PROYECTO FINAL ASIR

David Del Río Pascual

ASIR

I.E.S. JULIÁN MARÍAS

Tabla de contenido

Introducción:

Objetivo:

La teoría de Darwin manifiesta que: “O te adaptas o mueres”. En el siglo XXI, las empresas han sufrido grandes cambios, pero el más evidente es la llamada transformación digital.

La transformación digital se refiere al uso de la tecnología en la empresa como uno de los pilares básicos del funcionamiento de la misma.

Para poder adaptarse a los nuevos modelos y formas de negocio del siglo XXI, no es suficiente que las empresas incorporen la tecnología, ya que ésta no produce por sí misma la digitalización. Es necesario un cambio generalizado en los procedimientos, rediseñar sus modelos subyacentes de negocio, así como sus métodos operativos.

La estrategia de la transformación digital permite mejorar la eficiencia en todos los procesos internos de la organización. Algunos de ellos son:

- Investigación y desarrollo
- Producción
- Ventas
- Marketing
- Gestión Humana
- Atención al cliente
- Calidad
- Finanzas
- ...

Desde mi punto de vista, se observan dos tipos claros de transformación digital:

- Transformación digital del lado del cliente.
- Transformación digital del lado de la empresa.

Como clientes, pedir cita para el médico desde una aplicación móvil, realizar un pedido a nuestra farmacia de confianza a través de WhatsApp, gestionar una incidencia con nuestra compañía telefónica a través de redes sociales o realizar todo tipo de compras online, son tareas que con frecuencia realizamos y que conllevan, en el lado de la empresa, un gran cambio en el modelo de negocio y en los procedimientos de la misma.

En las empresas existen tareas muy interesantes a llevar a cabo como ejemplo de transformación digital de procedimientos internos, si bien es cierto que el cliente no verá reflejado directamente en él mismo dichos cambios, se verá beneficiado de la mejora del rendimiento del negocio gracias a ellos, por lo que son imprescindibles.

En este proyecto nos vamos a centrar en los ejemplos de transformación digital del lado de la empresa, implantando múltiples servicios que potenciarán el funcionamiento de la misma.

Justificación:

En muchas de las ponencias sobre transformación digital a las que he asistido, se ha manifestado por parte de los empresarios participantes, múltiples quejas asociadas al alto coste de la misma.

Si bien es cierto que puede no ser barata, tampoco tiene que ser cara.

Desde mi punto de vista, creo que disponemos de muchas herramientas para afrontar la transformación digital del lado del cliente sin que suponga un gran desembolso para un negocio. Es difícil compaginar cambios de procedimientos a nivel interno sin relacionarlo directamente con un sustancial aumento del gasto.

Actualmente muchos empresarios desean cambiar su modelo de negocio y metodologías internas adaptándose a los nuevos tiempos; desembolsan grandes cantidades de dinero en Hardware (comprando grandes servidores e infrautilizándolos), Software (que en muchos casos conlleva gastos extras de mantenimiento), u otro tipo de soluciones que suponen un gran desembolso económico.

Este proyecto nace con la idea de implantar diversos procedimientos con el menor coste posible y el mayor número de ventajas.

La justificación principal del mismo es concienciar a las empresas de la posibilidad de realizar la transformación digital con las premisas del ahorro de costes en hardware, software y mantenimiento sin olvidar ni dejar de lado en ningún momento el disponer de una solución fiable, fácil de mantener y de gran calidad.

Personalmente elegí este proyecto para demostrar el potencial y las grandes funcionalidades que se pueden llevar a cabo en un ordenador de bajo coste como la Raspberry Pi.

Además, la realización del mismo me permitirá poner en práctica una gran parte de los conocimientos adquiridos durante los cursos de primero y segundo de ASIR.

Conocimientos adquiridos en ASIR usados en el proyecto

- Arquitectura de nuestro ordenador (Raspberry Pi)
- Instalación, gestión y administración de Software en Linux.
- Manejo y administración de repositorios.
- Gestión de ficheros.
- Manejo de consola de comandos.
- Gestión de usuarios y grupos.
- Administración de recursos.
- Principios de seguridad informática.
- Gestión de procesos e hilos de procesos.
- Administración de redes locales.
- Administración de redes nateadas.
- Administración remota Linux.
- Scripting en Linux.
- Gestión de copias de seguridad.
- Programación de tareas.
- …

Propuesta detallada:

Como bien he explicado anteriormente, el proyecto trata sobre la implantación de múltiples servicios informáticos, todos ellos reunidos en un mismo servidor. En este caso en el que nuestro principal objetivo es la Digitalización Low Cost, es imposible hablar de bajo coste y no pensar automáticamente en el mini ordenador nº1 por excelencia, la Raspberry Pi.

Nuestro objetivo en la realización de este proyecto es explotar al 100% las posibilidades de este ordenador, siendo conscientes de los recursos de los que disponemos.

Por si andáis un poco perdidos…

¿Qué es la Raspberry Pi?

Según Wikipedia, la Raspberry Pi es un computador de placa reducida, computador de placa única o computador de placa simple (SBC) de bajo coste, desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas.

Su sistema operativo oficial, el que usaremos en el proyecto, es Raspbian, una versión adaptada de Debian, aunque hemos de conocer que existen otras muchas alternativas, tales como: Ubuntu Mate, Snappy Ubuntu Core, Windows 10 IOT Core, OSMC, Libreelec, Pinet...

El diseño de la Raspberry Pi incluye un procesador central, un procesador gráfico, y memoria RAM. No incluye ni disco duro (ya que usa MicroSD), ni fuente de alimentación y tampoco carcasa.

Con el paso del tiempo, y visto el éxito que tuvo la primera Raspberry Pi, han salido diversas versiones, mejorando así las anteriores, dando lugar a los siguientes modelos expuestos según su orden de aparición:

- Raspberry Pi Model A+
- Raspberry Pi Model B
- Raspberry Pi Model B+
- Raspberry Pi 2 Model B
- Raspberry Pi 3 Model B

Si deseas conocer un poco más acerca de la Raspberry Pi aquí te dejo una serie de enlaces que he recopilado, y que considero muy interesantes para aprender un poco más de este magnífico dispositivo:

Web oficial: https://www.raspberrypi.org/

Foro de discusión oficial: https://www.raspberrypi.org/forums/

Wikipedia: https://es.wikipedia.org/wiki/Raspberry_Pi

Para la realización de este proyecto dispongo del último modelo de Raspberry Pi, que gracias a ser más potente que sus antecesoras, me permitirá llevar a cabo más tareas en ella.

RASPBERRY PI 3 MODEL B
Procesador de 4 Núcleos ARMv8 de 64-bit a 1.2GHZ
802.11n Wireless LAN
Bluetooth 4.1 de bajo consumo
1GB RAM
4 puertos USB
40 GPIO pins
1 Puerto HDMI
1 puerto Ethernet
Conector de audio combinado de 3,5 mm y vídeo compuesto.
Interfaz de Cámara (CSI)
Interfaz de Display (DSI)
Slot para memoria MicroSD
Procesamiento gráfico VideoCore IV 3D graphics core

¿Qué vamos a hacer en la Raspberry Pi?

A continuación, se listan las diversas acciones que vamos a realizar en nuestra Raspberry Pi:

1. Instalación de Sistema Operativo (Raspbian).
2. Configuraciones iniciales de Sistema Operativo.
3. Configuración de acceso remoto.
4. Servicio 1º: Sistema de videovigilancia (Motion).
5. Servicio 2º: Crear nube privada (NextCloud).
6. Servicio 3º: Sistema de monitorización de red.
7. Configuraciones Extra

Diseño de la propuesta:

Todos los servicios anteriormente mencionados podrán ser accesibles desde la Red Local. Además realizaremos un "nateo" de los mismos en el Router para que puedan ser utilizados a través de Internet y, por ejemplo, poder ver desde nuestra casa, el servicio de videovigilancia implantado.

Nateo o traducción de direcciones de red o NAT: Es un mecanismo utilizado por Routers IP para intercambiar paquetes entre dos redes que asignan mutuamente direcciones incompatibles.

En nuestro caso realizaremos una redirección de puertos para poder acceder a los servicios desde fuera.

Una redirección de puertos es la acción de redirigir un puerto de red de un nodo de red a otro. Esta técnica puede permitir que un usuario externo tenga acceso a un puerto en una dirección IP privada (dentro de una LAN) desde el exterior vía un Router con NAT activado.

La redirección de puertos permite que ordenadores remotos se conecten a un ordenador concreto dentro de una LAN privada.

Evaluación de costes

A continuación, se detallan los costes de realización del proyecto, no se han incluido los costes asociados al personal informático que desempeña estas tareas.

Resumen de costes:

Descripción	Unidades	Precio Unitario	Importe total
Raspberry Pi 3 model B	1	38,9862	38,9862
Rii Mini X1 teclado inalámbrico	1	19,1059	19,1059
Aukru Caja + Disipador de Claor	1	7,45	7,45
Cargador Micro USB 5V 3000mA	1	7,45	7,45
MicroSD HC1 16GB Samsung	1	8,2	8,2
		Importe Total	81,1921

Instalación y configuraciones iniciales de Raspbian:

Es necesario dotar de un sistema operativo a nuestra Raspberry Pi 3, por lo que accederemos al apartado de descargas de su Web para descargar Raspbian, el sistema operativo oficial de Raspberry https://www.raspberrypi.org/downloads/raspbian/

En nuestro caso descargaremos Raspbian en la siguiente versión:

RASPBIAN JESSIE WITH PIXEL
Image with PIXEL desktop based on Debian Jessie
Versión: Abril 2017
Fecha de lanzamiento: 10/04/2017

Es necesario disponer de una tarjeta microSD para instalar el S.O, que posteriormente introduciremos en la Raspberry Pi. En este proyecto usaré una tarjeta SAMSUNG de 16GB microSD HC I de clase 10.

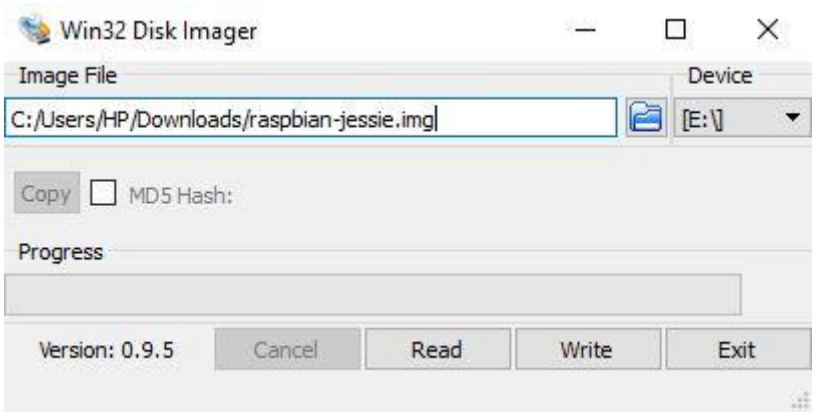
Instalación de Raspbian:

Una vez descargado Raspbian vamos a instalarlo en nuestra tarjeta microSD. En este caso usaremos el Software Win32DiskImager que podemos obtener desde el siguiente link:

https://sourceforge.net/projects/win32diskimager/

Win32DiskImager es un Software de uso muy sencillo, tan sólo es necesario introducir la tarjeta microSD en nuestro ordenador, seleccionarla en la opción "Device" y pulsar sobre el botón de la carpeta para elegir la imagen que vamos a instalar en el dispositivo.

Una vez seleccionada la imagen y el dispositivo, tan sólo es necesario pulsar sobre el botón Write para comenzar con la instalación.

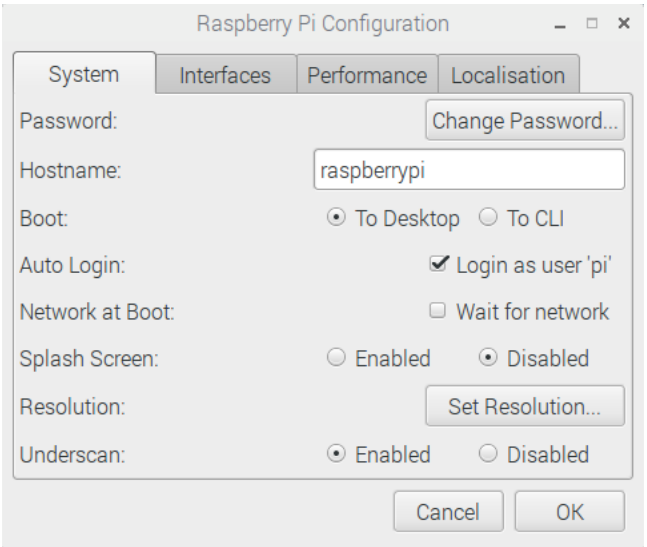


Configuraciones iniciales de Raspbian:

A continuación, vamos a comenzar a realizar las configuraciones iniciales más recomendadas para nuestro pequeño servidor.

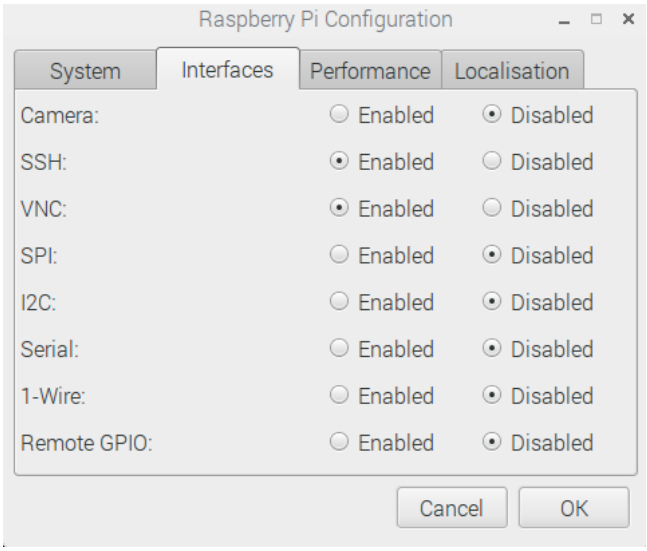
Desde el menú inicio iremos al apartado preferencias y una vez dentro haremos clic sobre "Configuración de Raspberry Pi".

Aparecerá un menú como el que se muestra en la imagen:



Es recomendable cambiar la contraseña para intentar mejorar la seguridad de nuestro sistema, la contraseña por defecto es "raspberry", nosotros la cambiaremos por la que se usará durante el resto del proyecto: "p@ssw0rd".

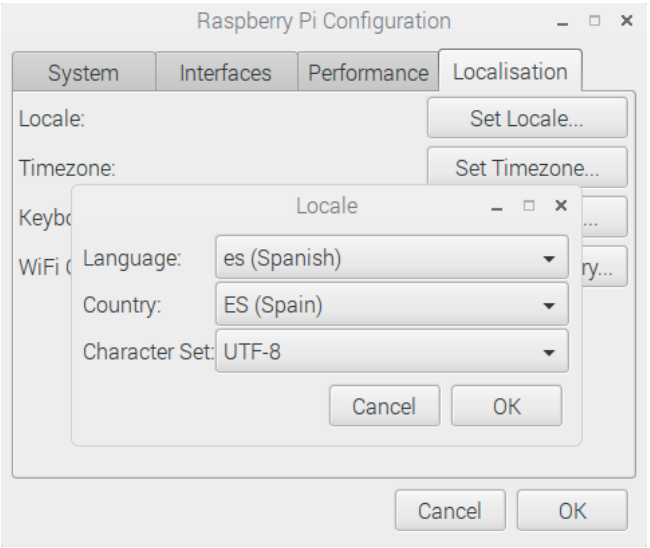
Una vez cambiada la contraseña accederemos al menú interfaces, en el cual podremos habilitar y deshabilitar lo que consideremos oportuno. Para este proyecto vamos a habilitar SSH y VNC por lo que el resultado del menú debería ser similar al siguiente:



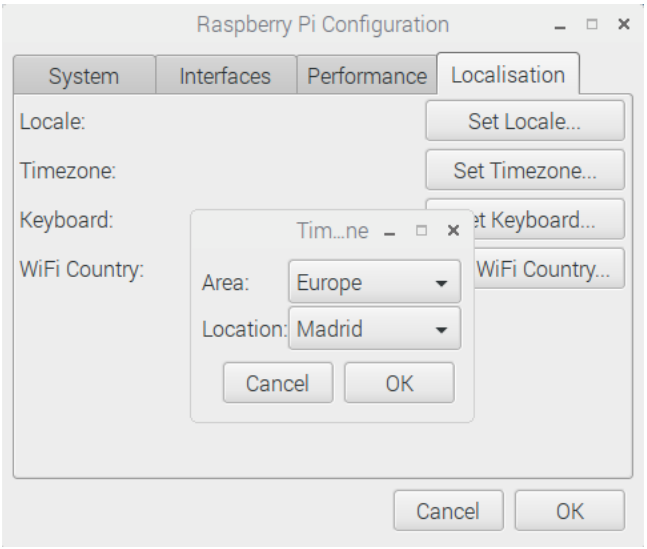
A continuación, pulsaremos sobre la pestaña configuración, en la que configuraremos diversas opciones regionales y de idioma.

Seguidamente se muestra el resultado de las diferentes parametrizaciones a realizar:

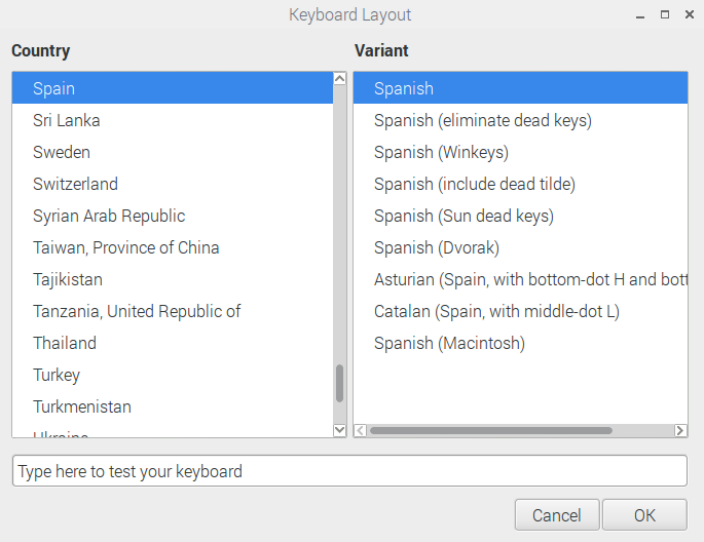
Configuración local:



Configuración de zona horaria:



Configuración de teclado:



El sistema nos pedirá un reinicio para aplicar los cambios, por lo que procederemos a realizarlo para continuar con la configuración.

Una vez reiniciado el sistema, deberíamos disponer del sistema operativo en el idioma español. A continuación, vamos a realizar una de las tareas más importantes a la hora de configurar nuestra Raspberry Pi 3, la expansión de la tarjeta microSD.

¿Qué significa expandir la tarjeta microSD? El sistema operativo instalado en nuestra tarjeta (después de flashear y arrancar la Raspberry Pi) solamente ocupa una pequeña parte del tamaño total de la tarjeta, ya que el S.O se instala en una partición con el espacio necesario (este espacio varía en función de la distribución flasheada).

Esto deja la mayoría de espacio de la tarjeta sin utilizar. Para poder aprovecharlo, expandiremos la partición para ocupar toda la tarjeta, de esta forma dispondremos de más espacio para la realización completa del proyecto y servicios que lo componen.

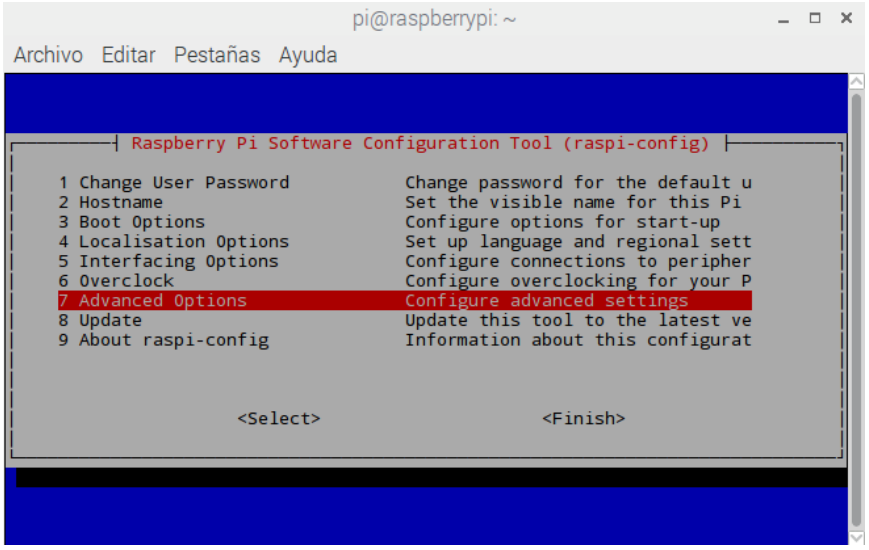
Desde una terminal ejecutaremos el siguiente comando:

CÓDIGO:

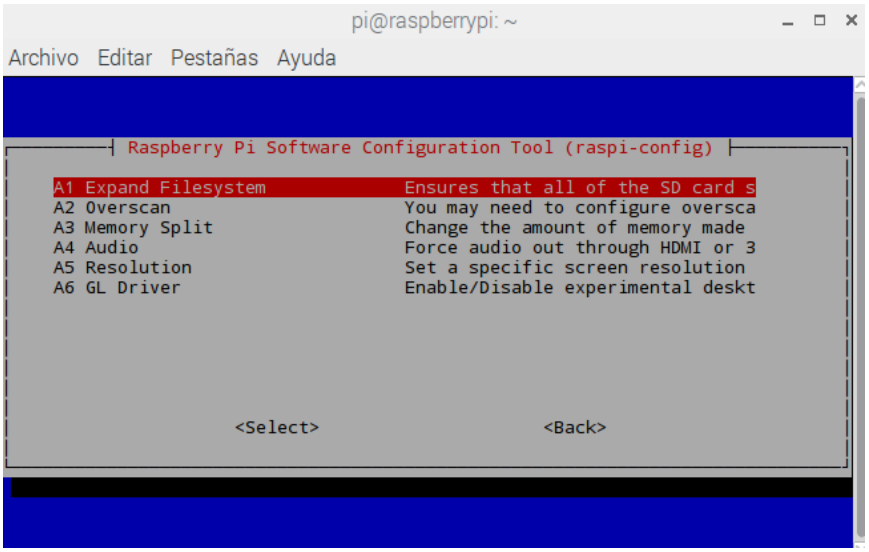
```
sudo raspi-config
```

Tras ejecutarlo, se nos abrirá la herramienta de configuración de Raspberry Pi, que, si bien es cierto que tiene muchas similitudes y semejanzas con el anterior menú gráfico, incluye ciertas opciones y características que el anterior carece de ellas, como la opción para expandir nuestra tarjeta.

Seleccionaremos el menú de opciones avanzadas:



Y elegiremos la opción “Expand Filesystem” para expandir nuestra tarjeta MicroSD.



Por último, el sistema nos pedirá un reinicio, tras ejecutarlo, podemos comprobar con el siguiente comando si la tarjeta MicroSD se ha expandido correctamente:

CÓDIGO:

```
sudo free
```

```
root@raspberrypi:/home/pi# free
             total       used       free     shared    buffers     cached
Mem:        945520     599756     345764      34664      57084     288280
-/+ buffers/cache:     254392     691128
Swap:       102396          0     102396
```

Estableciendo IP estática en Raspberry Pi 3:

Algo muy importante es tener siempre localizada en nuestra red la Raspberry, además nos facilitará la tarea de implantar diversos servicios tener una IP estática en Raspberry Pi 3.

