

INSTITUTO UNIVERSITARIO AERONÁUTICO

INFORME TFG

ISPSTATS

**Detección automática de equipos de red y recolección de estadísticas para
ISPs**

Ingeniería de Sistemas

Alumno: Alejandro Barturen

Tutor: Ing. Mariano García Mattio

Dedicatoria

A mis familias.

Agradecimientos

A mis colegas y amigos del grupo iglú y de ochoquince.

Título del Proyecto

ISPSTATS: Detección automática de equipos de red y recolección de estadísticas para ISPs

Hoja de Aceptación del Trabajo Final

APÉNDICE I



I NSTITUTO
U NIVERSITARIO
A ERONAUTICO

FECHA: 11/12/2016

FACULTAD: INGENIERÍA

DEPARTAMENTO: SISTEMAS

SOLICITUD DE TEMA DE TRABAJO FINAL DE GRADO / TRABAJO FINAL DE PREGRADO

Sr. Director de Departamento:

Me dirijo a Ud. a fin de solicitar la aprobación del tema de TRABAJO FINAL DE GRADO que propongo a continuación:

TEMA:

Nombre: ISPSTATS

Descripción: Detección automática de equipos de red y recolección de estadísticas para ISPs

Es un trabajo multidisciplinario Sí X No
Otro/s Dpto/s que participan: ---

Tutor de Trabajo Final:

Nombre: ING. MARIANO GARCÍA MATTIO

Cargo: DOCENTE

Dirección Personal o Laboral: LOPEZ Y PLANES 382 - RÍO CEBALLOS - CÓRDOBA

Tel: 03543 450528

e-mail: MAGM@IUA.EDU.AR

Firma del Tutor:

Datos del Alumno:

Nombre y Apellido: ALEJANDRO BARTUREN

Carrera: INGENIERÍA DE SISTEMAS

Matrícula: 30409091

Materias que faltan aprobar: ---

Dirección: CALLE 49 NRO 1131

Localidad: GENERAL PICO (6360) Provincia: LA PAMPA

Tel: 02302 15 598 500

e-mail: ALE.BARTUREN@GMAIL.COM

Córdoba: 11/12/2016 Firma del Alumno:

Índice

Carátula.....	1
Dedicatoria.....	2
Agradecimientos.....	2
Título del Proyecto.....	2
Hoja de Aceptación del Trabajo Final.....	3
Índice.....	4
Glosario.....	5
Resumen.....	5
Palabras Clave.....	6
Introducción.....	6
Objetivo del Proyecto.....	9
Destinatarios.....	11
Beneficios.....	11
Estudio Técnico.....	13
Desarrollo del Trabajo: Metodología.....	15
Desarrollo del Trabajo: Resumen Técnico.....	21
Desarrollo del Trabajo: Actividades Realizadas.....	24
Desarrollo del Trabajo: Resultados Alcanzados.....	25
Inversión Requerida.....	38
Proyección de Costos de Operación y Mantenimiento.....	38
Análisis de Viabilidad Comercial.....	39
Análisis Financiero.....	39
Estudio Ambiental.....	39
Estudio Social.....	40
Evaluación Económica.....	40
Conclusiones.....	40
Bibliografía.....	42
Anexo 1: Diagrama de Clases.....	43
Anexo 2: Diagrama Entidad-Relación (DER).....	44

Glosario

ISP: Internet Service Provider (Proveedor de Servicios de Internet)

RRDB: Round Robin Database (Base de Datos Round Robin o de tamaño fijo)

Resumen

El espíritu del proyecto es el de reducir la carga de trabajo asociada al registro e identificación inicial de los equipos que componen la infraestructura de transporte de un ISP, así como el mantenimiento de este listado.

Se automatizará lo más posible el proceso, de manera tal que el software sea capaz de detectar los equipos con que se cuenta y sus credenciales de acceso, así como su marca, modelo y versión de firmware. En función de esa información, se seleccionará luego la plantilla de recopilación de estadísticas correspondiente. Finalmente, el sistema hará uso de estas plantillas para presentar un panel de estadísticas al usuario.

La metodología a utilizar será kanban, usando como herramienta de soporte trello.

El resultado del proyecto será la posibilidad de brindar al operador un sencillo panel de configuración donde, luego de responder algunas preguntas, pueda confiar en que el software hará el reconocimiento necesario para ofrecer estadísticas confiables sobre los equipos, de manera automática.

Las dos consecuencias principales serán, por un lado, el ahorro de tiempo y esfuerzo en la creación del panel de estadísticas, así como de su manutención, y, por el otro, la mejor capacidad para la toma de decisiones del ISP al contar con información relevante, confiable y actualizada sobre la performance de su red.

Palabras Clave

ISP, Estadísticas

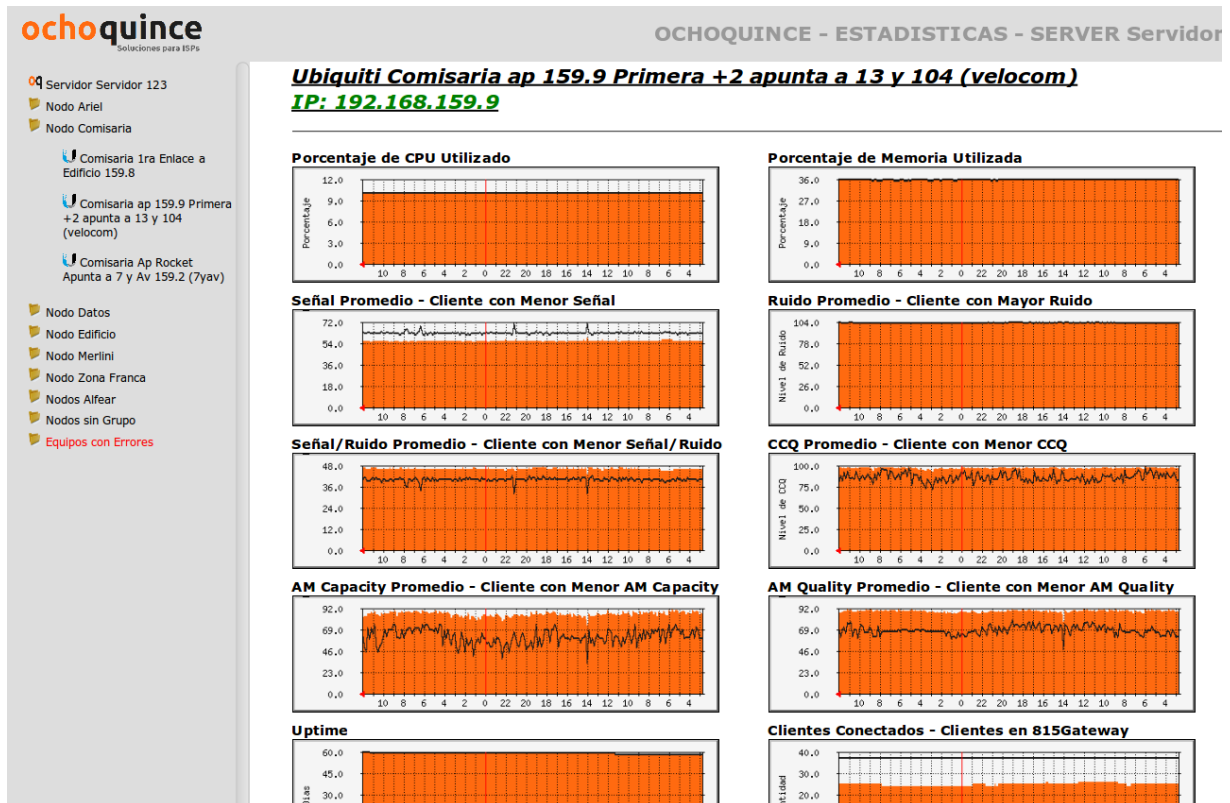
Introducción

Ochoquinze es una empresa que se dedica al diseño y desarrollo de soluciones para ISPs (Proveedores de Internet, Internet Service Provider, por su sigla en inglés).

