



Universitat Autònoma
de Barcelona

SISTEMA DE MONITORIZACIÓN DE SERVIDORES LINUX

Memoria del proyecto
de Ingeniería Técnica en
Informática de Sistemas
realizado por
Víctor Arrebola Real
y dirigido por
Marc Talló Sendra

Escuela de Ingeniería
Sabadell, Septiembre de 2010

El bajo firmante, Marc Talló Sendra,
profesor de la Escuela de Ingeniería de la UAB,

CERTIFICA:

Que le trabajo al que corresponde la presente
memoria ha sido realizado bajo su dirección
por Víctor Arrebola Real

I para que conste firma la presente.
Sabadell, Septiembre de 2010

Firmado: Marc Talló Sendra
Director

Presentación

Antes de comenzar, me gustaría agradecer al señor Marc Tallò Sendra el soporte y la confianza que ha tenido en mi y en este proyecto. También dar las gracias a los compañeros, amigos y familiares que me han dado su apoyo.

Así pues, comencemos con la exposición del proyecto:

La necesidad de monitorizar redes ha estado presente casi desde que existe la necesidad de usarlas y con la evolución de las nuevas tecnologías en los últimos años, las redes de computadoras han sufrido un crecimiento enorme, permitiendo que estas sean cada vez mayores en cuanto a tamaño y complejidad.

Siendo varios los motivos que nos pueden llevar a querer monitorizar una red, el más común es el hecho de saber cuando algo esta fallando, permitiéndonos incluso poder llegar a predecir situaciones determinadas y facilitar posibles planes de ampliación.

Partiendo como base de la mezcla de pasión y curiosidad que siento por la arquitectura y funcionamiento de las redes de computadoras, entenderemos la elección de desarrollar este proyecto, y además hacerlo siempre al nivel más bajo posible, evitando el uso de frameworks, librerías, y software de terceros, en la medida de lo posible.

Índice

1 – Introducción	1
1.1 – Presentación	1
1.2 – Objetivos	1
1.3 – Estado del arte	1
1.4 – Estructura de la memoria	2
 2 - Estudio de viabilidad	 4
2.1 – Introducción	4
2.2 – Objetivos	5
2.2.1 – Esquematización de los objetivos	7
2.2.2 – Prioridad de los objetivos	8
2.2.3 – Partes interesadas	9
2.3 – Requisitos del proyecto	10
2.3.1 – Requisitos funcionales	10
2.3.2 – Requisitos no funcionales	10
2.3.3 – Restricciones del sistema	11
2.4 – Estado del arte	11
2.4.1 – Alternativa 1: Nagios	11
2.4.2 – Alternativa 2: Hobbit	12
2.4.3 – Alternativa 3: Zenos	12
2.4.4 – Alternativa 4: Desarrollo a medida	13
2.5 – Planificación	13
2.5.1 – Recursos	14
2.5.2 – Tareas	15
2.5.3 – Planificación temporal	16
2.6 – Evaluación de riesgos	17
2.6.1 – Lista de riesgos	17
2.6.2 – Catalogación de los riesgos	18
2.6.3 – Plan de contingencia	18
2.7 – Presupuesto	18
2.8 – Conclusiones	19
 3 – Fundamentos teóricos	 20
3.1 – Que es GNU/Linux?	20
3.2 – Que es GNU?	21
3.3 – Que es el software libre?	21
3.4 – Que es un sistema de monitorización de red?	22
 4 – Análisis	 23
4.1 – Software utilizado	23
4.2 – Lenguajes utilizados	25
4.3 – Protocolo SNMP vs. diseño propio	27
4.4 – Diseño del sistema	28
4.5 – Base de datos	29
4.6 – Diagramas de casos de uso	31

5 – Implementación	34
5.1 – Arquitectura	34
5.1.1 – Aplicación central	35
5.1.2 – Script servidor	35
5.1.3 – Script cliente	35
5.2 – Mapa Web	37
5.3 – Descripción de interfaces	38
5.3.1 – Login	38
5.3.2 – Home	38
5.3.3 – Listado de nodos	40
5.3.4 – Monitorización de nodos	41
5.3.5 – Configuración de nodos	43
5.3.6 – Listado de usuarios	44
5.3.7 – Configuración de usuarios	44
5.3.8 – Alarmas	44
5.3.9 – Descarga del cliente	47
5.3.10 – Ayuda	47
6 – Pruebas	48
6.1 – Entorno de pruebas	48
6.2 – Navegadores Web	49
6.3 – Distribuciones GNU/Linux	49
6.4 – Disponibilidad de conexión	50
7 – Conclusiones	52
7.1 – Conclusiones	52
7.2 – Futuras ampliaciones	52
8 – Bibliografía	54
Anexo I. Manual de usuario	56

1 - Introducción

1.1 - Presentación

El presente proyecto pretende cubrir las necesidades de monitorización de una red por definir. Principalmente deberán ser monitorizados tanto los servidores como las estaciones de trabajo.

Todos estos host tendrán en común que funcionarán con sistemas operativos GNU/Linux, intentando en la medida de lo posible abarcar diversas distribuciones, aunque principalmente se trabajará para las basadas en Debian, y más concretamente en Ubuntu.

1.2 - Objetivos

Los objetivos básicos que debe cubrir la aplicación son tres:

- Monitorización de los nodos de red en tiempo real.
- Creación de un sistema de alarmas que permita alertar vía e-mail y/o sms cuando algunos de los aspectos parametrizados supere los rangos establecidos.
- La aplicación debe ser accesible desde cualquier nodo de la red, y deberá ser valorada la opción de poder acceder desde fuera de la misma siempre y cuando se disponga de medios que lo hagan posible como la permisión de acceso por parte de los proxys de red, o conexiones VPN.

Cabe destacar que un requisito no funcional, será tratar de desarrollar la totalidad del proyecto, usando para ello únicamente *software libre*. Este requisito abarca desde los entornos de desarrollo, sistemas operativos y bases de datos, hasta los lenguajes de programación y tecnologías utilizadas, sin pasar por alto las herramientas ofimáticas utilizadas para elaborar la documentación del mismo.

1.3 - Estado del arte

Dado que en la actualidad existen multitud de aplicaciones orientadas a la monitorización de redes, deberá valorarse el coste y las funcionalidades de las mismas, en comparación al desarrollo de un proyecto a medida.

Algunas de las aplicaciones más conocidas y extendidas en entornos de trabajo bajo sistemas operativos Unix o GNU/Linux son:

- Nagios
- Hobbbit
- Zenos

Nos centraremos en el análisis de estos tres productos para valorar los pros y contras de cada uno de ellos.

1.4 - Estructura de la memoria

Incluyendo el apartado que nos ocupa, la memoria consta de 8 capítulos, a través de los cuales se expondrá toda la información relativa al desarrollo del proyecto.

Capítulo 2: Estudio de Viabilidad

En este capítulo se analizarán desde un punto de vista más técnico los objetivos expuestos en esta introducción y se realizará un estudio con el fin de verificar la viabilidad del proyecto.

Capítulo 3: Fundamentos teóricos

En un documento en el que detalla el desarrollo de un proyecto en un ámbito como es la Ingeniería Informática, se dan por supuestos muchos conceptos técnicos. Sin embargo, hay ciertos aspectos más específicos que deben ser explicados con más detalle. Será en este apartado en el que se explicarán y resumirán algunos de estos conceptos.

Capítulo 4: Análisis

Detallaremos el software utilizado en el desarrollo del proyecto, y analizaremos el diseño del sistema, así como sus casos de uso y la estructura de la base de datos de la aplicación.

Capítulo 5: Implementación

Capítulo dedicado a los detalles relativos a la implementación del sistema. Explicaremos detalladamente la arquitectura de la aplicación, así como los detalles técnicos relacionados con el funcionamiento de la misma.

Capítulo 6: Pruebas

Se expondrán detalladamente la totalidad de los test realizados, conjuntamente con las valoraciones de los resultados obtenidos, así como el entorno en el que han sido realizados los mismos.

Capítulo 7: Conclusiones

En este apartado repasaremos las conclusiones resultantes de la elaboración de este proyecto, y analizaremos posibles líneas futuras de ampliación del mismo.

Capítulo 8: Bibliografía

Referencia de la documentación consultada.

2 - Estudio de viabilidad

2.1 - Introducción

Son varias las razones por las que podemos querer monitorizar una red, pero la más común es el deseo de saber cuando algo esta fallando. Además de para encontrar fallos, monitorizar sirve para poder predecir posibles situaciones y actuar en consecuencia para prevenirlas. Si sabemos, por ejemplo, que uno de los discos duros de nuestro servidor se encuentra al 95% de su capacidad, podemos anticiparnos y realizar una ampliación antes de quedarnos sin espacio.

Otra ventaja de la monitorización es que facilita los planes de ampliación. Por ejemplo, una tendencia que busca en el pasado y proporciona datos de históricos nos informará sobre que discos o cuotas de usuario se ven incrementadas en un 10% mensual, ayudándonos a decidir si en X meses será necesario un nuevo servidor de ficheros o no.

Cualquiera de las tareas que implica monitorizar una red deben ser automatizadas, pues nos seria imposible analizar personalmente todo lo sucedido, debido a que en todo momento hay bytes de información volando a través de los cables, servidores de bases de datos aceptando y registrando transacciones y CPUs ejecutando millones de instrucciones por segundo.

Podemos clasificar las aplicaciones de monitorización en tres áreas distintas:

- Monitorización de estados
- Monitorización de rendimientos
- Monitorización de registros o logs

En la actualidad existe multitud de software relacionado con la monitorización de redes en general y de servidores en particular, y es que la necesidad de monitorizar redes ha estado presente casi desde que existe la necesidad de usarlas. Para acotar un poco la cantidad de aplicaciones existentes, nos centraremos en las que trabajan en entornos Linux.

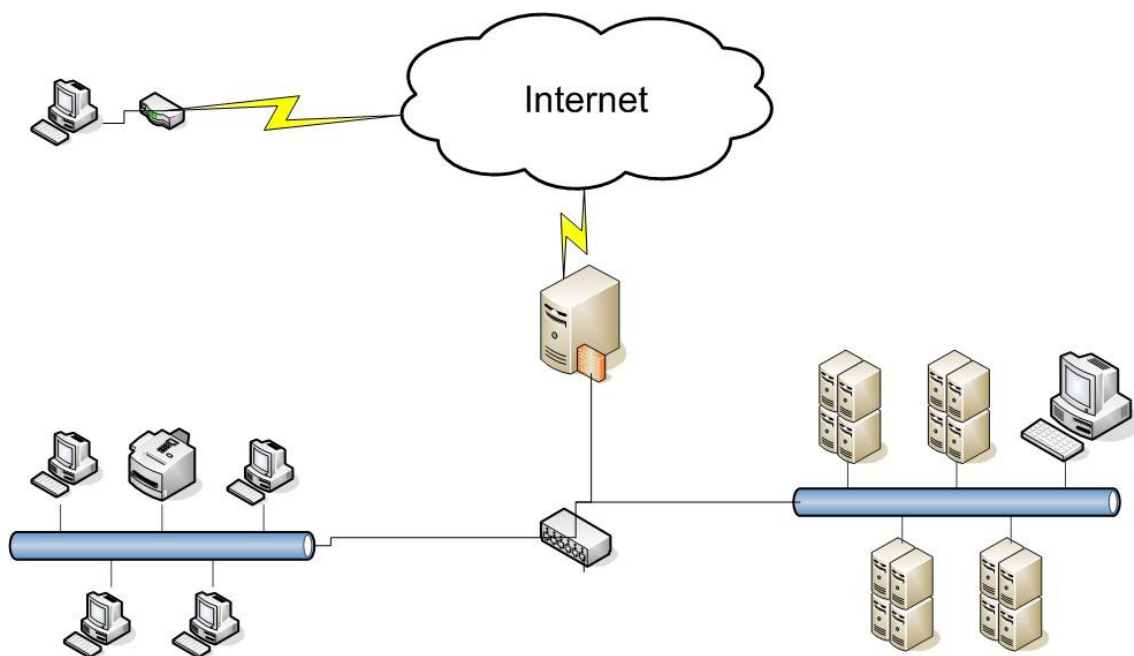
Dentro de estas cotas podemos encontrar desde aplicaciones como Nagios, tan potente y extensible, como complicada de configurar, hasta “monit”, una aplicación aparentemente sencilla y ligera, pero que es capaz de controlar y monitorizar procesos, servicios, archivos, directorios y otras variables del sistema, tanto local como remotamente. Hobbit, Munin, mon y Zenos son sólo algunos ejemplos más de aplicaciones de monitorización en entornos Linux.

Descartando el sobresaturado mercado de aplicaciones orientadas a grandes redes por motivos relacionados con los recursos y el tiempo disponible, el proyecto a desarrollar se centrará en posibles implantaciones en redes de tamaños de pequeños y medios.

La idea principal del proyecto es crear un software ligero, facilitando al máximo la configuración y el uso de este por parte del usuario final.

2.2 - Objetivos

Este proyecto parte desde cero y su implantación no esta orientada a ningún tipo de red en concreto. Debido a esta condición inicial, se partirá de una topología de red básica, intentando cumplir tres objetivos, dos de los cuales son considerados como prioritarios más un tercero secundario.



Los objetivos prioritarios serán:

- Monitorización de los nodos de red en tiempo real.
- Creación de un sistema de alarmas que permita alertar vía e-mail y/o sms cuando algunos de los aspectos parametrizados supere los rangos establecidos.

- La aplicación debe ser accesible desde cualquier nodo de la red, y deberá ser valorada la opción de poder acceder desde fuera de la misma siempre y cuando se disponga de medios que lo hagan posible como la permisión de acceso por parte de los proxys de red, o conexiones VPN.

La solución obvia para permitir el acceso a la aplicación desde cualquier equipo, tanto dentro como fuera de la red, es desarrollar una interfaz Web que pueda ser accesible desde cualquier navegador. En este apartado cabe destacar que dadas las características del proyecto, descartaremos el acceso a la interfaz desde smartphones u otros dispositivos móviles, centrando los esfuerzos de desarrollo en los principales navegadores Web para estaciones de trabajo. Por lo que se buscara obtener una completa compatibilidad con:

- Mozilla Firefox 3.3 o superior
- Google Chrome 5.0 o superior
- Internet Explorer 6.0 o superior
- Opera 9.0 o superior

Para cumplir el requisito de la monitorización en tiempo real, deberemos hacer uso de la tecnología AJAX, pudiendo de esta manera ofrecer una interfaz Web dinámica en tiempo real de calidad.

En cuanto al sistema de alarmas, se estudiará la posibilidad de utilizar el protocolo SNMP para obtener datos de los equipos vigilados, y establecer *traps*, que nos alerten de manera automática cuando se produzca algún tipo de alerta sobre los valores configurados.

Una alternativa al uso del protocolo SNMP, sería desarrollar un modelo de aplicación cliente-servidor, para poder establecer comunicaciones entre la aplicación central y los host monitorizados, tanto para obtener información, como para enviar las alarmas.

Para evitar posibles usos indebidos de la aplicación que puedan comprometer el sistema, se crearan dos perfiles de usuario distintos:

- Administrador
Tendrá acceso ilimitado a la aplicación
- Usuario
Se le limitaran opciones tales como altas, bajas y edición tanto de usuarios y hosts, como de alarmas. Sin embargo podrá consultar toda la información relacionada con los campos anteriormente mencionados, además de tener acceso a la monitorización en tiempo real.

El desarrollo se llevará a cabo siguiendo un método lineal, el cual deberá pasar por las siguientes fases:

- 1 - Formación
- 2 - Estudio y diseño del sistema
- 3 - Desarrollo de la interfaz gráfica
- 4 - Implementación de las bases de datos
- 5 - Desarrollo de la aplicación central
- 6 - Desarrollo y configuración de los clientes
- 7 - Implementación, cohesión y pruebas locales del sistema
- 8 - Pruebas en real
- 9 - Documentación del proyecto y manual de usuario

2.2.1 - Esquematización de los objetivos

1 – Formación

O1.1 - Incluye el estudio referente a materias desconocidas o profundizar en conocimientos específicos como el desarrollo con la tecnología AJAX, el protocolo SNMP, la configuración del servidor Apache2 o la obtención de los parámetros deseados a monitorizar en sistemas Linux.

2 – Estudio y diseño del sistema

O2.1 - Concretar el funcionamiento, los módulos necesarios y la arquitectura de estos

3 – Desarrollo de la interfaz gráfica

O3.1 – Interfaz básica desarrollada con HTML, CSS, PHP y Javascript

O3.2 – Control de usuarios a la interfaz de la aplicación

O3.3 – Implementación de un motor AJAX para la interfaz grafica

4 – Implementación de la base de datos

O4.1 – Cumplimiento de los estándares de BBDD

O4.2 – Desarrollo e implementación del la BBDD

5 – Desarrollo de la aplicación central

O5.1 - Control de acceso a la aplicación

O5.2 – Gestión de usuarios

O5.3 – Gestión de dispositivos de red

O5.5 – Monitorización y muestreo de los parámetros de los dispositivos de red

O5.6 – Detección de dispositivos de red

O5.7 – Sistema de alertas (Email, SMS)

O5.8 – Monitor físico de alertas

6 – Desarrollo y configuración de los clientes

O6.1 – Desarrollo de la aplicación cliente encargada de suministrar la información.