Como primer paso, debemos tener en cuenta el tipo de conexión a Internet con la que nuestra Raspberry se conecta a la red. Abriremos una consola de comandos en la cual ejecutaremos el comando ifconfig:

CÓDIGO:

```
sudo ifconfig
```

Dicho comando nos permite conocer información sobre la red, concretamente los adaptadores de red de los que disponemos y, sí tenemos asignados o no dirección IP a alguno de los adaptadores.

Deberemos conocer el adaptador que queremos establecer con IP estática, en mi será "WLAN0".

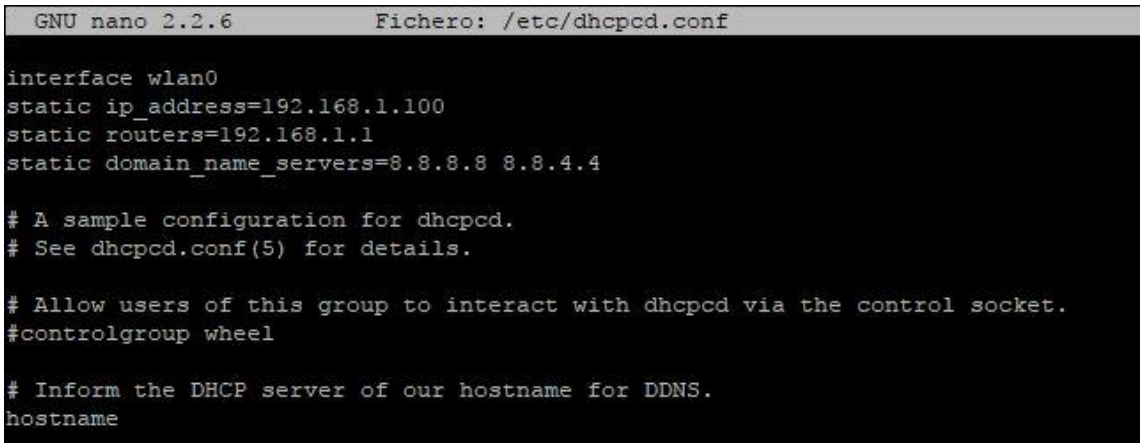
Para establecer una IP estática a nuestra Raspberry Pi vamos a editar el archivo /etc/dhcpcd.conf , para ello ejecutaremos el siguiente comando:

CÓDIGO:

```
sudo nano /etc/dhcpcd.conf
```

De tal manera que el archivo tendría que tener una estructura similar a la siguiente:

- static ip_address=(Aquí deberemos poner la dirección IP que queremos asignar)
- static routers=(Aquí deberemos poner la dirección IP de nuestro Router)
- static domain_name_servers=(Aquí deberemos poner las direcciones IP de los servidores DNS que deseamos usar, en este caso he puesto las dos de Google)

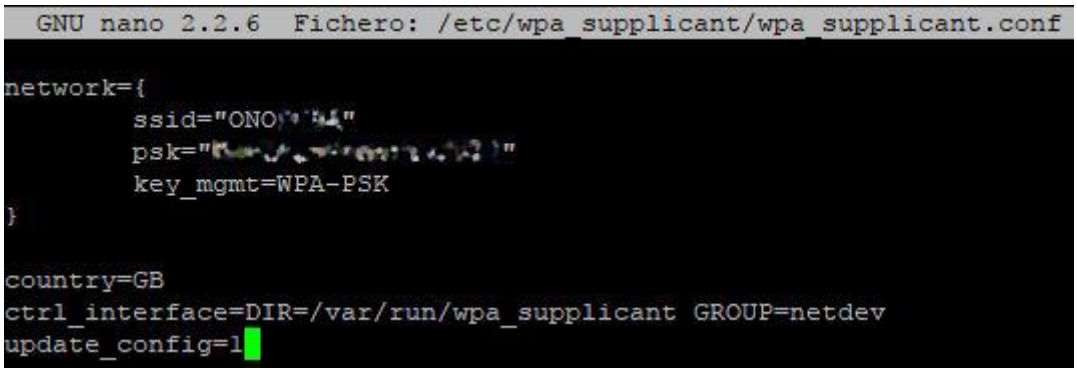


A continuación, vamos a editar el archivo /etc/wpa_supplicant/wpa_supplicant.conf , para ello ejecutaremos el siguiente comando:

CÓDIGO:

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

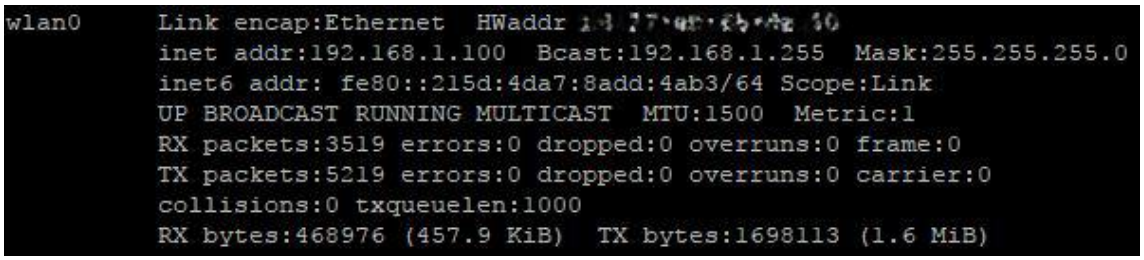
- ssid: deberemos añadir el nombre de nuestra red.
- psk: deberemos añadir la contraseña de nuestra red.
- key_mgmt: deberemos añadir el protocolo de seguridad que usar nuestra red.



Con un poco de suerte y tras reiniciar nuestro sistema ya tendremos nuestra IP estática. Tan sólo bastará ejecutar el comando ifconfig para comprobar que nuestra IP asignada es la que anteriormente habíamos establecido en el archivo de configuración.

CÓDIGO:

```
sudo ifconfig
```



Implantando múltiples servicios:

A continuación, vamos a configurar y poner en servicio en la Raspberry Pi 3 los siguientes servicios:

- Sistema de videovigilancia.
- Sistema de almacenamiento en la nube.
- Sistema de monitorización de red.
- Sistema de telefonía IP (VOIP)

Sistema de videovigilancia:

Introducción:

A continuación, vamos a implantar un sistema de videovigilancia a través de nuestro pequeño servidor, algo esencial hoy en día en la mayoría de los negocios, debido a la necesidad de proteger físicamente nuestro entorno.

Si bien es cierto que cada día la seguridad lógica cobra más y más sentido en nuestra vida, no podemos dejar de lado la seguridad física, por lo que, creo que no se puede hablar de seguridad física y no hablar de un sistema de videovigilancia. En este caso constará de una cámara de seguridad configurada para emitir a través de internet y poder ser visualizada desde cualquier lugar.

¿Qué vamos a usar?

Para la implantación de este sistema usaremos nuestro mini servidor Raspberry Pi 3, una cámara y el software "motion".

Puesto que ya hemos dedicado mucho tiempo a hablar de la Raspberry no indagaremos más en ella, pero sí dedicaremos un par de párrafos a la cámara y el software elegido para la implantación de dicho sistema.

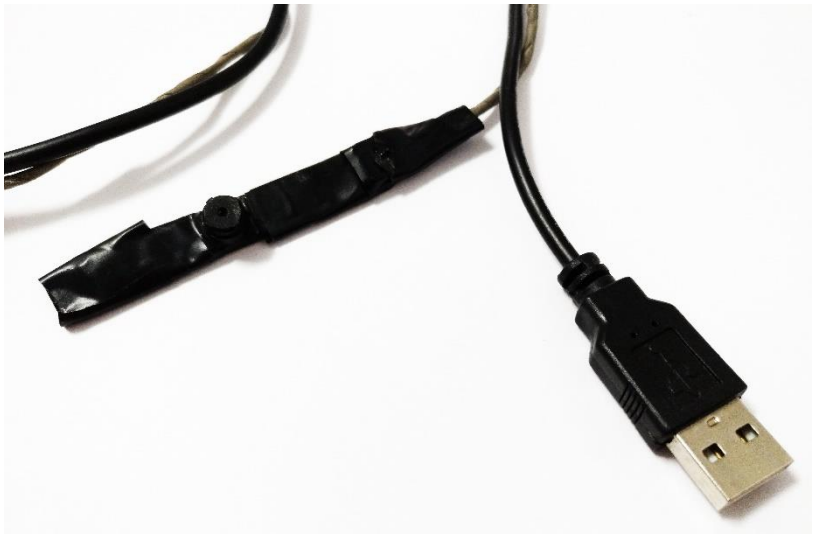
Cámara de videovigilancia

La cámara elegida para la implantación de un sistema como éste podría ser cualquiera que dispusiese de una conexión por USB, pero puesto que una de las principales premisas del proyecto es el ahorro de costes, vamos a usar una Webcam extraída de un antiguo ordenador portátil que ya no está operativo.

Una vez extraída la cámara, necesitamos un cable USB. Cortaremos dicho cable por la mitad y realizaremos las conexiones necesarias para transformar una cámara integrada a un portátil sin ninguna utilidad a una operativa cámara por USB.

Esquema de conexiones:

Imágenes de la cámara:



Cámara de seguridad con el cable USB.

Detalle de la cámara de seguridad.

Comparación tamaño de cámara con pila.

Software Motion

Motion es un programa que monitoriza la señal de vídeo de una o más cámaras y es capaz de detectar si una parte significante de una imagen ha cambiado. En otras palabras, es capaz de detectar movimiento.

El programa está escrito en C y hecho para el sistema operativo Linux.

Motion está basado en la línea de comandos. No tiene ninguna interfaz gráfica de usuario. Todo el setup se basa en archivos de configuración (archivos simples de texto que pueden ser editados con cualquier editor de texto plano).

Principales características de Motion:

- Opción de importar vídeo desde múltiples cámaras.
- Guardar imágenes cuando la señal de vídeo de la cámara detecte movimiento.
- Crear archivos de vídeo que contengan el evento cuando la cámara detecta movimiento.
- Ejecutar un programa externo cuando el movimiento es detectado.
- Ejecutar un programa externo al comienzo de un evento de varios movimientos detectados.
- Ejecutar un programa externo al final de un evento de varios movimientos detectados.
- Ejecutar un programa externo cuando una imagen es guardada.
- Streaming de vídeo en directo.
- Hacer fotografías de manera automatizadas o regulares en un intervalo de tiempo.
- Hacer fotografías de manera automatizadas o regulares en un intervalo de tiempo usando cron.
- Alimentar eventos a una base de datos MySQL, PostgreSQL o SQLite3.
- Configurable por el usuario y definido por el usuario en la pantalla.
- Visualización mediante una simple interfaz Web.
- Control automático de reducción de ruido en imagen.
- Altamente configurable la introducción de texto sobre las imágenes.
- Altamente configurable la definición de los nombres de archivos de las imágenes y archivos de vídeos guardados.

Hardware soportado por Motion:

Motion soporta la entrada de vídeo a través de dos tipos de recursos: Los dispositivos standard video4linux (/dev/video0) y las cámaras de red. Motion no dispone de drivers para cámaras. Si el dispositivo funciona correctamente con otro software común de vídeo, funcionará con Motion y viceversa. En ocasiones es conveniente primero hacer funcionar la cámara con otro software y después utilizar esas opciones de conexión con Motion.

Instalación y configuración de Software Motion

Para comenzar vamos a realizar una serie de comprobaciones básicas:

Comprobar que el sistema está actualizado:

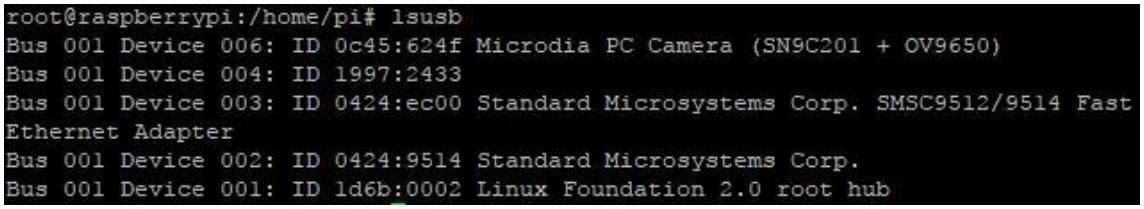
CÓDIGO:

```
sudo apt-get update

sudo apt-get upgrade
```

Comprobar que el sistema detecta la cámara:

CÓDIGO:

```
sudo lsusb
```



(Como vemos en la imagen, el sistema ha detectado en el Bus 001 con código de dispositivo 023 e identificador 0c45:624f nuestra cámara con el nombre Microdia PC Camera (SN9C201 + 0V9650).

Si nuestra Raspberry Pi ha detectado correctamente nuestra cámara podemos continuar con el siguiente paso, la instalación del Software Motion, para ello ejecutaremos el siguiente comando:

CÓDIGO:

```
sudo apt-get install motion
```

Una vez instalado motion vamos a proceder a editar su respectivo archivo de configuración: sudo nano /etc/motion/motion.conf. Editaremos el archivo configurándolo tal y cómo se muestra a continuación.

```
Daemon on
stream_localhost off
stream_maxrate 100
Width 640
Height 480
Framerate 100
Output_pictures off
ffmpeg_output_movies off
```

Seguidamente, vamos a configurar la autenticación para aportarle un extra de seguridad a nuestro sistema de videovigilancia. Para ello, y trabajando sobre el mismo archivo editaremos las siguientes líneas estableciéndolas tal y cómo se muestra en la imagen.

```
# Set the authentication method (default: 0)
# 0 = disabled
# 1 = Basic authentication
# 2 = MD5 digest (the safer authentication)
stream_auth_method 2

# Authentication for the stream. Syntax username:password
# Default: not defined (Disabled)
stream_authentication administrador:p@ssw0rd2017
```

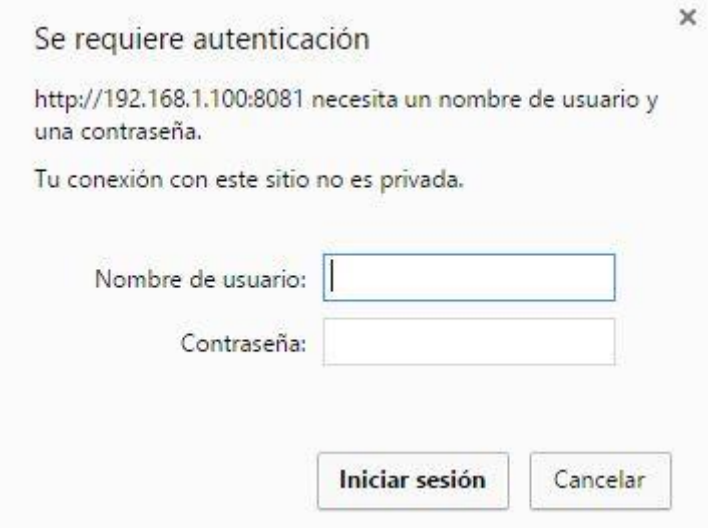
Por último, iniciaremos el servicio motion con el siguiente comando:

CÓDIGO:

```
sudo motion -n
```

Para comprobar el correcto funcionamiento de nuestra cámara de seguridad nos bastará con acceder a la dirección http://192.168.1.100:8081/ (Es decir la dirección IP de la Raspberry y el puerto que usa el Software Motion, en nuestro caso el puerto 8081. Dicho puerto puede configurarse a través del archivo de configuración /etc/motion/motion.conf.

Al acceder a la página Web se nos solicitarán nuestras credenciales de acceso al sistema:



Demostración autenticación

Una vez introducidas nuestras credenciales de acceso podremos ver el Streaming de nuestro sistema de videovigilancia.

Demostración funcionamiento cámara de seguridad

Debido a un problema arbitrario que hace que el servicio motion se congele puntualmente crearemos un script llamado *reiniciarmotion.sh* para usarlo en caso de que lo detectemos. El script contiene las siguientes líneas:

CÓDIGO:

```
proceso=`ps -aux | grep motion | head -n 1 | tr -s " " ":" | cut -d ":" -f2`

kill $proceso | sleep 10 |  motion -n
```

Configuración aplicación Android:

Como tarea complementaria vamos a instalar en nuestro dispositivo Android una aplicación que nos permita visualizar la cámara de seguridad sin necesidad de acceder a la misma a través del navegador Web.

Desde el mercado de apps de Android (Play Store) buscaremos la aplicación “IP Cam Viewer Basic”.

Añadiremos el tipo de cámara, en nuestro caso seleccionaremos “URL jpeg/mjpeg genérica”:

Configuraremos los datos referentes estableciéndoles tal y cómo se muestran en la siguiente imagen:

Pulsaremos sobre el botón test para comprobar la conexión con nuestra cámara de videovigilancia, debiendo obtener un mensaje de confirmación de la conexión. Para finalizar si volvemos a la página principal de la aplicación comprobaremos como obtenemos a la perfección la imagen de la cámara de seguridad:

Demostración visualización de vídeo desde App

Almacenamiento en la nube:

Introducción:

Proveniente del Inglés Cloud Storage, el almacenamiento en la nube es un modelo de almacenamiento de datos en el cual la información está alojada generalmente en múltiples servidores y en ocasiones en múltiples localizaciones.

Por lo general el entorno físico es administrado por una empresa de alojamiento. Los proveedores del almacenamiento en la nube son responsables de mantener los datos disponibles y accesibles y el entorno físico protegido.

Ventajas del almacenamiento en la nube:

1. Acceder a tus documentos - Desde cualquier lugar.
2. Trabajar con otros - Desde cualquier lugar.
3. Copias de seguridad de los archivos.
4. Seguridad de los datos.
5. Ahorro de costes.

Acceso a los datos: La nube nos permite acceder a los datos desde cualquier lugar, generalmente accederemos a ellos mediante un software específico, una aplicación o una página Web.

Trabajar con otros: El almacenamiento en la nube nos permitirá trabajar con otras personas compartir archivos con grupos o usuarios concretos con los que deseamos trabajar conjuntamente.

Copias de seguridad de los datos: La mayoría de los proveedores de almacenamiento en la nube disponen de servicios de copias de seguridad para garantizar la disponibilidad de nuestra información.

Seguridad de los datos: Los proveedores de servicio garantizan la seguridad de nuestros datos y la disponibilidad de los mismos.

Ahorro de costes: Reducción de la inversión inicial, reducción de facturas de consumos energéticos, reducción de costes de mantenimientos y gestión, reducción de costes de hardware…

Nextcloud. ¿Qué es?

Es un proyecto de software libre, creado inicialmente por el mismo creador de Owncloud, Frank Karlitschek, con el objetivo de que los usuarios recuperen el control sobre sus datos. La finalidad del producto es proporcionar a las organizaciones y a los particulares un control sobre su información y datos, facilitando la sincronización y el intercambio de ficheros entre dispositivos. Además, incorpora otras herramientas que permiten comunicarse por audio y vídeo vía WebRTC de manera segura.

Características de Nextcloud:

- Software libre.
- La seguridad como prioridad.
- Gestionar el flujo de trabajo.
- Cliente para dispositivos móviles o de escritorio.
- Posibilidad de almacenamiento externo.
- Calendario y agenda de contactos.
- Llamadas de audio y vídeo seguras.
- Integración con Active Directory, LDAP, Kerberos…
- Contraseñas integradas.
- Cuotas de usuario.
- Monitorización de la actividad del servidor.
- Trackeo de los cambios en archivos.
- Visualización y edición de documentos con Collabora.
- Apps propias de NextCloud.
- Interfaz amigable.
- Previews de archivos.
- Fácil personalización y configuración.

Instalación y configuración de Nextcloud.

Vamos a proceder a instalar Nextcloud en nuestra Raspberry Pi usando conjuntamente Apache2, PHP7 y MariaDB para la base de datos.

Podemos configurar NextCloud con las siguientes bases de datos:

- SQLite.
- Mysql.
- MariaDB.
- PostgreSQL.

Usaremos una de las combinaciones más recomendadas: MariaDB + APACHE2 + PHP7.

Instalación y configuración de Apache, PHP 7 y sus módulos correspondientes:

Para poder instalar correctamente PHP 7 y sus módulos, es necesario editar el archivo /etc/apt/sources.list por lo que ejecutaremos el siguiente comando:

CÓDIGO:

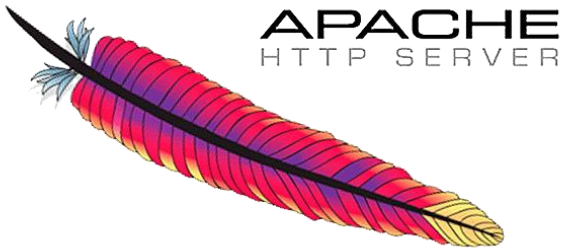
```
sudo nano /etc/apt/sources.list
```

Debajo de las entradas ya existentes en el archivo añadiremos las siguientes:

CÓDIGO:

```
deb http://repozytorium.mati75.eu/raspbian jessie-backports main contrib non-free
#deb-src http://repozytorium.mati75.eu/raspbian jessie-backports main contrib non-free
```

A continuación, necesitamos añadir un par de certificados para permitirnos usar los recursos con apt-get. Usaremos los siguientes comandos:



CÓDIGO:

```
sudo gpg --keyserver pgpkeys.mit.edu
--recv-key CCD91D6111A06851
sudo gpg --armor --export CCD91D6111A06851 | sudo apt-key add -
```

Por último, necesitamos actualizar los paquetes ejecutando el siguiente comando:

CÓDIGO:

```
sudo apt-get update
```

A continuación, instalaremos apache2, mariadb-server y libapache2-mod-php7.0

CÓDIGO:

```
apt-get install apache2 mariadb-server libapache2-mod-php7.0
```

Después de la instalación de MariaDB server, el software pedirá la creación de una contraseña de root, en nuestro caso usaremos la contraseña 'p@ssw0rd'. Es importante recordarla ya que necesitaremos conocerla durante la configuración de base de datos de Nextcloud.

Ahora ejecutaremos el siguiente comando, que nos va a permitir instalar los siguientes paquetes: php7.0-gd php7.0-json php7.0-mysql php7.0-curl php7.0-mbstring.

CÓDIGO:

```
sudo apt-get install php7.0-gd php7.0-json php7.0-mysql php7.0-curl php7.0-mbstring
```

Es necesario instalar php5-imagick, ejecutaremos el siguiente comando:

CÓDIGO:

```
sudo apt-get install php5-imagick
```

Para finalizar con la instalación de PHP7 instalaremos los siguientes paquetes: php7.0-intl php7.0-mcrypt php7.0-xml php7.0-zip

CÓDIGO:

```
sudo apt-get install php7.0-intl php7.0-mcrypt php7.0-xml php7.0-zip
```

Descarga de NextCloud y configuración de Apache.

Crearemos la siguiente carpeta.

CÓDIGO:

```
sudo mkdir /var/www
```

Nos dirigimos a la carpeta anteriormente creada.

