Desarrollo con tecnologías emergentes



TG3\_Final

Monitorización de redes y servidores



Contenido

[1. Autores del trabajo, planificación y entrega 3](#_Toc481951225)

[1.1 Autores 3](#_Toc481951226)

[1.2 Planificación 3](#_Toc481951227)

[1.3 Entrega 3](#_Toc481951228)

[2. Requisitos del prototipo a implementar 4](#_Toc481951229)

[2.1 Requisitos funcionales 4](#_Toc481951230)

[2.2 Otros requisitos 5](#_Toc481951231)

[3. Criterios de comparación en la implementación 7](#_Toc481951232)

[3.1 Criterio 1: Tiempo empleado para el desarrollo 7](#_Toc481951233)

[3.2 Criterio 2: Peso del programa 7](#_Toc481951234)

[3.3 Criterio 3: Tiempo de arranque de tecnologías 7](#_Toc481951235)

[3.4 Criterio 4: Recursos necesarios de CPU. 7](#_Toc481951236)

[3.5 Criterio 5: Recursos necesarios de Memoria RAM 7](#_Toc481951237)

[3.6 Criterio 6: Precio 8](#_Toc481951238)

[3.7 Criterio 7: Multiplataforma 8](#_Toc481951239)

[3.8 Criterio 8: Facilidad de aprendizaje 8](#_Toc481951240)

[3.9 Criterio 9: Almacenamiento 8](#_Toc481951241)

[3.10 Criterio 10: Actualizaciones 8](#_Toc481951242)

[Tipo de valor: Bolleano (Sí / No). 8](#_Toc481951243)

[4. Proyecto de implementación de un prototipo del sistema utilizando Icinga 9](#_Toc481951244)

[4.1 Documentación de diseño 9](#_Toc481951245)

[4.2 Documentación de construcción 13](#_Toc481951246)

[4.3 Documentación de pruebas 13](#_Toc481951247)

[4.4 Documentación de instalación 13](#_Toc481951248)

[4.5 Manual de usuario 18](#_Toc481951249)

[5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B 21](#_Toc481951250)

[5.1 Documentación de diseño 21](#_Toc481951251)

[5.2 Documentación de construcción 26](#_Toc481951253)

[5.3 Documentación de pruebas 26](#_Toc481951254)

[5.4 Documentación de instalación 27](#_Toc481951255)

[5.5 Manual de usuario 31](#_Toc481951256)

[6. Comparación de las dos implementaciones 39](#_Toc481951257)

[6.1 Evaluación de los criterios en la implementación de Icinga 39](#_Toc481951258)

[6.2 Evaluación de los criterios en la implementación de PandoraFMS 39](#_Toc481951259)

[7. Comparación de la implementación de las tecnologías 40](#_Toc481951260)

[8. Conclusiones 42](#_Toc481951261)

# 1. Autores del trabajo, planificación y entrega

## 1.1 Autores

Pertenecemos al grupo T6

Integrantes del grupo:

* Sergio Martín Míguez (Coordinador)
* Iván Alejandro Marugán
* Daniel Corral García
* Juan Felipe Martín Martín
* Silvia del Valle Recio

## 1.2 Planificación

Como ya sabemos, el peso de esta práctica es de un 30%, y por tanto requiere una dedicación por parte de cada alumno de 45 horas del total, lo que al final supone un total de 150 horas.

El desarrollo de la planificación que hemos llevado a cabo para este trabajo, lo hemos realizado usando la herramienta GanttPro. Seguidamente te adjuntamos el link para su posterior visualización:

[Planificación en GanttPro](https://app.ganttpro.com/shared/token/5b30b32d30dbb34e1e4b1e2786b38e261778552d979e6bec9889bdb94966f6b1/181235)

El reparto de las tareas realizadas por cada miembro del grupo ha sido equitativo, con un total de 45 horas para cada uno, más un total de 10 horas para la realización y preparación de los documentos finales de Word y Powerpoint.

## 1.3 Entrega

Procedemos a adjuntarte el enlace (URL) al repositorio en GitHub que hemos creado, donde se encuentra nuestro trabajo y los respectivos archivos de cada uno.

[GitHub Grupo T6 - TG3](https://github.com/sergiomartinm/TG3)

En este repositorio, hemos incluido una carpeta TG3 donde cada uno de los colaboradores hemos subido nuestras respectivas partes, y a su vez, en la rama master hemos incluido los dos archivos finales requeridos:

* Trabajo terminado: TG3\_final.docx
* Presentación del trabajo: TG3\_final.pptx

# 2. Requisitos del prototipo a implementar

El objetivo de este apartado se basa en reflejar cuales son cada uno de los requisitos de ambas tecnologías (Icinga y PandoraFMS), puesto que una de las condiciones de este apartado, es que los requisitos del prototipo deben ser cumplidos por las dos tecnologías que vamos a desarrollar. Por lo tanto, el conjunto de los requisitos deben ser compartidos por ambas tecnologías como punto de partida.

Con el objetivo de que quede bien especificado, vamos a intentar detallar al máximo cada requisito, de manera que consigamos mostrar que ambas tecnologías tienen la misma funcionalidad.

## 2.1 Requisitos funcionales

Como ya hemos comentado anteriormente, los requisitos funcionales deben ser los mismos para las dos implementaciones.

En la siguiente tabla se indicará el catálogo de requisitos funcionales del sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de requisito** | RF-001 | | |
| **Nombre de requisito** | Mapa de red | | |
| **Tipo** | Funcional | Datos Rendimiento Interfaz Seguridad Otros | |
| **Prioridad del requisito** | Alta/Esencial | Media/Deseado | Baja/ Opcional |
| **Descripción** | Esta funcionalidad permite visualizar gráficamente los nodos y sus relaciones, agentes, módulos y grupos disponibles para el usuario. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de requisito** | RF-002 | | |
| **Nombre de requisito** | Vistas | | |
| **Tipo** | Funcional | Datos Rendimiento Interfaz Seguridad Otros | |
| **Prioridad del requisito** | Alta/Esencial | Media/Deseado | Baja/ Opcional |
| **Descripción** | Los sistemas ofrecen muchos tipos de vistas con el objetivo de adecuarse a las necesidades del usuario. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de requisito** | RF-003 | | |
| **Nombre de requisito** | Estadísticas | | |
| **Tipo** | Funcional | Datos Rendimiento Interfaz Seguridad Otros | |
| **Prioridad del requisito** | Alta/Esencial | Media/Deseado | Baja/ Opcional |
| **Descripción** | Los sistemas ofrecen la posibilidad de crear estadísticas personalizadas con los datos recopilados. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de requisito** | RF-004 | | |
| **Nombre de requisito** | Informes | | |
| **Tipo** | Funcional | Datos Rendimiento Interfaz Seguridad Otros | |
| **Prioridad del requisito** | Alta/Esencial | Media/Deseado | Baja/ Opcional |
| **Descripción** | Los sistemas ofrecen la posibilidad de crear informes personalizados con los datos recopilados. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de requisito** | RF-005 | | |
| **Nombre de requisito** | Alertas | | |
| **Tipo** | Funcional | Datos Rendimiento Interfaz Seguridad Otros | |
| **Prioridad del requisito** | Alta/Esencial | Media/Deseado | Baja/ Opcional |
| **Descripción** | Los sistemas ofrecen la posibilidad de crear alertas de distintos módulos (CPU, memoria, disco…) | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de requisito** | RF-006 | | |
| **Nombre de requisito** | Exportar datos | | |
| **Tipo** | Funcional | Datos Rendimiento Interfaz Seguridad Otros | |
| **Prioridad del requisito** | Alta/Esencial | Media/Deseado | Baja/ Opcional |
| **Descripción** | Permite exportar los datos recopilados por la herramienta en un periodo de tiempo establecido. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de requisito** | RF-007 | | |
| **Nombre de requisito** | Gestor de agente software | | |
| **Tipo** | Funcional | Datos Rendimiento Interfaz Seguridad Otros | |
| **Prioridad del requisito** | Alta/Esencial | Media/Deseado | Baja/ Opcional |
| **Descripción** | Permite gestionar los distintos equipos que están monitorizados. | | |

## 2.2 Otros requisitos

El resto de los requisitos, los hemos recogido en la siguiente tabla, donde se indicará el catálogo de requisitos no funcionales del sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de requisito** | RNF-001 | | |
| **Nombre de requisito** | Visualización de eventos | | |
| **Tipo** | Funcional | Datos Rendimiento Interfaz Seguridad Otros | |
| **Prioridad del requisito** | Alta/Esencial | Media/Deseado | Baja/ Opcional |
| **Descripción** | Permite ver las diferentes opciones de cada evento en tiempo real. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de requisito** | RNF-002 | | |
| **Nombre de requisito** | Usuarios conectados | | |
| **Tipo** | Funcional | Datos Rendimiento Interfaz Seguridad Otros | |
| **Prioridad del requisito** | Alta/Esencial | Media/Deseado | Baja/ Opcional |
| **Descripción** | Permite ver los diferentes usuarios conectados en tiempo real. | | |
| **Número de requisito** | RNF-003 | | |
| **Nombre de requisito** | Desconexión programada | | |
| **Tipo** | Funcional | Datos Rendimiento Interfaz Seguridad Otros | |
| **Prioridad del requisito** | Alta/Esencial | Media/Deseado | Baja/ Opcional |
| **Descripción** | Permite planificar la parada del servicio cuando un usuario lo desee. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Número de requisito** | RNF-004 | | |
| **Nombre de requisito** | Acceso seguro | | |
| **Tipo** | Funcional | Datos Rendimiento Interfaz Seguridad Otros | |
| **Prioridad del requisito** | Alta/Esencial | Media/Deseado | Baja/ Opcional |
| **Descripción** | Permite conectar a los sistemas mediante acceso seguro. | | |

# 3. Criterios de comparación en la implementación

El objetivo de este apartado se basa en reflejar cuales son cada uno de los criterios de comparación de ambas tecnologías (Icinga y PandoraFMS), puesto que si no, no se podría realizar una comparación clara y concisa de la implementación de ambas tecnologías.

Con el objetivo de que quede bien especificado, vamos a intentar detallar al máximo cada criterio, de manera que consigamos mostrar que ambas tecnologías tienen la misma funcionalidad.