Una de las herramientas que se proveen es un panel de estadísticas donde el ISP puede conocer la performance de los distintos equipos que componen su infraestructura de transporte, es decir, aquellos equipos que viven dentro de la red entre el acceso a internet del propio ISP, y su clientes. Es claro que no hay un enlace directo entre el domicilio de cada cliente y este acceso final, sino que existen varios equipos, conectados entre si, encargados de ir haciendo la distribución.

Si imaginamos un esquema de árbol, en la raíz encontraríamos el router WAN del ISP, a partir del cual irían creciendo ramas de nodos interconectados, llegando finalmente a las hojas, que serian los clientes finales. Este panel de estadísticas hace foco en las ramas del mencionado árbol, cuyo correcto desempeño es fundamental para brindar un buen servicio, un mantra repetido en la industria es: “no basta con tener buena internet, es necesario poder repartirla”, y explica que un único cuello de botella es suficiente para arruinar la percepción del servicio de todos los clientes que tiene detrás, naturalmente, mientras más cercano se encuentre el cuello de botella a la raíz del árbol, mayor será la cantidad de clientes afectados.

Por lo expuesto anteriormente, cae de maduro que un ISP preocupado por la calidad debe ser especialmente vigilante del comportamiento de su red. Para ayudar en este sentido, uno de los productos provistos por ochoquinze, en particular, 815Gateway, provee varias herramientas como ser monitoreo, geolocalización, configuración masiva, diagnóstico, entre otras cosas. Una de esta herramientas es un panel de estadísticas donde se provee información sobre el desempeño de la red. Un típico panel se ve de esta manera:



Y consiste en un árbol, del lado izquierdo, donde se encuentran agrupados los nodos de acuerdo a algún criterio elegido pro el usuario. En este caso se trata de un WISP (Wireless Internet Service Provider), por lo que cada grupo hace referencia a una torre, dentro de la cual se encuentran los equipos allí presentes. Para cada uno se muestran los siguientes valores:

- Porcentaje de CPU utilizado
- Porcentaje de Memoria utilizado
- Señal Promedio + Peor nivel de Señal encontrado
- Ruido Promedio + Peor nivel de Ruido encontrado
- Relación Señal sobre Ruido Promedio + Peor Señal/Ruido encontrado
- CCQ Promedio + Peor CCQ encontrado
- Tiempo en funcionamiento del equipo (uptime)
- Cantidad de clientes conectados + cantidad de clientes teóricos
- Tráfico en cada interfaz (eth0 y ath0)
- Tráfico originado del equipo
- Latencia hacia el equipo

Dependiendo de marca y modelo, es posible que se registren otros valores adicionales. Hay equipos que, por ejemplo, reportan voltaje de entrada o porcentaje de

almacenamiento utilizado, también existen protocolos propietarios de acuerdo al fabricante que, dependiendo del caso, reportan valores que son interesantes de graficar.

El problema encontrado es que para que esta herramientas puedan funcionar, es necesario cargar todos los equipos presentes en la red, uno por uno, de forma manual. Por otro lado, muchas veces esta información, se encuentra incompleta o desactualizada, se provee en un formato de difícil acceso, o directamente no está disponible.

La idea de este proyecto, entonces, consiste en automatizar la detección y carga de estos equipos, para que luego de responder una simples preguntas, el operador pueda dejar al software hacer su trabajo y contar con el panel completo y actualizado de forma automática.

Objetivo del Proyecto

Objetivo general: Generar el panel de estadísticas y mantener actualizado el listado de equipos a consultar de forma automática.

Objetivo Específico 1: Crear los modelos necesarios para la captura de la información a utilizar. A saber:

- Perfiles de detección
- Registros de ejecución
- Equipos
- Perfiles de recolección
- Perfiles de identificación
- Modelos auxiliares

Objetivo Específico 2: Crear un panel de administración web donde puedan configurarse y consultarse todos los modelos creados

Objetivo Específico 3: Permitir la ejecución de detecciones, de acuerdo a los parámetros configurados, y proveer un registro de los resultados de cada ejecución.

Objetivo Específico 4: Mantener actualizado el listado de equipos conocidos con sus respectivos parámetros.

Objetivo Específico 5: Permitir la administración y configuración de perfiles de recolección de estadísticas en función de marca, modelo y versión de firmware de cada equipo conocido.

Objetivo Específico 6: Permitir la administración y configuración de perfiles de detección automática de equipamiento en función de una serie de comandos a ejecutar y el resultado esperado para cada uno.

Objetivo Específico 7: Permitir la visualización de las estadísticas recolectadas para los equipos conocidos.

Destinatarios

El autor del presente es socio fundador de ochoquince, empresa que se verá beneficiada con este desarrollo, al permitir a sus clientes disponer de una herramienta superadora de la actual.

Beneficios

El principal beneficio económico viene por el lado del ahorro de costos por mayor eficiencia.

En promedio, la configuración de la herramienta lleva 3 días. Esto se debe principalmente a que existe un ida y vuelta entre el ISP y la empresa para obtener la información, una vez que la misma está disponible y se encuentra verificada, la carga consiste en un día de trabajo. Luego, aunque es más difícil de medir, también se gasta tiempo en mantener la herramienta actualizada a medida que se cambian equipos, se dan de baja algunos y se agregan otros nuevos durante el normal funcionamiento del ISP a lo largo del tiempo.

Si asumimos que la configuración lleva un día de trabajo (8 horas reloj) a un empleado, y asumimos como costo promedio por hora de trabajo de un ingeniero, por parte de la empresa, de unos AR\$ 700,00 (a Octubre del 2016), por otro lado asumimos también dos implementaciones mensuales y, finalmente, asumimos que el tiempo de configuración puede reducirse a una hora, tenemos entonces 14 horas de trabajo mensuales de ahorro, lo que arroja unos AR \$9,800,00 mensuales de beneficio neto.

Esta información se obtuvo de los registros contables de la empresa y de la ticketera que se utiliza para trazar el trabajo realizado diariamente, y se asume que será cada vez más beneficiosa a lo largo del tiempo si la empresa sigue creciendo y se consigue hacer una mayor cantidad de implementaciones por mes.

Fuera de lo económico, se observan beneficios como menor tiempo de implementación, mayor satisfacción por utilizar mejor tecnología, así como reducción de trabajo engorroso y repetitivo.

Estudio Técnico

La selección de tecnologías no requirió de gran análisis, simplemente se aprovechan las tecnologías que ya están en juego, considerando que el desarrollo debe formar parte de un producto que ya tiene 8 años en el mercado y sobre el que trabaja un equipo de unas 10 personas, por lo que resulta ventajoso aprovechar las fortalezas con las que se cuentan gracias a la experiencia de venir trabajando sobre una misma base desde hace ya un tiempo considerable.

En cuanto a la definición de procesos, es claro que el análisis resulta beneficioso ya que, precisamente, se apunta a simplificar un proceso existente: en vez de cargar los equipos de forma manual, se deben explicitar aspectos sobre el conjunto de los mismos y luego dejar que el software haga su trabajo. Apuntamos a reducir la carga de 8 horas a 1 hora por implementación. El análisis económico se realizó en el punto anterior.

