# Implementación de un servidor para el análisis y visualización del estado de la red de la EPSEVG

Marc Ramiro Ramos

Escola Politècnica Superior d'Enginyeria de Vilanova i la Geltrú, UPC – EPSEVG

Ingeniería informática

## Resumen

En este proyecto se pretende crear una herramienta de monitorización de redes de área local para su uso en la EPSEVG. Esta, debe ser capaz de generar estadísticas de uso de los equipos detectados, recibir alertas vía email y visualizar en cualquier momento, los equipos que se encuentran conectados a las redes monitorizadas. Para lograr estos objetivos, se ha desarrollado una aplicación web que permite crear sistemas de escaneado personalizables. Gracias a este sistema, se consigue monitorizar la red de aulas informáticas de la escuela en los distintos horarios establecidos.

Dentro de la aplicación, se ha implementado un sistema de clasificación de equipos, permitiendo agrupar los distintos dispositivos detectados en grupos para su mejor gestión. Con este sistema de clasificación, se ha podido proceder a la creación de un sistema de generación de estadísticas por grupos, permitiendo analizar el uso que se les da a las máquinas de las distintas aulas informáticas de la escuela, analizando previamente el uso individual de cada una de ellas.

Con la información recopilada por los distintos sistemas de escaneado, se ha desarrollado un sistema de alertas que permite a los administradores recibir notificaciones por correo.

Para que los administradores puedan interactuar con la aplicación, se han desarrollado dos interfaces distintas. La interfaz por defecto cuenta con un diseño minimalista y amigable que permite localizar de forma rápida los sistemas de escaneado configurados y todos los equipos detectados por ellos. La interfaz de administración cuenta con acceso a todas las configuraciones y datos recopilados por la aplicación. Desde esta última, los administradores pueden exprimir al máximo la aplicación.

Para que todo lo anterior haya sido posible, la aplicación web se ha integrado en un servidor dedicado dentro de la EPSEVG. Durante el proyecto se han analizado que componentes se requerían para que la aplicación pudiese ser funcional. Cuándo estos han sido seleccionados, se ha procedido a la integración de los mismos en el servidor, configurándolos para que trabajasen con la aplicación desarrollada.

## 1. Introducción

Actualmente existe una gran cantidad de dispositivos distintos, capaces de acceder a las diferentes redes. Esto provoca que la administración de las redes, hoy en día, no sea para nada trivial. El dinamismo en la red producido por toda esta serie de dispositivos, hace que los administradores de redes, deban contar con diversas herramientas capaces de facilitarles la gestión, ayudándoles a mantenerlas bajo control y previniéndolas de los posibles peligros que pudiesen surgir. Además algunas de estas herramientas les permiten visualizar el estado de todos los sistemas de la red en cualquier momento.

Una de las tareas que llevan a cabo los administradores de redes, se trata del rastreo continuo de los dispositivos que se encuentran en ellas. A esta tarea se la conoce como monitorización. Mediante un procesado de los datos analizados en la red, se puede extraer cierta información que proporciona conocimiento, sobre los sistemas que se encuentran en ese momento conectados. Además de conocer que sistemas se encuentran presentes, se puede obtener información detallada de cada uno de ellos. Por ejemplo, es posible conocer que servicios se están ejecutando en un equipo e incluso que sistema operativo se está usando. Según la información recopilada de los dispositivos, los administradores pueden tomar decisiones consecuentes al uso.

El propósito de este trabajo de fin de grado, es el de desarrollar una de estas herramientas de monitorización y su despliegue en un servidor dedicado, con la configuración adecuada. Esta herramienta, estará dotada de varias funcionalidades que permitirán a los administradores de las redes, conocer el uso que se les da a los equipos monitorizados e incluso recibir alertas, informando que sistemas se encuentran activos en un rango determinado de horas.

## 2. Objetivos

Los objetivos fijados para el alcance del proyecto se pueden englobar en dos bloques:

- Desarrollo de una aplicación web minimalista y amigable que conste de las siguientes funcionalidades:
    - Creación de sistemas de escaneado personalizables, permitiendo al administrador definir la red a monitorizar, mediante el

formato CIDR, y la frecuencia de ejecución del escaneado.

- Detección de los equipos conectados a las redes monitorizadas, utilizando el protocolo ARP.
- Detección del sistema operativo usado por los equipos previamente detectados, mediante el análisis de los servicios en ejecución en cada uno de ellos.
- Visionado en tiempo real de los equipos pertenecientes a las redes monitorizadas, desde una sección dedicada en la interfaz web.
- Clasificación de los equipos en grupos, permitiendo organizar los equipos de forma personalizada.
- Cálculo de estadísticas de uso de los equipos en las redes monitorizadas, permitiendo escoger de que grupos de equipos se quiere analizar el uso.
- Creación de dos tipos de alertas las cuales se notificarán vía email:
  - Detección de equipos activos en un rango de horas, permitiendo al administrador determinar el rango y en qué momento del día se va a recibir la notificación.
  - Detección de equipos inactivos, permitiendo al administrador determinar el número de días que debe estar un equipo sin detectarse, para que se adjunte a la notificación, y en qué momento del día se va a recibir el aviso.