CÓDIGO:

```
cd /var/www
```

Vamos a proceder a descargar e instalar la última versión de Nextcloud, por lo que la descargaremos de los repositorios oficiales:

CÓDIGO:

```
sudo wget https://download.nextcloud.com/server/releases/nextcloud-10.0.0.zip
```

Lo descomprimiremos con el siguiente comando:

CÓDIGO:

```
sudo unzip nextcloud-*.zip
```

Una vez haya concluido la descompresión, borraremos el archivo descargado:

CÓDIGO:

```
sudo rm nextcloud-*.zip
```

Ahora mismo deberíamos tener ubicada la carpeta de Nextcloud en /var/www/nextcloud

Editaremos el siguiente archivo relativo a Apache, referenciando la ruta hacia la carpeta Nextcloud:

CÓDIGO:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Editaremos el archivo añadiendo los siguientes parámetros justo debajo de <VirtualHost*:80>

```
Alias /nextcloud "/var/www/nextcloud/"
<Directory /var/www/nextcloud/>
 Options +FollowSymlinks
 AllowOverride All
SetEnv HOME /var/www/nextcloud
SetEnv HTTP_HOME /var/www/nextcloud
```

```
<IfModule mod_dav.c>
 Dav off
</IfModule>
</Directory>
```

Por último, editaremos la siguiente línea de texto:

```
DocumentRoot /var/www/html
```

Lo modificaremos por:

```
DocumentRoot /var/www/nextcloud
```

Guardamos los cambios y salimos de la edición del fichero.

A continuación, estableceremos los permisos a Apache sobre la carpeta Nextcloud con el siguiente comando:

CÓDIGO:

```
sudo chown www-data:www-data -R /var/www/nextcloud
```

Vamos a activar los módulos necesarios para Apache, introduciendo uno por uno los siguientes comandos:

CODIGO: a2enmod rewrite
CODIGO: a2enmod headers
CODIGO: a2enmod env
CODIGO: a2enmod dir
CODIGO: a2enmod mime
CODIGO: a2enmod setenvif

Para guardar y que se recojan correctamente los cambios reiniciaremos Apache:

CÓDIGO:

```
sudo service apache2 restart
```

Creación de la base de datos:

Para acceder a MariaDB usaremos el mismo comando que para iniciar MYSQL:

CÓDIGO:

```
sudo mysql -u root -p
```

Solicitará una contraseña, introduciremos la que pusimos en el proceso de instalación de MariaDB, en caso de no haberla establecido previamente, la contraseña por defecto es: “mariadb”.

Una vez dentro del sistema ejecutaremos los siguientes comandos para crear la base de datos:

Creación de base de datos:

CÓDIGO:

```
create Database nextcloud;
```

Creación de usuario:

CÓDIGO:

```
create user david@localhost identified by ‘p@ssw0rd’;
```

Concesión de privilegios al usuario para que pueda acceder a la base de datos:

CÓDIGO:

```
grant all privileges on nextcloud.* to david@localhost identified by ‘p@ssw0rd’;
```

Refrescamos los privilegios:

CÓDIGO:

```
flush privileges;
```

Saldremos del panel MariaDB ejecutando el siguiente comando:

CÓDIGO:

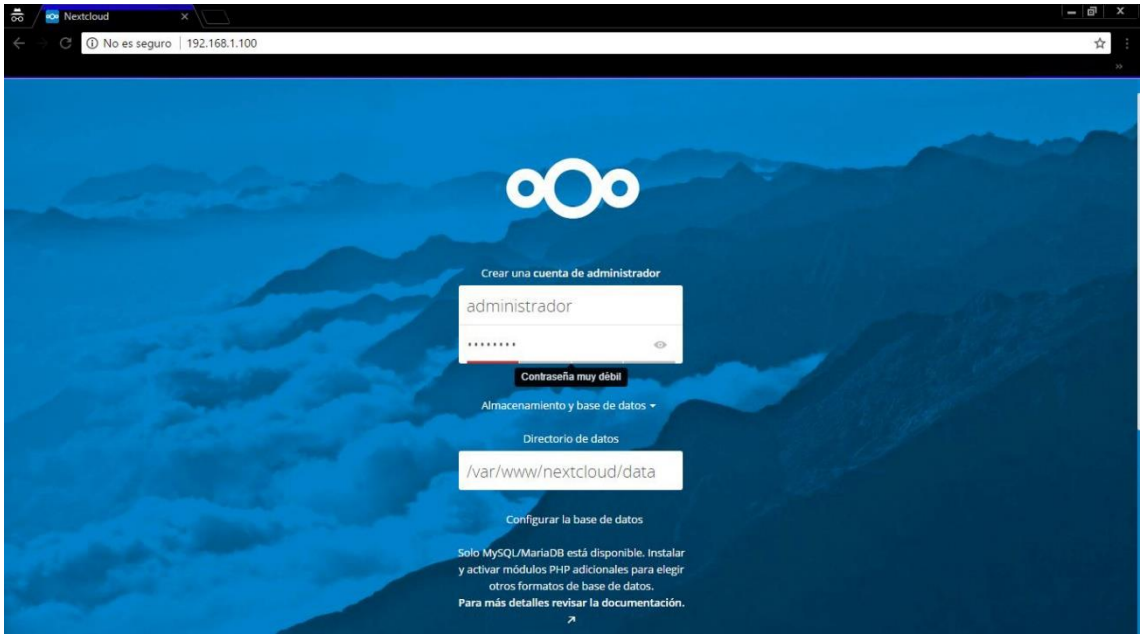
```
\q
```

Configuración inicial de NextCloud:

Una vez creada nuestra base de datos, entraremos a nuestro navegador Web poniendo la dirección IP de nuestra Raspberry PI:

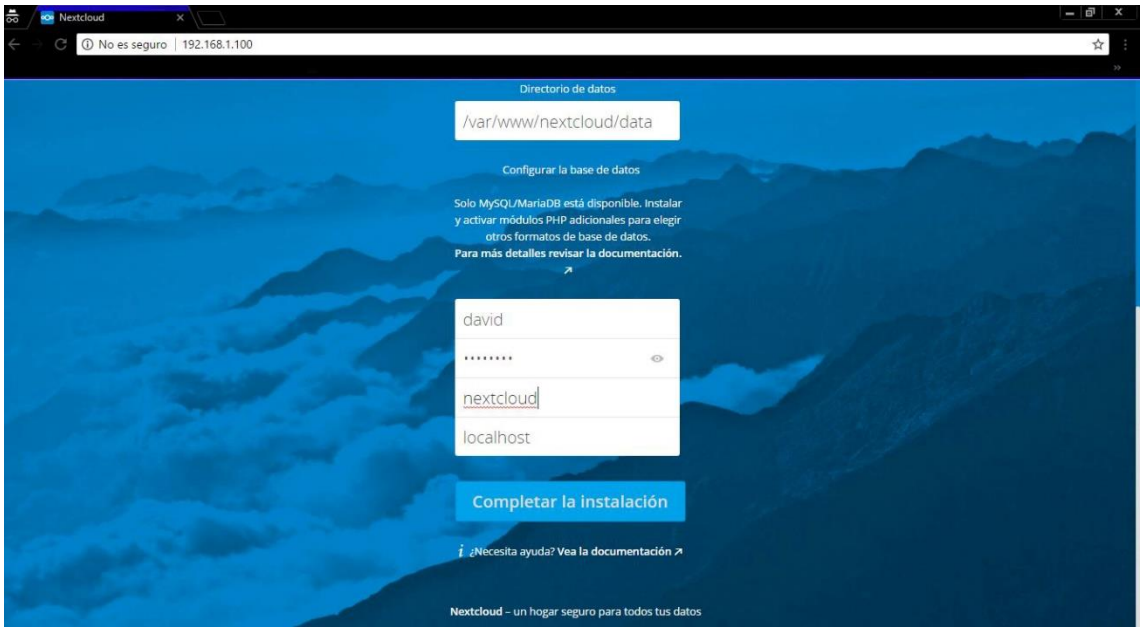
EJEMPLO:

http://192.168.1.100



Demostración funcionamiento Nextcloud

Una vez introducida la URL con la dirección IP de nuestra Raspberry, debemos crear una cuenta de administrador, como nombre de usuario estableceremos “administrador” y como contraseña estableceremos “p@ssw0rd”. Esta cuenta de usuario es la que utilizaremos para entrar en nuestro NextCloud.



En directorio de datos, seleccionaremos la carpeta donde queremos guardar todos nuestros archivos, por defecto:

CÓDIDO:
/var/www/nextcloud/data

Si queremos guardar los datos en un disco duro, dirigiremos la ruta hacia la carpeta automontada del disco, ejemplo /media/disco1.

Ahora añadiremos los datos con los que anteriormente creamos la base de datos:

USUARIO BASE DE DATOS:
david

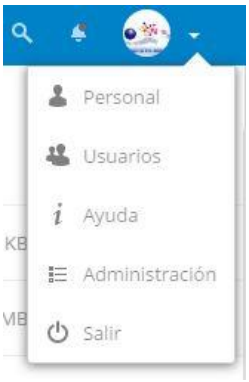
CONTRASEÑA DE LA BASE DE DATOS:
p@ssw0rd

NOMBRE DE LA BASE DE DATOS:
nextcloud

HOST DE LA BASE DE DATOS:
Localhost

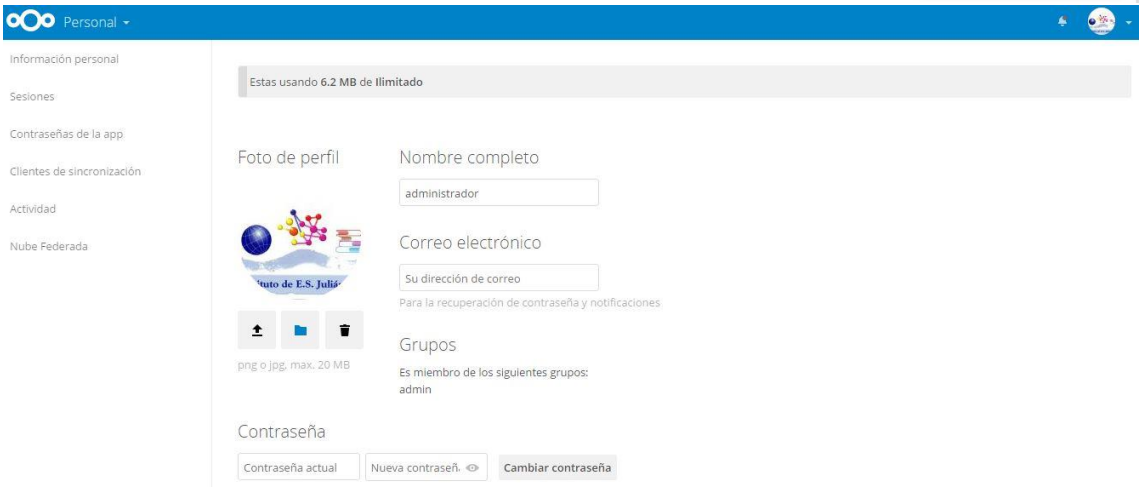
Pulsaremos sobre “Completar la instalación”, esperaremos unos minutos mientras se crean las tablas necesarias para crear el sistema y accederemos automáticamente a nuestro NextCloud.

Configuraciones iniciales de Nextcloud:



En la esquina superior derecha haremos clic en “administrador” y a continuación en “personal”.

En el usuario administrador podemos establecer un nombre completo y un correo electrónico para la recuperación de contraseña y notificaciones. También podremos restablecer nuestra contraseña.



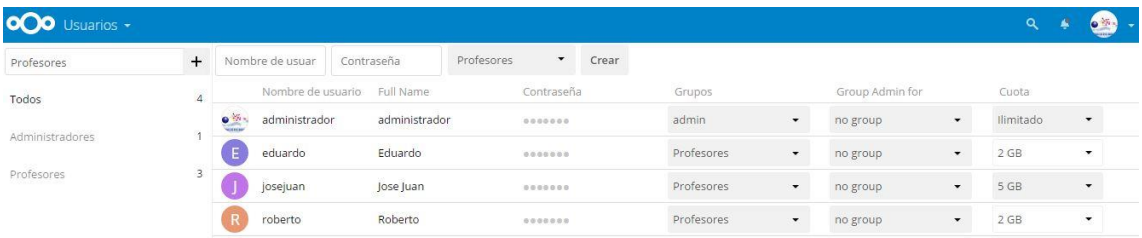
Desde esta pantalla también podemos observar un apartado “Sesiones” en el cual observaremos las sesiones actualmente conectadas a la cuenta.

También existe un apartado muy interesante denominado “Actividad” en el cual podemos establecer cuándo y cómo se nos notifican los cambios y actividades referentes a los archivos. Se podrá configurar la frecuencia con la que se envían los correos electrónicos.

Al final de esta pantalla podremos ver nuestra ID de Nube Federada, en nuestro caso administrador@192.168.1.100

Si volvemos al menú, podemos observar el apartado “Usuarios”, en el cual trataremos todo lo referente a la gestión de usuarios y grupos de Nextcloud. Vamos a crear el grupo profesores, en el que como usuarios de ejemplo añadiremos los siguientes: “José Juan”, “Eduardo” y “Roberto”.

Una vez creados los usuarios y asignarlos al grupo configuraremos una cuota de uso máximo de disco por usuario, estableciendo así 2 GB a “Eduardo” y “Roberto”, y 5 GB a “José Juan”.



En el menú también podemos acceder a la sección “Ayuda”, en la cual tenemos a nuestra disposición toda la documentación oficial referente a Nextcloud.

Por último, accederemos posiblemente al menú más importante, el de administración. Una vez dentro recorreremos las pestañas que lo componen.

En la pestaña “Server settings” se muestra Cron, el programador de tareas, permite ejecutar tareas a través de AJAX, Webcron y Cron.

Ajax es la opción predeterminada, desafortunadamente es el sistema menos confiable, cada vez que un usuario visita la página Nextcloud, se ejecuta un solo trabajo en segundo plano. La ventaja de este mecanismo es que no requiere acceso al sistema ni registro con un servicio de terceros. La desventaja reside en la comparación con el servicio Webcron, que requiere visitas regulares a la página para que se active.

Webcron apunta a un servicio externo Webcron ,como por ejemplo; https://www.easycron.com/ en el que asegurarás que los trabajos en segundo plano se ejecutarán regularmente. Para usar este tipo de servicio con el servidor, debemos ser capaz de acceder al servidor usando internet. Por ejemplo:

URL a introducir: http[s]://<dominio-del-servidor>/nextcloud/cron.php

Usar la característica Cron del sistema operativo es el método preferido para ejecutar tareas regulares. Este método habilita la ejecución programada de tareas sin las inherentes limitaciones que el servidor Web pudiera tener.

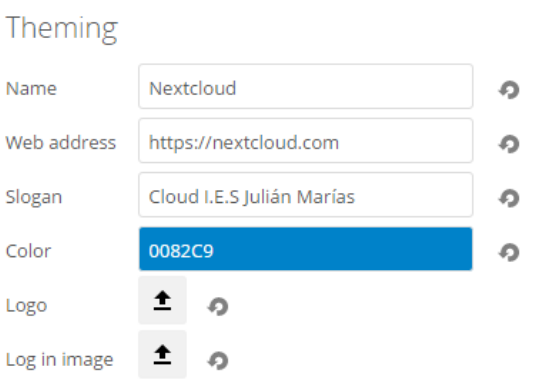
En la pestaña “Server Settings” podemos también ver la versión que disponemos de Nextcloud y el canal de actualización pudiendo elegir entre las opciones stable, daily, beta y production.

La pestaña “Server Info” permite ver la carga de la CPU de nuestro servidor, así como el uso de memoria. También podemos ver datos como los usuarios activos, el almacenamiento, la versión de PHP y de la base de datos. Podemos monitorizar dicha información desde una herramienta externa usando la siguiente url:

http://192.168.1.100/ocs/v2.php/apps/serverinfo/api/v1/info

La pestaña “sharing” nos permite gestionar lo relacionado con los archivos y el “Cloud Federado” (La federación nos permite conectarnos con otros servidores de confianza para intercambiar directorios).

La pestaña “theming” nos permite gestionar lo referente al tema en uso de Nextcloud y ciertos parámetros de personalización:



Si lo configuramos correctamente podemos conseguir un resultado de personalización corporativo similar al siguiente:

Llegamos a una de las pestañas más interesantes del menú de administración, la pestaña "encryption" que nos permitirá cifrar los datos de nuestra nube privada.

Pulsaremos sobre "Habilitar cifrado en el servidor". El sistema nos advierte que la encriptación por sí sola no garantiza la seguridad del sistema, que aumenta el tamaño de los archivos y que siempre debemos tener una copia de nuestros datos. Habiéndolas leído pulsaremos sobre el botón "Habilitar cifrado".

A continuación, vamos a habilitar un módulo de cifrado en el menú de aplicaciones. Lo activaremos para continuar el proceso.

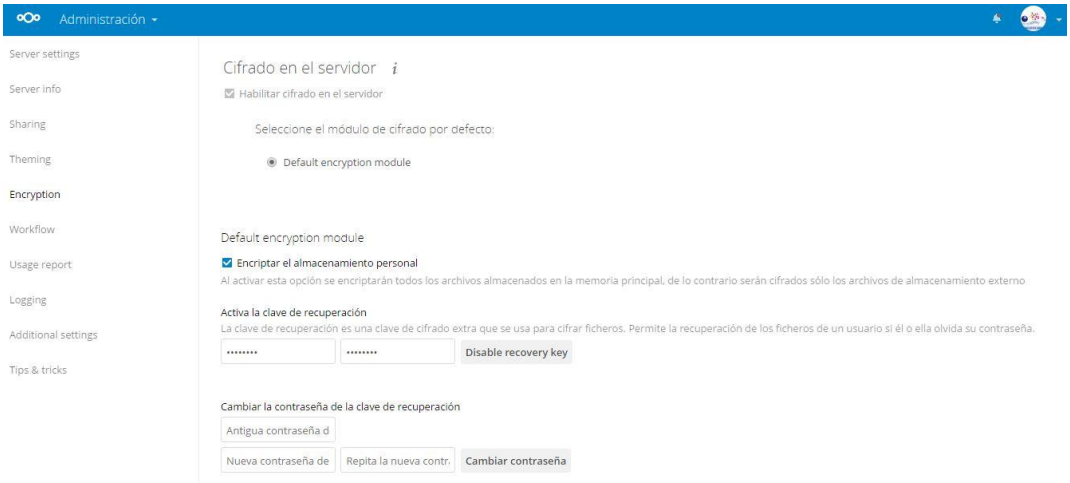
Módulo encriptación de Nextcloud

Volveremos al menú administración, a la pestaña "Encryption", en ella veremos que ahora aparece el módulo que acabamos de activar en nuestro servidor. El sistema nos advierte que la aplicación de encriptación está habilitada pero las claves no están inicializadas, para inicializarlas tan sólo debemos de salir del sistema y volver a iniciar sesión.

Una vez hecha dicha acción, y habiendo vuelto al menú Administración>Encryption , comprobaremos que sale una casilla con un tic de verificación que establece que se encriptará el almacenamiento personal. Al activar esa opción se encriptarán todos los archivos almacenados en la memoria principal.

Por último, vamos a habilitar las claves de recuperación de ficheros. Si uno de los usuarios pierde su contraseña de acceso a Nextcloud, sus archivos serían irrecuperables. Para evitar esto se creará una clave de recuperación. Si accedemos a la sección Encryption de la página de administración podremos establecer dicha clave.

Tras todos los cambios realizados en el apartado Encryption, tendremos nuestra nube encriptada, y la configuración debería ser similar a la siguiente:

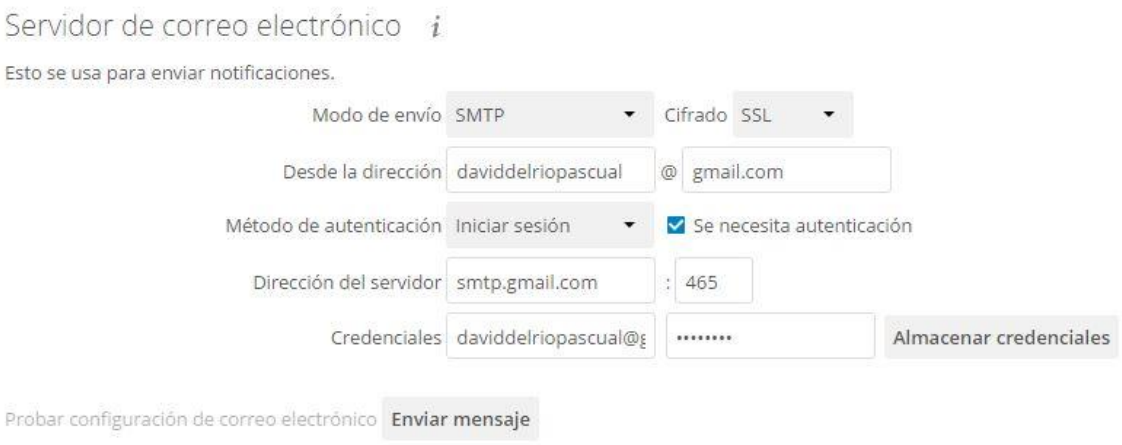


El apartado Workflow del menú administración nos permitirá usar etiquetas colaborativas para gestionar permisos de acceso a ficheros.

La pestaña “Usage report” permitirá mandar datos anónimos de uso a Nextcloud, nosotros obviaremos esta opción.

La pestaña logging permite llevar un registro de los errores que suceden en el sistema.

Desde la pestaña “additional settings” podremos configurar acciones tan importantes como el envío de correos electrónicos para las notificaciones. Será necesario establecer ciertos parámetros tales como la autenticación, servidor SMTP, datos de autenticación…



Desde el apartado “Additional settings” también podremos configurar el tamaño máximo de subida de ficheros, que lo estableceremos en 200MB. Con PHP-FPM podrían tardar hasta 5 minutos en surgir efecto los cambios en el servidor.

Por último, podemos configurar las políticas de contraseñas, aunque nosotros no lo haremos, ya que durante todo el proyecto usaremos la contraseña “p@ssw0rd”.

Tras haber finalizado todas las anteriores opciones habremos terminado de configurar Nextcloud.

Copias de seguridad Nextcloud:

A continuación, vamos a implementar unos scripts que automáticamente gestionarán copias de seguridad.

Siguiendo las premisas de ahorro de costes, usaremos un disco duro antiguo de un portátil acoplado a una carcasa por conexión USB como dispositivo hardware donde se almacenarán las copias.

Lo primero que debemos de hacer es formatear nuestro disco duro. Ejecutaremos el siguiente comando para conocer los dispositivos de almacenamiento de los que dispone el sistema.

