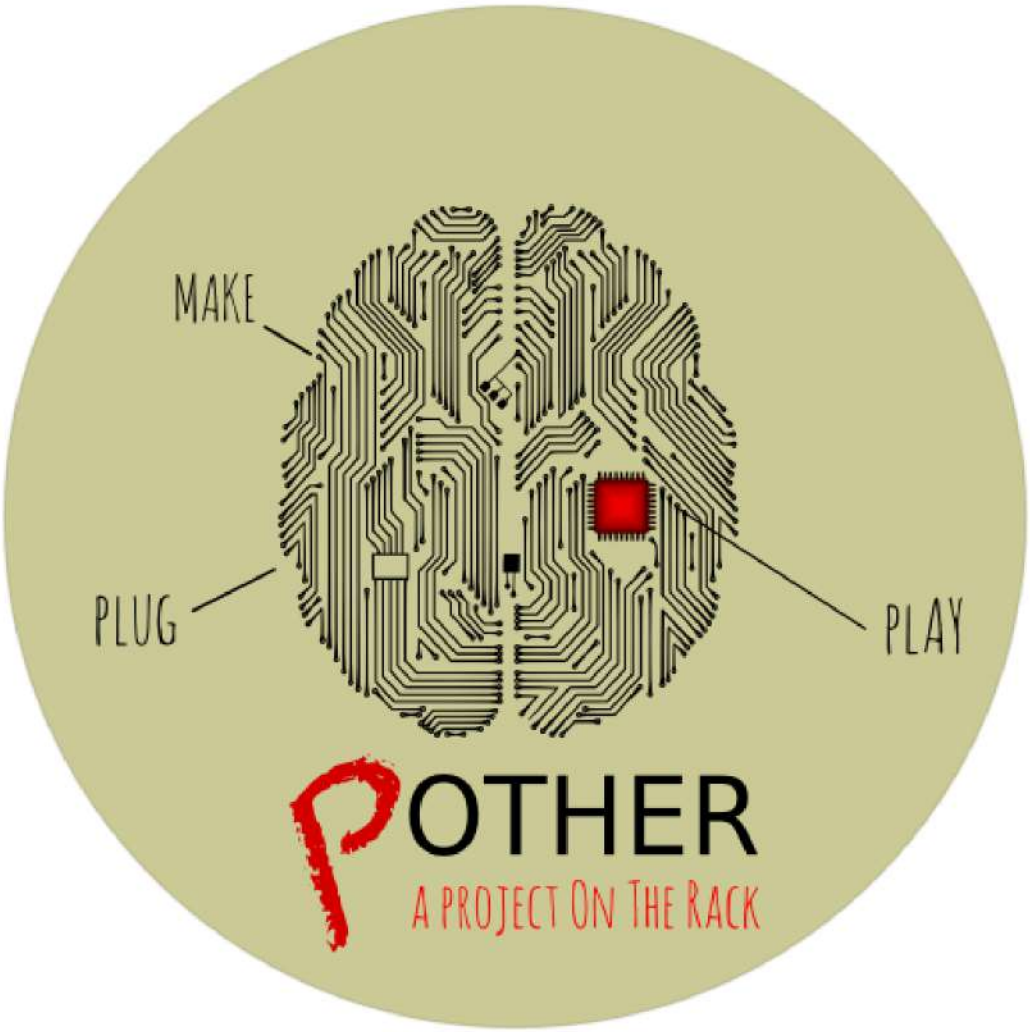
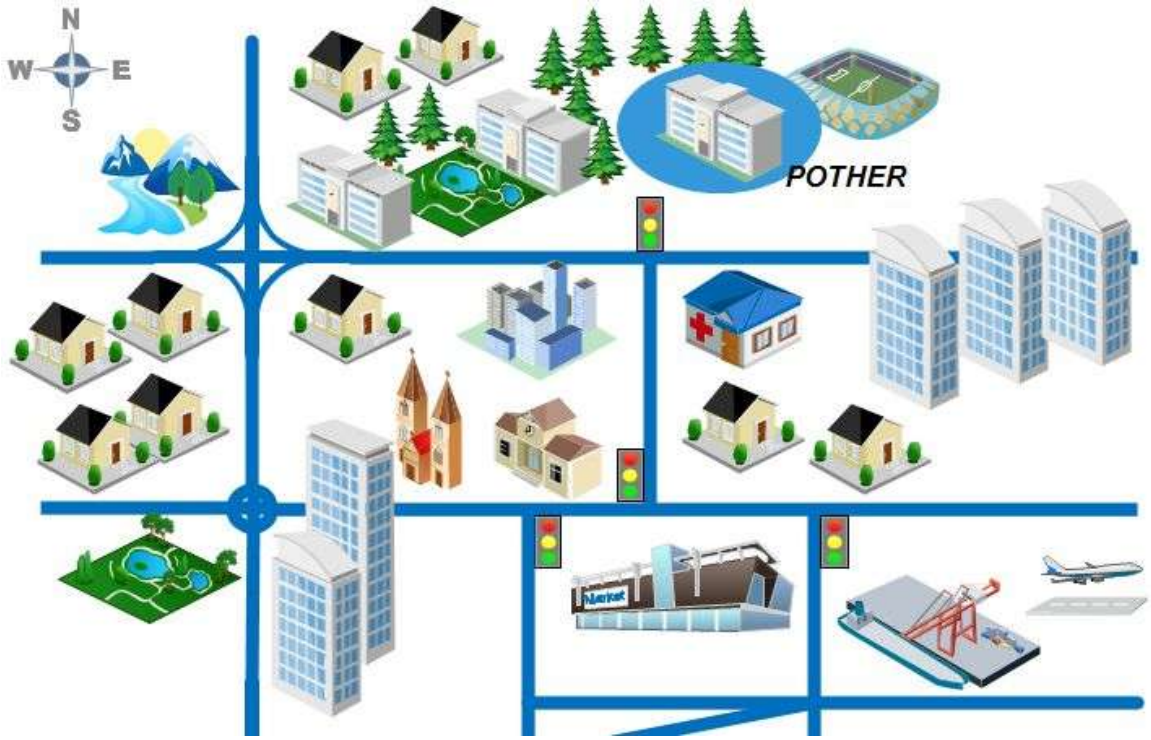