- Despliegue de la herramienta en un servidor dedicado dentro de la universidad.
  - Análisis de los componentes requeridos
  - Integración de la aplicación web
  - Configuración de los componentes seleccionados

# 3. Análisis

Debido a la naturaleza de la herramienta desarrollada a lo largo del proyecto, es indispensable analizar las necesidades a tener en cuenta para su correcto funcionamiento.

### 3.1 Sistema Operativo

Dado que la aplicación va a estar viviendo dentro de una máquina virtual creada exclusivamente para ella en la EPSEVG; primeramente se debe escoger qué sistema operativo será el encargado de gestionar los recursos que se le proporcionarán.

El hecho de ser usado en la universidad, implica que cuanto menos impacto económico tenga hacia ella mejor, por lo que se descarta el uso de software propietario. Por lo tanto el sistema operativo que se usará, debe ser de software libre y gratuito.

El sistema operativo debe ser lo más liviano posible, con el fin de consumir los recursos únicamente indispensables para su funcionamiento. Esto implica prescindir de la típica interfaz gráfica que incluyen hoy en día la mayoría de sistemas operativos y otros servicios de los que no se va a hacer uso.

Finalmente, el sistema operativo a usar, debe contar con un buen soporte por parte de la comunidad y de sus desarrolladores, manteniéndolo al día frente a posibles amenazas.

Se escoge Debian [1] como sistema operativo.

### 3.2 Lenguaje de programación interpretado en el back-end

Hoy en día, la inmensa mayoría de aplicaciones web son dinámicas, esto quiere decir, que se permite al usuario interactuar con ella ofreciendo una experiencia más satisfactoria. Así mismo, la aplicación web planteada para este proyecto ofrecerá estas características. Para desempeñar esta tarea no basta con usar HTML y CSS, pues estos están muy limitados; se tienen que combinar con otro tipo de lenguajes de programación. Estos lenguajes son interpretados y ejecutados directamente tanto en el navegador del usuario (JavaScript), como en el servidor (Python, PHP).

En la aplicación web de este proyecto, la mayor parte del trabajo se va a desempeñar en el lado del servidor. Para la selección del lenguaje a utilizar, se tendrá en cuenta los siguientes aspectos:

- Popularidad.
- Usabilidad.
- Facilidad de aprendizaje.
- Rapidez de ejecución.

Tras comparar los tres lenguajes más usados, se selecciona Python [2], en concreto su versión 3.4.2, como lenguaje de programación por su fácil curva de aprendizaje y su rapidez de ejecución.

### 3.3 Framework

Existen una serie de tareas realizadas en el desarrollo web, las cuales se repiten constantemente. Un framework permite aligerar la carga de trabajo del desarrollador, permitiendo abstraerle de esas tareas asociadas al desarrollo web. Por ejemplo, la mayoría de los frameworks disponibles cuentan con librerías para el acceso a la base de datos, motores para las plantillas y cuentan con un sistema de gestión de sesiones. Además, promueven la reusabilidad del código, permitiendo al desarrollador no repetirse para cada aplicación que desarrolle.

Para este proyecto se pretende usar un framework que siga el patrón de desarrollo MVC. Este patrón separa el modelo de datos de la lógica de negocio y de la interfaz de usuario. Esto permite programar de forma modular, promoviendo el principio DRY, don’t repeat yourself. Django [3] es el framework usado en este proyecto.

### 3.4 Sistema de gestión de bases de datos

Debido a la naturaleza de la herramienta, es necesario almacenar grandes cantidades de datos con el fin de poder analizarlos y ofrecer información útil sobre ellos. Dicho esto, queda claro que es necesario almacenar los datos en

una base de datos, ya que no es viable almacenarlos en ficheros planos.

Dada la naturaleza de los datos que se obtendrán, se hace uso de un sistema gestor de bases de datos relacional; dejando de lado las bases de datos no relacionales. Como ocurre con el resto de necesidades, es importante que el sistema utilizado sea gratuito. Siguiendo las recomendaciones de Django, el SGDB que funciona más eficientemente con él, se trata de PostgreSQL [4].

### 3.5 Servidor web

Ya que la aplicación que se desarrolla es una aplicación web, se debe poder recibir peticiones HTTP de parte de los clientes que quieran hacer uso de ella. Para eso es necesario disponer de un software que se encargue de gestionar las conexiones entre el cliente y el servidor, aceptando o rechazando las peticiones HTTP, entregando las páginas web solicitadas, etc.

Puesto que el uso que se le va a dar a esta aplicación, a nivel de conexiones recibidas, será muy limitado (únicamente los administradores de la EPSEVG dispondrán de acceso a la herramienta), se busca un servidor web ligero y a la vez rápido.

Nginx [5] es el servidor web usado.