O6.2 – Aplicación de configuración/instalación automatizada de los clientes.

O6.3 – Aplicación de configuración para soporte de herramientas basadas en mrtg.

7 – Implementación, cohesión y pruebas locales del sistema.

O7.1 – Diseño de las pruebas del sistema

O7.2 – Aplicación de los modelos de test

8 – Pruebas en real

O8.1 – Diseño y montaje de la red de pruebas.

O8.2 – Aplicación de los modelos de test en real

9 - Documentación del proyecto y manual de usuario

O9.1 - Redacción del documento del PFC.

O9.2 – Redacción del los respectivos manuales de usuario.

2.2.2 - Priorización de los objetivos

	Crítico	Prioritario	Secundario
O1.1	X		
O2.1	X		
O3.1	X		
O3.2		X	
O3.3			X
O4.1		X	
O4.2	X		
O5.1		X	
O5.2	X		
O5.3	X		
O5.4			
O5.5	X		
O5.6			X
O5.7		X	
O5.8			X
O6.1	X		
O6.2		X	
O6.3		X	
O7.1	X		
O7.1	X		
O8.1	X		
O8.2	X		
O9.1	X		
O9.2		X	

2.2.3 - Partes interesadas

Se describirán todas las partes interesadas en el proyecto, como los stakeholders, los perfiles de usuario que interactuarán con la aplicación, y los componentes del equipo de trabajo.

- Stakeholders

Nombre	Descripción	Responsabilidad
SH1	Responsable de la entidad	Patrocinador. Aprobación del proyecto. Participa en su definición y realiza seguimiento del proyecto.
SH2	Administrador, usuario experto	Participa en la definición de requisitos, suministro de información, representa al usuario tipo. Participa en la validación del proyecto.
SH3	Director del proyecto	Supervisa el trabajo del alumno. Avalúa el proyecto.

- Perfiles de usuario

Nombre	Perfil	Responsabilidad
U1	Administrador del sistema	Administrador del sistema.
U2	Usuario experto	Usuarios con acceso a la aplicación con un grado de conocimiento alto sobre la red.

- Equipo de trabajo

Nombre	Descripción	Responsabilidad
PT1	Encargado de proyecto	Define, gestiona, planifica y controla el proyecto
PT2	Analista	Colabora con el encargado de proyecto en el estudio de viabilidad y la planificación. Analiza la aplicación: arquitectura, metodología, especificaciones, estándares. Participa en el diseño y la validación
PT3	Programador	Diseña y desarrolla la aplicación de acuerdo con el análisis y planificación prevista. Participa en el proceso de validación e implantación.
PT4	Técnico de pruebas	Participa en el diseño de las pruebas internas y externas. Realiza las pruebas y participa en el proceso de control de calidad.
PT5	Director del proyecto y tutor	Supervisa el trabajo del alumno, también realiza funciones de stakeholder.

2.3 - Requisitos del proyecto

A continuación se exponen de forma esquematizada los requisitos del proyecto:

2.3.1 - Requisitos funcionales

- Control de acceso de los usuarios a la aplicación
- Mantenimiento (altas/bajas/modificaciones) de los dispositivos a monitorizar
- Mantenimiento (altas/bajas/modificaciones) de los usuarios de la aplicación
- Generación de informes
- Alertas vía mail
- Alertas vía sms
- Sistema de detección de dispositivos
- Panel luminoso de alertas
- Sistema de ayuda en línea

2.3.2 - Requisitos no funcionales

- Tolerancia a errores y acciones incorrectas
- Seguridad física del servidor
- Garantizar la disponibilidad y estabilidad

- Documentación del proyecto y manuales de usuario
- Normalización de la base de datos y acceso según el estándar SQL 99 (ISO/IEC 9072:1999)

2.3.3 - Restricciones del sistema

- La aplicación se tiene que implementar en un entorno Linux
- El proyecto ha de estar finalizado antes del 20 de Junio de 2010
- En la medida de lo posible, las herramientas que se usaran para su desarrollo serán software libre.

2.4 - Estado del arte

Tal y como se comentó en la introducción, evaluaremos algunas de las soluciones software existentes en el mercado orientadas a la monitorización de redes.

Para cada una de las alternativas se incluirá un breve resumen descriptivo de la aplicación, un resumen de sus funcionalidades más destacadas, el coste de la misma y finalmente una breve conclusión.

Procedemos a evaluar cada una de las alternativas:

2.4.1 - Alternativa 1: Nagios

Sistema open source de monitorización de redes ampliamente utilizado, que vigila los equipos (hardware) y servicios (software) que se especifiquen, alertando cuando el comportamiento de los mismos se aparta del especificado.

Funcionalidades:

- Monitorización de servicios de red.
- Monitorización de los recursos de equipos hardware.
- Monitorización remota, a través de túneles SSL cifrados o SSH.
- Chequeo de servicios paralizados.
- Permite distinguir entre host caídos y host inaccesibles.
- Notificaciones a los contactos cuando ocurren problemas en servicios o hosts, así como cuando son resueltos.
- Visualización del estado de la red en tiempo real a través de interfaz Web, con la posibilidad de generar informes y gráficas de comportamiento de los sistemas monitorizados.

Costes:

Software bajo licencia GNU General Public License Versión 2 publicada por la Free Software Foundation.

Resumen:

Nagios es una herramienta muy potente y extensible utilizada ampliamente en el mundo profesional por muchos administradores de sistemas para la monitorización de redes. El mayor inconveniente de esta aplicación es su curva de aprendizaje, ya requiere mucho tiempo y experiencia instalar y configurar correctamente la aplicación.

2.4.2 - Alternativa 2: Hobbit

Hobbit es un software de monitorización ideal para redes de tamaño medio. Funciona como un sistema centralizado, por lo que necesitamos un servidor central más un software cliente en cada máquina que queramos monitorizar.

Características:

- Interfaz de administración GUI.
- Binarios preempaquetados para distintos sistemas operativos con gran cantidad de plugins.
- Notificación por E-mail o SMS.
- Fácilmente configurable.
- Proporciona soporte comercial

Costes:

Software bajo licencia GPL Open source.

Resumen:

Se trata de una aplicación ligera e intuitiva capaz de cumplir de sobras con las expectativas para las que fue creado. Interfaz gráfica muy básica y sencilla aunque extremadamente funcional.

La versión release más reciente data de Agosto de 2006, aunque actualmente se ha hecho cargo del proyecto un nuevo grupo de desarrolladores bajo el nombre de Xymon, para el que si podemos encontrar versiones algo más recientes.

2.4.3 - Alternativa 3: Zenos

Zenoss nos proporciona una herramienta de monitorización muy versátil y con una interfaz muy amigable e intuitiva.

Características:

- Monitoreo de disponibilidad de dispositivos en la red utilizando SNMP
- Monitoreo de servicios de red (HTTP, POP3, NNTP, SNMP, FTP).
- Monitoreo de recursos de máquinas anfitrionas (Microprocesador, utilización de disco) en la mayoría de los sistemas operativos.
- Herramientas de gestión de eventos para anotar las alertas de un sistema.
- Detecta automáticamente recursos de una red y cambios en su configuración.

Costes:

Licencia Pública General de GNU (GPL) versión 2.

Resumen:

A pesar de su relativa facilidad de uso, se combinan una instalación bastante complicada, y una potencia, aunque aceptable, ni mucho menos comparable con la de Nagios.

2.4.4 - Alternativa 4: Desarrollo de una aplicación de monitorización de servidores

Permite ajustarse a los requisitos del patrocinador o a los recursos disponibles del sistema a monitorizar.

Características:

- Control de acceso de los usuarios a la aplicación
- Mantenimiento (altas/bajas/modificaciones) de los dispositivos a monitorizar
- Mantenimiento (altas/bajas/modificaciones) de los usuarios de la aplicación
- Generación de informes
- Alertas vía mail
- Alertas vía sms
- Sistema de detección de dispositivos
- Panel luminoso de alertas
- Sistema de ayuda en línea.

Costes:

Según la planificación prevista

2.5 - Planificación

El proyecto se desarrollará de Diciembre de 2009 a Junio de 2010, con una dedicación de 15 horas semanales. El total de horas dedicadas al proyecto será de 310 horas.



- Fecha de inicio: 9 de diciembre de 2009.
- Fecha de finalización: 9 de Mayo de 2010.

Las herramientas de planificación i control del proyecto serán DotProject y la suite OpenOffice.

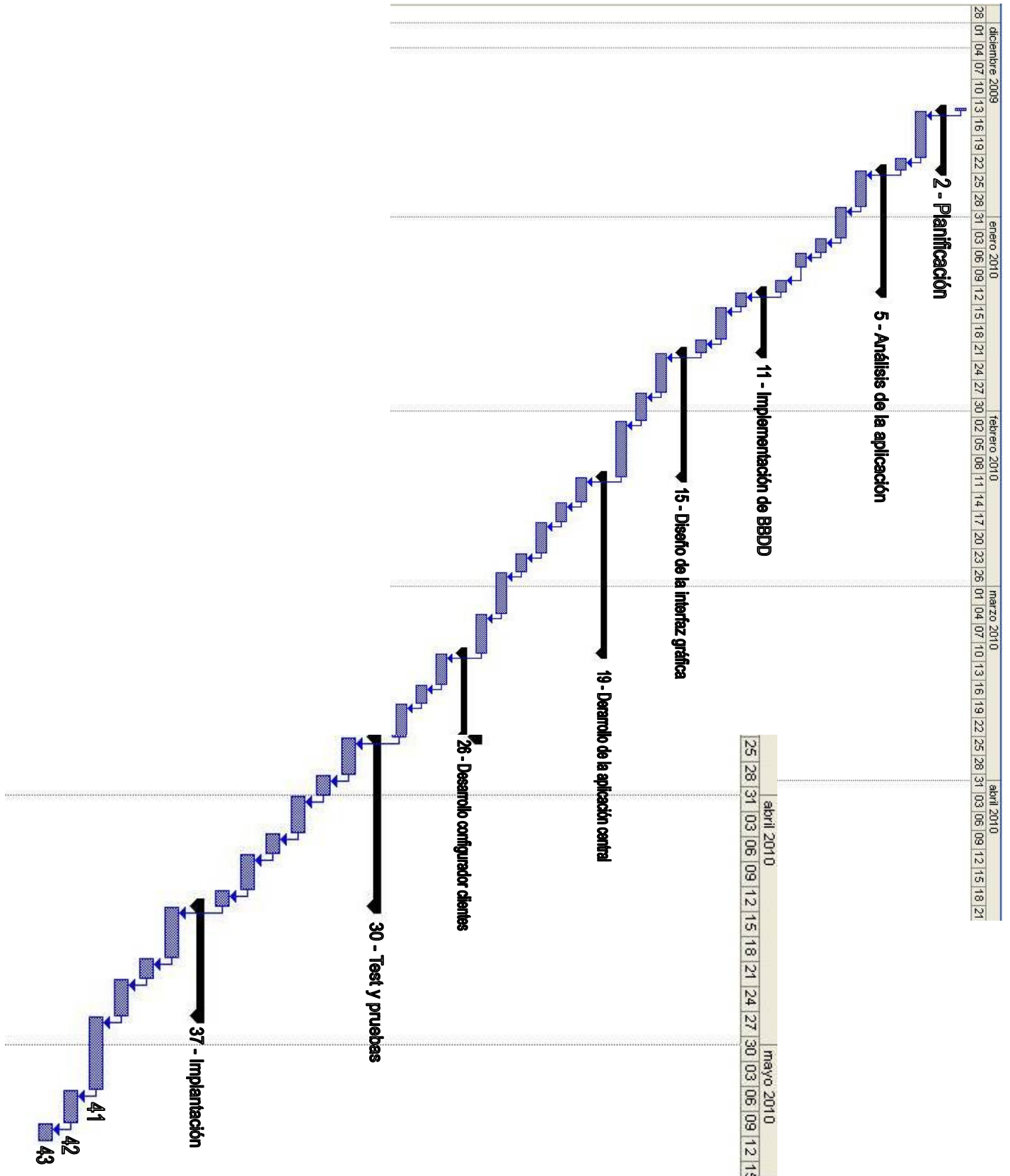
2.5.1 - Recursos del proyecto: Humanos i materiales

Recursos humanos	Valoración
Jefe de proyecto	100 €/h
Analista	50 €/h
Administrador	40 €/h
Programador	30 €/h
Técnico de pruebas	20 €/h

2.5.2 - Tareas del proyecto

		Nombre de tarea	Duración	Predecesoras
1		Inicio del proyecto: asignación i matriculación del proyecto	2 horas	
2		Planificación		
3		Estudio de viabilidad	30 horas	1
4		Aprobación del Estudio de Viabilidad (Punto de control)	1 hora	3
5		Análisis de la aplicación		
6		Análisis de requisitos (casos de uso)	15 horas	4
7		Análisis de datos (base de datos)	10 horas	6
8		Análisis de seguridad y legalidad	5 horas	7
9		Documentación del análisis	3 horas	8
10		Aprobación del análisis (Punto de control)	1 hora	9
11		Implementación de la base de datos		
12		Preparación del entorno	3 horas	10
13		Desarrollo de la BBDD	10 horas	12
14		Test	2 horas	13
15		Diseño de la interfaz gráfica		
16		Interfaz básica	20 horas	14
17		Control de acceso usuarios	5 horas	16
18		Diseño y desarrollo motor AJAX	40 horas	17
19		Desarrollo de la aplicación central		
20		Preparación entorno de desarrollo	2 horas	18
21		Módulo gestión usuarios	10 horas	20
22		Módulo gestión dispositivos de red	10 horas	21
23		Monitorización y muestreo de parametros	20 horas	22
24		Módulo de detección de dispositivos de red	20 horas	23
25		Módulo de sistema de alertas	20 horas	24
26		Desarrollo configurador clientes		
27		Configurador clientes basados en Debian	10 horas	25
28		Configurador otras arquitecturas	10 horas	27
29		Configurador MRTG	10 horas	28
30		Test y pruebas		
31		Pruebas unitarias	5 horas	29
32		Pruebas de integración	5 horas	31
33		Pruebas de estrés	5 horas	32
34		Pruebas de comunicación y alto rendimiento	5 horas	33
35		Documentación de desarrollo y test	2 horas	34
36		Aprobación del desarrollo y pruebas (Punto de control)	1 hora	35
37		Implantación		
38		Diseño y montaje de la red de pruebas	20 horas	36
39		Instalación	5 horas	38
40		Pruebas en real	4 horas	39
41		Generación de documentos (memoria del proyecto)	40 horas	40
42		Cierre del proyecto	1 hora	41
43		Defensa del proyecto	2 horas	42

2.5.3 - Planificación temporal



2.6 - Evaluación de riesgos

2.6.1 - Lista de riesgos

R1. Planificación temporal optimista:

Estudio de viabilidad. No se acaba en la fecha prevista, aumentan los recursos.

R2. Falta de alguna tarea necesaria:

Estudio de viabilidad. No se cumplen los objetivos del proyecto

R3. Cambio en los requisitos:

Estudio de viabilidad, análisis. Retardo en el desarrollo y resultado.

R4. Equipo del proyecto demasiado reducido:

Estudio de viabilidad. Retardo en la finalización del proyecto, no se cumplen los objetivos del proyecto.

R5. Herramientas de desarrollo inadecuadas:

Implementación. Retardo en la finalización del proyecto, menor calidad, ...

R6. Dificultad para acceder a los stakeholders:

Estudio de viabilidad, análisis, pruebas, formación. Faltan requisitos o son inadecuados, retardos, insatisfacción de los usuarios.

R7. No se realiza correctamente la fase de test:

Desarrollo, implantación. Falta de calidad, deficiencias en la operatividad insatisfacción de los usuarios pérdidas económicas.

R8. Incumplimiento de alguna norma, reglamento o legislación:

En cualquier fase. No se cumplen los objetivos, repercusiones legales.

R9. Falta de implantación de medidas de seguridad:

Estudio de viabilidad, análisis, desarrollo. Pérdida de información, incumplimiento legal, pérdidas económicas.

R10. Abandono del proyecto antes de la finalización:

En cualquier fase. Pérdidas económicas, frustración. Posibles repercusiones legales.

2.6.2 - Catalogación de riesgos

	Probabilidad	Impacto
R1	Alta	Critico
R2	Alta	Critico
R3	Alta	Marginal
R4	Alta	Critico
R5	Baja	Critico
R6	Baja	Critico
R7	Alta	Critico
R8	Media	Critico
R9	Alta	Critico
R10	Baja	Catastrófico

2.6.3 - Plan de contingencia

	Soluciones a adoptar
R1	Aplazar alguna funcionalidad, afrontar posibles perdidas, contratar un seguro.
R2	Revisar el estudio de viabilidad, modificar la planificación
R3	Renegociar con el cliente, aplazar funcionalidad, modificar planificación y presupuesto.
R4	Solicitar un aplazamiento, negociar con el cliente, afrontar perdidas.
R5	Mejorar la formación del equipo. Prevenir herramientas alternativas, mejorar la calidad.
R6	Fijar un calendario de reuniones, mejorar el contacto con el cliente.
R7	Diseñar los test con antelación, realizar tests automáticos, negociar contrato de mantenimiento, dar garantías, afrontar perdidas económicas.
R8	Revisar las normas y legislación, consultar a un experto, afrontar posibles repercusiones de carácter penal.
R9	Revisar la seguridad en cada fase, aplicar políticas de seguridad activas.
R10	No tiene solución.