## 3.1 Criterio 1: Tiempo empleado para el desarrollo

*Nombre del criterio: Tiempo empleado para el desarrollo.*

*Descripción: Número de horas empleadas en la configuración y puesta a punto del sistema.*

*Tipo de valor: Numérico (horas).*

## 3.2 Criterio 2: Peso del programa

*Nombre del criterio: Peso del programa.*

*Descripción: Cantidad de MegaBytes que ocupa el software de las tecnologías.*

*Tipo de valor: Numérico (MB).*

## 3.3 Criterio 3: Tiempo de arranque de tecnologías

*Nombre del criterio: Tiempo de arranque de tecnologías.*

*Descripción: Tiempo que tarda la tecnología en ponerse en funcionamiento desde que se ejecuta.*

*Tipo de valor: Numérico (segundos).*

## 3.4 Criterio 4: Recursos necesarios de CPU.

*Nombre del criterio:* Recursos necesarios de CPU*.*

*Descripción: Media de consumo de recursos de CPU por parte de la tecnología.*

*Tipo de valor: Porcentaje.*

## 3.5 Criterio 5: Recursos necesarios de Memoria RAM

*Nombre del criterio: Recursos necesarios de Memoria RAM.*

*Descripción: Media de consumo de recursos de Memoria RAM por parte de la tecnología.*

*Tipo de valor: Numérico (KiloBytes).*

## 3.6 Criterio 6: Precio

*Nombre del criterio: Precio.*

*Descripción: Dinero invertido en la instalación y configuración de ambas tecnologías.*

*Tipo de valor: Numérico (Pesetas).*

## 3.7 Criterio 7: Multiplataforma

*Nombre del criterio: Multiplataforma.*

*Descripción: Número de plataformas que soportan las tecnologías.*

*Tipo de valor: Numérico.*

## 3.8 Criterio 8: Facilidad de aprendizaje

*Nombre del criterio: Facilidad de aprendizaje.*

*Descripción: Número de horas empleadas en el aprendizaje para una correcta ejecución del sistema.*

*Tipo de valor: Numérico (horas).*

## 3.9 Criterio 9: Almacenamiento

*Nombre del criterio:* Almacenamiento*.*

*Descripción: Cantidad que ocupa en el disco la instalación de las tecnologías.*

*Tipo de valor: Numérico (MegaBytes).*

## 3.10 Criterio 10: Actualizaciones

*Nombre del criterio:* Actualizaciones*.*

*Descripción: Indicar si existen actualizaciones en las interfaces de las tecnologías..*

## *Tipo de valor: Bolleano (Sí / No).*

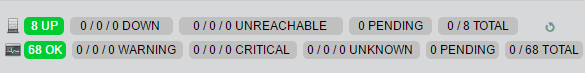
# 4. Proyecto de implementación de un prototipo del sistema utilizando Icinga

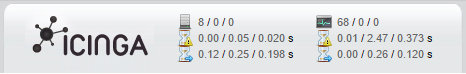
## 4.1 Documentación de diseño

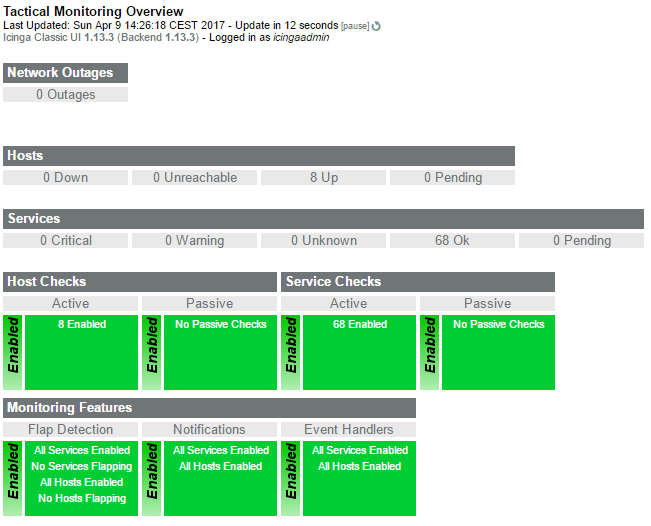
#### PÁGINA PRINCIPAL + HEADER

Se encuentra un resumen completo del todo sistema de monitorización.

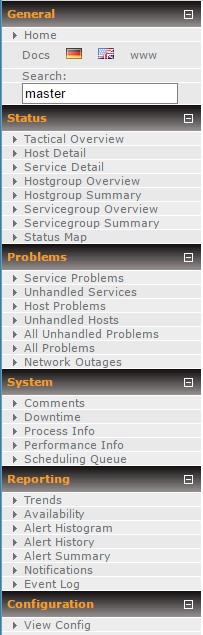








#### SIDEBAR

Se encuentran todos los accesos al aplicativo:

* **Status**

Apartados donde se resumen los servicios, hosts y status map del sistema de monitorización.

* **Problems**

Histórico de fallos de servidores, servicios, red…

* **System**

Información sobre los comentarios de los servicios, si se ha configurado una baja manualmente de los servidores o servicios. Además, se muestra un resumen de las estadísticas de uso del propio sistema.

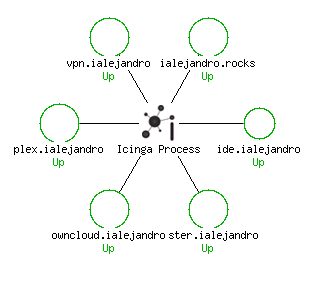
* **Reporting**

Resumen de históricos y generación de informes para su estudio posterior.

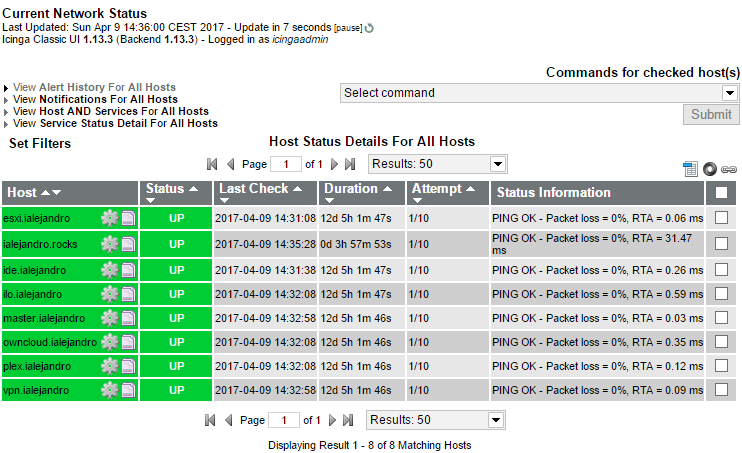
* **Configuration**

Se encuentran todos los parámetros que se están aplicando al aplicativo.

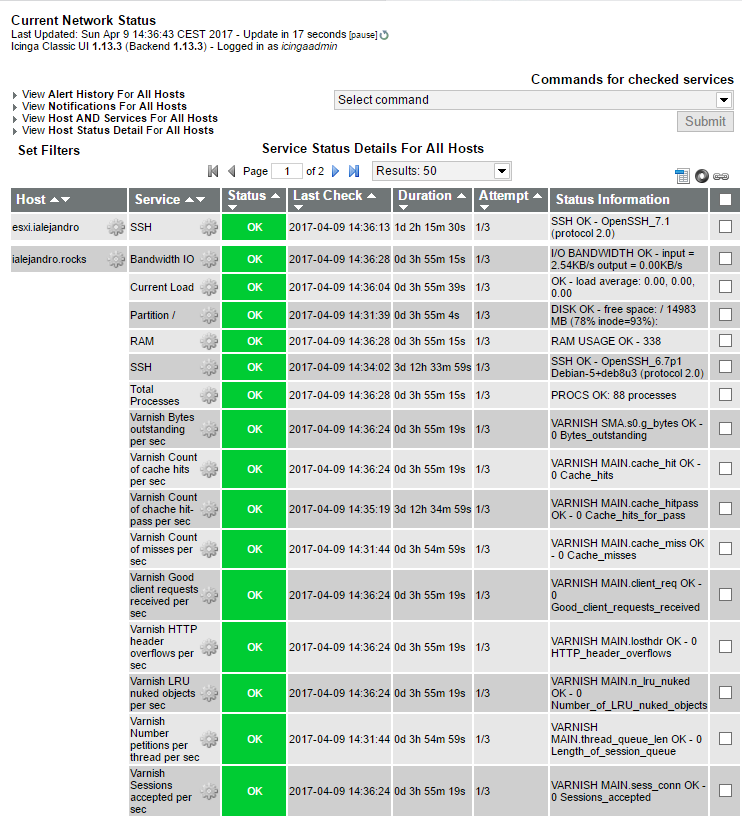
#### STATUS: STATUS MAP



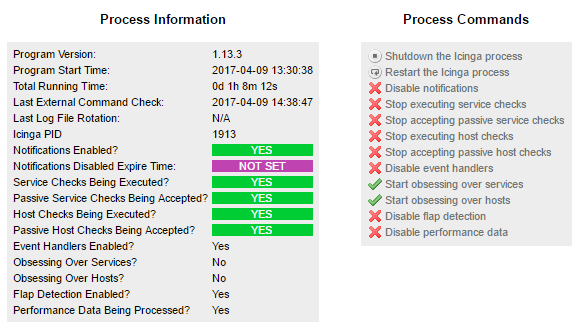
#### STATUS: RESUMEN DE LOS SERVIDORES



#### STATUS: RESUMEN DE LOS SERVICIOS



#### SYSTEM: PROCESS INFO



#### REPORTING: ALERT HISTORY



## 4.2 Documentación de construcción

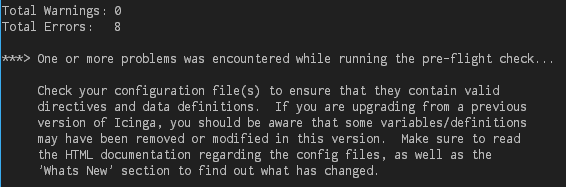
La instalación de Icinga es mediante paquetes pre-compilados, por tanto, solo se verían los comentarios del propio desarrollador. En este proyecto no hemos hecho ningún aporte al código.

## 4.3 Documentación de pruebas

La instalación ha sido mediante repositorios, por tanto, no hubo problemas. La mayoría de los problemas que aparecen en Icinga suelen ser cuando se aplica o se escribe una configuración errónea.