***Carlos del Cerro, Óscar Ramírez y David Burgos***

# *Índice*

# *Introducción*

La empresa Pother (Project on the rack) ha solicitado los servicios a nuestra empresa COD para poner en marcha el proyecto Project On The Rack. El proyecto se realizará en la UEM en el campus de Villaviciosa, concretamente el proyecto estará en el aula C308 y desde ahí se dará el servicio.




Nosotros les ofreceremos las soluciones hardware y software para resolver este supuesto que nos plantean y resolver así el supuesto que nos plantean. Este proyecto tendrá como objetivos principales:

- Poner a punto los servidores.
- Instalar, configurar y administrar los SO.
- Configurar las Raspberrys.
- Instalar, configurar y administrar la BBDD.

- Creación de un portal que dará servicio de entrada a los CMS de Wordpress.
- Implantar un cluster de alta disponibilidad.
- Implantar seguridad física y lógica.
- Realizar auditoria del proyecto.
- Usar las tecnologías utilizadas en clase.
- Dar seguridad a los sistemas que pondremos en marcha.

# *Empresa Code Or Digit*

Somos una empresa de consultoría y soporte IT que da soluciones a PYMES fundada en Enero de 2017. Queremos satisfacer las necesidades que el mercado tecnológico demanda en la actualidad, aportando soluciones tanto software como hardware.