Tanto el establecimiento de los medios puestos en juego como la previsión de la infraestructura requerida son también de sencillo análisis: el autor realizará el desarrollo en su tiempo libre, fuera del horario de oficina, utilizando los medios con los que la empresa ya cuenta y que se encuentran ociosos durante esos momentos.

La estimación de los rendimientos esperados se realizó también en el punto anterior.

Habiendo hecho estas salvedades, podemos comentar que para la implementación del desarrollo se utilizará el framework de programación web Django (junto a algunos complementos adicionales), utilizando el lenguaje de programación python y algunas de sus librerías. Mysql será el motor de base de datos, Apache el servidor web y se utilizará GNU/Linux (Debian en particular) como sistema operativo de base. La capacidad de realizar ping ICMP y de iniciar sesiones SSH también será provista por el sistema operativo. Para el almacenamiento de las estadísticas en un formato compacto y eficiente, así como para la generación de los gráficos a mostrar, se utilizará rrdtool.

Para las actividades de desarrollo se utilizará Vagrant junto con Virtualbox, ejecutando Debian en su versión 8, ejecutándose todo esto, a su vez, sobre una PC utilizando Ubuntu en su versión más reciente.

Desarrollo del Trabajo: Metodología

La metodología a utilizar será una adaptación de Kanban, usando como herramienta de soporte trello.

Las funcionalidades a implementar son las siguientes:

- Detección
- Identificación
- Recolección
- Visualización

Detección

Se asume que los equipos a detectar responden ping ICMP. El primer paso es la creación de un perfil de detección que incluya:

- Rangos de direcciones IP
- Listados de puertos SSH y combinaciones de [usuario + contraseña]

Luego procederemos a ejecutar el perfil de detección, para ello realizaremos un ping ICMP (5 paquetes) a cada dirección IP presente dentro de los rangos definidos. Si una dirección IP responde ping, intentaremos entonces iniciar una sesión SSH de acuerdo a los puertos y credenciales provistos (por puerto primero y por combinaciones de usuario y contraseña después). Si el acceso por SSH resulta exitoso, probaremos los perfiles de detección que estén activos hasta que identifiquemos el equipo, o bien agotemos los perfiles disponibles.

Al finalizar esta ejecución, crearemos un reporte que incluirá:

- Fecha y hora de la ejecución

- Perfil de detección ejecutado
- Cantidad y listado de direcciones IP que no respondieron ping
- Cantidad y listado de direcciones IP que si respondieron ping
- Cantidad y listado de direcciones IP que, habiendo respondido ping, no pudieron accederse por SSH
- Cantidad y listado de direcciones IP que respondieron ping y que pudieron accederse por SSH, pero que no pudieron identificarse utilizando los perfiles de detección disponibles
- Cantidad y listado de equipos detectados correctamente

En otras palabras, el resumen del resultado de una ejecución debiera verse parecida a la siguiente tabla:

ETAPA	SI	NO
PING	300	100
SSH	250	50
IDENTIFICACIÓN	120	130

Finalmente, procederemos a actualizar el listado de equipos conocidos. Para cada equipo que pudo accederse por SSH, independientemente de si pudo ser identificado o no, revisaremos si se trata de un equipo nuevo (presente en el listado) o no. Si se trata de un equipo nuevo, instanciaremos el modelo con todos los datos que conocemos, aquellos que no conozcamos quedarán en blanco. Si se trata de un equipo existente, actualizaremos sus campos de acuerdo a las novedades que haya arrojado la ejecución de la detección.

De esta manera, el usuario podrá ir evaluando que tan efectiva es la detección, como para saber si necesita ajustar parámetros, o bien crear nuevos perfiles de identificación para mejorar la proporción de equipos correctamente identificados.

Es importante en este momento explicar cómo se compondrá un equipo. El modelo en cuestión contará con los siguientes campos:

Activo	Indica si queremos juntar estadísticas o no. Por defecto es falso. Este es el único campo editable por el usuario
--------	---

Marca	Todos estos campos se completarán automáticamente, de ser posible, de acuerdo a los resultados de la última detección que aplique
Modelo	
Versión de firmware	
Hostname	
Dirección IP	
Usuario SSH	
Contraseña SSH	
Puerto SSH	

La última vez que intentamos recolectar estadísticas para este equipo
Si la última recolección fue exitosa o no (por ejemplo porque el equipo estaba apagado, o porque ocurrió algún error con el script de recolección)

El perfil de detección que se usó para crearlo, y la fecha de creación
El perfil de detección que lo actualizo por última vez, y la fecha de la última modificación

Conociendo esta información para cada equipo encontrado, el usuario dispondrá de un listado actualizado (suponiendo que se ejecuta la detección periódicamente) de los equipos que componen su red, podrá también elegir cuáles quiere graficar y cuáles no, ver cuáles equipos no se están identificando correctamente como para mejorar los perfiles de identificación y verificar si hay equipos correctamente identificados pero que no se están pudiendo graficar, como para ver si hay algún problema con los scripts o si le pasa algo a ese equipo.

Identificación

Se permitirá al usuario configurar perfiles que contengan la información necesaria para identificar marca, modelo o versión de firmware de un equipo por SSH.

Los perfiles contarán con los siguientes campos:

- Tipo de identificación: Este campo indica si vamos a identificar marca, modelo o versión de firmware
- Nombre del perfil
- Comando a ejecutar

- Valor Esperado
- Padre: Otro perfil de identificación de tipo "marca", o nulo.

El funcionamiento será el siguiente:

En primer lugar, queremos identificar la marca del equipo (es decir, el fabricante). Para ello iremos recorriendo todos los perfiles de tipo "marca", ejecutando por SSH el comando correspondiente, y comparando la salida del comando con el valor esperado. Si la salida no coincide, el comando arroja algún error, da timeout, u ocurre alguna otra excepción, entonces sabemos que no se trata de un equipo de esta marca y podemos pasar al siguiente.

Al finalizar esta etapa, o bien agotamos todos los perfiles y no pudimos identificar el equipo (en cuyo caso los campos de marca, modelo y firmware quedarán vacíos), o bien tuvimos éxito y ya conocemos la marca del equipo en cuestión.

Conociendo la marca, recorreremos ahora, en primer lugar, los perfiles de identificación de tipo "modelo" cuyo padre sea el comando de identificación de tipo "marca" que tuvo éxito. Ejecutaremos los comandos de identificación de equipo, de la misma forma que los de marca, hasta que encontremos el equipo correspondiente o agotemos todos los perfiles. Si tenemos éxito, ya conocemos el modelo del equipo, de lo contrario, este campo quedará en blanco. En segundo lugar, y de la misma manera, intentaremos identificar la versión de firmware del modelo, hasta que tengamos éxito o agotemos los perfiles.

Separar los perfiles de identificación de esta forma nos permite reutilizar los comandos de identificación de marca, como así también agilizar las tareas de detección (porque podemos abortar temprano si la detección de marca falla). Revisando los resultados de cada ejecución, podremos conocer qué equipos no estamos pudiendo identificar, y podremos también crear entonces nuevos perfiles de detección, o mejorar los existentes, sin necesidad de modificar el software.