### 3.6 Aplicaciones de escaneado

Una parte de la aplicación se dedica a la recopilación de información sobre los dispositivos que se encuentran conectados a las redes monitorizadas. A continuación se especifica que información se desea obtener:

- Detección de los dispositivos activos y conectados a la red. Ya que la aplicación está pensada para monitorizar redes de área local, en las que debe estar presente el servidor, la detección de los equipos se realizará mediante peticiones ARP.
- Detección del sistema operativo. La detección del sistema operativo se hará en base a las direcciones IP detectadas en el escaneado anterior. Se analizará una serie de servicios característicos de los sistemas operativos más conocidos, con tal de identificarlos.
- Detección del nombre de los dispositivos. Mediante consultas al DNS local, se obtendrán los nombres de los dispositivos de la red.

Nmap [6] será el encargado de realizar las detecciones.

### 3.7 Arquitectura del sistema

A continuación, en la Figura 1 se muestra un esquema indicando cómo están relacionados los distintos componentes definidos anteriormente, excluyendo Nmap pues no afecta al funcionamiento del servidor. Se puede observar como Nginx es el encargado de ofrecer visibilidad del servidor hacia el exterior, pudiendo encontrarse las redes locales e internet. PostgreSQL podría estar situado tanto dentro como fuera del servidor puesto que ofrece la posibilidad de conexión remota; en este proyecto se encuentra en el interior del servidor.

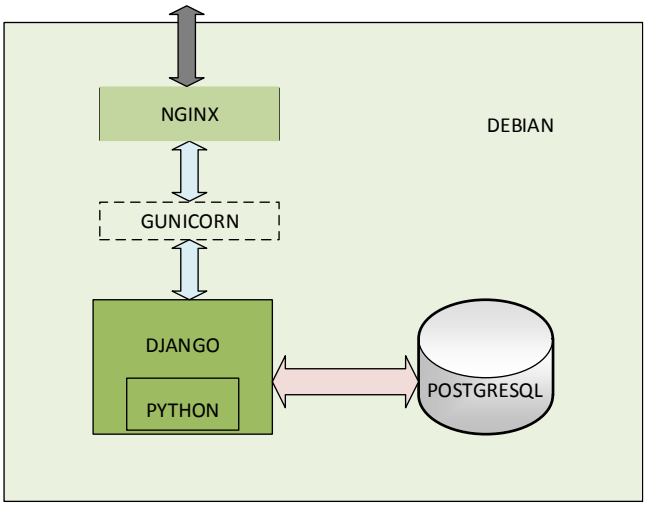


***Fig 1.*** *Arquitectura del sistema*

## 4. Diseño de la interfaz web

Dada la naturaleza de la aplicación desarrollada, el administrador interactúa con ella mediante dos interfaces web propiamente desarrolladas. A continuación se detallará cuáles son estas interfaces, qué información se muestra en cada una de ellas y cuál es su propósito.

### 4.1 Tipos de interfaz

La aplicación web consta de dos interfaces distintas, pero a la vez relacionadas entre ellas. La primera consta de un diseño minimalista y amigable, que permite tener una visión muy rápida, sobre los escáneres configurados y sobre el estado de todos los equipos detectados. Esta es la interfaz que se observa por defecto al acceder a la aplicación. En el desarrollo de esta interfaz, se ha usado Bootstrap [7] para poder ofrecer un diseño responsive, pudiéndose adaptar a todo tipo de pantallas.

La segunda interfaz cuenta con un diseño más complejo, para que el administrador pueda interactuar con todos los módulos desarrollados. Es en esta interfaz, dónde se puede exprimir a fondo la aplicación, permitiendo a los administradores disponer de todos los datos y configuraciones generadas para la aplicación web. Esta interfaz cuenta con las opciones de ordenación, búsqueda y filtrado ofreciendo una experiencia más practica al administrador.

### 4.2 Vistas de la interfaz minimalista

En este apartado se detallarán las vistas que ofrece la interfaz web minimalista desarrollada para este proyecto. Para cada una de ellas, se informarán de ciertos puntos a destacar.

#### 4.2.1 Página de inicio de sesión

Esta es la primera vista que se observa al acceder a la web. Para poder ofrecer privacidad de los datos recopilados, la aplicación consta de una pantalla de inicio de sesión. Ya que durante el desarrollo del proyecto, se han estado haciendo pruebas en remoto con el servidor ubicado en la EPSEVG, se ha implementado este sistema de login para poder proteger la información interna de la red de la escuela.

### 4.2.2 Página principal

Cuando se ha realizado con éxito el login, frente al administrador, aparecerá una pantalla con todos los escáneres configurados hasta ese momento. Esta vista se analizará siguiendo la numeración dispuesta en la Figura 2.

