



MEMORIA

**HERRAMIENTA DE MONITORIZACIÓN DE SERVIDORES Y EQUIPOS DE
RED i2basqueNMS**

MEMORIA TRABAJO FIN DE GRADO

DATOS DE LA ALUMNA O DEL ALUMNO

NOMBRE: AARON

APELLIDOS: OJEMBARRENA LOZANO

FDO.:

FECHA:

DATOS DEL DIRECTOR O DE LA DIRECTORA

NOMBRE: BEGO

APELLIDOS: FERRERO MARTÍN

DEPARTAMENTO: LENGUAJES Y SISTEMAS
INFORMÁTICOS

FDO.:

FECHA:

RESUMEN

La idea central de este TFG fue planteada por i2Basque, un proyecto financiado por el Gobierno Vasco que, actualmente, forma parte de la fundación IKERBASQUE. El objetivo de i2Basque es vertebrar la comunidad de I+D+I en el País Vasco, proporcionando infraestructuras de telecomunicaciones y servicios TIC a los agentes de la Red Vasca de Ciencia y Tecnología.

Con el fin de mantener una vigilancia constante sobre el estado en tiempo real de todos los equipos y servicios asociados a la red I+D+I del País Vasco, se propuso la construcción y configuración de una herramienta de monitorización de servidores y equipos de red denominada **i2BasqueNMS** (i2Basque Network Monitoring System).

Existen numerosas y variadas herramientas Open Source para este fin, pero ninguna de ellas da respuesta, por si sola, a todas las necesidades de i2Basque.

En el presente documento se aborda la construcción y configuración de dicha herramienta de monitorización

INDICE DE CONTENIDOS

RESUMEN	I
INDICE DE CONTENIDOS.....	III
INDICE DE ILUSTRACIONES.....	V
INDICE DE TABLAS.....	IX
1 INTRODUCCIÓN	1
2 DOCUMENTO DE OBJETIVOS DEL PROYECTO.....	3
2.1 DESCRIPCIÓN DEL TRABAJO	3
2.2 OBJETIVOS.....	3
2.3 ARQUITECTURA.....	4
2.4 HERRAMIENTAS	4
2.5 PLANIFICACIÓN DEL TRABAJO	5
2.6 PLANIFICACIÓN TEMPORAL.....	6
2.7 RIESGOS.....	8
2.8 EVALUACIÓN ECONÓMICA.....	10
3 ANTECEDENTES	13
3.1 OPENNMS.....	14
3.2 HYPERIC.....	15
3.3 PANDORAFMS	17
3.4 NAGIOS	19
3.5 SHINKEN	21
3.6 CONCLUSIONES.....	22
4 CAPTURA DE REQUISITOS.....	25
4.1 FUNCIONALIDADES SOFTWARE	25
4.2 COMPATIBILIDADES HARDWARE	26
5 ANALISIS Y DISEÑO.....	31
5.1 PLUGINS AÑADIDOS A SHINKEN	31
5.2 INTERFAZ WEB PARA SHINKEN.....	31
5.3 GRÁFICOS DE USO DE LOS EQUIPOS Y SERVICIOS MONITORIZADOS.....	33
5.4 VISUALIZACIÓN DE MAPAS	34
5.5 TERMINAL WEB PARA CONEXIÓN DIRECTA CON EL SERVIDOR	34
5.6 AUTO-DETECCIÓN DE LOS EQUIPOS Y LOS SERVICIOS DISPONIBLES EN LA RED	35
5.7 ENVÍO DE AVISOS E INFORMES DEL ESTADO DE LA RED	35
6 DESARROLLO.....	37
6.1 PRIMER PROTOTIPO	37
6.2 SEGUNDO PROTOTIPO.....	50
6.3 TERCER PROTOTIPO.....	58
6.4 DESARROLLO EXTRA.....	63
7 VERIFICACIÓN DE I2BASQUENMS	67
7.1 INSTALACIÓN DE LA VERSIÓN ORIGINAL.....	67
7.2 INSTALACIÓN DE LA VERSIÓN PARA RASPBERRY PI: SCRIPT DE INSTALACIÓN.....	69
7.3 INSTALACIÓN DE LA VERSIÓN PARA RASPBERRY PI: IMAGEN DE DISCO.....	70
8 COMPARATIVA ENTRE LA PLANIFICACIÓN INICIAL Y LA REAL	73
9 CONCLUSIONES Y TRABAJO FUTURO	75

10	BIBLIOGRAFIA.....	77
----	-------------------	----

INDICE DE ILUSTRACIONES

ILUSTRACIÓN 2-01: ARQUITECTURA CLIENTE - SERVIDOR I2BASQUENMS	4
ILUSTRACIÓN 2-02: TIEMPO ESTIMADO DE CADA UNA DE LAS TAREAS	7
ILUSTRACIÓN 2-03: DIAGRAMA GANTT CON LA ESTIMACIÓN DE TIEMPO POR CADA TAREA.....	7
ILUSTRACIÓN 3-01: INTERFAZ WEB PRINCIPAL DE OPENNMS.	14
ILUSTRACIÓN 3-02: GESTOR DE MAPAS DE OPENNMS.....	15
ILUSTRACIÓN 3-03: INTERFAZ WEB PRINCIPAL DE HYPERIC.	16
ILUSTRACIÓN 3-04: INTERFAZ WEB CON LAS GRÁFICAS PROPORCIONADAS POR HYPERIC.	17
ILUSTRACIÓN 3-05: INTERFAZ WEB PRINCIPAL DE PANDORAFMS.....	18
ILUSTRACIÓN 3-06: INTERFAZ WEB CON LAS UNIONES ENTRE EQUIPOS DE UNA RED EN PANDORAFMS.	18
ILUSTRACIÓN 3-07: INTERFAZ WEB PRINCIPAL DE NAGIOS.....	19
ILUSTRACIÓN 3-08: INTERFAZ WEB WEBUI DE SHINKEN, EN LA PESTAÑA DONDE APARECEN TODOS LOS EQUIPOS.	21
ILUSTRACIÓN 5-01: PANTALLA PRINCIPAL DE LA INTERFAZ WEB MULTISITE.....	32
ILUSTRACIÓN 5-02: PANTALLA DE INFORMACIÓN COMPLETA DE TODOS LOS EQUIPOS DE LA INTERFAZ WEB WEBUI.	32
ILUSTRACIÓN 5-03:INTERFAZ WEB DE GRAPHITE.....	33
ILUSTRACIÓN 5-04: INTERFAZ WEB DE PNP4NAGIOS.....	33
ILUSTRACIÓN 5-05: INTERFAZ WEB CON EL MAPA DE EUROPA DE NAGVIS.	34
ILUSTRACIÓN 5-06: SOFTWARE NMAP EJECUTÁNDOSE EN UNA TERMINAL.	35
ILUSTRACIÓN 5-07: ESQUEMA COMPLETO DE LA ESTRUCTURA DE LA SOLUCIÓN DISEÑADA	36
ILUSTRACIÓN 6-01: BOTONES QUE OCASIONABAN ERROR EN EL DIRECCIONAMIENTO A PNP4NAGIOS	38
ILUSTRACIÓN 6-02: BOTONES CON DIRECCIONAMIENTO ERRÓNEO A LA DOCUMENTACIÓN	38
ILUSTRACIÓN 6-03: INICIALIZACIÓN DE SHINKEN.....	43
ILUSTRACIÓN 6-04: INICIALIZACIÓN DE PNP4NAGIOS.....	44
ILUSTRACIÓN 6-05: VARIOS EQUIPOS AÑADIDOS A SHINKEN PARA SER MONITORIZADOS	44
ILUSTRACIÓN 6-06: REINICIO DE SHINKEN TRAS A ADICIÓN DE EQUIPOS PARA SU MONITORIZACIÓN	45
ILUSTRACIÓN 6-07: EQUIPOS DE LAS MARCAS DELL, CISCO Y JUNIPER SIENDO MONITORIZADOS CORRECTAMENTE POR SHINKEN.....	45
ILUSTRACIÓN 6-08: SECCIÓN DASHBOARD DE SHINKEN	46
ILUSTRACIÓN 6-09: REALIZACIÓN DE GRÁFICAS EN PNP4NAGIOS CON LOS DATOS RECABADOS DE LA RED POR SHINKEN	46
ILUSTRACIÓN 6-10: CUADRO DE BÚSQUEDA DE EQUIPOS EN PNP4NAGIOS	47
ILUSTRACIÓN 6-11: FILTROS PARA LA BÚSQUEDA DE EQUIPOS EN SHINKEN	47

ILUSTRACIÓN 6-12: ALARMA POR EMAIL INDICANDO ERRORES DE CONEXIÓN EN LAS INTERFACES DE RED DEL ROUTER JUNIPER-M10I	48
ILUSTRACIÓN 6-13: BUSQUEDA INDEXADA DE EQUIPOS EN SHINKEN	48
ILUSTRACIÓN 6-14: DOCUMENTO PDF CREADO A PARTIR DE LAS GRÁFICAS DE PNP4NAGIOS	49
ILUSTRACIÓN 6-15: INTERFAZ WEB DE PNP4NAGIOS PULSAR EL BOTÓN PARA QUE MUESTRE LOS DATOS DE LAS ÚLTIMAS 4 HORAS.....	49
ILUSTRACIÓN 6-16: GRÁFICAS CREADAS POR PNP4NAGIOS EN LA INTERFAZ DE SHINKEN.....	50
ILUSTRACIÓN 6-17: LOGIN DE NAGVIS	53
ILUSTRACIÓN 6-18: FOTOGRAFÍA AÉREA DE I2BASQUE CON LOS DATOS DE UNO DE LOS SERVIDORES MONITORIZADOS.....	54
ILUSTRACIÓN 6-19: NAGVIS MOSTRANDO, SOBRE UNA FOTO AÉREA DE LA FACULTAD DE VITORIA, LOS DATOS DE UNO DE LOS EQUIPOS MONITORIZADOS	54
ILUSTRACIÓN 6-20: SECCIÓN DE DATOS RESPECTO A LA MÁQUINA SELECCIONADA EN NAGVIS	55
ILUSTRACIÓN 6-21: INTERFAZ WEB CON LAS OPCIONES PARA ADMINISTRAR LAS IMÁGENES DE LOS FONDOS DE LOS MAPAS	55
ILUSTRACIÓN 6-22: INTERFAZ WEB CON LAS OPCIONES PARA ADMINISTRAR LOS MAPAS, DONDE SE VISUALIZA LA IMAGEN DE FONDO SUBIDA POR EL ADMINISTRADOR.....	56
ILUSTRACIÓN 6-23: MAPA CREADO EN NAGVIS A PARTIR DE UNO DE LOS FONDOS AÑADIDOS POR EL ADMINISTRADOR.....	56
ILUSTRACIÓN 6-24: MAPA DE NAGVIS DEL PAÍS VASCO CON UN ENLACE A OTRO MAPA DE BILBAO... 57	
ILUSTRACIÓN 6-25: MAPA DE NAGVIS CON UNA VISTA POR SATÉLITE DE LA ESCUELA DE INGENIEROS DE BILBAO CON UNO DE SUS EQUIPOS SIENDO MONITORIZADO	57
ILUSTRACIÓN 6-26: MAPA DE NAGVIS DE LA ESCUELA DE INGENIEROS DE BILBAO CON UNA MARCA AZUL INDICANDO QUE EL EQUIPO YA NO ESTÁ SIENDO MONITORIZADO POR I2BASQUENMS.....	58
ILUSTRACIÓN 6-27: INTERFAZ WEB DE SHINKEN CON EL MENÚ DESPLEGABLE AÑADIDO.....	60
ILUSTRACIÓN 6-28: INTERFAZ WEB DE NAGVIS TRAS LA ADICIÓN DE LOS LINKS AL RESTO DE PROGRAMAS DE I2BASQUENMS.....	60
ILUSTRACIÓN 6-29: INTERFAZ WEB DE PNP4NAGIOS TRAS LA ADICIÓN DE LOS LINKS AL RESTO DE PROGRAMAS DE I2BASQUENMS.....	60
ILUSTRACIÓN 6-30: INTERFAZ WEB DE SHINKEN CON LA OPCIÓN TERMINAL SELECCIONADA	61
ILUSTRACIÓN 6-31: TERMINAL WEB DANDO ACCESO DIRECTO AL ADMINISTRADOR AL SERVIDOR DONDE SE ENCUENTRA INSTALADO EL SISTEMA DE MONITORIZACIÓN.....	61
ILUSTRACIÓN 6-32: CONEXIÓN REMOTA CON EL SERVIDOR A TRAVÉS DE LA TERMINAL AÑADIDA A LA SOLUCIÓN. INSERCIÓN DEL COMANDO PARA EJECUTAR EL ESCANEADO DE LA RED	62
ILUSTRACIÓN 6-33: ESCANEADO DE LA RED MONITORIZADA MEDIANTE EL USO DE NMAP A TRAVÉS DE LA CONEXIÓN REMOTA.....	62
ILUSTRACIÓN 6-34 ESQUEMA DEL DISEÑO PARA RASPBERRY PI.....	64
ILUSTRACIÓN 7-01: INICIO DEL SCRIPT DE INSTALACIÓN EN SU VERSIÓN PARA DEBIAN 7 WHEEZY X86_64BIT.....	67
ILUSTRACIÓN 7-02: A MITAD DE LA INSTALACIÓN DE I2BASQUENMS EN DEBIAN 7 WHEEZY X86_64BIT	68

ILUSTRACIÓN 7-03: FINAL DE LA INSTALACIÓN DE I2BASQUENMS. UNA VEZ SE INSTALA, SE CONFIGURA Y SE SOLUCIONAN LOS PROBLEMAS, SE REINICIA TANTO APACHE COMO SHINKEN	68
ILUSTRACIÓN 7-04: CONFIGURACIÓN INICIAL DE LA INSTALACIÓN EN UNA RASPBERRY PI	69
ILUSTRACIÓN 7-05: ACTUALIZACIÓN DE LA VERSIÓN MINIMIZADA DE DEBIAN PARA RASPBERRY PI ..	70
ILUSTRACIÓN 7-06: IMAGEN DE DISCO DE DEBIAN CON I2BASQUENMS PREINSTALADO DESCARGADA	70
ILUSTRACIÓN 7-07: GRABACIÓN DE LA IMAGEN DE DEBIAN JUNTO CON I2BASQUENMS CREADA PARA ESTE TFG, EN UNA TARJETA SD	71
ILUSTRACIÓN 8-01: COMPARACIÓN GRÁFICA ENTRE LA ESTIMACIÓN INICIAL Y EL TIEMPO REAL.....	74

INDICE DE TABLAS

TABLA 2-1: ESTIMACIÓN DE TIEMPO INICIAL DE LA DURACIÓN DEL PROYECTO.....	5
TABLA 2-2: SALARIO DEL INFORMÁTICO CON LOS DESCUENTOS DEL IRPF Y SEGURIDAD SOCIAL APLICADOS	10
TABLA 2-3: COSTE MONETARIO DE LAS APLICACIONES USADAS DURANTE EL PROYECTO.....	11
TABLA 2-4: COSTE DE LOS EQUIPOS UTILIZADOS EN EL PROYECTO	11
TABLA 2-5: AMORTIZACIÓN DE LOS EQUIPOS UTILIZADOS DURANTE LA REALIZACIÓN DEL PROYECTO	12
TABLA 2-6: OTROS GASTOS DERIVADOS DE LA REALIZACIÓN DEL PROYECTO	12
TABLA 2-7: COSTE MONETARIO TOTAL DEL PROYECTO.....	12
TABLA 3-1: COMPARACIÓN GENERAL ENTRE LAS DISTINTAS SOLUCIONES ESTUDIADAS	24
TABLA 4-1: SERVIDOR EN EL QUE SE INSTALARÁ LA SOLUCIÓN.....	26
TABLA 4-2: ROUTERS DE LA MARCA JUNIPER QUE SE DISPONDRÁ EN LA RED	26
TABLA 4-3: CARACTERÍSTICAS DE LOS ROUTERS DE LA TABLA 4-2.....	26
TABLA 4-4: EQUIPOS DE LA MARCA INFINERA QUE SE DISPONDRÁN EN LA RED.....	27
TABLA 4-5: CARACTERÍSTICA PRINCIPAL DE LOS EQUIPOS ÓPTICOS INFINERA DEL CUADRO 4-4.....	27
TABLA 4-6: SWITCHES DE LA MARCA JUNIPER QUE SE DISPONDRÁN EN LA RED.....	27
TABLA 4-7: CARACTERÍSTICAS PRINCIPALES DE LOS SWITCHES DE LA TABLA 4-6.	27
TABLA 4-8: SERVIDORES DE LOS QUE SE DISPONDRÁN EN LA RED FINAL DE I2BASQUE	28
TABLA 4-9: CARACTERÍSTICAS DE LOS SERVIDORES DE LA TABLA 4-8.....	29
TABLA 6-1: CARACTERÍSTICAS PRINCIPALES DE LOS DISTINTOS MODELOS DE RASPBERRY PI	63
TABLA 8-1: COMPARACIÓN ENTRE LA ESTIMACIÓN DE TIEMPO INICIAL Y EL TIEMPO REAL.....	74

1 INTRODUCCIÓN

El Trabajo Fin de Grado (TFG) aborda la construcción y configuración de una herramienta de monitorización de servidores y equipos de red denominada **i2BasqueNMS** (i2Basque Network Monitoring System). En el presente documento se utilizará el término *equipo* para referirse a dispositivos hardware, tales como routers, switches, servidores, ...; las piezas internas de los equipos se identificarán como *componentes*, y se denominarán *servicios* a los diferentes servicios proporcionados por los equipos, tales como correo, firewall, compartición de archivos en red, etc.

Un buen proceso de monitorización de red mantiene a su administrador informado en todo momento sobre la situación de cada uno de los equipos y servicios de la misma, permitiéndole anticiparse a las incidencias que pudieran surgir. Así, el administrador podrá resolver situaciones problemáticas antes de que ocurran, evitando o minimizando su impacto. Detectar a tiempo fallos en un servidor, un router o un switch puede evitar situaciones graves, tales como la paralización de la producción en una cadena de montaje o el colapso de la unidad de urgencias de un hospital.

La idea central de este TFG fue planteada por i2Basque, un proyecto financiado por el Gobierno Vasco que, actualmente, forma parte de la fundación IKERBASQUE. El objetivo de i2Basque es vertebrar la comunidad de I+D+I en el País Vasco, proporcionando infraestructuras de telecomunicaciones y servicios TIC a los agentes de la Red Vasca de Ciencia y Tecnología, entre los que se encuentran:

- Universidades públicas y privadas: UPV/EHU, Deusto, TECNUN, Mondragón, UNED.
- Centros científicos y tecnológicos.
- Hospitales e investigación biomédica.
- Entidades gestoras de I+D.

Aunque se plantea como objetivo fundamental de este trabajo que la herramienta de monitorización final sea válida para cualquier red, durante su construcción y validación, las pruebas se llevarán a cabo utilizando la infraestructura de i2basque. Para ello, ha facilitado una maqueta con los siguientes equipos:

- Un switch Juniper EX4200.
- Un router Juniper M10i.
- Un router Cisco2960G.
- Un servidor físico DELL PowerEdge 1950 con el software de virtualización ProxMox VE3 sobre el que hay 3 servidores virtuales con distintos servicios habilitados.

A continuación se detalla el documento de objetivos del proyecto, para seguir con un estudio de antecedentes, una captura de requisitos para determinar las necesidades de i2Basque, el análisis y diseño de la solución de acuerdo con los requisitos establecidos, el desarrollo de la solución y su posterior verificación.

2 DOCUMENTO DE OBJETIVOS DEL PROYECTO

2.1 Descripción del trabajo

La herramienta de monitorización planteada por i2Basque para este TGF, consistirá en la unión de un conjunto de aplicaciones OpenSource con distintas funcionalidades. Que se adaptarán para que trabajen en conjunto con el objetivo de ofrecer a cualquier administrador de red un sistema de monitorización de redes completo.

2.2 Objetivos

Atendiendo a las necesidades planteadas por i2Basque, los objetivos de la herramienta que se construirá en este TFG son:

- Conocer en cualquier momento la cantidad de RAM, HDD, CPU y ancho de banda de las conexiones de red utilizadas en los diferentes equipos.
- Configurar los avisos sobre fallos o niveles críticos en los equipos, tales como alarmas cuando se detecta que la carga de un procesador es excesiva, que alguno de los servicios ha caído, etc. Estos avisos se enviarán al administrador a través del correo electrónico.
- Permitir una configuración sencilla, con una interfaz web que muestre, sin ningún esfuerzo de interpretación adicional ni engorrosos procesos de navegación, los datos más relevantes.
- Ofrecer la posibilidad de crear mapas en los que se muestre la localización de los equipos y servicios de la red.
- Generar dinámicamente informes sobre su estado.

Actualmente, existen herramientas que ofrecen algunos de estos servicios, pero ninguna de ellas da respuesta por si sola a todas las necesidades planteadas. En consecuencia, la primera tarea de este trabajo será la revisión de dichas herramientas y la elaboración de un estudio comparativo entre ellas para determinar qué requisitos de los establecidos por i2Basque cubre cada una de ellas.

En el siguientes apartado se realiza un estudio de algunas de las herramientas de monitorización OpenSource más conocidas, que son: OpenNMS, Hyperic, PandoraFMS, Nagios y Shinken.

2.3 Arquitectura

La arquitectura de i2BasqueNMS será Cliente-Servidor (véase Ilustración 2-01).

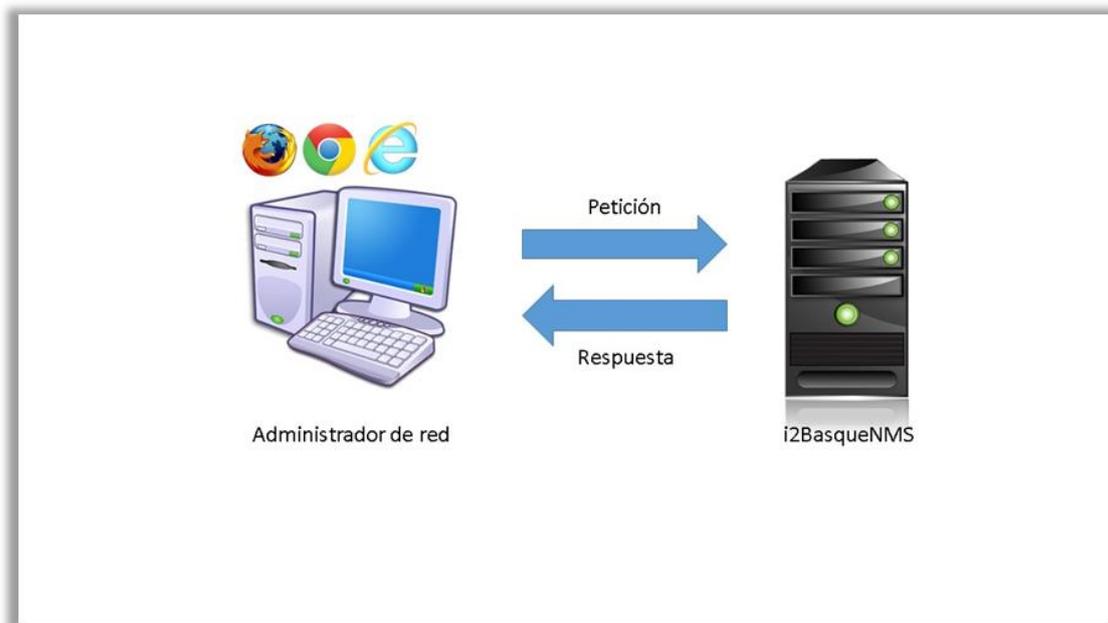


Ilustración 2-01: Arquitectura Cliente - Servidor i2BasqueNMS

El administrador de red deberá interactuar con i2BasqueNMS a través de sus interfaces web, tanto para informarse del estado de la red como para la configuración de la herramienta de monitorización.