2.7 - Presupuesto

Debido a que la totalidad del software utilizado para el desarrollo del proyecto se trata de software libre, no aplica el valorar los costes del mismo, así que únicamente estimaremos los costes de personal:

Estimación costes de personal

Costes de personal imputables directamente al proyecto.

Estimación costes de personal		
Jefe de proyecto	38 h	3.800 €
Analista	50 h	2.500 €
Administrador	15 h	600 €
Programador	150 h	4.500 €
Técnico de pruebas	25 h	500 €
Total:		11.900 €

2.8 - Conclusiones

Una vez analizados los costes y la planificación del proyecto, llegamos a la conclusión de que la realización del mismo es **viable**.

Con este estudio hemos podido comprobar que los objetivos prioritarios del proyecto pueden llegar a cumplirse con facilidad, y dado que la previsión no ha sido demasiado ajustada, podrían llegar a llevarse a cabo algunos de los objetivos secundarios planteados.

3 - Fundamentos teóricos

En este capítulo se realizará una breve descripción de algunos de los conceptos teóricos que abarca el desarrollo de este proyecto.

Como se ha expuesto en capítulos anteriores, este proyecto se ha desarrollado prácticamente en su totalidad usando software libre, tanto para la edición de la memoria con la suite ofimática OpenOffice, como para el desarrollo de la aplicación mediante el uso de entornos de desarrollo gráficos, aplicaciones como Apache para el servidor, o lenguajes de programación y base de datos sujetos a licencias libres. Es por eso de vital importancia, que además de explicar detalladamente en qué consiste un sistema de monitorización de redes, aclarar conceptos como *software libre*, *GNU/Linux* o la *licencia GPL*.

3.1 - Qué es GNU/Linux?

GNU/Linux es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado Linux, que es usado con herramientas de sistema GNU. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL (Licencia Pública General de GNY) y otra serie de licencias libres.

A pesar de que Linux (núcleo) es, en sentido estricto, el sistema operativo, parte fundamental de la interacción entre el núcleo y el usuario (o los programas de aplicación) se maneja usualmente con las herramientas del proyecto GNU o de otros proyectos como GNOME. Sin embargo, una parte significativa de la comunidad, así como muchos medios generales y especializados, prefieren utilizar el término Linux para referirse a la unión de ambos proyectos.

A las variantes de esta unión de programas y tecnologías, a las que se les adicionan diversos programas de aplicación de propósitos específicos o generales se las denomina distribuciones. Su objetivo consiste en ofrecer ediciones que cumplan con las necesidades de un determinado grupo de usuarios. Algunas de ellas son especialmente conocidas por su uso en servidores y supercomputadoras, donde tiene la cuota más importante del mercado. Según un informe de IDC, GNU/Linux es utilizado por el 78% de los principales 500 servidores del mundo. Con menor cuota de mercado el sistema GNU/Linux también es usado en el segmento de las computadoras de escritorio, portátiles, computadoras de bolsillo, teléfonos móviles, sistemas embebidos, videoconsolas y otros dispositivos.

La distribución elegida para desarrollar el proyecto en un principio fue Debian, que es de sobras conocida por su estabilidad y utilidad, aunque ciertamente implica una cierta complejidad de uso en usuarios no avanzados. Es por eso que se optó por Ubuntu, una distribución basada en Debian, que actualmente goza de gran popularidad en gran medida debido a la estabilidad y eficiencia heredada de Debian, y una gran facilidad de

uso, con una GUI (Graphic User Interface) amigable e intuitiva que facilita su uso a todo tipo de usuarios.

La versión de ubuntu elegida fue la 9.10 denominada Karmic Koala, publicada en Octubre de 2009 y soportada hasta Abril de 2011. La comunidad de Ubuntu gira alrededor de las ideas expresadas en la Filosofía Ubuntu: que el software debe estar disponible de forma gratuita, que las herramientas de software deben poder ser utilizadas por la gente en su idioma local, y que la gente debe tener la libertad de personalizar y alterar su software de la manera que necesiten. Por esos motivos, según comentan en la propia página oficial de Ubuntu:

- Ubuntu siempre será gratuito y no tiene costes adicionales en la “enterprise edition”, haciendo accesible el mejor trabajo a cualquiera en los mismos términos de gratuidad.
- Usa lo mejor en infraestructura de traducciones y accesibilidad que la comunidad de software libre es capaz de ofrecer, para hacer que Ubuntu sea utilizable por el mayor número de personas posible.
- Se publica de manera regular y predecible; se publica una nueva versión cada seis meses. Puede usar la versión estable actual o ayudar a mejorar la versión actualmente en desarrollo. Cada versión está soportada al menos durante 18 meses.
- Ubuntu está totalmente comprometido con los principios del desarrollo de software de código abierto; animando a la gente a utilizar software de código abierto, a mejorarlo y a compartirlo.

3.2 - Que es el proyecto GNU?

Aunque hayamos comentado el binomio GNU/Linux, merece la pena hacer hincapié y resaltar la importancia del proyecto GNU, ya que habitualmente suele pasarse por alto, sin darle la importancia que le pertoca.

El Proyecto GNU fue lanzado en Enero de 1984 por Richard Stallman, para desarrollar un sistema operativo completo estilo Unix compuesto de software libre: El sistema GNU. Actualmente se usan ampliamente variantes del sistema operativo GNU, que usa el núcleo Linux.

El proyecto GNU está fuertemente relacionado con la filosofía del software libre, que es central en los proyectos que derivan de él, como Ubuntu.

3.3 - Que es el software libre?

“Software Libre” no significa que no haga falta pagar por él. Lo que significa es que le puede dar al software el uso que quiera: el código fuente que forma la base del software libre está disponible para que cualquiera lo descargue, modifique y en general usarlo como le parezca. Además de los beneficios ideológicos, esta libertad trae ventajas técnicas: cuando un programa es desarrollado, el sudor que los autores han invertido en

él puede ser aprovechado para construir sobre él. Por el contrario, el software que no es libre no se presta para que otros construyan sobre él, sino que hay que empezar desde cero.

En el libro escrito por Richard M. Stallman, “Software libre para una sociedad libre”, encontramos la siguiente definición de software libre seguido de las libertades que este debe cumplir para ser libre:

El “software libre” es una cuestión de libertad, no de precio. Para comprender este concepto, debemos pensar en la acepción de libre como en “libertad de expresión” y no como en “barra libre de cerveza”.

Con software libre nos referimos a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Nos referimos especialmente a cuatro clases de libertad para los usuarios de software:

- *Libertad 0: la libertad para ejecutar el programa sea cual sea nuestro propósito.*
- *Libertad 1: la libertad para estudiar el funcionamiento del programa y adaptarlo a tus necesidades – el acceso al código fuente es condición indispensable para esto.*
- *Libertad 2: la libertad para redistribuir copias y ayudar así a tu vecino.*
- *Libertad 3: la libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad – el acceso al código fuente es condición indispensable para esto.*

Que es un sistema de monitorización de red?

En una red nos encontramos con que en cada momento hay docenas de procesos funcionando, archivos volando a través de los cables, servidores Web sirviendo páginas, servidores de datos aceptando y registrando transacciones, y CPUs ejecutando millones de instrucciones por segundo.

Es por esto por lo que es necesario un sistema que controle lo sucedido en nuestra red y nos informe de sucesos que pueden ser de nuestro interés, ya que nos es imposible controlar en todo momento lo que esta sucediendo, las 24 horas al día, los 365 días del año.

Hay infinidad de parámetros que nos puede interesar que analice un sistema de monitorización de red, pero en nuestro caso nos centraremos en la monitorización de servidores, y concretamente en los consumos de CPU, memoria RAM, espacio en disco, y el estado de los servicios que seleccionemos.

4 - Análisis

Uno de los principales objetivos, ha sido diseñar y desarrollar la totalidad del proyecto, utilizando para ello única y exclusivamente software libre.

A continuación se repasarán las herramientas, lenguajes y tecnologías utilizadas analizándolas brevemente, para acabar exponiendo las cuestiones relacionadas con el diseño de la aplicación.

4.1 - Software utilizado

Ubuntu

Ubuntu ha sido la distribución Linux elegida para desarrollar el proyecto. Se trata de una distribución Linux basada en Debian GNU/Linux que proporciona un sistema operativo actualizado y estable para el usuario medio, con un fuerte enfoque en la facilidad de uso y de instalación del sistema. Es por esto que se ha centrado el desarrollo del proyecto en buscar la compatibilidad total con esta distribución, aunque también es totalmente compatible con la mayoría de distribuciones como Mandriva, Red Hat, Fedora, Debian, Open Suse, etc.

MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario propiedad de Sun Microsystems, y ésta a su vez de Oracle Corporation.

Licencia

MySQL es desarrollado como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso.

La licencia GNU GPL de MySQL obliga a que la distribución de cualquier producto derivado (aplicación) se haga bajo esa misma licencia. Si un desarrollador desea incorporar MySQL en su producto pero desea distribuirlo bajo otra licencia que no sea la GNU GPL, puede adquirir una licencia comercial de MySQL.

MyISAM

MyISAM es la tecnología de almacenamiento de datos usada por defecto por el sistema administrador de bases de datos relacionales MySQL. Este tipo de tablas están basadas en el formato ISAM pero con nuevas extensiones. El formato ISAM (Indexed Sequential Acces Method), se trata de un método para almacenar información a la que se pueda acceder rápidamente.

La principal característica de este tipo de almacenamiento es la gran velocidad que obtiene en las consultas, ya que no tiene que hacer comprobaciones de la integridad referencial, ni bloquear las tablas para realizar las operaciones por la ausencia de características de atomicidad. Este tipo de tablas está especialmente indicado para sistemas que no tienen un número elevado de inserciones, como es nuestro caso.

Tal y como hemos comentado, dado que en nuestra aplicación hay baja concurrencia en la modificación de datos y en cambio el entorno es más intensivo en la lectura de datos, hace de MySQL el motor ideal para la base de datos de nuestra aplicación.

Apache

El servidor HTTP Apache es un servidor Web http de código abierto desarrollado dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

La versión utilizada es la XX, la última release estable publicada en el momento de iniciar el diseño de la aplicación.

La decisión de utilizar el servidor HTTP Apache como servidor Web del proyecto se ha basado en características como su modularidad, su robustez, fiabilidad y eficiencia. También se han tenido en cuenta aspectos como el hecho de ser de código abierto, y la popularidad de la que goza, lo cual facilita la consecución de ayuda y soporte técnico. Además Apache es el componente de servidor Web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python, lo cual, debido a la utilización de estas mismas tecnologías en el desarrollo del proyecto, hace que esta elección sea la más adecuada.

Licencia

La licencia de software bajo la cual el software de la fundación Apache es distribuido es una parte distintiva de la historia de Apache HTTP Server y de la comunidad de código abierto. La Licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original.

La *Free Software Foundation* no considera a la Licencia Apache como compatible con la versión 2 de la *GNU General Public License* (GPL), en la cual el software licenciado bajo la Apache License no puede ser integrado con software distribuido bajo la GPL

Sin embargo, la versión 3 de la GPL incluye una provisión (Sección 7e) que le permite ser compatible con licencias que tienen cláusulas de represalia de patentes, incluyendo a la Licencia Apache.

Aptana Studio

Aptana Studio ha sido el entorno de desarrollo elegido debido a que se trata de una distribución focalizada en el desarrollo para aplicaciones Web 2.0, con soporte a HTML, CSS y Javascript, así como otras tecnologías utilizadas como PHP o AJAX.

La versión utilizada es la Aptana Studio Community Edition, que es la versión gratuita, y contiene la mayoría de las funcionalidades del IDE, como edición, debugging, sincronización y administración de proyectos. Con soporte para todas las tecnologías que veníamos comentando.

La versión utilizada se encuentra licenciada bajo una licencia dual: GPL V3 y la freeware APL (Aptana Public License).

4.2 - Lenguajes utilizados

PHP

Aunque la cantidad de ventajas que ofrece PHP es enorme, las que han propiciado su elección como lenguaje de programación Web del lado del servidor han sido las siguientes:

- Completamente orientado al desarrollo de aplicaciones Web dinámicas con acceso a información almacenada en una Base de Datos.
- El código fuente escrito, es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destacando su conectividad con MySQL y PostgreSQL, las dos opciones principales para utilizar en el proyecto.
- Capacidad de expandir su potencial mediante la utilización de módulos.
- Posee una amplia documentación.
- Es libre.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- Tiene manejo de excepciones.

JavaScript

Es el lenguaje de programación utilizado para aplicar dinamismo a la parte del cliente que opera desde el navegador Web. Es utilizado tanto para aplicar dinamismo a la interfaz Web mediante la implementación DOM de la que viene provisto y el lenguaje CSS, como para realizar peticiones de datos de forma asíncrona mediante AJAX al servidor. Gracias al uso de este lenguaje combinado con la tecnología AJAX, podemos obtener y actualizar dinámicamente el contenido de la Web mostrada, sin necesidad de volver a cargar la pagina al completo.

Perl

Perl es un lenguaje de propósito general originalmente desarrollado para la manipulación de texto y que ahora es utilizado para un amplio rango de tareas incluyendo administración de sistemas, desarrollo Web, programación en red, y más.

Precisamente estas características son las que lo hacen idóneo para implementar los scripts que obtendrán los datos de los sistemas y realizarán la comunicación vía sockets con las interfaces desarrolladas tanto en este mismo lenguaje como en PHP. También se ha tenido en cuenta para su elección su integración y las facilidades que ofrece para trabajar en entornos Linux.

HTML y CSS

Como no podía ser de otra manera, al tratarse de una aplicación con interfaz Web, la base de la implementación y el diseño y posicionamiento que se muestran a través del navegador se ha realizado con estos dos lenguajes.

AWK

Para la obtención de algunos datos relacionados con el entorno del sistema operativo se ha utilizado este lenguaje de programación conjuntamente con comandos Bash.

4.3 - Protocolo SNMP vs. diseño propio

Inicialmente se estudió la posibilidad de usar el protocolo simple de administración de red SNMP para realizar las tareas de monitorización de equipos dado que este protocolo ofrece gran facilidad para obtener información de administración de dispositivos de red.

Para poder obtener dicha información de los distintos dispositivos que nos interesen, debemos configurar el respectivo demonio o servicio de SNMP en las máquinas cliente o dispositivos administrados. Una vez configurado el cliente de SNMP en los dispositivos a monitorizar, podemos mediante polling, obtener de estos la información necesaria para mostrarla en nuestra aplicación. También podemos configurar en ellos lo que se conoce como Traps, que son un comando de notificación usado por los dispositivos administrados para reportar eventos en forma asíncrona. Cuando un cierto tipo de evento ocurre, es decir, cuando alguno de los parámetros que estamos controlando, se escapa del margen acotado, el dispositivo administrado envía una notificación a nuestra aplicación central (NMS). Mediante la correcta configuración de los Traps, solucionaríamos fácilmente el sistema de alertas del proyecto que nos ocupa.

Además, dada la estandarización de este protocolo, podríamos llegar a monitorizar no tan solo servidores, sino workstations, impresoras, routers, y una extensa lista de dispositivos de red con unos mínimos cambios.

Hasta aquí todo han sido halagos hacia este protocolo, por lo que en el momento de realizar los estudios previos al inicio del diseño y codificación del proyecto, SNMP era el candidato ideal sobre el cual basar el sistema de monitorización.

Pero eso solo fue sobre el papel, ya que en el momento de realizar pruebas con las primeras versiones de la aplicación, se vio de forma clara que los resultados de estas no eran satisfactorios, pues imposibilitaban o complicaban en exceso algunos de los requisitos funcionales y/o características con las que se deseaba que contara el proyecto. Pero fueron concretamente dos, las que impulsaron la decisión de eliminar el uso del protocolo SNMP.

La primera fue la intención de crear una aplicación cliente que fuera lo más sencilla posible de instalar y configurar en los dispositivos a monitorizar. En los primeros test realizados, se pudo observar, que la instalación en entornos Windows era relativamente sencilla, pero al dar el paso a entornos con sistema operativo Linux, la configuración no resultaba tan trivial, llegando a complicarse en exceso en algunas distribuciones concretas. Con el uso de una aplicación cliente propia, nos vemos obligados a descartar su uso en dispositivos como routers, impresoras, y equipos que no corran un sistema operativo Linux, o al menos momentáneamente, dejándolo para posteriores líneas de ampliación del proyecto. Sin embargo, la instalación y configuración se convierte en un proceso muy sencillo en cualquier entorno Linux.

La segunda característica que impulso el diseño de aplicaciones cliente propias, fue el deseo de tener total libertad para controlar el funcionamiento y la obtención de datos deseados. El uso de los Traps del protocolo SNMP, no se acababa de adaptar al funcionamiento deseado para el sistema de alarmas del sistema de monitorización.