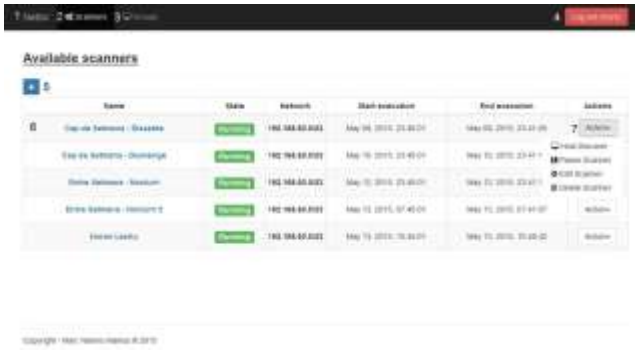


***Fig 2.*** *Página inicial*

1. **Acceso a la interfaz de administración.** Este enlace está situado en la barra de navegación, presente en toda la interfaz minimalista. Pulsando en *Nettor* se accede a la interfaz de administración.
2. **Acceso a la vista de escáneres disponibles.** Pulsando en *Scanners*, se accede a la pantalla principal, donde se muestra el listado de todos los escáneres disponibles.
3. **Acceso a la vista de grupos.** Pulsando en *Groups*, se accede a la pantalla donde se encuentran todos los grupos de equipos detectados por la aplicación.
4. **Logout de la aplicación.** Este botón muestra el usuario que está haciendo uso de la aplicación. Cuando se pulsa, se procede al cierre de sesión del usuario.
5. **Añadir nuevo escáner.** Este botón conduce al administrador a la vista de creación de escáneres disponible en la interfaz de administración.
6. **Tabla con los escáneres disponibles.** En esta tabla, se muestran por filas, todos los escáneres que se encuentran configurados en la aplicación. Se muestran los siguientes campos:
    a. **Name.** Nombre dado al escáner. Pulsando en el enlace, se redirige la vista a la interfaz de administración, para la modificación del escáner.
    b. **State.** Estado en el que se encuentra el escáner. Se dará más información en el próximo capítulo.
    c. **Network.** Red monitorizada por el escáner.
    d. **Start execution.** Inició de la última ejecución del escáner.
    e. **Ens execution.** Fin de la última ejecución del escáner.
7. **Acciones disponibles.** En este desplegable se muestra una serie de acciones disponibles para cada escáner. A continuación se indica el propósito de cada una de ellas:
    a. **Host discover.** Permite la ejecución de un único escaneo a la red definida.
    b. **Pause Scanner.** Esta acción cambia el estado del escáner a *Paused*.
    c. **Edit Scanner.** Redirige la vista a la interfaz de administración para la modificación del escáner.
    d. **Delete Scanner.** Elimina el escáner.

### 4.2.3 Página de grupos

En esta página, se muestra un listado de todos los grupos detectados en las redes escaneadas y los grupos creados manualmente. A continuación se describe el significado de cada uno de los campos:

1. **Name**. Nombre del grupo. Este puede ser en base a los equipos que pertenecen a él o simplemente es el nombre descriptivo que se le ha dado manualmente. El grupo *Unknown*, es el grupo por defecto, donde se añaden los equipos que no se han clasificado automáticamente. Pulsando en el nombre, se redirige página a la pantalla de equipos.
2. **Hosts up**. Este valor indica el número de equipos detectados por el último escaneo realizado.
3. **Hosts down**. Indica el número de equipos no detectados por el último escaneo realizado.
4. **Total Hosts**. Este valor indica la cantidad de sistemas que pertenecen a cada grupo.
5. **Statistiscs**. Este enlace, redirige la vista a la interfaz de administración, para el visionado de estadísticas diarias del grupo.

### 4.2.4 Página de equipos

En esta página, se muestra un listado de todos los equipos detectados, clasificados por su grupo, en las redes escaneadas. Esta página se analizará siguiendo la numeración dispuesta en la Figura 3.

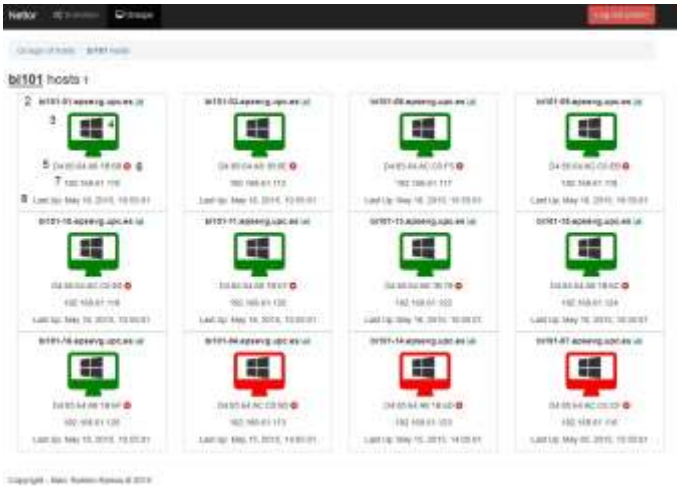


***Fig 3.*** *Página de equipos*

1. **Nombre del grupo** al que pertenecen los equipos mostrados.
2. **Nombre del equipo.** A su lado se dispone de un enlace a la vista de estadísticas diarias de este.

3. **Indicador del estado del equipo**. Si su color es verde, indica que el sistema se ha detectado durante ejecución del último escaneo. Por el contrario, si se encuentra rojo, indica que el sistema no ha sido detectado.
4. **Sistema operativo** que se ha detectado en ese equipo, durante último escaneado ejecutado dónde ha sido detectado.
5. **Dirección MAC** del equipo.
6. Botón para enviar el equipo a la **lista negra.**
7. **Dirección IP** del equipo, detectada durante el último escaneado ejecutado.
8. **Fecha de última detección** del equipo.