2.4 Herramientas

Las herramientas que se utilizarán para la realización de este TFG serán:

- Microsoft Office Word 2013: Aplicación perteneciente a la empresa Microsoft, con la que se creará la documentación del proyecto.
- Putty: Aplicación gratuita que permite la conexión remota a equipos mediante los protocolos Telnet y SSH. Dicha aplicación se utilizará en la realización de conexiones remotas a los equipos prestados por i2Basque. Especialmente a los servidores virtuales, ya que en ellos se realizará casi todo el desarrollo y las pruebas en su totalidad.
- Firefox: Navegador web perteneciente a la empresa Mozilla, con la que se probarán las distintas funcionalidades de i2BasqueNMS según se vayan añadiendo.
- Chrome: Navegador web perteneciente a la empresa Google, con la que se probarán las distintas funcionalidades de i2BasqueNMS según se vayan añadiendo.
- Microsoft Paint: Aplicación de dibujo y recorte fotográfico perteneciente a Microsoft, con la que se recortará y se adaptará el tamaño de las imágenes de la documentación.

- GanttProject: Aplicación libre que permite la creación de esquemas Gantt. Se usará para la creación de los esquemas Gantt pertenecientes a la planificación temporal.
- ProxMox VE 3: Aplicación Open Source de virtualización de equipos. Sobre esta aplicación se encuentran las máquinas virtuales sobre las que se hará el desarrollo y las pruebas de este TFG.
- Nano: Aplicación Open Source de edición de textos para Linux. Se utilizará para la creación de los scripts y la realización de las modificaciones y adaptaciones necesarias a lo largo del desarrollo del proyecto.

2.5 Planificación del trabajo

En este apartado se reflejará el tiempo estimado, en semanas, para llevar a cabo el trabajo requerido para completar todos los objetivos fijados en este TFG (véase *Tabla 2-1*).

Debido al uso de lenguajes de programación no impartidos durante el Graduado o por la necesidad de poseer un nivel mayor de alguno ya conocido, antes de la realización del proyecto, deberá existir un tiempo de aprendizaje.

Actividad	Estimación de tiempo (semanas)
Estudio de: bash scripting, php, html, perl, funcionamiento del protocolo snmp, etc	3 semanas
Estudio de antecedentes	1 semana
Captura de requisitos	1 semana
Diseño de i2BasqueNMS	2 semanas
Desarrollo del primer prototipo	3 semanas
Pruebas del primer prototipo	1 semana
Desarrollo del segundo prototipo	3 semanas
Pruebas del segundo prototipo	1 semana
Desarrollo del tercer prototipo	3 semanas
Pruebas del tercer prototipo	1 semana
Verificación de la correcta instalación y posterior funcionamiento de i2BasqueNMS	2 semanas
Tiempo total estimado	21 semanas

Tabla 2-1: Estimación de tiempo inicial de la duración del proyecto

2.6 Planificación temporal

Debido a que el alumno dispone de 4 asignaturas por evaluar, no se podrá dedicar por completo a la realización del TFG. Por lo tanto, se ha estimado que dispondrá de aproximadamente 3 horas y media al día entre semana dependiendo de la carga de trabajo por parte de las asignaturas pendientes.

En las siguientes páginas se muestra el diagrama Gantt con la estimación de tiempo para cada una de las tareas (*véase Ilustraciones 2-02 y 2-03*).

Nombre	Fecha de inicio	Fecha de fin
• Estudio de: bash scripting, php, html, perl, funcionamiento del protocolo snmp, etc	3/02/14	21/02/14
• Estudio de antecedentes	24/02/14	28/02/14
• Captura de requisitos	3/03/14	7/03/14
• Diseño de i2BasqueNMS	10/03/14	21/03/14
• Desarrollo del primer prototipo	24/03/14	11/04/14
• Pruebas del primer prototipo	14/04/14	18/04/14
• Desarrollo del segundo prototipo	21/04/14	9/05/14
• Pruebas del segundo prototipo	12/05/14	16/05/14
• Desarrollo del tercer prototipo	19/05/14	6/06/14
• Pruebas del tercer prototipo	9/06/14	13/06/14
• Verificación de la correcta instalación y posterior funcionamiento de i2BasqueNMS	16/06/14	27/06/14

Ilustración 2-02: Tiempo estimado de cada una de las tareas

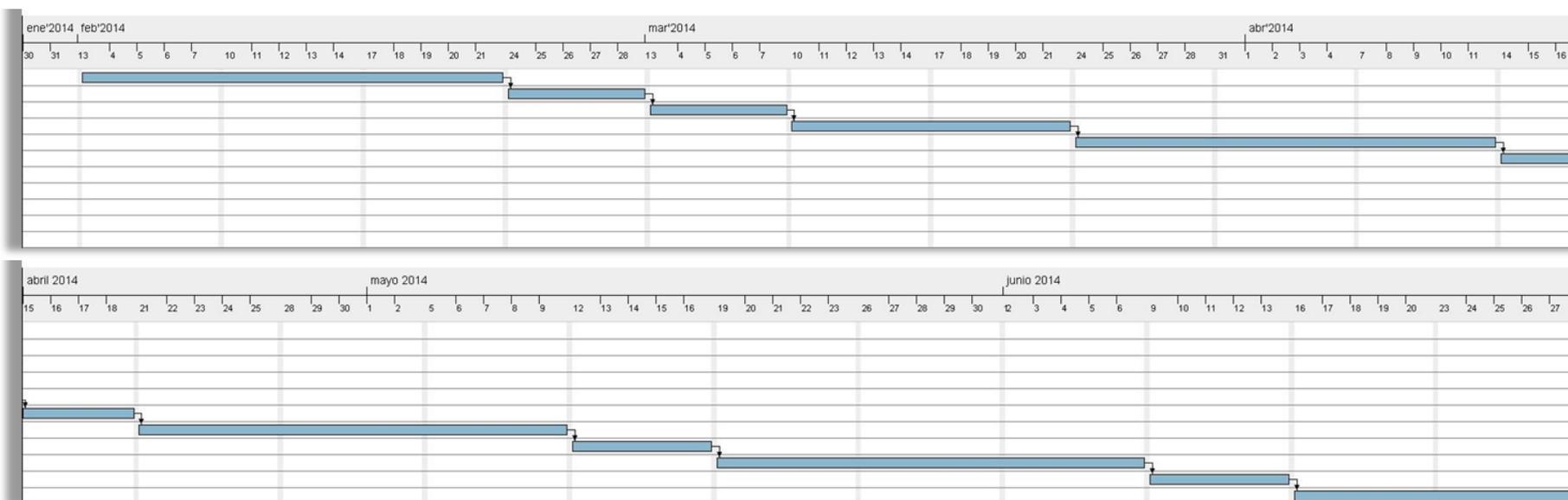


Ilustración 2-03: Diagrama Gantt con la estimación de tiempo por cada tarea

2.7 Riesgos

Existen riesgos que podrían afectar a la planificación temporal del proyecto de manera considerable.

Los más relevantes son:

a) Problemas de seguridad en los servidores virtuales cedidos por i2Basque

Descripción: Podrían producirse fallos de seguridad debido a la instalación de diversos servicios ligados al desarrollo de la aplicación.

Efecto: De producirse este riesgo, si el fallo de seguridad no se solucionara en un corto período de tiempo, podría causar el apagado de los servidores virtuales por el administrador de red de i2Basque, ya que esta se encuentra ligada a RedIRIS (red española para Interconexión de los Recursos Informáticos de las universidades y centros de investigación).

Modo de evitar/mitigar: Revisar cada uno de los servicios que se instalan para asegurarse que ninguno se quede sin ningún tipo de seguridad.

Solución: Filtrar, mediante el firewall de Linux, todos los puertos usados por cada uno de los servicios instalados.

Probabilidad: Alta

Impacto: Alto

b) Problemas de software

Descripción: Podrían producirse errores de compatibilidad, infecciones de virus informáticos, etc.

Efecto: De producirse este riesgo, podría causar pérdida de parte del trabajo que habría que rehacer.

Modo de evitar/mitigar: Mantener el antivirus actualizado, así como crear copias de seguridad de todos los documentos y ficheros creados.

Solución: Poner en cuarentena los ficheros infectados y recuperar la mayor parte posible del trabajo mediante las copias de seguridad.

Probabilidad: Baja

Impacto: Media

c) Problemas de hardware

Descripción: Podrían producirse fallos a nivel físico en los equipos utilizados para el desarrollo de i2BasqueNMS.

Efecto: De producirse este riesgo en el equipo propiedad del alumno, podría causar pérdida de parte del trabajo que habría que rehacer. De producirse en el hardware

prestado por i2Basque, podría repercutir en la calidad de las pruebas realizadas durante el desarrollo.

Modo de evitar/mitigar: Mantener un seguimiento del estado de todos los equipos utilizados.

Solución: Reparar o sustituir el hardware afectado y recuperar la información perdida a través de copias de seguridad.

Probabilidad: Baja

Impacto: Media

d) Dificultades en el desarrollo

Descripción: Podrían producirse dificultades durante el desarrollo, por la necesidad de un mayor conocimiento sobre alguna área concreta por parte del alumno, o por la falta de información por parte de los desarrolladores sobre el software Open Source que haya que modificar.

Efecto: De producirse este riesgo causaría un incremento de tiempo en la realización de alguna de las tareas.

Modo de evitar/mitigar: Documentarse lo mejor posible tanto a nivel de conocimientos por parte del alumno, como en todo lo referente al diseño de las aplicaciones Open Source que vayan a modificarse.

Solución: Empleo de una mayor cantidad de tiempo para adquirir mayores conocimientos y mejor documentación.

Probabilidad: Alta

Impacto: Alta

e) Carga de trabajo en las asignaturas que quedan por evaluar al alumno

Descripción: Podrían producirse períodos de carga de trabajo en las asignaturas pendientes de evaluación que dispone el alumno. Este riesgo puede verse incrementado en períodos de tiempo concretos como entregas de trabajos o períodos de exámenes.

Efecto: De producirse este riesgo causaría un incremento de tiempo en la realización de alguna de las tareas.

Modo de evitar/mitigar: Llevar lo más adelantado posible la materia impartida, así como los trabajos a realizar en cada una de las asignaturas.

Solución: Emplear el tiempo que sea posible a la realización de pequeñas partes de las tareas.

Probabilidad: Alta

Impacto: Alta

f) Mala planificación temporal de las tareas

Descripción: Al tratarse de un trabajo en el que se debe estudiar, comprender la estructura, modificar y configurar diversos programas, es posible que la planificación temporal estimada para ciertas tareas no sea la adecuada.

Efecto: Retraso en la finalización del proyecto.

Modo de evitar/mitigar: Intentar estimar el tiempo otorgado a cada tarea lo más acertadamente posible.

Solución: Aprovechar lo máximo posible el tiempo disponible o, en caso necesario, retrasar la fecha de entrega.

Probabilidad: Alta

Impacto: Alta

2.8 Evaluación económica

Aunque el proyecto tiene como fin la creación de una herramienta de monitorización de redes gratuita y Open Source, el coste monetario de un proyecto de estas características podría dividirse en los siguientes apartados:

a) Salario

Dado que el alumno, por motivos académicos, dispondrá de 3 horas y media al día para la realización del proyecto y la estimación temporal es de 21 semanas, esto da como resultado una estimación de 368 horas.

En la siguiente tabla se indica el salario junto con los descuentos por IRPF y Seguridad Social en el supuesto de que se cobrara 20€/hora (véase Tabla 2-2).

Salario Neto	368 horas * 20€/hora	7360€
Seguridad Social	7360€*0.12 (12%)	-883€
IRPF	7360€*0.20 (20%)	-1472€
Salario Total	7360€-883€-1766€	5005€

Tabla 2-2: Salario del informático con los descuentos del IRPF y Seguridad Social aplicados

b) Coste del Software

Debido a que todas las aplicaciones utilizadas, exceptuando Microsoft Office Word 2013, son gratuitas, el coste del software utilizado para la creación de i2BasqueNMS será (véase Tabla 2-3):

Aplicación	Coste monetario
Microsoft Office Word 2013	135€

Putty	0€
Firefox	0€
Chrome	0€
Microsoft Paint	0€
GanttProject	0€
ProxMox VE 3	0€
Nano	0€
Coste Total	135€

Tabla 2-3: Coste monetario de las aplicaciones usadas durante el proyecto

c) Inmovilizado de material y otros gastos

Para el diseño, creación y pruebas de la herramienta de monitorización se necesitarán los siguientes equipos (véase Tabla 2-4):

Equipo	Coste monetario
Ordenador portátil	700€
Servidor con máquinas virtuales	422€
Switch Juniper	2976€
Router Juniper	2000€
Router Cisco	70€
Total	6168€

Tabla 2-4: Coste de los equipos utilizados en el proyecto

En la siguiente tabla se encuentran las amortizaciones de los equipos, en base a su amortización anual, durante la realización del proyecto (véase Tabla 2-5):

Equipo		Amortización
<i>(La vida útil de los equipos utilizados es aproximadamente de 7 años)</i>	<i>(La estimación temporal del proyecto es de 21 semanas, lo que aproximadamente son 5 meses)</i>	
Ordenador portátil	$((700/7)€ * 5 \text{ meses}) / 12 \text{ meses}$	41,7€
Servidor con máquinas virtuales	$((422/7)€ * 5 \text{ meses}) / 12 \text{ meses}$	25,12€
Switch Juniper	$((2976/7)€ * 5 \text{ meses}) / 12 \text{ meses}$	177,14€
Router Juniper	$((2000/7)€ * 5 \text{ meses}) / 12 \text{ meses}$	119,04€
Router Cisco	$((70/7)€ * 5 \text{ meses}) / 12 \text{ meses}$	4,17€

Amortización Total		367,17€
---------------------------	--	----------------

Tabla 2-5: Amortización de los equipos utilizados durante la realización del proyecto

En la siguiente tabla se muestran otros gastos derivados de la realización del proyecto (véase Tabla 2-6):

Otros gastos	Coste monetario
Gasto medio de electricidad	90€
Desplazamientos (1,19€ * 40 viajes)	47,6€
Total	137,6€

Tabla 2-6: Otros gastos derivados de la realización del proyecto

d) Coste total del proyecto

El coste final del proyecto sería lo reflejado en la siguiente tabla (véase Tabla 2-7).

Salario	5005€
Amortización de los equipos	367,17€
Otros gastos	137,6€
Coste total del proyecto	5509,77€

Tabla 2-7: Coste monetario total del proyecto

3 ANTECEDENTES

La creciente expansión del uso de redes locales en las empresas para distintos fines, tales como la compartición interna de información o de elementos hardware (impresoras, discos duros, ...) entre sus trabajadores ha llevado a que cada día las redes locales sean más amplias y complejas. Esta complejidad dificulta las tareas de control y mantenimiento de los equipos de la red (switches, routers, servidores, ...), haciendo necesario el uso de herramientas que simplifiquen su monitorización y gestión.

Existen numerosas y variadas herramientas para este fin y la mayoría de empresas fabricantes de equipos de red, ofrecen versiones de prueba de software de monitorización para los equipos que suministran. En algunos casos, estos productos de prueba limitan el número de equipos y servicios que permiten monitorizar, como por ejemplo la herramienta PRTG de la compañía Paessler (<http://www.es.paessler.com/prtg>); en otros casos, limitan el número de días de uso gratuito, como la versión de prueba de Nagios para empresas. Dentro del entorno de Software Libre, también existen herramientas con este propósito, siendo éstas las que se usarán en este proyecto.

A continuación se realiza un estudio comparativo entre algunos de los distintos sistemas de monitorización OpenSource existentes.

3.1 OpenNMS

Fue la primera plataforma de administración de red que surgió a nivel empresarial. Comenzó a desarrollarse bajo licencia OpenSource.(véase *Ilustración 3-01*) Actualmente cuenta con una comunidad que da soporte al proyecto en sí, mientras que una organización se encarga del soporte comercial (cursillos de entrenamiento a los usuarios, soluciones personalizadas para empresas...)

Su página oficial es: <http://www.opennms.org/index.php>

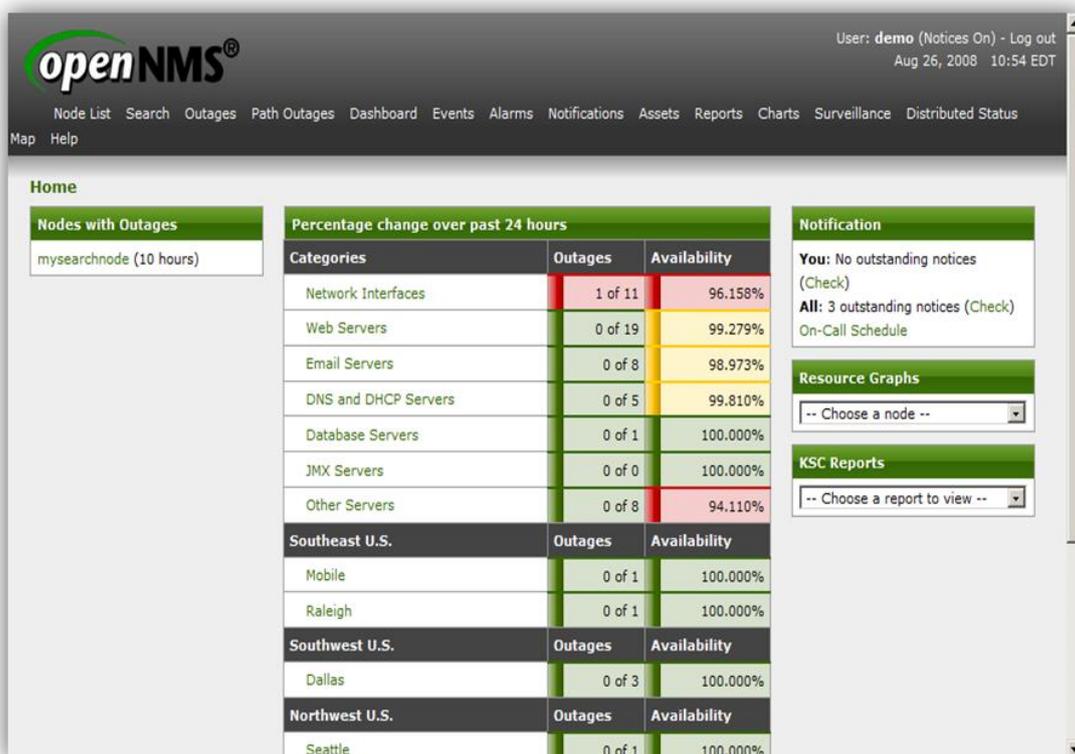


Ilustración 3-01: Interfaz web principal de OpenNMS.

Sus principales características son:

- Es capaz de realizar gráficas sencillas del estado de los equipos de la red.
- Puede realizar informes de auditorías de la red, permitiendo conocer su estado general o el de alguno de sus equipos en particular, durante periodos de tiempo establecidos por el administrador.
- Puede realizar estadísticas de uso a partir de los datos extraídos de la red.
- Es capaz de auto-descubrir servicios y equipos de una red, es decir, detectar de manera automática tanto los equipos como los servicios disponibles en ella.
- Admite el protocolo de monitorización SNMP, aunque también es compatible con sistemas basados en otros protocolos como WMI o JMX.
- Es compatible con el protocolo de envío de mensajes de registro Syslog. Esto le permite llevar un registro de accesos con contraseñas equivocadas, variaciones del funcionamiento normal del sistema, etc.
- Se le pueden añadir scripts externos de la comunidad.

- Posee algunos plugins y complementos soportados por la propia comunidad de OpenNMS.
- Gracias al soporte de su comunidad, el nivel de dificultad para la realización de plugins para OpenNMS es medio.
- Contiene un sistema de alerta de fallos en la red por correo electrónico.
- Su aplicación web permite el control total de la red; no sólo su monitorización sino también la gestión remota de algunas opciones de los equipos de red.
- Se puede realizar una monitorización distribuida de la red.
- Es compatible con distintos métodos de almacenamiento de datos como Jrobin, RRDtool o PostgreSQL.
- Se encuentra bajo la licencia GPL.
- Permite ubicar geográficamente los equipos monitorizados (véase Ilustración 3-02).

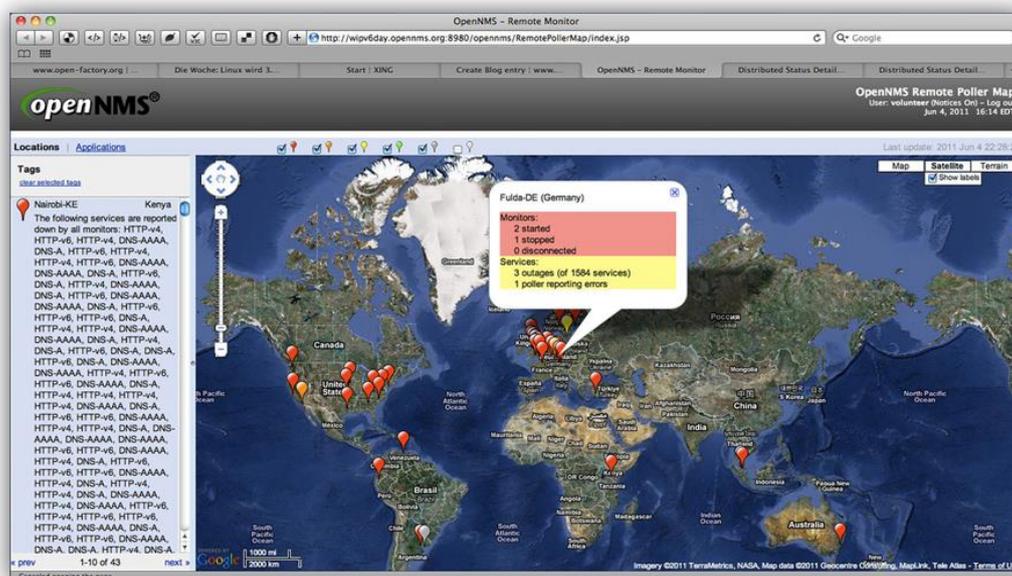


Ilustración 3-02: Gestor de mapas de OpenNMS.