Recolección

Se permitirá al usuario configurar perfiles de recolección de estadísticas. Estos perfiles cuentan con los siguientes campos:

- Nombre del perfil
- Listado de combinaciones de marca, modelo y firmware para los que aplica.
- Listado de valores a graficar, por ejemplo:
 - Título: Porcentaje de Uso de CPU
 - Clave: `cpu_percent`
 - Unidad: %
 - Tipo: Gauge
- Título: Subida Interfaz ath0
 - Clave: `ath0_up_bits`
 - Unidad: bps
 - Tipo: Derive
- Un archivo de texto a ejecutar

Los archivos (scripts) se pueden subir desde la interfaz web y se pueden reutilizar entre perfiles de recolección .

Reciben:

- El pk del equipo del cual queremos recolectar estadísticas.

Devuelven:

- Un listado de clave y valor a almacenar, separados por un espacio, uno por línea. Por ejemplo:

```
cpu_percent 22
ath0_up_bits 5998653214
```

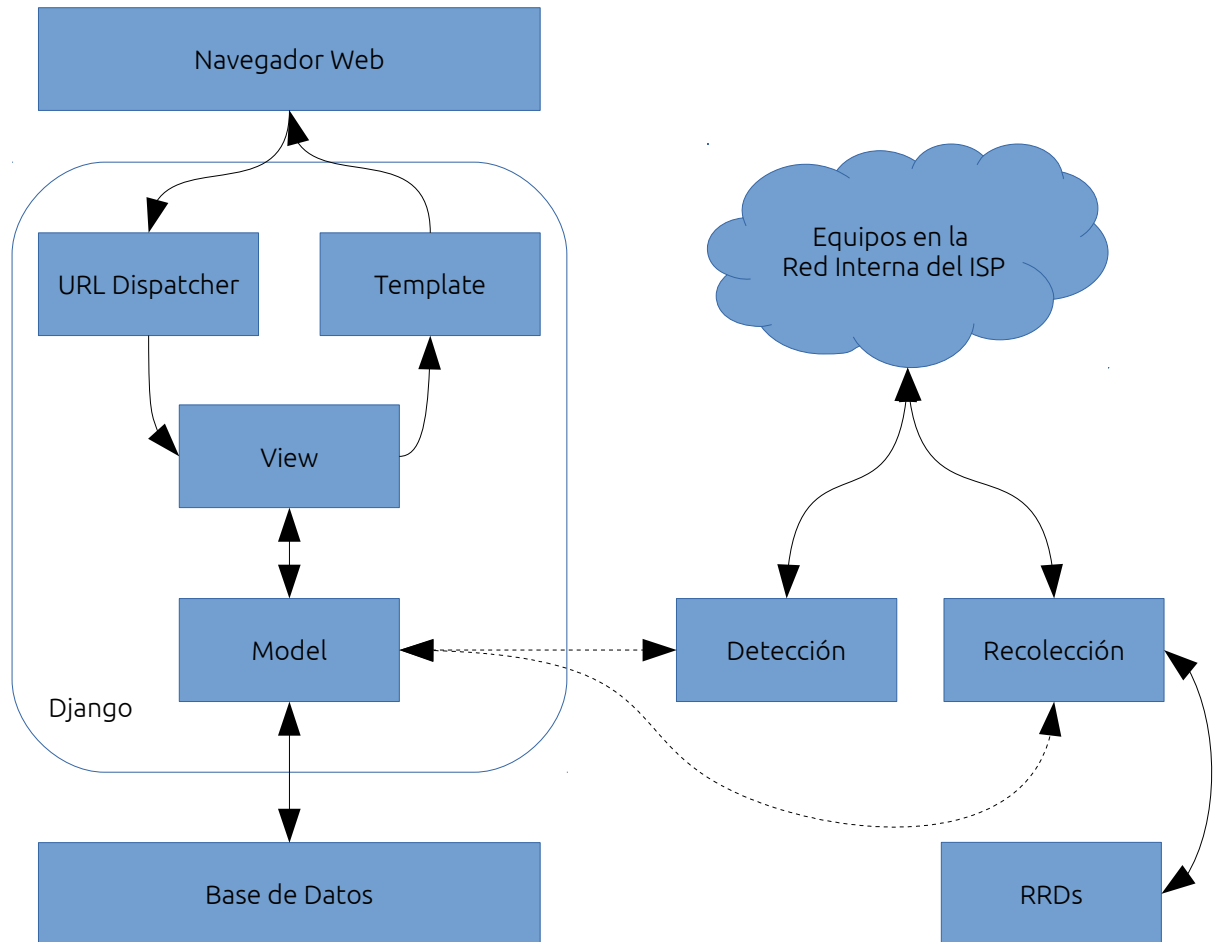
De esta manera se puede ir actualizando la herramienta sin tener que hacer una actualización del software en sí mismo, al encontrar una oportunidad, se mejorarán los scripts (o se crearán nuevos) que consecuentemente puedan subirse, logrando así mejores gráficos sin necesidad de actualizar el software.

Luego, de manera automática y periódica, se recorrerán los equipos activos, para todos aquellos que tengan correctamente identificado marca, modelo y firmware , si existe algún modelo de recolección que aplique para las tres cosas, recorreremos el listado de valores a graficar, verificando que exista un archivo .rrd para cada clave , luego ejecutaremos el script de recolección correspondiente , interpretaremos la salida del script, y actualizaremos cada rrd con el valor obtenido , finalmente, actualizaremos en el equipo los valores de ultima recolección, indicando éxito si el script se ejecuto correctamente, o error en caso contrario.

Visualización

Ofreceremos al usuario todos los equipos activos para que seleccione las estadísticas de cuál de ellos desea visualizar, una vez seleccionado, le mostraremos un resumen de la información del equipo , las curvas que correspondan (obteniendo la información de cada rrd) de acuerdo al perfil de recolección utilizado y la opción de seleccionar el periodo a visualizar (ultimas 24hs, ultima semana, ultimo mes o ultimo año)

Desarrollo del Trabajo: Resumen Técnico



El **Navegador Web** es la herramienta que utiliza el usuario para interactuar con el sistema. Realiza peticiones y recibe respuestas HTTP.

El **URL Dispatcher** mapea la URL requerida a una función de vista (view) y llama a dicha función con el objeto HttpRequest correspondiente.

La función de vista **View** realiza la acción requerida para responder la petición del cliente, la cual típicamente involucra leer o escribir en la Base de Datos haciendo uso de los modelos (model) disponibles.

Cada modelo **Model** es una clase Python que representa los datos de una entidad y su comportamiento, pudiendo definirse también de qué manera los modelos se relacionan

entre sí. Usualmente cada modelo se almacena en una tabla de una Base de Datos relacional.

Una vez realizada la tarea requerida, la función de vista *view* devuelve un objeto de tipo *HttpResponse* al Navegador Web, usualmente luego de pasar los datos a través de una plantilla (**Template**), la cual facilita la generación de HTML para resolver la presentación de los datos a devolver.

Apuntando a la simplicidad del esquema, algunas cosas se dejaron intencionalmente fuera del mismo. Cabe destacar que **Django** se ejecutará como una aplicación WSGI (el estándar de Python para aplicaciones y servidores Web), por lo que en realidad la comunicación con el Navegador Web tiene una capa intermedia en la cual se ubican Apache + mod_wsgi, donde Apache es el servidor web y mod_wsgi es un módulo para Apache capaz de alojar aplicaciones WSGI Python (incluyendo a Django, obviamente).

En el espíritu del punto anterior, también cabe mencionar que Django provee un ORM para la manipulación de los datos y la consiguiente interacción con la **Base de Datos**. Si bien puede utilizarse directamente SQL, para los fines de la aplicación a desarrollar el ORM disponible resulta suficiente.