# 5. Implementación

Siguiendo la filosofía de Django, la aplicación web se ha implementado de forma modular. Así, se han desarrollado cuatro módulos diferenciados por las tareas que desempeñan.

### 5.1 Módulo de escaneado

La base del funcionamiento de la aplicación desarrollada en este proyecto, se encuentra en las detecciones expuestas en el punto 3.6. Para poder realizar estas detecciones, este módulo consta de dos partes.

La primera parte, ofrece al administrador la capacidad de programar escaneados personalizados. Esto quiere decir, que se le da la posibilidad de indicar qué red se desea monitorizar y cada cuanto tiempo se debe ejecutar el escaneado. La red especificada debe estar en el formato CIDR, permitiendo monitorizar grandes cantidades de equipos, sin tener que especificar sus direcciones IP una a una. El tiempo especificado, permite indicar que días de la semana y durante que rango de horas se desea monitorizar la red. Finalmente, para cada escáner creado, se ofrece la opción de registrar los datos obtenidos en un histórico, para su posterior análisis y generación de estadísticas; esto se detallará en los próximos apartados.

La segunda parte, consta de los procesos internos de la aplicación, encargados de ejecutar el escaneado cómo y cuándo toca. Para ello, hay un indicador del estado en que se encuentran. Existen tres estados disponibles:

- Running. Indica que se encuentra operativo.
- Paused. Indica que el escáner se encuentra pausado de forma indefinida.
- Stopped. Indica que se trata de un nuevo escáner, sin previa activación.

Cuándo un escáner se activa y por lo tanto pasa al estado *Running*, la aplicación añade una nueva entrada en el cron del servidor, con la configuración temporal del escáner introducida previamente. Una vez se cumple la condición del tiempo en el cron, este ejecuta una serie de scripts que interactúan con Nmap y el resto de los módulos, analizando la red definida en el escáner.

El primer script ejecutado, se encarga de detectar los dispositivos activos de la red en ese instante, almacenando por un lado la información de estos y por el otro, almacenando en un fichero, un listado de todas las IPs detectadas. Además, dentro de la misma ejecución de Nmap, se hacen las consultas inversas al DNS para obtener el nombre de los equipos.

El segundo script toma el fichero de IPs de entrada y ejecuta la detección del sistema operativo sobre ellas, almacenando el resultado en un fichero.

### 5.2 Módulo de clasificación

Como se ha indicado en el apartado anterior, al finalizar la ejecución de los scripts que interactúan con Nmap, se crean dos ficheros distintos. El primero de ellos incluye la información básica de los equipos, indicando su nombre (en el caso de que este registrado en el servidor DNS), dirección MAC y dirección IP. El segundo de ellos obvia la resolución inversa de nombres y únicamente contiene las direcciones MAC e IP y el sistema operativo.

Este módulo cuenta con dos propósitos. El primero de ellos es el de tratar toda esa información e incluirla en la base de datos. Para ello, se utiliza una herramienta llamada analizador sintáctico o parser, que permite crear una estructura de datos a partir de un fichero XML. Con esta estructura de datos, se coteja por el campo *hostname* para obtener los nombres de las máquinas. En base a este campo, mediante expresiones regulares propiamente creadas para la universidad, se extrae el nombre de las aulas de la escuela. Un ejemplo de resultado obtenido con el *hostname* ai109-03.epsevg.upc.es, sería el aula ai109. Todos los sistemas que no dispongan de un nombre contemplado por las expresiones regulares, se añaden a un grupo genérico llamado *Unknown*.

Una vez disponible toda esta información tratada, se consulta la base de datos y si no existe ningún registro para un equipo, identificándose por su dirección MAC, se añade. En caso de que exista un registro previo y aparezca entre los resultados, se actualiza su información. Para todos los sistemas que no han sido detectados, se les supone como inactivos, no conectados a la red. Una vez se dispone de la información de los sistemas operativos usados por los sistemas conectados a la red, esta se actualiza en la base de datos. Cabe destacar, que para cada sistema detectado, se mantiene la fecha en la que se ha detectado por última vez. Como se verá en el apartado del módulo de alertas, esta información será crucial para su funcionamiento.

El otro propósito de este módulo, es el de permitir al administrador organizar conjuntos de equipos de forma personalizada, pudiendo crear grupos de sistemas y añadiendo equipos a estos. Esto es útil a la hora de la generación de estadísticas, pues en cada grupo se puede especificar si se quieren obtener estadísticas de los equipos que pertenecen a él o no. Lo mismo ocurre con las alertas, permitiendo seleccionar de qué grupos de dispositivos se desea obtener notificaciones.