- Es capaz de mostrar mapas de la red a partir de una especificación que permite describir gráficos vectoriales bidimensionales. Dicha especificación se denomina SVG (Scalable Vector Graphics).
- Es compatible con el protocolo LDAP para el control de acceso a la aplicación.
- No existen eventos enfocados única y exclusivamente a este software, pero se dan charlas sobre él en diferentes eventos relacionados con la administración de redes.

3.2 Hyperic

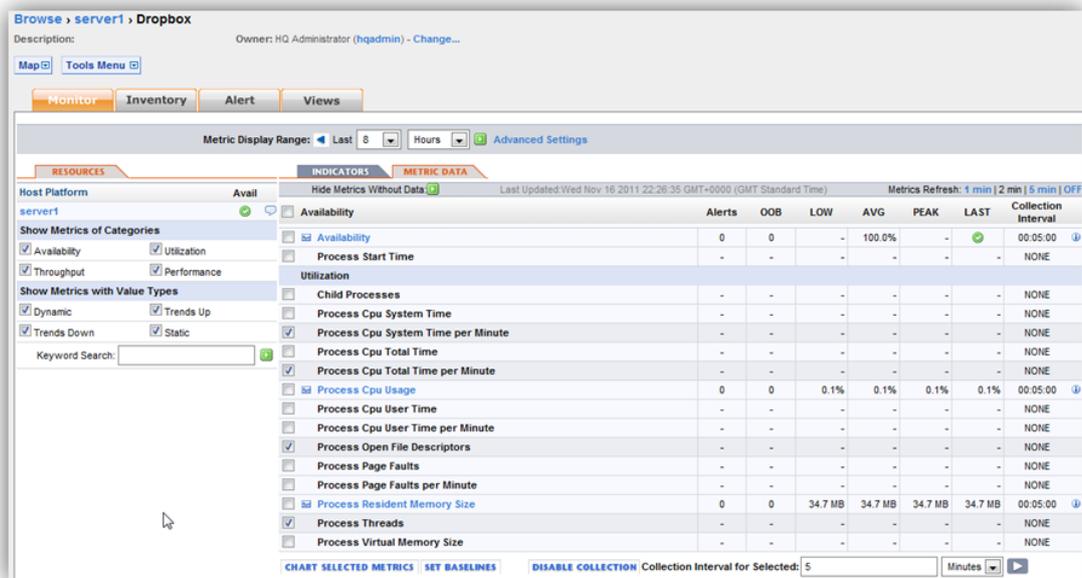
Surge de una división de la empresa VMWare (<http://www.vmware.com/es>). Incluye una parte OpenSource y otra comercial (véase Ilustración 3-03).

Su página oficial es: <http://www.hyperic.com/>

Sus principales características son:

- Es capaz de mostrar graficas de uso de los distintos equipos y servicios de la red (véase Ilustración 3-04).

- Permite dividir los equipos de la red en grupos lógicos, facilitando el visionado del estado de un servicio concreto. Es decir, permite agrupar los mismos servicios o equipos de características similares para poder tener acceso directo a toda la información genérica disponible sobre un servicio o a las características específicas de algunos equipos de la red.
- Puede auto-descubrir servicios y equipos de la red monitorizada. Es decir, puede detectar tanto los servicios como los equipos de red conectados a la red monitorizada, de manera automática.



- **Ilustración 3-03: Interfaz web principal de Hyperic.**

- Es compatible con el protocolo SNMP.
- Admite scripts externos realizados por la comunidad o por la cualquier empresa donde se use.
- Posee algunos plugins y complementos, pero no tantos como otras soluciones analizadas.
- La creación de complementos es sencilla.
- Contiene un sistema de alertas basado en el envío de correos electrónicos a los usuarios indicados por el administrador.
- Su aplicación web permite un control total sobre los equipos de red.
- Permite una monitorización distribuida de la red.
- Es compatible con distintos métodos de almacenamiento de datos como PostgreSQL, MySQL u Oracle.
- Es software libre, aunque posee alguna parte comercial como cursillos o manuales para empresas.
- Está bajo la licencia GPL.



Ilustración 3-04: Interfaz web con las gráficas proporcionadas por Hyperic.

3.3 PandoraFMS

Es un software de monitorización de redes de código abierto desarrollado por Ártica ST. Posee varios tipos de licencias, OpenSource, Small Enterprise y Enterprise (véase Ilustración 3-05).

Su página oficial es: <http://pandorafms.com>

Sus principales características son:

- Realiza gráficas del estado de los equipos de red.
- Puede realizar informes de auditorías de la red en cualquier momento o en momentos previamente establecidos por el administrador. Es posible ver el estado de los equipos o servicios en cualquier momento o enviar notificaciones en momentos previamente establecidos por el administrador.
- Es capaz de dividir los equipos de la red en grupos lógicos para un mejor seguimiento de determinados servicios, y así poder visualizar la información de todos los equipos o servicios similares de una sola vez.
- Puede realizar estadísticas de uso a partir de los datos recogidos en la red.
- En su versión de pago, utilizando las distintas estadísticas realizadas, es capaz de predecir fallos que podrían ocurrir en la red.
- En su versión de pago, descubre los servicios y equipos de la red. Es decir, puede descubrir automáticamente los equipos conectados a la red, así como los servicios disponibles.



Ilustración 3-05: Interfaz web principal de PandoraFMS.

- Es compatible con el protocolo SNMP.
- La monitorización se realiza mediante los agentes SNMP de los distintos equipos. Mediante unos programas instalados y configurados en los equipos de red, PandoraFMS recaba toda la información sobre el estado de éstos.
- Es compatible con el protocolo de envío de mensajes de registro Syslog, lo que le permite llevar un registro de accesos con contraseñas equivocadas, variaciones del funcionamiento normal del sistema, etc.
- Admite scripts externos.



Ilustración 3-06: Interfaz web con las uniones entre equipos de una red en PandoraFMS.

- Dispone de plugins y complementos, aunque en gran medida restringidos a las versiones de pago.

- La realización y añadido de complementos propios para Pandora es sencilla.
- Contiene un sistema de alertas basado en correos electrónicos enviados a los usuarios seleccionados previamente por el administrador.
- La aplicación web propia de Pandora permite un control total sobre la red.
- Permite una monitorización distribuida de la red.
- Su método de almacenamiento de datos por defecto es MySQL, aunque mediante plugins es compatible con otros como MongoDB.
- La versión libre de Pandora se encuentra bajo la licencia GPL.
- La versión de pago es capaz de realizar mapas de la red (véase Ilustración 3-06).
- Posee un control de acceso granular, es decir, permite que el administrador configure hasta qué punto tienen acceso a la información de la red los distintos usuarios.
- Existen eventos, sobre todo dirigidos a empresas, para dar a conocer el software.

3.4 Nagios

Es un software OpenSource, licenciado bajo la GNU General Public License v.2 por la Free Software Foundation. También ofrece servicios comerciales de soporte y formación, además de libros y manuales (véase Ilustración 3-07).

Su página oficial es: <http://www.nagios.org/>

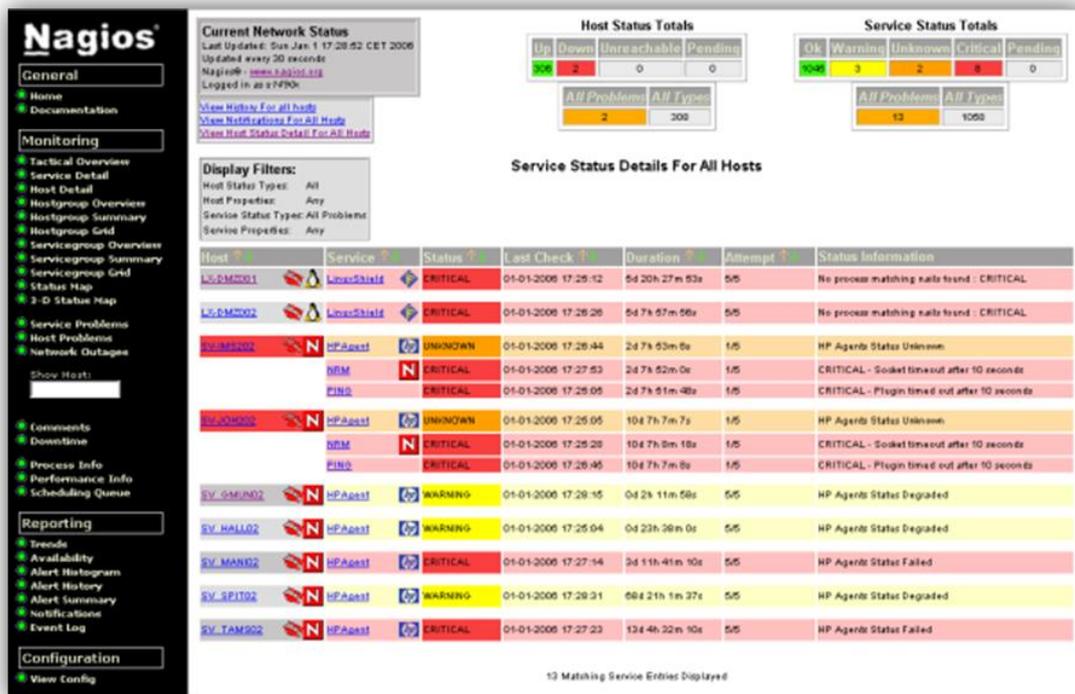


Ilustración 3-07: Interfaz web principal de Nagios.

Sus principales características son:

- Es capaz de realizar gráficas para representar el uso de diversos equipos de la red así como de sus componentes.
- Puede elaborar informes de auditorías de la red.

- Permite la división y el filtrado de los equipos mediante la generación de grupos lógicos, que posibilitando que el administrador visualice todos los servicios o equipos similares, de una sola vez.
- Puede realizar estadísticas del uso de la red a partir de los datos recogidos.
- En su versión de pago, es capaz de realizar posibles predicciones basadas en las estadísticas de uso previamente realizadas, lo que facilita al administrador la prevención de errores en la red.
- Su versión de pago es capaz de auto-descubrir equipos y servicios. Es decir, localizar y especificar dichos equipos, así como los servicios disponibles en ellos.
- Es compatible con el protocolo SNMP.
- La versión de pago también es compatible con el protocolo de red Syslog, que le permite realizar un seguimiento de los accesos correctos o incorrectos en los equipos monitorizados.
- Posee una gran cantidad de plugins y complementos desarrollados por la empresa encargada de Nagios y/o por su comunidad, además de admitir scripts externos.
- Aunque es posible añadir plugins propios de los usuarios, resulta complicado, ya que al disponer de una versión de pago, no muestran ningún interés en que la versión gratuita llegue a poseer las mismas funcionalidades.
- El sistema de aviso de fallos se basa en la utilización de alertas por correo electrónico o por SMS. El aviso por correo es gratuito y se realiza a través de un servicio llamado “sendmail” instalado en el propio servidor de monitorización. Sin embargo, los avisos por SMS requieren disponer de un aparato con tarjeta SIM conectado al servidor (centurión) para poder enviar los avisos a través de la de red móvil GSM. En su defecto, es posible subcontratar el servicio de envío de SMS a una empresa externa.
- Tiene su propia aplicación web, aunque los administradores suelen usar soluciones de terceros como Centreon. La web oficial de este es: <https://www.centreon.com/en/products/centreon/centreon-engine/>
- Es posible realizar una monitorización distribuida de la red mediante la inclusión de varios sistemas de monitorización, ya sea como soporte en caso de fallo en el principal o por cuestiones de mantenimiento de este. Si la red es muy amplia, también es posible que la monitorización se pueda distribuir entre distintos servidores, para que todo el trabajo no recaiga sobre un solo servidor central.
- Por si mismo solo es compatible con SQL, aunque mediante plugins es posible usar otros métodos de almacenamiento de datos como MongoDB.
- La versión libre de Nagios está bajo la licencia GPL.
- Solo puede mostrar mapas a través de un plugin llamado NagVis.
- Posee seguridad en el acceso al propio Nagios, así como en las conexiones realizadas vía SSH con los distintos equipos de la red.

Como apunte cabe destacar que, aunque la comunidad de Nagios está cada día más dividida, en uno de sus últimos intentos por mantenerse en la cúspide de los sistemas de monitorización libres (Septiembre de 2013), se añadió compatibilidad con el protocolo Ipv6.

3.5 Shinken

Se trata de una re-implementación de Nagios en Python [6], compatible con todos sus plugins y complementos pero con la ventaja de que su programación modular le permite gestionar un mayor número de componentes de red. Concretamente, Nagios, a duras penas, es capaz de monitorizar 10000 equipos y servicios (cifra nada desdeñable cuando se presentó en 1999, pero limitada de cara al futuro), mientras que Shinken ha llegado a monitorizar entre 100000 y 150000 sin mayor problema (véase *Ilustración 3-08*).

Su página oficial es: <http://shinken-monitoring.org/>

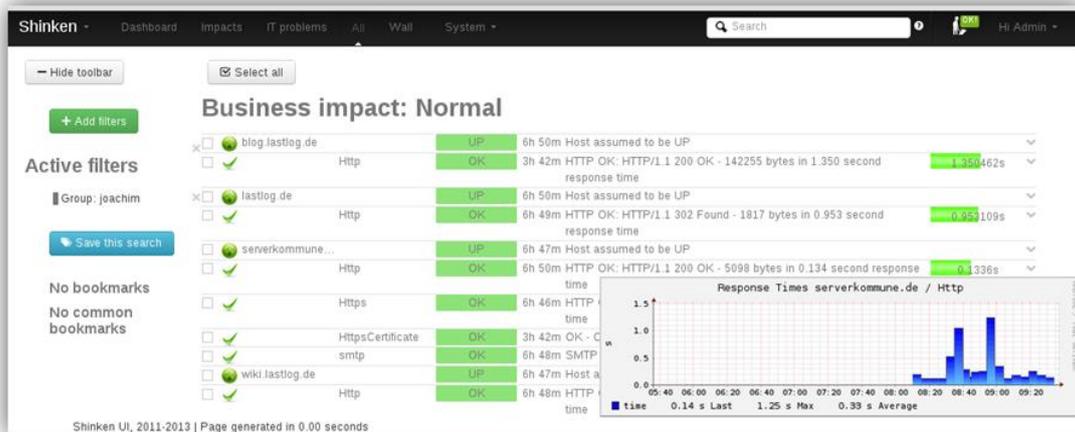


Ilustración 3-08: Interfaz web WebUI de Shinken, en la pestaña donde aparecen todos los equipos.

Sus principales características son:

- Es capaz de mostrar gráficas del estado de los equipos de la red, así como de sus componentes concretos (véase *ilustración 3-08*).
- Además de enviar avisos al producirse errores en la red; Shinken también puede enviar informes de su estado o del de ciertos servicios y/o componentes de algunos equipos que puedan resultar vitales. Dichos avisos, los configura el administrador del sistema, indicando el nombre del servicio, equipo o componente, el destinatario y el intervalo de tiempo en el que se deben enviarse. Se llaman informes SLA.
- Divide la red en grupos lógicos, permitiendo filtrar equipos por sus servicios y no sólo por su pertenencia a alguna subred.
- Con todos los datos que posee de la red, es capaz de realizar estadísticas que permitan al administrador conocer valores precisos del uso general que hacen los usuarios de la red y, por ende, qué partes de la red podrían ser más proclives a fallos y/o vitales para el funcionamiento de esta.
- Descubre equipos y servicios de la red a través de un plugin soportado por la propia comunidad de Shinken.
- Es compatible con el protocolo de gestión de redes, SNMP y con Syslog a través de un plugin
- La monitorización de los equipos se realiza mediante los agentes SNMP instalados en dichos equipos.

- Permite usar scripts externos realizados.
- Posee plugins y complementos propios como el de Autodescubrimiento de equipos y servicios, pero también es compatible con todos los de Nagios.
- Admite plugins realizados por usuarios de Shinken; no obstante, aunque la comunidad da un gran soporte, no es tan sencillo implementar y añadir plugins propios como en otras soluciones de monitorización. Posee un sistema de alertas a través de correos electrónicos y/o mensajes por SMS. Los avisos por correo se realizan a través de un servicio instalado en el propio servidor que se ejecuta través del comando "mail". Para los envíos SMS, es necesario poseer un elemento hardware llamado centurión o contratar un servicio externo de envío de SMS.
- Tiene su propia aplicación web llamada WebUI, aunque como es compatible con los complementos de Nagios, se le pueden añadir otras opciones web de terceros.
- Puede realizar una monitorización distribuida de los equipos de red, existiendo en una sola red varios sistemas de monitorización, ya sea como apoyo por si uno falla o para dividir la carga si la red es muy amplia.
- Es compatible con distintos métodos de almacenamiento de datos, algunos de ellos a través de plugins como es el caso de MongoDB.
- Se encuentra bajo la licencia AGPL.
- Puede realizar mapas simples de red en los que se muestran las distintas conexiones entre los equipos.
- Posee seguridad en el control de acceso a los datos de la red, así como en la conexión a los distintos componentes mediante SSH.

3.6 Conclusiones

Dado que la idea de este trabajo, así como la infraestructura para realizar las pruebas fueron cedidas por I2Basque, sus necesidades, opiniones y experiencia en la gestión y monitorización de redes marcaron el punto de partida para hacer el estudio de antecedentes.

En las primeras conversaciones propusieron considerar la posibilidad de usar Nagios como sistema base de monitorización, resaltando la necesidad de realizar un estudio previo para determinar su validez de cara a cubrir todas las necesidades planteadas. En este estudio se detectó el deterioro de la comunidad OpenSource de esta herramienta, así como y el creciente abandono del soporte ofrecido por los propios desarrolladores, que estaban optando por centrarse en la solución de pago en detrimento de la opción libre. A la vista de esta situación, se decidió optar por otro sistema libre de características similares.

Tras el estudio de las distintas posibilidades disponibles se determinó que Shinken podría ser una buena opción, dado que:

- Mientras que el resto de soluciones poseen opciones privativas o directamente se están convirtiendo en soluciones únicamente de pago, Shinken es un software completamente OpenSource, con una comunidad cada vez mayor.
- Todo el proyecto está disponible en GitHub, lo que permite a cualquier usuario adaptarlo a sus necesidades e incluso colaborar para mejorarlo o dar solución a los bugs que pudiera tener.

- Aun tratándose de una solución completamente libre, permite monitorizar desde los servicios hasta el último componente hardware de los equipos de red, de forma similar a lo ofrecido por otras soluciones de pago. También incluye alertas que se emiten al producirse el más mínimo fallo en la red, y contempla el envío de informes sobre el estado de los servicios, equipos y componentes en períodos de tiempo establecidos por el administrador.
- Aunque es posible añadir interfaces gráficas de terceros, la interfaz propia de Shinken proporciona toda la información del estado de la red sin saturar a los usuarios, además de resultar más “sencilla” que otras posibilidades. Esta interfaz permite, entre otras funcionalidades, filtrar, buscar y ver información general de la red, información de grupos de equipos o servicios, información del estado de equipos o servicios concretos, información explícita de un componente hardware de un equipo concreto de la red.
- Se trata de un sistema de monitorización con grandes posibilidades de ampliación y mejora, tanto a partir de plugins (entre los que se encuentran los gratuitos de Nagios, puesto que Shinken es una re-implementación de éste), como con la modificación directa de su código disponible en GitHub. Esto permite una mejor adaptación del software a las necesidades concretas de cada empresa.
- Debido en gran parte a su programación modular y multi-hilo, el nivel de escalabilidad de Shinken de cara al futuro es muy superior al de Nagios, dándole la capacidad de monitorizar un mayor número de equipos y de aprovechar mejor las capacidades de los servidores en los que se instale.

En el siguiente cuadro se resumen las características principales de los softwares de monitorización revisados (*véase Tabla 3-1*).

A continuación de la Tabla 3-1 se presentan los requisitos que debe satisfacer la herramienta de monitorización planteada por i2Basque.

Nombre	Gráficas	Informes SLA	Grupos lógicos	Estadísticas	Predicción de estadísticas	Autodiscover	Agentes	SNMP	Syslog	Scripts externos	Complementos (Plugins)	Creación de complementos	Alertas	Aplicación web	Monitorización distribuida	Método de almacenaje de datos	Licencia	Mapas	Seguridad	Eventos
Shinken	Si	Si	Si	Si	No	Si	Si	P*	P*	Si	Si (Compatible con todos los de Nagios)	Media	Si	Solo visualización	Si	Flatfile, MySQL, Oracle, Graphite, cs Sqlite, MongoDB	AGPL	Si	Si	Si
Nagios	Si	Si	Si	Si	Si*	Si*	Si	P*	Si*	Si	Si	Media	Si	Solo visualización	Si	SQL	GPL; Comercial (versión Enterprise)	No	Si	Si
OpenNMS	Si	Si	No	Si	No	Si	SNMP, WMI, JMX, usando NRPE	Si	Si	Si	Si	Media	Enrutamiento, escalas y horarios	Control total	cliente mínimo o snmp proxy	jrobin, RRDtool PostgreSQL	GPL	Pantallas dinámicas con Svg	LDAP y Pantallas con dashboard	Si
Hyperic	Si	Si*	Si	Si*	Si*	Si	Si	Si	Si*	Si	Si	Fácil	Si	Control total	Si	PostgreSQL, MySQL, Oracle	GPL; Comercial (versión Enterprise)	No	Si	Si
Pandora	Si	En tiempo real o programados	Si	Si	Si	Si	Con y sin agente	Si	Si	Si	Si	Fácil	Si	Control total	Si	MySQL	GPL	Mapas de red automáticos definibles por el usuario con edición interactiva	Control de acceso granular afinado	Si

Si* = en la versión de pago; P*= Mediante plugin

Tabla 3-1: comparación general entre las distintas soluciones estudiada

4 CAPTURA DE REQUISITOS

Debido a que el proyecto tiene como fin la monitorización de equipos y servicios de red, se comentan por separado la captura de requisitos para las funcionalidades software y la compatibilidad con distintos tipos de hardware.

4.1 Funcionalidades software

En términos de funcionalidades software, desde el comienzo de este trabajo se ha partido de dos premisas:

1. La herramienta final i2BasqueNMS debe funcionar sobre un sistema Linux.
2. Utilizará software libre.

Otros requisitos adicionales propuestos desde i2basque para la solución de monitorización son:

- Una base software de monitorización con alta escalabilidad de cara a poder abordar futuras expansiones de la red.
- Una sencilla interfaz web que permita administrar la red de manera fácil e intuitiva.
- Permitir visualizar gráficos de uso de los distintos equipos de red monitorizados.
- Visualización de mapas con la localización de los distintos equipos y servicios de la red.
- Auto-detección de los equipos de la red y de los servicios que ofrece cada uno de ellos.
- Informar mediante el envío de e-mails, de problemas que surjan en la red.
- Enviar, a través de emails, informes del estado de la red en períodos de tiempo establecidos por el administrador.

