Config: muestra configuración de cualquier objeto del sistema.

4.8.2. PNP4Nagios

La interfaz Web de *PNP4Nagios* utiliza la siguiente estructura de ficheros:

El directorio físico de la Web es:

- /usr/local/pnp4nagios/share
    - Los directorios y ficheros que contiene son:
        - /application (aplicación)
        - /media (multimedia)
        - /templates (plantillas)
        - /templates.dist (plantillas ya incluidas)
        - Index.php (fichero índice .php)

La visualización del frontal en un navegador se rige por la siguiente forma:

Content	Sidebar

Tabla 8: Estructura frontal Web PNP4Nagios

Y un ejemplo del aspecto real:

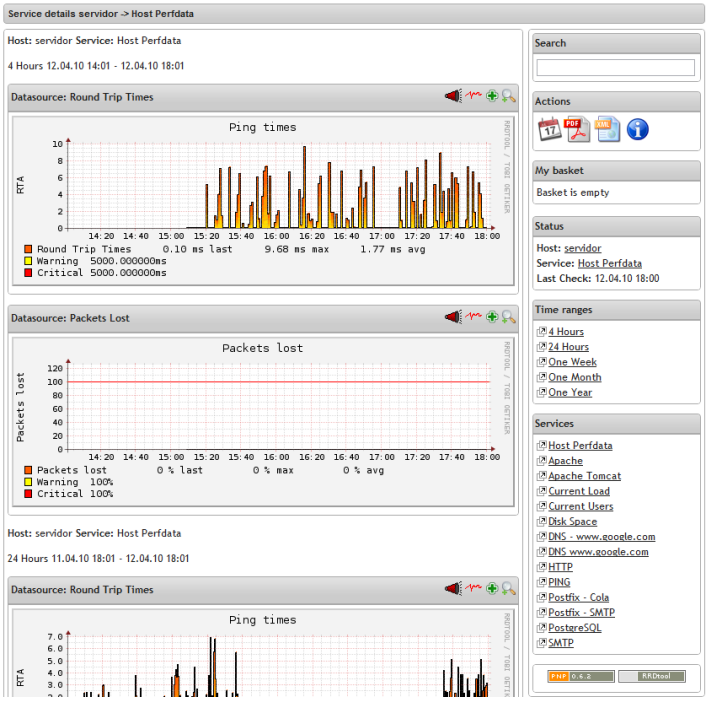


Ilustración 27: Visualización de PNP4Nagios

Funciones:

- Solicita el login de acceso, que es el mismo que el de Nagios. No lo solicita en caso de que ya se haya logueado anteriormente para acceder a Nagios.
- En la parte derecha se encuentra la columna lateral con todas las posibilidades que ofrece el sistema:
  - Search: realizar búsquedas en la Web.
  - Actions: definir rangos de tiempo para representar los gráficos de los monitores, generar a pdf o a xml y visualizar las estadísticas internas del "process_perfdata.pl".
  - Status: mostrar el host, servicio y último chequeo de los monitores que se están representando gráficamente.
  - Time ranges: definir los rangos horarios en los que se representan los datos en los gráficos.
  - Services: listar los diferentes servicios del host actual que permiten ser representados gráficamente.

5. Implementación

Una vez descrito el análisis y diseño de todo el sistema es el momento de detallar las implementaciones realizadas para cubrir todas las funcionalidades requeridas.

5.1. Introducción

En este capítulo se describen las implementaciones que se han llevado a cabo en el sistema.

5.2. Servicios a monitorizar

A continuación se explican las implementaciones iniciales de los servicios a monitorizar en relación a las instalaciones y configuraciones básicas.

5.2.1. OpenSSH

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: openssh-server.

5.2.2. Apache

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: apache2.

5.2.3. Tomcat

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: tomcat6.

5.2.4. MySQL

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: mysql-server-5.1.
- Creación del usuario "root" con permisos totales.

- Para el acceso remoto se modifica el fichero "my.conf" y se comenta la línea "#bind-address = 127.0.0.1".

5.2.5. PostgreSQL

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: postgresql-8.4.
- Creación del usuario "postgres".
- Para el acceso remoto se modifica el fichero "postgresql.conf" y se modifica la línea "listen_addresses = '*'".
- También se modifica el fichero "pg_hba.conf" con la línea "host all all 0.0.0.0 0.0.0.0 md5".

5.2.6. Bind

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: bind9.

5.2.7. Postfix

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: postfix.
- Renombrar fichero "main.cf.debian" a "main.cf".

5.2.8. ClamAV

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: clamav-daemon y clamav.

5.2.9. Amavis

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: amavisd-new.

5.2.10. Spamassassin

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: spamassassin.
- Modificar el fichero "spamassassin" con la línea "ENABLED=1".

5.2.11. Samba

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: samba.

5.3. Nagios

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: nagios3.
- Generar los ficheros de configuración determinados en el diseño del sistema.
- Definir los ficheros y directorios de configuración en "nagios.cfg".

5.4. Nagios-Plugins

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: nagios-plugins.
- Definir el directorio de plugins en "nagios.cfg".

5.5. PNP4Nagios

- Descarga de la herramienta desde la web de *PNP4Ngios*.
- Instalación de la herramienta a partir del código fuente.
- Definir los comandos para procesar los datos de rendimiento para PNP4Nagios en "nagios.cfg".
- Habilitar el proceso de recolección de datos para *PNP4Nagios* en "nagios.cfg".
- Activación de los pop-ups de los servicios en los monitores de *Nagios* en el fichero "servicio_generico.cfg".
- Activación de los pop-ups de los hosts en los monitores de *Nagios* en el fichero "host_genericos.cfg".

- Definición de la generación de gráficas en los monitores de servicios y hosts en el fichero "comandos.cfg".

A partir de éste punto, se crea la estructura de monitorización del sistema definiendo los hosts y servicios a monitorizar, chequeos, intervalos de chequeos, notificaciones, contactos y auto recuperaciones de servicios:

- Definición de los hosts y dependencias entre ellos en los ficheros "hosts.cfg" y "host_generico.cfg".
- Definición de los grupos de hosts en el fichero "grupos_hosts.cfg".
- Definición de los servicios en los ficheros "servicios.cfg" y "servico_generico.cfg".
- Enlazar los chequeos definidos con los plugins de nagios correspondientes.
- Crear los plugins necesarios no incorporados en el paquete "nagios-plugins".
- Definición de los contactos en el fichero "contactos.cfg".
- Definición de los comandos de notificación de alertas y receptores de las alertas en el fichero "contactos.cfg".
- Definición de los intervalos de chequeos en el fichero "tiempos.cfg".
- Definición de las notificaciones por email y SMS en el fichero "comandos.cfg".
- Definición de las auto recuperaciones de los servicios en el fichero "comandos.cfg".
- Configuración de los permisos de acceso de los usuarios al frontal web del sistema en el fichero "cgi.cfg".
- Enlazar rutas específicas del sistema a variables en el fichero "resource.cfg".

5.6. NDOUtils

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: ndoutils-nagios3-mysql.
- Definir la conexión con NDOutils para el volcado de datos en *MySQL* en el fichero "nagios.cfg".
- Definir socket y puerto en el fichero "ndomod.cfg".
- Definir credenciales de conexión en el fichero "ndo2db.cfg".

5.7. Apache

- Enlazar el directorio físico del frontal *Nagios* con el alias para el navegador y con los usuarios autorizados en el sistema en el fichero "apache2.conf".
- Enlazar el directorio físico del frontal *PNP4Nagios* con el alias para el navegador y con los usuarios autorizados en el sistema en el fichero "pnp4nagios.conf".

5.8. Postfix

- Definir el servidor SMTP utilizado en el fichero "main.cf".
- Definir el fichero "sasl_passwd" y "generic" en el fichero "main.cf".
- Definir credenciales de la cuenta externa utilizada en el fichero "sasl_passwd".
- Definir la relación entre la cuenta local del sistema y la cuenta externa en el fichero "generic".

5.9. Gnokii

- Instalación por defecto desde el "APT-GET INSTALL" de Ubuntu.
- Paquetes instalados: gnokii.
- Definir puerto, modelo del móvil utilizado y conexión en el fichero "gnokiirc".

5.10. Usuarios, grupos y permisos

5.10.1. Nagios y PNP4Nagios

El usuario "nagios" es el propio de la herramienta y ha de pertenecer al grupo "dialout", para disponer de permisos sobre el puerto serial al que se conecta el dispositivo móvil para las notificaciones SMS. La siguiente imagen muestra las líneas de los ficheros de usuarios y grupos del servidor llamados "passwd" y "group" respectivamente y ubicados, ambos, en "/etc".