Con este cambio se perdió la gran facilidad de diseño y desarrollo que nos brindaba el uso del protocolo SNMP, e hizo aumentar considerablemente la complejidad del proyecto. Por otra parte se gana flexibilidad absoluta ya que se al diseñar y programar desde cero las aplicaciones tanto del cliente, como del servidor, y como consecuencia diseñar y decidir el proceso de entre ambos.

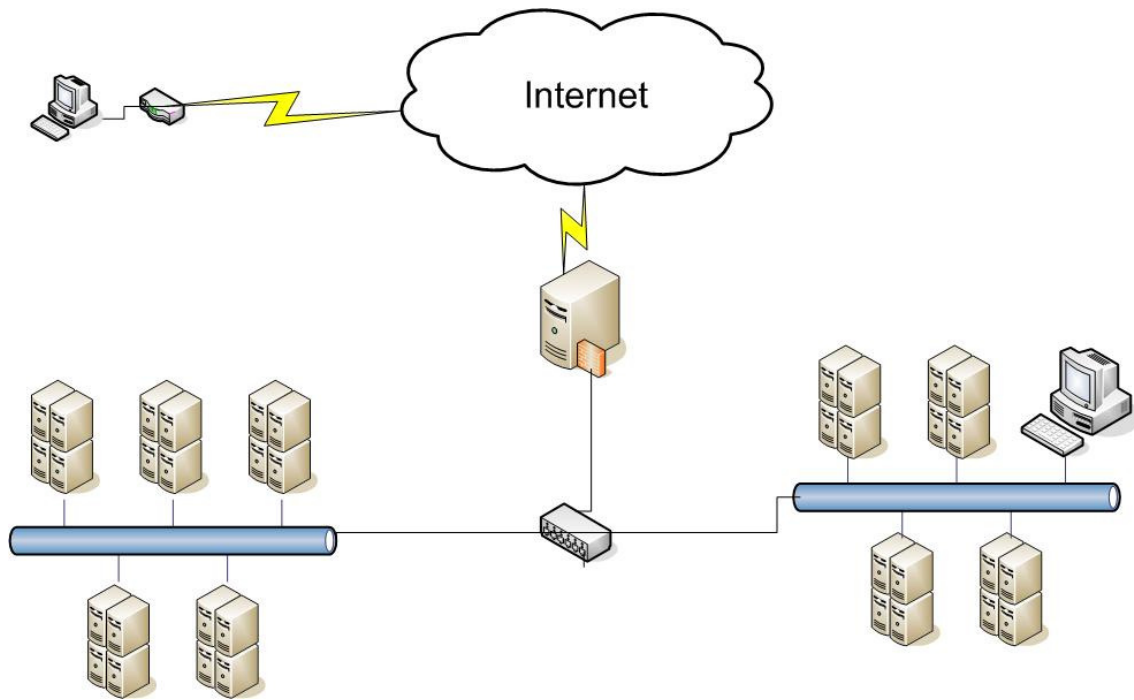
4.4 – Diseño del sistema

Una vez tomada la decisión de abandonar el uso del protocolo SNMP para obtener datos de los dispositivos a monitorizar, se optó por diseñar una nueva arquitectura de comunicación con dos premisas a cumplir bajo cualquier concepto:

- **Scripts cliente-servidor:** La mejor opción para obtener información de los dispositivos de red, ha sido desarrollar un script en Perl que, corriendo como demonio en los nodos a monitorizar, atenderá las peticiones que lleguen desde el servidor, y servirá los datos correspondientes. Con esto podremos monitorizar su estado y configurarlo desde el servidor central.

Por otra parte, tendremos otro script funcionando en el servidor central, el cual atenderá las peticiones que le envíe cualquiera de los nodos monitorizados y efectuará el proceso correspondiente. Con este segundo caso, nos referimos básicamente al proceso de alertas, ya que serán los propios nodos los que tengan conciencia de que aspectos de su sistema han de controlar, y sobre que parámetros deberán enviar una alerta. Por su parte el servidor central, recibirá dicha alerta y la procesará correspondientemente.

- **Acceso desde Internet:** Otro de los principales propósitos que hay que respetar, es el acceso a la aplicación, no solo localmente desde una estación de trabajo, o servidor, sino desde cualquier nodo de la red, o incluso desde fuera de ella, por ejemplo desde Internet. Para ello el frontend de la aplicación deberá basarse en un interfaz Web, accesible desde cualquier navegador.



4. 5 - Base de datos

MySQL ha sido el sistema gestor de base de datos elegido para la implementación de la base de datos de la aplicación.

A parte de las características que de sobras conocidas que hacen de MySQL un potente gestor, hay que mencionar la integración, compatibilidad y facilidad de uso que proporciona conjuntamente con el servidor Apache y el lenguaje PHP. Además de ser un producto licenciado bajo GPL.

Debido al funcionamiento de la aplicación no ha sido necesario crear un complejo diseño, ni establecer relaciones, pues con un diseño simple cubre a la perfección las necesidades básicas. En el caso de posteriores ampliaciones que se comentaran más adelante si que sería necesario modificar el diseño, estableciendo relaciones entre las tablas y añadiendo nuevas.

Base de datos PFC

El conjunto de tablas que almacenan la totalidad de la información que debe manejar la aplicación del lado del servidor, se encuentran reunidas en la base de datos llamada PFC.

A continuación veremos el conjunto de tablas que forman PFC, enumerando y describiendo los campos que las forman.

Tabla nodos

En esta tabla almacenamos toda la información necesaria sobre cada nodo que es dado de alta en la aplicación.

Campo	Tipo	Null	Key	Extra
id	int(5)	NO	Primaria	auto_increment
nombre	varchar(14)	SI		
ip	varchar(15)	SI		
mac	varchar(17)	SI		
tipo	varchar(20)	SI		
descripcion	varchar(150)	SI		

id: es el identificador único.

nombre: nombre de red del host.

ip: dirección IP del host.

mac: dirección MAC del host

tipo: nos informa del tipo host (workstation, server, ...)

descripcion: En este campo es posible añadir información relativa al host, como podrían ser características, funcionalidades que ofrece, fecha de alta o ubicación física.

Tabla usuarios

Campo	Tipo	Null	Key	Extra
id	int(4)	NO	Primaria	auto_increment
nombre	varchar(14)	SI		
password	varchar(14)	SI		
descripcion	varchar(60)	SI		
privilegios	tinyint(4)	SI		
ultimolog	date	SI		
activo	tinyint(1)	SI		

id: identificador único. Es autoincremental, por lo que se incrementara de forma automática cada vez que añadamos un nuevo usuario en el sistema.

nombre: nombre del usuario.

password: password del usuario.

descripcion: campo en el que podremos introducir una breve descripción sobre el usuario, como fecha de alta, fecha de baja, tipo de usuario, etc.

privilegios: mediante este campo se asignan los permisos del usuario a la hora de interactuar con la aplicación. Se ha elegido un valor entero y no un booleano para poder facilitar posibles ampliaciones de inclusión de nuevos perfiles de usuario.

ultimolog: última fecha en la que el usuario realizo un login en el sistema.

activo: nos indica si el usuario se encuentra activado o desactivado, permitiendo denegar el acceso al sistema a un usuario sin necesidad de eliminar su perfil.

Tabla alarmas

Campo	Tipo	Null	Key	Extra
id	int(5)	NO	Primaria	auto_increment
tipo	varchar(8)			
fecha	varchar(10)			
hora	varchar(5)			
umbral	varchar(10)			
consumo	varchar(10)			
alarma	varchar(6)			
host	varchar(14)			
atendida	varchar(1)			

id: identificador único. Es autoincremental, por lo que se incrementara de forma automática cada vez que añadamos un nuevo usuario en el sistema.

tipo: especifica si se trata de una alarma por consumo de CPU, consumo de memoria RAM, espacio en disco, o si se ha alterado el estado de un servicio monitorizado.

fecha y hora: especifica la fecha y la hora en la que se produjo la alarma en el equipo cliente.

umbral: es el umbral que especificamos para que una vez sobre pasado dicho consumo o cambiado el estado de un servicio, será cuando el equipo cliente genere una alarma.

consumo: consumo real, en el momento en el que se sobrepasa el umbral.

alarma: especifica si debemos tratar el sobrepaso del valor umbral como un aviso o como una alerta.

host: nombre del host en el que se ha producido la alarma

atendida: especifica si la alarma ha sido avistada por el usuario y atendida.

4.6 - Diagramas de casos de uso

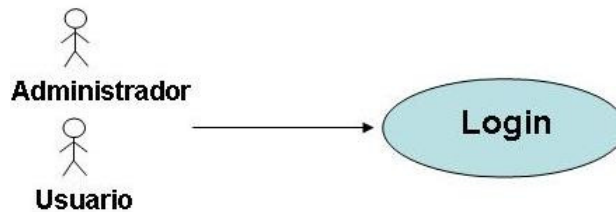
A continuación se exponen los diagramas de los casos de uso diseñados para establecer los requisitos de comportamiento. El diseño de estos es primordial en el diseño de cualquier proyecto para permitirnos capturar los requisitos potenciales del nuevo sistema a desarrollar.

Como veremos, cada caso de uso nos proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir el objetivo específico de cada uno.

Acceso a la aplicación

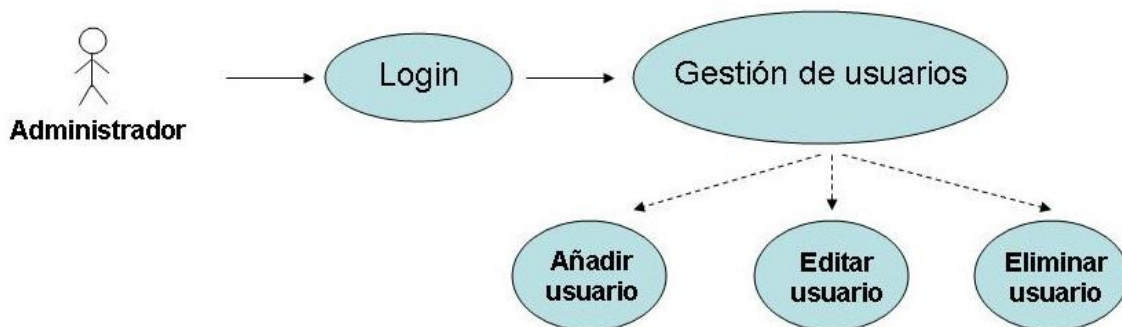
En el momento de conectar con la aplicación, la interfaz inicial es un interfaz de login. Para poder acceder, los usuarios deberán identificarse mediante un nombre de usuario y un password. Los datos serán verificados, permitiendo o denegando el acceso, dependiendo de si el usuario es válido en el sistema o no.

Los actores pueden ser tanto administradores, como usuarios.



Gestión de usuarios

Los usuarios del sistema, únicamente pueden ser gestionados por un administrador. Los usuarios sin permisos no podrán añadir, eliminar o editar usuarios, ni tan siquiera tendrán acceso a la información completa de los mismos. Únicamente podrán visualizar los listados de usuarios en los que se muestra un breve resumen de los datos de cada uno.

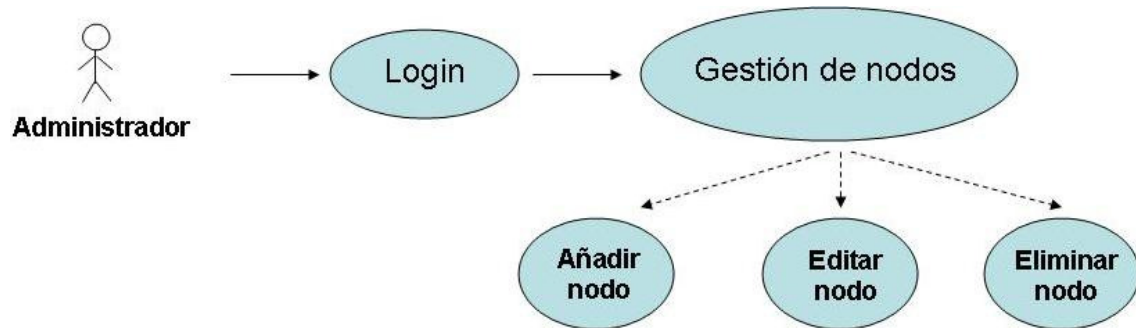


Gestión de nodos

Para poder acceder a las interfaces correspondientes y gestionar los nodos del sistema, el usuario deberá tener permisos de administrador.

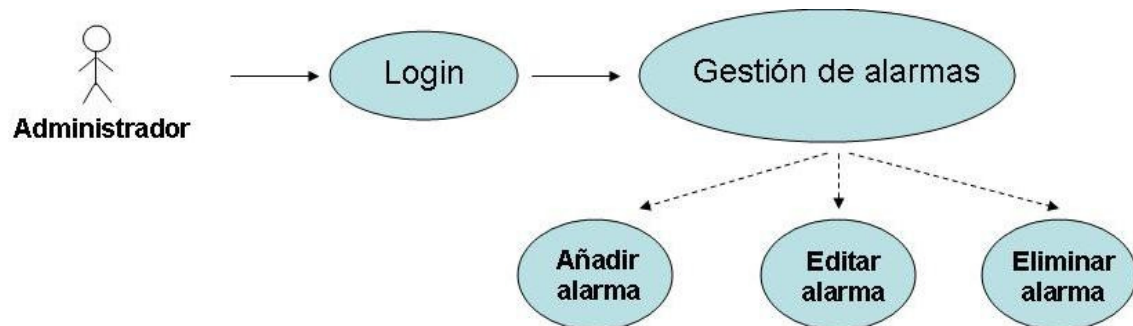
Al contrario que en el caso del acceso a la información de los usuarios del sistema, en este caso, el perfil de usuario simple, si que podrá acceder sin restricciones a los listados, y a la totalidad de la información relativa a los nodos dados de alta en la aplicación. Sin embargo, no podrá añadir nuevos nodos, ni eliminar o editar los existentes.

En las interfaces de monitorización en tiempo real, las opciones que puedan comprometer la estabilidad de los nodos observados, permanecerán desactivadas para excepto para los usuarios con perfil de administrador.



Gestión de alarmas

Únicamente los usuarios con perfil de administrador, podrán gestionar las alarmas del sistema.



5 - Implementación

A lo largo de este capítulo, comentaremos los aspectos más técnicos relativos a la implementación del proyecto.

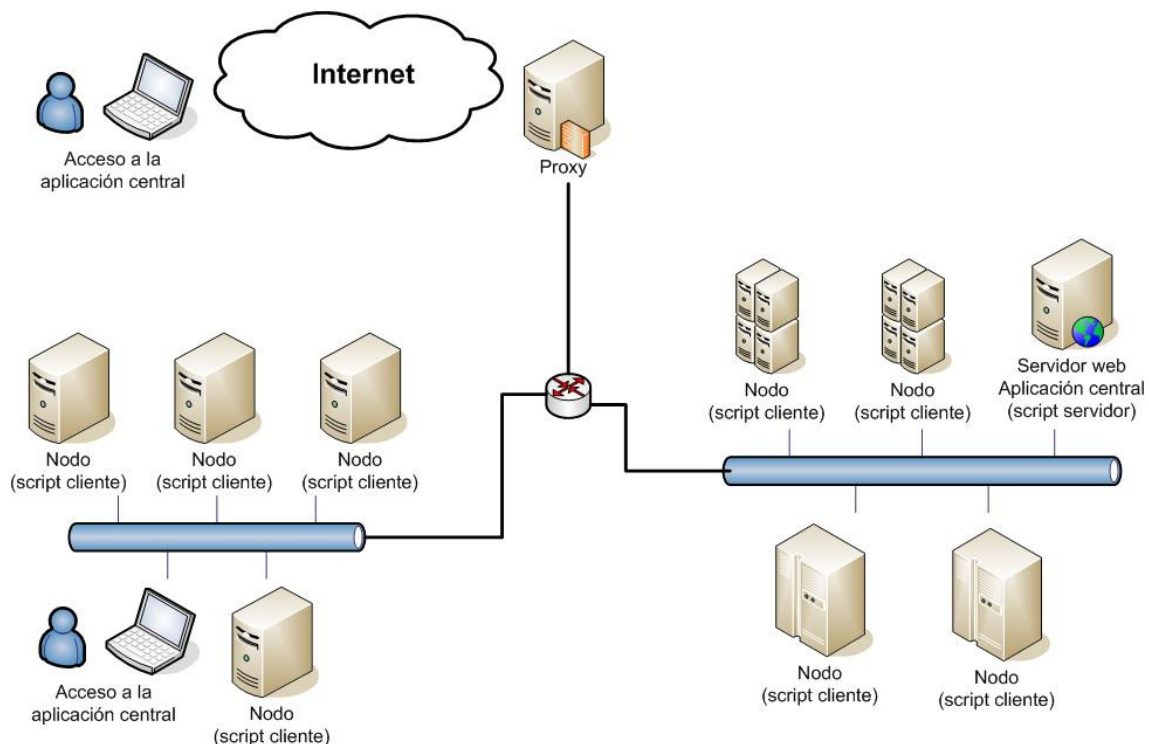
Comenzaremos detallando la arquitectura de la aplicación y su funcionamiento desde un punto de vista más general, para ir profundizando en cada uno de los apartados más significativos de la misma.