CÓDIGO:

```
sudo fdisk -l | grep /dev/
```

Si observamos la siguiente imagen podemos ver cómo se detectan los discos “/dev/mmcblk0” (la tarjeta microSD de la Raspberry Pi) y “/dev/sda” (el disco duro que he conectado mediante USB en el que realizaremos las copias).

```
Disk /dev/mmcblk0: 14,7 GiB, 15719727104 bytes, 30702592 sectors
/dev/mmcblk0p1        8192   137215   129024    63M  c W95 FAT32 (LBA)
/dev/mmcblk0p2      137216 30702591 30565376 14,6G 83 Linux
Disk /dev/sda: 74,5 GiB, 80026361856 bytes, 156301488 sectors
```

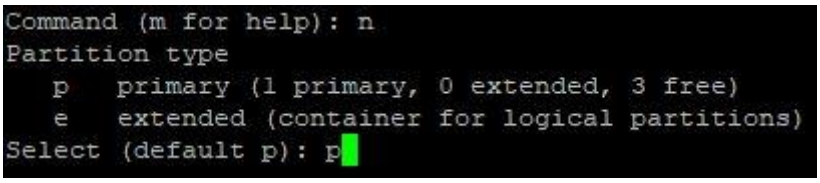
A continuación, ejecutaremos el siguiente comando para seleccionar el disco /dev/sda:

CÓDIGO:

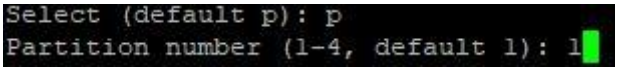
```
sudo fdisk /dev/sda
```

```
root@raspberrypi:/home/pi# fdisk /dev/sda

Welcome to fdisk (util-linux 2.25.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help):
```

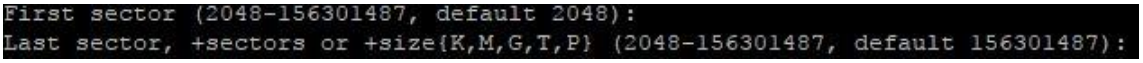
Pulsaremos la tecla “n” para crear una nueva partición y elegiremos la opción “p” (primaria):



Escribiremos “1” ya que es la primera partición que creamos en el disco duro.

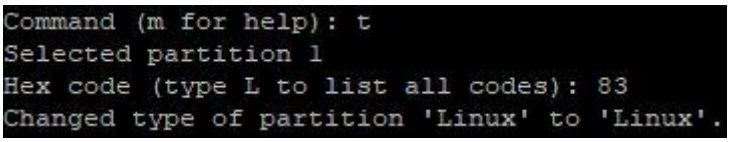


A continuación, estableceremos los valores por defecto en lo relativo a los sectores del disco.

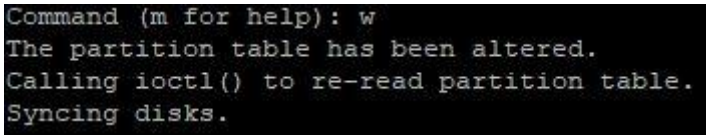


Pulsaremos la tecla “t” para establecer el tipo de partición. Puesto que la vamos a formatear en el formato de archivos “Ext.4” estableceremos “Linux” como tipo de partición. Para conocer el listado de tipos de particiones y sus códigos hexadecimales asociados tan sólo bastará con ejecutar la tecla “L”.

En nuestro caso estableceremos como código el “83”, el código Hexadecimal de “Ext4”.



Para finalizar con la herramienta de particionado “fdisk”, teclearemos “w” para escribir los cambios en la tabla de particionado.

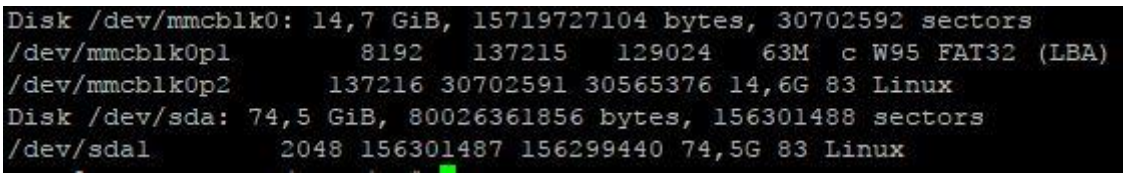


Si volvemos a ejecutar el siguiente comando:

CÓDIGO:

```
sudo fdisk -l | grep /dev/
```

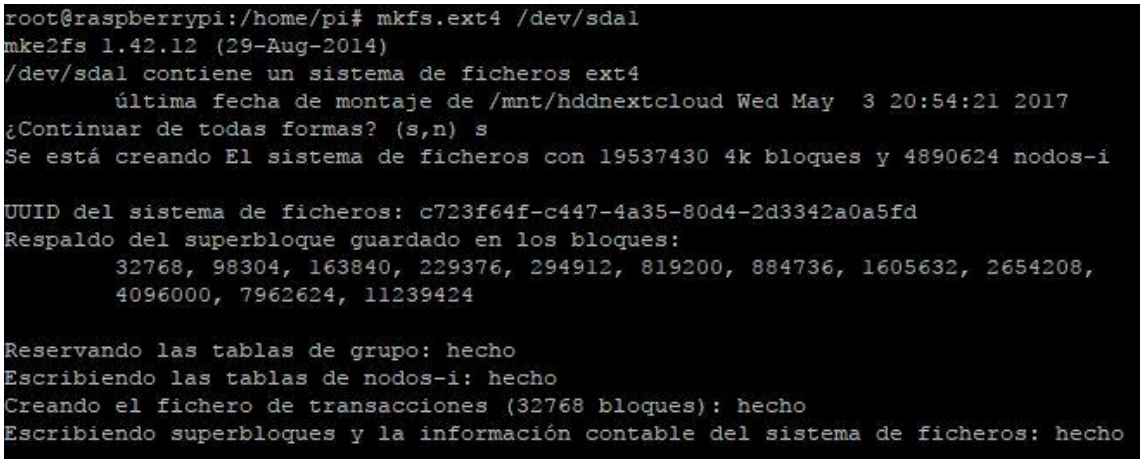
Comprobaremos que se ha creado correctamente la partición en el disco “/dev/sda”, con la nomenclatura “/dev/sda1”.



Por último, necesitamos darle formato a dicha partición por lo que ejecutaremos el siguiente comando:

CÓDIGO:

```
sudo mkfs.ext4 /dev/sda1
```



A continuación, debemos montar permanentemente la partición en una carpeta. Crearemos la carpeta “hddnextcloud” en /mnt/ con el siguiente comando:

CÓDIGO:

```
sudo mkdir /mnt/hddnextcloud
```

Una vez creada la carpeta vamos a montarla permanentemente, para ello editaremos el fichero /etc/fstab con el siguiente comando:

CÓDIGO:

```
sudo nano /etc/fstab
```

Añadiendo la siguiente línea:

CÓDIGO:

```
/dev/sda1      /mnt/hddnextcloud      ext4    defaults 0      0
```

Es necesario reiniciar el sistema para que recoja los cambios y el disco duro quede montado permanentemente en la carpeta /mnt/hddnextcloud

CÓDIGO:

```
sudo shutdown -r now
```

Tras reiniciar el sistema ejecutaremos el siguiente comando para comprobar que la carpeta se ha montado automáticamente:

CÓDIGO:

```
sudo mount | grep /sda
```

Crearemos varios scripts: backupnextcloud.sh y borradobackupnextcloud.sh

Script backupnextcloud.sh

El script backupnextcloud.sh tiene tres funciones principales:

- Crear la carpeta correspondiente a la copia de seguridad en /mnt/hddnextcloud con el nombre “copianextcloud” y la marca de la fecha actual.
- Crear copia de seguridad de la carpeta /var/www/nextcloud en /mnt/hddnextcloud/copianextcloud(fechaactual) con el nombre “nextcloud-dirbkp” y la marca de la fecha actual.
- Crear copia de seguridad de la base de datos MariaDB en /mnt/hddnextcloud/copianextcloud(fechaactual) con el nombre nextcloud-sqlbkp y la marca de la fecha actual.

CÓDIGO:

```
#!/bin/bash

#Creamos la carpeta

mkdir /mnt/hddnextcloud/copianextcloud_`date +"%Y%m%d"` 2>/dev/null

#Copia de la carpeta de Nextcloud

rsync -Aax /var/www/nextcloud/ /mnt/hddnextcloud/copianextcloud_`date +"%Y%m%d"`/nextcloud-dirbkp_`date +"%Y%m%d"`/ 2>/dev/null

#Copia de la base de datos

mysqldump --single-transaction -u root -pp@ssw0rd nextcloud > /mnt/hddnextcloud/copianextcloud_`date +"%Y%m%d"`/nextcloud-sqlbkp_`date +"%Y%m%d"`.bak 2>/dev/null
```

Ubicaremos el script en la carpeta /bin y asignaremos los permisos 755

7 = Lectura+Escritura+Ejecución

5 = Lectura+Ejecución

Para ello ejecutaremos el siguiente comando:

CÓDIGO:

```
chmod 755 backupnextcloud.sh
```

Podemos comprobar el correcto funcionamiento del script, ejecutándolo independientemente de dónde se esté situado en la consola, es decir no es necesario estar ubicado en la carpeta donde se encuentra el script. Esto sucede al almacenar un script en /bin/

Script borradobackupnextcloud.sh

El script borradobackupnextcloud.sh tiene una función principal:

- Busca en el directorio /mnt/hddnextcloud la existencia de alguna carpeta creada hace más de 7 días. Las carpetas que encuentre serán borradas. Es decir, las copias de seguridad nunca sobrepasarán los 7 días de duración en el sistema.

CÓDIGO:

```
sudo find /mnt/hddnextcloud/ -type d -name 'copia*' -mtime +7 -exec rm -rf {} \; 2>/dev/null
```

Ubicaremos el script en la carpeta /bin y asignaremos los permisos 755

7 = Lectura+Escritura+Ejecución

5 = Lectura+Ejecución

Para ello ejecutaremos el siguiente comando:

CÓDIGO:

```
sudo chmod 755 backupnextcloud.sh
```

Al igual que el anterior script, al haberlo ubicado en la carpeta /bin/ se podrá ejecutar desde cualquier lugar de la consola de comandos.

Programación automática ejecución Scripts:

A continuación, vamos a programar la ejecución automática de los scripts anteriormente creados, de tal manera que diariamente se ejecutará el script de borrado de las copias que tienen más de 7 días a las 21:00 pm y diariamente también una copia de seguridad de todo el sistema Nextcloud a las 21:05 pm.

Para programar la ejecución de los scripts ejecutaremos el siguiente comando:

CÓDIGO:

```
sudo crontab -e
```

```
# m h  dom mon dow   command

0 21 * * * /bin/borradobackupnextcloud.sh
5 21 * * * /bin/backupnextcloud.sh
```

Resultado ejecución "crontab -e"

Monitorización de la red:

Introducción:

La monitorización de redes consiste en el uso de un sistema que constantemente monitoriza una red de ordenadores buscando componentes lentos o fallidos, notificando al administrador de la red (vía email, teléfono, u otras alarmas) en caso de cortes.

Normalmente las únicas métricas de medición son tiempo de respuesta, disponibilidad y tiempo de funcionamiento, aunque las métricas de consistencia y fiabilidad están empezando a ganar popularidad.

Icinga:

Icinga es un sistema de código abierto de monitorización de redes y ordenadores. Originalmente fue creado como una bifurcación de Nagios en 2009.

Icinga incorpora nuevas características tales como una moderna interfaz de usuario de estilo Web 2.0, conectores de bases de datos adicionales (MySQL, Oracle, y PostgreSQL) y una API que permite a los administradores integrar numerosas extensiones sin complicadas modificaciones del núcleo de Icinga.

A continuación, se detallan las características principales de Icinga:

Características de Icinga:

Monitorización:

- Monitorización de servicios de red (SMTP, POP3, HTTP, NNTP, ping, etc).
- Monitorización de recursos de host (Carga de CPU, uso de disco, etc).
- Monitorización de componentes de servidor (switches, routers, temperatura y sensores de humedad…).
- Plugin de diseño sencillo que permite a los usuarios desarrollar fácilmente sus propios controles de servicio.
- Controles de servicio paralelos.
- Capacidad para definir jerarquías en la red.
- Capacidad para definir manejadores de eventos.

Notificaciones:

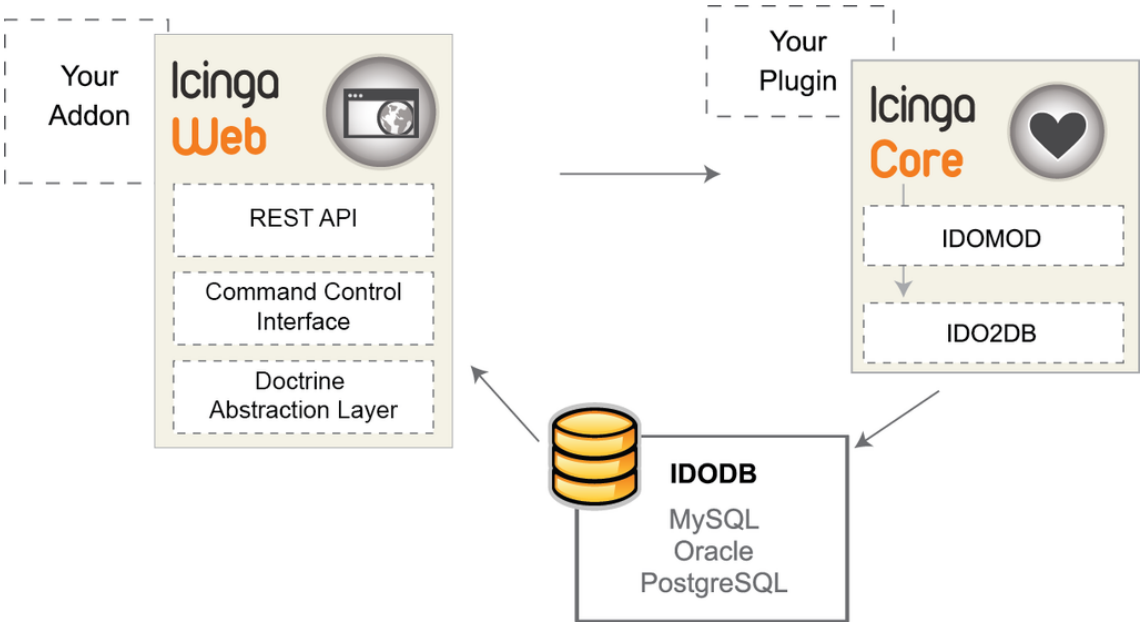
- Notificación a las personas de contacto cuando ocurren problemas de servicio o de host.
- Escalada de alertas a otros usuarios o canales de comunicación.

Visualización e informes:

- Dos interfaces de usuario opcionales (Icinga Classic UI e Icinga Web) para la visualización del estado del host, servicio, mapas de red, informes, registros...
- Informes basados en plantillas, por ejemplo, los 10 principales hosts o servicios problemáticos, sinopsis del entorno completo de supervisión, informes de disponibilidad, etc.
- Repositorio de informes con diferentes niveles de acceso, generación y distribución automática de informes.
- Extensión opcional que distingue entre eventos críticos de tiempos de inactividad planificados y no planificados y periodos de reconocimiento.
- Informes de utilización de la capacidad.
- Rendimiento gráfico a través de complementos como PNP4Nagios, NagiosGrapher e InGraph.

Arquitectura:

El núcleo de Icinga está escrito en C y tiene una arquitectura modular con núcleo autónomo, interfaz de usuario y base de datos en la que los usuarios pueden integrar varios complementos.



Topología Icinga2

Instalación y configuración de Icinga2:

A continuación, vamos a instalar y configurar Icinga2 en nuestra Raspberry Pi. La instalación es relativamente compleja por lo que es necesario seguir correctamente los pasos que a continuación se detallan para obtener una correcta instalación del sistema de monitorización de red.

Para comenzar debemos obtener "software-properties-common", dicho software nos proporciona una abstracción de los repositorios apt usados. Nos permitirá administrar fácilmente la distribución y fuentes independientes del proveedor de Software.

Instalación de software-properties-common:

CÓDIGO:

```
sudo apt install software-properties-common
```

A continuación, vamos a obtener el repositorio necesario para descargar Icinga2, para ello ejecutaremos el siguiente código:

CÓDIGO:

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

Deberíamos obtener un mensaje de confirmación ("OK")

Y lo añadiremos a nuestros repositorios:

CÓDIGO:

```
sudo echo 'deb http://debmon.org/debmon debmon-jessie main' >/etc/apt/sources.list.d/debmon.list
```

Actualizamos nuestros repositorios:

CÓDIGO:

```
sudo apt-get update
```

Y procedemos a realizar la instalación de Icinga2:

CÓDIGO:

```
sudo apt-get install icinga2
```

Habilitamos el servicio que usa el Software de monitorización:

CÓDIGO:

```
sudo systemctl enable icinga2.service
```

Y lo iniciamos mediante el siguiente código:

CÓDIGO:

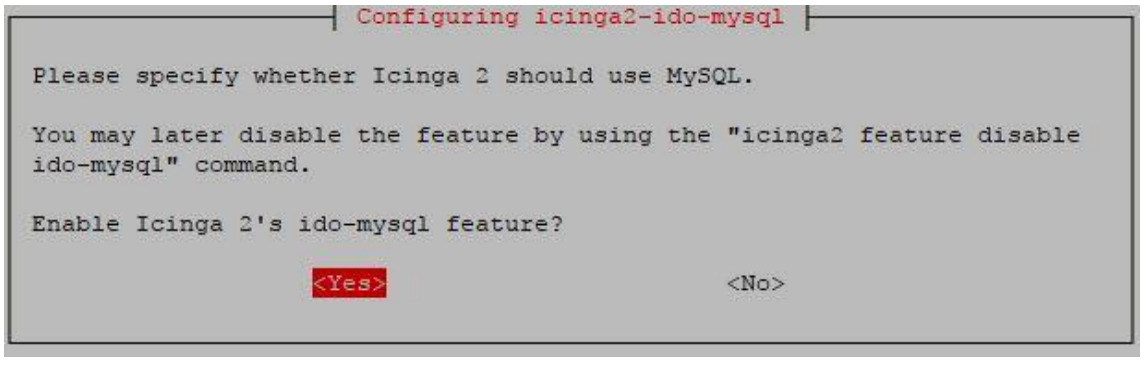
```
sudo systemctl start icinga2.service
```

Instalaremos los plugins necesarios de Icinga2:

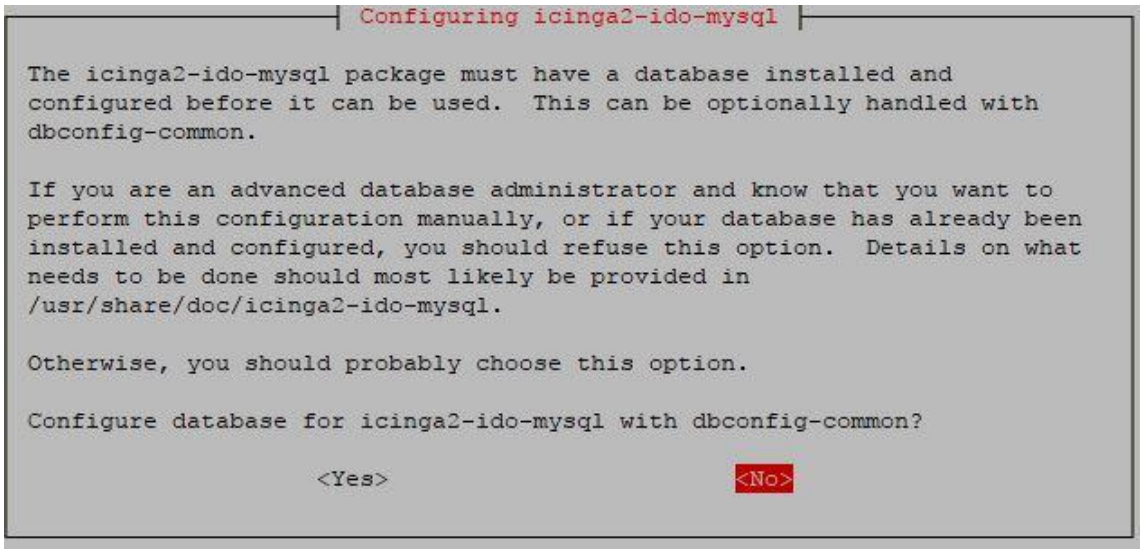
CÓDIGO:

```
sudo apt-get install nagios-plugins icingaweb2 icinga2-ido-mysql
```

A continuación, se mostrará una pantalla que pregunta si deseamos usar la característica de Icinga2 “ido-mysql”, seleccionaremos “yes” ya que será el módulo mediante el cual crearemos la conexión con nuestra base de datos.



Puesto que no nos interesa que la creación de la base de datos se configure automáticamente mediante “dbconfig-common” seleccionaremos “No” en el siguiente menú:



Es necesario disponer de un gestor de base de datos, en este caso se usará MariaDB que ya fue instalado anteriormente al realizar la instalación y configuración de la Nube Privada Nextcloud, por lo que no será necesario volver a instalarlo.

Conectaremos con el sistema gestor de bases de datos (MariaDB) mediante la ejecución del siguiente comando:

CÓDIGO:

```
sudo mysql -u root -p
```

Crearemos la base de datos que usará Icinga2 para almacenar datos:

CÓDIGO:

```
create database icinga2;
```

Crearemos un usuario denominado “icinga2” con contraseña “p@ssw0rd” y le concederemos los permisos de selección, inserción, actualización, borrado, eliminación, creación de vista y ejecución:

CÓDIGO:

```
GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, EXECUTE ON icinga2.*TO 'icinga2'@'localhost' IDENTIFIED BY 'p@ssw0rd';
```

Refrescaremos los privilegios para que comiencen a aplicarse:

CÓDIGO:

```
flush privileges;
```

A continuación, saldremos del sistema gestor de base de datos y editaremos los parámetros de conexión entre Icinga2 y la base de datos Mysql para establecerlos acordes a la base de datos y usuario que acabamos de crear.

Es necesario editar el archivo /etc/icinga2/features-available/ido-mysql.conf mediante el siguiente código:

CÓDIGO:

```
sudo nano /etc/icinga2/features-available/ido-mysql.conf
```

Habilitaremos el módulo que gestiona la conexión con la base de datos anteriormente creada mediante la ejecución del siguiente código:

CÓDIGO:

```
sudo icinga2 feature enable ido-mysql
```

Vamos a volver a conectar con nuestro sistema gestor de base de datos MariaDB para crear esta vez la base de datos que usará la interfaz Web de nuestro sistema de monitorización Icinga2:

Conectamos con el sistema gestor:

CÓDIGO:

```
sudo mysql -u root -p
```

Crearemos la base de datos:

CÓDIGO:

```
create database icingaweb;
```

Crearemos el usuario “icingaweb” concediéndole el permiso SUPER que nos permitirá conectarnos, incluso si se ha superado el número máximo de conexiones y realizar otras operaciones como la configuración de variables globales del servidor y gestión de procesos.

CÓDIGO:

```
GRANT SUPER ON *.* to 'icingaweb'@'localhost' IDENTIFIED BY 'p@ssw0rd';
```

Concederemos los permisos de selección, actualización, borrado, eliminación, creación de vista, indexación y ejecución a “icingaweb”.

CÓDIGO:

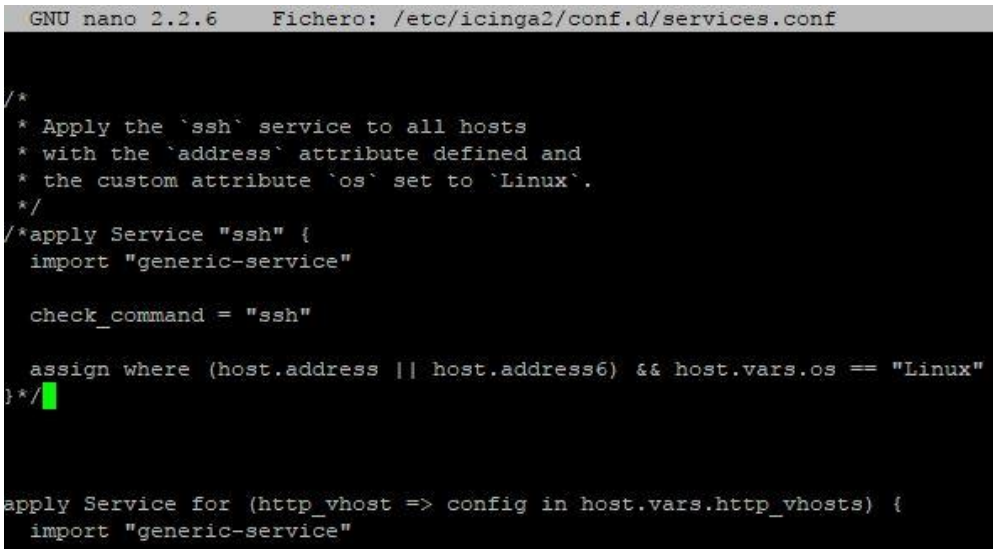
```
GRANT SELECT, INSERT, UPDATE, DELETE, DROP, CREATE VIEW, INDEX, EXECUTE ON icingawebdb.*TO 'icingaweb'@'localhost' IDENTIFIED BY 'p@ssw0rd';
```

A continuación, saldremos del sistema gestor de bases de datos.