Como extra, también se ha incluido un plugin de Nagios que permita la monitorización de equipos con sistema operativo Windows.

4.2 Compatibilidades hardware

La máquina que se usará para la monitorización de la red será un nodo de un DELL PowerEdge C6220, con las siguientes características (véase Tabla 4-1):

	<ul style="list-style-type: none"> • 2 x Xeon E5-2630 @ 3.30 GHz (24 núcleos) • 32 GB RAM • 2 x 1TB Disco duro • 2 x Intel I350 Gigabit
---	---

Tabla 4-1: servidor en el que se instalará la solución.

Los equipos a monitorizar (a día de hoy) son 39, de los tipos y modelos que se muestran a continuación.

Equipos de enrutamiento

a) Nueve routers Juniper (véase Tablas 4-2 y 4-3):

<p>4 x Juniper M40e</p> 	<p>1 x Juniper MX240</p>  <hr/> <p>4 x Juniper MX80</p> 
--	---

Tabla 4-2: Routers de la marca Juniper que se dispondrá en la red

Características / Equipos de red	Juniper M40e	Juniper MX240	Juniper MX80
Conexión óptica Half-Duplex	51,2 Gbit/s	240 Gbit/s	-
Conexión óptica Full-Duplex	25,6 Gbit/s	120 Gbit/s	-
Ranura FPC	8 ranuras FPC	2 ranuras FPC	2 ranuras FPC
Full-Duplex a través de cada slot	3,2 Gbit/s	-	-
Conexión Ethernet 10Gigabit	0	24/48	20
Conexiones PCI por chasis	32	-	-
Numero de chasis por rack	2	2	4
Redundancia	Si	Si	Si

Tabla 4-3: características de los routers de la tabla 4-2

b) Doce equipos ópticos Infinera (véase Tablas 4-4 y 4-5):

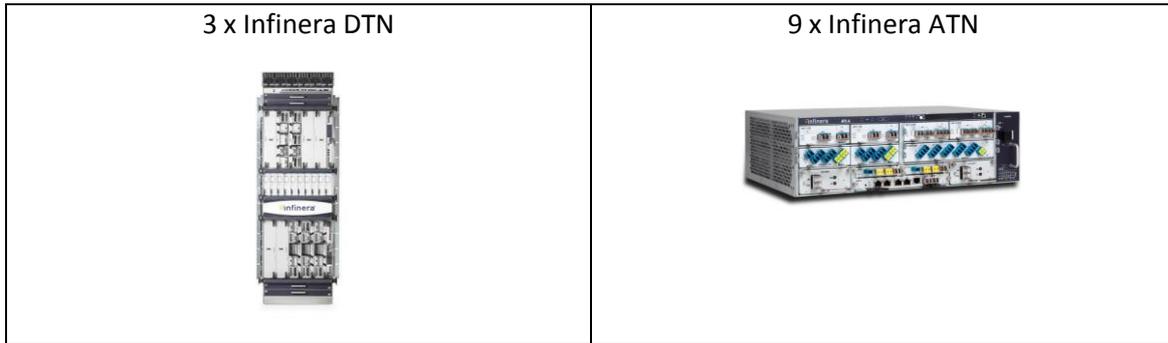


Tabla 4-4: equipos de la marca Infinera que se dispondrán en la red

Característica / Equipo	<u>Infinera DTN</u>	<u>Infinera ATN</u>
Capacidad de transmisión	6,4Tb/s	800Gb/s

Tabla 4-5: característica principal de los equipos ópticos Infinera del cuadro 4-4

c) Ocho Switches Juniper (véase Tablas 4-6 y 4-7):

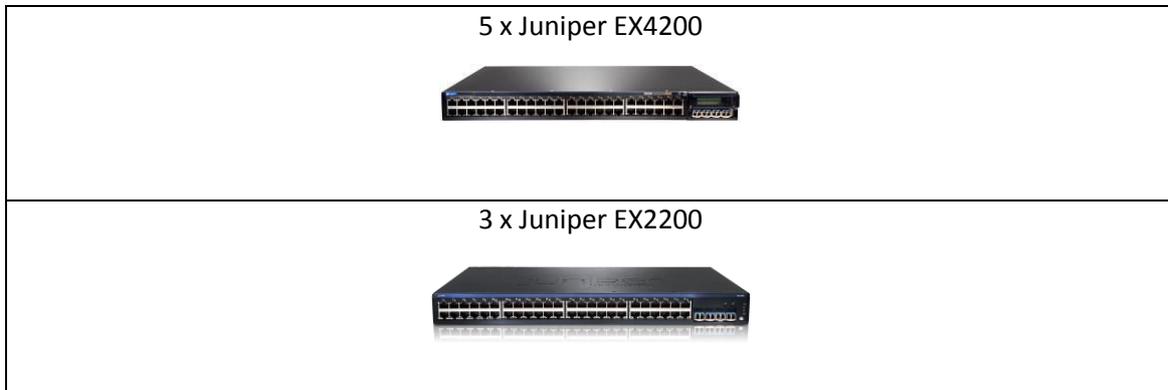


Tabla 4-6: switches de la marca Juniper que se dispondrán en la red

Características / Equipos	<u>Juniper EX4200</u>	<u>Juniper EX2200</u>
Puertos 10/100/1000BASE-T	24 / 48	24 / 48
10-Gigabit Ethernet	4	4

Tabla 4-7: características principales de los switches de la tabla 4-6.

Equipos servidores

a) Diez servidores de distintos fabricantes (véase Tablas 4-8 y 4-9):

1 x DELL PowerEdge C6220 	1 x DELL PowerEdge C6100 
2 x DELL PowerEdge 1950 	1 x SUN Fire X4150 
1 x SUN Sparc Enterprise T5120 	2 x Apple XSERVE 
1 x Fujitsu RX200S7 	1 x Fujitsu ETERNUS DX60 

Tabla 4-8: Servidores de los que se dispondrán en la red final de i2Basque

Características / Equipos	DELL PowerEdge C6220	DELL PowerEdge C6100	DELL PowerEdge 1950	SUN Fire X4150	SUN Sparc Enterprise T5120	Apple XSERVE	Fujitsu RX200S7	Fujitsu ETERNUS DX60
Sistema Operativo	Microsoft Windows Server 2012 R2	Novell SUSE Linux Enterprise Server 11 SP2	Microsoft Windows Server 2012 R2	Solaris 10 OS	Solaris 10 OS	Mac OS X server	Microsoft Windows Server 2012 R2	Microsoft® Windows Server® 2012 R2 Datacenter
CPU	2 CPUs de hasta 8 núcleos Intel XEON por CPU	2 CPUs de hasta 4 núcleos Intel XEON por CPU	2 CPUs de hasta 4 núcleos Intel XEON por CPU	2 CPUs de hasta 4 núcleos Intel XEON por CPU	1 CPU con 1 núcleo UltraSPARC	2 CPUs, hasta 2 núcleos Power G5 por CPU	2 CPUs, hasta 8 núcleos Intel XEON por CPU	1 CPU, hasta 4 núcleos Intel XEON
Memoria RAM	64BG	64GB	32GB	64 GB	64 GB	16GB	64BG	64GB
Capacidad de Almacenamiento	36TB	Sata 24TB / SAS 36TB	2TB Raid 1	300GB	873GB	1.5TB	768GB	28.8TB
Conexiones de Red	Controladora Intel Ethernet i350 - 2 x 1Gb Ethernet 1 x 100Mb Ethernet	Intel 82576 – 2 x Gb Ethernet y 1 x 100Mb Ethernet	Broadcom NetXtreme II Dual 5708 Gigabit Ethernet	4 puertos, 10/100/1000 Mbps	4 puertos, 10/100/1000 Mbps	2 puertos Gigabit Ethernet	I350, 2 x 10/100/1000 Mbit/s	Ethernet (1000 Base-T / 100 Base-TX / 10 Base-T)

Tabla 4-9: Características de los servidores de la tabla 4-8

5 ANALISIS Y DISEÑO

Para diseñar la solución completa i2BasqueNMS de cara a los requisitos en términos de monitorización exigidos por el cliente (i2Basque), ha sido necesaria la inclusión, junto al software central Shinken, de diversos plugins de Nagios (precursor de Shinken), así como de un plugin externo. Así mismo, para que el software de monitorización cumpla los requisitos, se han añadido 4 programas externos.

5.1 Plugins añadidos a Shinken

Dado que es imposible conocer los equipos o servicios que a largo plazo, se añadirán a la red del cliente, se ha considerado apropiado la inclusión de diversos plugins que, junto a los propios de Shinken, incrementen la compatibilidad i2BasqueNMS con el mayor número de equipos y servicios posibles. Dichos plugins son:

- a) **nagios-plugins:** pack con plugins de monitorización basados en el protocolo SNMP contenidos por defecto en Nagios
- b) **check_netint:** plugin inicialmente diseñado para Nagios, cuya finalidad es mejorar la monitorización de las interfaces de red de los equipos.
- c) **check_nwc_health:** plugin disponible en GitHub, cuyo fin es la monitorización de componentes de red. Abarca componentes de la mayoría de las marcas y modelos más conocidos, tales como: Cisco, Nortel, HP, Allied Telesyn, etc.
- d) **check_snmp_bandwidth:** Plugin para monitorizar el estado y el uso de los puertos de los switches de una red.
- e) **manubulon:** pack de plugins creado con el propósito de abarcar el mayor número de componentes de red posibles.

5.2 Interfaz web para Shinken

En vista de las preferencias del cliente respecto al aspecto y usabilidad deseada de la interfaz web para Shinken, fue necesario escoger entre una interfaz web llamada Multisite y otra (propia de Shinken) llamada WebUI.

a) Multisite

Como se puede apreciar en la, la interfaz web Multisite (véase *Ilustración 5-01*) satura al administrador de la red con demasiada información en pantalla al mismo tiempo.



Ilustración 5-01: Pantalla principal de la interfaz web Multisite.

b) WebUI

Mientras que la interfaz web WebUI ofrece una visión de los datos de la red más ligera, tal y como se puede observar (véase ilustración 5-02).

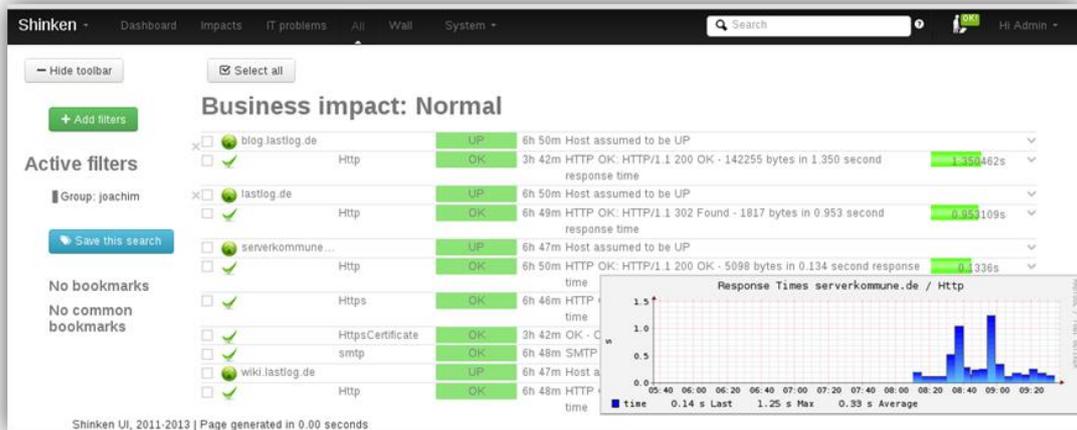


Ilustración 5-02: Pantalla de información completa de todos los equipos de la interfaz web WebUI.

En base a la claridad con la que ofrece los datos WebUI (véase Ilustración 5-02) en comparación con la saturación ofrecida por Multisite (véase Ilustración 5-01), se decidió optar por el uso de la primera.

5.3 Gráficos de uso de los equipos y servicios monitorizados

Para dar respuesta al requisito de generar gráficas de uso a partir de los datos ofrecidos por los equipos y servicios monitorizados, se barajaron dos opciones viables:

a) Graphite



Ilustración 5-03: Interfaz web de Graphite

b) PNP4Nagios



Ilustración 5-04: Interfaz web de PNP4Nagios

Como se puede observar (véase Ilustraciones 5-03 y 5-04) ambas opciones ofrecen gráficas similares, pero PNP4Nagios [8] lleva más tiempo en desarrollo, lo que le aporta

mayor confianza en cuanto a su estabilidad con respecto a Graphite. Por ello, se decidió usar PNP4Nagios como programa para la creación de gráficos a partir de los datos de monitorización de la red aportados por Shinken.

5.4 Visualización de mapas

Para mostrar mapas y/o imágenes de los equipos o servicios monitorizados junto con sus datos en tiempo real, solo se ha encontrado una opción OpenSource disponible: NagVis [9] (véase Ilustración 5-05).



Ilustración 5-05: Interfaz web con el mapa de Europa de NagVis.

Por lo tanto, la única opción posible fue integrar este software desarrollado para Nagios, como parte de la solución diseñada con Shinken.

5.5 Terminal web para conexión directa con el servidor

Con el fin de facilitar la administración del sistema, se ha optado por añadir a i2BasqueNMS un terminal que permita al administrador una conexión directa y segura al servidor desde la propia interfaz web. Para ello se ha optado por el uso de Shellinabox. Un terminal web al que se accede a través del puerto 4200 mediante una conexión segura SSL [10].

5.6 Auto-detección de los equipos y los servicios disponibles en la red

Para que la solución sea capaz de escanear la red en busca de equipos y servicios, se ha recurrido al programa Nmap [11] (véase *Ilustración 5-06*) al que se accede mediante el uso de la terminal web añadida a la herramienta de monitorización.

No existen más opciones OpenSource para este fin. Esta funcionalidad suele ser una de las más solicitadas por los administradores y por lo tanto una de las funciones más publicitadas por los desarrolladores.

```
Starting Nmap 6.00 ( http://nmap.org ) at 2015-03-23 09:56 CET
Nmap scan report for 150.241.128.121
Host is up (0.00058s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    filtered telnet
MAC Address: 00:05:05:D8:08:5D (Juniper Networks)
Device type: router
Running: Juniper JUNOS 7.X|9.X
OS CPE: cpe:/o:juniper:junos:7 cpe:/o:juniper:junos:8
OS details: Juniper Networks M10 or M320 router (JUNOS 7.4R3.4), Juniper Networks M series router (JUNOS 8.3R2.8)
Network Distance: 1 hop

Nmap scan report for 150.241.128.122
Host is up (0.00075s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    filtered telnet
MAC Address: 80:71:1F:CF:ED:40 (Juniper Networks)
Device type: router|switch
Running: Juniper JUNOS 8.X|9.X|10.X
OS CPE: cpe:/o:juniper:junos:8 cpe:/o:juniper:junos:9 cpe:/o:juniper:junos:10
OS details: Juniper Networks J2320 router; or EX2200, EX3200, EX4200, or EX8200 switch (JUNOS 8.5 - 10.0)
Network Distance: 1 hop

Nmap scan report for 150.241.128.123
Host is up (0.00017s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
3128/tcp  open  squid-http
6000/tcp  open  X11
6001/tcp  open  X11:1
6002/tcp  open  X11:2
MAC Address: 00:15:C5:E8:E0:FF (Dell)
No exact OS matches for host (If you know what OS is running on it, see http://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=6.00%E=4%D=3/23%OT=22%CT=1%CU=39080%PV=N%DS=1%DC=D%G=Y%M=0015C5%T
OS:M=550FD560%P=x86_64-unknown-linux-gnu)SEQ(SP=106%GCD=1%ISR=109%TI=24%CI=2
OS:YI=1%TS=A)OPS(OI=M5B4ST1LNW%OZ=M5B4ST1LNW%OG=M5B4NNV1LNW%O4=M5B4ST11
OS:NBW%OS=M5B4ST1LNW%O6=M5B4ST11)WIN(WI=3890%WZ=3890%WG=3890%WA=3890%MS=38
OS:901%MG=3890)ECN(R=Y%DF=Y%T=41%W=3908%O=M5B4NNV71CC%YQ=)RI(R=Y%DF=Y%T=4
OS:1%RS=0%A=S%AF=RS%RD=0%Q)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=41%W=0%IS=A%AS=RF%
OS:%RD=0%Q)T5(R=Y%DF=Y%T=41%W=0%IS=S%AF=AR%O=1%RD=0%Q)T6(R=Y%DF=Y%T=41
OS:W=0%IS=A%AS=RF%RD=0%Q)T7(R=Y%DF=Y%T=41%W=0%IS=2%A=S%AF=AR%O=1%RD=0%Q
OS:U)U1(R=Y%DF=N%T=41%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y
OS:DFI=N%T=41%CD=S)
```

Ilustración 5-06: Software Nmap ejecutándose en una terminal.

5.7 Envío de avisos e informes del estado de la red

Existen diversas opciones al respecto, pero desde i2Basque se comunicó que su administrador de red se encargará de gestionar el servicio de correo y que por lo tanto no se debe cambiar nada al respecto. Aunque en i2Basque se usa la versión estable de Debian de 64bits, que por defecto lleva integrado Exim 4, el administrador usa Postfix.

Por último se muestra un esquema de la estructura completa de la solución diseñada (véase *Ilustración 5-07*). Se puede apreciar cómo Shinken junto con los plugins añadidos, se interconecta con las aplicaciones externas PNP4Nagios, Nagvis y el servidor de correo, ya sea a través del módulo Liveness, o directamente como en el caso del servidor de correo

También se puede apreciar como Shinken, PNP4Nagios y NagVis engloban una solución cerrada de cara a la interfaz web con la que interactúa el usuario. Y como, a través de la terminal web añadida en dicha interfaz, el usuario tiene la posibilidad de conectar directamente con el servidor y ejecutar la aplicación Nmap instalada en este, así como de administrar el servicio de monitorización al completo.

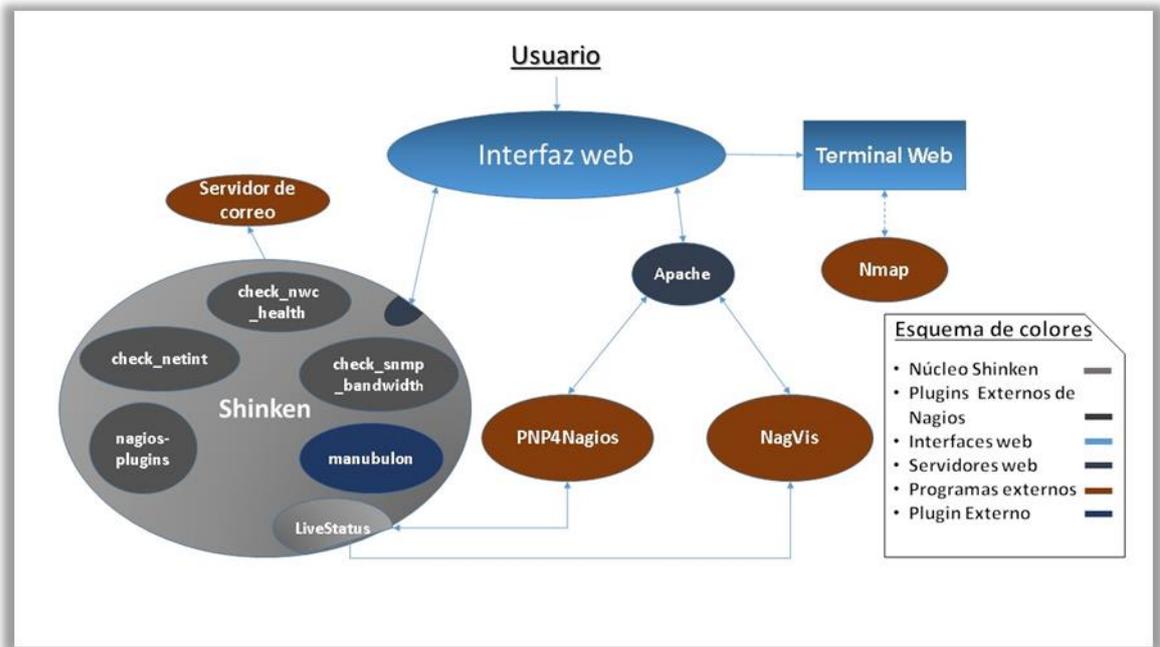


Ilustración 5-07: Esquema completo de la estructura de la solución diseñada

Todas las pruebas realizadas de los distintos prototipos de la solución que se detallan en el siguiente apartado de la memoria se realizan sobre una maqueta, proporcionada por i2Basque, que contiene equipos y servicios similares a los que se encontrarán en la red final.

6 DESARROLLO

En la implementación del proyecto se ha seguido un proceso de desarrollo basado en prototipos incrementales. En este proceso, cada prototipo añade funcionalidades, modificaciones o correcciones al anterior, hasta conseguir el adecuado a la solución de monitorización i2BasqueNMS diseñada.

6.1 Primer prototipo

En el primer prototipo se construyó el núcleo de la herramienta objetivo de este proyecto. Dicho núcleo debe ser capaz de monitorizar la red, enviar avisos e informes a través de correo electrónico y realizar gráficas con los datos recabados durante la monitorización. Para conseguirlo fue necesario configurar los distintos componentes software seleccionados durante la fase de análisis y sus interconexiones así como comprobar que todos se inician correctamente.

Herramientas integradas

Para construir este prototipo se utilizó Shinken como herramienta básica de monitorización y PNP4Nagios para la construcción de gráficas de uso.

La primera acción fue conseguir que ambas herramientas se inicializaran correctamente, proceso que no resultó tan trivial como parecía.

Inicialización

- Shinken:

En la inicialización de Shinken surgieron algunos problemas que se describen a continuación, junto con sus causas y la solución desarrollada para solventarlos.

- a. El script de instalación ofrecido por los desarrolladores daba problemas. tras revisarlo, se detectó que esos problemas eran debidos a que este script realiza una comprobación para determinar si el sistema dispone de los paquetes necesarios para la correcta instalación de Shinken. En teoría, si no encuentra alguno de estos paquetes, lo instala automáticamente, pero la realidad es que no lo hace, por lo que posteriormente se producen errores graves durante la ejecución de Shinken [7].
- b. La generación de las gráficas que deberían mostrar información de los equipos monitorizadas dio problemas. Los fallos surgían al generar las gráficas de cualquiera de los componentes monitorizados, tras pulsar en la opción de tiempo "1 day" y seguidamente al botón "show more" (véase *Ilustración 6-01*). Finalmente, se determinó que este fallo se producía porque la url que debía re-direccionar al usuario a la página de PNP4Nagios con las gráficas de dicho componente era incorrecta.