En paralelo al sistema anteriormente descrito se encuentra el funcionamiento de las herramientas de **Detección** de Equipos y **Recolección** de Estadísticas. Estas herramientas pueden iniciarse bajo demanda del usuario, o bien ejecutarse periódicamente de forma automática. Durante su ejecución, ambas consultarán a la Base de Datos a través de los objetos Model para obtener la información que necesitan para cumplir sus objetivos, y luego utilizarán esa información para interpelar los **Equipos en la Red Interna del ISP**, ejecutar algún comando sobre ellos y, luego de trabajar con los resultados obtenidos, volver a comunicarse con la Base de Datos para registrar su ejecución.

Por otro lado, la herramienta de Recolección en particular también se comunicará con los archivos **RRD** (por Round Robin Database) para almacenar las estadísticas obtenidas. Estos archivos se mantiene fuera de la base de datos relacional ya que utilizan un funcionamiento especial para mantener su tamaño constante a lo largo del tiempo y proveen herramientas de compresión y visualización diseñadas específicamente para datos almacenados en forma de serie de tiempo (timeseries).

Bibliografía: Documentación oficial de Django disponible en:

<https://docs.djangoproject.com/en/1.9/>

Desarrollo del Trabajo: Actividades Realizadas

Como se explicó en el punto 6 (Metodología), Las funcionalidades a implementar se dividen en las siguientes áreas:

- Detección
- Identificación
- Recolección
- Visualización

El orden está dado por la naturaleza de las tareas, es imposible implementar alguna de ellas si antes no están implementadas todas las anteriores.

Empezaremos por implementar la Detección, al finalizar esta etapa esperamos contar con la posibilidad de ejecutar una detección parcial, hasta el punto de saber a qué equipos podemos ingresar exitosamente por SSH.

En segundo lugar, implementaremos los perfiles de Identificación y luego volveremos a repasar la etapa anterior, a esta altura, una ejecución exitosa debiera además de indicar los equipos a los que podemos acceder, completar correctamente los datos de aquellos equipos que hayan sido exitosamente identificados.

A esta altura, es natural que sigamos con la implementación de la recolección, ya que contamos con toda la información necesaria para empezar a recolectar estadísticas.

A continuación, una vez los archivos RRD se estén alimentando correctamente, implementaremos el panel de visualización para que el usuario pueda disfrutar de las estadísticas recolectadas.

Luego, dedicaremos tiempo a la creación de múltiples perfiles de identificación y recolección, y documentaremos este proceso de forma tal que todo el personal de la empresa sepa cómo darle seguimiento y mantenimiento a la nueva herramienta.

Finalmente, se pondrá la herramienta en producción, realizando la correspondiente actualización al producto y distribuyéndolo a nuestros clientes.

Desarrollo del Trabajo: Resultados

Alcanzados

Las funcionalidades implementadas fueron:

- 1) Detección
- 2) Identificación
- 3) Recolección
- 4) Visualización

Detección

Para implementar la detección, se creó un modelo llamado “Perfil de Detección”. El mismo permite capturar la información necesaria para realizar una detección, y tiene la siguiente forma:

Administración de Django
BIENVENIDO/A, ADMIN. [VER SITIO](#) / [CAMBIAR CONTRASEÑA](#) / [CERRAR SESIÓN](#)

Inicio > Deteccion > Perfiles de Detección > Redes 158 y 159

Modificar Perfil de Detección HISTORIA

Identificación

Nombre:

SUBREDES IP

DIRECCIÓN IP	MÁSCARA	¿ELIMINAR?
192.168.158.0/24 <input style="width: 100%;" type="text" value="192.168.158.0"/>	<input style="width: 100%;" type="text" value="/24 - 255.255.255.0"/>	<input type="checkbox"/>
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text" value="/24 - 255.255.255.0"/>	<input type="checkbox"/>

[+ Agregar otro/a Subred IP](#)

PUERTOS

PUERTO	¿ELIMINAR?
15915 <input style="width: 100%;" type="text" value="15915"/>	<input type="checkbox"/>
22000 <input style="width: 100%;" type="text" value="22000"/>	<input type="checkbox"/>
<input style="width: 100%;" type="text"/>	<input type="checkbox"/>

[+ Agregar otro/a Puerto](#)

CREDENCIALES

USUARIO	CLAVE	¿ELIMINAR?
ubnt : velodato <input style="width: 100%;" type="text" value="ubnt"/>	<input style="width: 100%;" type="text" value="12341234"/>	<input type="checkbox"/>
ubnt : Cata5314 <input style="width: 100%;" type="text" value="ubnt"/>	<input style="width: 100%;" type="text" value="12341234234"/>	<input type="checkbox"/>
<input style="width: 100%;" type="text"/>	<input style="width: 100%;" type="text"/>	<input type="checkbox"/>

[+ Agregar otro/a Credencial](#)

Eliminar
Guardar y agregar otro
Guardar y continuar editando
GUARDAR

Habiendo introducido la información necesaria, podremos luego ejecutar el perfil tantas veces como sea necesario. Una ejecución de la herramienta empieza por la selección del perfil a ejecutar:

Administración de Django

Inicio

Ejecutar Detección de Equipos

Seleccione perfil a ejecutar

Redes 158 y 159

Comenzar

Y continúa con el panel de resultados, el cual se va completando a medida que aparece nueva información. Unos resultado de ejemplo pueden verse a continuación:

Al comienzo:

Administración de Django

Inicio

Ejecutar Detección de Equipos

Ping ICMP

ejecutando...

Acceso SSH

esperando Ping ICMP...

Identificación

esperando Acceso SSH...

Durante la ejecución:

Administración de Django

Inicio

Ejecutar Detección de Equipos

Ping ICMP

- Direcciones: 254
- Responden Ping ICMP: 32 (12.60%) [Ver Listado](#)
- No responden Ping ICMP: 222 (87.40%) [Ver Listado](#)

Acceso SSH

- Direcciones: 32
- Se acceden por SSH: 26 (81.25%) [Ver Listado](#)
- No se acceden por SSH: 6 (18.75%) [Ver Listado](#)

Identificación

ejecutando...

Al finalizar:



Ejecutar Detección de Equipos

Ping ICMP

- Direcciones: 254
- Responden Ping ICMP: 32 (12.60%) [Ver Listado](#)
- No responden Ping ICMP: 222 (87.40%) [Ver Listado](#)

Acceso SSH

- Direcciones: 32
- Se acceden por SSH: 26 (81.25%) [Ver Listado](#)
- No se acceden por SSH: 6 (18.75%) [Ver Listado](#)

Identificación

- Direcciones: 26
- Se identificaron: 26 (100.00%) [Ver Listado](#)
- No se identificaron: 0 (0.00%) [Ver Listado](#)

Los links de “Ver Listado” muestran los hosts que se reconocieron en cada etapa, por ejemplo:

Ping ICMP

- Direcciones: 254
- Responden Ping ICMP: 32 (12.60%) [Ver Listado](#)
- No responden Ping ICMP: 222 (87.40%) [Ver Listado](#)

Acceso SSH

- Direcciones: 32
- Se acceden por SSH: 26 (81.25%) [Ver Listado](#)
- No se acceden por SSH: 6 (18.75%) [Ver Listado](#)
 - 192.168.158.1
 - 192.168.158.9
 - 192.168.158.49
 - 192.168.158.135
 - 192.168.158.141
 - 192.168.158.171

Identificación

- Direcciones: 26
- Se identificación: 26 (100.00%) [Ver Listado](#)
 - 192.168.158.72
 - Marca: Ubiquiti
 - Modelo: NanoStation M5
 - Firmware: XM.v5.6.4
 - 192.168.158.138
 - Marca: Ubiquiti
 - Modelo: NanoStation M5
 - Firmware: XM.v5.5.6
 - 192.168.158.75
 - Marca: Ubiquiti
 - Modelo: NanoStation M5