Antes de iniciar, reiniciar o recargar el servicio de Icinga se debe comprobar que la sintaxis de la configuración modificada es correcta. Si no, nos dará el siguiente fallo cuando se compruebe la sintaxis.





## 4.4 Documentación de instalación

#### PRE-REQUISITOS

* Añadiendo los repositorios de https://debmon.org/. Debmon (Debian Monitoring) es un proyecto de Debian enfocado a la monitorización.

wget -O - http://debmon.org/debmon/repo.key 2>/dev/null | apt-key add -

echo "deb http://debmon.org/debmon debmon-jessie main" >> /etc/apt/source.list

apt-get update

* Instalamos MySQL Server para guardar los datos de ido2db

apt-get install mysql-server-5.5 libdbd-mysql mysql-client-5.5

* Instalamos Apache2 + PHP5 para Icinga Classic UI

apt-get install apache2 php5 php5-common php5-cgi php5-cli php5-json php5-readline php5-gd

#### INSTALACIÓN

* Instalamos los paquetes de Icinga1:

apt-get install icinga icinga-doc icinga-web nagios-plugins icinga-idoutils

* Configuracion en la propia instalación:
  + En mitad de la instalación nos pedirá una contraseña para el usuario administrador: icingaadmin.
  + Cuando llegue a la configuración de ido2db (idoutils) necesitará un usuario y contraseña para escribir en la base de datos, previamente instalada (en los pre-requisitos).

**IDOUtils (ido2db: icinga data objects to database)**: Consiste en guardar los datos de los objetos en una base de datos.

#### POST-INSTALACIÓN: PNP4Nagios + RRDCached – Gráficas

* Instalación de PNP4Nagios + RRDCached:

apt-get install pnp4nagios pnp4nagios-bin pnp4nagios-web rrdcached

* Configuración de PNP4Nagios. Modificar el fichero "/etc/pnp4nagios/config.php":

$conf[‘nagios\_base’] = “/cgi-bin/icinga”;

* Configuración con Bulk Mode (carga masiva) + NPCD. Modificar el fichero "/etc/icinga/icinga.cfg":

process\_performance\_data=1

* Modificamos el fichero "/etc/default/npcd":

RUN=”yes”

* Añadimos las nuevas templates para PNP4Nagios en la configuración de Icinga. Modificamos el fichero "/etc/icinga/objects/pnptemplates.cfg" añadiendo dos nuevos objetos.

define host {

name host-pnp

action\_url /pnp4nagios/index.php/graph?host=$HOSTNAME$&srv=\_HOST\_' class='tips' rel='/pnp4nagios/index.php/popup?host=$HOSTNAME$&srv=\_HOST\_

register 0 ; Lo convierte en plantilla

}

define service {

name srv-pnp

action\_url /pnp4nagios/index.php/graph?host=$HOSTNAME$&srv=$SERVICEDESC$' class='tips' rel='/pnp4nagios/index.php/popup?host=$HOSTNAME$&srv=$SERVICEDESC$

register 0 ; Lo convierte en plantilla

}

Para terminar añadimos estas plantillas a los objetos genéricos de servidores (**/etc/icinga/objects/generic-host.cfg**) y servicios (**/etc/icinga/objects/generic-service.cfg**)

# Servidores

define host{

name generic-host ; The name of this host template

use pnp-hst ; <-- PNP4Nagios

notifications\_enabled 1 ; Host notifications are enabled

event\_handler\_enabled 1 ; Host event handler is enabled

flap\_detection\_enabled 1 ; Flap detection is enabled

failure\_prediction\_enabled 1 ; Failure prediction is enabled

process\_perf\_data 1 ; Process performance data

retain\_status\_information 1 ; Retain status information across program restarts

retain\_nonstatus\_information 1 ; Retain non-status information across program restarts

check\_command check-host-alive

max\_check\_attempts 10

notification\_interval 0

notification\_period 24x7

notification\_options d,u,r

register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL HOST, JUST A TEMPLATE!

}

# Servicios

define service{

name generic-service ; The 'name' of this service template

use pnp-svc ; <-- PNP4Nagios

active\_checks\_enabled 1 ; Active service checks are enabled

passive\_checks\_enabled 1 ; Passive service checks are enabled/accepted

parallelize\_check 1 ; Active service checks should be parallelized (disabling this can lead to major performance problems)

obsess\_over\_service 1 ; We should obsess over this service (if necessary)

check\_freshness 0 ; Default is to NOT check service 'freshness'

notifications\_enabled 1 ; Service notifications are enabled

event\_handler\_enabled 1 ; Service event handler is enabled

flap\_detection\_enabled 1 ; Flap detection is enabled

failure\_prediction\_enabled 1 ; Failure prediction is enabled

process\_perf\_data 1 ; Process performance data

retain\_status\_information 1 ; Retain status information across program restarts

retain\_nonstatus\_information 1 ; Retain non-status information across program restarts

notification\_interval 0 ; Only send notifications on status change by default.

is\_volatile 0

check\_period 24x7

normal\_check\_interval 5

retry\_check\_interval 1

max\_check\_attempts 3

notification\_period 24x7

notification\_options w,u,c,r

register 0 ; DONT REGISTER THIS DEFINITION - ITS NOT A REAL SERVICE, JUST A TEMPLATE!

}

* Configurar rrdcached. Creamos el directorio donde se guardarán los datos para pintar en las gráficas:

mkdir -p /var/cache/rrdcached  
chown nagios:www-data /var/cache/rrdcached

* Modificamos la configuración de rrdcached (/etc/default/rrdcached):

OPTS=”-w 1800 -z 1800 -j /var/cache/rrdcached -s nagios -m 0660 -l unix:/var/run/rrdcached.sock”

* Configuramos PNP4Nagios para el uso de rrdcached. Modificamos el fichero "/etc/pnp4nagios/process\_perfdata.cfg"

RRD\_DAEMON\_OPTS = unix:/var/run/rrdcached.sock

* Modificamos el fichero "/etc/pnp4nagios/config.php"

$conf[‘RRD\_DAEMON\_OPTS’] = ‘unix:/var/run/rrdcached.sock’;

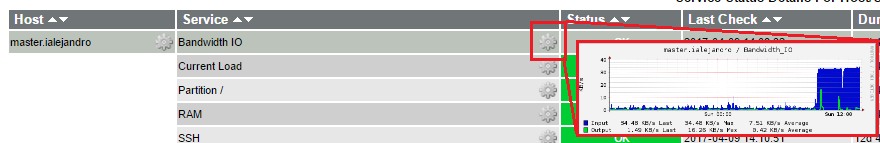
* Añadimos el usuario de "nagios" a "www-data":

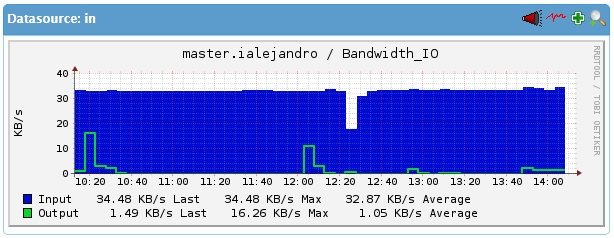
usermod -G nagios -a www-data

* Ahora reiniciamos e iniciamos los servicios correspondientes:

service npcd start  
service rrdcached restart  
service icinga restart  
service apache2 restart

Nos quedaría una gráfica así:





## 4.5 Manual de usuario

#### CONFIGURACIÓN

Icinga, por defecto, se instala en “**/etc/icinga**”:

* Explicación de jerarquía / directorios:

/etc/icinga -> Configuración de Icinga  
\  
|-> modules/ -> módulos de Icinga (ej. PNP4Nagios, Idoutils...)  
\  
|-> objects/ -> configuración de los objetos de Icinga (servidores, servicios, comandos, contactos, plantillas...)  
\  
|-> stylesheets/ -> estilos de interfaz Classic-UI

Explicación de ficheros de conf. importantes. Todos estos ficheros se encuentran en la raíz de instalación (/etc/icinga):

* **apache2.conf**: fichero de configuración para el VirtualHost de Icinga.
* **cgi.cfg**: fichero de configuracion de cgi. para dar accesos a los usuarios, configuración de muestra de diseños, rotaciones de log, horarios...
* **commands.cfg**: fichero donde se definen los comandos para envío de alertas, carga de gráficas (PNP4Nagios)...
* **icinga.cfg**: fichero de configuración del aplicativo, ejemplo: niveles de log, tipos de chequeos, etc.
* **htpasswd.users**: fichero donde se encuentran las credenciales que carga Apache2 para que accedan al aplicativo.

**¿Tipos de objetos?**

* Guía oficial: <https://docs.icinga.com/latest/en/objectdefinitions.html>

**Se explicarán los más usados:**

* **contact**: objeto que define un contacto que será usado para cuando surja algún problema le llegue la alerta.
* **contactgroup**: objeto para agrupar contactos, se usa para mandar una alerta a un grupo en específico en vez de a usuarios concretos que pertenecen al mismo departamento.
* **command**: objeto para la definición de un comando.
* **host**: objecto que define un servidor el cual vamos a monitorizar.
* **hostgroup**: objeto para agrupar servidores, se usa para cuando varios servidores comparten las mismas características asignarles un servicio al grupo en vez de hacerlo por servidor.
* **service**: objeto que define el servicio el cual vamos a chequear.
* **servicegroup**: objeto para agrupar servicios que pertenecen a un mismo campo o tecnología. Ejemplo: particiones de discos.
* **servicedependency**: objeto que sirve para crear servicios dependientes de otros. Por tanto, con esta definición evitamos que lleguen alertas de los hijos cuando falle el padre, ya que damos por hecho que el fallo del padre desencadenará fallo en los hijos.

**¿Cómo se cargan las configuraciones?**

El aplicativo carga por defecto todos los ficheros que terminen con la extensión ".cfg", independientemente del directorio en el que esté. Icinga recorre cada uno de directorios y va cargando todas las configuraciones, comandos, contactos, objetos...