Dicho todo esto, también se cuenta con la capacidad de personalizar los nombres de los equipos detectados y como ya se ha mencionado, el grupo al que pertenecen. Para que esta información introducida manualmente no sea sobrescrita por la aplicación en cada escaneado, se ha añadido una opción de bloqueo para que se mantenga. Además, este módulo ofrece la posibilidad de añadir sistemas a una lista negra. Con esto se consigue mantener

un registro de los equipos no deseados y a la vez estos no interfieren con la aplicación. Si los equipos no deseados se eliminaran directamente, en futuros escaneados podrían volver a surgir, pudiendo entorpecer el trabajo de los administradores.

### 5.3 Módulo de estadísticas

Este módulo se encarga de recopilar la información obtenida de los escaneos, con el fin de generar estadísticas tanto para los equipos, como para los grupos de equipos. Para ello, se hace uso de los escáneres habilitados para recopilar estadísticas. Cuando se ejecuta un escáner de este tipo y finaliza la ejecución del módulo de clasificación, es cuando empieza a trabajar este módulo. Se ofrece la posibilidad de realizar extracciones de datos con el formato CSV, permitiendo al administrador importar los datos a hojas de Excel y poder interactuar con ellos.

Con tal de dar veracidad a las estadísticas generadas, ha sido necesario realizar un estudio sobre el tiempo medio de duración de un escaneo. Se han realizado varios escaneos durante varios días, en distintos momentos, dónde la congestión de la red iba variando. Después de varias pruebas, se ha calculado que el tiempo medio de escaneado es de 3 minutos. Para asegurar su correcta ejecución, se ha establecido un tiempo mínimo entre escaneos de 5 minutos, obteniendo un número de muestras por hora y equipo de 12. Además, se ha establecido que el tiempo máximo entre escaneos sea de 20 minutos, en cuyo caso, el número de muestras por hora y equipo sería de 3. Como se entiende, como más muestras se disponga de un equipo, mayor veracidad tendrán las estadísticas calculadas para este.

Por un lado contamos con la tabla *Historic_Hosts*. Para todos los equipos que pertenezcan a un grupo con la opción de estadísticas habilitada y que no se encuentren en la lista negra; se crea un registro (una muestra) en este histórico. Aquí, se detalla toda la información de los sistemas (nombre, IP, MAC, sistema operativo, estado y grupo al que pertenece) y el momento exacto en el que se ha añadido. En base a la información de este histórico, para cada equipo, se calcula la diferencia de tiempo entre escaneados. Esta información calculada, se almacena en la tabla de estadísticas encargada de mantener un registro de los equipos por horas. En esta tabla se indica el número de minutos que han sido detectados en cada una de ellas. Se puede observar que en la tabla aparecen porcentajes. En el caso de esta, el porcentaje de actividad, se calcula sobre los sesenta minutos que dispone una hora y el número de minutos que ha sido detectado el equipo. El porcentaje de uso del sistema operativo, se calcula sobre el número de minutos en cada uno de ellos sobre el tiempo total de actividad de una máquina.

El siguiente paso es el registro de entradas en la tabla de grupos por hora. Acumulando los valores de los equipos de un mismo grupo y una misma hora, se obtiene el uso total de un grupo durante una hora.

El resto de tablas son esencialmente datos acumulados de sus predecesoras. Por ejemplo, la tabla de equipos por día, contiene un acumulado de toda la información disponible en la tabla de equipos por hora, tomando como referencia el día en cuestión. Lo mismo ocurre con la tabla de equipos por semana que contiene un acumulado, de toda la información disponible de equipos por día, tomando como referencia la semana en cuestión.

Se dispone de una tabla de configuración dónde los administradores deben configurar el tiempo que se ejecuta el escáner de estadísticas. Estos valores introducidos se usan en el momento de calcular los porcentajes que se ven en las otras tablas

### 5.4 Módulo de alertas

La finalidad de este último módulo, es la de ofrecer la posibilidad de configurar dos tipos de alertas distintos y enviar notificaciones vía e-mail, a los administradores registrados en la aplicación.

Este módulo, solo se ejecuta con los escáneres definidos sin la opción de estadísticas habilitada. Esto es así para poder reducir la carga de trabajo del servidor y poder separar funcionalidades distintas.

#### 5.4.1 Alerta de equipos inactivos durante x días

El propósito de esta alerta es el de informar a los administradores, de la existencia de equipos no detectados por más o igual a un número de días establecido. Su utilidad reside en poder aportar proactividad a los administradores, notificándoles de posibles averías en algunos equipos de la red. Así, se consigue mejorar el servicio ofrecido, pudiéndose adelantar a las posibles quejas por parte de los usuarios, sobre el mal estado de los equipos de la escuela.

Para desempeñar esta tarea, se han tenido que dividir las funcionalidades. Por un lado se cuenta con la creación y configuración de este tipo de alertas. En este punto, los administradores pueden introducir un nombre para identificar a la alerta, el número de días que debe permanecer un equipo sin detectarse en la red, los grupos de máquinas que se quieren mantener controlados, la hora a la que se quiere recibir la notificación diaria y la opción de activar o desactivar la alerta.