Identificación

En esta sección es donde cargamos “Identificadores”. Los mismo son comandos a ejecutar para identificar un equipo ya sea por Marca, Modelo o Versión de Firmware, en función de un valor esperado. Durante las ejecuciones de la herramienta de detección, a los equipos a los que puede accederse correctamente por ssh se los intenta identificar ejecutando estos comandos hasta que alguno coincide, o bien agotamos los comandos disponibles. Un identificador tiene la siguiente forma:

Administración de Django BIENVENIDO/A. ADMIN. VER SITIO / CAMBIAR CONTRASEÑA / CERRAR SESIÓN



Inicio › Identificación › Identificadores › TOUGHSwitch

Modificar Identificador HISTORIA

Identificación

Nombre:

Tipo:

Padre:  

Comando

Comando a Ejecutar:

Valor Esperado:

A medida que la herramienta de detección va identificando equipos, va creando entradas nuevas (si se trataba de un equipo desconocido) o bien va actualizando las existentes en la tabla de Equipos. La misma se ve de esta manera:

Administración de Django
BIENVENIDO/A, ADMIN. [VER SITIO](#) / [CAMBIAR CONTRASEÑA](#) / [CERRAR SESIÓN](#)

Inicio · Equipos · Equipos

Seleccione Equipo a modificar AGREGAR EQUIPO +

Acción: 0 de 30 seleccionados/as

<input type="checkbox"/>	DIRECCIÓN IP	ACTIVO	1	ESTADÍSTICAS	MARCA	2	MODELO	3	VERSIÓN DE FIRMWARE	FECHA DE ÚLTIMA RECOLECCIÓN
<input type="checkbox"/>	192.168.158.32	✓		Visualizar	Ubiquiti		NanoStation Loco M5		XM.v5.5.6	-
<input type="checkbox"/>	192.168.158.78	✓		Visualizar	Ubiquiti		NanoStation Loco M5		XM.v5.5.6	-
<input type="checkbox"/>	192.168.158.75	✓		Visualizar	Ubiquiti		NanoStation M5		XM.v5.5.6	-
<input type="checkbox"/>	192.168.158.72	✓		Visualizar	Ubiquiti		NanoStation M5		XM.v5.6.4	-
<input type="checkbox"/>	192.168.158.249	✗		Visualizar	Ubiquiti		NanoBridge M5		XM.v5.6.4	-
<input type="checkbox"/>	192.168.158.159	✗		Visualizar	Ubiquiti		NanoBridge M5		XM.v5.6.4	-
<input type="checkbox"/>	192.168.158.110	✗		Visualizar	Ubiquiti		NanoBridge M5		XM.v5.5.6	-
<input type="checkbox"/>	192.168.158.170	✗		Visualizar	Ubiquiti		NanoStation Loco M5		XM.v5.6.1	-
<input type="checkbox"/>	192.168.158.153	✗		Visualizar	Ubiquiti		NanoStation Loco M5		XM.v5.5.6	-
<input type="checkbox"/>	192.168.158.132	✗		Visualizar	Ubiquiti		NanoStation Loco M5		XM.v5.5.6	-
<input type="checkbox"/>	192.168.158.193	✗		Visualizar	Ubiquiti		NanoStation Loco M5		XM.v5.6.6	-
<input type="checkbox"/>	192.168.158.66	✗		Visualizar	Ubiquiti		NanoStation Loco M5		XM.v5.5.6	-
<input type="checkbox"/>	192.168.158.64	✗		Visualizar	Ubiquiti		NanoStation Loco M5		XM.v5.5.6	-

Y un equipo puntual se ve de esta forma:

Administración de Django
BIENVENIDO/A. ADMIN. [VER SITIO](#) / [CAMBIAR CONTRASEÑA](#) / [CERRAR SESIÓN](#)

Inicio · Equipos · Equipos · Ubiquiti NanoStation Loco M5 - 192.168.158.32

Modificar Equipo HISTORIA

Identificación

Dirección IP:

Activo

Marca: ✎ +

Modelo: ✎ +

Versión de Firmware: ✎ +

Credenciales

Puerto SSH:

Usuario SSH:

Clave SSH:

Detección

Perfil de Detección (Creación): ✎ +

Fecha de Detección (Creación):
Fecha: Hoy | 📅
Hora: Ahora | ⌚

Perfil de Detección (Última Modificación): ✎ +

Fecha de Detección (Última Modificación):
Fecha: Hoy | 📅
Hora: Ahora | ⌚

Recolección

Fecha de Última Recolección:
Fecha: Hoy | 📅
Hora: Ahora | ⌚

¿Última Recolección Exitosa?

Eliminar
Guardar y agregar otro
Guardar y continuar editando
GUARDAR

Recolección

A continuación implementamos los “Perfiles de Recolección”, los cuales operan sobre los equipos que se encuentran activos. Estos perfiles definen un script a ejecutar, una serie de valor esperados para cada ejecución del script, y las combinaciones de marca, modelo y firmware para los que aplica.

Un perfil de recolección tiene la siguiente forma:

Modificar Perfil de Recolección

HISTORIA

Identificación

Nombre:

Script

Script:

```
# -*- coding: utf-8 -*-

import os
import sys
import django

sys.path.insert(0, '/opt/stats/')
os.environ["DJANGO_SETTINGS_MODULE"] = "stats.settings"
django.setup()
```

VALORES				
TITULO	CLAVE	UNIDAD	TIPO	¿ELIMINAR?
ValorRecoleccion object	<input type="text" value="Nivel de Señal"/>	<input type="text" value="signal"/>	<input type="text" value="db"/>	<input type="text" value="Gauge"/> <input type="checkbox"/>
ValorRecoleccion object	<input type="text" value="Ruido"/>	<input type="text" value="noise"/>	<input type="text" value="db"/>	<input type="text" value="Gauge"/> <input type="checkbox"/>
ValorRecoleccion object	<input type="text" value="Señal/Ruido"/>	<input type="text" value="signal_noise"/>	<input type="text" value="percent"/>	<input type="text" value="Gauge"/> <input type="checkbox"/>
ValorRecoleccion object	<input type="text" value="CCQ"/>	<input type="text" value="ccq"/>	<input type="text" value="db"/>	<input type="text" value="Gauge"/> <input type="checkbox"/>
ValorRecoleccion object	<input type="text" value="Tiempo en Funcionamiento"/>	<input type="text" value="uptime"/>	<input type="text" value="segundos"/>	<input type="text" value="Gauge"/> <input type="checkbox"/>
ValorRecoleccion object	<input type="text" value="RxBits/seg"/>	<input type="text" value="wlan_rx_bits"/>	<input type="text" value="bits/s"/>	<input type="text" value="Derive"/> <input type="checkbox"/>
ValorRecoleccion object	<input type="text" value="RxPackets/seg"/>	<input type="text" value="wlan_rx_packets"/>	<input type="text" value="packets/s"/>	<input type="text" value="Derive"/> <input type="checkbox"/>
ValorRecoleccion object	<input type="text" value="RxErros/seg"/>	<input type="text" value="wlan_rx_errors"/>	<input type="text" value="packets/s"/>	<input type="text" value="Derive"/> <input type="checkbox"/>
ValorRecoleccion object	<input type="text" value="TxBits/seg"/>	<input type="text" value="wlan_tx_bits"/>	<input type="text" value="bits/s"/>	<input type="text" value="Derive"/> <input type="checkbox"/>
ValorRecoleccion object	<input type="text" value="TxPackets/seg"/>	<input type="text" value="wlan_tx_packets"/>	<input type="text" value="packets/s"/>	<input type="text" value="Derive"/> <input type="checkbox"/>
ValorRecoleccion object	<input type="text" value="TxErros/seg"/>	<input type="text" value="wlan_tx_errors"/>	<input type="text" value="packets/s"/>	<input type="text" value="Derive"/> <input type="checkbox"/>
ValorRecoleccion object	<input type="text" value="Latencia"/>	<input type="text" value="latency"/>	<input type="text" value="ms"/>	<input type="text" value="Gauge"/> <input type="checkbox"/>
	<input type="text"/>	<input type="text"/>	<input type="text" value="-----"/>	<input type="checkbox"/>