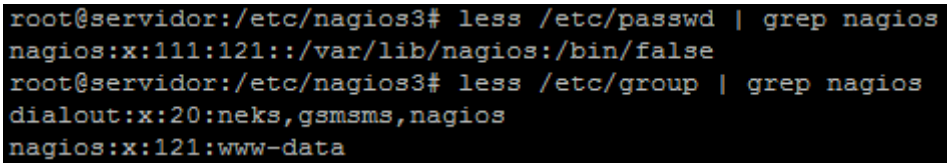


Ilustración 28: Usuario y grupo Nagios

El fichero "HTPASSWD.USERS" de *Nagios* muestra el contenido siguiente:

```
#usuario nagiosadmin, password encriptado

nagiosadmin:Dm7wyip81KgnA

#usuario usuario1, password encriptado

usuario1:Vycm8eE50dOTo
```

Además, es necesario que el usuario de *Apache* "www-data" disponga de permisos de lectura en los ficheros y directorios de *Nagios* y *PNP4Nagios*, para poder acceder desde el entorno Web a toda la información de monitorización y configuración.

5.10.2. Apache

El usuario "www-data" es de *Apache* y se requiere que pertenezca al grupo "www-data" y también al grupo "nagios". La siguiente imagen muestra esta información:

```
root@servidor:/etc/nagios3# less /etc/passwd | grep www-data
www-data:x:33:33:www-data:/var/www:/bin/sh
root@servidor:/etc/nagios3# less /etc/group | grep www-data
www-data:x:33:
nagios:x:121:www-data
root@servidor:/etc/nagios3# less /etc/group
```

Ilustración 29: Usuario y grupo Apache

5.10.3. MySQL

El usuario "mysql" es el propio de la herramienta. No es necesario que exista ningún usuario en el grupo "mysql". La siguiente imagen muestra esta información:

```
root@servidor:/etc/nagios3# less /etc/passwd | grep mysql
mysql:x:105:114:MySQL Server,,,:/var/lib/mysql:/bin/false
root@servidor:/etc/nagios3# less /etc/group | grep mysql
mysql:x:114:
```

Ilustración 30: Usuario y grupo MySQL

El usuario "root" de MySQL, es necesario para la completa gestión del gestor de bases de datos. La siguiente imagen muestra el usuario con todos los permisos:

```
1 select * from user where User='root' && Host='servidor'
```

39 column(s) x 1 row(s) in last result set. Duration for 1 query: 0.000 sec.

Host	User	Password						
servidor	root	*A4723AF477B9055321889532267F2FB98F1EF316	Y	Y	Y	Y	Y	Y

Ilustración 31: Usuario root en MySQL

Es necesaria la creación de un usuario para la conexión a la base de datos *MySQL* NDOUtils. La siguiente imagen muestra el usuario con los permisos mínimos:

```
1 select * from user where User='ndoutils' && Host='servidor';
```

39 column(s) x 1 row(s) in last result set. Duration for 1 query: 0.000 sec.

Host	User	Password						
servidor	ndoutils	*22C4DBAA3BE683C67EE29BED7B42AF8BE5696135	N	N	N	N	N	N

Ilustración 32: Usuario ndoutils en MySQL

5.10.4. Postfix

El usuario "postfix" es el propio de la herramienta. No es necesario que exista ningún usuario en el grupo "postfix". La siguiente imagen muestra esta información:

```
root@servidor:/etc/nagios3# less /etc/passwd | grep postfix
postfix:x:108:117::/var/spool/postfix:/bin/false
root@servidor:/etc/nagios3# less /etc/group | grep postfix
postfix:x:117:
```

Ilustración 33: Usuario y grupo Postfix

6. Pruebas

Una vez descrito, analizado e implementado el sistema de monitorización es el momento de realizar todas las pruebas pertinentes para verificar que se cumplen las funcionalidades requeridas sobre un entorno de monitorización configurado para éste fin.

6.1. Introducción

En este capítulo se describe, por un lado, una configuración de prueba donde se trabaja con los servicios instalados en el servidor Linux simulando un entorno de trabajo real y escalable y, por otro, se verifican las funcionalidades que permite el sistema.

6.2. Entorno de pruebas

Se trabaja en un entorno donde existen los siguientes dispositivos:

- Servidor: máquina Linux que contiene el sistema de monitorización y los servicios Linux a monitorizar.
- Cliente: máquina Windows. Se controla su estado en la red.
- PDA: dispositivo Windows Mobile. Se controla su estado en la red.
- Google: no es un dispositivo físico. Se utiliza para verificar que la conexión a Internet está operativa.
- Router: dispositivo de red. Se controla su estado en la red. Los demás dispositivos cuelgan de él.

En la siguiente imagen, correspondiente al apartado "Status Map" del frontal de *Nagios*, se muestran los dispositivos configurados:

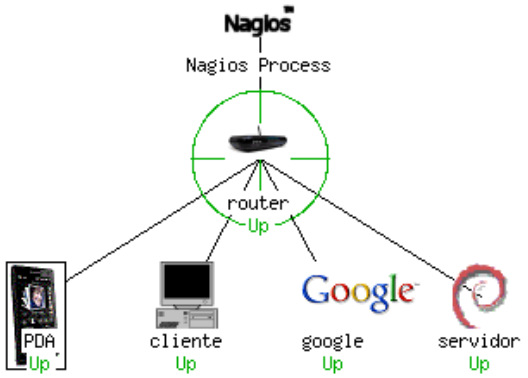


Ilustración 34: Estado de la red monitorizada en Nagios

A nivel de servicios y monitores configurados, existen monitores de "ping" que controlan la comunicación en la red de todos los dispositivos.

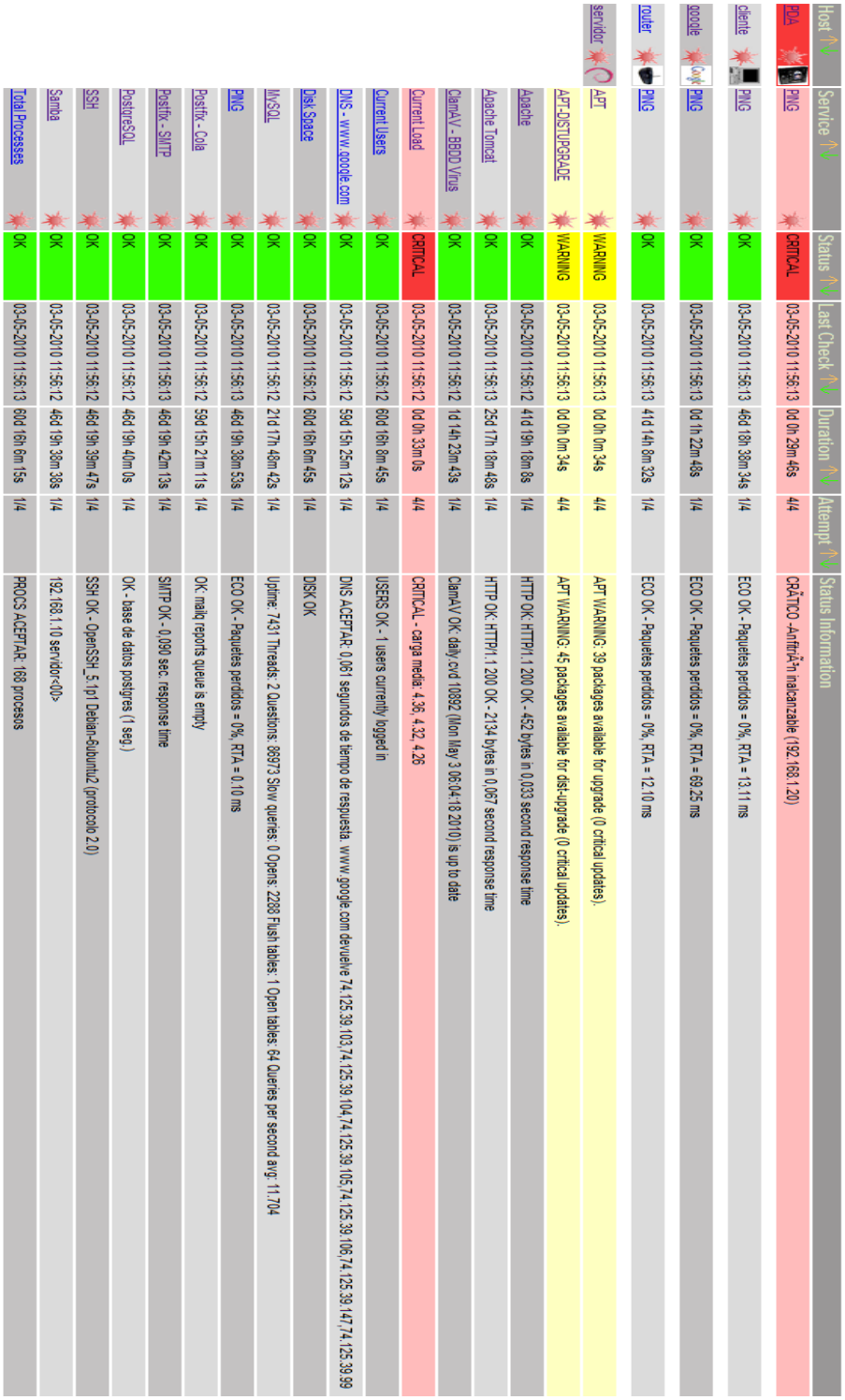
Referente a los servicios del servidor Linux, se monitorizan los siguientes:

- Ping: estado en la red.
- Apache: control del servicio y funcionamiento Web.
- Apache Tomcat: control del servicio y funcionamiento Web.
- ClamAV: base de datos de virus actualizada.
- DNS: control del servicio.
- MySQL: control del servicio y estado de la base de datos.
- Postfix: control del servicio y estado de la cola.
- PostgreSQL: control del servicio y estado de la base de datos.
- SSH: control del servicio.
- Samba: control del servicio.

Adicionalmente se han configurado otros monitores:

- APT: control de paquetes actualizados en el sistema.
- Carga del sistema: control de la carga de memoria en el sistema.
- Usuarios activos: control de usuarios activos del sistema.
- Espacio en disco: control del espacio libre /ocupado en disco.
- Procesos totales del sistema: control nº procesos totales del sistema.

En la siguiente imagen se muestran los monitores descritos, correspondientes al apartado "Service Detail" del frontal Web de *Nagios*:



Host	Service	Status	Last Check	Duration	Attempt	Status Information
PDA	PING	CRITICAL	03-05-2010 11:56:13	0d 0h 29m 46s	4/4	CRÃTICO -ArrihÃ³n inalcanzable (192.168.1.20)
cliente	PING	OK	03-05-2010 11:56:13	46d 18h 39m 34s	1/4	ECO OK - Paquetes perdidos = 0%, RTA = 13.11 ms
google	PING	OK	03-05-2010 11:56:13	0d 1h 22m 48s	1/4	ECO OK - Paquetes perdidos = 0%, RTA = 69.25 ms
router	PING	OK	03-05-2010 11:56:13	41d 14h 8m 32s	1/4	ECO OK - Paquetes perdidos = 0%, RTA = 12.10 ms
servidor	APT	WARNING	03-05-2010 11:56:13	0d 0h 0m 34s	4/4	APT WARNING: 39 packages available for upgrade (0 critical updates).
	APT-DISTUPGRADE	WARNING	03-05-2010 11:56:13	0d 0h 0m 34s	4/4	APT WARNING: 45 packages available for dist-upgrade (0 critical updates).
	Apache	OK	03-05-2010 11:56:12	41d 19h 18m 8s	1/4	HTTP OK: HTTP/1.1 200 OK - 452 bytes in 0.033 second response time
	Apache Tomcat	OK	03-05-2010 11:56:13	25d 17h 18m 48s	1/4	HTTP OK: HTTP/1.1 200 OK - 2134 bytes in 0,067 second response time
	ClamAV - BBDD Virus	OK	03-05-2010 11:56:12	1d 14h 23m 43s	1/4	ClamAV OK: daily.cvd 10892 (Mon May 3 06:04:18 2010) is up to date
	Current Load	CRITICAL	03-05-2010 11:56:12	0d 0h 33m 0s	4/4	CRITICAL - carga media: 4.36, 4.32, 4.26
	Current Users	OK	03-05-2010 11:56:12	60d 16h 8m 45s	1/4	USERS OK - 1 users currently logged in
	DNS - www.qoogle.com	OK	03-05-2010 11:56:12	59d 15h 25m 12s	1/4	DNS ACEPTAR: 0,061 segundos de tiempo de respuesta. www.google.com devuelve 74.125.39.103,74.125.39.104,74.125.39.105,74.125.39.106,74.125.39.147,74.125.39.99
	Disk Space	OK	03-05-2010 11:56:12	60d 16h 6m 45s	1/4	DISK OK
	MySQL	OK	03-05-2010 11:56:12	21d 17h 40m 42s	1/4	Uptime: 7431 Threads: 2 Questions: 89973 Slow queries: 0 Opens: 2288 Flush tables: 1 Open tables: 64 Queries per second avg: 11.704
	PING	OK	03-05-2010 11:56:13	46d 19h 38m 53s	1/4	ECO OK - Paquetes perdidos = 0%, RTA = 0.10 ms
	Postfix - Cola	OK	03-05-2010 11:56:12	59d 15h 21m 11s	1/4	OK: mailq reports queue is empty
	Postfix - SMTP	OK	03-05-2010 11:56:13	46d 19h 42m 13s	1/4	SMTP OK - 0.090 sec. response time
	PostgreSQL	OK	03-05-2010 11:56:12	46d 19h 40m 0s	1/4	OK - base de datos postgres (1 seg.)
	SSH	OK	03-05-2010 11:56:12	46d 19h 39m 47s	1/4	SSH OK - OpenSSH_5.1p1 Debian-6ubuntu2 (protocolo 2.0)
	Samba	OK	03-05-2010 11:56:12	46d 19h 38m 38s	1/4	192.168.1.10 servidor<0D>
	Total Processes	OK	03-05-2010 11:56:13	60d 16h 6m 15s	1/4	PROCS ACEPTAR: 168 procesos

Ilustración 35: Detalle de monitores en Nagios

6.3. Funcionalidades

Se han realizado las pruebas pertinentes para verificar que se cumplen las siguientes funcionalidades en el sistema de monitorización:

- Notificación de alertas vía correo electrónico:

Se ha verificado con un monitor configurado, en este caso el de *Apache*, que notifique mediante correo electrónico la fallada del servicio, que se ha parado manual y voluntariamente. Concretamente, la notificación es la siguiente:

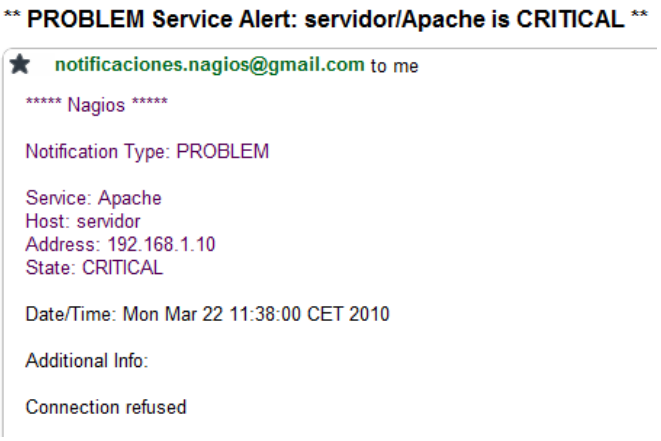


Ilustración 36: Notificación error en email

Y la notificación, una vez levantado el servicio, es la siguiente:

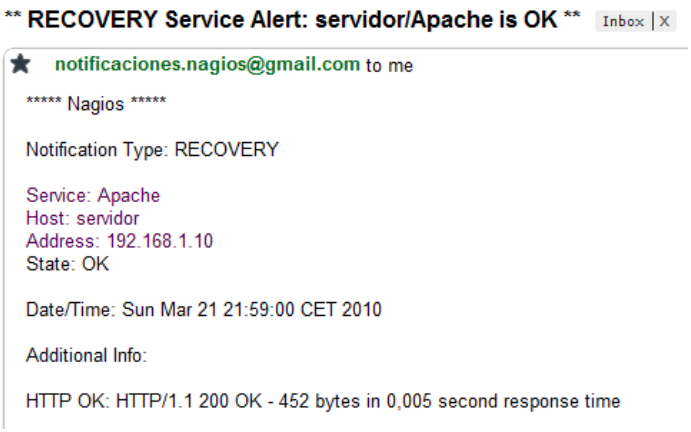


Ilustración 37: Notificación OK en email

- Notificación de alertas vía SMS:

El mismo monitor anterior, configurado también para notificar alertas vía SMS, genera la siguiente alerta ante el fallo del servicio:

"Nagios – PROBLEM: servidor/Apache is CRITICAL"

Y la notificación, una vez levantado el servicio, es la siguiente:

"Nagios – RECOVERY: servidor/Apache is OK"

- Auto recuperación de servicios:

El mismo monitor anterior está configurado también para auto recuperarse en caso de fallar. Se verifica como, parando manualmente el servicio desde la línea de comandos, pasados unos segundos el servicio se levanta nuevamente de manera automática.