#### EJEMPLOS

1. Servidores (hosts)

define host{  
 use ialejandro-servers  
 host\_name master.ialejandro  
 address master.ialejandro.hogar  
 hostgroups hogar,debian-hosts  
}

1. Grupo de Servidores (hostgroup)

define hostgroup {  
 hostgroup\_name debian-hosts  
 alias Debian GNU/Linux Hosts

}

1. Servicios (services)

define service {  
 use generic-service  
 hostgroup\_name hogar  
 service\_description SSH  
 check\_command check\_ssh  
}

1. Grupo de Servicios (servicegroup)

define servicegroup {  
 servicegroup\_name varnish  
 alias Varnish Checks  
}

1. Contactos (contact)

define contact {  
 contact\_name tg\_dte  
 use generic-contact  
 alias TG Bot  
 service\_notification\_commands notify-service-by-telegram  
 host\_notification\_commands notify-host-by-telegram  
}

1. Grupo de Contactos (contactgroup)

define contactgroup{  
 contactgroup\_name admins  
 alias Nagios Administrators  
 members root,tg\_dte  
}

1. Comandos (command)

# Definición de notificaciones de servidores por Telegram  
define command{  
 command\_name notify-host-by-telegram  
 command\_line /usr/bin/curl -X POST --data chat\_id="XXXXXXXXX" -  
data text="<b>/!\/!\ Alerta de Icinga /!\/!\</b>%0A%0A<b>Notification Type:</b> $NOTIFICATIONTYPE$%0A<b>Host:</b> $HOSTNAME$%0A<b>State:</b> $HOSTSTATE$%0A<b>Address:</b> $HOSTADDRESS$%0A<b>Info:</b> $HOSTOUTPUT$%0A%0A<b>Date/Time:</b> $LONGDATETIME$%0A%0A<b>Comments:</b> $NOTIFICATIONCOMMENT$" --data parse\_mode="HTML" <https://api.telegram.org/botXXXXXX/sendMessage>  
}

# Definición de chequeo de discos mediante SSH  
define command {  
 command\_name check\_disk\_ssh  
 command\_line $USER2$$HOSTADDRESS$ /usr/lib/nagios/plugins/check\_disk -w '$ARG1$' -c '$ARG2$' -p '$ARG3$'  
}

#### COMANDOS A DESTACAR

* Servicio de Icinga

systemctl [status|restart|reload|stop|start] icinga  
service [status|restart|reload|stop|start] icinga  
/etc/init.d/icinga [status|estart|reload|stop|start]

* Comprobar sintaxis

icinga -v /etc/init.d/icinga.cfg

#### DIRECTORIOS A DESTACAR

* /etc/icinga

Configuración global de Icinga

* /etc/pnp4nagios

Configuración global de PNP4nagios

* /usr/lib/nagios/plugins/

Se encuentran todos los scripts para chequear servicios.

* /var/lib/pnp4nagios/perfdata

Guarda toda la info para rrdcached y pintarlas en PNP4Nagios.

# 5. Proyecto de implementación de un prototipo del sistema utilizando la tecnología B

En este apartado, vamos a realizar una documentación en la que quede perfectamente detallado el proyecto de implementación utilizando PandoraFMS. Para que quede bien conciso, haremos especial detalle tanto en los requisitos funcionales como en los no funcionales definidos anteriormente en el apartado 2.

Hemos decidido dividir la documentación en los siguientes apartados descritos en el enunciado de la práctica:

* Documentación de diseño
* Documentación de construcción
* Documentación de pruebas
* Documentación de instalación
* Manual de usuario

## 5.1 Documentación de diseño

A continuación, se expone la documentación del diseño, en la que se hace una descripción del diseño del prototipo, incluyendo las principales pantallas que definen de una manera detallada la interfaz de usuario de Pandora FMS.

#### PÁGINA PRINCIPAL + HEADER

En esta vista de la página principal de Pandora, se muestra un resumen completo de todo el sistema de monitorización.

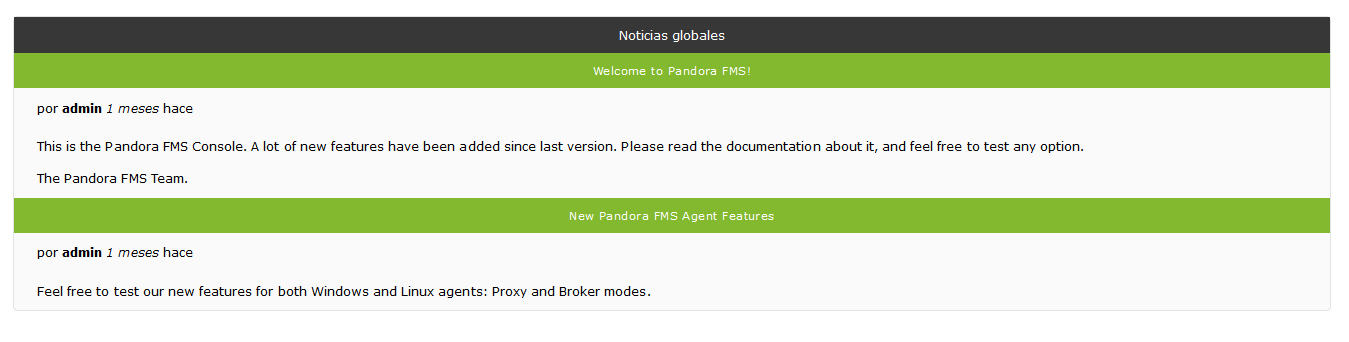
## 

Analizando punto por punto:

En la barra de encabezado están las siguientes opciones en orden: Asistente de PandoraFMS, Estado de los sistemas, modo auto refresco, advertencias, ayuda general, log out y mi usuario (donde se puede editar)



Más abajo se encuentra la feed de noticias:

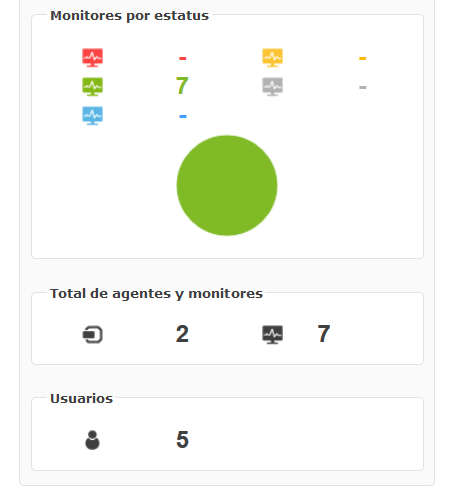


Y en la parte inferior de estas, se encuentra el historial de actividad de cada usuario:



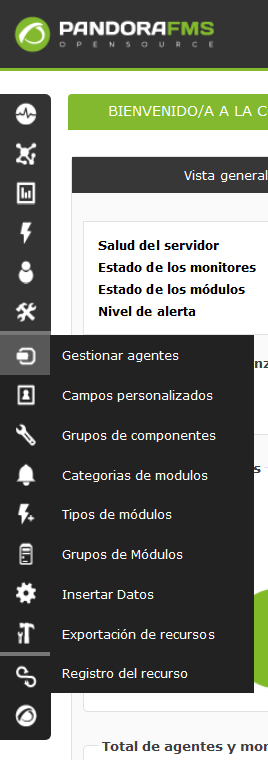
En la parte de la izquierda hay un resumen en el que se encuentra una vista general de PandoraFMS, donde se puede observar la salud del servidor, el estado de los monitores, el estado de los módulos, y el nivel de alertas.

Además se pueden ver las alertas que hay definidas, y las alertas que se han disparado a tiempo real.

Seguidamente podemos ver un resumen a modo gráfico (circular), de los diferentes estados que pueden tener los monitores, que pueden ser, monitores en estado crítico, monitores en estado normal, en warning, en estado no iniciado, y en estado desconocido.

Más abajo, se pueden ver todos los agentes y monitores que se encuentran monitorizados en un determinado momento.

Y abajo del todo, se pueden ver los usuarios que se encuentran definidos para este sistema de monitorización

A su vez, PandoraFMS ofrece una SIDEBAR donde se pueden realizar multitud de apartados como:

* Vistas y SNMP.
* Mapa de red y consola visual.
* Ver eventos, estadísticas, archivo CSV, eventos

sonoros, evento Matrix.

* Editar usuario, chat web, incidentes, mensajes, usuarios conectados.
* Exportar datos, desconexión programada, vista R reconocida, repositorio.
* Gestionar agentes, campos personalizados,

grupos de componentes, categorías de módulos

tipos de módulos, grupos de módulos, insertar

datos, exportación de recursos, y registro de recursos.

* Configuración de grupos de agentes, etiquetas de

Módulos, gestión de usuarios, gestionar perfiles,

usuarios conectados.

* Componentes de red, plantillas de módulos,

Operaciones masivas.

* Lista de alertas, plantillas, acciones, comandos,

lista de días especiales, alertas SNMP.