Es necesario editar el archivo /etc/icinga2/conf.d/services.conf y comentar el párrafo referente a “ssh” ya que si no el servicio relativo a Icinga2 no iniciará correctamente y saldrá un error.

CÓDIGO:

```
sudo nano /etc/icinga2/conf.d/services.conf
```



A continuación, habilitaremos el servicio icinga2 y comprobaremos su estado mediante la ejecución del siguiente comando:

CÓDIGO:

```
sudo systemctl enable icinga2.service && sudo systemctl status icinga2.service
```

Por último, reiniciaremos el servicio apache2, el servicio MySQL y el servicio Icinga2:

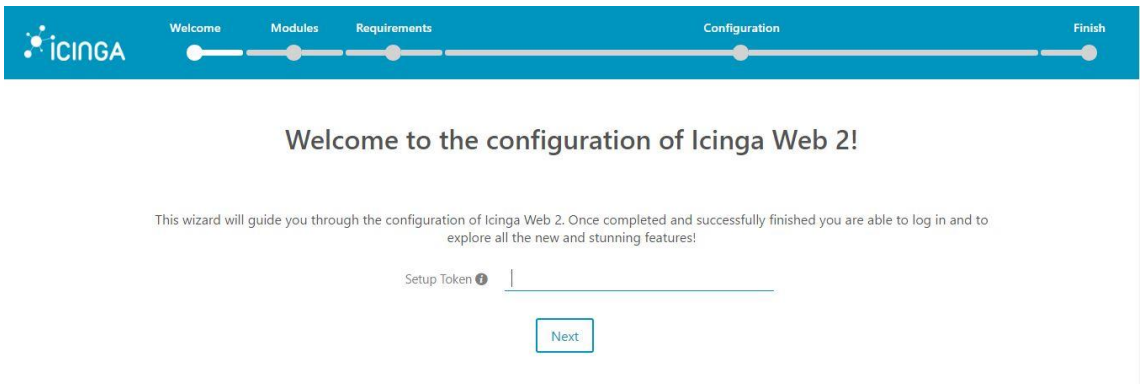
CÓDIGO:

```
sudo systemctl restart apache2.service mysql.service icinga2.service
```

A continuación, para comenzar con la instalación y configuración de icingaweb2 entraremos desde el navegador en la siguiente URL:

CÓDIGO:

```
http://192.168.1.100/icingaweb2/setup
```



Inicio configuración Icingaweb2

Veremos que nos solicita un “token”, que no disponemos, por lo que vamos a generarlo.

Ejecutaremos el siguiente comando relativo a la configuración de directorio y de grupo de icingaweb2:

CÓDIGO:

```
sudo icingacli setup config directory --group icingaweb2;
```

Para posteriormente ejecutar el comando que creará el token necesario para introducir en la Web, es necesario copiar el código que nos generará el siguiente comando:

CÓDIGO:

```
sudo icingacli setup token create;
```

```
root@raspberrypi:/home/pi# icingacli setup token create;
The newly generated setup token is: d20a82dec99d2818
root@raspberrypi:/home/pi#
```

Propagaremos la base de datos con el siguiente comando:

CÓDIGO:

```
mysql -u root -p icinga2 < /usr/share/icinga2-ido-mysql/schema/mysql.sql
```

A continuación, reiniciaremos los servicios apache2, MySQL e Icinga2 mediante el siguiente comando:

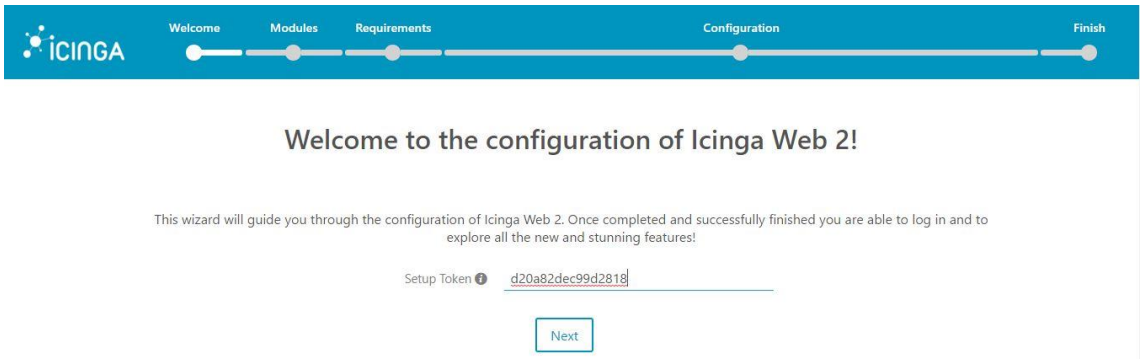
CÓDIGO:

```
systemctl restart apache2.service mysql.service icinga2.service
```

Accederemos a la siguiente dirección de nuevo:

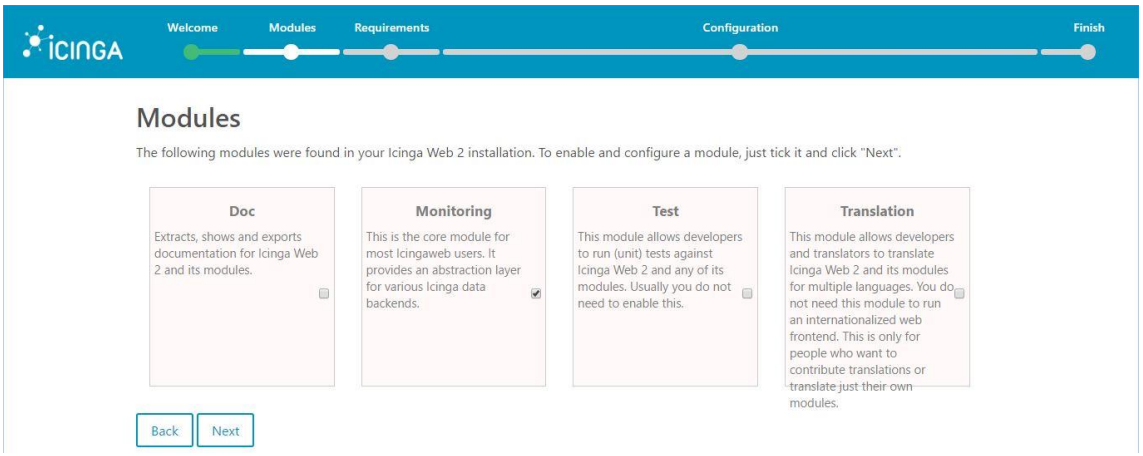
CÓDIGO:

```
http://192.168.1.100/icingaweb2/setup
```

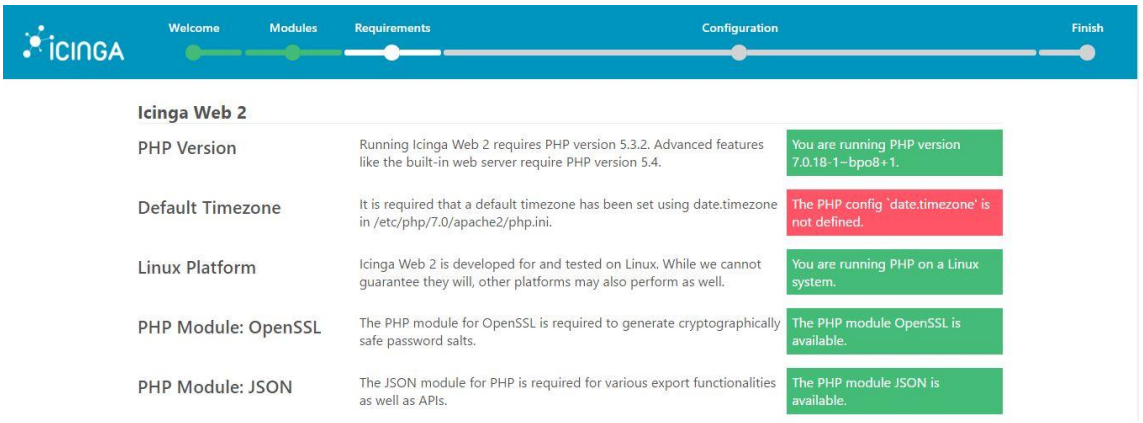


Y esta vez copiaremos el “token” anteriormente generado y pulsaremos sobre “next”.

Seleccionaremos el módulo que vamos a usar, en nuestro caso el de monitorización, pulsaremos “next” para continuar el proceso.



Como podemos comprobar, al intentar continuar sale un mensaje de error en rojo:



Dicho error nos especifica que es necesario configurar el parámetro date.timezone del archivo de PHP /etc/php/7.0/apache2/php.ini estableciéndolo así en: “Europe/Madrid”.

Por lo que vamos a editar el archivo /etc/php/7.0/apache2/php.ini

CÓDIGO:

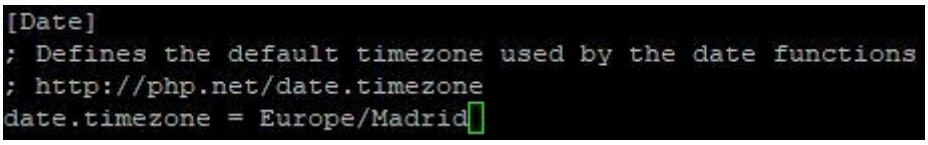
```
sudo nano /etc/php/7.0/apache2/php.ini
```

El valor de date.timezone lo obtendremos de la siguiente URL oficial de PHP:

CÓDIGO:

```
http://php.net/manual/es/timezones.europe.php
```

En nuestro caso lo descomentaremos y estableceremos en Europe/Madrid

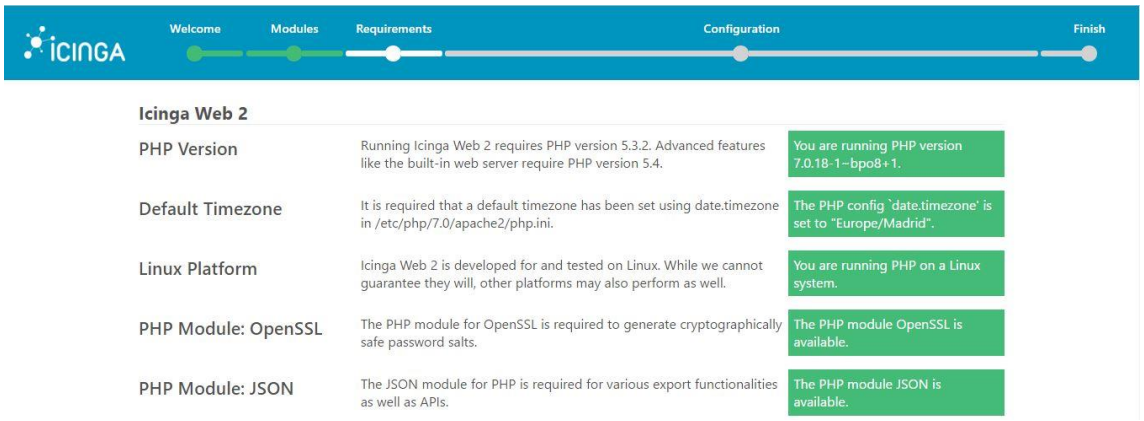


Reiniciaremos el servicio apache2, MySQL e Icinga2 para que se recojan correctamente los cambios:

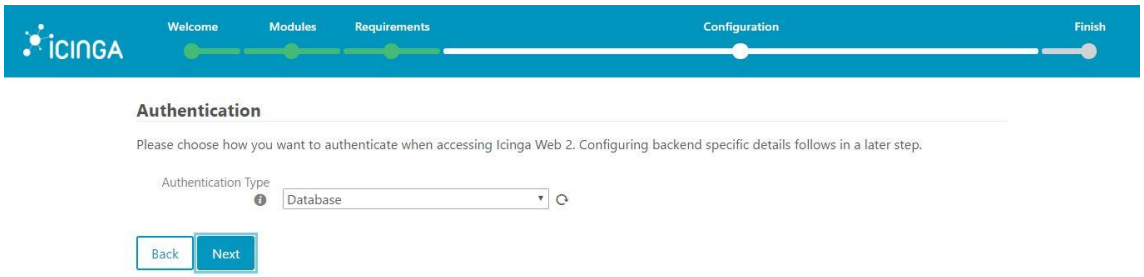
CÓDIGO:

```
sudo systemctl restart apache2.service mysql.service icinga2.service
```

Actualizaremos la página Web en la que aparecía el error date.timezone y veremos que dicho error se ha corregido:

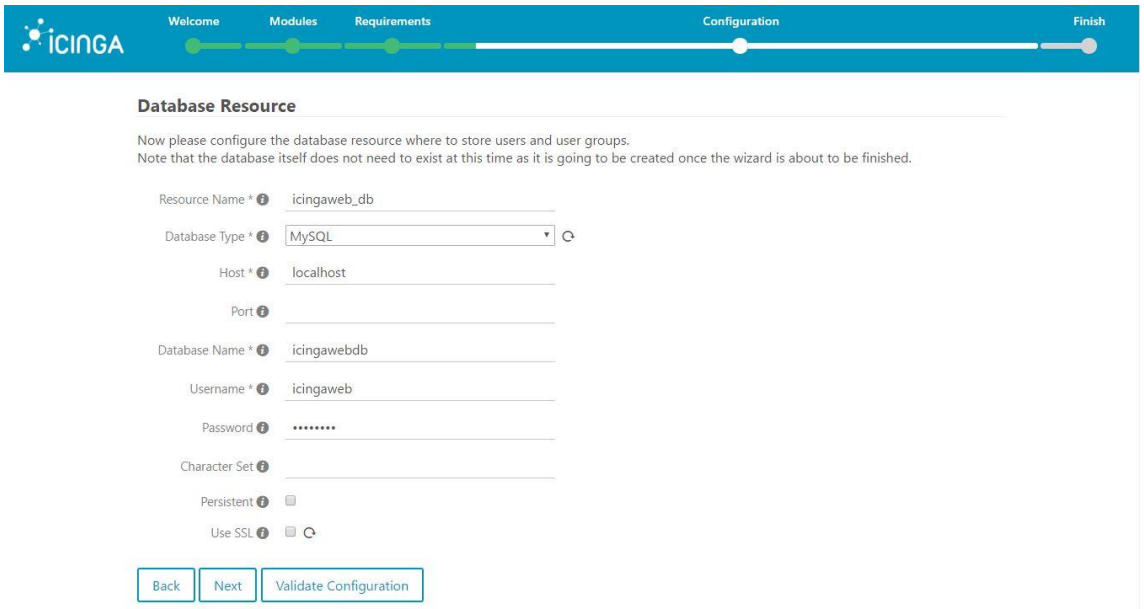


Ya que nuestro método de autenticación será contra una base de datos, estableceremos el valor en "Database" y pulsaremos sobre "Next".



A continuación, hemos de configurar el nombre de nuestra base de datos, el usuario y la contraseña estableciendo así los siguientes valores:

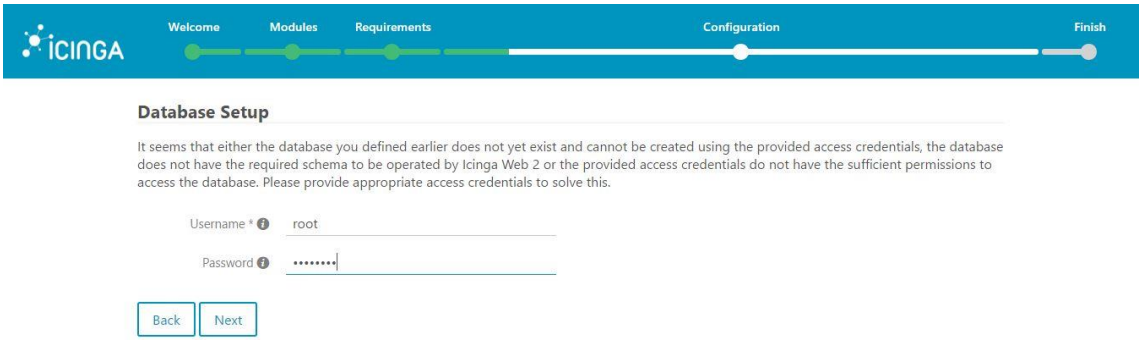
Nombre de BBDD:	icingawebdb
Username:	icingaweb
Password:	p@ssw0rd



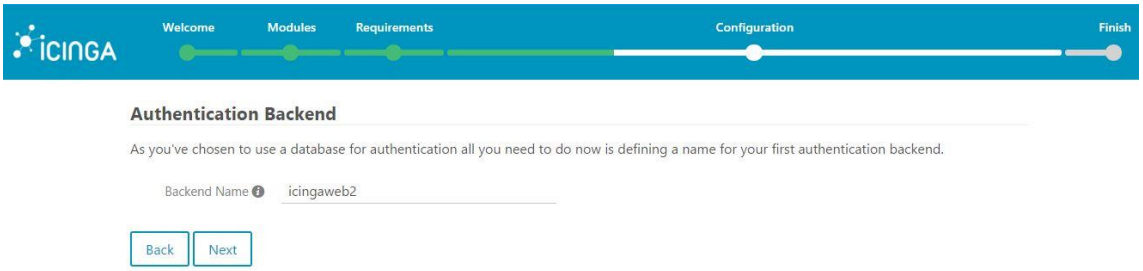
Pulsaremos de nuevo en “next”.

Estableceremos los parámetros de conexión root con la base de datos:

Usuario:	root
Password:	p@ssw0rd

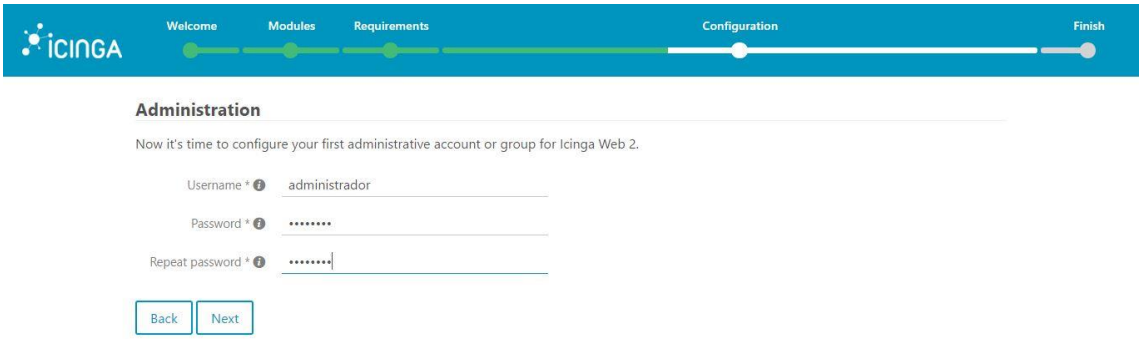


Como hemos elegido usar una base de datos para realizar la autenticación, es necesario definir un nombre para la primera autenticación en el lado del servidor. Estableceremos dicho parámetro en “icingaweb2”.



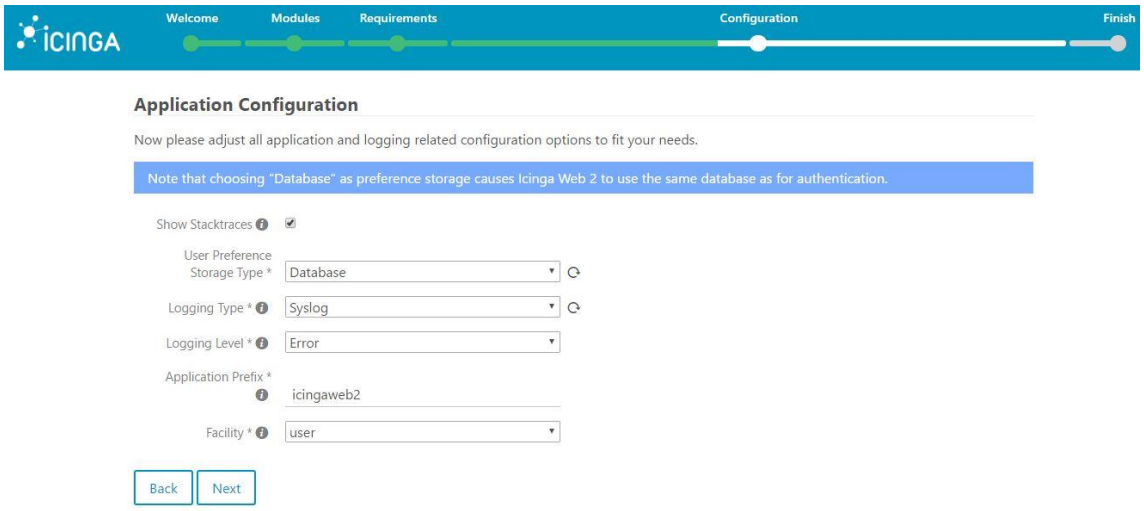
Crearemos el siguiente usuario para loguearnos en el sistema de monitorización Icinga2 a través de la interfaz Web:

Usuario:	administrador
Contraseña:	p@ssw0rd



A continuación, pulsaremos de nuevo sobre el botón “next”.

Comprobaremos y ajustaremos los datos relativos a las opciones de aplicación y log del sistema, estableciendo así los valores tal y como se muestran a continuación:

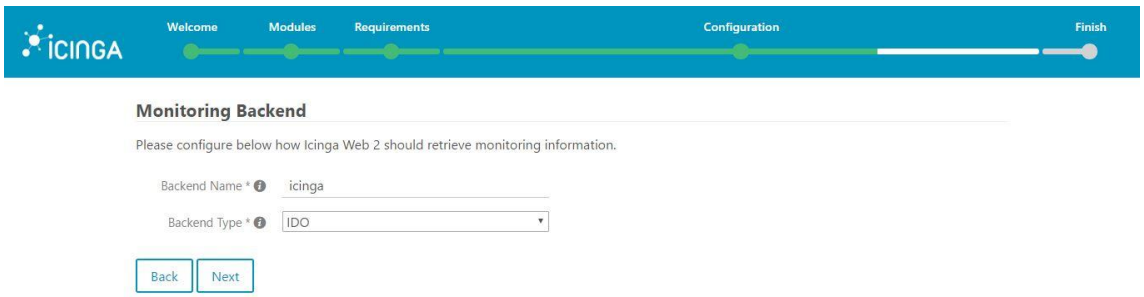


Pulsaremos de nuevo sobre “next”.

Deberíamos visualizar un resumen de la configuración que hemos realizado hasta el momento, si es correcta pulsaremos sobre “next”.