Una vez vista la arquitectura, nos centraremos en la interfaz de usuario, en la que veremos las funcionalidades a medida que detallamos el mapa web de la aplicación.

5.1 - Arquitectura

Como hemos visto en capítulos anteriores, el proyecto consta de una aplicación central, y unas aplicaciones cliente desarrolladas en Perl, que funcionan en los nodos monitorizados.

El planteamiento inicial es un modelo cliente-servidor, aunque por temas funcionales, no siempre actúa como servidor la aplicación central y los nodos como clientes, sino que por ejemplo para la monitorización de los nodos, es la aplicación central, la que actúa como cliente realizando la petición de datos y estos como servidores devolviendo la información.



5.1.1 - Aplicación central

Se trata del núcleo del proyecto, y consta de los scripts Perl, las interfaces Web, y la base de datos. Principalmente todo el conjunto debe hallarse instalado en el mismo servidor.

El acceso se lleva a cabo desde un navegador Web, desde donde podremos acceder a las interfaces de usuario. Desde estas, y según en la parte de la aplicación en la que nos encontremos, nos mostrara la información correspondiente, obteniéndola de los host mediante peticiones directas a través de sockets en el caso de la monitorización de nodos, de la base de datos ubicada en el propio servidor, o de ficheros temporales también ubicados en el propio servidor.

5.1.2 - Script servidor

Se trata de un script sencillo, cuya única finalidad es crear un socket que permanece a la escucha de peticiones entrantes.

Dichas conexiones se producen en el momento que se genera un aviso o una alerta en cualquiera de los nodos, y estos envían la información relativa a lo sucedido a este script. La información simplemente es almacenada en un fichero temporal, para ser procesada por la aplicación central, que a intervalos de tiempo predefinidos, analiza la existencia y el contenido de dicho fichero.

En el caso de que la aplicación central encuentre alguna alarma almacenada en el fichero temporal, la procesara, decidiendo según su nivel de prioridad que acciones tomar, como enviar un e-mail en el caso de los avisos, o un sms en el caso de las alarmas. Una vez llevadas a cabo estas acciones, la información es guardada en la base de datos y publicada inmediatamente en la interfaz gráfica. Tal y como veremos más adelante, cuando comentemos las interfaces Web, veremos que en cada una de ellas hay un espacio reservado para la publicación de estos sucesos.

5.1.3 - Script nodos

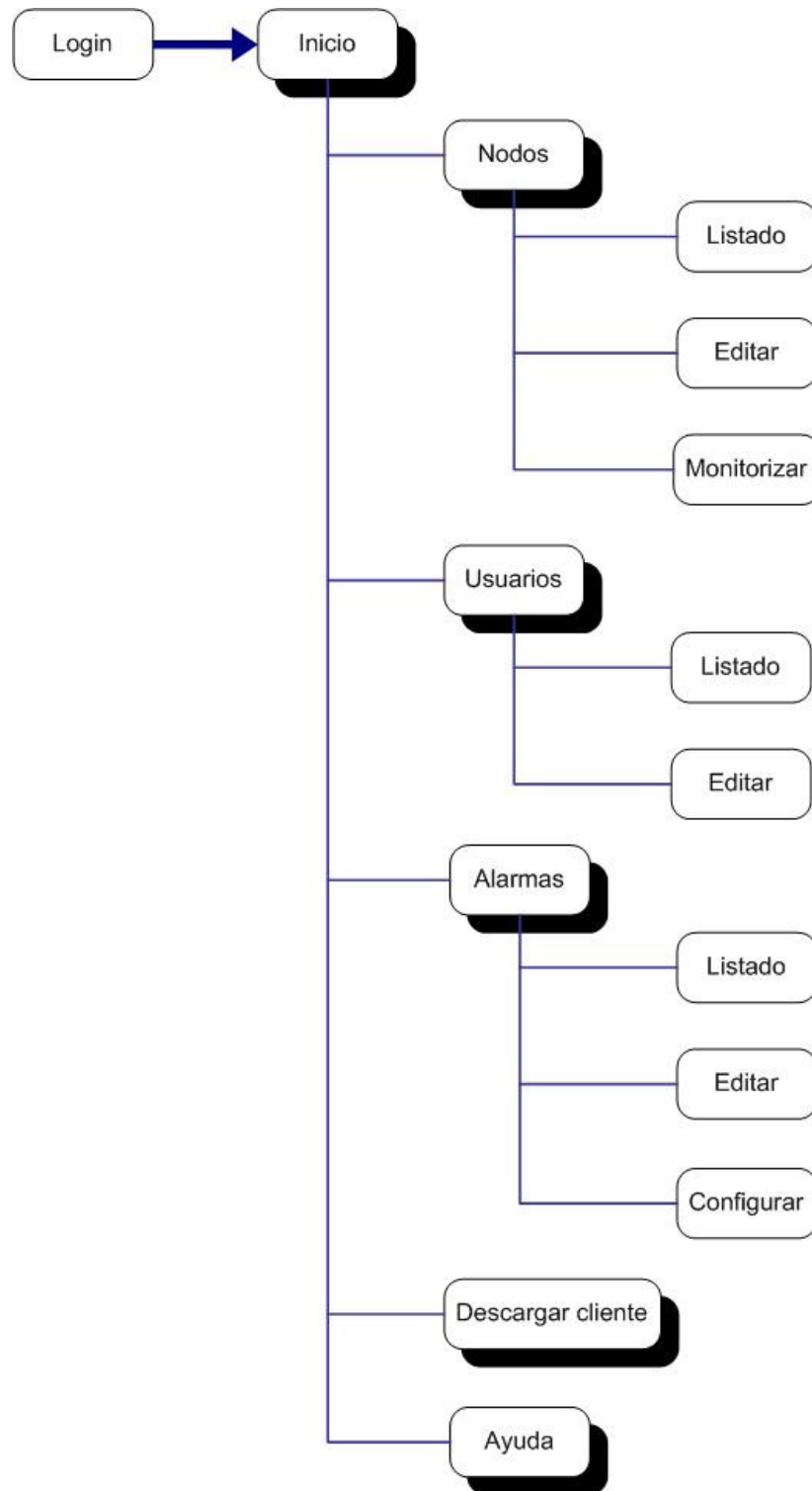
Este script es el que se encuentra procesándose en todas y cada una de las estaciones monitorizadas. Como hemos comentado anteriormente, el script ha sido desarrollado en Perl, y es iniciado como un demonio del sistema. Consta de diversos archivos que constituyen las librerías donde se encuentran todas las funciones adicionales que dan soporte al script principal.

A pesar de que es iniciado desde un mismo fichero, consta de dos aplicaciones que trabajan por separado. La primera es la encargada de crear un socket que permanece a la espera de peticiones por parte del servidor. Dichas peticiones pueden ser para obtener información sobre el estado del host, o bien para modificar su configuración, como sucede en el caso de las alarmas, o en el listado de servicios del sistema de los que deseamos visualizar su estado.

La segunda aplicación es la encargada de manejar los procesos relativos al sistema de alarmas. Esto es debido a que tanto las alertas como los avisos, no están controlados desde la aplicación central, sino que es cada nodo el que se responsabiliza de su propia monitorización en este aspecto. Es por ello que cada cierto intervalo de tiempo predefinido, realiza las comprobaciones sobre su propio estado, comparándolo con los umbrales de alarma establecidos, y decidiendo si enviar un aviso, una alerta, o si todo esta funcionando correctamente. En el caso de que alguno de los umbrales configurados se vea rebasado, el script creara un socket que conectara con el script ubicado en el servidor central para enviarle la información de lo que esta sucediendo para que este segundo lo valore y sea procesado adecuadamente, realizando las acciones pertinentes.

Obviamente cada uno de ellos trabaja en puertos distintos, para evitar que puedan producirse denegaciones de servicio.

5.2 - Mapa Web



5.3 - Descripción de interfaces

5.3.1 - Login

Interfaz inicial de la aplicación, que proporciona el control de acceso a la misma.

El control de acceso se permite mediante la validación de un nombre de usuario y su password correspondiente. Tras comprobar la existencia de texto en los campos, y su formato, enviamos la información de forma asíncrona utilizando la tecnología AJAX, a la pagina “login.php”, la cual mediante el acceso a la base de datos de usuarios, nos confirmará si son o no correctos la pareja nombre de usuario y password, devolviendo dicha respuesta de forma asíncrona a la interfaz, que permitirá el acceso a la aplicación o nos denegara el acceso informándonos del tipo de error ocurrido.

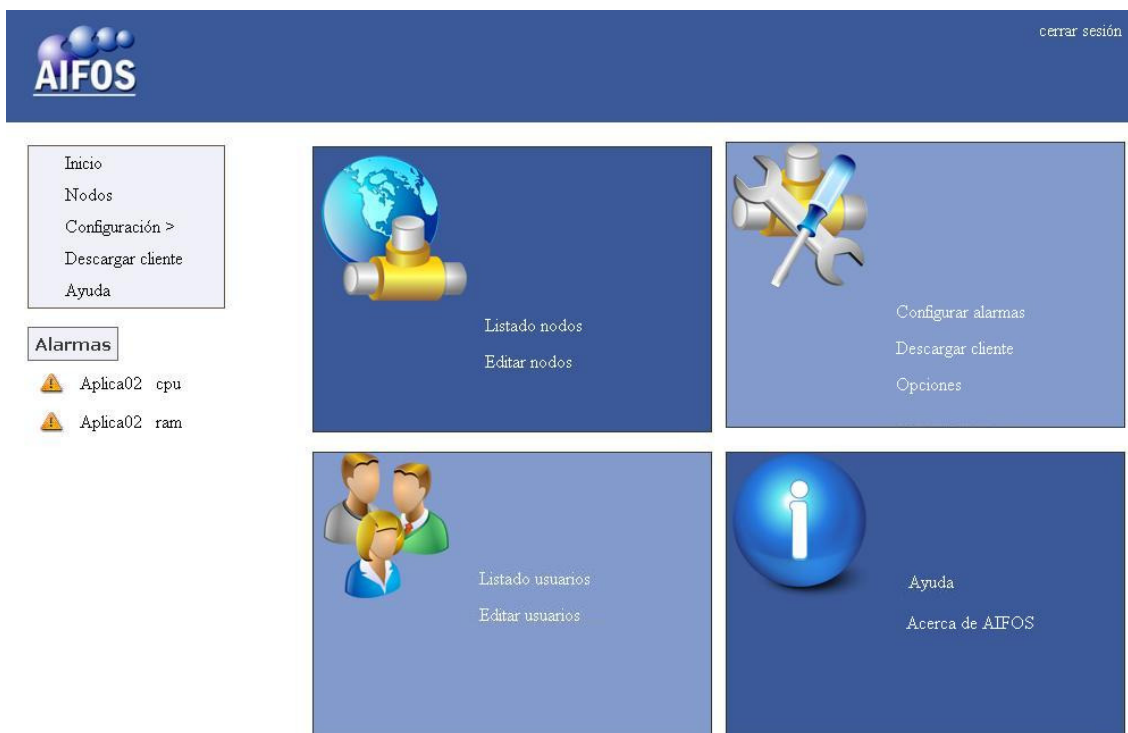


Usuario:

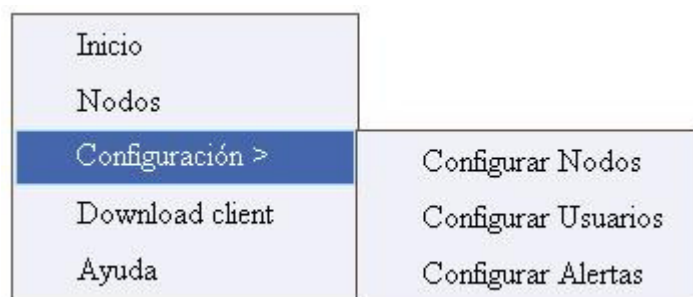
Password:

5.3.2 - Home

Esta se trata de la página inicial de la aplicación, la cual nos sirve de guía, facilitando el acceso a las diversas funcionalidades de la aplicación, separando el acceso a las mismas de forma temática.



En esta interfaz vemos por primera vez el menú situado en la parte izquierda, que nos acompañara en todas las interfaces posteriores, facilitando en todo momento el acceso a cualquier funcionalidad de la aplicación.




Al igual que en el resto de interfaces y situado debajo del menú, nos encontramos con la ventana de alarmas, en la que se nos mostraran en tiempo real las alarmas que se hayan producido y que aún no hayan sido marcadas como atendidas por algún usuario. En esta ventana se muestra un pequeño resumen de la alarma, indicando la gravedad de la misma, si se trata de un aviso o una alerta, el nombre del host en que se ha producido y el tipo de campo monitorizado al que afecta: consumo de cpu, memoria RAM, espacio de disco, o servicios. Desde cualquiera de los avisos que nos aparecen, podemos acceder mediante el link correspondiente, a la pantalla de publicación de alarmas, en la que veremos con todo detalle la información de lo ocurrido. Profundizaremos más en este tema y en las funcionalidades que ofrece el sistema de alarmas en el apartado dedicado a su interfaz.



En el momento de acceder a la aplicación, se iniciara una sesión con el perfil de usuario con el se ha autenticado en la pantalla de login. Esta sesión se mantendrá para todo el conjunto de páginas que forman la aplicación, permitiendo en cada caso el acceso a las funcionalidades pertinentes, según el perfil de usuario y los permisos del mismo. Es por eso que en el encabezado superior, se nos mostrará siempre la opción de “cerrar sesión”, para poder abandonar de forma controlada la aplicación.

5.3.3 - Listado de Nodos

En esta interfaz se nos muestra un listado de todos los nodos dados de alta en la base de datos, junto con una breve descripción de cada uno, los datos más relevantes y el estado de su conexión a la red. Dicho estado es mostrado en tiempo real gracias a la implementación de funciones que hacen uso de la tecnología AJAX que van comprobando periódicamente la conexión de dichos nodos y modificando su estado inmediatamente en caso de que este cambie, sin necesidad de volver a cargar la página. A través de esta interfaz podemos acceder a la interfaz encargada de mostrar los parámetros monitorizados en tiempo real, mediante el link incluido en el nombre del host. También podemos acceder mediante los links correspondientes, a la edición de la información del nodo, o a la configuración de alarmas del mismo.


cerrar sesión

Inicio

Nodos

Configuración >

Descargar cliente

Ayuda

Nombre	IP	MAC	Tipo	Estado		
walu-laptop	127.0.0.1	00:13:02:dd:9e:f7	servidor		Editar	Alarmas
<u>Aplica02</u>	192.168.0.130	00:17:de:48:a5:08	servidor		Editar	Alarmas
Mail01	192.168.0.5	00:17:12:54:a5:09	servidor		Editar	Alarmas
Aplica01	192.168.0.129	00:17:12:58:a5:07	servidor		Editar	Alarmas
Aplica01	192.168.0.6	00:54:dd:48:a5:01	servidor		Editar	Alarmas

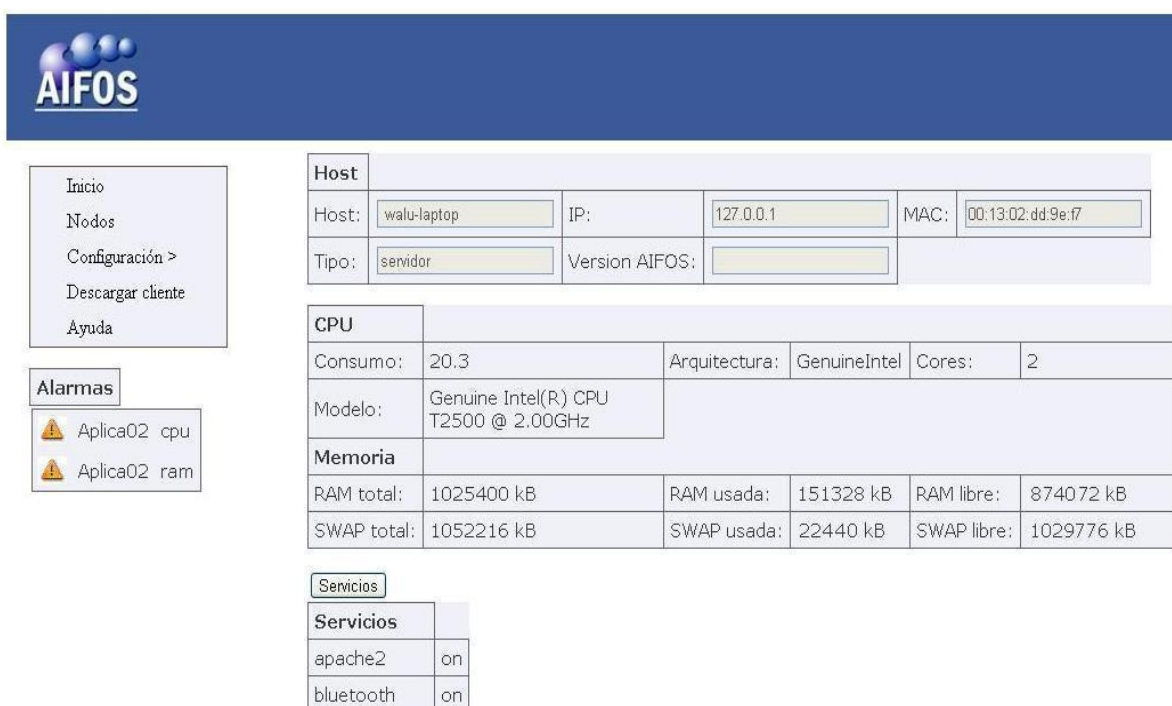
Alarmas

Aplica02 cpu
 Aplica02 ram

5.3.4 - Monitorización de nodos

Esta es una de las interfaces que más relevancia tiene dentro de la aplicación. En ella se nos muestran los parámetros monitorizados en tiempo real.