* Filtro de eventos, eventos personalizados y

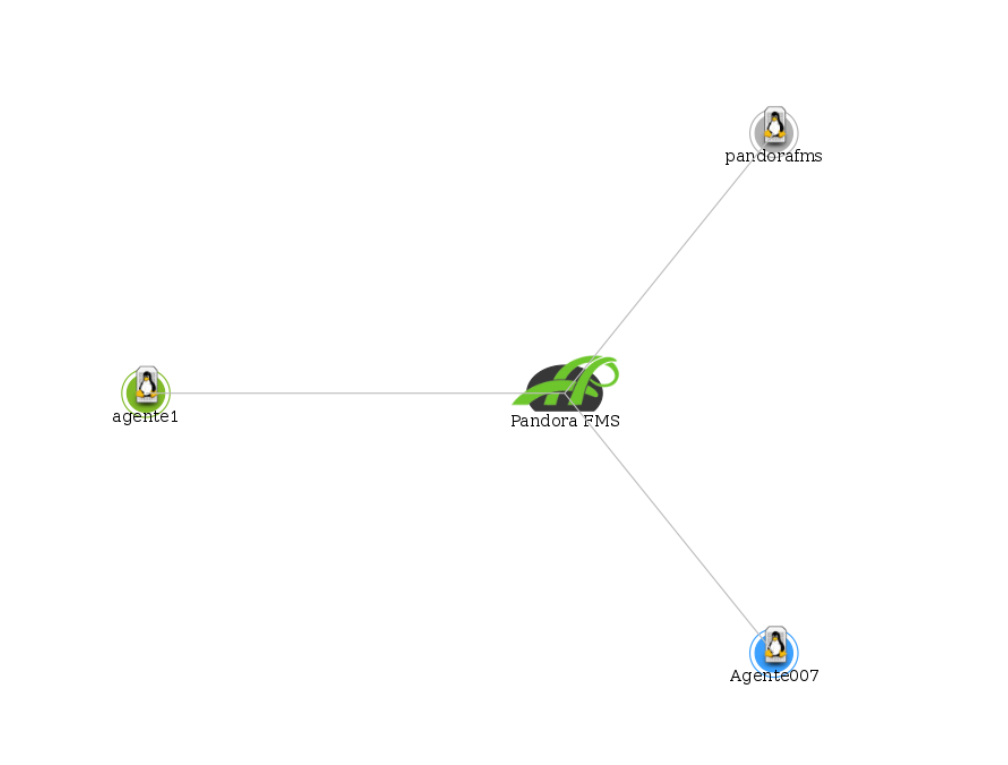
respuestas de eventos

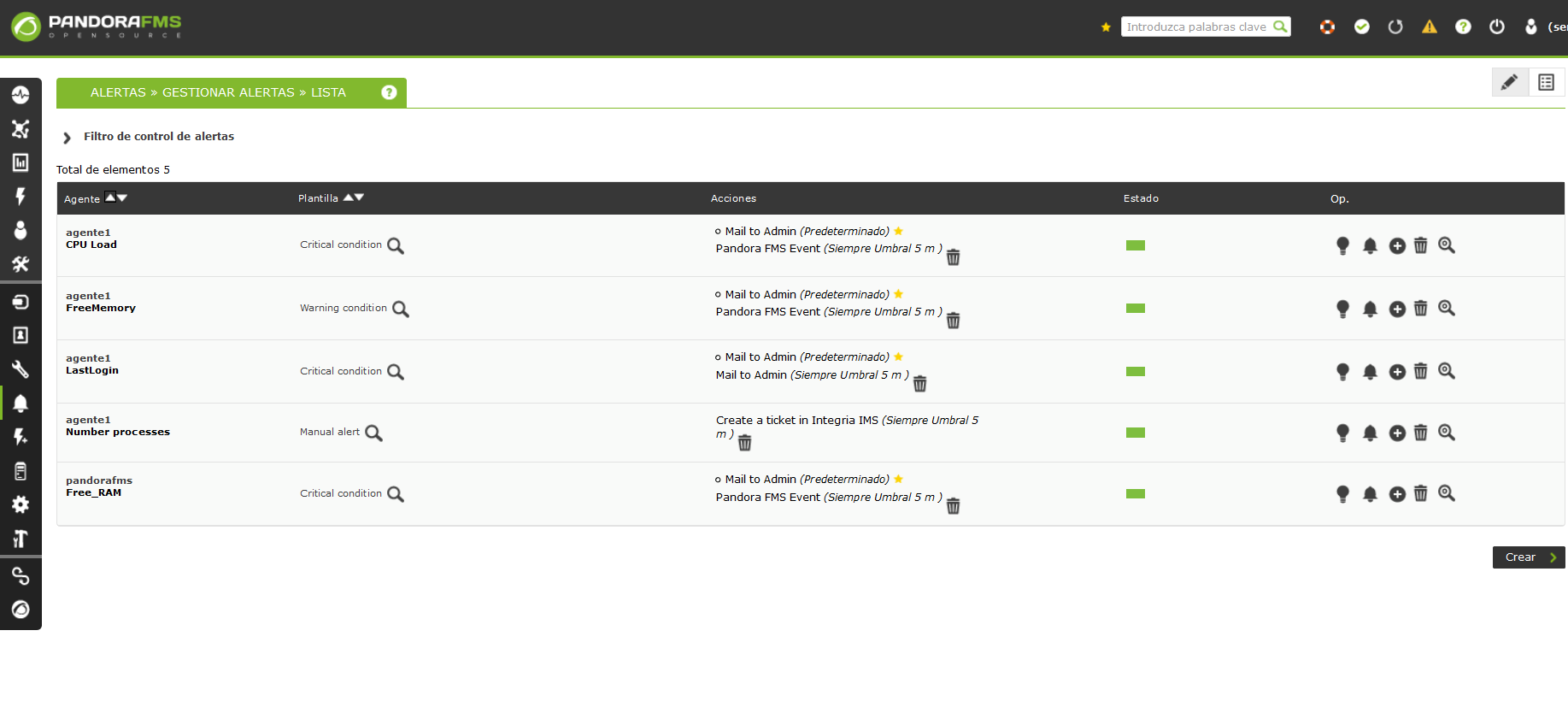
* Gestionar servidores, tareas de reconocimiento,

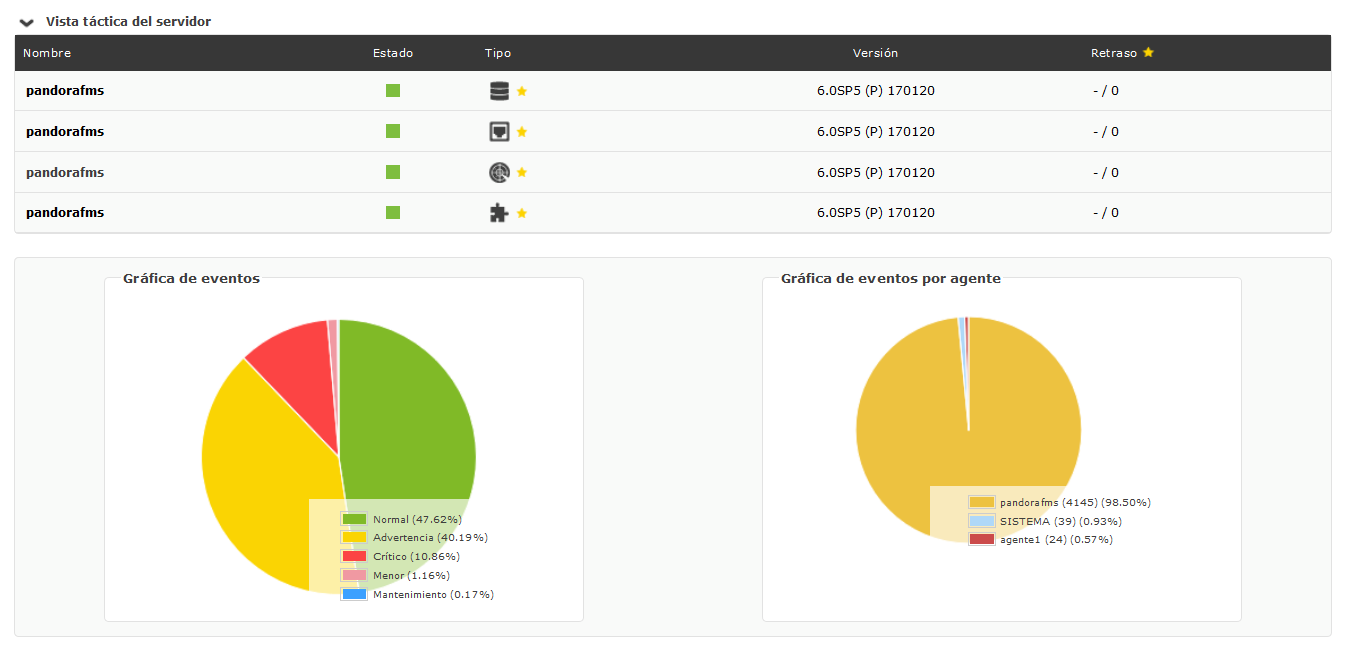
Plugins, recon script, registro plugin.

* Configuración, editar SO, licencia
* Auditoría del sistema, enlaces, información de diagnostico, noticias del sistema, gestor de archivos, gestión BBDD, información de Sistema, verificación API, gerente de extensión
* Module library, PandoraFMS, PandoraFMS Manual, report a bug, suggest new feature.
* Update manager sin conexión, update manager con conexión, opciones de update manager, mensajes.

MAPA DE RED



HISTORIAL DE ALERTAS:

VISTA TÁCTICA DEL SERVIDOR:

## 5.2 Documentación de construcción

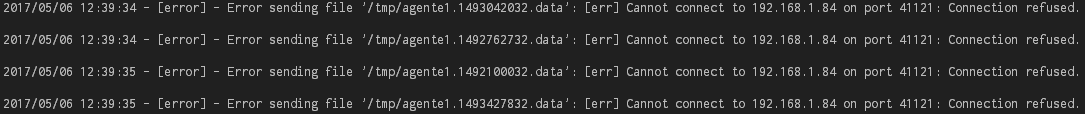
La instalación de PandoraFMS es mediante paquetes pre-compilados, por tanto, solo se verían los comentarios del propio desarrollador. En este proyecto no hemos hecho ningún aporte al código.

## 5.3 Documentación de pruebas

La instalación de PandoraFMS se ha realizado igual que la de Icinga, mediante repositorios, pero al contrario que ésta, hubo fallos al principio por falta de dependencias:

snmp snmpd libtime-format-perl libxml-simple-perl libdbi-perl libnetaddr-ip-perl libhtml-parser-perl wmi-client xprobe2 nmap libmail-sendmail-perl traceroute libio-socket-inet6-perl libhtml-tree-perl php5 libapache2-mod-php5 apache2 mysql-server php5-gd php5-mysql php-pear php5-snmp php-db php-gettext graphviz mysql-client php5-curl php5-xmlrpc php5-ldap libsnmp-perl snmp-mibs-downloader libio-socket-multicast-perl libsnmp-perl libjson-perl

Por lo tanto, fallaba la instalación de los paquetes. Una vez instalado seguían surgiendo fallos:



Fallos de conexión constantes con el agente (máquina 192.168.1.86) y el servidor principal: PandoraFMS (máquina 192.168.1.84).

Seguía fallando la configuración del master, PandoraFMS tiene un servicio: tentacle\_serverd. Este servicio no está arrancado por defecto, y nos llevó un tiempo encontrar el fallo.

**Tentacle** se encarga de recibir toda la información de los agentes de PandoraFMS que se encuentran en la red y de pasárselos al servicio de PandoraFMS, para que este los analice y trate la información recibida.