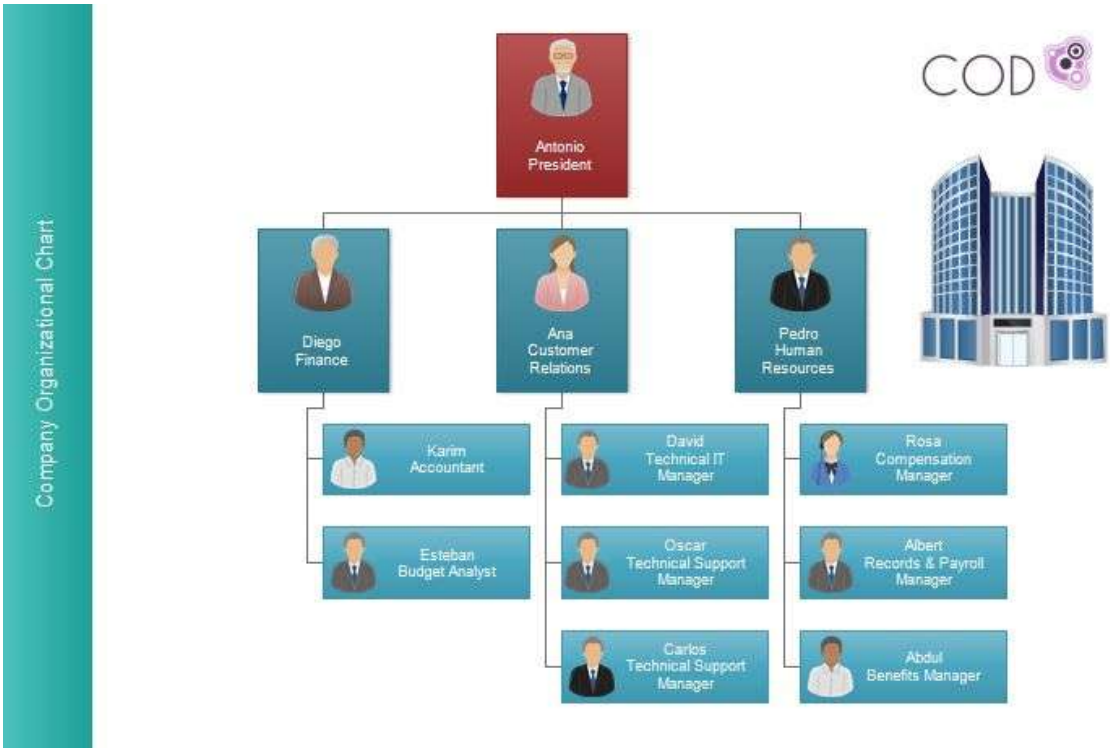
Utilizamos las tecnologías más avanzadas, aportamos personas idóneas para cada proyecto y con un alto nivel de cualificación.

Nuestro portfolio de actividades abarca desde:

- Mantenimiento y soporte tecnológico.
- Gestión de aplicaciones y desarrollo del software.
- Análisis y Diseño de soluciones.

El conocimiento de nuestros técnicos abarca las principales entornos tecnológicos utilizados en el mercado: Bases de datos (Oracle, SQL Server, MySQL), Lenguajes de Programación (Javascript, PHP, HTML, CSS, XML , Scripting), Aplicaciones de gestión empresarial.

Para ello basamos nuestra metodología en 4 pilares fundamentales:
-Trabajo en Equipo
-Cercanía
-Proactividad
-Flexibilidad




# *Miembros del equipo COD*

El equipo asignado para realizar el proyecto Pother es el equipo de técnicos COD está formado por tres alumnos de 2º Administración de Sistemas Informáticos en Red, los cuales, darán una solución al proyecto integrador Project On The Rack.