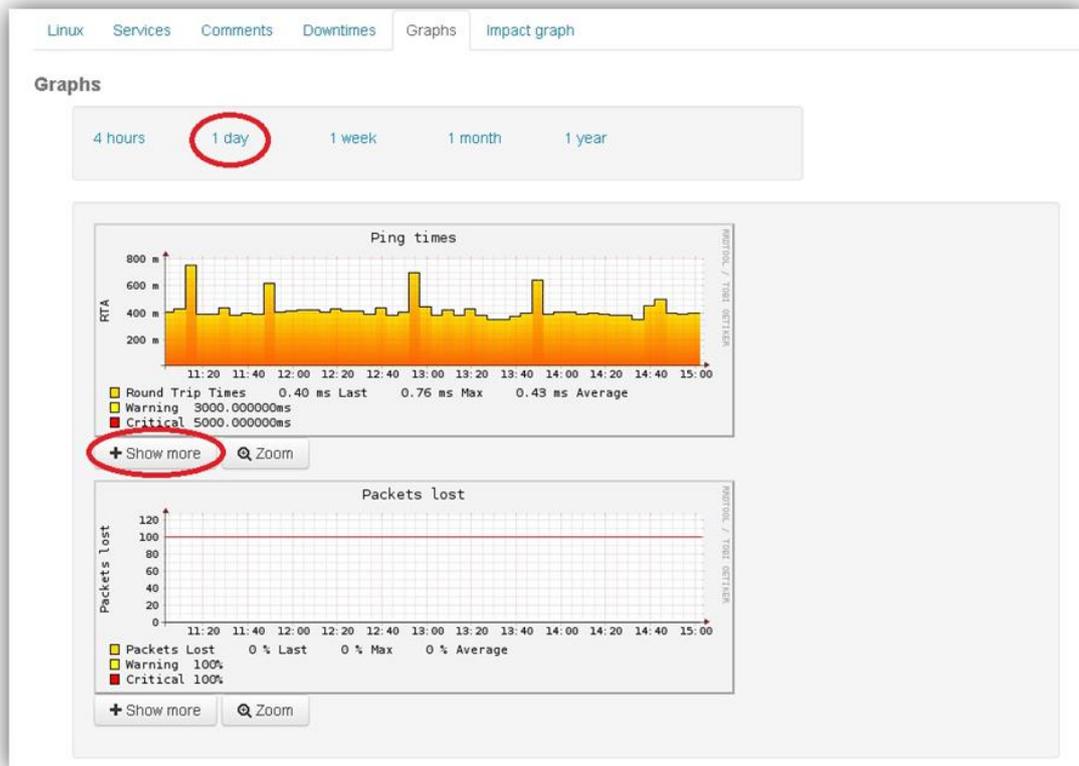


Ilustración 6-01: Botones que ocasionaban error en el direccionamiento a PNP4Nagios

- c. Dos botones de la interfaz web WebUI (véase Ilustración 6-02), que deberían direccionar al usuario a la documentación oficial, direccionan a un servidor en desuso que tenían los desarrolladores.

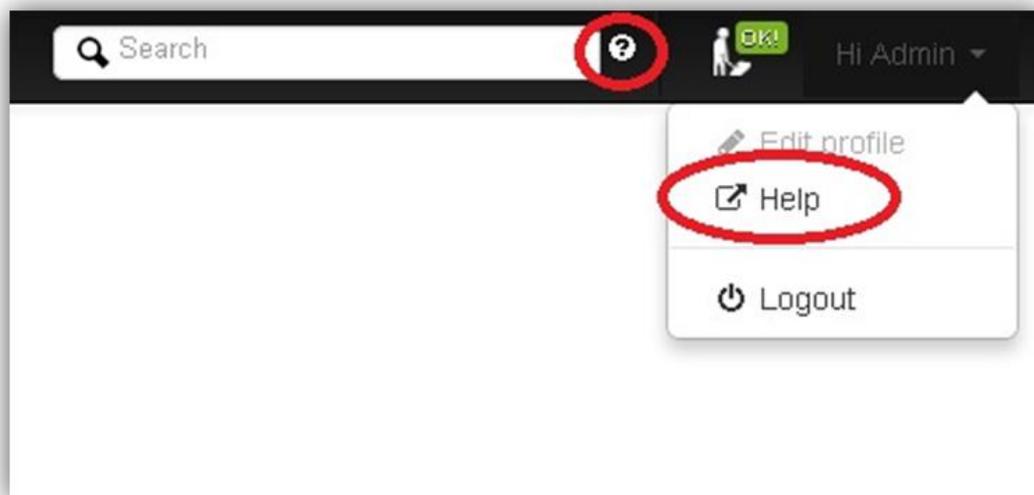


Ilustración 6-02: Botones con direccionamiento erróneo a la documentación

- d. El último error se detectó en la pestaña Dashboard (la sección de WebUI donde se muestra de manera resumida, mediante widgets el estado de la red) y estaba provocado por la ausencia de un módulo de mongodb. Es de resaltar que en la instalación de Shinken ni tan siquiera se comprobaba la existencia de este módulo, pese a ser imprescindible para utilizar los widgets necesarios para el correcto funcionamiento del Dashboard.

Soluciones a los problemas surgidos:

- Solución para “a)”: Tras investigar cuales son cada uno de los paquetes de software necesarios para el correcto funcionamiento de Shinken, he creado un script que instala, de una manera completamente automática para el usuario, dichos paquetes. Los paquetes que se instalan en dicho script son:
 - o *python-paramiko*.
 - o *python-kombu*.
 - o *python-mysqldb*.
 - o *python-simplejson*.
 - o *python-netifaces*.
 - o *gawk*.
 - o *syslog-ng-mod-mongodb*.
- Solución para “b)”: Con el fin de solucionar el error de direccionamiento del botón “Show more” al seleccionar “1 Day” en la sección de graficas de cualquier dispositivo, se sustituye la plantilla “eldetail.tpl”, perteneciente a la interfaz gráfica de Shinken WebUI, por una versión corregida del mismo. Para modificar dicha plantilla hubo que revisar sus 764 líneas para comprender y solucionar los errores de la parte encargada de mostrar información de programas externos como PNP4Nagios.
- Solución para “c)”: Para solucionar el error de direccionamiento a la documentación de los botones de la interfaz WebUI hubo que corregir la plantilla “header_element.tpl” perteneciente a la código de dicha interfaz web. Y para ello hubo que comprender la estructura de la programación de 131 líneas del archivo y crear un pequeño script que sustituyese la versión original con errores por la corregida.
- Solución para “d)”: El error de la sección Dashboard se solucionó mediante la instalación del paquete:
 - *python-pymongo*.
- Para llegar a la certeza de que el archivo que faltaba de instalar era el mencionado, hubo que revisar los logs de inicio de Shinken.

En vista de que ya se disponía de un script que instalaba los paquetes que causaban el error mencionado en el apartado “a)”, se aprovechó para adjuntar la instalación de este archivo en dicho script y, de ese modo, ir automatizando poco a poco las soluciones a los errores que iban surgiendo.

- **PNP4Nagios**

En la instalación e inicialización de PNP4Nagios también surgen ciertos problemas:

- a. Nada más instalar PNP4Nagios no funcionaba. Esto es debido a que los desarrolladores no indicaban en ningún momento la importancia de tener el módulo de re-escritura de Apache instalado y activado para el funcionamiento del programa. Cabe señalar que siendo un módulo vital para el funcionamiento de PNP4Nagios, tampoco se activaba durante su instalación.
- b. Al hacer filtrados de información y tras diferentes pruebas se observó que las métricas internas utilizadas por pnp4nagios para realizar los cálculos de los intervalos de tiempo no obedecían a ninguna lógica. así se detectó que para realizar los cálculos diarios consideraba que el día tiene 25 horas; sin embargo, cuando los intervalos a tratar eran semanas, meses y años asumía que un día tiene 24 horas. además, interpretaba que un mes tiene 32 días y un año 380.
- c. El botón de la interfaz gráfica de PNP4Nagios para filtrar los datos de 1 día, indicaba 25 horas y no 24 horas como debía.

Soluciones a los problemas surgidos:

- Solución para “a)”: Para solucionar el problema descrito en “a)”, se ha creado un script que instala y activa automáticamente el módulo “allowoverride” de Apache, necesario para el correcto funcionamiento de la aplicación. Así como reinicia Apache para que los cambios surjan efecto.
- Solución para “b)”: El problema de las métricas descrito en “b)”. se ha solucionado creando un script en el que se sustituyen los archivos de pnp4nagios que contienen las métricas usadas por defecto (“config.php” y “config_local.php”), por versiones nuevas de los mismos con métricas adecuadas a la realidad (días de 24 horas, meses de 31 días y años de 365 días).

Para poder sustituir los parámetros necesarios, ha sido necesario estudiar la estructura que compone la interfaz web de PNP4Nagios y de ese modo ser poder ubicar los archivos en los que debían realizarse los cambios [8].

- Solución para “c)”: En los mismos archivos “config.php” y “config_local.php” de la solución “b)”, se modifica el botón por defecto de las 25 horas para que corresponda a la opción de 24 horas que debiera ser.

Interconexión

Tras la inicialización de ambas herramientas, se procedió a conectar Shinken con PNP4Nagios. para ello. En primer lugar fue necesario activar el módulo LiveStatus de Shinken. Este módulo es la interfaz responsable de la interacción entre Shinken y otros programas externos.

La interacción de Shinken con el servidor de correo del sistema se realiza a través del comando "mail".

Monitorización

Una vez interconectados Shinken, PN4Nagios y el servicio de correo electrónico, se procedió a realizar las pruebas de monitorización sobre la maqueta de red proporcionada por i2Basque.

En las pruebas de monitorización con el primer prototipo se detectaron algunos problemas:

- a. Shinken no ofrece soporte para el hardware de Juniper.
- b. La monitorización de las interfaces de red de cualquiera de los equipos siempre indicaba un fallo crítico. Pero, sin embargo, mostraba que el estado de cada una de ellas era correcto. Esta información resultaba contradictoria y tras indagar en el código del plugin encargado de la monitorización de este aspecto de los equipos, se detectó el bug que causaba esta inconsistencia: una condición que se evaluaba a true cuando no se estaban usando todas las interfaces de red del hardware monitorizado y que provocaba que, acto seguido, se activase el aviso de fallo crítico [1].
- c. El uso de las interfaces de red de los equipos no se monitorizan adecuadamente, lo que provoca que la información resultante sea incorrecta. Debido a que esa información era incorrecta, no es posible crear las gráficas correspondientes [2].

Soluciones a los errores surgidos en el primer prototipo

- Solución para "a)": Para añadir a Shinken la capacidad de monitorizar el hardware de Juniper, y solucionar el problema "a)", fue necesario crear un pack específico para monitorizar hardware de esta marca. Para construir este pack se tomó como base el oficial de Cisco. Después se creó un script para incluir dicho pack en el conjunto de packs disponibles en Shinken que se encuentran en la carpeta "/usr/local/shinken/etc/packs/". También se modificó uno de los scripts de monitorización escrito en Perl "check_nwc_health" para que admitiese equipos Juniper. Una vez modificado dicho script de monitorización se creó otro script para sustituir el original por el modificado en la carpeta de Shinken "/usr/local/shinken/libexec/".

- Solución para “b”): Se solucionó programando una nueva versión del plugin causante del error y construyendo un script, que sustituyera dicho plugin por la versión corregida [2].
- Como se trata del mismo plugin del problema descrito en “a)”, se tuvo que modificar más a fondo. Llegados a este punto se decidió descargar el código fuente del plugin disponible en GitHub, modificarlo y compilarlo.
- Para sustituir el original por el modificado se siguió usando el mismo script que se creó en la solución “a)”.
- Solución para “c)”: Para solucionar el error del uso de las conexiones de red, se sustituye el pack de monitorización de sistemas Linux por un pack del mismo modificado, cuyo chequeo de las conexiones ha sido cambiado para que recaiga sobre el plugin de monitorización compilado en la solución “b)”, que lo realiza de manera adecuada [3].
- Debido a que el instalador de Shinken copia tal cual la carpeta de packs de la carpeta de instalación hasta la ruta del sistema operativo donde se instala la aplicación en sí, se ha sustituido el pack de sistemas Linux original por el modificado en la misma carpeta comprimida de la instalación.

Los archivos instalados y/o modificados para el correcto funcionamiento del primer prototipo han sido:

- python-pymongo (extensión de Python que posibilita el acceso a bases de datos mongo desde programas escritos en Python)
- python-paramiko (se trata de una librería que permite que los scripts de Python se comuniquen mediante el protocolo SSH2 en caso de ser necesario)
- syslog-ng-mod-mongodb (es un plugin de MongoDB. Se trata de una mejora del demonio de gestión del login del sistema)
- python-kombu (una interfaz de mensajería interna, es decir, una interfaz que permite enviar mensajes entre distintos procesos a Python)
- python-mysqldb (se trata de un driver de MySQL escrito en Python)
- python-simplejson (un codificador/decodificador de JSON para Python)
- python-netifaces (librería que permite a los programadores de Python acceder a las interfaces de red de la maquina local, y obtener las direcciones de estas)
- gawk (es una re-implementacion de awk para GNU)
- rsync
- check_nwc_health (plugin de monitorización de Shinken)

- commands.cfg (archivo de comandos a los que accede el pack de Linux para la monitorización de dichos sistemas)
- eltdetail.tpl (Shinken)
- graph_content.php (PNP4Nagios)
- config.php (PNP4Nagios)
- config_local.php (PNP4Nagios)
- default (Apache)

Los scripts creados para solucionar los diversos fallos detectados durante el desarrollo del prototipo 1, se han agrupado en un solo script, cuya ejecución instala el prototipo 1 del programa de monitorización objetivo de este trabajo.

A continuación se muestran las pruebas realizadas al primer prototipo:

- Inicio de Shinken: Tras solucionar los problemas del instalador de Shinken, se comprueba el correcto inicio de todos los módulos que lo componen (véase *Ilustración 6-03*).

```

aaron@pfc-server-03: ~
root@pfc-server-03:/home/aaron# service shinken start
Starting scheduler:
. ok
Starting poller:
. ok
Starting reactionner:
. ok
Starting broker:
. ok
Starting receiver:
. ok
Starting arbiter:
. ok
root@pfc-server-03:/home/aaron# █

```

Ilustración 6-03: Inicialización de Shinken

- Inicio de PNP4Nagios: Una vez solucionado el problema con el módulo de apache, se comprueba el correcto inicio del programa encargado de la realización de las gráficas PNP4Nagios (véase *Ilustración 6-04*).

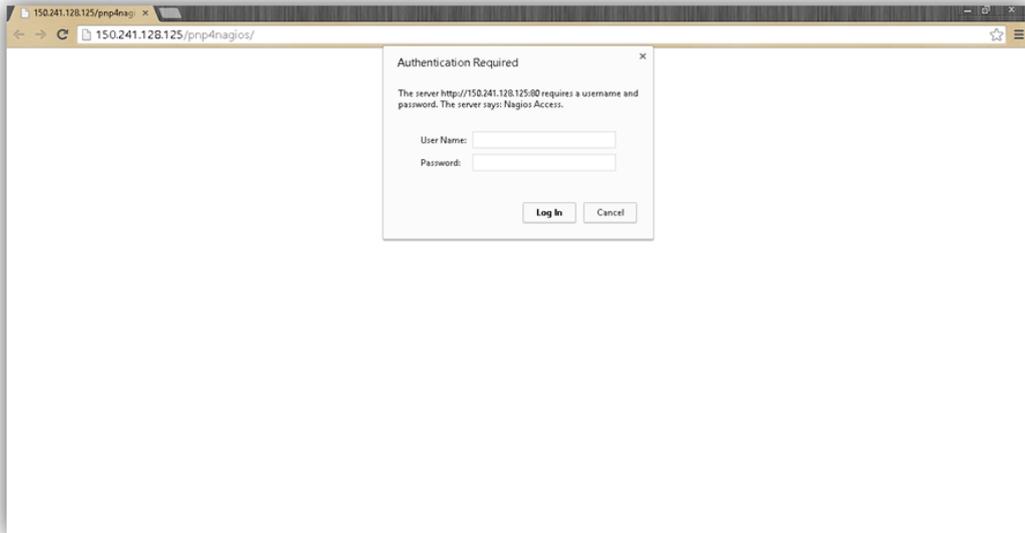


Ilustración 6-04: Inicialización de PNP4Nagios

- Añadir equipos: Comprobación de que Shinken no da ningún error tras añadir varios equipos (véase *Ilustración 6-05*) al reiniciarse (véase *Ilustración 6-06*).

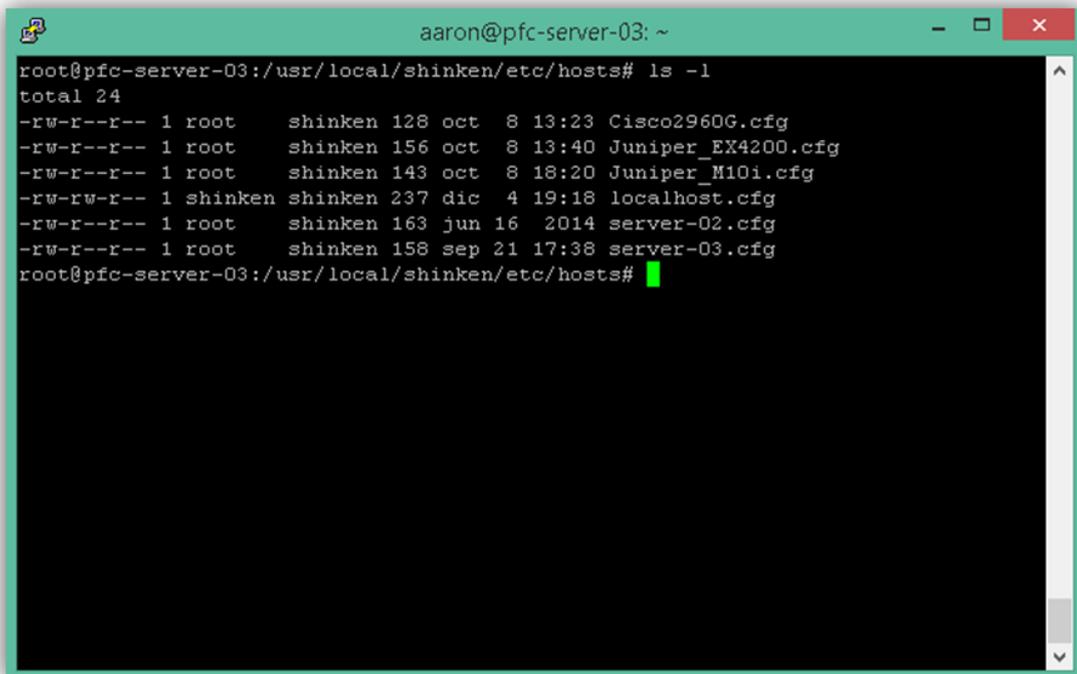


Ilustración 6-05: Varios equipos añadidos a Shinken para ser monitorizados

```

aaron@pfc-server-03: ~
root@pfc-server-03:/usr/local/shinken/etc/hosts# service shinken restart
Restarting scheduler
. ok
Restarting poller
. ok
Restarting reactionner
. ok
Restarting broker
. ok
Restarting receiver
. ok
Restarting arbiter
Doing config check
. ok
. ok
root@pfc-server-03:/usr/local/shinken/etc/hosts#

```

Ilustración 6-06: Reinicio de Shinken tras a adición de equipos para su monitorización

- Monitorización de los equipos añadidos: Tras modificar el pack de Linux para solventar el problema con la monitorización de las interfaces de red, crear un pack Juniper nuevo para ofrecer soporte a dicha marca y compilar uno de los plugins de monitorización después de modificar su código, se comprobó que todos los equipos eran correctamente monitorizados por Shinken (véase *Ilustración 6-07*).

The screenshot shows the i2BasqueNMS monitoring interface. At the top, there are navigation tabs for Dashboard, Impacts, IT problems, All, Wall, and System. A search bar and user profile (Hi Admin) are also visible. The main content area displays a table of monitored devices and their components. The overall business impact is reported as 'Normal'.

Device	Status	Component	Value
Cisco2960G	UP	PING	OK - Packet loss = 0%, RTA = 0.77 ms
		Cpu	OK - cpu 1 usage (5 min avg.) is 6.00%
		Hardware	OK - environmental hardware working fine
		InterfaceErrors	OK - interface Vlan1 errors in 0.00/s out 0.00/s, interface GigabitEthernet0/1 errors in 0.00/s out 0.00/s
		InterfaceUsage	OK - interface Vlan1 usage is in 0.00% (6425.93Bts/s) out 0.00% (8955.26Bts/s), interface GigabitEthernet0/2 is down/up, GigabitEthernet0/1 is up/down
DELL_PowerEd...	UP	PING	OK - Packet loss = 0%, RTA = 0.39 ms
		Cpu	OK - 4 CPU, average load 8.0% < 80% - OK
		Disks	OK - (<90%) All selected storages
		Load	OK - Load: 0.05 0.12 0.09 - OK
		Memory	OK - mempool Processor usage is 22.63%, mempool Driver text usage is 0.00%, mempool I/O usage is 39
Juniper-EX4200	UP	PING	OK - Packet loss = 0%, RTA = 4.03 ms
		Cpu	OK
		Hardware	OK
		InterfaceErrors	OK - interface lo usage is in 0.00% (0.00Bts/s) out 0.00% (0.00Bts/s), interface eth0 usage is in 0.00/s out 0.00/s, interface dsc errors in 0.00/s out 0.00/s, interface

Ilustración 6-07: Equipos de las marcas Dell, Cisco y Juniper siendo monitorizados correctamente por Shinken

- Correcto funcionamiento del Dashboard: Tras incluir el paquete de mongoDB necesario para el correcto funcionamiento de la sección Dashboard de Shinken, se comprobó que éste funciona sin ningún problema (véase Ilustración 6-08).