- Control de los monitores mediante gráficas:

El mismo monitor anterior está configurado también para generar gráficas mediante la herramienta *PNP4Nagios*. Ésta información se puede observar desde el frontal Web de *Nagios*, vía Pop-Up, como muestra la siguiente imagen:

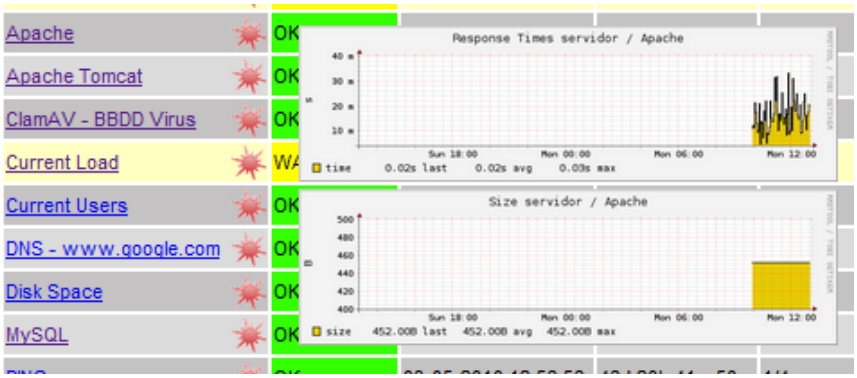


Ilustración 38: Gráficas mediante Pop-Up en Nagios

También se puede acceder, haciendo clic en el botón del monitor referente a *PNP4Nagios*:

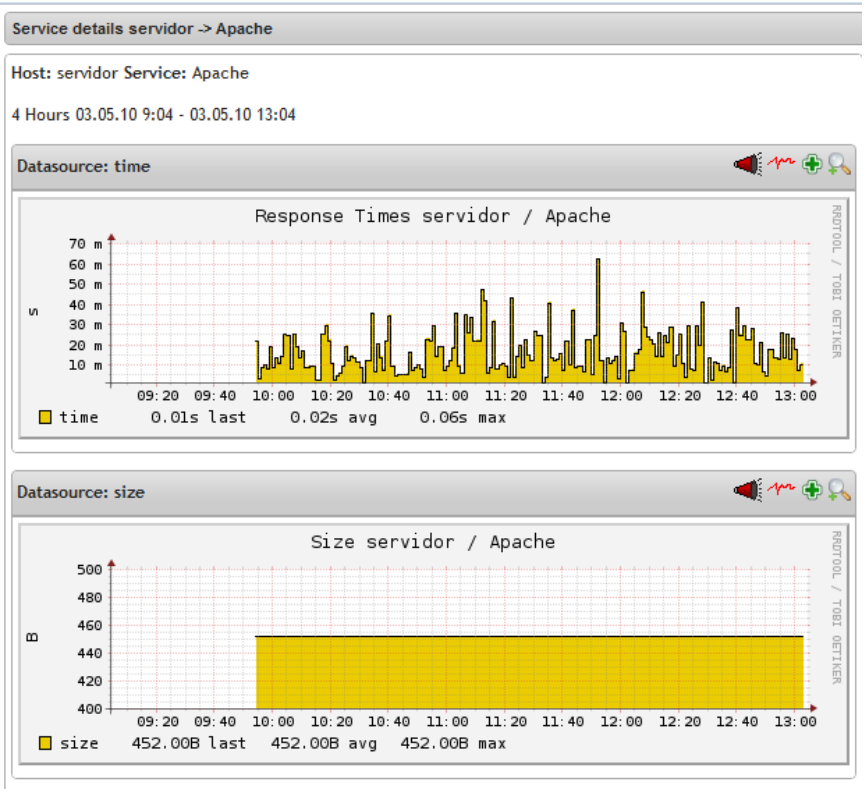


Ilustración 39: Gráficas en PNP4Nagios

7. Conclusiones

Para finalizar esta memoria, se describen las conclusiones finales que se extraen del trabajo realizado en el transcurso de este proyecto.

7.1. Introducción

En este capítulo se detallan las desviaciones sucedidas desde el inicio del proyecto comparando la planificación inicial y final. También se describen posibles ampliaciones para continuar el proyecto. Finalmente se expresan las conclusiones finales.

7.2. Desviaciones

Realmente se ha planificado correctamente el tiempo para la realización de las diferentes tareas al inicio del proyecto. No obstante, algunas etapas del proyecto han ocupado algo más del tiempo inicial esperado. En la siguiente imagen se pude observar que las etapas 9 y 10 ocupan 10 y 20 horas respectivamente:

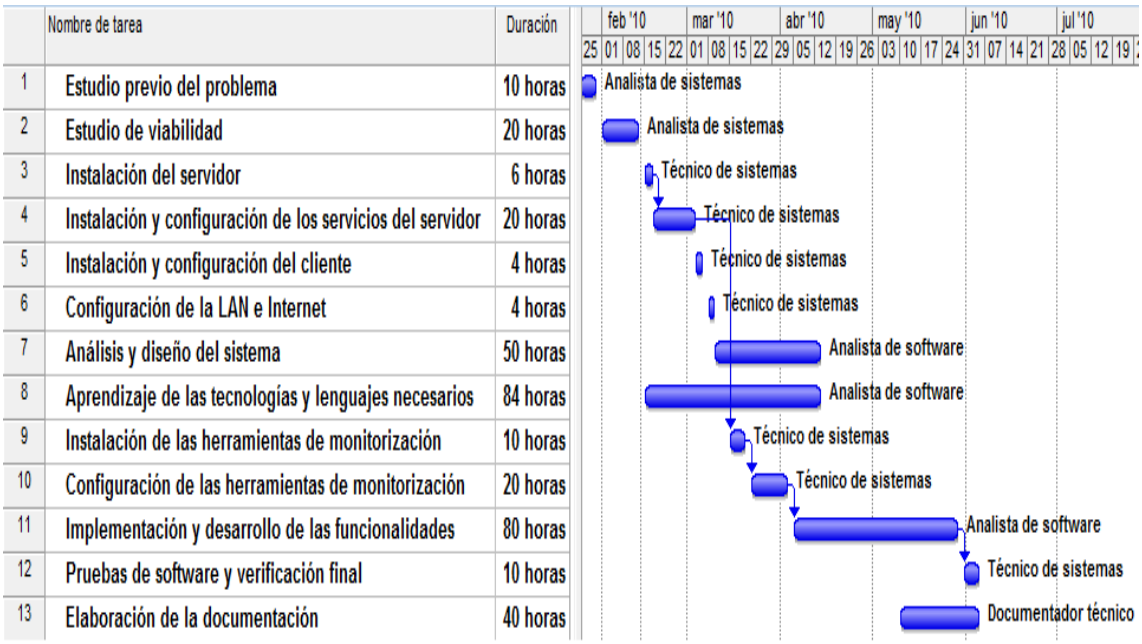
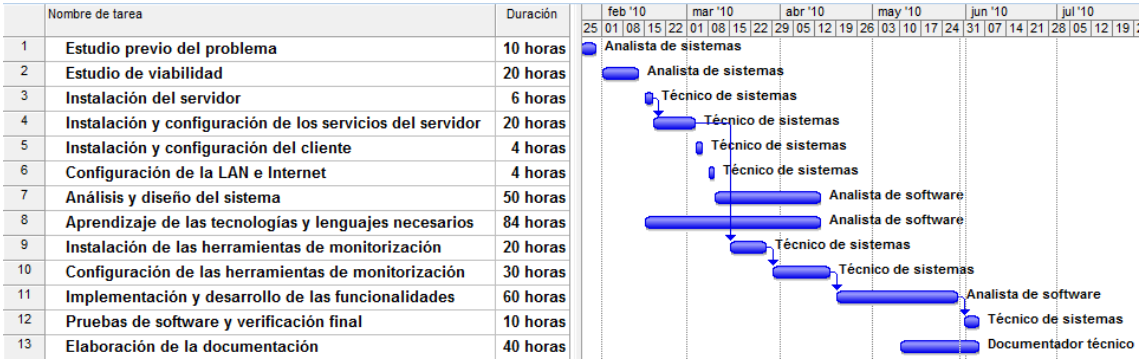