Carlos del Cerro Tenorio
Óscar Ramírez
David Burgos

## *Asignaturas que engloban el proyecto*

- Administración de Sistemas Operativos
- Servicios de Red e Internet
- Seguridad y Alta Disponibilidad
- Implantación de Aplicaciones Web
- Sistemas Gestores de Bases de Datos

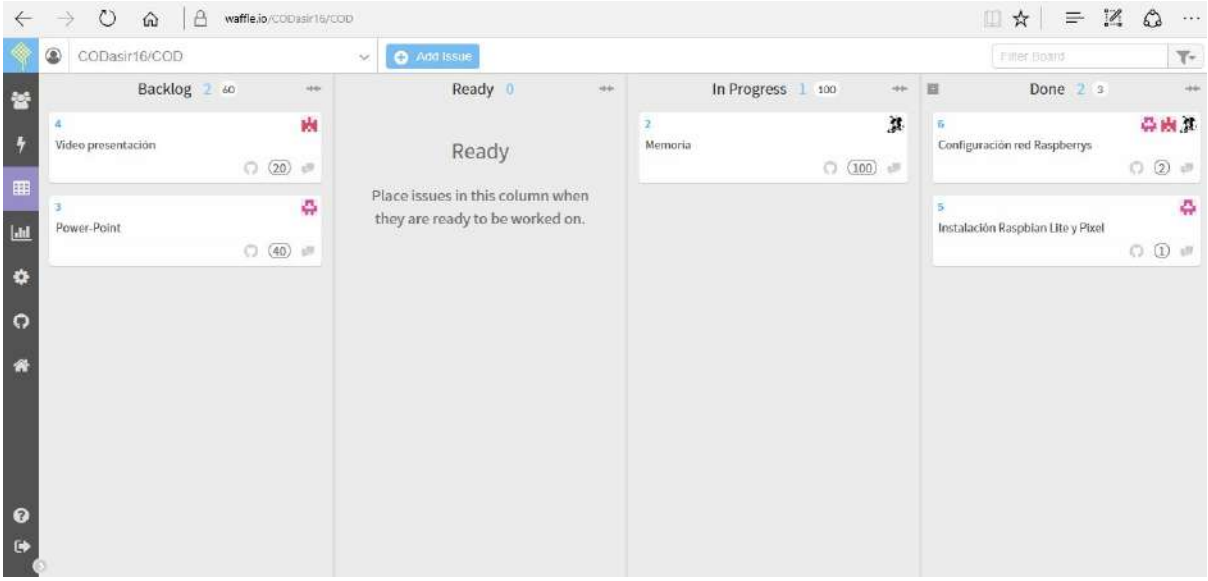
## *Tecnologías utilizadas*

- Servidores Rack, sobremesa y Raspberrys.
- Sistemas operativos Windows y Linux.

- Equipos y material de red (switchs, routers, cableado, …).
- Software HTTP, FTP, proxy…
- Software opensource para la implantación del sistema y desarrollo de la aplicación web.
- HTML, CSS, Javascript y PHP.
- MySQL y DIA.
- Software de clonezilla para la implantación de la maquetas de los equipos por red y equipo a equipo.
- Aplicación web kanban  https://waffle.io .

### *Metodología ágil*

Para ser buenos gestores de nuestro proyecto integrador utilizaremos los métodos y técnicas ágiles como Scrum o Kanban (popularizada en Japón). Esta metodología de post-it que se mueven en un tablero es de gran ayuda para el trabajo en equipo. Nosotros utilizaremos Waffle que es una herramienta que está enlazada con GitHub para proyectos software. Para ello crearemos un tablero donde el responsable o Scrum Master pondrá las tareas e irá actualizando el tablero. Un buen Scrum Master necesita saber cómo integrar estos métodos dentro de su proceso de desarrollo en el equipo COD para el buen resultado del proyecto integrador. Algunos miembros del equipo trabajamos el año pasado con una herramienta llamada Trello para realizar el proyecto integrador Star Walls de 1ºASIR.