[+ Agregar otro/a Valor](#)

COMBINACIONES			
MARCA	MODELO	FIRMWARE	¿ELIMINAR?
CombinacionRecoleccion object	<input type="text" value="Ubiquiti"/>	<input type="text" value="NanoStation M5"/>	<input type="text" value="XM.v5.6.4"/> <input type="checkbox"/>
CombinacionRecoleccion object	<input type="text" value="Ubiquiti"/>	<input type="text" value="NanoStation M5"/>	<input type="text" value="XM.v5.5.6"/> <input type="checkbox"/>
CombinacionRecoleccion object	<input type="text" value="Ubiquiti"/>	<input type="text" value="NanoStation Loco M5"/>	<input type="text" value="XM.v5.5.6"/> <input type="checkbox"/>
	<input type="text" value="-----"/>	<input type="text" value="-----"/>	<input type="text" value="-----"/> <input type="checkbox"/>

[+ Agregar otro/a Combinacion](#)

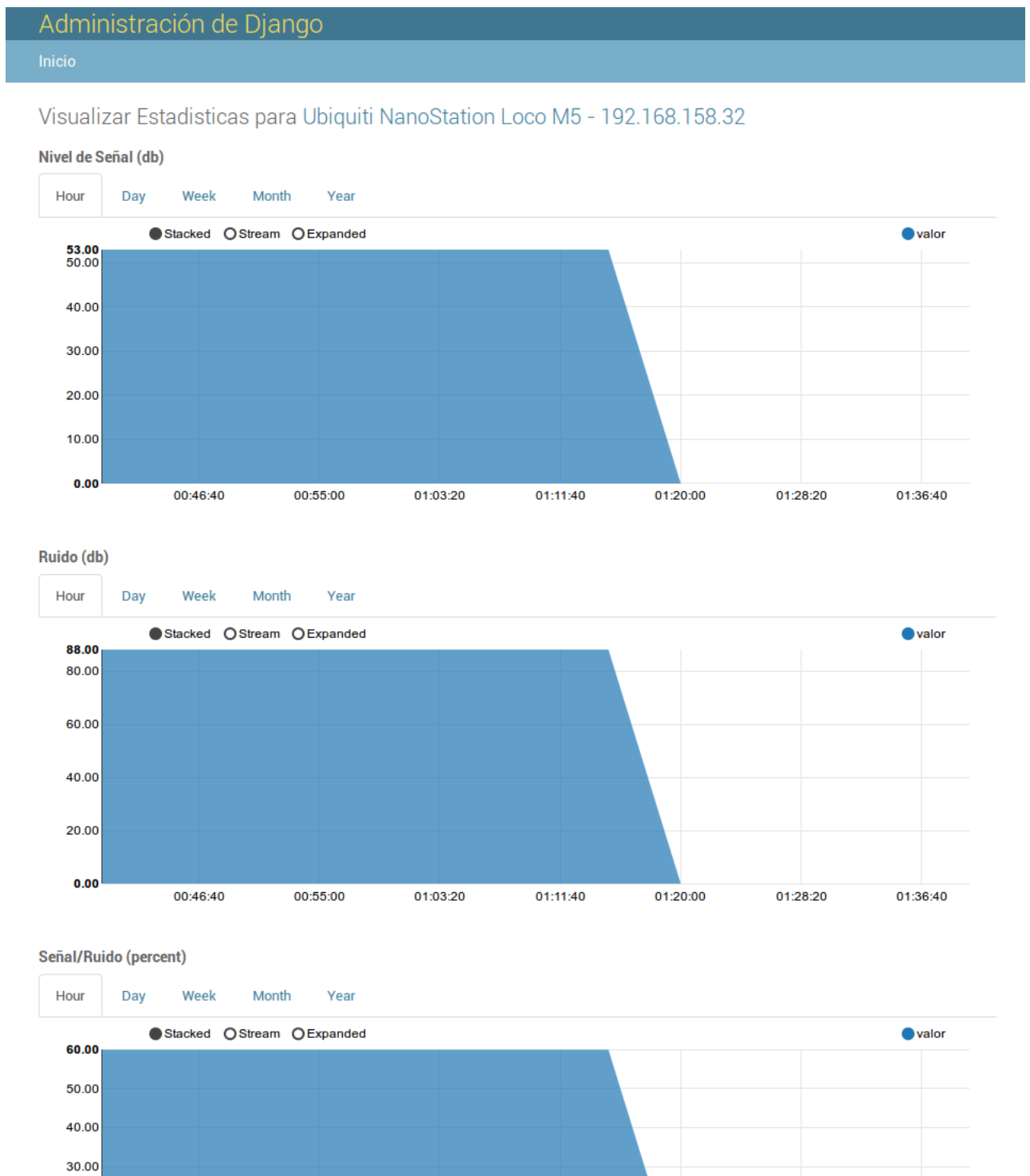
Luego, con cierta periodicidad, el sistema recorre los perfiles de recolección creados, identifica los equipos activos para los cuales aplica, y ejecuta el script en cuestión, una vez por equipo. A continuación, almacena los valores obtenidos en las bases de datos rrdbs correspondientes (si existieran, caso contrario las crea automáticamente).

A modo de ejemplo, una ejecución de un perfil de recolección sobre un equipo se vería se la siguiente forma:

```
# python /opt/stats/recoleccion/scripts/test.py 86
latency 3
signal 53
noise 88
signal_noise 60
ccq 909
uptime 2093272
wlan_rx_bits 1484826008
wlan_rx_packets 303420422
wlan_rx_errors 0
wlan_tx_bits 20332906712
wlan_tx_packets 14500927
wlan_tx_errors 0
```

Visualización

Finalmente, el sistema permite visualizar las estadísticas recolectadas para cada equipo, generando los gráficos de forma dinámica a partir de la información almacenada en las rrdbs. Las estadísticas de un equipo puntual tienen la siguiente forma:



y pueden consultarse tanto por hora como por día, semana, mes y año. Las mismas se refrescan automáticamente cada 5 minutos, acompañando el ciclo de actualización de los perfiles de recolección.

Inversión Requerida

El proyecto se eligió y diseñó para que no sea necesaria inversión de capital alguno (únicamente se invertirá tiempo, el cual será obtenido del que el autor vaya teniendo disponible de acuerdo a sus circunstancias).

A nivel de **recursos humanos**, este es un proyecto de una sola persona.

En cuanto a **infraestructura edilicia** el autor utilizará su actual domicilio y oficina, esta última en horarios no laborales.

Se cuenta actualmente con el **equipamiento e instrumental** necesario (una PC y conexión a internet, según se indicó en el punto 5).