Ilustración 40: Planificación inicial

En la planificación final se puede ver que se han destinado más horas a ambas tareas:



	Nombre de tarea	Duración
1	Estudio previo del problema	10 horas
2	Estudio de viabilidad	20 horas
3	Instalación del servidor	6 horas
4	Instalación y configuración de los servicios del servidor	20 horas
5	Instalación y configuración del cliente	4 horas
6	Configuración de la LAN e Internet	4 horas
7	Análisis y diseño del sistema	50 horas
8	Aprendizaje de las tecnologías y lenguajes necesarios	84 horas
9	Instalación de las herramientas de monitorización	20 horas
10	Configuración de las herramientas de monitorización	30 horas
11	Implementación y desarrollo de las funcionalidades	60 horas
12	Pruebas de software y verificación final	10 horas
13	Elaboración de la documentación	40 horas

Ilustración 41: Planificación final.

Finalmente, también se puede ver que se ha podido destinar menos tiempo a la etapa 11, llegando a implementar satisfactoriamente las funcionalidades requeridas en ese tiempo.

A nivel global, ésta modificación no ha perjudicado en el resultado final del proyecto, en el que se han cumplido exitosamente todas las fechas de entregas y seguimientos hasta su finalización.

7.3. Ampliaciones

Se pueden destacar diferentes ampliaciones para la continuación de éste proyecto:

- Implementación de más funcionalidades al sistema *Nagios*:
  - Volcado de datos en otros gestores como *Oracle* o *PostgreSQL*.
  - Diferentes añadidos como *NagVis*, para la visualización de las infraestructuras de red monitorizadas.
  - Diferentes frontales para la gestión Web como *Centreon*.
- Modificar el entorno red y monitorización:
  - Ampliar los servidores y dedicar una máquina a la monitorización de toda la infraestructura.

- Implantar una arquitectura con **NRPE** (Nagios Remote Plugin Executor):

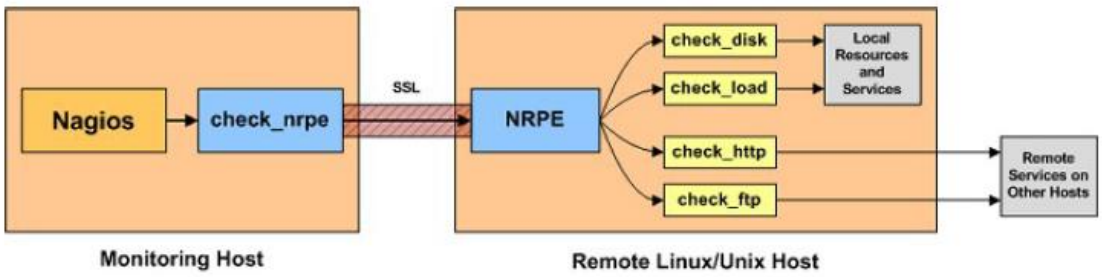


Ilustración 42: Sistema con NRPE

Como se puede observar en la imagen anterior, el servidor dedicado a la monitorización contiene el módulo de chequeo "check_nrpe" y en cada servidor monitorización existen todos los chequeos necesarios para cada servicio, gestionados por el "NRPE" que se comunica con el sistema *Nagios* del servidor.

Es posible utilizar el protocolo SSH en substitución al NRPE. SSH se considera más seguro pero aumenta la carga en las máquinas. Es más eficiente utilizar NRPE implementando una comunicación segura con SSL.

- Crear un entorno virtual de servidores con diferentes servicios para ser monitorizados desde otro servidor físico o virtual.

7.4. Conclusiones finales

Inicialmente se planteaba la implantación de un sistema de monitorización desarrollando la herramienta utilizada desde el inicio. Después de estudiar en el sector las herramientas más utilizadas y competentes y analizar las posibilidades que existían se optó por la elección de diferentes herramientas Open Source para cumplir con todas las funcionalidades requeridas.

Por lo tanto, se puede concluir con que se han cumplido los objetivos propuestos en los plazos horarios establecidos consiguiendo las metas inicialmente planteadas.

Tal y como se expresa en el apartado anterior: "Desviaciones" de éste capítulo, ha habido algún retraso en dos tareas referentes a la instalación u configuración de las herramientas de monitorización. El motivo ha venido dado por la variedad de herramientas y complementados que se han probado e implantado en el sistema de monitorización. Esto no ha repercutido en el desarrollo del proyecto puesto que finalmente se ha podido limitar la fase de implementación llegando con éxito a cumplir con todas las funcionalidades estipuladas.

A nivel personal, ha sido un proyecto que me ha motivado puesto que ya había trabajado con herramientas de monitorización en el mundo laboral y encontraba interesante poder ampliar mis conocimientos en el tema y encaminar mi perfil profesional hacia la administración y control de los sistemas, que es la parte de la informática que más me atrae.

8. Bibliografía

Se han utilizado diferentes fuentes de información:

- Sistema operativo del servidor:
  - CentOS 4.8.: http://centos.org/
  - Debian 5.0.4.: http://www.debian.org/index.es.html
  - Fedora 12.: http://fedoraproject.org/es/
  - Mandriva 2010.: http://www2.mandriva.com/es/
  - OpenSUSE 11.2.: http://es.opensuse.org/Bienvenidos_a_openSUSE.org
  - Ubuntu 9.10 Server.: http://www.ubuntu.com/

- Servicios del servidor:
  - Servidor de acceso remoto:
    - Open SSH: http://www.openssh.com/
  - Servidor Web:
    - Apache Tomcat: http://tomcat.apache.org/
  - Servidor de BBDD:
    - MySQL: http://www.mysql.com/
    - PostgreSQL: http://www.postgresql.org/
  - Servidor de DNS:
    - Bind: https://www.isc.org/products/BIND
  - Servidor de E-mail:
    - Postfix: http://www.postfix.org/
    - Sendmail: http://www.sendmail.org/
  - Servidor de antivirus:
    - ClamAV: http://www.clamav.net/
  - Servidor de antispam:
    - SpamAssassin: http://spamassassin.apache.org/
  - Servidor de archivos:
    - Samba: http://www.samba.org/

- Sistema operativo del cliente:
  - Windows 7 Ultimate: http://www.microsoft.com/windows/windows-7/

Software del cliente:

- Cliente SSH:
  - PuTTY: http://www.putty.org/
  - WinSCP: http://winscp.net/eng/docs/lang:es
- Navegador Web:
  - Microsoft Internet Explorer: http://www.microsoft.com/spain/windows/internet-explorer/
  - Mozilla Firefox: http://www.mozilla-europe.org/es/firefox/
- Herramientas Ofimáticas:
  - Microsoft Office 2007: http://office.microsoft.com/es-ES/default.aspx
  - Microsoft Project 2007: http://office.microsoft.com/es-es/project/default.asp.