Cuando una alerta se encuentra activada, al final de la ejecución de los escáneres, se comprueba la fecha de ultima detección de los equipos; pertenecientes a los grupos seleccionados en la configuración. Si esta fecha es anterior al número de días especificados en la configuración, en base de la fecha de ejecución del escáner ejecutado, se añade un registro en del sistema. Con tal de mantener un registro de todas las detecciones que se han realizado, se ha creado una tabla dentro de la base de datos que contiene la información relacionada para cada sistema y la alerta activada.

#### 5.4.2 Alerta de equipos activos entre rango de horas

El propósito de esta alerta es el de informar a los administradores, de la existencia de equipos conectados a la red, dentro de un rango de horas establecido. Su utilidad reside en detectar equipos en horas en las que no debería encontrarse ninguna máquina encendida en la escuela. Además, con una buena organización de los equipos clasificados en grupos, se podría aprovechar para la detección de equipos ajenos a la escuela en horas fuera de lo habitual. Este último caso, podría significar la presencia de alguna maquina potencialmente peligrosa dentro del entorno de la escuela. Su creación es muy similar al otro tipo de alerta mostrada. Cada alerta creada dispone de un nombre que la identificará, el rango de horas a querer

revisar, los grupos que se quieren tener en cuenta, en que momento del día se desea obtener la notificación y si se quiere activar o no. Es importante mencionar, que en este tipo de alertas, la hora de envío de la notificación, debe estar dentro del rango de horas revisadas. El funcionamiento interno también es muy similar al anterior tipo de alerta. Cuando un escáner se está ejecutando dentro del rango de horas establecido por una alerta, los equipos detectados se registran en una tabla para tener constancia de ellos.

#### 5.4.3 Configuración del servidor de correos

Para que la aplicación sea capaz de enviar emails, es necesario especificarle que servidor y cuenta de correo se debe utilizar. Para ello, se ha añadido a la aplicación la capacidad de configurar un servidor de correo. Para el testeo de la aplicación, se ha creado una cuenta de Gmail encargada de enviar las notificaciones.

### 5.5 Módulo de autenticación

Django incorpora en su propio motor, un sistema de autenticación de usuarios. Para este proyecto se ha integrado este sistema. Con esto se consigue por un lado, lo mostrado en el capítulo anterior, ofreciendo una pantalla de login para poder acceder a la aplicación.

Este sistema, también es usado en la notificación de alertas por correo, pues sus destinatarios son los usuarios registrados en él. Por esto es importante que durante el registro de todos los administradores, se especifique una dirección de correo válida para poder recibir las alertas deseadas.

## 6. Integración

En este capítulo se documenta el proceso de integración de la herramienta desarrollada, en el servidor dedicado. Se documenta como se han configurado los distintos componentes vistos en el capítulo 3, para funcionar en harmonía con la aplicación. Todos los tecnicismos, tales como las instrucciones realizadas y las configuraciones, se encuentran en los anexos.

### 6.1 Pasos previos

Antes de empezar a configurar los diversos componentes, se deben efectuar una serie de acciones previas en el servidor.

- Creación del usuario *Nettor*, propietario de la aplicación web y de la base de datos. Se debe crear un usuario con los mínimos privilegios posibles en el sistema, por temas de seguridad.
- Revisión de los servidores DNS configurados. Si no están bien configurados, la aplicación no será capaz de realizar las consultas inversas a los DNS, para poder detectar los nombres de las máquinas.
- Revisión de las interfaces de red. Se debe comprobar que el servidor forme parte de las redes que se quieran monitorizar, estando estas presentes en el servidor en forma de interfaz.

### 6.2 Configuraciones

#### 6.2.1 Lenguaje de programación - Python

Se debe instalar la versión 3.4.2 de Python en el servidor. Una vez instalada, se debe crear un entorno virtual donde residirá la aplicación web.

#### 6.2.2 SGBD - PostgreSQL

La configuración de PostgreSQL es muy sencilla. Simplemente se debe crear una base de datos, nombrada *NettorDB*, la cual será la contenedora de toda la información de la aplicación. El usuario propietario de esta base de datos, debe ser *Nettor*.

#### 6.2.3 Herramienta de detección - Nmap

Para poder utilizar la detección de equipos mediante consultas ARP, se requieren privilegios de root. Para ello, se deberá crear una entrada en el fichero de *sudoers*, especificando que el usuario *Nettor* dispone de permisos totales para ejecutar Nmap, sin solicitud de contraseña.

#### 6.2.4 Servidor web – Nginx

Esta parte es la más delicada de configurar. Por un lado se configura el servidor para que trabaje únicamente con conexiones cifradas, usando HTTPS. Esto permitirá cifrar la conexión entre el cliente y el servidor, protegiendo la integridad y la confidencialidad de los datos transmitidos. Simplemente con la instalación de *mod_ssl* y la creación de unos certificados, se puede habilitar esta opción.

Por el otro lado se debe configurar la comunicación con la aplicación web.