Comenzaremos con la configuración del módulo de monitorización de Icinga Web 2.

Configuraremos el método por el cual Icinga Web 2 recibirá la información de monitorización. Estableciendo Backend Name en “icinga” y Backend Type en “IDO”:

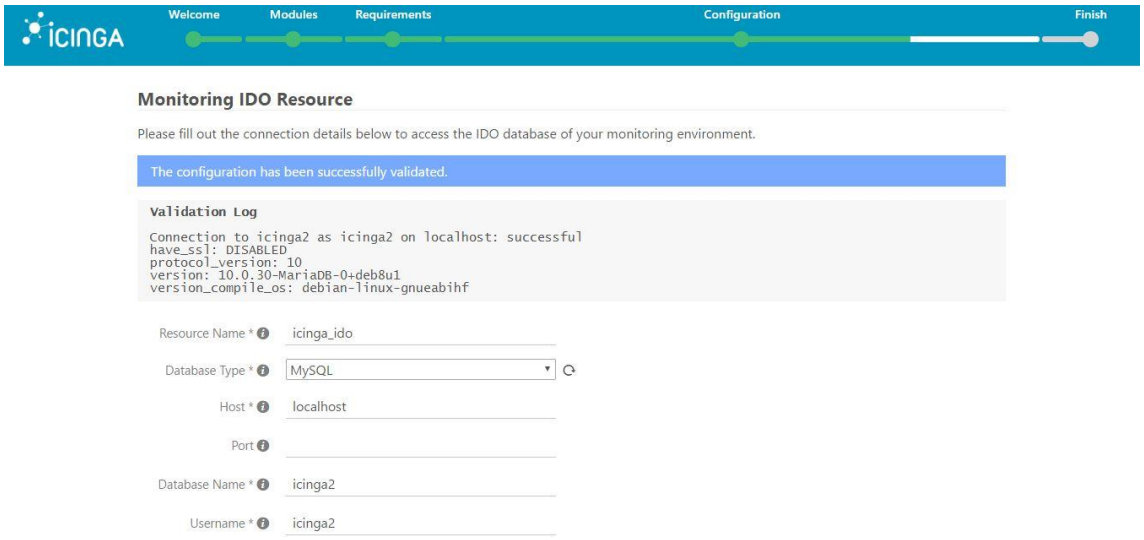


Pulsaremos sobre el botón next.

Ahora vamos a comprobar los valores de la base de datos de icinga2, que deberían ser los siguientes:

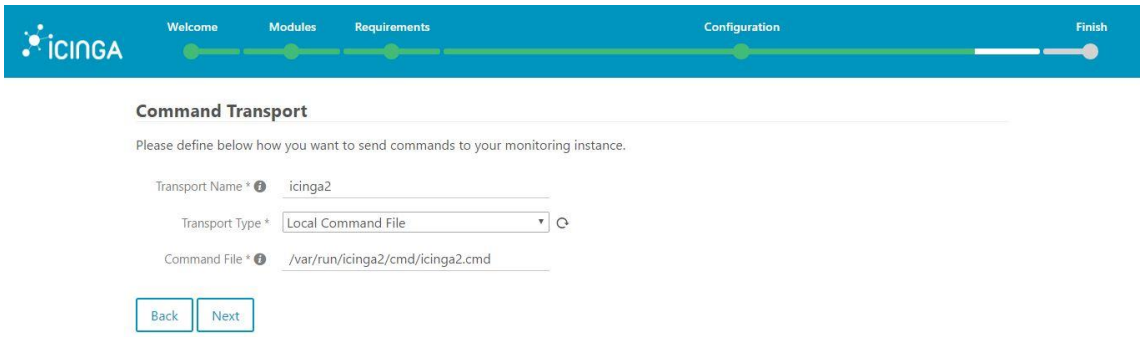
Nombre de BBDD:	icinga2
Usuario:	icinga2
Password:	p@ssw0rd

Pulsaremos sobre Validate configuration para comprobar la conexión con la base de datos y pulsaremos sobre el botón next.



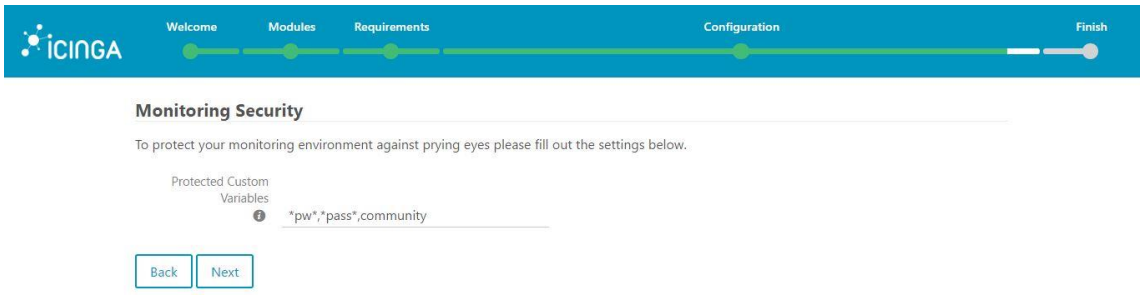
De nuevo pulsaremos sobre el botón next.

Definiremos cómo queremos que se manden los comandos a nuestra instancia de monitorización, estableciendo así los valores que se muestran a continuación:

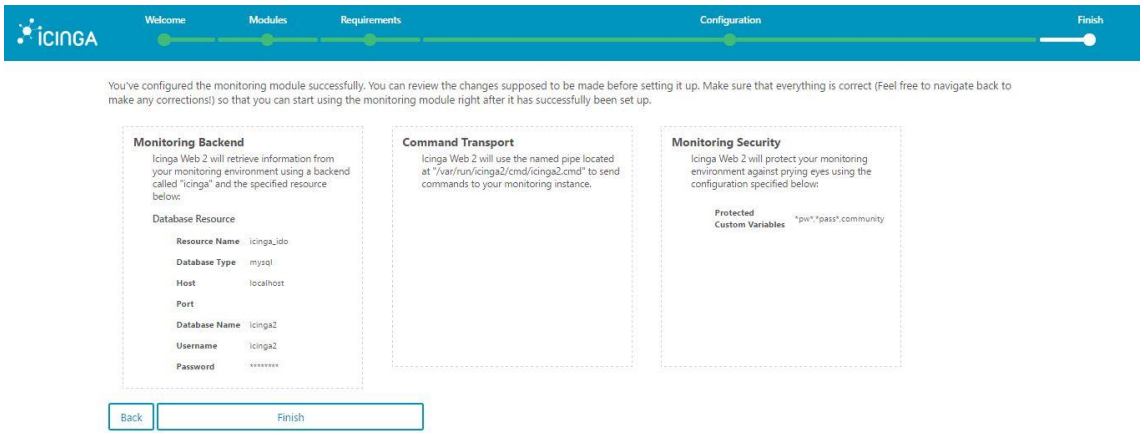


Pulsaremos el botón next.

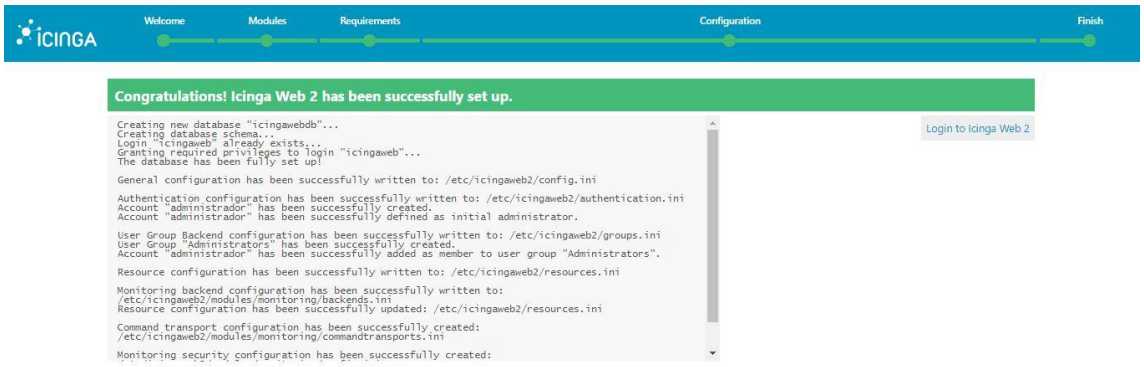
Para proteger el entorno estableceremos las variables “pw”, “pass” y “community”:



Por último, se mostrará un resumen de la configuración final del módulo de monitorización. Para finalizar la configuración del sistema pulsaremos sobre el botón “Finish”.



Icinga Web 2 mostrará un mensaje de confirmación de la correcta instalación del sistema de monitorización:



Haremos clic en “Login to Icinga Web 2” para iniciar sesión en el sistema.

A continuación, iniciaremos sesión con usuario “administrador” y contraseña “p@ssw0rd”:

Panel inicio sesión Icingaweb2

Ya habremos iniciado sesión correctamente en el sistema.

Por último, vamos a añadir un host a monitorizar. Editaremos el archivo /etc/icinga2/conf.d/hosts.conf con el siguiente comando:

CÓDIGO:

```
sudo nano /etc/icinga2/conf.d/hosts.conf
```

Y añadiremos lo siguiente al final del archivo:

CÓDIGO:

```
object Host "vm-centreon2"{

import "generic-host"

address="192.168.1.7"

display_name="Windows 10"

vars.os="Windows"

}
```

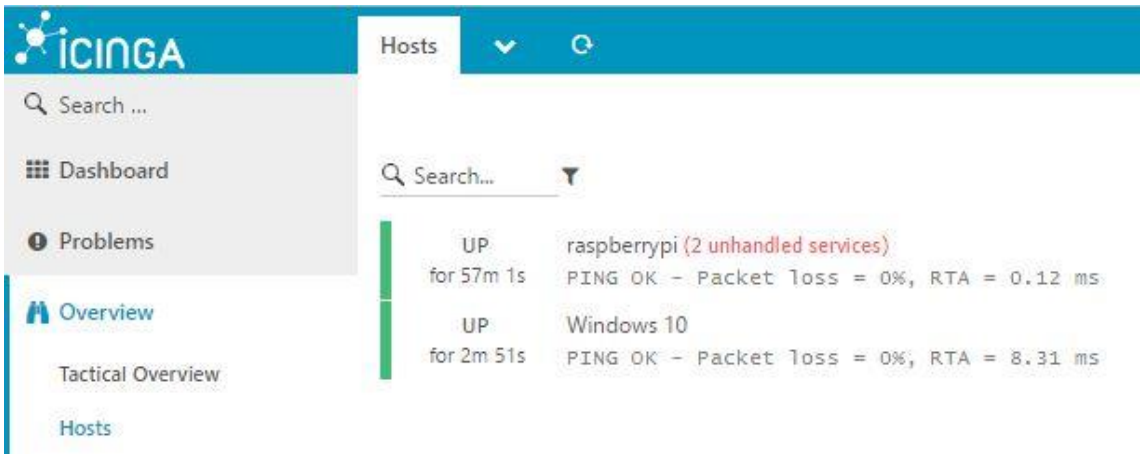
Archivo hosts.conf

Por último, para que se realicen los cambios, debemos recargar la configuración ejecutando el siguiente comando:

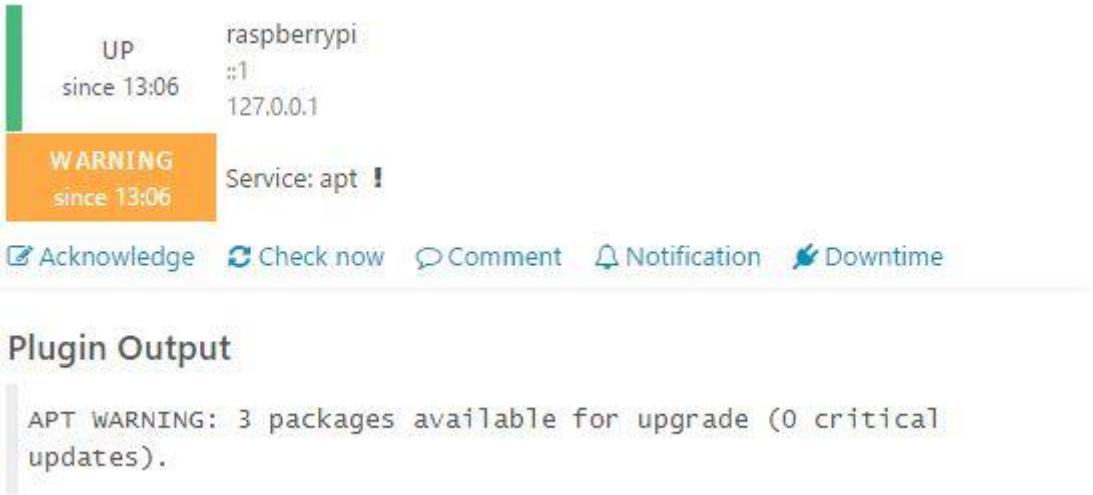
CÓDIGO:

```
sudo service icinga2 reload
```

Si volvemos a la interfaz Web podemos comprobar desde el apartado de “Hosts” como detecta nuestro equipo Windows 10.

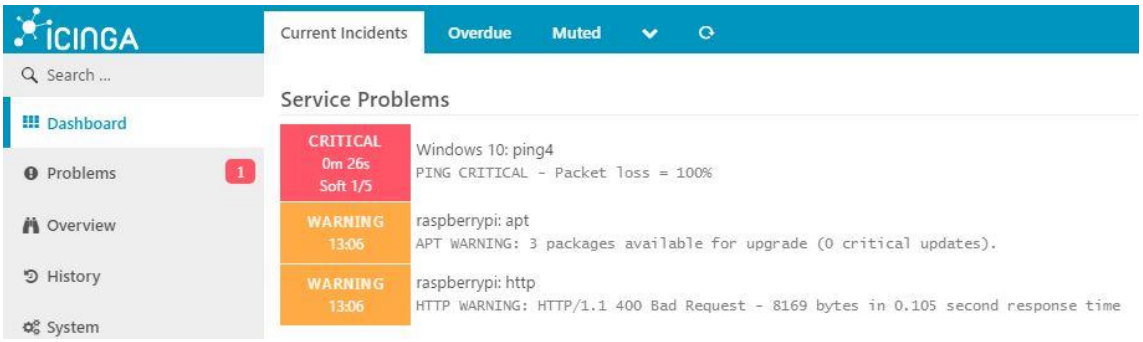


Como podemos comprobar en la anterior imagen, el sistema de monitorización Icinga2 ha detectado “dos servicios no gestionados”. Si hacemos clic en la advertencia, podemos comprobar cómo uno de los avisos, es la disponibilidad de tres paquetes a actualizar en nuestro servidor RaspberryPi. Creo que este ejemplo muestra a la perfección el enorme potencial de este software de monitorización.

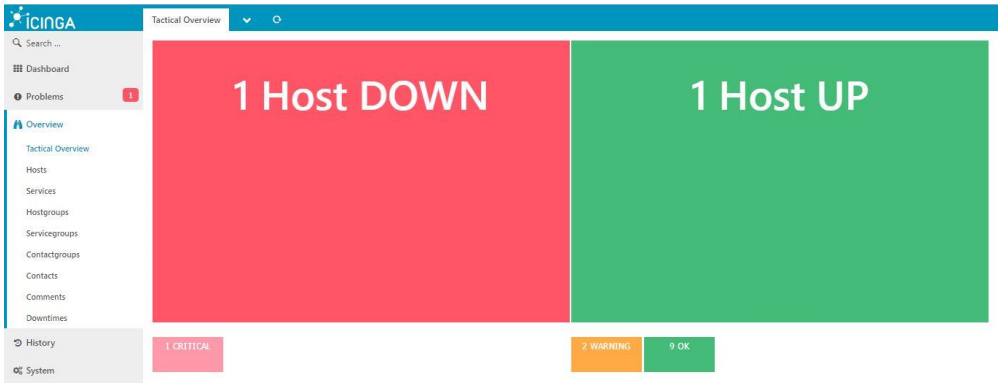


Supongamos que el equipo Windows 10, que anteriormente añadimos, es un equipo crítico en nuestra infraestructura, por lo que vamos a proceder a apagarlo para ver cómo reacciona nuestro software de monitorización.

Nada más proceder con el apagado del equipo, Icinga2 muestra un problema crítico, notificando la falta de comunicación con el equipo.



Tras intentar durante 2 minutos la comunicación con dicho equipo, procede a establecerlo cómo "caído" en el apartado "tactical overview".



Configuración notificaciones a través de correo electrónico:

Vamos a configurar nuestro sistema de monitorización Icinga2, para que nos notifique automáticamente mediante el envío de un correo electrónico, cuando detecte que un equipo, previamente seleccionado de nuestra red, se apague o pierda la conexión.

Vamos a comenzar listando las características disponibles de Icinga2, para ello ejecutaremos el siguiente comando:

CÓDIGO:

```
sudo icinga2 feature list
```

Habilitaremos la característica de notificación del sistema de monitorización:

CÓDIGO:

```
sudo icinga2 feature enable notification command
```

Reiniciaremos el servicio Icinga2 para que comience a usar la característica de notificación:

CÓDIGO:

```
sudo service icinga2 restart
```

A continuación, editaremos el archivo /etc/icinga2/conf.d/users.conf , en dicho archivo se encuentran los diferentes usuarios y grupos que forman parte de nuestro sistema de monitorización. Modificaremos el archivo ejecutando el siguiente comando:

CÓDIGO:

```
sudo nano /etc/icinga2/conf.d/users.conf
```

Editaremos el Email del usuario que hay creado (administrador), estableciendo el email al que queremos que lleguen las notificaciones:

CÓDIGO:

```
daviddelriopascual@gmail.com
```

A continuación, instalaremos los paquetes “ssmtp” y “mailutils”, el software necesario para permitirnos enviar correos electrónicos desde nuestro servidor:

CÓDIGO:

```
sudo apt-get install ssmtp mailutils
```

Vamos a editar el archivo /etc/ssmtp/ssmtp.conf , en él estableceremos la configuración de nuestro proveedor de correo electrónico, en mi caso Gmail:

CÓDIGO:

```
sudo nano /etc/ssmtp/ssmtp.conf
```

Editaremos la línea “mailhub” estableciéndola tal y cómo se muestra a continuación:

CÓDIGO:

```
mailhub=smtp.gmail.com:587
```

Y añadiremos el siguiente código al final del archivo:

CÓDIGO:

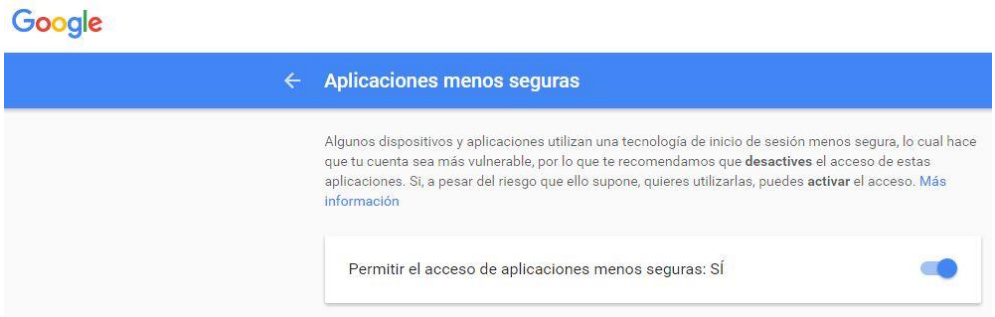
```
# Usuario y contraseña de la cuenta Gmail. Usuario con el prefijo "@".

# Ej. usuario@ejemplo.com

AuthUser=mensajesraspberry@gmail.com

AuthPass=p@ssw0rd2017



# Esto es para usar conexiones STARTTLS

UseSTARTTLS=YES
```

Para que funcione correctamente el envío de mensajes a través de un servidor Gmail, necesitamos habilitar la opción de Google “Permitir que aplicaciones menos seguras accedan a las cuentas”. Para ello accederemos al siguiente link:

CÓDIGO:

```
https://myaccount.google.com/lesssecureapps?pli=1
```

Y estableceremos en “Sí” la opción “Permitir el acceso de aplicaciones menos seguras”.



Panel “Aplicaciones menos seguras”

A continuación, ejecutaremos el siguiente comando para comprobar si funciona correctamente el envío de un correo electrónico desde nuestro sistema.

CÓDIGO:

```
sudo echo "Cuerpo del correo" | mail -s "Asunto del Correo" daviddelriopascual@gmail.com
```

Vamos a añadir un nuevo host a nuestro sistema de monitorización, para ello editaremos el archivo /etc/icinga2/conf.d/hosts.conf

CÓDIGO:

```
sudo nano /etc/icinga2/conf.d/hosts.conf
```

Y añadiremos al final del archivo el siguiente código:

CÓDIGO:

```
      object Host "vm-centreon5"{
       import "generic-host"
       address="192.168.1.150"
       display_name="Windows10"
       vars.os="Windows"
       vars.notification_type = "mail"
       vars.notification["mail"] = {
       /* The UserGroup `icingaadmins` is defined in `users.conf`. */
       groups = [ "icingaadmins" ]
        }
}
```

Por último, reiniciaremos Icinga2 para que recoja los cambios y configuraciones que hemos realizado en el sistema, para ello ejecutaremos los siguientes comandos:

CÓDIGO:

```
sudo service icinga2 reload

sudo service icinga2 restart
```

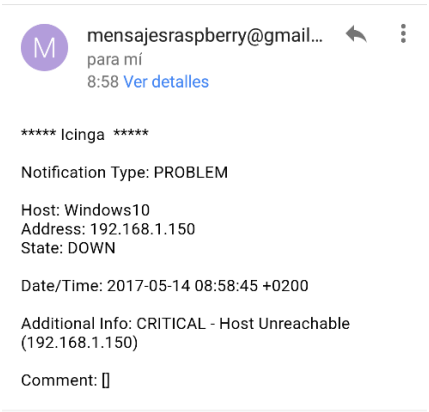
Configuraremos el archivo /etc/icinga2/conf.d/users.conf para establecer múltiples usuarios pertenecientes al grupo de los administradores. A continuación, se muestra un ejemplo:

Ejemplo de archivo users con dos usuarios:

```
/**
 * The example user 'icingaadmin' and the example
 * group 'icingaadmins'.
 */

object User "daviddrp" {
  import "generic-user"

  display_name = "David Del Río Pascual"
  groups = [ "icingaadmins" ]

  email = "daviddelriopascual@gmail.com"
}

object UserGroup "icingaadmins" {
  display_name = "Icinga 2 Admin Group"
}


object User "josejlt" {
  import "generic-user"

  display_name = "Jose Juan [illegible]"
  groups = [ "icingaadmins" ]

  email = "[illegible]"
}

object User "eduardotb" {
  import "generic-user"

  display_name = "Eduardo [illegible]"
  groups = [ "icingaadmins" ]

  email = "[illegible]"
}
```

Archivo users.conf

En la siguiente imagen se puede observar cómo se recibe correctamente el mensaje de notificación en un dispositivo Android.



Telefonía IP (VOIP):

Introducción ¿Qué es la telefonía IP?

Atendiendo a su definición, telefonía IP o VOIP (Voice Over Internet Protocol), es un conjunto de recursos que hacen posible que la señal de voz viaje a través de Internet empleando el protocolo IP. La señal viaja en forma digital, en paquetes de datos en lugar de enviarla en forma analógica a través de circuitos utilizables solo por telefonía convencional.