- Herramientas de monitorización:
  - Cacti: http://www.cacti.net/
  - God: http://god.rubyforge.org/
  - JFFNMS: http://www.jffnms.org/
  - Monit: http://mmonit.com/monit/
  - Munin: http://munin.projects.linpro.no/
  - Nagios: http://www.nagios.org/
  - Osmius: http://www.osmius.net/es/
  - PandoraFMS: http://pandorafms.org/
  - Zabbix: http://www.zabbix.org/
  - Zenoss: http://www.zenoss.com/

- Complementos, plugins de Nagios y otras herramientas:
  - http://exchange.nagios.org/
  - http://nagiosplugins.org/
  - PNP4Nagios: http://www.pnp4nagios.org/
  - RRDTools: http://oss.oetiker.ch/rrdtool/
  - Gnokii: http://www.gnokii.org/

9. Apéndice

9.1. Índice de ilustraciones

9.2. Códigos Fuente

9.2.1. Nagios y PNP4Nagios

Fichero de configuración principal "NAGIOS.CFG":

```
#Definición de los directorios y ficheros de configuración.
cfg_dir=/etc/nagios3/conf.d/
cfg_file=/etc/nagios3/comandos.cfg
#definición de los plugins de Nagios.
cfg_dir=/etc/nagios-plugins/config/
#Definición de diferentes ficheros de recursos, logs, comandos, etc.
resource_file=/etc/nagios3/resource.cfg
log_file=/var/log/nagios3/nagios.log
log_archive_path=/var/log/nagios3/archives/
status_file=/var/cache/nagios3/status.dat
p1_file=/usr/lib/nagios3/p1.pl
#Comandos para procesar los datos de rendimiento para PNP4Nagios.
service_perfdata_command=process-service-perfdata
host_perfdata_command=process-host-perfdata
object_cache_file=/var/cache/nagios3/objects.cache
precached_object_file=/var/lib/nagios3/objects.precache
command_file=/var/lib/nagios3/rw/nagios.cmd
lock_file=/var/run/nagios3/nagios3.pid
temp_file=/var/cache/nagios3/nagios.tmp
temp_path=/tmp
check_result_path=/var/lib/nagios3/spool/checkresults
debug_file=/var/log/nagios3/nagios.debug
#Conexión con NDOUTILS, para el volcado de datos en MYSQL.
broker_module=/usr/lib/ndoutils/ndomod-mysql-3x.o config_file=/etc/nagios3/ndomod.cfg
#Diferentes parámetros de configuración del sistema.
event_broker_options=-1
nagios_user=nagios
nagios_group=nagios
enable_notifications=1
execute_service_checks=1
```

```
accept_passive_service_checks=1

enable_event_handlers=1

check_external_commands=1

command_check_interval=-1

#Habilitar el proceso de recolección de datos para PNP4Nagios.

process_performance_data=1

host_perfdata_file_mode=2

service_perfdata_file_mode=2

check_for_orphaned_services=0

check_service_freshness=1

date_format=euro

illegal_object_name_chars=~!$%^&*"|'<>?,()=

illegal_macro_output_chars=`~$^&"|'<>

host_inter_check_delay_method=s

max_host_check_spread=30

debug_level=0
```

Fichero "SERVICIOS.CFG":

```
#Apache.

define service {

       host_name                       servidor

        service_description             Apache

        check_command                   check_http

        event_handler                   restart-apache2

       use                             generic-service,srv-pnp

        notification_interval           0

        }

#SSH.

define service {

       host_name                       servidor

       service_description             SSH

        check_command                   check_ssh
```

```
        use                             generic-service,srv-pnp

         notification_interval           0

         }

#Ping (comunicación en la red). Warning > 100ms o > 20% paquetes perdidos, critical > 500 o > 60% paquetes perdidos.

define service {

        service_description             PING

         check_command                   check_ping!100.0,20%!500.0,60%

        use                             generic-service,srv-pnp

         notification_interval           0

        host_name                       servidor, router, cliente, PDA, google

         }

#Espacio en disco.  Warning < 20%, critical < 10%.

         define service{

         use                             generic-service,srv-pnp

         host_name                       servidor

         service_description             Disk Space

         check_command                   check_all_disks!20%!10%

         }

#Usuarios. Warning > 20 usuarios, critical > 50 usuarios.

define service{

         use                             generic-service,srv-pnp

         host_name                       servidor

         service_description             Current Users

         check_command                   check_users!20!50

         }

#Procesos totales. Warning > 250 procesos, critical > 400 procesos.

define service{

         use                             generic-service,srv-pnp

         host_name                       servidor

         service_description             Total Processes

          check_command                   check_procs!250!400

         }

#Carga. Warning > 7, 6, 5, critical > 10, 9, 8.

define service{
```

```
        use                             generic-service,srv-pnp
        host_name                       servidor
        service_description             Current Load
         check_command                   check_load!7.0!6.0!5.0!10.0!9.0!8.0
        }
#MySQL
define service{
        use generic-service,srv-pnp
        host_name servidor
        service_description MySQL
        check_command check_mysql_cmdlinecred
        }
#PostgreSQL
define service {
    host_name                       servidor
        service_description             PostgreSQL
        check_command                   check_pgsql
    use                             generic-service,srv-pnp
        }
#Tomcat
define service {
    host_name                       servidor
        service_description             Apache Tomcat
        check_command                   check_Tomcat
        event_handler                   restart-tomcat
    use                             generic-service,srv-pnp
        }
#DNS
define service {
    host_name                       servidor
        service_description             DNS - www.google.com
        check_command                   check_dns
    use                             generic-service,srv-pnp
        }
```

```
#SMTP (para postfix)
define service {
    host_name                       servidor
        service_description             Postfix - SMTP
        check_command                   check_smtp
    use                             generic-service,srv-pnp
        }
#Cola (para postfix)
define service {
    host_name                       servidor
        service_description             Postfix - Cola
        check_command                   check_mailq_postfix
    use                             generic-service,srv-pnp
        }
#Paquetes para actualizar distribución con APT
define service {
    host_name                       servidor
        service_description             APT-DISTUPGRADE
        check_command                   check_apt_distupgrade
    use                             generic-service,srv-pnp
        }
#Paquetes para actualizar con APT
        define service {
    host_name                       servidor
        service_description             APT
        check_command                   check_apt
    use                             generic-service,srv-pnp
        }
#ClamAV
        define service {
    host_name                       servidor
        service_description             ClamAV - BBDD Virus
        check_command                   check_clamav
    use                             generic-service,srv-pnp
```

```
}
#Samba
        define service {
    host_name                       servidor
        service_description             Samba
        check_command                   check_nmb
    use                             generic-service,srv-pnp
}
```

Fichero "SERVICIO_GENERICO.CFG":

```
define service{
name                            generic-service
active_checks_enabled           1
passive_checks_enabled          1
parallelize_check               1
obsess_over_service             1
check_freshness                 0
notifications_enabled           1
event_handler_enabled           1
flap_detection_enabled          1
failure_prediction_enabled      1
process_perf_data               1
retain_status_information       1
retain_nonstatus_information    1
notification_interval           0
is_volatile                     0
check_period                    24x7
normal_check_interval           5
retry_check_interval            1
max_check_attempts              4
notification_period             24x7
notification_options            w,u,c,r
contact_groups                  admins
```

```
register                        0

}

define service {

name       srv-pnp

#Activación del Pop-Up de PNP4Nagios (servicios) en Nagios.

action_url /pnp4nagios/graph?host=$HOSTNAME$&srv=$SERVICEDESC$' class='tips'
rel='/pnp4nagios/popup?host=$HOSTNAME$&srv=$SERVICEDESC$



register   0

}
```

Fichero "HOSTS.CFG":

```
#Router

define host {

       host_name   router

       alias       router

       address     192.168.1.1

       use         generic-host,host-pnp

       icon_image          comtrend.gif

       statusmap_image      comtrend.gd2

       }

#Servidor

define host{

       use                     generic-host,host-pnp

       host_name               servidor

       alias                   servidor

       address                 192.168.1.10

       icon_image                   debian.gif

       statusmap_image      debian.gd2

         parents router

       }

#Cliente

define host{

        use                    generic-host,host-pnp
```

```
        host_name               cliente
        alias                   cliente
        address                 192.168.1.4
        icon_image              cliente.gif
       statusmap_image          cliente.gd2
         parents router
        }
#PDA
define host{
        use                     generic-host,host-pnp
        host_name               PDA
        alias                   PDA
        address                 192.168.1.20
       icon_image              htc.gif
       statusmap_image          htc.gd2
         parents router
        }
#Google-Verifica internet.
define host{
        use                     generic-host,host-pnp
        host_name               google
        alias                   google - Internet
        address                 www.google.com
        icon_image              google.gif
       statusmap_image          google.gd2
         parents router
        }
```

Fichero "GRUPOS_HOSTS.CFG":

```
define hostgroup {
#miembros del grupo servidores
       hostgroup_name  servidores
                alias           servidores
```

```
                members         servidor

        }
define hostgroup {
#miembros del grupo clientes
         hostgroup_name  clientes
                alias           clientes
                members         cliente, PDA
         }
define hostgroup {
#miembros del grupo dispositivos
        hostgroup_name  dispositivos
                alias           dispositivos
                members         router
         }
```