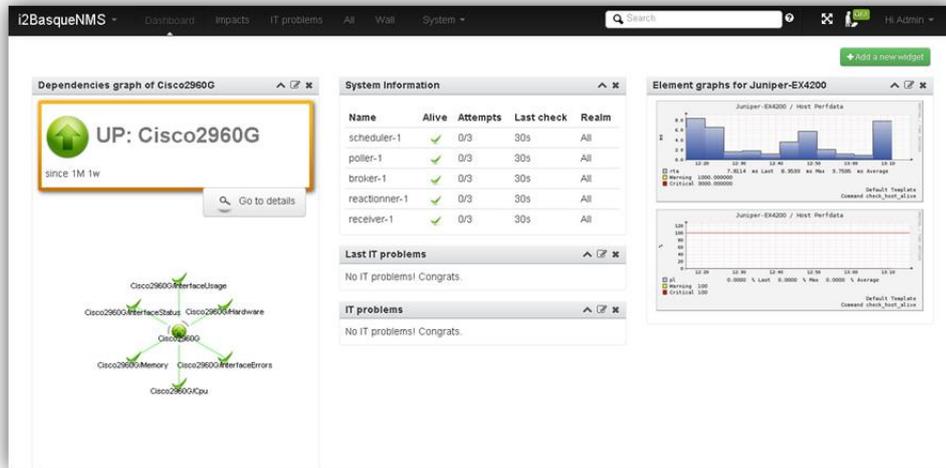


Ilustración 6-08: Sección Dashboard de Shinken

- Realización de gráficas mediante PNP4Nagios usando los datos recabados por Shinken: Tras la activación de la interfaz LiveStatus para interconectar Shinken con los programas externos, se comprobó que PNP4Nagios crea gráficas con los datos recabados por Shinken en la monitorización de la red (véase Ilustración 6-09).

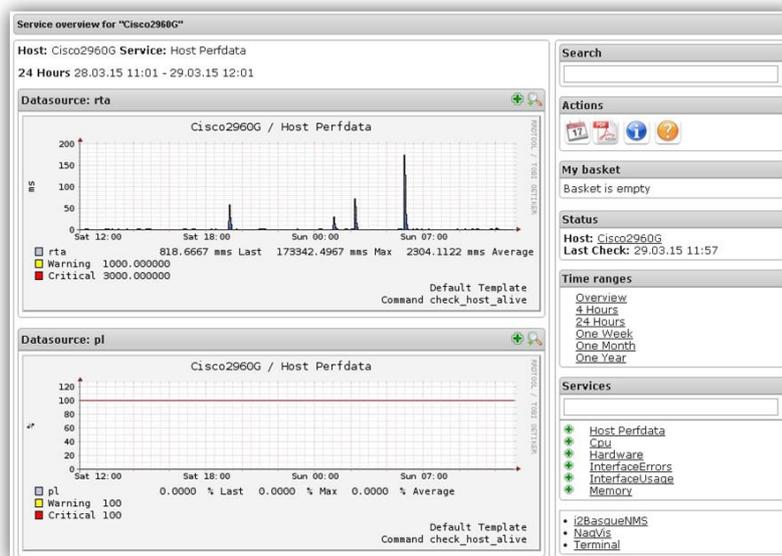


Ilustración 6-09: Realización de gráficas en PNP4Nagios con los datos recabados de la red por Shinken

- Comprobación del correcto funcionamiento del cuadro de búsqueda indexada de PNP4Nagios: Se comprobó que tras la activación del módulo de apache y de la interfaz LiveStatus, el cuadro de búsqueda de equipos de PNP4Nagios funciona correctamente (véase Ilustración 6-10).

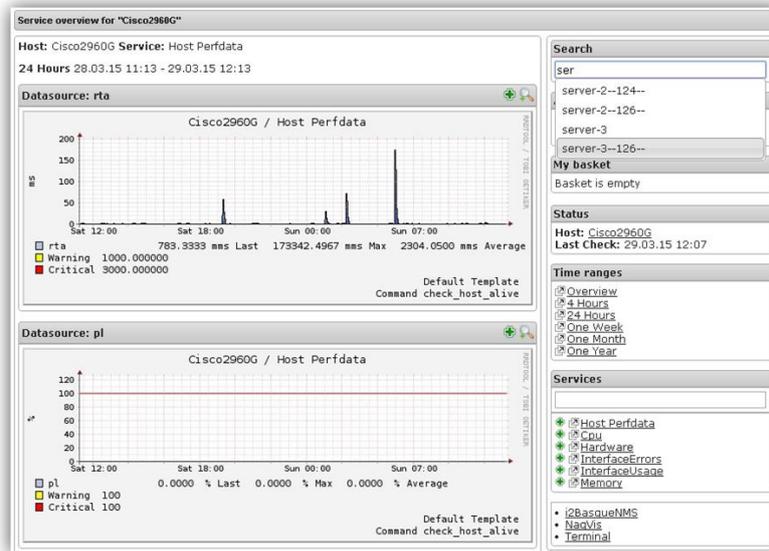


Ilustración 6-10: Cuadro de búsqueda de equipos en PNP4Nagios

- Prueba de los filtros añadidos en la pestaña ALL: Se comprobó que en Shinken funciona correctamente el filtrado de grupos de equipos por nombre, tipos (servidores, routers, etc), servicios (ftp, http, etc) (véase Ilustración 6-11).

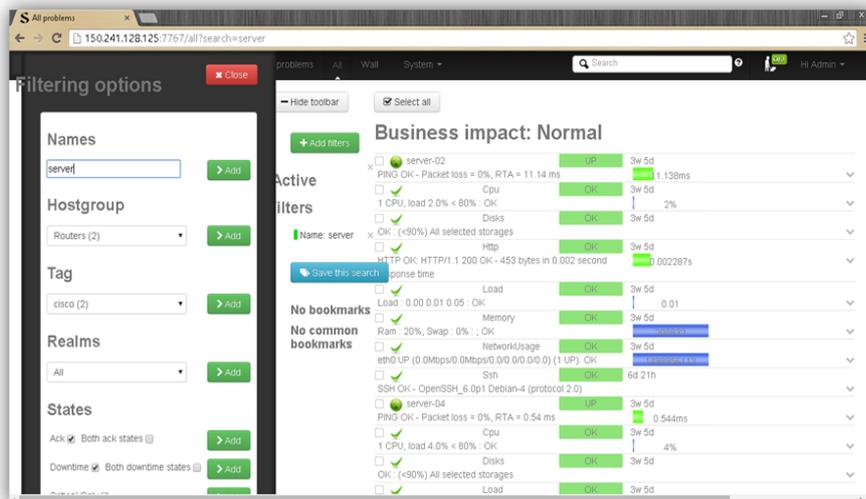


Ilustración 6-11: Filtros para la búsqueda de equipos en Shinken

- Prueba del funcionamiento de la alarma vía email: Se comprobó, provocando un error, que Shinken avisa de dicho fallo enviando un email al administrador de la red (véase Ilustración 6-12).

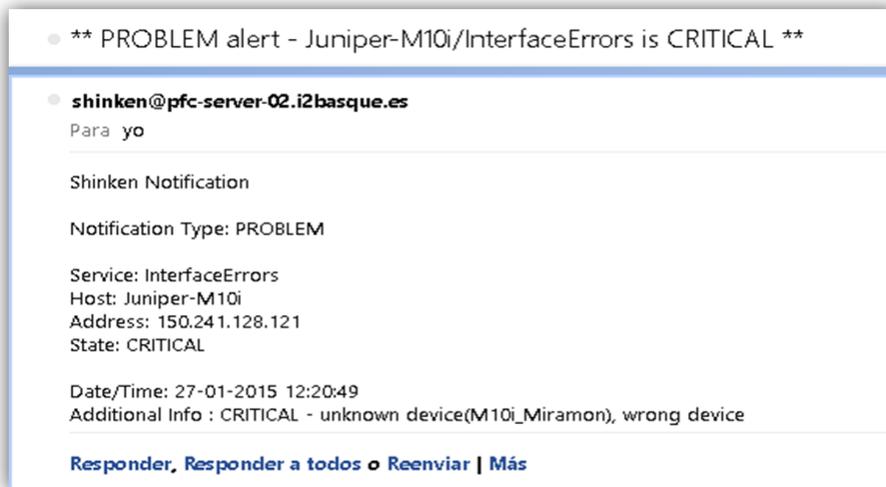


Ilustración 6-12: Alarma por email indicando errores de conexión en las interfaces de red del router Juniper-M10i

- Prueba del buscador indexado de equipos: Comprobación de que el buscador de Shinken permite buscar equipos concretos correctamente (véase Ilustración 6-13).

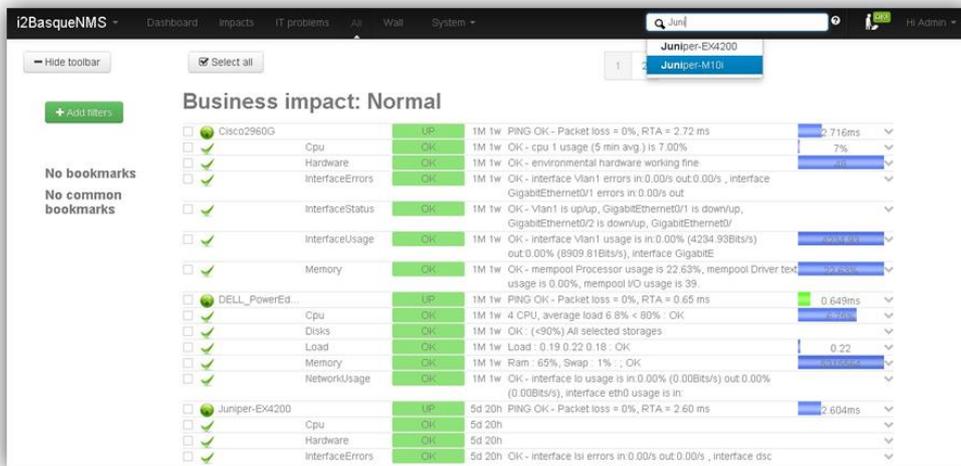


Ilustración 6-13: Búsqueda indexada de equipos en Shinken

- Prueba del funcionamiento de la creación de PDFs con las gráficas de los equipos monitorizados: Comprobación de que es posible crear documentos PDF a partir de las gráficas generadas por PNP4Nagios (véase Ilustración 6-14).

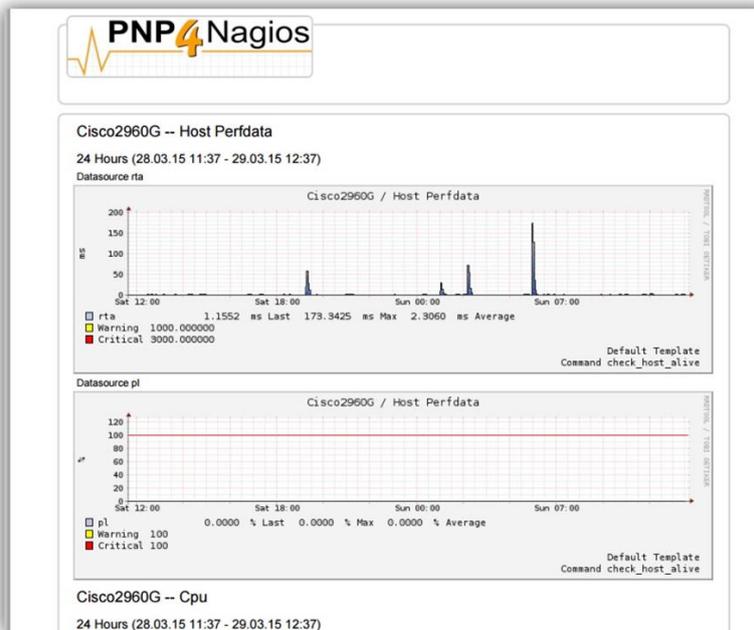


Ilustración 6-14: Documento PDF creado a partir de las gráficas de PNP4Nagios

- Prueba del funcionamiento de los filtros de tiempo en PNP4Nagios: Comprobación de que PNP4Nagios permite al usuario filtrar los datos en base a periodos de tiempo deseados (véase Ilustración 6-15).

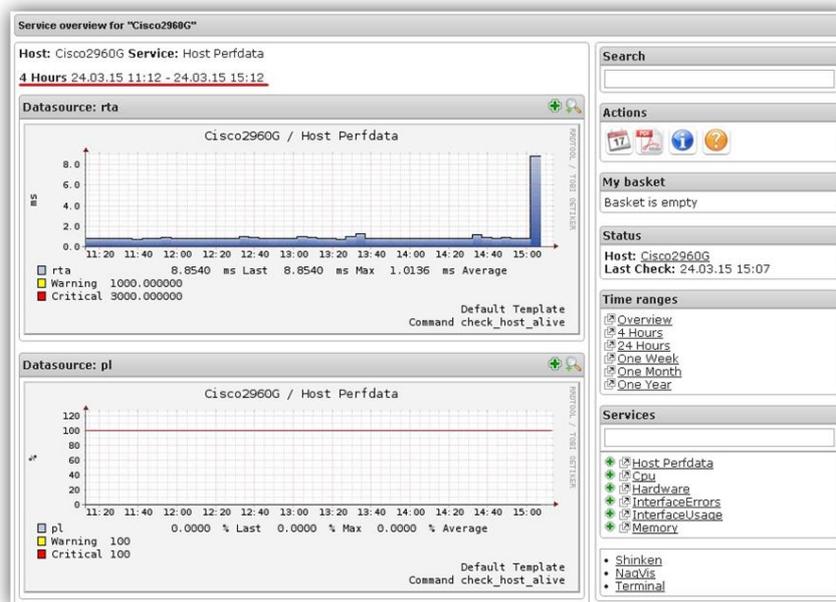


Ilustración 6-15: Interfaz web de PNP4Nagios pulsar el botón para que muestre los datos de las últimas 4 horas

- Prueba de las gráficas creadas por PNP4Nagios dentro de Shinken: Tras solucionar el problema de direccionamiento al pulsar "1 Day" y, a continuación, "Show more" para activar el módulo LiveStatus, se comprobó

que las gráficas creadas por PNP4Nagios se muestran adecuadamente en la interfaz de Shinken (véase *Ilustración 6-16*).



Ilustración 6-16: Gráficas creadas por PNP4Nagios en la interfaz de Shinken

6.2 Segundo prototipo

En el segundo prototipo, se amplió el primero con una nueva opción que permite al administrador añadir imágenes de la topología de la red, desde las que se puede visualizar el estado de los componentes en tiempo real.

Herramientas integradas

Se añadió el software NagVis al primer prototipo para incluir la funcionalidad de mapas en la herramienta de monitorización.

Para ello, antes de integrarla en el primer prototipo, fue necesario comprobar que, de forma independiente, NagVis se inicializaba y funcionaba correctamente.

Inicialización

- Nagvis:

En la ejecución de NagVis surgieron algunos problemas que se describen a continuación, junto con sus causas y la solución desarrollada para solventarlos.

- Se produjo un problema al intentar acceder a la configuración general de NagVis a través de la interfaz web:

“Error: (0) Array to string conversión”

Se decidió investigar por los foros antes de lanzarse a modificar directamente el código. Tras la investigación, resultó tratarse de un problema común en la última versión estable disponible para Debian 7

Wheezy. Y los mismos desarrolladores indicaban que lo solucionarían para la siguiente versión. Como se desconocía cuando estaría dicha versión en los repositorios de Debian 7 Wheezy, se decidió solucionarlo por cuenta propia.

Dicho problema se producía en el paso previo a mostrar en la interfaz web el contenido del fichero de configuración "*nagvis.ini.php*", cuando se convertía el contenido de un array con los datos de la configuración a strings.

- b. Se producía un error al guardar los cambios en la configuración general. Ya que aún solucionando el problema descrito en el apartado "a)", debido a que el fichero "*nagvis.ini.php*" no contaba con el permiso de escritura necesario, el administrador no podía modificarlo.
- c. Las figuras añadidas por el administrador a la aplicación fallaban por un error en la programación del software. Cuando se intentaban añadir en alguno de los mapas, estas se mostraban como una imagen que decía "Not Found".

Soluciones a los errores de NagVis

- Solución para "a)": Para solucionar el error se ha tenido que modificar el código del archivo "*WuiViewEditMainCfg.php*", encargado de la lectura, visualización y escritura de las posibles modificaciones de la configuración general de NagVis, almacenada en el fichero "*nagvis.ini.php*".

Para solucionar el error fue necesario revisar el software para determinar que archivos de la interfaz web de NagVis era el encargado de su configuración general. Tras comprender la estructura de programación seguida por los desarrolladores, resultó que el archivo a modificar era el "*WuiViewEditMainCfg.php*". Una vez localizado el archivo hubo que revisar sus 207 líneas de código para resolver el problema que existía al pasar la información de array a string de cara a mostrar los datos en la interfaz web. Una vez localizada la línea se procedió a copiar el archivo, se creó una modificación del mismo y finalmente se creó un script para que se sustituyera el original por el modificado en la instalación.

- Solución para "b)": Tras solucionar el problema del apartado "a)", se produjo el problema esperado con respecto a los permisos del archivo "*nagvis.ini.php*". Para solucionar el problema del apartado "b)" se ha creado un script que modifica automáticamente los permisos del archivo durante la propia instalación de i2BasqueNMS.
- Solución para "c)": Para solucionar el error del apartado "b)" se ha tenido que modificar el archivo, llamado "*NagVisShape.php*", encargado de

mostrar en la interfaz web las figuras subidas por el administrador. Una vez modificado el archivo, se creó un script para sustituir el archivo original por el modificado durante la instalación de i2BasqueNMS.

El problema lo causaban unas líneas de código correspondientes a la implementación de una nueva función, prevista para versiones posteriores, que a los desarrolladores se les había olvidado eliminar [9].

Interconexión

Shinken y NagVis se interconectan mediante el módulo LivesStatus que ya se había activado en el primer prototipo.

Monitorización

Una vez interconectados el primero prototipo y NagVis, se procedió a realizar las pruebas pertinentes de cara asegurar la total integración y el correcto funcionamiento de la nueva aplicación.

Al iniciar la monitorización con el nuevo software, no se detectó ningún problema cuando se incluyeron los equipos monitorizados por Shinken en los mapas de NagVis.

Por otro lado, tras añadir un equipo a cualquier mapa en NagVis y pulsar sobre el icono de éste, se producía el siguiente error:

- a. Re-direccionamientos erróneos desde NagVis hasta Shinken al pulsar sobre cualquier equipo, servicio o grupo de equipos visualizado en los mapas.

Esto era debido que NagVis no se diseñó para que funcionara junto a Shinken, y por lo tanto, el vínculo creado para cada icono de los mapas correspondían a direcciones de Nagios [3].

Solucion al error surgido en el segundo prototipo

- Solución para “a)”: Para solucionar el problema fue necesario localizar las líneas de configuración encargadas del re-direccionamiento en el archivo “*nagvis.ini.php*”. Una vez localizadas se tuvo que crear una versión modificada del archivo para que contuviese las direcciones “plantilla” junto con una variable IP, que representa la dirección IP de cada equipo representado en los mapas, de cada una de ellas. Después se creó un script para sustituir el archivo original por el modificado y un segundo script cuya ejecución capturaba la dirección IP del servidor y la sobrescribía sobre la variable IP de las plantillas.

Los ficheros modificados y los scripts creados para el correcto funcionamiento del segundo prototipo son:

- WuiViewEditMainCfg.php
- nagvis.ini.php
- NagVisShape.php
- Script para las direcciones IP de NagVis: NagVisIPs.sh

A continuación se muestran las pruebas realizadas al segundo prototipo:

- Inicio de Nagvis: Comprobación del correcto inicio del software de mapas Nagvis (véase *Ilustración 6-17*).

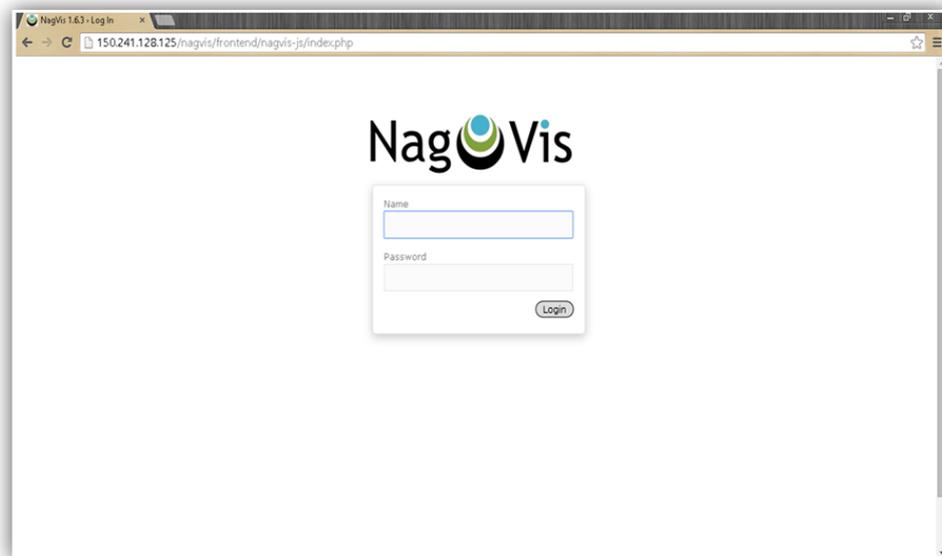


Ilustración 6-17: Login de NagVis

- Correcto funcionamiento de NagVis mostrando en los mapas geográficos los datos recabados por Shinken en tiempo real: Comprobación de que NagVis muestra los datos de los componentes de red monitorizados por Shinken a tiempo real (véase *Ilustración 6-18*).

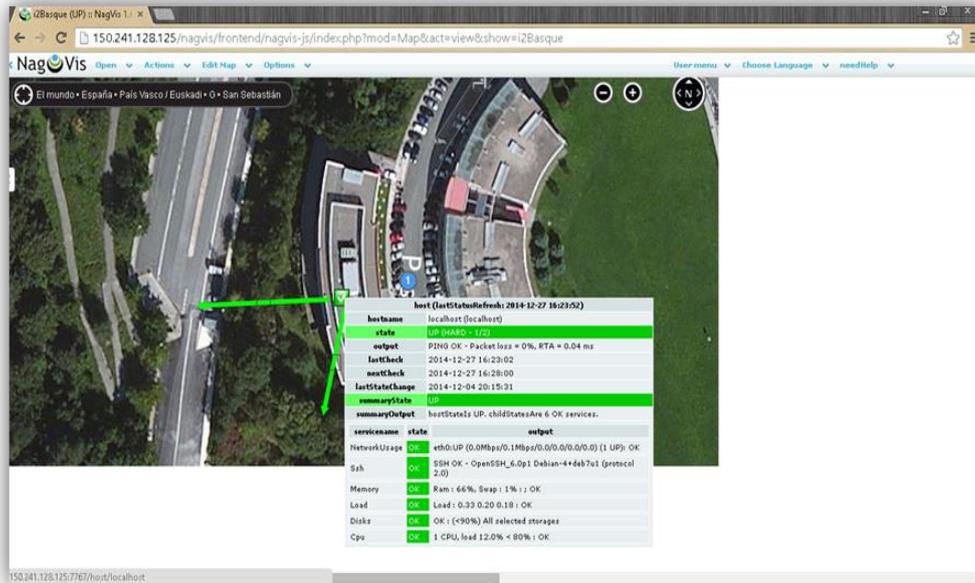


Ilustración 6-18: Fotografía aérea de i2Basque con los datos de uno de los servidores monitorizados

- Correcto direccionamiento desde NagVis hasta Shinken: Comprobación del correcto direccionamiento desde NagVis (véase Ilustración 6-19) hasta Shinken (véase Ilustración 6-20) al pulsar sobre cualquiera de los equipos de red mostrados en los mapas.