El tráfico de voz sobre IP puede circular por cualquier red IP, incluyendo aquellas conectadas a Internet, como por ejemplo las LAN, por lo que es una opción válida en cualquier empresa.

Ventajas de la telefonía IP:

- Integración con el sistema informático.
- Ahorro de costes.
- Eficiencia en la localización de personas.
- Integración de SmartPhones.
- Libertad en la elección del operador.
- Libertad en la elección de equipos.
- Aprovechamiento del equipamiento existente.
- Facilidad de instalación.
- Posibilidad de grabación de llamadas.
- Operadora virtual.
- Fácil multiconferencia.
- Normas de uso, fechas y horas.
- Mejora en la atención al cliente.
- Monitorización.
- Recepción de faxes por email.
- Recepción de mensajes de voz por email.
- Escalable.
- Actualizable.
- Tranquilidad frente a averías.
- Soporte remoto.

Asterisk:

Asterisk en un framework para crear aplicaciones de comunicación. Es capaz de convertir cualquier sistema en un servidor de comunicaciones. Lo usan desde pequeños negocios, hasta grandes empresas, call centers, aerolíneas… Es gratis y de código libre.

Lanzado hace más de 10 años y constantemente desarrollado por la comunidad open source, Asterisk se ha convertido en uno de los servidores de comunicaciones más ricos en funcionalidades, escalables y sofisticados de los disponibles en la actualidad.

Ventajas de Asterisk:

- Funcionalidad.
- Escalabilidad.
- Competitividad en coste.
- Interoperabilidad y flexibilidad.

Instalación y configuración de Asterisk:

Vamos a comenzar con la instalación y configuración de Asterisk en nuestro pequeño servidor. Lo primero que debemos hacer es instalar el paquete Asterisk, por lo que ejecutaremos el siguiente comando:

CÓDIGO:

```
sudo apt-get install Asterisk
```

Tras finalizar el proceso de instalación es necesario realizar diversas configuraciones. Procederemos a editar el archivo /etc/asterisk/sip.conf

CÓDIGO:

```
sudo nano /etc/asterisk/sip.conf
```

Modificaremos dicho archivo. Para comenzar, estableceremos los parámetros generales de nuestro servidor Asterisk:

CÓDIGO:

```
[general]

port= 5060

bindaddr= 0.0.0.0

allow= all codecs

context= default
```

Tal y cómo podemos ver en la anterior imagen el puerto que usará nuestro servidor Asterisk será el 5060.

Vamos a configurar los clientes que usarán nuestro sistema de telefonía, en este caso añadiremos 4 usuarios con la siguiente estructura:

```
[1001]
type= friend
host= dynamic
username= David
secret= p@ssw0rd
callerid="David" <1001>

[1002]
type= friend
host= dynamic
username= 1002
secret= p@ssw0rd
callerid="José Juan" <1002>

[1003]
type= friend
host= dynamic
username= 1003
secret= p@ssw0rd
callerid="Eduardo" <1003>

[1004]
type= friend
host= dynamic
username= 1004
secret= p@ssw0rd
callerid="Roberto" <1004>
```

```
[1005]

type= friend

host= dynamic

username= 1005

secret= p@ssw0rd

callerid="Pepe" <1005>
```

(El usuario Pepe se ha creado para la realización de pruebas).

Por último, vamos a configurar el archivo /etc/asterisk/extensions.conf , probablemente el archivo más importante de Asterisk. Tiene como misión establecer el plan de llamadas de cada extensión.

```
[default]

exten => 1001,1,Dial(SIP/1001,30)

exten => 1002,1,Dial(SIP/1002,30)

exten => 1003,1,Dial(SIP/1003,30)

exten => 1004,1,Dial(SIP/1004,30)

exten => 1005,1,Dial(SIP/1005,30)
```

Ejecutaremos el siguiente comando para reiniciar Asterisk y que recoja los cambios realizados:

CÓDIGO:

```
sudo service asterisk restart
```

Comprobación VOIP

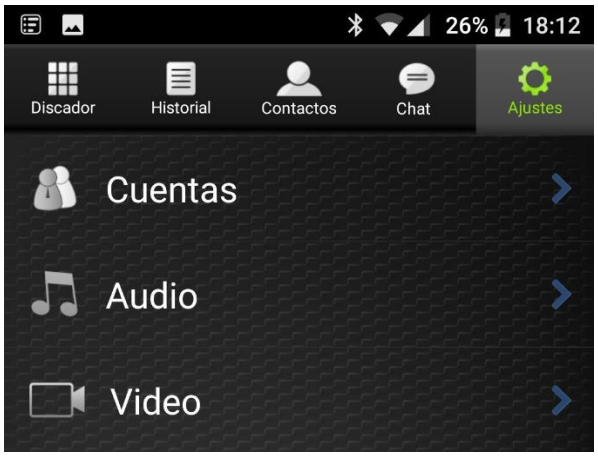
Ya tenemos nuestro sistema VOIP en marcha, ahora tan sólo hace falta comprobar su correcto funcionamiento, serán necesarios dos dispositivos Android con los que realizaremos las pruebas.

A continuación, se mostrará el proceso de instalación de Zoiper IAX SIP VOIP Softphone en un dispositivo, pero deberemos hacerlo en los dos, configurando diferentes cuentas en cada uno de ellos.

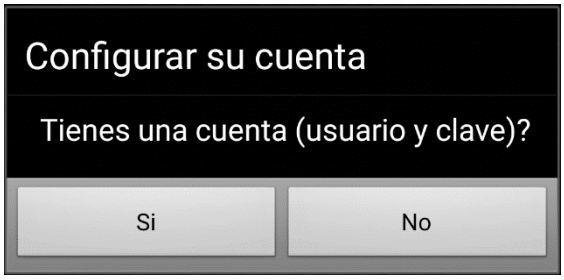
Descargaremos la aplicación desde el mercado de aplicaciones de Android:

App Zoiper IAX SIP VOIP Softphone en Play Store (Android)

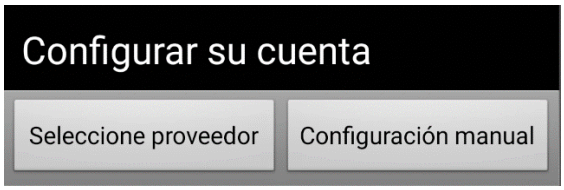
Una vez dentro de la aplicación iremos al menú de Ajustes>Cuentas.



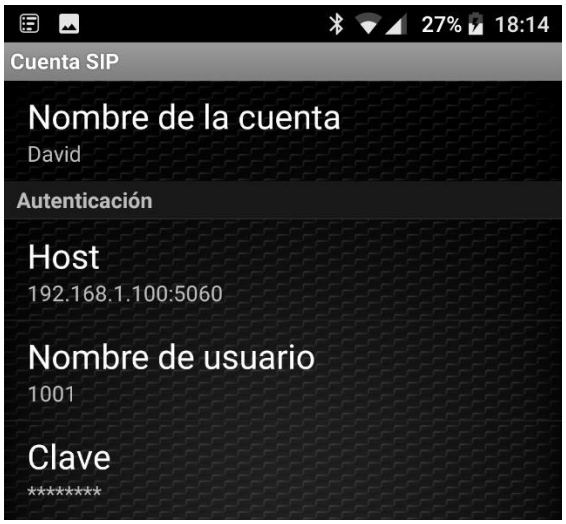
Crearemos una cuenta. El software preguntará si disponemos un nombre de usuario y clave, pulsaremos sobre “Si”.



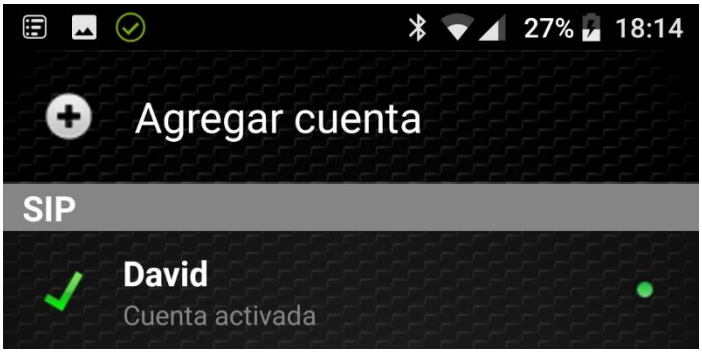
Seleccionaremos configuración manual para, posteriormente, establecer los parámetros necesarios.



Elegiremos SIP como tipo de cuenta y estableceremos los datos de uno de nuestros usuarios. En el ejemplo se ha usado la cuenta “David”. Tal y cómo se puede ver en la imagen, hemos de establecer en el apartado “Host” la dirección IP de nuestro servidor con la siguiente estructura: “IP:PUERTO”. Por último, estableceremos el nombre de usuario y clave.



Tras guardar los cambios, veremos que se ha activado la cuenta en nuestro dispositivo.



En el otro dispositivo configuraremos la aplicación con el usuario “Pepe”. Desde dicho móvil, llamaremos a la cuenta anteriormente configurada (David). En la siguiente imagen se muestra cómo desde el dispositivo de David (extensión 1001) se recibe la llamada del usuario de prueba “Pepe” (extensión 1005) por lo que nuestro sistema funciona perfectamente.

Ejemplo llamada entrante a través de Asterisk

Con esto finalizará la configuración de nuestro sistema de voz sobre IP en nuestra Raspberry Pi.

Configuración Asterisk cuentas IAX (posibilidad de realización de llamadas fuera de la red local)

1. Abriremos el puerto 4569 en el Router
2. Debemos editar el archivo /etc/asterisk/iax.conf añadiendo la configuración por defecto:

```
[general]
bindport=4569
bindaddr=192.168.1.100
delayreject=yes
srvlookup=yes
accountcode=lss0101
language=en
disallow=all
allow=ulaw
allow=alaw
allow=gsm
```

3. Añadir usuarios debajo de la anterior configuración en el archivo /etc/asterisk/iax.conf

```
[1010]
type=friend ; tipo friend es peer y user a la vez
host=dynamic ; si el cliente no se conecta siempre desde un IP determinado hay $
secret=1234 ;contraseña
context=default ; contexto asociado a este usuario en extensions.conf
mailbox=1234@default ; casilla de los correo de voz del usuario
qualify=yes; para averiguar periódicamente con un ping si el usuario está conectado
callerid = "David Del Río" ; identificativo de llamada del usuario
```

4. Añadir extensiones en archivo /etc/asterisk/extensions.conf

```
exten => 1010,1,Dial(IAX2/1010,30)
```

Configuraciones adicionales:

Realizaremos una serie de configuraciones adicionales que aportarán nuevas funciones y un plus de seguridad.

Fail2ban:

Introducción:

Fail2ban es una aplicación escrita en Python usada para la prevención de intrusos en un sistema, que actúa penalizando o bloqueando las conexiones remotas que intentan accesos por fuerza bruta.

Fail2ban busca en los registros (logs) de los programas que se especifiquen las reglas que el usuario decida para poder aplicar una penalización. La penalización puede ser bloquear la aplicación que ha fallado en un determinado puerto, bloquearla para todos los puertos, etc. Las penalizaciones, así como las reglas, son definidas por el usuario.

Habitualmente, si las IP de ataque se prohíben por un lapso prudencial de tiempo, la sobrecarga de red provocada por los ataques baja y, también se reduce la probabilidad de que un ataque de fuerza bruta basada en diccionarios tenga éxito.

Después de una sucesión de intentos fallidos, Fail2ban (en función de la configuración determinada por el usuario) decidirá la acción a realizar sobre la IP que originó el problema. Puede simplemente notificar por e-mail del suceso, denegar el acceso a la IP atacante, bloquearla en determinados puertos y habilitarla en otros, etc.

Servicios que soporta:

Actualmente Fail2ban establece filtros para Apache, sshd, qmail, vsftpd, lighttpd, Postfix y Courier Mail Server.

Instalación y configuración de Fail2ban

Para instalar Fail2ban tan sólo es necesario la ejecución del siguiente comando:

CÓDIGO:

```
sudo apt-get install fail2ban
```

Tras instalarlo vamos a proceder a editar su archivo de configuración para establecer ciertos parámetros.

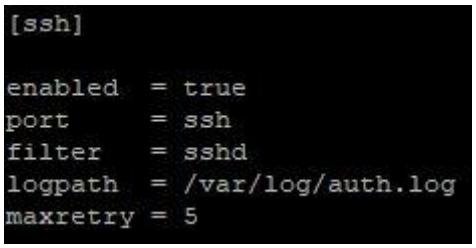
CÓDIGO:

```
sudo nano /etc/fail2ban/jail.conf
```

Comprobaremos que el valor ignoreip esté establecido tal y cómo se muestra a continuación, para que el tráfico proveniente de la red local no lo banee.

```
ignoreip = 192.168.1.0/24
```

Tal y como se ha mencionado anteriormente, Fail2ban permite monitorizar múltiples servicios, en nuestro caso, y para testear su correcto funcionamiento, configuraremos los valores referentes a ssh estableciéndolos tal y como se muestran a continuación:



Configuraremos una de las opciones más interesantes, la posibilidad de recibir notificaciones por correo electrónico.

Es necesario tener instalado y configurado previamente “mailutils” y “ssmtp”, obviaremos este paso puesto que en la configuración de Icinga2 ya lo realizamos.

En el archivo /etc/fail2ban/jail.conf modificaremos la siguiente línea por nuestro correo electrónico:

CÓDIGO:

```
destemail = daviddelriopascual@gmail.com
```

A continuación, buscaremos la siguiente línea:

CÓDIGO:

```
action = %(action_)s
```

Y la sustituiremos por la siguiente:

CÓDIGO:

```
action = %(action_mw)s
```

Para finalizar reiniciaremos el servicio Fail2ban.

CÓDIGO:

```
sudo /etc/init.d/fail2ban restart
```

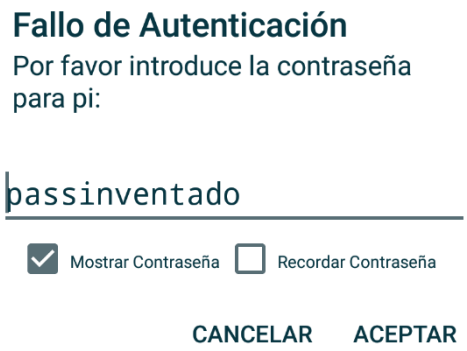
La aplicación ya estaría configurada para banear a los usuarios que mediante el protocolo SSH realizasen 5 intentos fallidos de conexión.

Comprobación del funcionamiento:

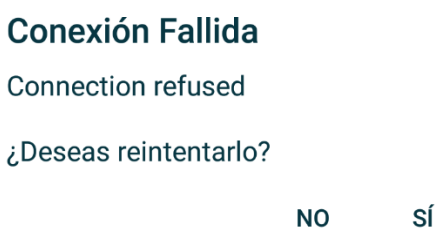
Vamos a comprobar el correcto funcionamiento de Fail2ban, haremos la comprobación desde la aplicación Android “JuiceSSH”, un poderoso cliente SSH para Android.

Configuraremos la nueva conexión con la Raspberry Pi:

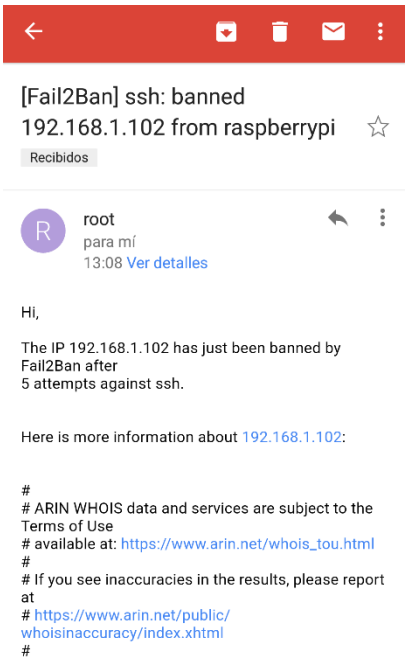
Una vez hayamos añadido la nueva conexión procederemos a usarla. La App nos solicitará una contraseña, nosotros la inventaremos para fallar en el proceso de autenticación hasta 5 veces (el parámetro establecido para banear una IP).



Tras cinco intentos fallidos de conexión usando la contraseña inventada, el sistema denegará el intento de conexión.



Además, inmediatamente recibiremos un correo electrónico notificando sobre el baneo de la IP implicada. En este caso la IP 192.168.1.102 (la de mi dispositivo Android).



Email de notificación recibido

La IP ha sido baneada durante el tiempo establecido en el parámetro bantime del archivo /etc/fail2ban/jail.conf.

Comprobar baneo IP:

Si ejecutamos el siguiente comando con la IP de nuestro sistema y devuelve algún valor, nuestra IP habrá sido baneada:

CÓDIGO:

```
sudo iptables -L -n | grep '192.168.1.102'
```

Quitar baneo a una IP específica:

Para eliminar el baneo a una IP específica debemos ejecutar el siguiente comando:

CÓDIGO:

```
sudo fail2ban-client set ssh unbanip 192.168.1.102
```

(Donde la IP 192.168.1.102 sería la IP del terminal en el que hemos sido baneados).

Por último, reiniciaremos el servicio Fail2ban para que el sistema recoja los cambios realizados:

CÓDIGO:

```
sudo /etc/init.d/fail2ban restart
```

Externalización de servicios

Para finalizar el proyecto, vamos a realizar la externalización de los servicios que componen la infraestructura.

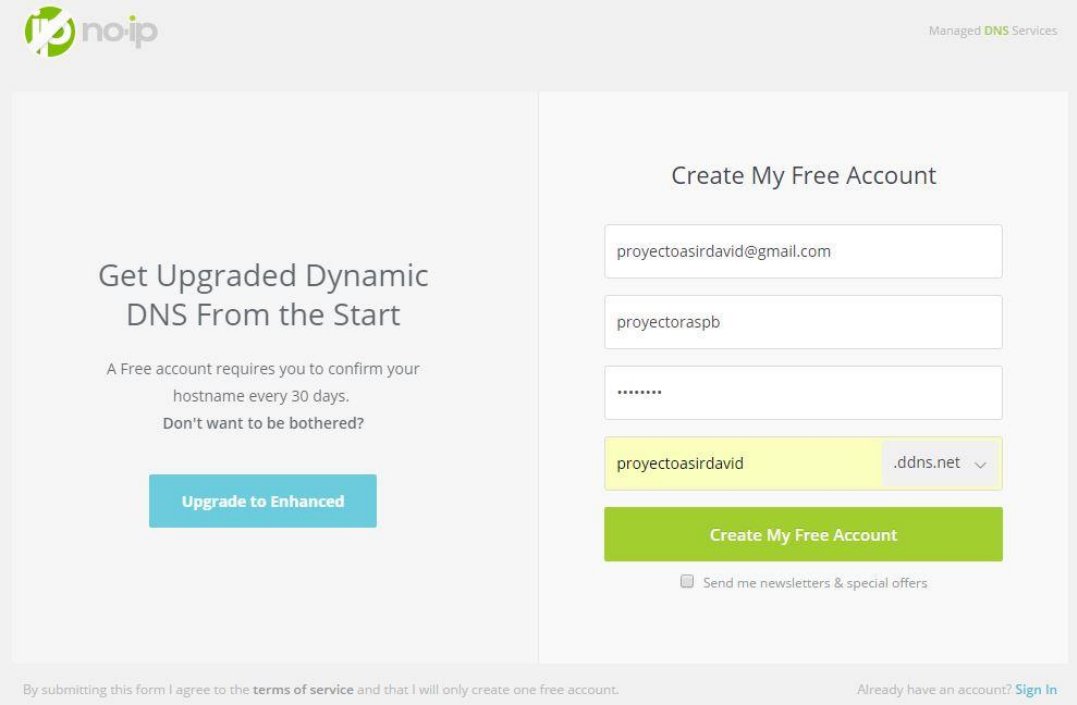
Utilizaremos el servicio https://www.noip.com/. No-IP es un proveedor de DNS dinámico que dispone de versión gratuita durante 30 días. Es un sistema que permite redireccionar automáticamente el dominio que hayamos configurado con nuestra IP externa real. Activaremos también el servicio cliente de actualización dinámica, que permitirá a NO-IP redireccionar nuestra IP y modificar el registro DNS en caso de que se produzca algún cambio en la IP externa.

Configuración NO-IP

Vamos a comenzar con la configuración del servicio NO-IP, para ello crearemos la cuenta que usaremos en la Web.

Previamente, se ha creado la cuenta de correo Gmail proyectoasirdavid@gmail.com, la cual usaremos para el registro en la web de NO-IP.

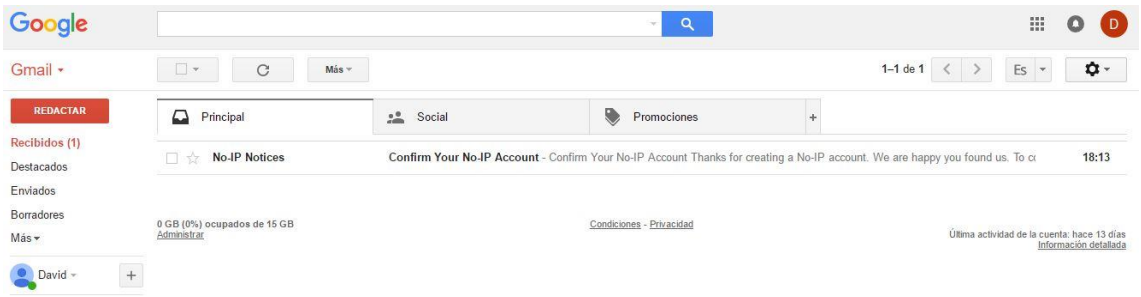
A continuación, en la próxima imagen, se muestran los datos de registro que se han usado para la creación de dicha cuenta:



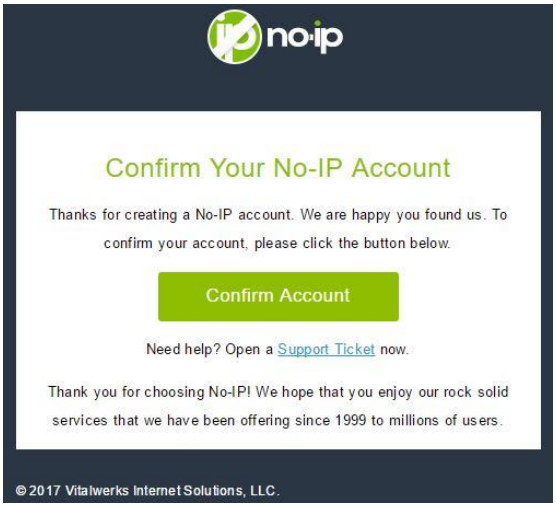
Registro NO-IP

Tal y como se puede ver en la imagen, como cuenta de correo electrónico se ha usado la anteriormente creada en Gmail, como nombre de usuario se usa “proyectoraspb” y como nombre de dominio, que será asociado con nuestra IP, el siguiente “proyectoasirdavid.ddns.net”.