Fichero "HOST_GENERICO.CFG":

```
define host{
name                            generic-host
notifications_enabled           1
event_handler_enabled           1
flap_detection_enabled          1
failure_prediction_enabled      1
process_perf_data               1
retain_status_information       1
retain_nonstatus_information    1
check_command                   check-host-alive
max_check_attempts              10
notification_interval           0
notification_period             24x7
notification_options            d,u,r
contact_groups                  admins
register                        0
}
```

```
define host {

name       host-pnp

#Activación del Pop-Up de PNP4Nagios (hosts) en Nagios.

action_url /pnp4nagios/graph?host=$HOSTNAME$&srv=_HOST_' class='tips'
rel='/pnp4nagios/popup?host=$HOSTNAME$&srv=_HOST_

register   0

}
```

Fichero "CONTACTOS.CFG":

```
define contact{

contact_name                    root

alias                           Root

service_notification_period     24x7

host_notification_period        24x7

service_notification_options    w,u,c,r

host_notification_options       d,r

#Comandos de notificación de alertas de servicios.

service_notification_commands   notify-service-by-email,notify-service-by-sms

#Comandos de notificación de alertas de hosts.

host_notification_commands      notify-host-by-email,notify-host-by-sms

#Receptor de alertas por email.

email                           carlos.ramos.gallardo@gmail.com

#Receptor de alertas por SMS.

pager                                   636667870

}

define contactgroup{

contactgroup_name       admins

alias                   Nagios Administrators

members                 root

}
```

Fichero "TIEMPOS.CFG":

```
define timeperiod{

timeperiod_name 24x7

alias           24 Hours A Day, 7 Days A Week

#funciona todo el tiempo.

sunday          00:00-24:00

monday          00:00-24:00

tuesday         00:00-24:00

wednesday       00:00-24:00

thursday        00:00-24:00

friday          00:00-24:00

saturday        00:00-24:00

}

define timeperiod{

#no funciona nunca.

timeperiod_name never

alias           Never

}
```

Fichero "COMANDOS.CFG":

```
#Notificación por email HOST.

define command{

command_name    notify-host-by-email

command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type:
$NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress: $HOSTADDRESS$\nInfo:
$HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /usr/bin/mail -s "** $NOTIFICATIONTYPE$ Host
Alert: $HOSTNAME$ is $HOSTSTATE$ **" $CONTACTEMAIL$

}

#Notificación por email SERVICIO.

define command{

command_name    notify-service-by-email

command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification Type:
$NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress: $HOSTADDRESS$\nState:
$SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional Info:\n\n$SERVICEOUTPUT$" |
/usr/bin/mail -s "** $NOTIFICATIONTYPE$ Service Alert: $HOSTALIAS$/$SERVICEDESC$ is
```

```
$SERVICESTATE$ **" $CONTACTEMAIL$

}

#Notificación por SMS SERVICIO.

define command {

command_name notify-service-by-sms

command_line /usr/bin/printf "Nagios - $NOTIFICATIONTYPE$ : $HOSTALIAS$/$SERVICEDESC$ is
$SERVICESTATE$ ($OUTPUT$)" | /usr/bin/gnokii --sendsms $CONTACTPAGER$ -r

}

#Notificación por SMS HOST.

define command {

command_name notify-host-by-sms

command_line /usr/bin/printf "Nagios - $NOTIFICATIONTYPE$ : Host $HOSTALIAS$ is $HOSTSTATE$
($OUTPUT$)" | /usr/bin/gnokii --sendsms $CONTACTPAGER$ -r

}

#Generación de gráficas de Servicios.

define command {

command_name    process-service-perfdata

command_line    /usr/bin/perl /usr/local/pnp4nagios/libexec/process_perfdata.pl

}

#Generación de gráficas de Hosts.

define command {

command_name    process-host-perfdata

command_line    /usr/bin/perl /usr/local/pnp4nagios/libexec/process_perfdata.pl -d HOSTPERFDATA

}

#Evento de reinicio de Apache

define command {

command_name restart-apache2

command_line  /usr/local/nagios/libexec/eventhandlers/restart-apache2 $SERVICESTATE$
$SERVICESTATETYPE$ $SERVICEATTEMPT$

}

#Evento de reinicio de Tomcat

define command {

command_name restart-tomcat

command_line  /usr/local/nagios/libexec/eventhandlers/restart-tomcat $SERVICESTATE$
$SERVICESTATETYPE$ $SERVICEATTEMPT$

}
```

Fichero "CGI.CFG":

```
main_config_file=/etc/nagios3/nagios.cfg

physical_html_path=/usr/share/nagios3/htdocs

url_html_path=/nagios3

show_context_help=1

use_pending_states=1

nagios_check_command=/usr/lib/nagios/plugins/check_nagios /var/cache/nagios3/status.dat 5
'/usr/sbin/nagios3'

use_authentication=1

use_ssl_authentication=0

#autorización a los contenidos del frontal Web por los usuarios definidos.

authorized_for_system_information=nagiosadmin

authorized_for_configuration_information=nagiosadmin

authorized_for_system_commands=nagiosadmin

authorized_for_all_services=nagiosadmin,usuario1

authorized_for_all_hosts=nagiosadmin,usuario1

authorized_for_all_service_commands=nagiosadmin

authorized_for_all_host_commands=nagiosadmin

default_statusmap_layout=5

default_statuswrl_layout=4

ping_syntax=/bin/ping -n -U -c 5 $HOSTADDRESS$

refresh_rate=90

escape_html_tags=1

action_url_target=_blank

notes_url_target=_blank

lock_author_names=1
```

Fichero "RESOURCE.CFG":

```
#contenidos de variables utilizadas por el sistema

$USER1$=/usr/local/nagios/libexec

$USER2$=public
```

9.2.2. Nagios-Plugins

En éste apartado solo se muestran las definiciones de los comandos utilizados y, el resto, se omiten.

Fichero "HTTP.CFG":

```
#Apache

define command{

        command_name    check_http

        command_line    /usr/lib/nagios/plugins/check_http -H '$HOSTADDRESS$' -I
'$HOSTADDRESS$'

}
```

```
#Tomcat

define command{

        command_name    check_Tomcat

        command_line    /usr/lib/nagios/plugins/check_http -H '$HOSTADDRESS$' -I
'$HOSTADDRESS$' -p 8080 }
```

Fichero "PING.CFG":

```
#Ping

define command{

        command_name    check_ping

        command_line    /usr/lib/nagios/plugins/check_ping -H '$HOSTADDRESS$' -w '$ARG1$' -c
'$ARG2$'

}
```

Fichero "SSH.CFG":

```
#SSH

define command{

        command_name    check_ssh

        command_line    /usr/lib/nagios/plugins/check_ssh '$HOSTADDRESS$'

        }
```

Fichero "DISK.CFG":

```
#Espacio en disco

define command{

        command_name    check_all_disks

        command_line    /usr/lib/nagios/plugins/check_disk -w '$ARG1$' -c '$ARG2$' -e
```

Fichero "USERS.CFG":

```
#Usuarios

define command{

        command_name    check_users

        command_line    /usr/lib/nagios/plugins/check_users -w '$ARG1$' -c '$ARG2$'

        }
```

Fichero "PROCS.CFG":

```
#Procesos

define command{

        command_name    check_procs

        command_line    /usr/lib/nagios/plugins/check_procs -w '$ARG1$' -c '$ARG2$'

        }       }
```

Fichero "LOAD.CFG":

```
#Carga

define command{

        command_name    check_load

        command_line    /usr/lib/nagios/plugins/check_load --warning='$ARG1$,$ARG2$,$ARG3$' --
critical='$ARG4$,$ARG5$,$ARG6$'

        }
```

Fichero "MYSQL.CFG":

```
#MySQL

define command{

        command_name    check_mysql_cmdlinecred

        command_line    /usr/lib/nagios/plugins/check_mysql -H '$HOSTADDRESS$' -u ndoutils -p
ndoutils

}
```

Fichero "PGSQL.CFG":

```
#PostgreSQL

define command{

        command_name    check_pgsql

        command_line    /usr/lib/nagios/plugins/check_pgsql -H '$HOSTADDRESS$' -d postgres -l
postgres_nagios -p postgres_nagios

        }
```

Fichero "DNS.CFG":

```
#DNS

define command{

        command_name    check_dns

        command_line    /usr/lib/nagios/plugins/check_dns -H www.google.com -s '$HOSTADDRESS$'

}
```

Fichero "MAIL.CFG":

```
#SMTP

define command {

        command_name    check_smtp

        command_line    /usr/lib/nagios/plugins/check_smtp -H '$HOSTADDRESS$'

}

#Cola Postfix

define command {

        command_name    check_mailq_postfix
```

```
        command_line    /usr/lib/nagios/plugins/check_mailq -w 5 -c 10 -M postfix
}
```