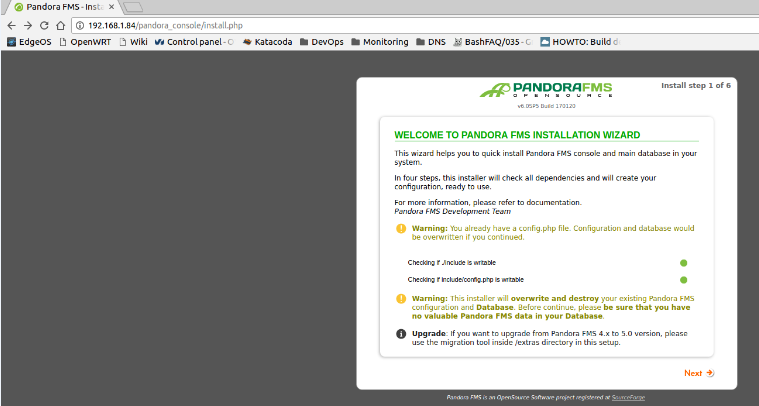
Tras solucionar este fallo, hemos podido utilizar con toda normalidad las funcionalidades de PandoraFMS.

## 5.4 Documentación de instalación

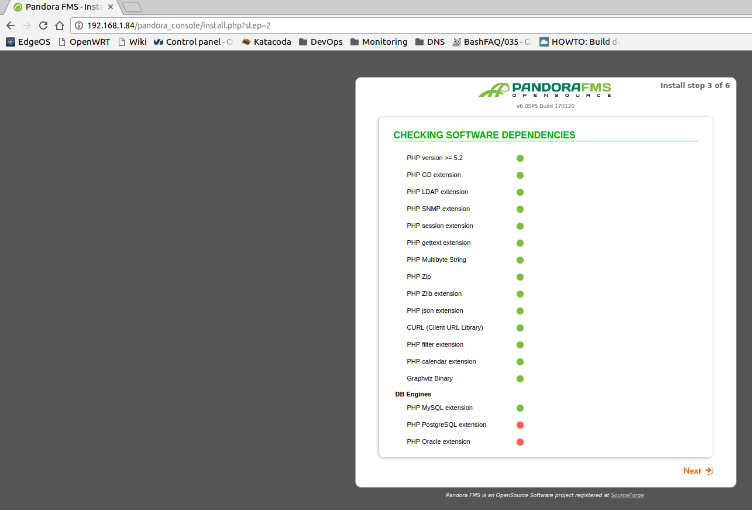
Se realiza la instalación de Pandora siguiendo los siguientes pasos:



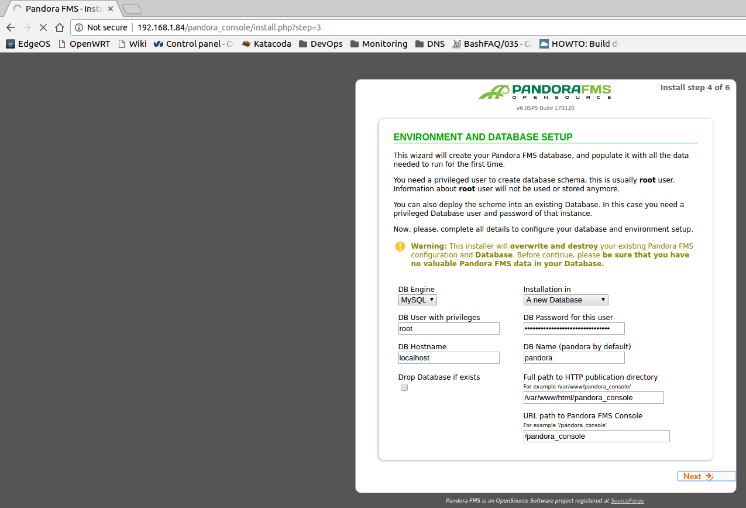
Instalamos la consola de Pandora:



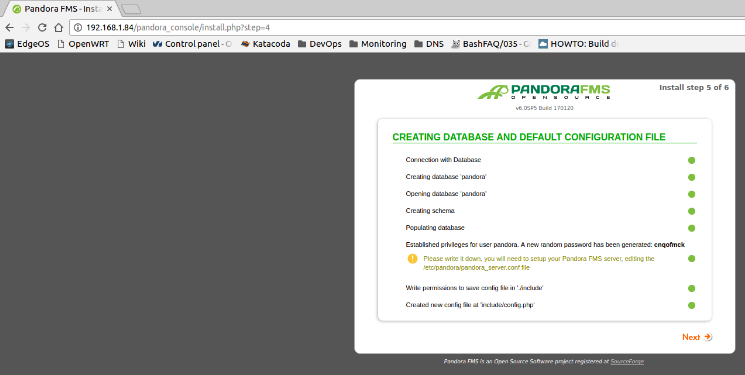
Comprobamos que el estado de las dependencias del software es correcto:



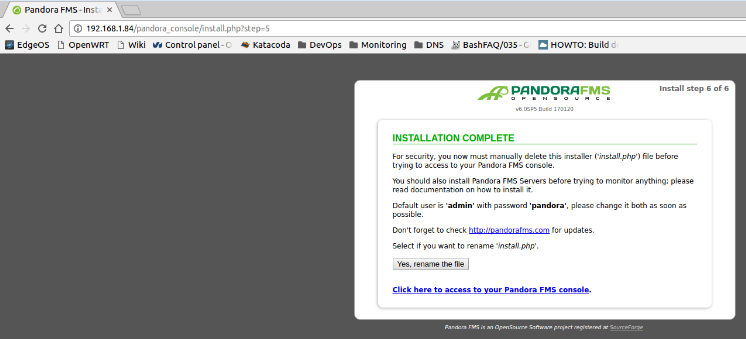
Preparación del entorno y la BBDD de Pandora:



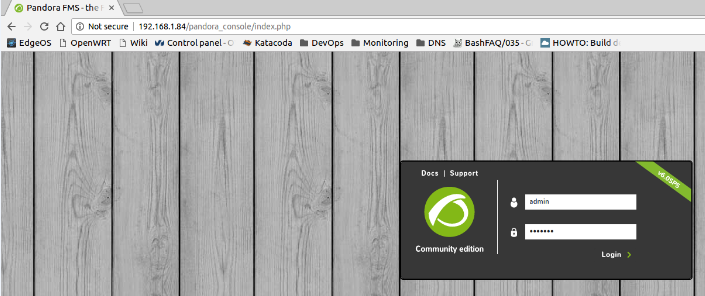
La creación de la BBDD se ha realizado correctamente:



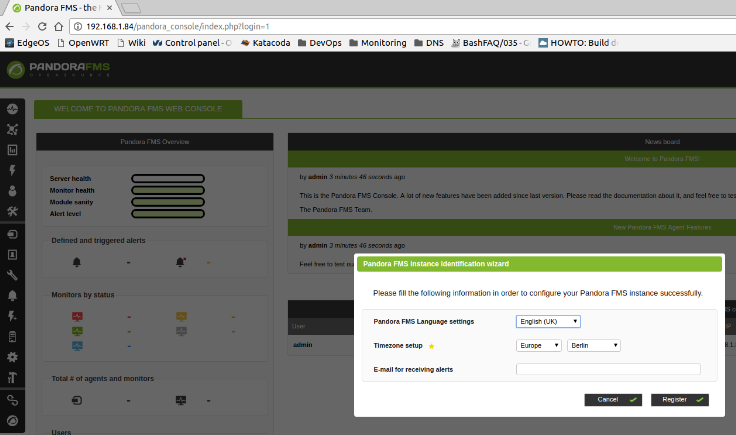
Instalación completada:



Tras completar la instalación introducimos por primera vez las credenciales de acceso:



Y, por último, introducimos el email para recibir alertas y seleccionamos el idioma en que vamos a trabajar:



## 5.5 Manual de usuario

Una vez tenemos la consola y el agente instalado. Comprobamos el archivo pandora\_agent.conf para revisar si la configuración es correcta:

# Base config file for Pandora FMS agents

# Version 6.0SP5, GNU/Linux

# Licensed under GPL license v2,

# Copyright (c) 2003-2014 Artica Soluciones Tecnologicas

# http://www.pandorafms.com

# General Parameters

# ==================

server\_ip 192.168.1.84

server\_path /var/spool/pandora/data\_in

temporal /tmp

logfile /var/log/pandora/pandora\_agent.log

#include /etc/pandora/pandora\_agent\_alt.conf

#broker\_agent name\_agent

# Interval in seconds, 300 by default

interval 300

# Group assigned for this agent (descriptive, p.e: Servers)

group Servers

# Module Definition

# =================

# System information

# vmstat syntax depends on linux distro and vmstat command version, please check before use it

module\_begin

module\_name CPU Load

module\_type generic\_data

module\_interval 1

module\_exec vmstat 1 2 | tail -1 | awk '{ print $13 }'

module\_max 100

module\_min 0

module\_description User CPU Usage (%)

module\_min\_warning 70

module\_max\_warning 90

module\_min\_critical 91

module\_max\_critical 100

module\_unit %

module\_end

# vmstat syntax depends on linux distro and vmstat command version, please check before use it

module\_begin

module\_name CPU IOWait

module\_type generic\_data

module\_interval 1

module\_exec vmstat 1 2 | tail -1 | awk '{ print $16 }'

module\_min\_warning 10

module\_min\_critical 16

module\_unit %

module\_end

#Get load average

module\_begin

module\_name Load Average

module\_type generic\_data

module\_exec cat /proc/loadavg | cut -d' ' -f1

module\_description Average process in CPU (Last minute)

module\_end

#IO Wait CPU ticks /sec

module\_begin

module\_name IOWaitCPU

module\_type generic\_data\_inc

module\_exec vmstat -s | grep "IO-wait cpu ticks" | awk '{ print $1 }'

module\_unit ticks/sec

module\_description Too much IOwait means IO bottleneck and performance problems. Check also LoadAVG.

module\_end

#Connected users

module\_begin

module\_name Connected users

module\_type generic\_data

module\_exec who | wc -l

module\_end

#Count total number of processes

module\_begin

module\_name Number processes

module\_type generic\_data

module\_exec ps aux | wc -l

module\_description Total processes

module\_unit processes

module\_end

# Async data example

module\_begin

module\_name LastLogin

module\_type async\_string

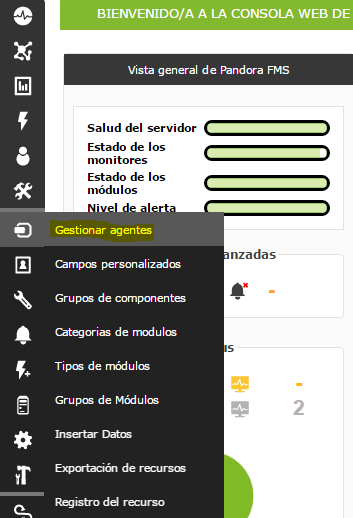
module\_exec last | head -1

module\_description Monitor last user loggin

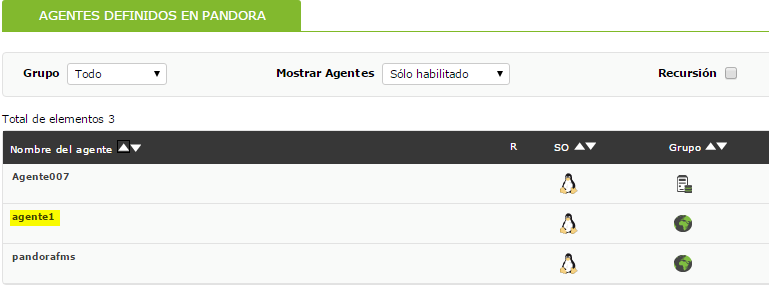
module\_end

RF-007: Gestor de agentes de software

Comprobamos que la configuración es correcta y revisamos que la consola reconoce la máquina que queremos monitorizar. Hacemos click en Gestionar agentes.