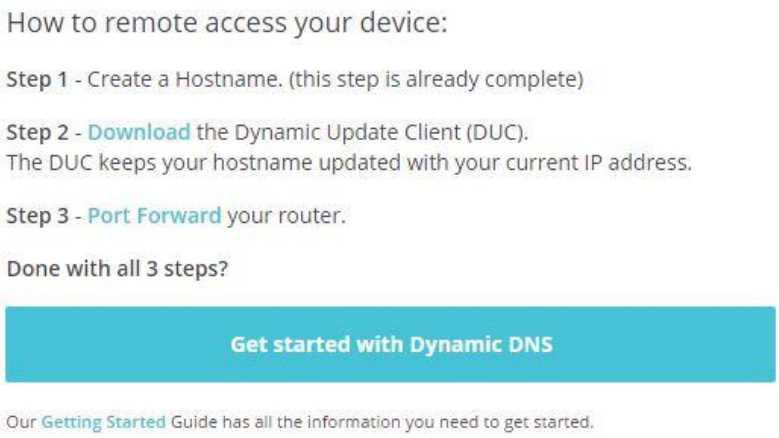
A continuación, el sistema mostrará un mensaje de confirmación, en el cual se solicita la validación de registro mediante correo electrónico.



Por último, confirmaremos la creación de nuestra cuenta.

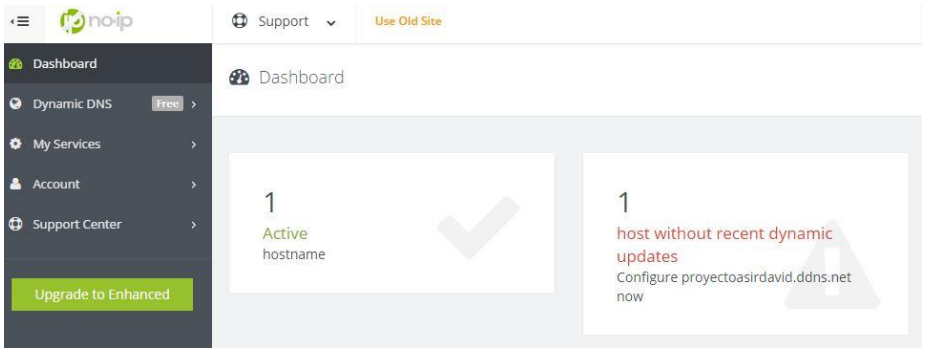


En la pantalla de bienvenida se mostrará un fácil y sencillo resumen, que nos permitirá conocer el procedimiento a seguir para acceder a nuestros servicios internos desde el exterior.

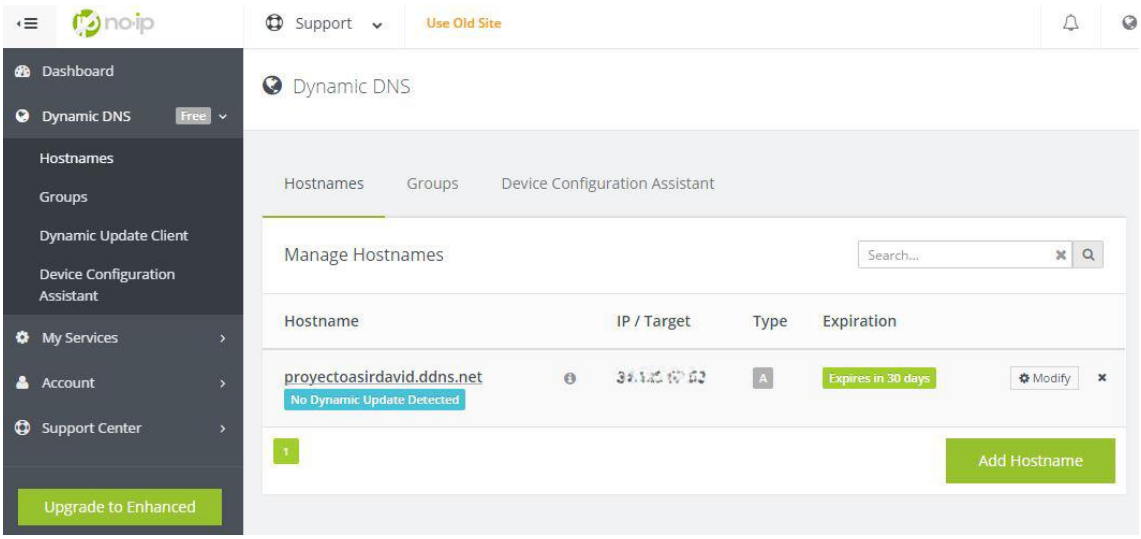


Resumen externalización de servicios

En el tablón de nuestra cuenta veremos un “host” activo.



A continuación, se muestra nuestro dominio “proyectoasirdavid.ddns.net” con la IP externa de nuestra organización asociada al mismo.



Reenvío de puertos en Router:

Para acceder a los servicios desde el exterior debemos configurar nuestro Router, para permitir el acceso a dispositivos de nuestra red local.

Accederemos a nuestro Router a través de la interfaz Web, para ello introduciremos la dirección del mismo en la barra de direcciones del navegador.

Seguidamente, iniciaremos sesión en el sistema mediante las credenciales de acceso.

Acceso a Router

En la siguiente imagen se muestra el resumen final de la configuración del reenvío de puertos realizado en nuestro Router.

Los puertos que han sido habilitados son los siguientes:

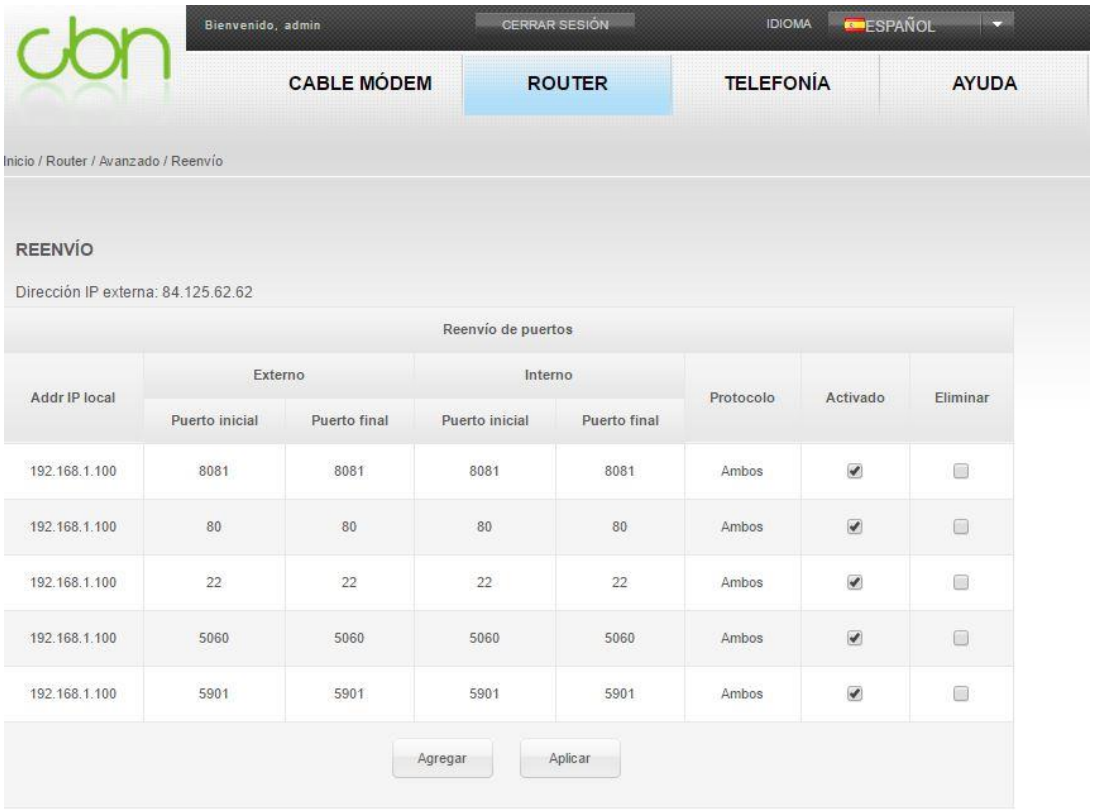
Puerto 8081: Cámara de videovigilancia.

Puerto 80: Nextcloud e Icinga2.

Puerto 22: SSH.

Puerto 5060: Asterisk.

Puerto 5901: VNC.



Resumen reenvío de puertos en Router

Instalación y configuración del cliente de actualización dinámica NO-IP

A continuación, vamos a instalar y configurar el cliente de actualización dinámica NO-IP, esta acción permitirá poder actualizar nuestra cuenta (y por consiguiente el dominio asociado) con la IP externa de nuestra organización, en caso de que haya cambiado.

Comenzaremos situándonos en la carpeta "/usr/local/src" en la que descargaremos dicho software:

CÓDIGO:

```
cd /usr/local/src
```

Descargaremos el cliente de actualización dinámica NO-IP:

CÓDIGO:

```
sudo wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
```

Descomprimiremos y extraeremos dicho software:

CÓDIGO:

```
sudo tar xzf noip-duc-linux.tar.gz
```

Accederemos a dicha carpeta:

CÓDIGO:

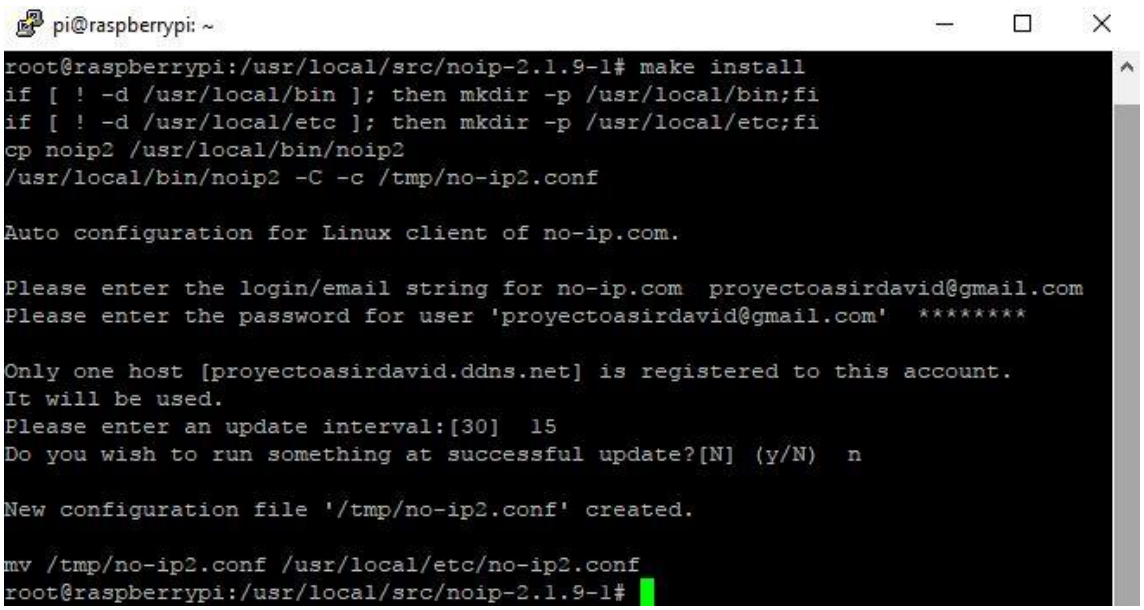
```
cd no-ip-*
```

Instalaremos el paquete:

CÓDIGO:

```
sudo make install
```

Estableceremos las credenciales de nuestra cuenta NO-IP y configuraremos el intervalo de comprobación de actualización de IP.



A continuación, crearemos el script que se iniciará al arrancar el equipo y que permitirá ejecutar el cliente de actualización dinámica NO-IP.

Para comenzar, crearemos el script en "/etc/init.d/" mediante la ejecución del siguiente comando:

CÓDIGO:

```
sudo nano /etc/init.d/noip2
```

Que estará formado por las siguientes líneas de código:

```
#!/bin/sh
### BEGIN INIT INFO
# Provides:          noip
# Required-Start:    $network
# Required-Stop:
# Should-Start:
# Default-Start:     1 2 3 4 5
# Default-Stop:
# Short-Description: Actualiza los registros DNS
# Description:       Actualiza los registros DNS de NoIP
### END INIT INFO

DAEMON=/usr/local/bin/noip2
NOIP_ARGS="-c /usr/local/etc/no-ip2.conf"

PATH=/sbin:/usr/sbin:/bin:/usr/bin:/usr/local/bin

. /lib/lsb/init-functions

case $1 in
start)
log_daemon_msg "Starting NoIP client" "noip2"
start-stop-daemon --start --exec $DAEMON --quiet --oknodo --startas $DAEMON -- $NOIP_ARGS
status=$?
log_end_msg $status
;;
stop)
log_daemon_msg "Stopping NoIP client" "noip2"
start-stop-daemon --stop --quiet --oknodo --exec $DAEMON
log_end_msg $?
;;
esac
```

Asignaremos permisos de ejecución al script mediante el siguiente comando:

CÓDIGO:

```
sudo chmod +x /etc/init.d/noip2
```

Actualizaremos los servicios de arranque y parada:

CÓDIGO:

```
sudo update-rc.d noip2 defaults
```

Mediante la ejecución del último comando ya habremos finalizado con la externalización de servicios de nuestro servidor.

Resumen URL acceso a los servicios:

Motion (Sistema de videovigilancia):

- URL acceso: http://proyectoasirdavid.ddns.net:8081/
- Usuario: “administrador”
- Contraseña: “p@ssw0rd2017”

Nextcloud (Nube privada):

- URL acceso: http://proyectoasirdavid.ddns.net/
- Usuario: “administrador”
- Contraseña: “p@ssw0rd”

IcingaWeb2 (Sistema de monitorización de red):

- URL acceso: http://proyectoasirdavid.ddns.net/icingaweb2/
- Usuario: “administrador”
- Contraseña: “p@ssw0rd”

Asterisk (Centralita VOIP):

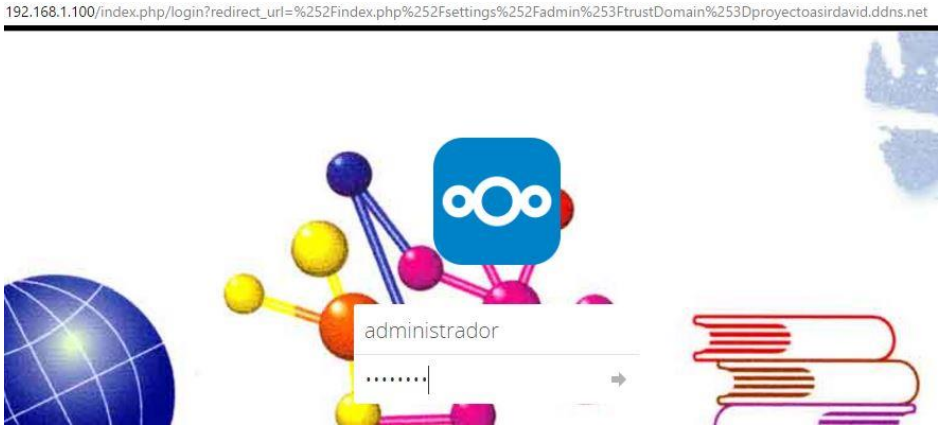
- Servidor: http://proyectoasirdavid.ddns.net:5060
- Usuario: “1001”
- Contraseña: “p@ssw0rd”

Añadir dominio seguro en Nextcloud

Podemos observar que, al intentar acceder a la nube privada anteriormente configurada, Nextcloud notifica el intento de acceso desde un dominio inseguro, por lo que es necesario añadir dicho dominio a la lista de dominios seguros. Para ello pulsaremos sobre el botón “Añadir proyectoasirdavid.ddns.net como dominio de confianza” (Hemos de asegurarnos de realizar esta acción desde la red local ya que el sistema redirigirá a una URL local para realizar la acción).

Error Nextcloud acceso desde dominio inseguro

Accederemos al sistema con las credenciales de administrador.



El sistema mostrará un mensaje de confirmación para “Agregar dominio de confianza”, pulsaremos sobre “Sí” para añadirlo.

Por último, comprobaremos cómo podemos acceder al dominio sin ningún tipo de advertencia.

Conclusiones finales

Es requisito indispensable para la finalización de este curso, la realización de este proyecto. No dudé ni un segundo en usar como elemento principal la Raspberry Pi.

Si bien es cierto que, al comenzar, no conocía en profundidad las grandes posibilidades que ofrece este dispositivo, mis ganas, curiosidad e ilusión me impulsaron a implicarme al 100% en este trabajo.

Siempre surgen dudas… Muy probablemente una de las principales, y que me ha acompañado durante todo este tiempo, ha sido la siguiente: "¿Será suficientemente potente este hardware para soportar lo que quiero implantar?".

A medida que iba haciendo el proyecto, implementando diversos servicios, configurando y programando diversas tareas, me iba dando cuenta de la realidad. La Raspberry Pi es un hardware con un sistema operativo extremadamente optimizado, que gestiona eficientemente sus recursos y que iba a permitir realizar a la perfección todas las ideas que deseaba implementar.

Diverso hardware conectado, hasta cuatro servicios funcionando en paralelo (Streaming en directo, almacenamiento en la nube, monitorización de red, centralita VOIP), múltiples servicios "backend", tales como sevidores Web, bases de datos, programación de tareas…

Muy sorprendido por el rendimiento del sistema y, cuando mi planteamiento inicial tan "sólo" era implementar un sistema de videovigilancia y una nube privada, tuve que aumentar el volumen de trabajo, ya que quería explotar al máximo los recursos de los que dispone este mini ordenador.

La realización de este proyecto ha permitido poner en práctica una gran variedad de conocimientos adquiridos durante el curso y realizar largas jornadas de estudio e investigación.

¿Objetivo inicial? Cumplido. Una empresa puede comenzar la digitalización de la misma con un coste ínfimo.

¿Resultado final? El funcionamiento de todo lo implantado, con un más que solvente rendimiento.

Agradecimientos:

No puedo acabar el proyecto sin agradecer al profesorado del instituto I.E.S Julián Marías su profesionalidad durante todo el curso, así como en lo personal, el magnífico trato recibido por su parte.

Webgrafía:

http://www.juancmejia.com/marketing-digital/transformacion-digital-que-es-beneficios-y-ejemplos-ebook-infografias-videos/

http://www.expansion.com/economia-digital/companias/2015/10/20/5621476cca47411f0d8b45bd.html

http://economia.elpais.com/economia/2016/11/17/actualidad/1479417547_378654.html

https://es.wikipedia.org/wiki/Raspberry_Pi

https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

https://es.wikipedia.org/wiki/Redirecci%C3%B3n_de_puertos

https://es.wikipedia.org/wiki/Traducci%C3%B3n_de_direcciones_de_red

https://en.wikipedia.org/wiki/Cloud_storage

https://colaboratorio.net/davidochobits/sysadmin/2016/nextcloud-almacenamiento-la-nube-mucho-mas-primera-parte/

http://www.levelcloud.net/why-levelcloud/cloud-education-center/advantages-and-disadvantages-of-cloud-computing/

https://nextcloud.com/features/

https://docs.nextcloud.com/server/11/admin_manual/installation/source_installation.html#ubuntu-installation-label

https://github.com/puphpet/puphpet/issues/2246

http://lifehacker.com/how-to-clone-your-raspberry-pi-sd-card-for-super-easy-r-1261113524

http://php.net/manual/es/book.imagick.php

https://docs.nextcloud.com/server/10/admin_manual/configuration_files/encryption_configuration.html

https://docs.nextcloud.com/server/9/admin_manual/configuration_server/background_jobs_configuration.html

https://support.google.com/a/answer/176600?hl=es

https://www.youtube.com/watch?v=aE9B0ghweCo

https://en.wikipedia.org/wiki/Icinga

https://es.wikipedia.org/wiki/Nagios

https://monitoring-portal.org/index.php?thread/33940-solved-icinga-web-2-installation/

http://php.net/manual/es/timezones.europe.php

http://root.ve.cx/2016/04/16/icinga-2-not-starting-after-update-icinga-2-4/

https://alejandronieto90.wordpress.com/2015/04/16/instalacion-y-configuracion-de-icinga-2-en-arquitectura-distribuida/

https://www.howtoforge.com/tutorial/install-icinga2-and-icingaweb2-on-centos-7/

https://github.com/Icinga/icingaweb2/blob/master/doc/02-Installation.md

http://www.aradaen.com/programacion/mysql/cheat-sheet-mysql-privilegios-de-usuario/

https://packages.debian.org/jessie/admin/icinga2-ido-mysql

http://packages.ubuntu.com/trusty/software-properties-common

http://searchdomino.techtarget.com/answer/What-does-it-mean-to-populate-a-database

https://packages.debian.org/jessie/admin/software-properties-common

http://frambuesa-pi.blogspot.com.es/2015/04/raspberry-pi-primeros-pasos-con.html

http://www.daviddelrio.es/configurar-ip-estatica-en-raspberry-pi-3/

https://monitoring-portal.org/index.php?thread/33662-solved-notification-definition-in-icinga-2/

https://docs.icinga.com/icinga2/latest/doc/module/icinga2/toc#!/icinga2/latest/doc/module/icinga2/chapter/monitoring-basics#using-apply-notifications

https://github.com/Icinga/icinga2/blob/master/etc/icinga2/conf.d/hosts.conf

https://www.youtube.com/watch?v=AFoHRjBfxmw

https://git.icinga.com/github-mirror/icinga2/blob/cdd5c0a716127f00a66c7a2718bed4763f98658b/doc/3.04-notifications.md

http://www.raspberry-projects.com/pi/software_utilities/email/ssmtp-to-send-emails

https://github.com/Icinga/icinga2/issues/3366

https://es.wikipedia.org/wiki/Fail2ban

https://geekytheory.com/tutorial-raspberry-pi-16-seguridad-con-fail2ban

http://www.doc.ic.ac.uk/~pg1712/blog/fail2ban-in-ubuntu/

http://www.alevsk.com/2015/08/forma-correcta-de-desbanear-direcciones-ips-en-fail2ban/

https://www.pantz.org/software/cron/croninfo.html

https://www.raspberrypi.org/documentation/linux/usage/cron.md

https://superuser.com/questions/327762/how-to-find-a-directory-on-linux

https://es.wikipedia.org/wiki/Fstab

http://www.linuxquestions.org/questions/linux-newbie-8/how-to-format-hard-drive-in-ext4-751168/

http://www.sagraramirez.es/index.php/sistema-operativo/114-conocer-el-tamano-que-ocupa-una-carpeta-en-linux-por-consola

https://www.trucoslinux.es/copia-de-archivos-con-rsync/

https://docs.nextcloud.com/server/10/admin_manual/maintenance/backup.html

https://www.zoiper.com/en/voip-softphone/download/zoiper3?cid=home-dlb#windows/step3

http://www.asterisk.org/get-started

http://www.todostartups.com/bloggers/22-motivos-y-ventajas-para-usar-telefonia-ip-en-tu-empresa-por-artaizasesoria

https://es.wikipedia.org/wiki/Voz_sobre_protocolo_de_internet

https://www.youtube.com/watch?v=heL_hrg5kXQ

http://www.voipforo.com/asterisk/configuracion-extensions-conf.php

https://www.mail-archive.com/asterisk-users@lists.digium.com/msg191074.html

https://openwebinars.net/blog/tutorial-asterisk-cli-command-line-interface/

http://foro.elhacker.net/dudas_generales/para_que_sirve_realmente_el_noip-t152155.0.html

https://en.wikipedia.org/wiki/No-IP

http://trasteandoarduino.com/2013/12/26/instalando-el-cliente-noip-en-la-raspberry-pi/

http://www.noip.com/support/knowledgebase/installing-the-linux-dynamic-update-client/

https://enavas.blogspot.com.es/2008/12/update-rcd-actualizando-el.html

https://www.voztovoice.org/?q=node/76