Fichero "APT.CFG":

```
#APT
define command{
        command_name    check_apt
        command_line    /usr/lib/nagios/plugins/check_apt
        }

#APT DIST-UPGRADE
define command{
        command_name    check_apt_distupgrade
        command_line    /usr/lib/nagios/plugins/check_apt -d
        }
```

Fichero "CLAMAVCFG":

```
#ClamAV actualizado
define command{
        command_name    check_clamav
        command_line    /usr/lib/nagios/plugins/check_clamav
        }
```

Fichero "SAMBA.CFG":

```
#Samba
define command{
        command_name    check_nmb
        command_line    /usr/bin/perl /usr/lib/nagios/plugins/check_nmb -H servidor
        }
```

9.2.3. NDOutils

Fichero "NDOMOD.CFG":

```
instance_name=default

#socket de salida

output_type=unixsocket

output=/var/cache/nagios3/ndo.sock

#puerto

tcp_port=5663

output_buffer_items=5000

buffer_file=/var/cache/nagios3/ndomod.tmp

file_rotation_interval=14400

file_rotation_timeout=60

reconnect_interval=15

reconnect_warning_interval=15

data_processing_options=-1

config_output_options=2
```

Fichero"NDO2DB.CFG":

```
#usuario de Nagios utilizado por NDOUtils

ndo2db_user=nagios

ndo2db_group=nagios

#socket

socket_type=unix

socket_name=/var/cache/nagios3/ndo.sock

#puerto

tcp_port=5663

#tipo de servidor BBDD

db_servertype=mysql

#host

db_host=localhost

#puerto para la conexión con la BBDD
```

```
db_port=3306

#nombre de la BBDD

db_name=ndoutils

db_prefix=nagios_

#usuario para la conexión con la BBDD NDOUTILS

db_user=ndoutils

db_pass=ndoutils

max_timedevents_age=1440

max_systemcommands_age=10080

max_servicechecks_age=10080

max_hostchecks_age=10080

max_eventhandlers_age=44640

debug_level=0

debug_verbosity=1

debug_file=@localstatedir@/ndo2db.debug

max_debug_file_size=1000000
```

9.2.4. Apache

Fichero "NAGIOS3.CONF":

```
ScriptAlias /cgi-bin/nagios3 /usr/lib/cgi-bin/nagios3

ScriptAlias /nagios3/cgi-bin /usr/lib/cgi-bin/nagios3

Alias /nagios3/stylesheets /etc/nagios3/stylesheets

#Indica alias para la dirección en el navegador Web y lo relaciona con el directorio real

Alias /nagios3 /usr/share/nagios3/htdocs

<DirectoryMatch (/usr/share/nagios3/htdocs|/usr/lib/cgi-bin/nagios3|/etc/nagios3/stylesheets)>

        Options FollowSymLinks

        DirectoryIndex index.html

        AllowOverride AuthConfig

        Order Allow,Deny

        Allow From All

        AuthName "Nagios Access"
```

```
        AuthType Basic
#indica el fichero que consulta para los usuarios autorizados
        AuthUserFile /etc/nagios3/htpasswd.users
        require valid-user
</DirectoryMatch>
```

Fichero "PNP4NAGIOS.CONF":

```
#Indica alias para la dirección en el navegador Web y lo relaciona con el directorio real
Alias /pnp4nagios "/usr/local/pnp4nagios/share"
<Directory "/usr/local/pnp4nagios/share">
        AllowOverride None
        Order allow,deny
        Allow from all
        AuthName "Nagios Access"
        AuthType Basic
#indica el fichero que consulta para los usuarios autorizados
        AuthUserFile /etc/nagios3/htpasswd.users
        Require valid-user
        <IfModule mod_rewrite.c>
                RewriteEngine On
                Options FollowSymLinks
                RewriteBase /pnp4nagios/
                RewriteRule ^(application|modules|system) - [F,L]
                RewriteCond %{REQUEST_FILENAME} !-f
                RewriteCond %{REQUEST_FILENAME} !-d
                RewriteRule .* index.php/$0 [PT,L]
        </IfModule>
</Directory>
```

9.2.5. Postfix

Fichero "MAIN.CF":

```
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)

biff = no

append_dot_mydomain = no

readme_directory = no

#Especifica el servidor SMTP configurado.

relayhost = [smtp.gmail.com]

smtp_use_tls = yes

smtp_sasl_auth_enable = yes

#indica el fichero de password

smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd

smtp_sasl_security_options = noanonymous

smtp_sasl_tls_security_options = noanonymous

#indica el fichero generic

smtp_generic_maps = hash:/etc/postfix/generic
```

Fichero "SASL_PASSWD":

```
#credenciales de de la cuenta externa

[smtp.gmail.com] notificaciones.nagios@gmail.com:n0t1f1c4c10n3s.n4g10s
```

Fichero "GENERIC":

```
#relaciona la cuenta local de root con la cuenta real externa

root@localhost notificaciones.nagios@gmail.com
```

9.2.6. Gnoki

Fichero "GNOKIIRC":

```
[global]

#puerto

port = /dev/ttyACM0

#modelo

model = AT

initlength = default

#conexión

connection = serial

use_locking = yes

serial_baudrate = 115200

smsc_timeout = 10

[xgnokii]

allow_breakage = 0

[gnokiid]

bindir = /usr/sbin/

[connect_script]

TELEPHONE = 12345678

[disconnect_script]

[logging]

debug = off

rlpdebug = off

xdebug = off
```

9.3. Glosario

ADSL +2: estándar de telecomunicaciones correspondiente al ITU (International Telecommunication Union) G.992.5.

BBDD: abreviación de Base de datos informática.

BIND: servidor de nombres de dominio (Berkeley Internet Name Daemon).

BSD: licencia de software (Berkeley Software Distribution).

C: lenguaje de programación de propósito general orientado a desarrollar sistemas y software.

C++: lenguaje de programación de propósito general orientado a objetos.

CGI: estándar de un servidor Web (Common Gateway Interface) relacionado con la generación de las páginas Web.

DNS: servidor de nombres de dominio (Domain Name System) que convierte dominios en direcciones lógicas IP y viceversa.

FTP: protocolo de red estándar (File Transfer Protocol) para la transferencia de ficheros en la red.

GCC: sistema de compilación de GNU (GNU Compiler Collection).

GD: librería gráfica Graphics Library relacionada con las imágenes.

GNU GPL: licencia de software libre (GNU General Public License).

GSM: sistema global de comunicaciones móviles (Global System for Mobile Communications).

HTTP: protocolo estándar de la capa de aplicación (Application Layer protocol).

HTTPS: protocolo estándar seguro de la capa de aplicación (Hypertext Transfer Protocol Secure) que implementa SSL/TLS.

IIS: servidor Web de Microsoft (Internet Information Services o Server).

IP: protocolo de Internet (Internet Protocol) que permite identificar dispositivos en una red.

JSP: procedimiento Java (Java stored procedure) ejecutados por la máquina virtual de Java.

LAN: red de área local (Local area network).

MIT: licencia de software libre (Massachusetts Institute of Technology).

NRPE: plugin ejecutor remoto de Nagios (Nagios Remote Plugin Executor).

PHP: lenguaje de scripts (Hypertext Preprocessor) orientado al desarrollo Web y páginas dinámicas.

RAM: método de acceso aleatorio a memoria (Random-access memory).

RPM: sistema de gestión de paquetes (Package Manager).

RRDTOOL: herramienta que utiliza el protocolo Round-Robin (round-robin database tool) en bases de datos circulares.

SFTP: protocolo de red seguro (Secure File Transfer Protocol).

SLA: nivel de servicio contratado (service level agreement).

SMS: servicio de mensajes cortos en una red GSM (Short Message Service).

SMTP: protocolo de transferencia de correo (Simple Mail Transfer Protocol).

SNMP: protocolo de gestión de red (Simple Network Management Protocol).

SQL: lenguaje de bases de datos (Structured Query Language).

SSH: protoco de red seguro (Secure Shell).

SSL: protocolo criptográfico de seguridad (Secure Socket Layer).

TCP: protocolo de control de la transmisión (Transmission Control Protocol).

TI: tecnologías de la información IT (Information technology).

UDP: protocolo de red (User Datagram Protocol).

UMTS: sistema de telecomunicaciones móviles universal (Universal Mobile Telecommunications System).

URL: localizador de recursos uniforme (Uniform Resource Locator).

X11: sistema de interface gráfica (X Window System).