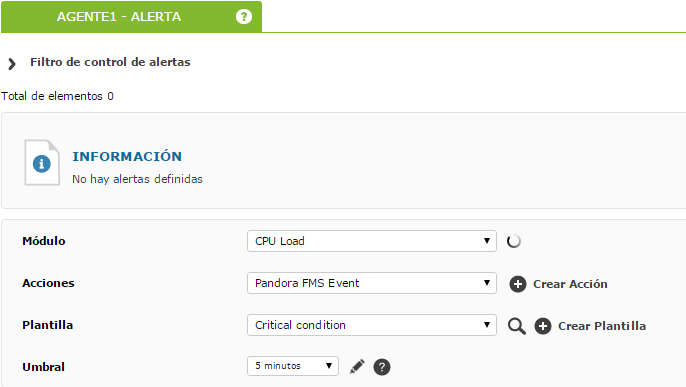


Comprobamos que nuestra máquina ha sido reconocida por la consola.

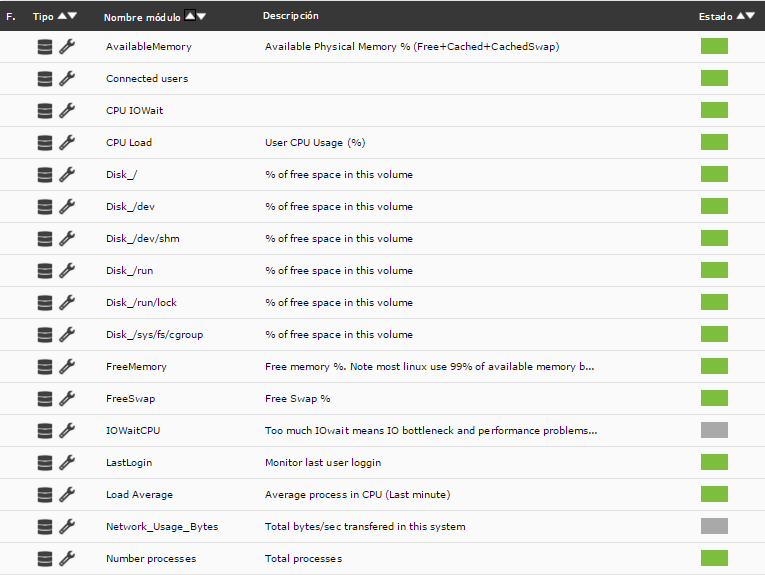


RF-005: Alertas

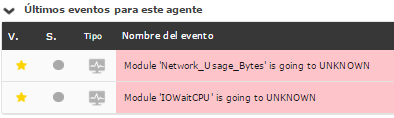
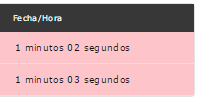
Añadimos al servidor todas las alertas (CPU, memoria, espacio en disco…). Por ejemplo, añadimos la alerta de CPU.



Estos son los módulos de alertas que tiene asignados el agente1.



Como podemos comprobar han salido alertas de CPU y red. Estos son los eventos:

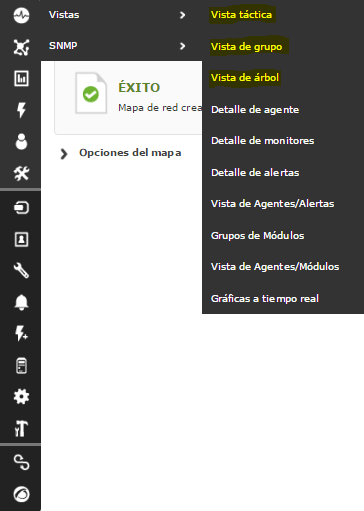


RF-001: Mapa de red

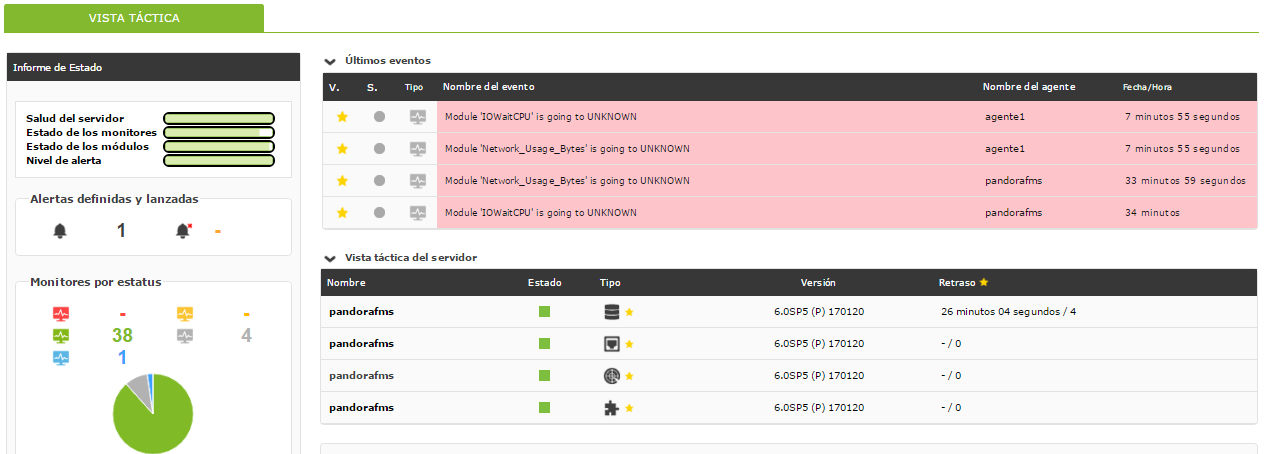
El mapa se ha mostrado en el apartado 5.1 y nos sirve para visualizar los equipos conectados al servidor central de Pandora.

RF-002: Vistas

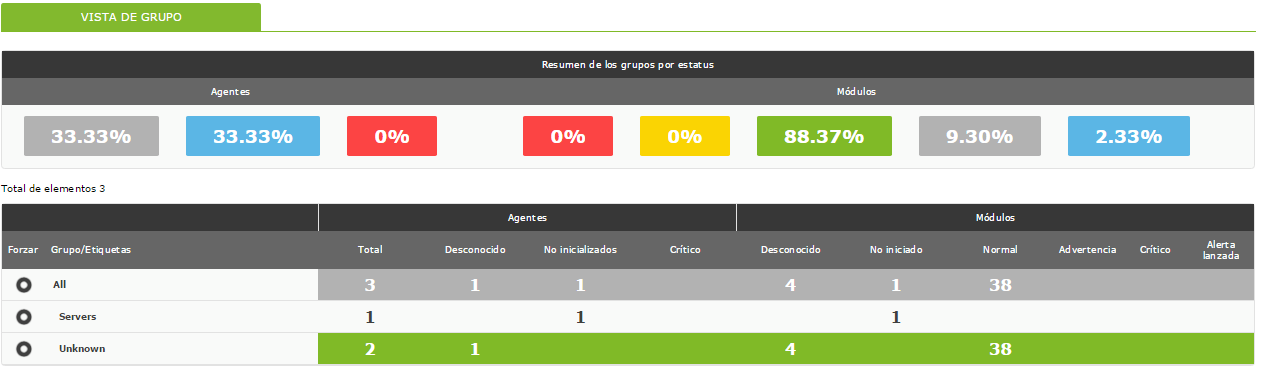
Las vistas son un requisito funcional en las que se permite al usuario tener una visión general de toda la monitorización. Estas son las vistas que nos ofrece Pandora:



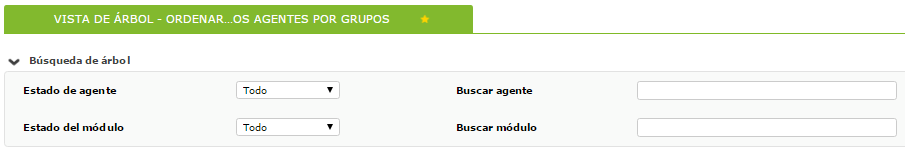
Vista táctica:



Vista de grupo:

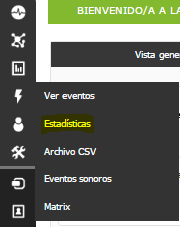


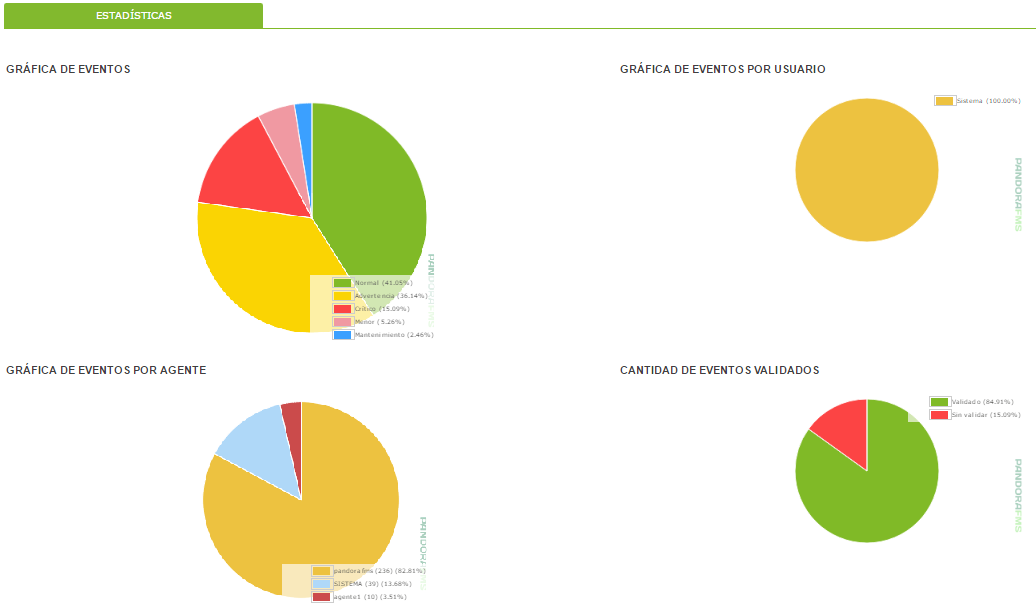
Vista de árbol:



RF-003: Estadísticas

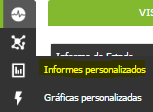
Con Pandora puedes consultar todas las estadísticas relacionadas con la monitorización de los sistemas.



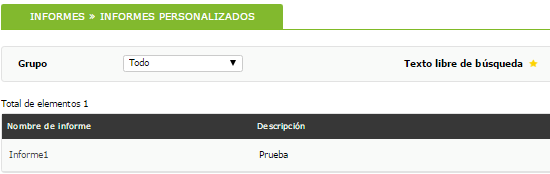


RF-004: Informes

Para consultar y crear informes personalizados:

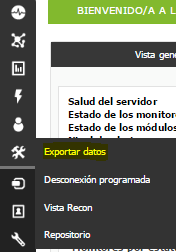


Hemos creado un informe de prueba:



RF-006: Exportar datos

El último de los requisitos funcionales es la exportación de datos.





# 6. Comparación de las dos implementaciones

En este apartado procedemos a analizar una evaluación exhaustiva de cada una de las tecnologías adecuándonos a los criterios establecidos en los apartados anteriores de manera que podamos obtener datos claros y concisos, que nos permitan poder decantarnos por una tecnología u otra.

## 6.1 Evaluación de los criterios en la implementación de Icinga

|  |  |
| --- | --- |
| CRITERIO | EVALUACIÓN |
| Criterio 1: Tiempo empleado para el desarrollo | 30 min. |
| Criterio 2: Peso del programa | 35 MB |
| Criterio 3: Tiempo de arranque de tecnologías | 3 segundos |
| Criterio 4: Recursos necesarios de CPU | 0.1 % |
| Criterio 5: Recursos necesarios de Memoria RAM | 20 MB |
| Criterio 6: Precio | Gratuito |
| Criterio 7: Multiplataforma | 2 |
| Criterio 8: Facilidad de aprendizaje | 6 |
| Criterio 9: Almacenamiento | 20 MB |
| Criterio 10: Actualizaciones | 1 |

## 6.2 Evaluación de los criterios en la implementación de PandoraFMS

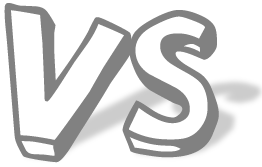
|  |  |
| --- | --- |
| CRITERIO | EVALUACIÓN |
| Criterio 1: Tiempo empleado para el desarrollo | 60 min. |
| Criterio 2: Peso del programa | 60 MB |
| Criterio 3: Tiempo de arranque de tecnologías | 4 segundos |
| Criterio 4: Recursos necesarios de CPU | 0.5 % |
| Criterio 5: Recursos necesarios de Memoria RAM | 200 MB |
| Criterio 6: Precio | Gratuito |
| Criterio 7: Multiplataforma | 3 |
| Criterio 8: Facilidad de aprendizaje | 3 |
| Criterio 9: Almacenamiento | 34 MB |
| Criterio 10: Actualizaciones | 6 |

# 7. Comparación de la implementación de las tecnologías

En este apartado vamos a proceder a realizar una profunda comparación en relación a los criterios descritos en el tercer apartado y obviamente a su posterior evaluación. Para ello hemos desarrollado la siguiente tabla, de acuerdo con la plantilla establecida por el profesor en el enunciado de la práctica.

Dicha tabla cuenta con todos y cada uno de los criterios establecidos anteriormente, con los datos sobre cada tecnología y con los comentarios pertinentes en caso de que la comparación no quedase bien definida, o hubiese cualquier aspecto que no quedase del todo claro.

Además, hemos añadido un apartado adicional a modo de conclusión en el que se detalla más concretamente que es lo que hemos sacado en claro a la hora de haber elaborado esta comparación.



|  |  |  |  |
| --- | --- | --- | --- |
| RITERIO | ICINGA | PANDORAFMS | COMENTARIOS |
| Criterio 1: Tiempo empleado para el desarrollo | 30 min. | 60 min. | Para el desarrollo de Icinga hemos necesitado la mitad de tiempo que para el desarrollo de PandoraFMS. |
| Criterio 2: Peso del programa | 35 MB | 60 MB | Icinga pesa menos que Pandora, ya que tiene una interfaz más simple. |
| Criterio 3: Tiempo de arranque de tecnologías | 3 segundos | 4 segundos | Ambas tecnologías tienen un arranque rápido y similar. |
| Criterio 4: Recursos necesarios de CPU | 0.1 % | 0.5 % | Pandora consume 5 veces más recursos de CPU que Icinga. |
| Criterio 5: Recursos necesarios de Memoria RAM | 20 MB | 200 MB | Icinga consume muchos menos recursos de memoria que Pandora. |
| Criterio 6: Precio | Gratuito | Gratuito | Hemos utilizado la versión gratuita de ambas tecnologías. |
| Criterio 7: Multiplataforma | 2 | 3 | Icinga está disponible para Windows y Unix. Mientras que Pandora también está disponible para FreeBSD. |
| Criterio 8: Facilidad de aprendizaje | 6 | 3 | Icinga es más complejo de aprender, pero ofrece funcionalidades más expertas. |
| Criterio 9: Almacenamiento | 20 MB | 34 MB | Pandora ocupa más espacio en disco que Icinga, pero actualmente el espacio de estos agentes son insignificantes. |
| Criterio 10: Actualizaciones | 1 | 6 | Icinga no ha tenido actualizaciones, mientras que Pandora está actualmente por la versión 6. |

# 8. Conclusiones

Los sistemas de monitorización son fundamentales a día de hoy para la continua supervisión de los sistemas, sobre todo en aquellos que son críticos para un servicio clave en una empresa.

Cabe destacar que ambas tecnologías son de las más prestigiosas del momento, en lo que a monitorización se refiere, debido a que las dos son muy completas y versátiles, y cubren las necesidades que cualquier empresa puede necesitar.

En varios puntos Icinga es superior a PandoraFMS, lo podemos dividir en secciones: instalación, aprendizaje/implementación y versatilidad.

* Instalación

La instalación de Icinga es bastante sencilla, tienes distintos métodos a la hora de instalarlo: puedes compilar el código fuente o instalarlo mediante los repositorios de la distribución GNU/Linux que uses. La instalación es automática, por ejemplo, en Debian/Ubuntu: apt-get install icinga. Y ya tienes instalado Icinga (con todas sus dependencias, por ejemplo: MySQL Server).

Por otro lado, PandoraFMS tienes que preparar todas las dependencias previamente antes de instalar los paquetes. En Debian/Ubuntu tienes que descargarlos e instalarlos en un orden en concreto:

1º. Core / Consola

2º. Servidor

3º. Agente

La instalación es manual, ya que no existen repositorios de PandoraFMS ni empaquetadores de terceros para hacerlo más fácil. Como conclusión, Icinga en este punto tiene una ventaja sobre PandoraFMS.

* Aprendizaje/Uso/Implementación

En PandoraFMS tienes una lista de módulos preparados ya en la página Web (hechos por la propia comunidad), pero crearlos tú mismo hace que sea todo mucho más complicado. Por ejemplo:

module\_begin

module\_name proctotal

module\_type generic\_data

module\_exec ps -A | tail --lines=+5 | wc -l

module\_description Total processes

module\_end

En cambio, en GNU/Linux puedes escribir en cualquier lenguaje (Perl, Bash, C, Java, Python...) el script (o alerta) y luego Icinga lo ejecuta. En Icinga, también encontramos scripts predefinidos:

check\_http -> chequea conexiones http.

check\_tcp -> chequea conexiones tcp.

Además, la comunidad es bastante más extensa para Icinga que para PandoraFMS en cuanto a scripts de monitorización (Github, por ejemplo, tiene librerías completas).

En cuanto aprendizaje, Icinga, tiene una documentación bastante más extensa y ejemplificada que PandoraFMS, lo que facilita el uso de Icinga. Lo único complicado de Icinga es que tienes que estar familiarizado con el entorno GNU/Linux ya que todo se configura mediante una shell, mientras que en PandoraFMS la configuración se hace mediante Web.

En cuanto al uso Icinga, tiene una interfaz bastante friendly y sencilla. PandoraFMS tiene bastantes submenús y puede llegar a ser un poco lioso encontrar la opción que quieres.

* Versatilidad

Hacer funcionar a Icinga y a PandoraFMS, tiene varias formas:

1º. Usar un agente nrpe

2º. Conexiones mediante ssh (ejecuciones remotas)

Icinga se adapta a cualquier Sistema Operativo cliente (Windows, GNU/Linux), pero el fallo que tiene es que no es compatible con Windows si queremos instalar Icinga en Windows Server. En cambio, PandoraFMS también se adapta a cualquier Sistema Operativo cliente y, además, se puede instalar en un Windows Server.

En cuanto a versatilidad, PandoraFMS tiene una ventaja si el cliente tiene distintos entornos (Windows y GNU/Linux) en su empresa. Pero si todo corriera bajo GNU/Linux, Icinga sería la opción a elegir.