Tal y como hemos comentado anteriormente, esta interfaz también implementa funciones que hacen uso de la tecnología AJAX, para poder mostrar dinámicamente los cambios producidos en el estado del nodo, sin necesidad de cargar de nuevo la página Web. Esto se consigue realizando la petición, recepción y muestreo de los datos mediante conexiones de forma asíncrona, lo cual produce un resultado dinámico y totalmente transparente para el usuario.



The screenshot displays the AIFOS monitoring interface. On the left, there is a navigation menu with options: Inicio, Nodos, Configuración >, Descargar cliente, and Ayuda. Below this is an 'Alarmas' section showing two active alarms: 'Aplica02 cpu' and 'Aplica02 ram', both indicated by yellow warning icons. The main content area is divided into several sections:

- Host**: A table showing host details.

Host:	walu-laptop	IP:	127.0.0.1	MAC:	00:13:02:dd:9e:f7
Tipo:	servidor	Version AIFOS:			
- CPU**: A table showing CPU usage and architecture.

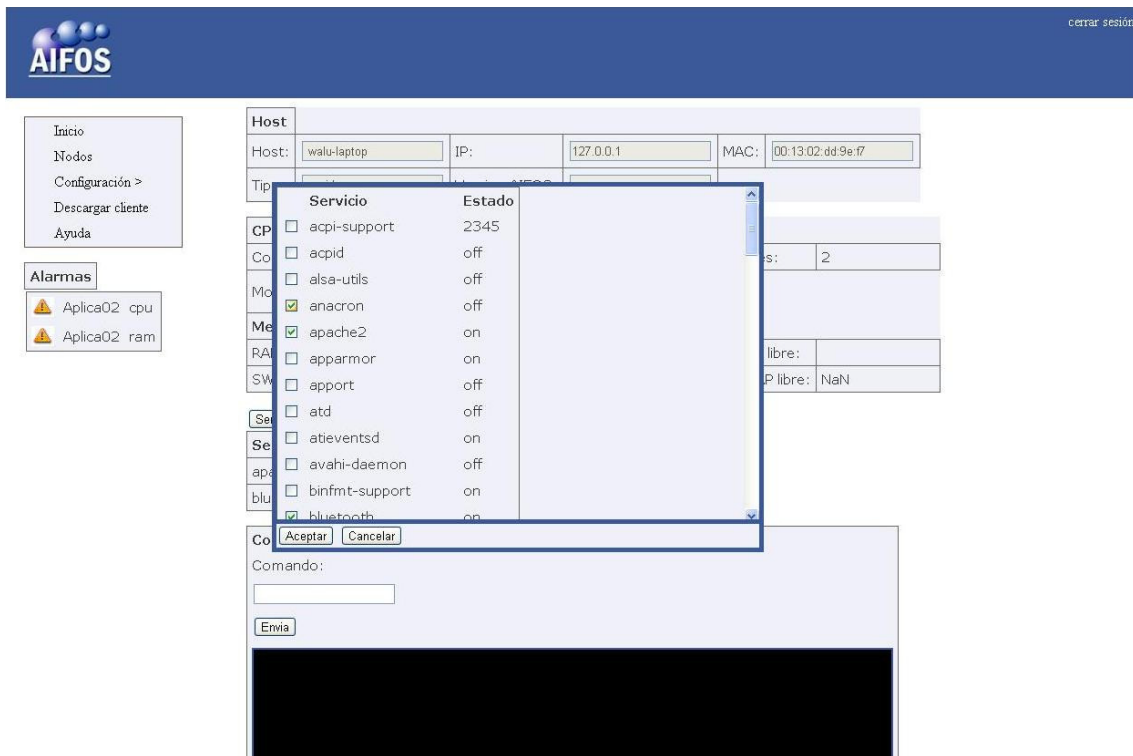
Consumo:	20.3	Arquitectura:	GenuineIntel	Cores:	2
Modelo:	Genuine Intel(R) CPU T2500 @ 2.00GHz				
- Memoria**: A table showing memory usage.

RAM total:	1025400 kB	RAM usada:	151328 kB	RAM libre:	874072 kB
SWAP total:	1052216 kB	SWAP usada:	22440 kB	SWAP libre:	1029776 kB
- Servicios**: A table showing the status of services.

apache2	on
bluetooth	on

La información relativa a la descripción del equipo, como el nombre de host, tipo de estación, etc. se consigue a través de una consulta a la base de datos ubicada en el servidor, mientras que la obtención de el resto de parámetros monitorizados, se obtienen directamente del nodo, a través de la creación de sockets, conectando directamente con la aplicación cliente instalada en el nodo monitorizado y publicando los datos.

Desde esta interfaz también podemos configurar los servicios que deseamos monitorizar, pudiendo seleccionarlos desde un listado que nos muestra todos los servicios existentes en el equipo. En la nueva ventana, se nos muestra tanto los servicios hallados en el sistema remoto, como su estado. Una vez seleccionados, se incluirán o se eliminarán de la monitorización en tiempo real.



En el caso de tratarse de un usuario con permisos de administrador, aparecerá en la interfaz un nuevo grupo de opciones que nos permitirán ejecutar comandos de consola en la estación monitorizada y obtener los resultados producidos por la ejecución del mismo, como si de una consola virtual se tratara. Por ejemplo, si ejecutáramos el comando 'ifconfig', se mostrara los detalles de las interfaces de red del nodo monitorizado. A continuación exponemos una captura de pantalla de la aplicación en la que se puede apreciar el resultado del envío del comando date.

Consola Virtual

Comando:

jue jul 8 00:37:57 CEST 2010

5.3.5 - Configuración de los nodos

Interfaz sencilla en la que se nos muestran todos los datos relativos a un nodo concreto, permitiéndonos la edición de los mismos.

AIFOS

Inicio
Nodos
Configuración >
Descargar cliente
Ayuda

Alarmas

⚠ Aplica02 cpu
⚠ Aplica02 ram

ID:

Nombre:

IP:

MAC:

Tipo:

Descripción:

Estado:

Inicialmente se muestra la información del nodo solicitado, y mediante las opciones situadas en la parte superior del formulario, podemos editarlas. Tanto la edición, como la creación o eliminación de nodos, utiliza la tecnología AJAX, para proporcionar al usuario una experiencia más agradable, al no tener que recargar la página cada vez que se lleve a cabo una acción.

5.3.6 - Listado de usuarios

Muestra un listado de todos los usuarios dados de alta en la aplicación, permitiendo añadir nuevos usuarios, eliminar existentes, y acceder a toda la información relativa a un usuario concreto, para poder consultar o modificar dicha información.



Nombre	Privilegios	Descripción	Activo	
Victor	1	Administrador	0	Editar
Carlos	2	Usuario	0	Editar

5.3.7 – Configuración de usuarios

De igual forma que con la configuración de nodos, esta interfaz nos muestra los datos relativos a un usuario concreto, permitiéndonos la edición del mismo.

Inicialmente se muestra la información del nodo solicitado, y mediante las opciones situadas en la parte superior del formulario, podemos editarlas. Tanto la edición, como la creación o eliminación de nodos, utiliza la tecnología AJAX, para proporcionar al usuario una experiencia más agradable, al no tener que recargar la página cada vez que se lleve a cabo una acción.

5.3.8 - Alarmas

Este apartado es conjuntamente con la interfaz dedicada a la monitorización de nodos en tiempo real, uno de los más relevantes dentro del proyecto, pues de encarga de gestionar todo lo relativo al sistema de alarmas que implementa el proyecto. Nos referimos en

este caso al mismo como apartado, en vez de interfaz como hemos echo hasta ahora, debido a que la funcionalidad del sistema de alarmas consta de varias interfaces visibles para el usuario, algunas dedicadas exclusivamente y otras que están incluidas en otras interfaces, como la ventana de alarmas que hemos comentado cuando detallábamos la interfaz 'home'. También consta de varios módulos no visibles para el usuario, que se encargan de revisar periódicamente la existencia de nuevas alarmas, y en el caso de que estas existan, de la generación y envío de las mismas mediante e-mail o sms.

A continuación procederemos a detallar el funcionamiento de las interfaces.

- Listado de alarmas

En esta interfaz se nos muestra un listado de todas las alarmas generadas por los distintos clientes, que aún no han sido atendidas.

Se muestra un breve resumen, en el que podemos observar la gravedad de la alarma, el host en el que se ha producido, el tipo de parámetro al que afecta, la fecha y hora en la que se produjo, el umbral del parámetro que especifico el usuario para esta alarma, y el consumo o estado real que se ha dado.

A través del link ubicado en el nombre del host, podemos acceder a la interfaz de monitorización del nodo, para comprobar el estado del mismo y decidir las acciones correspondientes a tomar, valorando el estado actual. Una vez valorado y llevadas a cabo las actuaciones correspondientes, podemos marcar como atendida la alarma, para que esta desaparezca tanto del listado de alarmas, como de la pantalla presente en todas las interfaces de la aplicación.

- Configuración de alarmas

Desde esta parte de la aplicación, es donde podemos configurar las alarmas para un nodo concreto.

Los aspectos que podemos configurar son:

- Consumo de CPU
- Consumo de memoria RAM
- Espacio libre en disco
- Servicios

Para los consumos de CPU, RAM y disco, la configuración es idéntica, ya que podemos definir un umbral de aviso y otro de alerta. El umbral de alerta siempre deberá ser más crítico que el de aviso. Por crítico se entiende, que por ejemplo en el caso del consumo de CPU, el umbral de alerta debe definir un consumo más elevado que el de aviso. En el caso de la memoria RAM y de espacio en disco, en los que se define la cantidad de memoria libre, el umbral de alerta, siempre deberá ser un valor inferior al de aviso.

En el caso de los servicios, se especificara un servicio existente en el host y se especificara cual es el estado para el que queremos que se nos avise si se produce. Por ejemplo, podemos seleccionar el servicio 'apache2' y seleccionar 'off'. Dicha configuración nos avisara si el estado del servicio 'apache2' pasa a ser 'off', lo que nos indicaría que se ha detenido. Obviamente también se puede configurar a la inversa, por si nos interesara saber si se ha activado un servicio que debería estar inactivo.

Todas las alarmas generadas por servicios, serán tratadas siempre como alertas, otorgándoles de esta manera siempre, el carácter más crítico.

A continuación de las opciones de configuración de las alarmas, nos aparece un listado de las alarmas configuradas en el mismo. Desde este listado podremos eliminar las que no deseemos.

No podremos modificar alarmas existentes, pero tampoco podremos generar dos alarmas distintas para un mismo tipo, como por ejemplo para el consumo de CPU. Por lo que si disponemos de una alarma configurada para el tipo CPU y deseamos configurarla, únicamente deberemos configurar las opciones de la misma, y esta será modificada sin la necesidad de que haya sido eliminada previamente.

Debido al diseño del sistema, dos condiciones indispensables para poder configurar alarmas en un host concreto, son que el equipo este dado de alta en la aplicación, y que en el momento de configurar la alarma, disponga de conexión de red. Estas dos condiciones son valoradas en el momento de acceder a la interfaz de configuración, de manera que si el equipo se encuentra dado de alta y con conexión a red, podremos configurarlo. En caso contrario, se nos avisara del error correspondiente, impidiéndonos el acceso a la interfaz. Esta comprobación es necesaria debido a que es el propio cliente el que almacena sus propias alarmas y se monitoriza a si mismo, interviniendo el servidor únicamente cuando se produce una alarma y esta es enviada desde el cliente, o bien cuando desea acceder al mismo para visualizar su configuración de alarmas.

- Envío de e-mail

Para realizar el envío de e-mail, se ha configurado PHP para usar un servidor local de mail, haciendo uso de la aplicación *sendmail*.

Una vez configurado correctamente, cada vez que la aplicación central recibe una alarma, esta es procesada, y si se trata de un aviso, se enviara un e-mail a la dirección de correo preestablecida a través de la interfaz en PHP correspondiente.

- Envío de SMS

Para el envío de SMS, se ha optado por hacer uso de los servicios ofrecidos por la empresa MENSATEK.

Mediante este sistema se consigue la comunicación entre los servidores de MENSATEK y nuestro servidor, utilizando el protocolo HTTP para el envío de

mensajes SMS. El proceso básico de comunicación es el siguiente: MENSATEK recibe la petición GET o POST de nuestro servidor según los parámetros especificados y procesa la petición.

Las peticiones se pueden realizar por HTTP o HTTPS (conexión segura) y los parámetros pueden ser enviados en peticiones GET o POST.

Para hacer uso de los servicios ofrecidos por MENSATEK, únicamente ha sido necesario incluir la API proporcionada, y crear un script en PHP, detallando los parámetros necesarios como número de móvil o móviles a los cuales deseamos enviar los SMS, remitente y el cuerpo del mensaje.

Para el envío de SMS, es necesario que el nivel de alarma sea de alerta, en cuyo caso se enviara un SMS al número de móvil o móviles especificados, y un e-mail a la dirección correspondiente.

5.3.9 - Descarga del script cliente

Nos permite descargar el script de instalación del cliente de la aplicación.

Este link es de gran utilidad, cuando necesitemos instalar el cliente en una estación en la que nos encontramos trabajando físicamente o mediante conexión remota. Únicamente tendremos que acceder a la aplicación desde cualquier navegador Web, y seleccionar esta opción para poder descargar en el equipo el script y proceder a su instalación.

5.3.10 - Ayuda

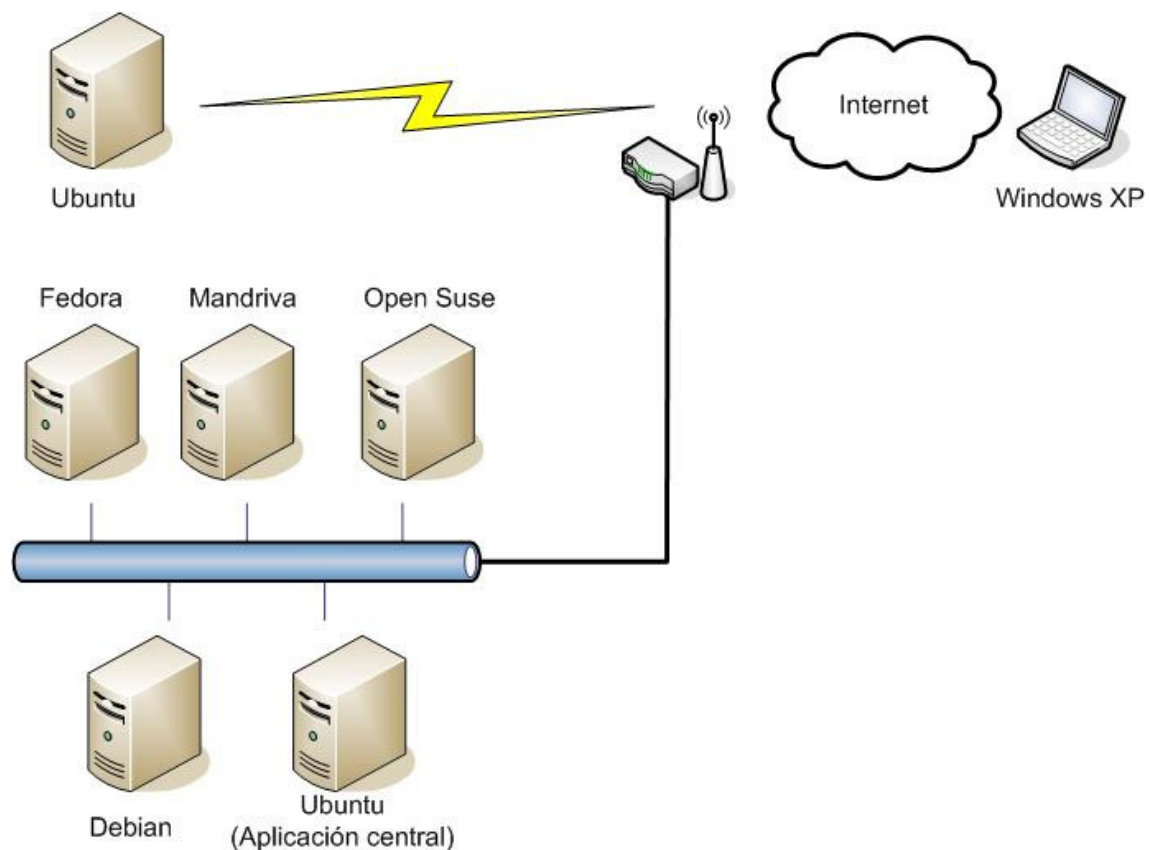
Nos proporciona acceso a la ayuda de la aplicación, que incluye el manual de usuario.