Ilustración 6-19: NagVis mostrando, sobre una foto aérea de la facultad de Vitoria, los datos de uno de los equipos monitorizados

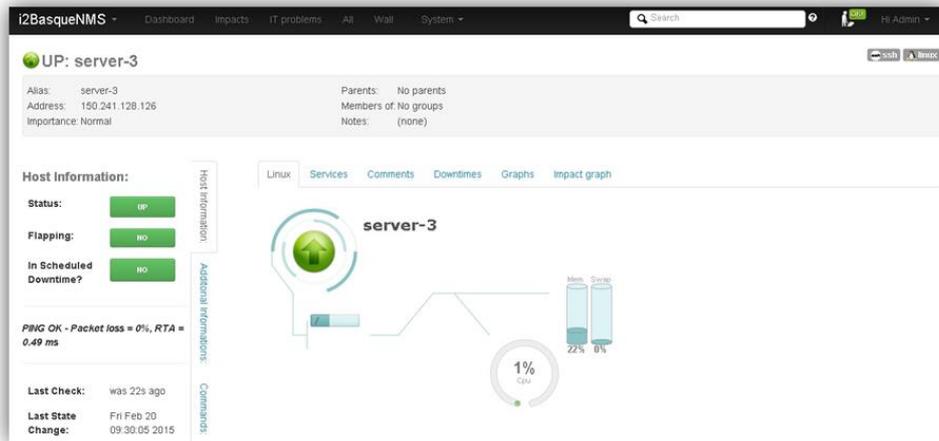


Ilustración 6-20: Sección de datos respecto a la máquina seleccionada en NagVis

- Prueba de adición de una imagen de fondo: Comprobación de la correcta adición de imágenes de mapas geográficos o equipos de red en NagVis (véase Ilustraciones 6-21 y 6-22)

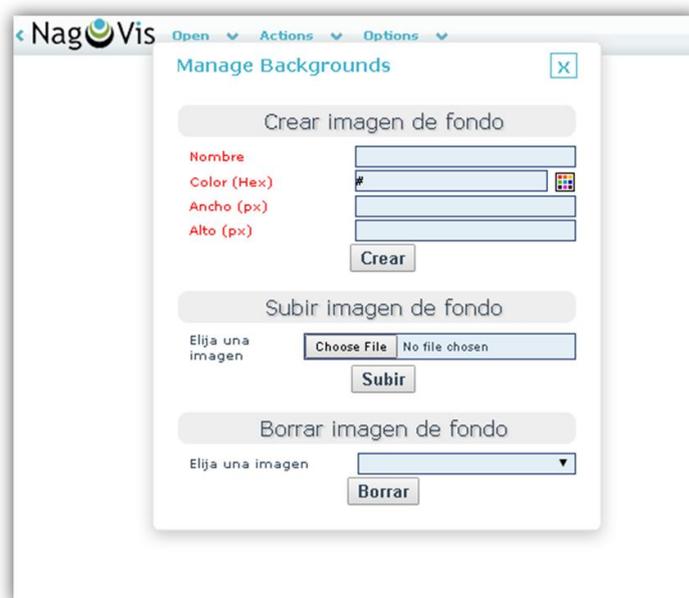


Ilustración 6-21: Interfaz web con las opciones para administrar las imágenes de los fondos de los mapas

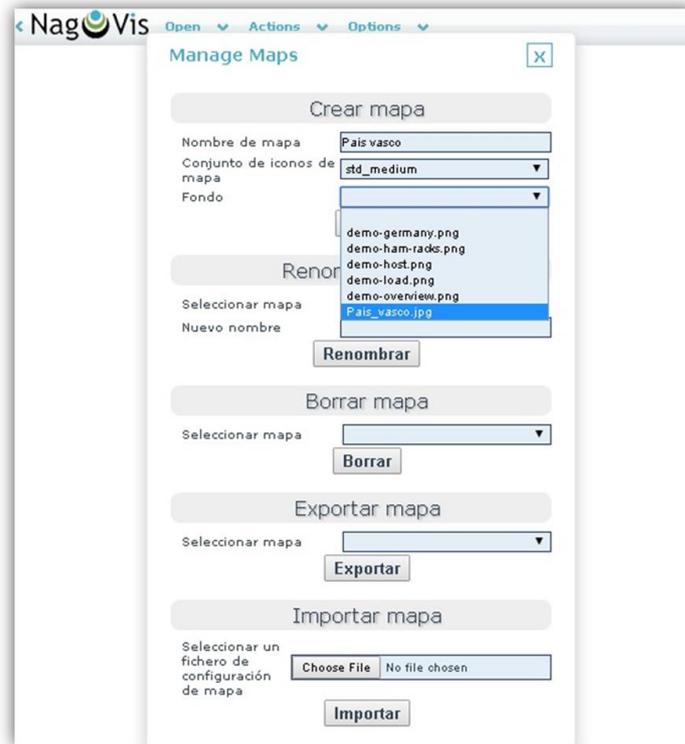


Ilustración 6-22: Interfaz web con las opciones para administrar los mapas, donde se visualiza la imagen de fondo subida por el administrador

- Prueba de adición de equipos nuevos a los mapas: Tras comprobar la correcta adición de los fondos para mapas, se comprobó que se pueden crear correctamente mapas nuevos con dichos fondos (véase *Ilustración 6-23*).

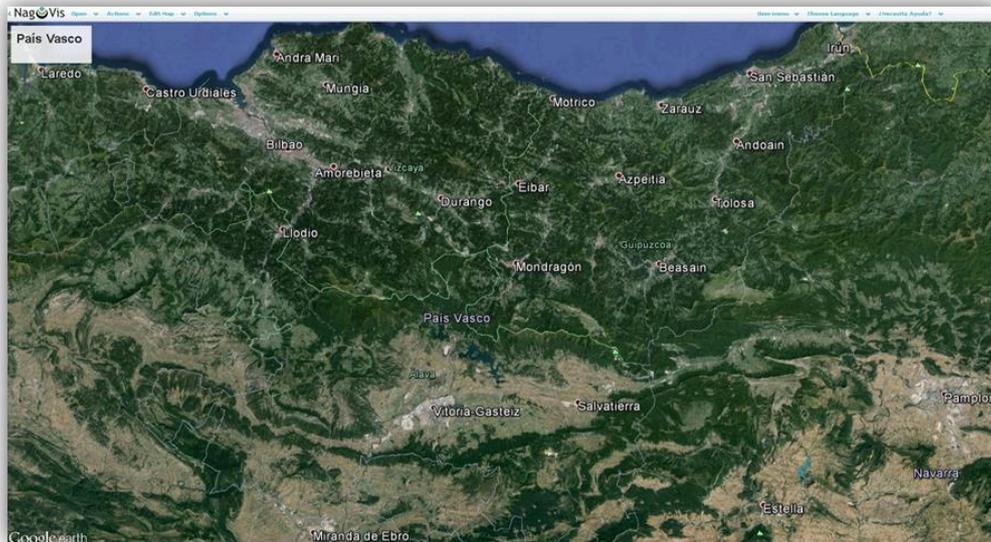


Ilustración 6-23: Mapa creado en NagVis a partir de uno de los fondos añadidos por el administrador

- Prueba de transición de un mapa a otro: Comprobación de la posible transición de un mapa a otro por parte de los usuarios (véase *Ilustraciones 6-24 y 6-25*).

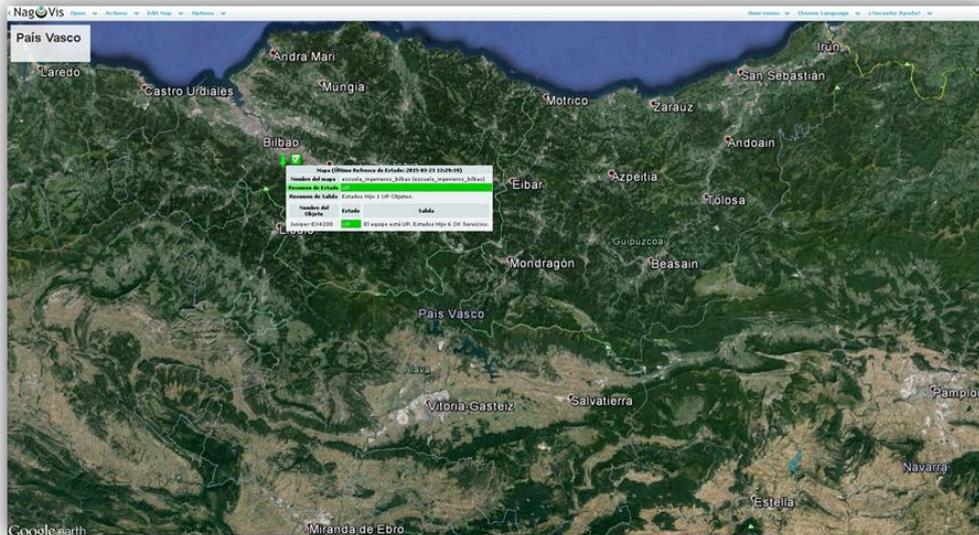


Ilustración 6-24: Mapa de NagVis del País Vasco con un enlace a otro mapa de Bilbao

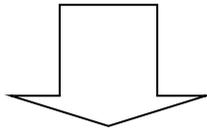


Ilustración 6-25: Mapa de Nagvis con una vista por satélite de la escuela de ingenieros de Bilbao con uno de sus equipos siendo monitorizado

- Prueba de que la eliminación de equipos desde Shinken no causa problemas en NagVis: Comprobación de la indicación en NagVis cuando se ha eliminado un equipo en Shinken (véase Ilustración 6-26).



Ilustración 6-26: Mapa de NagVis de la escuela de ingenieros de Bilbao con una marca azul indicando que el equipo ya no está siendo monitorizado por i2BasqueNMS

6.3 Tercer prototipo

el tercer y último prototipo debía ser capaz de, además de todas las funcionalidades contempladas por el prototipo anterior, permitir al administrador conectarse vía terminal desde la propia aplicación web al servidor donde se encuentre instalado i2basquenms. además, tras conectarse vía remota al servidor, el administrador debía ser capaz de realizar un escaneo de la red monitorizada para descubrir nuevos equipos o servicios que se encuentren disponibles en esta.

el administrador debían poder navegar entre las aplicaciones que forman i2basquenms gracias a la modificación de las interfaces gráficas de cada uno de los programas. dichas modificaciones consistieron en añadir, a cada aplicación integrada, un menú que permitiera dirigirse a cualquiera de las aplicaciones en cualquier momento.

Herramientas integradas

En el tercer prototipo se incluyeron Nmap, Shellinabox, los cambios en WebUI y en las interfaces gráficas tanto de PNP4nagios y NagVis, al segundo prototipo

Inicialización

- **Nmap**

La herramienta usada para el escaneo de equipos y servicios en la red, Nmap, se instaló y se inició sin problemas.

- **Shellinabox (Terminal remota)**

En la instalación de la terminal remota Shellinabox, que sirve para que el administrador de la red se conecte directamente con el servidor, surgieron algunas dificultades causadas por los certificados. Aparte de esto, se inició sin problemas.

- **Interfaces web modificadas**

La modificación de las interfaces web para navegar entre las aplicaciones integradas en i2BasqueNMS resultó bastante costosa, requiriendo llegar a entender cómo los desarrolladores habían diseñado y programado cada uno de los programas a modificar. No obstante, tras repetidos intentos y correcciones, se consiguió que funcionara adecuadamente. Una vez modificadas las interfaces gráficas de cada uno de los programas, se inician sin problemas y las modificaciones efectuadas se visualizan correctamente.

Para que las modificaciones en las interfaces web fueran persistentes, se crearon copias modificadas de los archivos implicados y un nuevo script para sustituir los originales por los modificados para la instalación de i2BasqueNMS.

Interconexión

Las modificaciones realizadas en las interfaces gráficas de las aplicaciones posibilitan la interconexión entre las aplicaciones. Dichas modificaciones añaden un menú que direcciona al usuario a la aplicación que elija.

Monitorización

Para realizar el descubrimiento de equipos y servicios en la red hay que usar Nmap desde el servidor donde está instalado. El administrador debe valerse de la Terminal web remota, a la que se puede acceder desde el menú añadido en la modificación de las interfaces web.

Se comprobó que en la monitorización de la red con Nmap los equipos y servicios nuevos aparecen correctamente especificados.

Gracias a los datos obtenidos por Nmap, el administrador puede añadir equipos nuevos en Shinken, o servicios nuevos en los equipos monitorizados.

Los archivos modificados para el correcto funcionamiento del tercer prototipo han sido:

- logo_box.php (PNP4Nagios)
- default.header.html (NagVis)
- header_element.tpl (Shinken)

A continuación se muestran las pruebas realizadas a este prototipo:

- Prueba de adición del menú en la interfaz WebUI de Shinken: Comprobación la correcta modificación de las interfaces WebUI, PNP4Nagios y NagVis (véase Ilustraciones 6-27, 6-28 y 6-29).

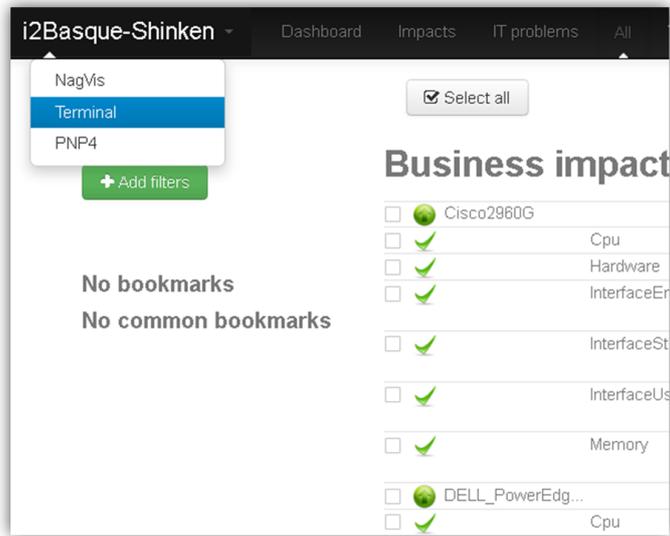


Ilustración 6-27: Interfaz web de Shinken con el menú desplegable añadido



Ilustración 6-28: Interfaz web de NagVis tras la adición de los links al resto de programas de i2BasqueNMS



Ilustración 6-29: Interfaz web de PNP4Nagios tras la adición de los links al resto de programas de i2BasqueNMS

- Correcto funcionamiento de la terminal web Shellinabox con conexión cifrada: Comprobación del correcto funcionamiento de la terminal web con conexión cifrada; permitiendo al administrador conectarse al servidor en el que se aloja i2BasqueNMS (véase Ilustraciones 6-30 y 6-31).

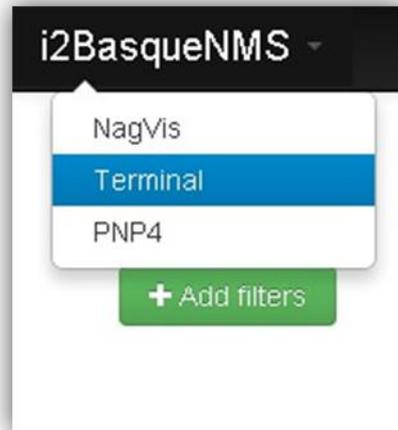


Ilustración 6-30: Interfaz web de Shinken con la opción *Terminal* seleccionada

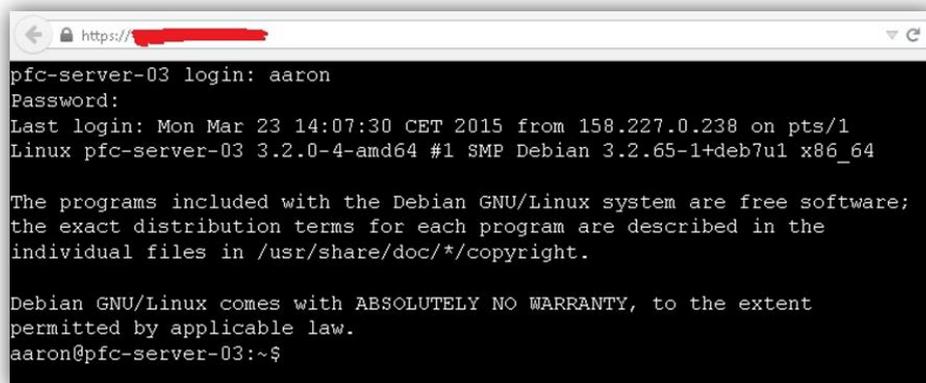


Ilustración 6-31: Terminal web dando acceso directo al administrador al servidor donde se encuentra instalado el sistema de monitorización

- Prueba de escaneo de la maqueta proporcionada por i2Basque, mediante Nmap a través de la terminal web: Comprobación de que, conectándose al servidor a través de la terminal web, es posible usar el software Nmap para el escaneo de la red administrada en busca de nuevos equipos y/o servicios (véase Ilustraciones 6-32 y 6-33)

```

pfc-server-03 login: aaron
Password:
Last login: Mon Mar 23 09:31:52 CET 2015 from 158.227.0.238 on pts/0
Linux pfc-server-03 3.2.0-4-amd64 #1 SMP Debian 3.2.65-1+deb7u1 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
aaron@pfc-server-03:~$ su
Contraseña:
root@pfc-server-03:/home/aaron# nmap -O 150.241.128.121

```

Ilustración 6-32: Conexión remota con el servidor a través de la Terminal añadida a i2BasqueNMS. Inserción del comando para ejecutar el escaneo de la red

```

Starting Nmap 6.00 ( http://nmap.org ) at 2015-03-23 09:56 CET
Nmap scan report for 150.241.128.121
Host is up (0.00058s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    filtered telnet
MAC Address: 00:05:85:08:08:5D (Juniper Networks)
Device type: router
Running: Juniper JUNOS 7.X|8.X
OS CPE: cpe:/o:juniper:junos:7 cpe:/o:juniper:junos:8
OS details: Juniper Networks M10 or M320 router (JUNOS 7.4R3.4), Juniper Networks M series router (JUNOS 8.3R2.0)
Network Distance: 1 hop

Nmap scan report for 150.241.128.122
Host is up (0.00075s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    filtered telnet
MAC Address: 80:71:1F:CF:ED:40 (Juniper Networks)
Device type: router|switch
Running: Juniper JUNOS 8.X|9.X|10.X
OS CPE: cpe:/o:juniper:junos:8 cpe:/o:juniper:junos:9 cpe:/o:juniper:junos:10
OS details: Juniper Networks J2320 router; or EX2200, EX3200, EX4200, or EX8200 switch (JUNOS 8.5 - 10.0)
Network Distance: 1 hop

Nmap scan report for 150.241.128.123
Host is up (0.00017s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
111/tcp   open  rpcbind
3128/tcp  open  squid-http
6000/tcp  open  X11
6001/tcp  open  X11:1
6002/tcp  open  X11:2
MAC Address: 00:15:C5:E8:E0:FF (Dell)
No exact OS matches for host (If you know what OS is running on it, see http://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=6.001E=4D=3/231OT=221CT=11CU=390801FV=N1DS=11DC=D1G=Y4M=0015C51T
OS:M=50FD56D1P=x86_64-unknown-linux-gnu)SEQ(SP=1061GCD=11ISR=1091TI=21CI=2
OS:11I=11TS=A)OPS(O1=M5B4ST11NW71O2=M5B4ST11NW71O3=M5B4NN11NW71O4=M5B4ST11
OS:NW71O5=M5B4ST11NW71O6=M5B4ST11)WIN(W1=38901W2=38901W3=38901W4=38901W5=38
OS:901W6=3890)ECN(R=Y1DF=Y1T=411W=39081O=M5B4NN11NW71CC=Y1Q=)T1(R=Y1DF=Y1T=4
OS:11S=01S=21F=R31RD=01Q)T2(R=M)T3(R=M)T4(R=Y1DF=Y1T=411W=01S=21F=R1O
OS:=1RD=01Q)T5(R=Y1DF=Y1T=411W=01S=21A=S+1F=AR1O=1RD=01Q)T6(R=Y1DF=Y1T=41
OS:1W=01S=21A=S+1F=R1O=1RD=01Q)T7(R=Y1DF=Y1T=411W=01S=21A=S+1F=AR1O=1RD=01Q
OS:)=U1(R=Y1DF=N1T=411IPL=1641UN=01R1PL=G1RID=G1RIPCK=G1RUCK=G1RUD=G)IE(R=Y
OS:1DF=N1T=411CD=8)

```

Ilustración 6-33: Escaneo de la red monitorizada utilizando Nmap a través de la conexión remota

6.4 Desarrollo Extra

Ampliación del diseño de i2BasqueNMS para Raspberry Pi y la posibilidad de realizar monitorización distribuida a bajo coste

Introducción

En previsión de un crecimiento de la red de i2Basque o como sistema de monitorización para redes locales pequeñas o medianas, se decidió modificar el diseño de i2BasqueNMS inicial para adaptarlo al limitado hardware de unos sistemas de bajo coste llamados Raspberry Pi y Raspberry Pi 2 [5]. Cabe destacar que la modificación funciona tanto en la versión de 2012 de estos sistemas, como en la más reciente de 2015. Sistemas cuyo precio oscila entre los 29 y 35 euros y cuyas características son (véase *Tabla 6-1*):

Características / Modelo	Raspberry Pi (2012)(modelo B)	Raspberry Pi (2015)(modelo B+)
CPU	700 MHz Single-Core ARM1176JZF-S	Broadcom ARM Cortex-A7 (SoC BCM2836) Quad-Core 900 MHz
RAM	512 MiB (compartidos con la GPU)	1 GB (compartidos con la GPU)
Almacenamiento	SD / MMC / ranura para SDIO	MicroSD
Conectividad por Red	10/100Ethernet (RJ-45)	10/100Ethernet (RJ-45)
Conectividad USB	2 x USB2.0	4 x USB2.0
Consumo energético	700mA, (3.5 W) / 5V vía Micro USB	800mA, (3.5 W) / 5V vía Micro USB

Tabla 6-1: Características principales de los distintos modelos de Raspberry Pi

Diferencias entre el diseño original y el de Raspberry Pi

Debido a que todos los recursos del sistema Raspberry serían usados para monitorizar una red, se cuenta con la libertad de modificar tanto el diseño inicial de i2BasqueNMS, así como el sistema operativo sobre el que se ejecuta [4][5].

En la figura que se encuentra a continuación se muestran los cambios realizados en el diseño (véase *Ilustración 6-34*).