#### 6.2.5 Aplicación web - Nettor

La aplicación web se encuentra comprimida en un fichero tar.gz en los anexos. Para integrarla en un nuevo servidor, simplemente se debe extraer su contenido en la carpeta /var/www/nettor/. Una vez estén todos los ficheros en esa carpeta, se deben cambiar los permisos del directorio y subdirectorios además del propietario de los mimos.

Se hace uso de una herramienta nombrada *supervisor* para poder levantar el servidor en caso de desconexiones causadas por cortes de luz o de línea.

El fichero /var/www/nettor/nettor/settings.py, contiene el corazón de la aplicación. Ahí se especifica valores clave para el funcionamiento de la herramienta tales como la conexión con la base de datos.

### 6.3 Distribución del sistema

Es posible distribuir el sistema de dos formas distintas.

- **Máquina virtual.** Ya que todo el desarrollo se ha realizado en una máquina virtual, se puede hacer una copia de esta y reutilizarla. En caso de seleccionar esta opción, únicamente se tendría que tener en cuenta los dos últimos pasos descritos en el apartado de pasos previos.
- **Instalación manual.** En los anexos se detallan las instrucciones llevadas a cabo para la instalación de todos los componentes y sus ficheros de configuración. Así como de la aplicación desarrollada. Simplemente se tendrían que tener en cuenta los dos apartados anteriores y la información disponible en los anexos para su correcta instalación.

## 7. Resultados

Se ha desarrollado una aplicación web segura, que junto a los componentes analizados, forma un sistema de monitorización que cumple las expectativas. El sistema obtenido permite la visualización en tiempo real de los equipos monitorizados, a través de una interfaz minimalista. Además es capaz de generar estadísticas de uso de los equipos y grupos de equipos, permitiendo a los administradores analizar la información recopilada. También, cuenta con la posibilidad de generar alertas, notificando a los administradores vía correo y acceder a todos los datos y configuraciones mediante una interfaz de administración. Finalmente se ofrecen elementos personalizables tales como la creación de grupos de equipos y la configuración de sistemas de escaneado.

En definitiva, el proyecto cumple con todos los objetivos fijados, puesto que tras su despliegue en el servidor de la escuela ha funcionado tal y como estaba planeado.

## 8. Trabajo futuro

Existen posibles mejoras para este trabajo, a continuación se indican algunas de ellas:

- Incorporación de Ajax para poder visualizar la información justo en el momento en que aparezca en la base de datos. Así se podría eliminar la etiqueta temporizadora de HTML presente en la interfaz básica.
- Generación de graficas en base a los datos de las estadísticas. Los datos recopilados y tratados se podrían ofrecer de una forma más visual dentro de la aplicación, sin tener que depender de herramientas externas como Excel.
- Nuevos tipos de alertas. Por ejemplo se podría implementar un tipo de escaneado que realizase detecciones de servicios en las máquinas. En caso de detectar algún servicio fuera de los parámetros establecidos, se generaría una alerta.
- Interacción directa con el servidor. Se podría habilitar una vista en la aplicación web, que permitiera conectarse al servidor mediante SSH.

## 9. Conclusiones

Con la realización de este proyecto, se han aprovechado los conocimientos adquiridos a lo largo de la carrera. El conocimiento de redes y de la administración de sistemas operativos y sus servicios, han jugado un papel muy importante. Por otro lado, al inicio de este se desconocía completamente, el funcionamiento de las aplicaciones web y de la mayoría de los componentes que forman parte del sistema generado. Esto ha llevado emplear muchísimo tiempo al estudio de todos estos y creo que el resultado final ha valido la pena.

Con este proyecto espero aportar mi granito de arena para mejorar los servicios ofrecidos en la escuela.

## Referencias


[1] Equipo de Debian. Acerca de Debian [en línea]. Debian, Página oficial, 2015 [fecha de consulta: 26 de Febrero del 2015]. Disponible en <https://www.debian.org/intro/about#what>.

[2] Equipo de Python. Wiki Python [en línea]. Python, Página oficial, 2015 [fecha de consulta: 28 de Febrero del 2015]. Disponible en <https://wiki.python.org/>.

[3] Equipo de Django. Meet Django [en línea]. Django, Página oficial, 2015 [fecha de consulta: 28 de Febrero del 2015]. Disponible en <https://www.djangoproject.com/>.

[4] Equipo de PostgreSQL. About PostgreSQL [en línea]. PostgreSQL, Página oficial, 2015 [fecha de consulta: 2 de Marzo del 2015]. Disponible en <http://www.postgresql.org/about/>.

[5] Equipo de Nginx. Nginx [en línea]. Nginx, Página oficial, 2015 [fecha de consulta: 2 de Marzo del 2015]. Disponible en <http://nginx.org/en/>.

[6] Equipo de Nmap. Nmap description [en línea]. Nmap, Página oficial, 2015 [fecha de consulta: 24 de Febrero del 2015]. Disponible en <https://nmap.org/book/man.html#man-description>.

[7] Equipo de Bootstrap. Bootstrap [en línea]. Bootstrap, Página oficial, 2015 [fecha de consulta: 30 de Marzo del 2015]. Disponible en <http://getbootstrap.com/>.