6 - Pruebas

6.1 - Entorno de pruebas

Se ha tratado de diseñar un entorno de pruebas que incluyera todos los aspectos posibles, tanto a nivel de los sistemas operativos utilizados por los servidores, como por la arquitectura de la red, sin olvidar el uso de distintos navegadores para acceder a la aplicación.

Se han utilizado siete estaciones, seis de ellas con distribuciones GNU/Linux y una de ellas con el sistema operativo Windows XP, con la finalidad de probar el acceso desde el navegador Internet Explorer. También se ha dispuesto de un router, que además desempeña el papel de proxy de la red, permitiendo además acceso a Internet. El router utilizado ha sido un Thomson SpeedTouch v6, con el que se ha creado una intranet con cinco equipos conectados físicamente a él, mas uno conectado de forma inalámbrica.



6.2 - Navegadores Web

Como se comento en el estudio de viabilidad, uno de los requisitos del sistema, es que pudiera ser accesible desde cualquiera de los principales navegadores Web existentes en la actualidad. Así pues recordemos que estos eran:

- Mozilla Firefox 3.3 o superior
- Google Chrome 5.0 o superior
- Internet Explorer 6.0 o superior
- Opera 9.0 o superior

Las pruebas realizadas han consistido en acceder a la aplicación desde cada uno de ellos, y simular una sesión de trabajo, en la que se navegara por las distintas interfaces de la aplicación, interactuando con ellas añadiendo o modificando configuraciones.

Todos superaron las pruebas satisfactoriamente. Esto es debido a que tanto las funcionalidades aportadas por la tecnología AJAX, la programación en JavaScript y la maquetación de las interfaces Web, que son comúnmente los apartados en los que suelen aparecer un mayor grado de incompatibilidad entre los distintos navegadores, fueron tenidas en cuenta de antemano.

6.3 - Distribuciones GNU/Linux

Otro de los requisitos iniciales, era que el proyecto fuera compatible con algunas de las distribuciones GNU/Linux, más reconocidas, evitando centrarse únicamente en alguna de ellas.

Para realizar las pruebas se han utilizado versiones estables de las siguientes distribuciones:

- Open Suse 10.2
- Fedora 12
- Mandriva 2010 Spring
- Ubuntu
- Debian 4.0 Etch

A parte de instalar el script cliente en cada uno de los servidores, el único dato que hay que tener en cuenta, es instalar las librerías de Perl adecuadas. En el caso que nos ocupa, las utilizadas fueron las de la versión 5.12.1, por lo que no podemos garantizar compatibilidad con versiones anteriores.

Al tratarse de distribuciones similares, basadas en un mismo núcleo, antes de iniciar el desarrollo del sistema se tuvo en cuenta el estudio de las herramientas como comandos, o los lenguajes soportados por las shell de los sistemas, que serian utilizados para obtener información de los servidores.

Las pruebas realizadas con cada una de las distribuciones ha sido similar a la efectuada para testear los navegadores Web, que consistió en simular una sesión de trabajo con la aplicación interactuando con la totalidad de las interfaces.

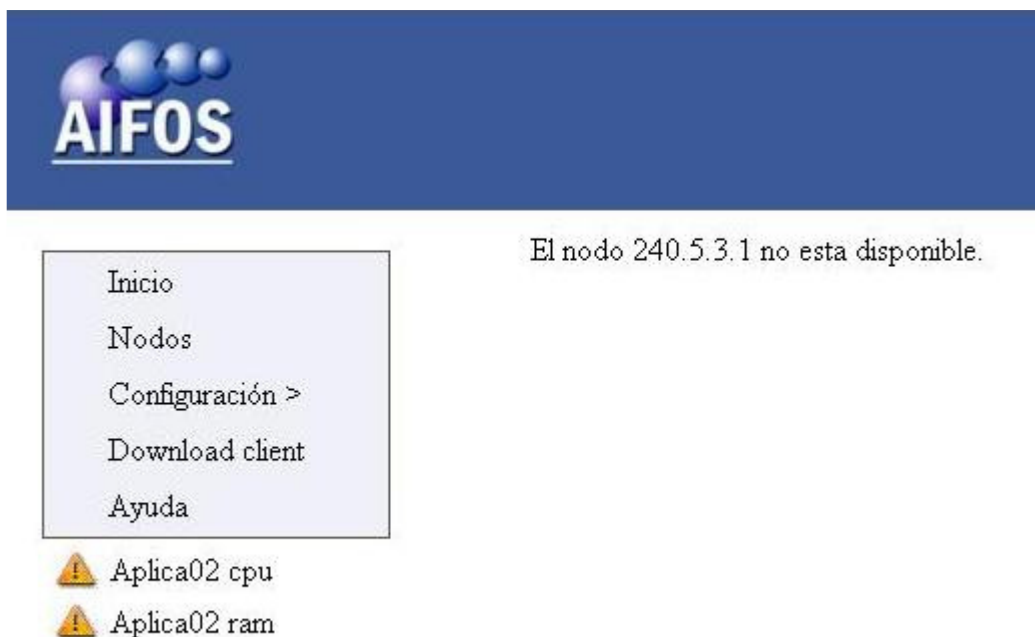
Las pruebas de este apartado fueron superadas correctamente por todas y cada una de las distribuciones.

6.4 - Disponibilidad de la conexión

Este es quizás uno de los apartados más importantes a la hora de testear la aplicación, ya que es sumamente importante para un sistema que trabaja en red, ser tolerante a errores relacionados con la disponibilidad de la conexión de los recursos con los que trabaja.

Dadas las características del sistema, estas pueden darse tanto a la hora de establecer conexión con un equipo, como cuando esta se pierde al estar conectado al mismo. El primer caso podría darse en el momento de establecer la configuración de las alarmas en un servidor, y donde tendríamos que comprobar el estado del equipo antes de permitir que se modifiquen. El segundo caso podría darse, en el momento en que estamos viendo el estado en tiempo real de uno de los host. En este caso ya nos encontramos conectados al equipo, pero que pasaría si por cualquier circunstancia, el equipo se viniera abajo inesperadamente. Es aquí donde, aun a pesar de que tenemos una conexión establecida, debemos detectar que ha cambiado el estado del equipo, controlar el error y informar adecuadamente al usuario.

A continuación, se muestra una captura de pantalla en la que se refleja como ha sido controlado un error de conexión:



Han sido comprobados ambos casos, tanto en distintos equipos, como accediendo desde distintos navegadores Web. La comprobación se ha realizado por una parte tratando de establecer conexión con equipos de los que previamente conocemos están apagados o sin conexión de red, y de igual manera, se han desconectado equipos en los que la aplicación se encontraba monitorizándolos en tiempo real. Todos los resultados han sido satisfactorios, controlando adecuadamente los errores, e informando adecuadamente al usuario.

7 - Conclusiones

7.1 – Conclusiones

Durante la realización del proyecto se ha podido comprobar la dificultad de cumplir la planificación temporal prevista. Inicialmente se había previsto que el tiempo no sería un problema, pero la implicación en ciertos proyectos paralelos, no contemplados durante la planificación del proyecto ha complicado mucho la dedicación temporal.

Las tecnologías de desarrollo de paginas Web ha crecido mucho durante los últimos años, haciendo que la programación de estas sea mucho más fácil obteniendo resultados mucho más interesantes para el usuario final. Aun así, la linealidad aun es demasiado grande, aunque cada vez esta más implantada la programación orientada a objetos, lo que nos aporta una mayor potencia y flexibilidad.

Uno de los objetivos del proyecto, a nivel personal, ha sido el estudio y aprendizaje del funcionamiento de sistemas de monitorización de red. Debido al tremendo interés y curiosidad que siento por el funcionamiento de las redes y por la programación, se ha tratado en todo momento, desarrollar el sistema al nivel más bajo posible, evitando siempre el uso de frameworks o librerías de funciones. La satisfacción una vez finalizado el mismo es indescriptible, tanto por el resultado, como por la cantidad de conocimientos adquiridos.

Por lo tanto la valoración final es muy positiva, dado que se han cumplido en mayor o menor medida todos los objetivos establecidos inicialmente, además de algunos que han ido surgiendo sobre la marcha durante el desarrollo del sistema.

7.2 – Futuras ampliaciones

Han sido bastantes las líneas de posibles ampliaciones futuras que se han ido detectando durante el desarrollo del proyecto. Esto es debido a que como hemos comentado anteriormente en el capítulo dedicado a fundamentos teóricos, un sistema de monitorización de redes, es un concepto de aplicación o sistema bastante general, con lo que puede abarcar multitud de campos y ofrecer infinidad de funcionalidades.

Algunas de las ampliaciones posibles que se podrían desarrollar serían:

- Detección automática de recursos de red: Se trataría de dotar al sistema de un módulo dedicado al escaneo y detección automática de dispositivos de red como servidores, estaciones de trabajo, routers, etc.
- Sistema de logs: Aunque si que es cierto que el sistema de alarmas mantiene un histórico de las alarmas sucedidas, esto mismo se podría aplicar tanto a los nodos, como a los usuarios. También es cierto que en el sistema de logs implementado para el sistema de alarmas podría mejorarse.

- Generar gráficos: Implementar funciones para visualizar estadísticas en formato gráfico. Esta ampliación iría ligada directamente al desarrollo de un sistema de logs.
- Mayor interacción con los nodos monitorizados: Si bien es cierto que la monitorización en tiempo real de nodos de red, ha superado con creces los objetivos establecidos inicialmente, siempre es posible y positivo dotar de mayor control al sistema, tanto a la hora de recabar mayor información, como para permitir una mayor interacción con dichos sistemas.

8 - Bibliografía

Lenguajes de programación

Dave W. Mercer, Allan Kent, Steven D. Nowicki, David Mercer, Dan Squier
Fundamentos PHP5. Ediciones Anaya Multymedia (Grupo ANAYA, S.A.), 2005
ISBN: 84-415-1805-X

Página oficial de PHP: Hypertext Preprocessor. The PHP Group
<http://www.php.net/>

R.Allen Wyke, Luke Duncan
Guía de referencia para programadores de PERL 5
Anaya Multimedia 1997
ISBN: 84-415-0446-6

Peter Wainwright
Professional Perl Programming
Wrox Press Ltd. 2001
ISBN: 1-861004-49-4

Tom Negrino, Dori Smith
Guía de aprendizaje Javascript
Pearson educación S.A. 5ª Edición 2005
ISBN: 84-205-4646-1

José López Quijado
Domine Javascript
RA-MA editorial. 2ª edición 2007
ISBN: 9788478977604

Nicholas C. Zakas, Jeremy McPeak, Joe Fawcett
Professional Ajax
Wiley Publishing Inc. 2ª edición 2007
ISBN: 978-0-470-10949-6

Christopher Murphy, Nicklas Persson
HTML y CSS
Anaya Multimedia. 1ª edición 2009
ISBN: 8441526117

Andy Budd, Cameron Moll, Simon Collison
CSS
Anaya Multimedia. 1ª edición 2007
ISBN: 8441521379

Bases de datos

MySQL 5.0 Reference Manual. Oracle

<http://downloads.mysql.com/docs/mysql-tutorial-excerpt-5.1-en.pdf>

Configuración del servidor

Mohammed J. Kabir

La biblia del Servidor Apache 2

Anaya Multimedia 2002

ISBN: 84-415-1468-2

Página oficial del proyecto Net-SNMP

<http://www.net-snmp.org/>

SMS

MENSATEK

Unidad de negocio de la empresa ASETEC Ingeniería de Sistemas (ASETEC GROUP).

<http://www.mensatek.com/>

Anexo I: Manual de usuario



Autor: Víctor Arrebola Real

Introducción

Aifos, es una herramienta de monitorización de redes, centrada en la monitorización de servidores y estaciones de trabajo que funcionen bajo sistemas operativos GNU/Linux.

El acceso a la aplicación central se realiza a través de un navegador Web, entre los que se recomiendan:

- Mozilla Firefox 3.3 o superior
- Google Chrome 5.0 o superior
- Internet Explorer 6.0 o superior

Para el correcto funcionamiento, las interfaces Web deberán ser servidas desde un servidor Web, preferiblemente Apache 2, en el que además se deberá ejecutar un script encargado de gestionar las comunicaciones entrantes provenientes de los host monitorizados. Para ello se deberá instalar el script cliente en cada uno de los hosts que se quiera monitorizar.

La aplicación central, formada por el conjunto de interfaces web, deberá tener acceso a la base de datos de la aplicación, pudiendo estar instalada o no en el mismo servidor físico.

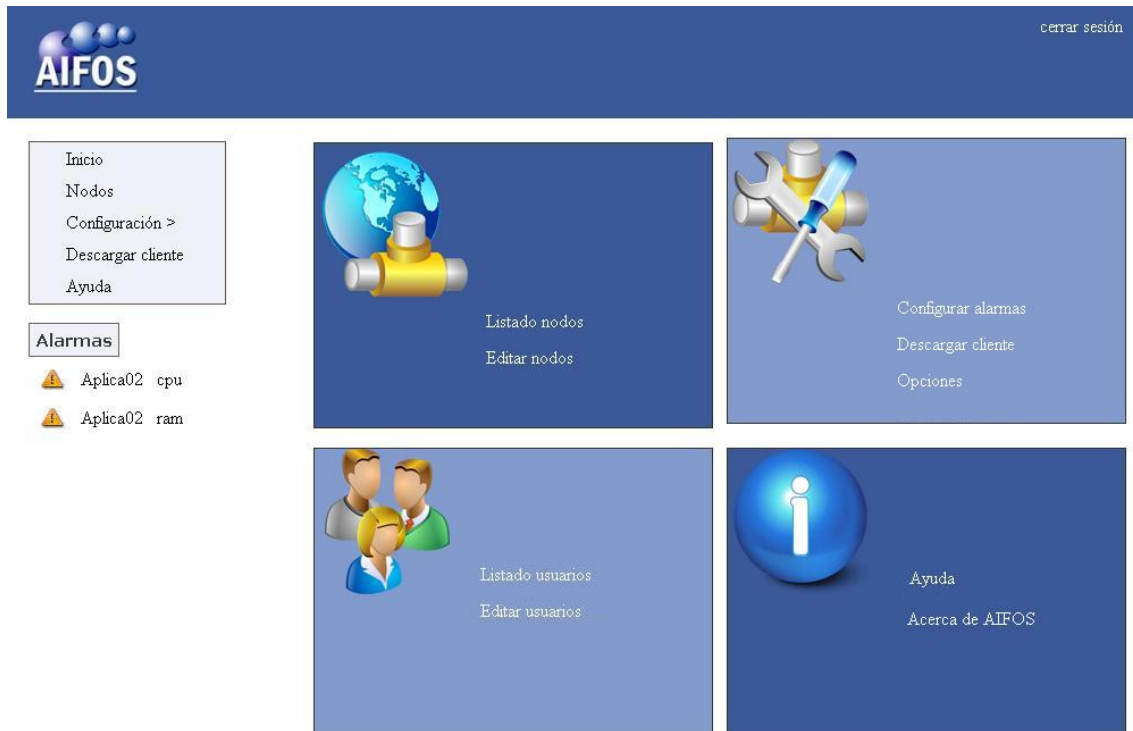
Inicio

Al intentar acceder a la aplicación, se nos presenta la interfaz de login, donde debemos introducir un usuario y su password correspondiente, que sean válidos en el sistema, ya que en caso contrario se denegara el acceso.



Una vez logeados, entraremos en la pagina de inicio, que consta de un menú en la parte superior izquierda, que nos acompañara a través de todas las interfaces, facilitándonos la navegación por la aplicación. Los componentes del menú son muy intuitivos, aunque serán detallados a lo largo del manual.

En la parte central encontramos expuesta de una forma más visual, un resumen de las interfaces que nos ofrece el sistema clasificadas por categorías.



En todo momento se nos ofrece en la parte superior derecha de la ventana, la opción de cerrar la sesión del usuario con el que nos encontramos trabajando, para volver así a la pantalla de login.

Otra de las funcionalidades que encontramos en esta pantalla inicial, y que de igual manera que el menú, nos acompañara por el resto de interfaces, es la ventana de alarmas. Su funcionalidad se detallará en el apartado relativo a las alarmas del sistema, aquí simplemente avanzamos, que se trata de un listado en el que se nos muestran las alarmas sucedidas y registradas por la aplicación central, pero que aun no han sido atendidas por el usuario.

Usuarios

En Aifos, podremos trabajar con dos perfiles distintos de usuario:

- usuario
- administrador

El primero será el utilizado por actores del sistema que podrán consultar toda la información que el mismo nos ofrece, pero no podrán modificar configuraciones, ni añadir, eliminar o editar la información relativa tanto a usuarios como a nodos.

El perfil administrador, tendrá un acceso sin restricciones a la aplicación, pudiendo dar de alta, editar o eliminar tanto nodos como usuarios, configurar nuevas alarmas en los nodos, modificar las configuraciones relativas al sistema de alarmas, y trabajar con partes de la interfaz de monitorización en tiempo real como la consola virtual.