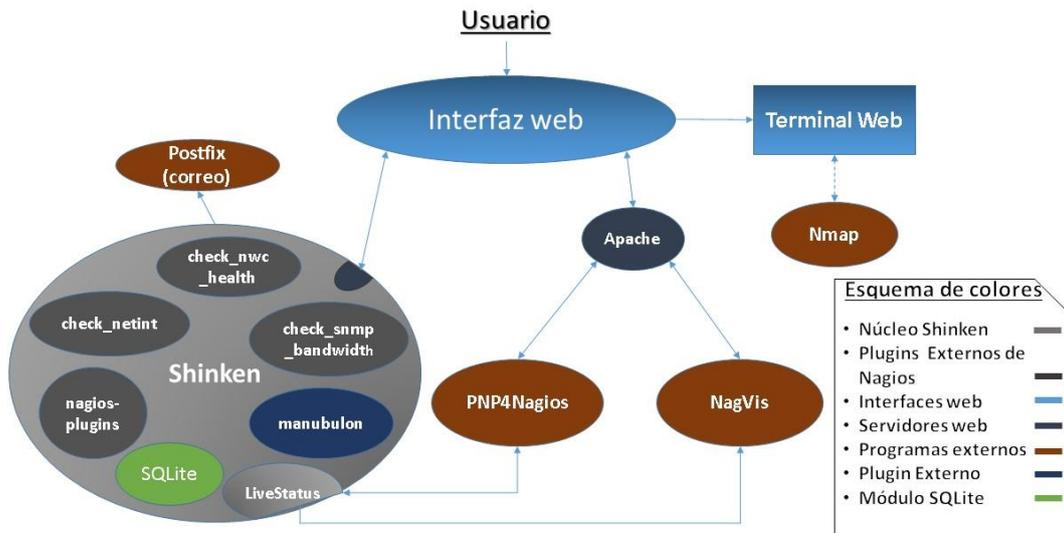


Ilustración 6-34 Esquema del diseño para Raspberry PI

Como se puede observar, debido a la total libertad de modificación, se ha configurado por defecto el servidor de correo Postfix y se ha añadido un módulo que permite a Shinken usar una base de datos SQLite, mucho más ligera que la MongoDB que usada por defecto.

El sistema operativo usado para crear la imagen de la versión de i2BasqueNMS para Raspberry Pi, es una modificación reducida de Debian para arquitectura armv6 y armv7 [12]. Dicho sistema apenas usa un 2% de la CPU y 18MB de los 512MB de la Raspberry Pi (2012). También se ha modificado el swappiness del sistema de 60 a 0. Dicho parámetro se encarga de indicar al sistema a que tanto por ciento de uso de memoria RAM debe empezar a paginar en el disco duro, cuanto mayor sea la cifra antes comenzará a paginar (en este caso en la tarjeta SD donde se encuentra instalado).

Para incrementar un poco más el rendimiento del sistema se ha añadido la opción "relatime" en el montaje de la partición principal del sistema. Dicha opción limita las actualizaciones de un parámetro de los archivos y directorios que indica la hora a la que se han accedido/modificado. No proporciona tanto incremento en el rendimiento como la opción de montaje "noatime", porque esta última desactiva completamente la actualización de dicho parámetro. Por otro lado, aunque "relatime" incremente menos el rendimiento, evita los errores que puedan ocasionarse en el servidor de correo Postfix instalado, debido a que chequea los indicadores de acceso de los archivos y directorios.

De cara a facilitar su instalación al usuario, la configuración de las IPs en PNP4Nagios y en NagVis se ha automatizado mediante dos scripts que se ejecutan en el inicio del sistema. Gracias a ello, se asegura un correcto funcionamiento del sistema de monitorización aun cambiando la IP otorgada a la Raspberry.

Para evitar que el usuario deba escanear la red para conocer la dirección IP otorgada por el servidor DHCP del router/switch a la Raspberry, se ha creado un script que al iniciarse el sistema recoge: la dirección IP, la MAC (único para cada equipo), la hora y la fecha actuales; y envía un email con los datos a la dirección de correo electrónico que el usuario haya especificado en la instalación.

La versión creada de i2BasqueNMS para estos pequeños ordenadores se presenta de dos maneras:

- Una imagen de disco, en cuyo caso lo los pasos que debe realizar el usuario son: descargar la imagen en un sistema Linux, montar la imagen, añadir su dirección email en un archivo, grabarla en una tarjeta SD/MicroSD e insertarla en la ranura de la Raspberry.
- Un archivo comprimido que contiene todo el código de la versión original de i2BasqueNMS pero adaptada para Raspberry Pi, incluido un script de instalación y otro de desinstalación. Cuyo fin es permitir que los usuarios que quieran usar su Raspberry para otros fines a parte de la monitorización, puedan hacerlo.

A continuación se realizarán instalaciones limpias tanto del diseño original como del diseño modificado para Raspberry Pi, con el fin de asegurar un correcto funcionamiento de los respectivos scripts de instalación.

7 VERIFICACIÓN DE I2BASQUENMS

7.1 Instalación de la versión original

La siguiente instalación se realiza sobre una máquina virtual con Debian 7 Wheezy x86_64bit.

El script creado para la instalación completa de i2BasqueNMS es la unión de todos los scripts creados durante el desarrollo. Esto ha dado como resultado un script con 200 líneas de código que automatiza tanto la instalación, la configuración y la corrección de todos los errores, encontrados durante el desarrollo, de los programas que conforman la herramienta de monitorización.

Las capturas (*véase Ilustraciones 7-01, 7-02 y 7-03*) corresponde a distintas fases de la ejecución del script.

```
Configurando libjs-jquery (1.7.2+dfsg-1) ...
Configurando python-simplejson (2.5.2-1) ...
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  python-netifaces
0 actualizados, 1 se instalarán, 0 para eliminar y 58 no actualizados.
Se necesita descargar 0 B/13,8 kB de archivos.
Se utilizarán 109 kB de espacio de disco adicional después de esta operación.
Seleccionando el paquete python-netifaces previamente no seleccionado.
(Leyendo la base de datos ... 45604 ficheros o directorios instalados actualmente.)
Desempaquetando python-netifaces (de ../python-netifaces_0.8-1_amd64.deb) ...
Configurando python-netifaces (0.8-1) ...
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
gawk ya está en su versión más reciente.
0 actualizados, 0 se instalarán, 0 para eliminar y 58 no actualizados.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
rsync ya está en su versión más reciente.
0 actualizados, 0 se instalarán, 0 para eliminar y 58 no actualizados.
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
lsb-release ya está en su versión más reciente.
0 actualizados, 0 se instalarán, 0 para eliminar y 58 no actualizados.
-----
Se instala Shinken, PNP4Nagios, NagVis y Nmap
Junto con:
1 plugin de Shinken
1 paquete de módulos de Nagios
4 módulos extra de Nagios
-----

> Found installation parameters
--> ETC=/usr/local/shinken/etc
--> VAR=/usr/local/shinken/var
--> LIBEXEC=/usr/local/shinken/libexec
--> TARGET=/usr/local/shinken
> Do you want to reuse those parameters? █
```

Ilustración 7-01: Inicio del script de instalación en su versión para Debian 7 Wheezy x86_64bit

```

> checking if shinken is installed in /usr/local/shinken
-----
| Install check_nwc_health plugin
-----
> Installing prerequisites
> Downloading check_nwc_health
> Extract check_nwc_health
> Build check_nwc_health.pl
> Install check_nwc_health.pl
> Found installation parameters
--> ETC=/usr/local/shinken/etc
--> VAR=/usr/local/shinken/var
--> LIBEXEC=/usr/local/shinken/libexec
--> TARGET=/usr/local/shinken
> checking if shinken is installed in /usr/local/shinken
-----
| Install install_check_netint
-----
> Installing pre-requisites.
> Downloading check_netint
> install check_netint
-----
Se copian los archivos modificados en las carpetas correspondientes
-----
Se soluciona el error de guardado de la configuracion general de NagVis
-----
Se soluciona el error del buscador de equipos monitorizados en PNP4Nagios
-----
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los siguientes paquetes se ELIMINARÁN:
 libapache2-mod-php5* php5*
0 actualizados, 0 se instalarán, 2 para eliminar y 36 no actualizados.
Se liberarán 9.241 kB después de esta operación.
¿Desea continuar [S/n]? █

```

Ilustración 7-02: A mitad de la instalación de i2BasqueNMS en Debian 7 Wheezy x86_64bit

```

[... ] Restarting web server: apache2[Mon Mar 30 14:32:09 2015] [warn] module wsgi_module is already loaded, skipping
... waiting [Mon Mar 30 14:32:10 2015] [warn] module wsgi_module is already loaded, skipping
. ok
Restarting scheduler
. ok
Restarting poller
. ok
Restarting reactionner
. ok
Restarting broker
. ok
Restarting receiver
. ok
Restarting arbiter
Doing config check
. ok
. ok
root@pfc-server-03:/home/aaron/i2BasqueShinken_RC3# █

```

Ilustración 7-03: Final de la instalación de i2BasqueNMS. Una vez se instala, se configura y se solucionan los problemas, se reinicia tanto Apache como Shinken

7.2 Instalación de la versión para Raspberry Pi: script de instalación

La siguiente instalación se realiza sobre una imagen modificada de Debian 7 para ARM, con el propósito de aligerar la carga del sistema, en una Raspberry Pi de 2011.

El script de instalación resulta de una modificación del creado para la instalación en la arquitectura x86_64bit, pero contemplando ciertos aspectos propios de la Raspberry Pi tales como: el tamaño de las particiones de la tarjeta SD donde se encuentra el sistema, el uso de un sistema de bases de datos más ligero que MongoDB, el uso de un servidor de correo ligero y, por último, ciertas mejoras de rendimiento tanto en el uso de la memoria RAM así como en las opciones de montaje de las particiones del sistema.

Antes de la ejecución del script de instalación es necesario añadir el correo electrónico del administrador en el script raspIP.sh, para que este le envíe un correo indicando los datos de conexión una vez se instale la herramienta i2BasqueNMS y se reinicie el sistema.

Las siguientes capturas (véase *Ilustraciones 7-04*, y *7-05*) corresponde a distintas fases de la ejecución del script.

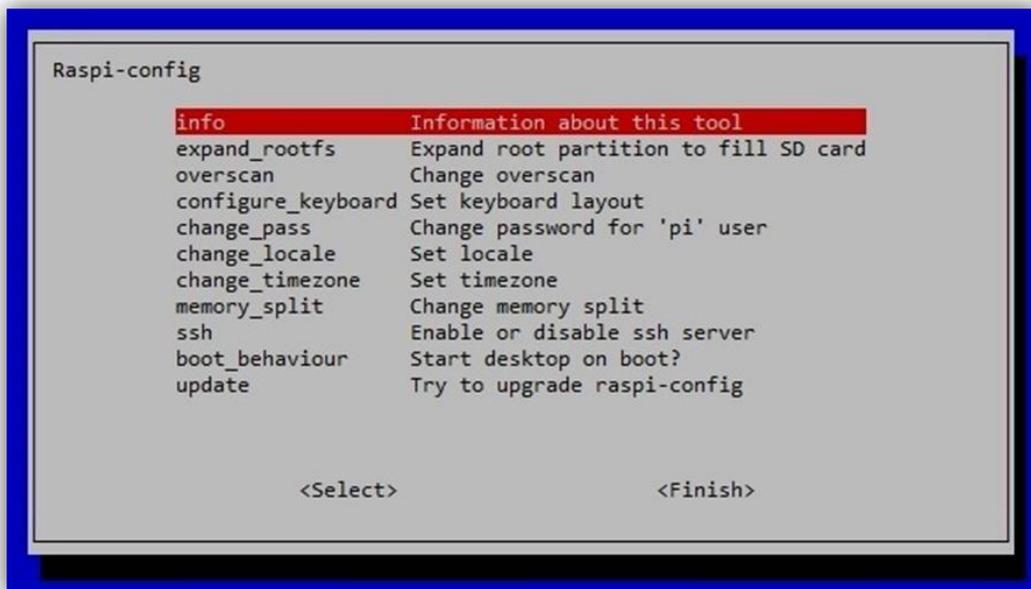


Ilustración 7-04: Configuración inicial de la instalación en una Raspberry Pi

```

Se actualizarán los siguientes paquetes:
comerr-dev dpkg dpkg-dev e2fslibs e2fsprogs file gnupg gpgv
libapache2-mod-php5 libc-bin libc-dev-bin libc6 libc6-dev libcomerr2
libcups2 libcupsimage2 libdpkg-perl libfreetype6 libgcrypt11 libgcrypt11-dev
libgd2-xpm libgnutls-dev libgnutls-openssl27 libgnutls26 libgnutlsxx27
libldap-2.4-2 libldap2-dev libmagic1 libssh2-1 libssl-dev libssl-doc
libssl1.0.0 libtasn1-3 libtasn1-3-dev libwbclient0 libxml2
linux-image-3.2.0-4-amd64 linux-libc-dev locales multiarch-support openssl
php-pear php5 php5-cli php5-common php5-gd php5-mysql php5-sqlite
samba-common samba-common-bin smbclient sudo tzdata
54 actualizados, 0 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 75,8 MB de archivos.
Se liberarán 2.804 kB después de esta operación.
¿Desea continuar [S/n]? S
Des:1 http://ftp.es.debian.org/debian/ wheezy-updates/main tzdata all 2015b-0whe
ezy1 [427 kB]
Des:2 http://security.debian.org/ wheezy/updates/main dpkg amd64 1.16.16 [2.662
kB]
Des:3 http://security.debian.org/ wheezy/updates/main libc-dev-bin amd64 2.13-38
+deb7u8 [226 kB]
Des:4 http://security.debian.org/ wheezy/updates/main libc6-dev amd64 2.13-38+de
b7u8 [2.665 kB]

```

Ilustración 7-05: Actualización de la versión minimizada de Debian para Raspberry Pi

7.3 Instalación de la versión para Raspberry Pi: imagen de disco

Con el propósito de facilitar la instalación lo máximo posible a los administradores que deseen monitorizar una red mediante una o varias Raspberry Pi, se ha creado una imagen pre-configurada de Debian más i2BasqueNMS.

Dicha imagen solo precisa la adición por parte del administrador de su correo electrónico del script “*PNP4NagiosIP.sh*” que se encuentra en */etc/init.d/* dentro de la imagen de disco. Para poder modificar dicho archivo se debe montar la imagen en un sistema Linux.

Tras la modificación de dicho archivo se debe desmontar la imagen del sistema, grabarla en una tarjeta SD, insertarla en la ranura de la Raspberry Pi y encenderla.

Las siguientes capturas (véase *Ilustraciones 7-06, 7-07*) corresponde a distintas fases de la ejecución de la instalación.

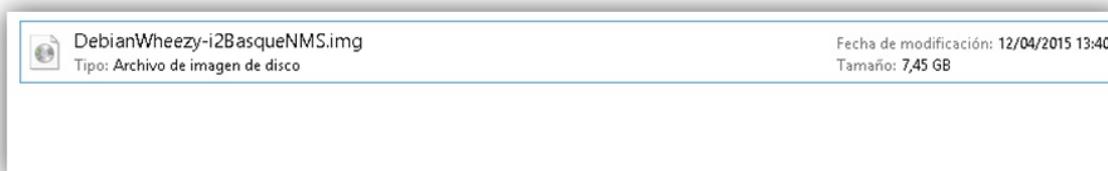


Ilustración 7-06: Imagen de disco de Debian con i2BasqueNMS preinstalado descargada

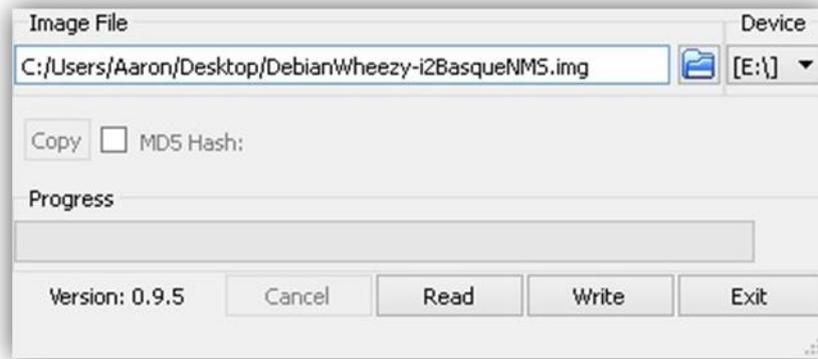


Ilustración 7-07: Grabación de la imagen de Debian junto con i2BasqueNMS creada para este TFG, en una tarjeta SD

8 COMPARATIVA ENTRE LA PLANIFICACIÓN INICIAL Y LA REAL

La planificación temporal inicial no ha podido ser cumplida principalmente por: dificultades no planificadas durante el desarrollo de los distintos prototipos y optimismo a la hora de suponer que el alumno podría disponer de 3 horas y media al día con diversas asignaturas por evaluar.

Por otra parte, también se ha añadido trabajo extra que no se encontraba planificado desde un principio, lo que ha incrementado algo más el tiempo necesario para la finalización del proyecto.

En la siguiente tabla se puede observar la comparativa entre el tiempo estimado y el real (véase *Tabla 8-1*).

Actividad	Estimación Inicial (semanas)	Tiempo Real (semanas)
Estudio de: bash scripting, php, html, perl, funcionamiento del protocolo snmp, etc	3 semanas	5 semanas
Estudio de antecedentes	1 semana	2 semanas
Captura de requisitos	1 semana	1 semana
Diseño de i2BasqueNMS	2 semanas	3 semanas
Desarrollo del primer prototipo	3 semanas	6 semanas
Pruebas del primer prototipo	1 semana	2 semanas
Desarrollo del segundo prototipo	3 semanas	5 semanas
Pruebas del segundo prototipo	1 semana	2 semanas
Desarrollo del tercer prototipo	3 semanas	3 semanas
Pruebas del tercer prototipo	1 semana	1 semana
Verificación de la correcta instalación y posterior funcionamiento	2 semanas	4 semanas

Trabajo Extra	No Planificado inicialmente	4 semanas
Tiempo total	21 semanas	38 semanas

Tabla 8-1: Comparación entre la estimación de tiempo inicial y el tiempo real

Como se puede observar en el siguiente gráfico, la diferencia entre la estimación inicial y el tiempo real es destacable (véase *Ilustración 8-01*).

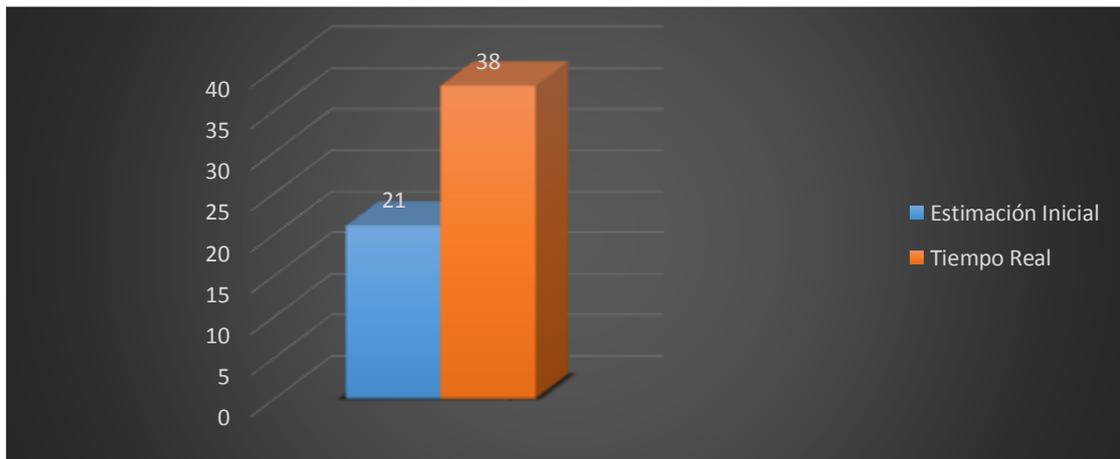


Ilustración 8-01: Comparación gráfica entre la estimación inicial y el tiempo real

9 CONCLUSIONES Y TRABAJO FUTURO

Para este TFG, i2Basque propuso la creación de una herramienta de monitorización (i2BasqueNMS) mediante la unificación de distintos programas OpenSource que juntos cubrieran las necesidades de monitorización de la red de la organización.

Al comienzo, se planteó Nagios como software central de la herramienta de monitorización. Tras investigar acerca de las distintas opciones disponibles, se llegó a la conclusión de que la mejor opción era Shinken junto con los programas: PNP4Nagios, NagVis, Shellinabox y Nmap. Una vez diseñada la estructura de i2BasqueNMS se procedió al desarrollo de tres prototipos incrementales, añadiendo en cada iteración nuevas funcionalidades a la herramienta.

En vista de los distintos problemas surgidos y solventados a lo largo de la construcción de los prototipos, se decidió crear un script que aunara todas las correcciones realizadas durante el desarrollo. Con este script se simplifica la instalación de i2BasqueNMS.

Las funcionalidades ofrecidas en la versión final son las mismas que las disponibles en gran parte de las herramientas de monitorización de pago, añadiendo la ventaja de ser gratuita, ya que combina software gratuito y Open Source.

Tras la construcción de una herramienta de monitorización capaz de cubrir las necesidades expuestas por i2Basque para este proyecto, existen otras funcionalidades que podrían añadirse en futuras ampliaciones, como por ejemplo:

- Implementación de un software intermedio que permita la gestión de los usuarios de manera unificada permitiendo al administrador crear y eliminar usuarios, así como gestionar cada uno de los permisos concedidos a estos para cada una de las aplicaciones que conforman i2BasqueNMS.
- Aumento del número de equipos y sistemas de diferentes marcas y modelos que puedan ser monitorizados mediante la herramienta i2BasqueNMS.
- Creación de nuevos diseños, tomando como referencia el actual, que estén más enfocados en la monitorización distribuida de la redes.

10 BIBLIOGRAFIA

- [1] Mauro, D., & Schmidt, K. (2005). *Essential snmp*. " O'Reilly Media, Inc."
- [2] Walsh, L. (2008). *SNMP MIB Handbook: essential guide to MIB development, use and diagnosis*. Wyndham.
- [3] Burgess, C. (2005). *The Nagios Book*.
- [4] Pollei, R. P. (2013). *Debian 7: System Administration Best Practices*. Packt Publishing Ltd.
- [5] Watson, M. A. (2014). *The BIG Book of Raspberry Pi*. Alison Watson.

Fuentes electrónicas:

- [6] Documentación oficial de Shinken: <http://shinken.readthedocs.org/en/latest/>
- [7] Foro oficial de Shinken: <http://forum.shinken-monitoring.org/>
- [8] Documentación oficial de PNP4Nagios: <https://docs.pnp4nagios.org/pnp-0.6/start>
- [9] Documentación oficial de NagVis: http://docs.nagvis.org/1.8/en_US/index.html
- [10] Página oficial de Shell in a Box: <https://code.google.com/p/shellinabox/>
- [11] Página oficial de Nmap: <http://nmap.org/book/man.html>
- [12] Página oficial de minibian: <https://minibianpi.wordpress.com/>