Finalmente, no se observan insumos asociados a ninguno de los anteriores puntos que merezcan mención, aunque si es importante destacar que para la realización de las pruebas es necesario contar con una red de transporte, donde se encuentren varios equipos de distintas marcas y modelos, funcionando en el mundo real (con tráfico, carga, clientes, etc.). Afortunadamente, el autor cuenta con varios clientes dispuestos a prestar sus redes sin costo para la consecución del objetivo planteado.

Proyección de Costos de Operación y Mantenimiento

Al tratarse de una adición a un producto existente, los costos de operación y mantenimiento serán absorbidos por este último. De todas formas, el espíritu del proyecto

apunta justamente a la reducción de estos costos por mayor eficiencia, eliminación de tareas manuales y reducción, por lo tanto, de posibles errores, motivos por los cuales los costos de operación y mantenimiento se esperan negativos.

Análisis de Viabilidad Comercial

Al tratarse de una adición a un producto existente, este punto no tiene sentido si analizamos únicamente el desarrollo planteado. Podríamos analizar la viabilidad comercial del producto en su conjunto, pero tal análisis está fuera de los alcances de este escrito. Si podemos mencionar que el producto en cuestión es altamente rentable, encontrándose hace ya 8 años en el mercado, contando con más de 350 instalaciones en actividad repartidas entre 7 países. El énfasis en la mejora de procesos y el aumento de la eficiencia que da origen a este trabajo, viene justamente de la mano de la inversión que pretende realizar la empresa para mejorar el funcionamiento interno y la escalabilidad de sus actividades de soporte.

Análisis Financiero

Consultar punto anterior.

Estudio Ambiental

Al tratarse de un proyecto de software, este punto no resiste un análisis profundo. Podríamos plantear que aumentar la eficiencia de este tipo de procesos seguramente aumente la eficiencia en la utilización de equipamiento informático y, por ende, reduzca el consumo eléctrico de los mismos. La diferencia obtenida, en realidad, no tiene suficiente importancia como para seguir profundizando en ese sentido, por más que se entienda que en la suma de estas pequeñas diferencias es en donde terminaremos obteniendo una diferencia significativa.

Estudio Social

El proyecto apunta a aumentar la eficiencia de un proceso interno de una empresa privada de alrededor de 10 personas. Podríamos, tal vez, plantear que contar con mejores herramientas permitirá a los ISPs brindar un mejor servicio y, de esa manera, pensar que el mejor acceso a internet de una parte de la sociedad es algo moralmente “bueno”, pero, al igual que en punto anterior, la diferencia no se supone significativa.

Por otro lado, la misión de la empresa si apunta, al menos en parte, en ese sentido y hemos tenido la satisfacción personal de ver comunidades revitalizadas luego de llevar internet a lugares geográficos alejados donde, en algunos casos, ni siquiera se contaba con acceso telefónico, pero, nuevamente, estas cosas tienen más que ver con la misión de la empresa en su conjunto que con el desarrollo puntual que motiva el presente escrito.

Evaluación Económica

Ver punto 12 (Beneficios Esperados).

Conclusiones

Punto por punto, la consecución de los objetivos fue de la siguiente manera:

Objetivo Específico 1: Crear los modelos necesarios para la captura de la información a utilizar. Este objetivo se alcanzó al **100%**, lo cual se ve reflejado en la creación de los distintos perfiles de detección, ejecución, etc; así como en la capacidad de representar el equipamiento del ISP y almacenar estadísticas sobre ellos.

Objetivo Específico 2: Crear un panel de administración web donde puedan configurarse y consultarse todos los modelos creados. Este objetivo se alcanzó al **100%**. Esto puede verse en las distintas capturas que acompañan al presente trabajo, la interfaz de usuario provista se accede desde un navegador web y permite la administración completa de la

totalidad de los modelos, como así también ejecutar las tareas de identificación y recolección o la visualización de las estadísticas.

Objetivo Específico 3: Permitir la ejecución de detecciones, de acuerdo a los parámetros configurados, y proveer un registro de los resultados de cada ejecución. Este objetivo se alcanzó al **100%**. La herramienta permite ejecutar una tarea de detección sobre un rango o rangos de direcciones IP, sobre los cuales identifica los equipos encontrados y crea automáticamente los modelos correspondientes, sin ninguna intervención del usuario.

Objetivo Específico 4: Mantener actualizado el listado de equipos conocidos con sus respectivos parámetros. Este objetivo se alcanzó al **100%**, y es una consecuencia del diseño de la herramienta de detección, la cual permite ejecuciones sucesivas sin destruir información anterior: si el perfil correspondiente a algún equipo cambia, esto se verá reflejado automáticamente en los modelos correspondientes, lo mismo ocurre si aparecen equipos nuevos, etc.

Objetivo Específico 5: Permitir la administración y configuración de perfiles de recolección de estadísticas en función de marca, modelo y versión de firmware de cada equipo conocido. Este objetivo se alcanzó al **100%**. Conociendo los equipos presentes en la red y la marca, modelos y versión de firmware de cada uno, la herramienta permite la recolección de estadísticas de todos ellos de forma automática. También se facilita el mantenimiento y la puesta a punto gracias a que la herramienta informa si alguna recolección deja de funcionar correctamente.

Objetivo Específico 6: Permitir la administración y configuración de perfiles de detección automática de equipamiento en función de una serie de comandos a ejecutar y el resultado esperado para cada uno. Este objetivo se alcanzó al **100%**. La herramienta permite cargar y reutilizar perfiles de detección que respetan la jerarquía planteada para optimizar tiempo y recursos, como así también mantener este listado actualizado a lo largo del tiempo de manera sencilla y eficiente, ya que cada perfil individual presenta poca complejidad, pudiéndose tener y administrar un número importante de ellos ordenado y por la mencionada jerarquía. Esto también facilita compartir la información entre distintas instalaciones de la herramienta.

Objetivo Específico 7: Permitir la visualización de las estadísticas recolectadas para los equipos conocidos. Este objetivo se alcanzó al **100%**. Las estadísticas que la herramienta va recolectando se almacenan de forma automática y eficiente en una base de datos round-robin, y luego se grafican las correspondientes curvas de forma automática cada vez que el usuario lo solicita. El panel de administración web permite buscar y ordenar para encontrar rápidamente el equipo que al usuario le interesa consultar, y ofrece los gráficos en cuestión con sólo un clic.

Por lo anteriormente expuesto, consideramos que nuestro **Objetivo general:** Generar el panel de estadísticas y mantener actualizado el listado de equipos a consultar de forma automática, se encuentra cumplimentado en un **100%**.

Es con satisfacción que llegamos a la conclusión de este trabajo. Los resultados obtenidos se consideran muy buenos, sobre todo teniendo en cuenta la relación entre el esfuerzo que implicó realizarlo contra el esfuerzo que esperamos ahorrarnos de ahora en adelante. La elección de las herramientas y tecnologías se comprueba buena también, ya que permitió el normal fluir del desarrollo sin mayores dificultades. Se agradece también la posibilidad de contar con redes “en el mundo real”, prestadas por nuestros clientes, para la realización de las pruebas y ajustes que resultaron necesarios. En adelante, consideramos que a medida que creamos nuevos perfiles de identificación y recolección, iremos obteniendo cada vez más valor de esta herramienta.

Bibliografía

Documentación Oficial de Django	https://docs.djangoproject.com/en/1.9/
Documentación Oficial de RRDTool	http://oss.oetiker.ch/rrdtool/doc/index.en.html
Documentación Oficial de 815Gateway	http://ochoquinque.com/sitio/815gateway.html

Anexo 2: Diagrama Entidad-Relación (DER)