La manera habitual de acceder a la información relativa a usuarios, es a través de: menú -> Configuración -> Configurar usuarios.

Con esta acción conseguiremos mostrar por pantalla un listado de los usuarios dados de alta en el sistema, junto con su información más relevante.



Nombre	Privilegios	Descripcion	Activo	
Victor	1	Administrador	0	Editar
Carlos	2	Usuario	0	Editar

Desde esta interfaz, siempre y cuando nos encontremos trabajando con un perfil de administrador, podremos dar de alta un nuevo usuario mediante el botón situado en la parte superior del listado, o editar alguno existente mediante la opción “editar” situada en la parte derecha del listado.

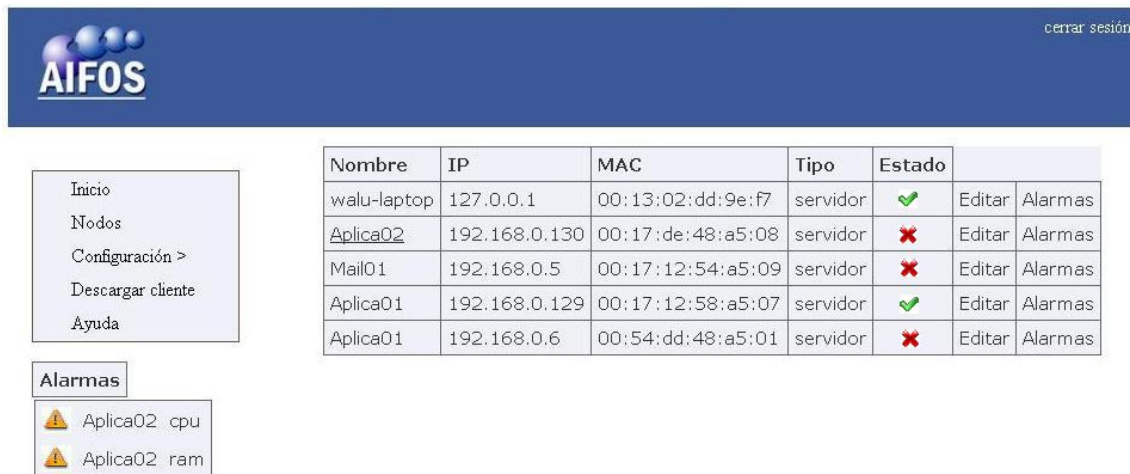
A través del formulario de edición, podremos consultar la totalidad de la información relativa a un usuario y modificarla.

Nodos

Como hemos comentado al inicio del manual, todos los host que deseemos monitorizar, deberán tener instalado y ejecutándose el script *cliente*.


Para poder acceder a las opciones relativas a los nodos, debemos dirigirnos a: menú -> Configuración -> Configurar nodos.

Se nos mostrara por pantalla un listado con todos los nodos dados de alta en el sistema, además de la información más relevante de cada uno de ellos. Desde este mismo listado, en la columna de estado, podemos ver el estado o conexión de red en tiempo real del nodo en cuestión.



Nombre	IP	MAC	Tipo	Estado		
walu-laptop	127.0.0.1	00:13:02:dd:9e:f7	servidor	✓	Editar	Alarmas
<u>Aplica02</u>	192.168.0.130	00:17:de:48:a5:08	servidor	✗	Editar	Alarmas
Mail01	192.168.0.5	00:17:12:54:a5:09	servidor	✗	Editar	Alarmas
Aplica01	192.168.0.129	00:17:12:58:a5:07	servidor	✓	Editar	Alarmas
Aplica01	192.168.0.6	00:54:dd:48:a5:01	servidor	✗	Editar	Alarmas

Para poder acceder a la monitorización en tiempo real de uno de los host, únicamente debemos clicar en su nombre. Echo esto, se nos mostrará la interfaz de monitorización en tiempo real.



Inicio

Nodos

Configuración >

Descargar cliente

Ayuda

Host			
Host:	walu-laptop	IP:	127.0.0.1
		MAC:	00:13:02:dd:9e:f7
Tipo:	servidor	Version AIFOS:	

CPU			
Consumo:	20.3	Arquitectura:	GenuineIntel
		Cores:	2
Modelo:	Genuine Intel(R) CPU T2500 @ 2.00GHz		

Memoria			
RAM total:	1025400 kB	RAM usada:	151328 kB
		RAM libre:	874072 kB
SWAP total:	1052216 kB	SWAP usada:	22440 kB
		SWAP libre:	1029776 kB

Servicios

Servicios	
apache2	on
bluetooth	on

Alarmas

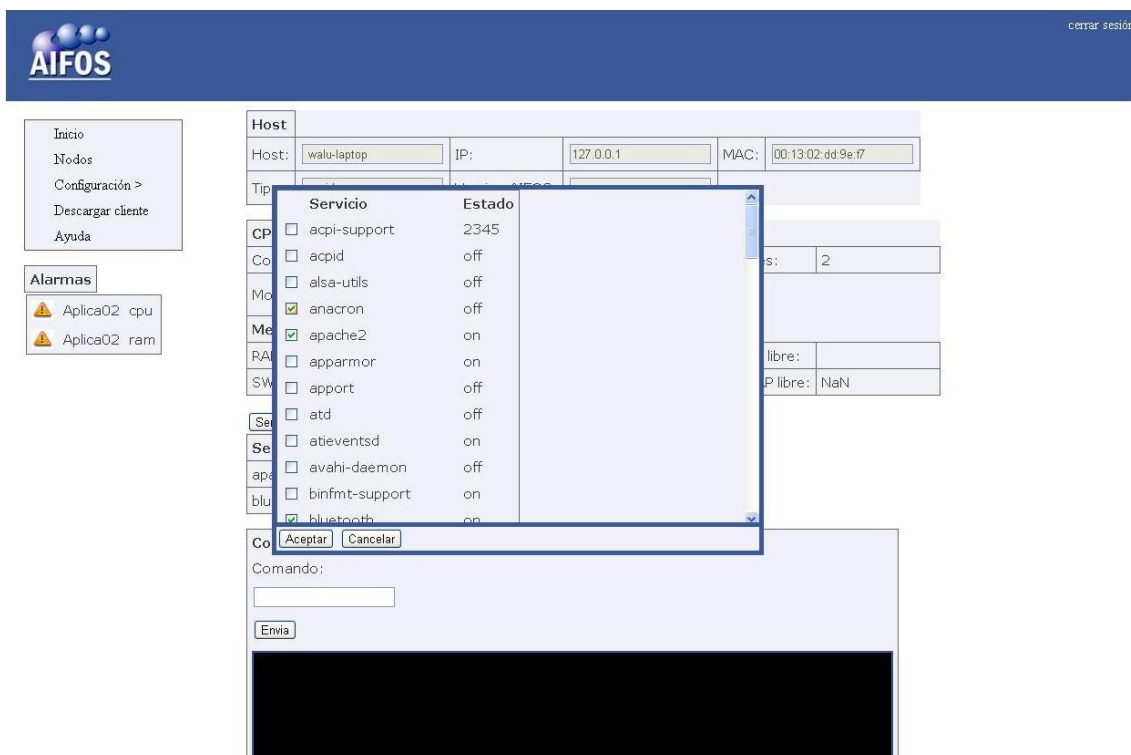
⚠ Aplica02 cpu

⚠ Aplica02 ram

Podemos diferenciar tres tablas diferentes. En la primera, se nos muestra la información relevante del nodo respecto a la red, obtenida de la base de datos del sistema. El resto de tablas, obtienen la información en tiempo real del nodo en cuestión, actualizándola a intervalos cortos de tiempo.

En la segunda tabla, encontramos toda la información relativa al tipo de procesador y su consumo actual, y la memoria tanto RAM como SWAP utilizada y disponible en el nodo.

En la tabla de servicios, se nos ofrece una visión del nombre y el estado de los servicios que previamente hayamos configurado para que aparezcan. Esto podemos hacerlo desde el botón situado en la parte superior de la tabla. Esta acción nos mostrará un listado de la totalidad de los servicios existentes en el nodo monitorizado y su estado actual. Podemos seleccionar o deseleccionar los que deseemos que nos aparezcan en la interfaz principal.



Las funcionalidades mostradas hasta el momento en la monitorización de nodos, son las accesibles tanto por el perfil de usuario, como por el perfil administrador. Sin embargo, a continuación exponemos una nueva funcionalidad que únicamente será mostrada si estamos trabajando con una sesión con un usuario con perfil de administrador, ya que la interacción con esta parte del sistema puede tener repercusiones en el funcionamiento normal del estado de la estación monitorizada.

Se trata de una simulación de línea de comandos, en la que podremos enviar y ejecutar remotamente comandos de shell y obtener los resultados producidos por la ejecución del comando. En la siguiente figura, se nos muestra un ejemplo de la interfaz, una vez ha ejecutado el comando “date”.

Consola Virtual
Comando:

jue jul 8 00:37:57 CEST 2010

Podremos ejecutar cualquier comando disponible en el sistema remoto, pero no podremos escalar directorios. El nivel de permisos con el que se ejecutan los comandos, es el mismo que el del usuario que ha iniciado el script en el nodo monitorizado.

Edición

Volviendo al listado de nodos, para cada uno de los que aparecen, tenemos en la parte derecha de la tabla una opción para editarlos. Desde este formulario, podremos consultar la totalidad de la información relativa a un nodo, y siempre y cuando nos encontremos trabajando con un perfil de administrador, eliminarlo, editarlo, o dar de alta uno nuevo.

[Inicio](#)[Nodos](#)[Configuración >](#)[Descargar cliente](#)[Ayuda](#)

Alarmas

 Aplica02 cpu Aplica02 ram[Editar](#)[Actualiza](#)[Cancelar](#)

ID:	<input type="text" value="4"/>
Nombre:	<input type="text" value="Aplica02"/>
IP:	<input type="text" value="192.168.0.130"/>
MAC:	<input type="text" value="00:17:de:48:a5:08"/>
Tipo:	<input type="text" value="servidor"/>
Descripción:	<input type="text" value="Servidor de aplicaciones"/>
Estado:	<input type="text"/>

Alarmas

Este apartado es conjuntamente con la interfaz dedicada a la monitorización de nodos en tiempo real, uno de los más relevantes dentro del proyecto, pues de encarga de gestionar todo lo relativo al sistema de alarmas que implementa el sistema Aifos.

Comenzaremos por explicar con más detalle la funcionalidad de la tabla de alarmas que nos acompaña en todas las interfaces en la parte izquierda de la pantalla, debajo del menú.



En este listado, las que nos aparecen, son todas las alarmas emitidas por los nodos, que aún no han sido marcadas como atendidas por ningún usuario. Para poder ver más información deberemos clicar en la que nos interese, y nos aparecerá toda la información relativa a la alarma seleccionada, con la opción de poder acceder a la monitorización en tiempo real del nodo para poder comprobar su estado, únicamente clickando en el nombre del host. Una vez llevadas acabo las acciones necesarias, debemos clicar en la opción “atendida”, para indicar que la alarma ha sido atendida, y no se muestre más en esta tabla.

Las alarmas se clasifican en dos tipos, según cual sea su gravedad:

Aviso: es el nivel de alarma considerado menos crítico, y asociado al cual, se procede a publicar la alarma en la aplicación central y enviar un e-mail a la dirección configurada.

Alerta: es el nivel de alarma más crítico. Junto a la publicación de la alarma en la aplicación central, se envía un e-mail a la dirección especificada, y además un SMS al o los teléfonos móviles especificados.

- Configuración de alarmas

Desde esta parte de la aplicación, es donde podemos configurar las alarmas para un nodo concreto.

Los aspectos que podemos configurar son:

- Consumo de CPU
- Consumo de memoria RAM
- Espacio libre en disco
- Servicios

Para los consumos de CPU, RAM y disco, la configuración es idéntica, ya que podemos definir un umbral de aviso y otro de alerta. El umbral de alerta siempre deberá ser más crítico que el de aviso. Por crítico se entiende, que por ejemplo en el caso del consumo de CPU, el umbral de alerta debe definir un consumo más elevado que el de aviso. En el caso de la memoria RAM y de espacio en disco, en los que se define la cantidad de memoria libre, el umbral de alerta, siempre deberá ser un valor inferior al de aviso.

En el caso de los servicios, se especificara un servicio existente en el host y se especificara cual es el estado para el que queremos que se nos avise si se produce. Por ejemplo, podemos seleccionar el servicio 'apache2' y seleccionar 'off'. Dicha configuración nos avisara si el estado del servicio 'apache2' pasa a ser 'off', lo que nos indicaría que se ha detenido. Obviamente también se puede configurar a la inversa, por si nos interesara saber si se ha activado un servicio que debería estar inactivo.

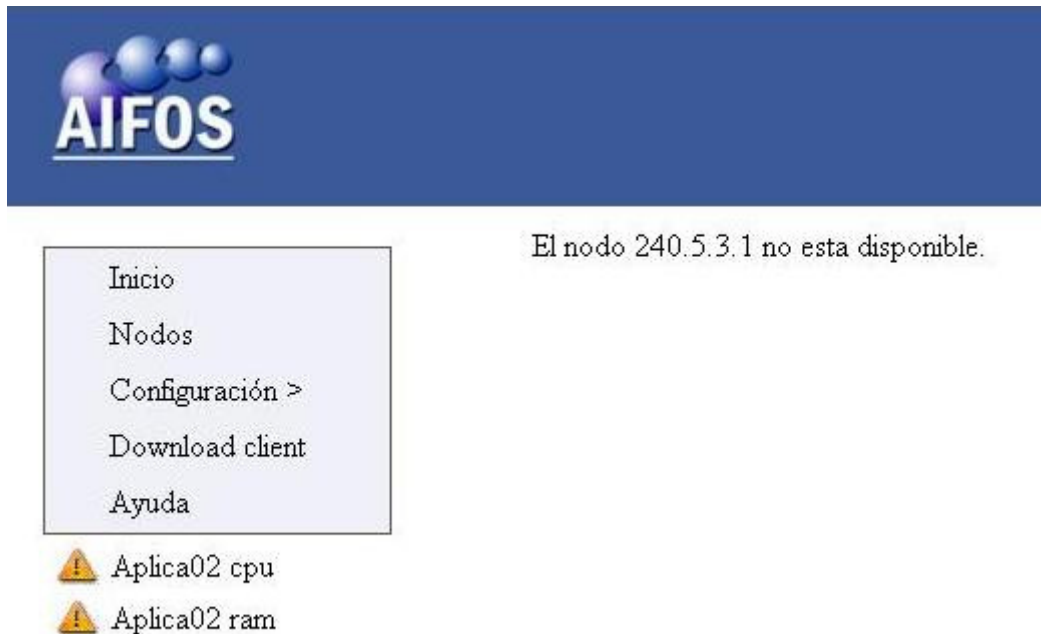
Todas las alarmas generadas por servicios, serán tratadas siempre como alertas, otorgándoles de esta manera siempre, el carácter más crítico.

A continuación de las opciones de configuración de las alarmas, nos aparece un listado de las alarmas configuradas en el mismo. Desde este listado podremos eliminar las que no deseemos.

No podremos modificar alarmas existentes, pero tampoco podremos generar dos alarmas distintas para un mismo tipo, como por ejemplo para el consumo de CPU. Por lo que si disponemos de una alarma configurada para el tipo CPU y deseamos configurarla, únicamente deberemos configurar las opciones de la misma, y esta será modificada sin la necesidad de que haya sido eliminada previamente.

Debido al diseño del sistema, dos condiciones indispensables para poder configurar alarmas en un host concreto, son que el equipo este dado de alta en la aplicación, y que en el momento de configurar la alarma, disponga de conexión de red. Estas dos condiciones son valoradas en el momento de acceder a la interfaz de configuración, de manera que si el equipo se encuentra dado de alta y con conexión a red, podremos configurarlo. En caso contrario, se nos avisara del error correspondiente, impidiéndonos el acceso a la interfaz. Esta comprobación es necesaria debido a que es el propio cliente

el que almacena sus propias alarmas y se monitoriza a si mismo, interviniendo el servidor únicamente cuando se produce una alarma y esta es enviada desde el cliente, o bien cuando desea acceder al mismo para visualizar su configuración de alarmas.



- Listado de alarmas

En esta interfaz se nos muestra un listado de todas las alarmas generadas por los distintos clientes, que aún no han sido atendidas.

Se muestra un breve resumen, en el que podemos observar la gravedad de la alarma, el host en el que se ha producido, el tipo de parámetro al que afecta, la fecha y hora en la que se produjo, el umbral del parámetro que especificó el usuario para esta alarma, y el consumo o estado real que se ha dado.

Como hemos comentado, través del link ubicado en el nombre del host, podemos acceder a la interfaz de monitorización del nodo, para comprobar el estado del mismo y decidir las acciones correspondientes a tomar, valorando el estado actual. Una vez valorado y llevadas a cabo las actuaciones correspondientes, podemos marcar como atendida la alarma, para que esta desaparezca tanto del listado de alarmas, como de la pantalla presente en todas las interfaces de la aplicación.