# Implantación y configuración de servidores

Lo primero que haremos será realizar un inventario de los servidores que tenemos disponibles para realizar el proyecto. Para ello instalaremos Ubuntu en los

servidores y una vez hecho esto, entraremos en el terminal de comandos. Realizaremos un inventariado con el comando "lshw" podremos ver todos los componentes conectados al equipo, pero a esto se le puede añadir el parámetro "-hml", lo que hará que el inventariado del equipo se muestre en código HTML para así poder verlo desde un navegador con un mejor diseño, además, hay que indicar en que se quiere exportar a un archivo .html.
Así que el comando utilizado para inventariar el servidor ubuntu fue **lshw -html > invent.html** .

**Inventario Stark**

| | |
|---|---|
| **id:** | ubuntu |
| **descripción:** | Chasis de montaje rack |
| **producto:** | Precision WorkStation R5400 () |
| **fabricante:** | Dell Inc. |
| **serie:** | 8PPQX4J |
| **anchura:** | 64 bits |
| **capacidades:** | *smbios-2.5 dmi-2.5 vsyscall32* |

(Todos los inventariados los tenemos en el anexo de inventarios).

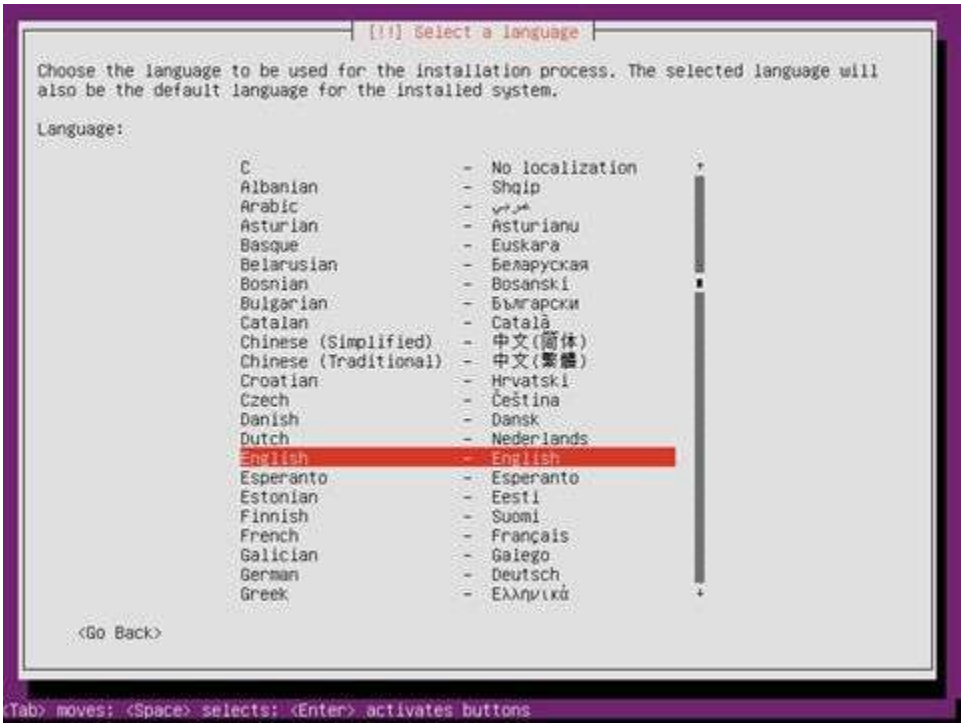
## *STARK (NAS)*

### *Instalación Ubuntu 16.04 Server.*

Seleccionamos lenguaje y comenzamos la instalación de Ubuntu Server 14.04.


Podemos ver el Asistente Gráfico de Ubuntu Server 14.04 (GUI, Graphical User Interface).


A continuación, elegimos el idioma que vamos a usar, la ubicación del teclado y a continuación nos detectará el HW.




Ponemos un nombre a la máquina y registramos un usuario. Durante la instalación tendrás que introducir una clave para el usuario (recomendado al menos 8 caracteres incluyendo números y letras). Podemos además cifrar nuestra carpeta personal. Y comenzaría la instalación del SO propiamente dicho.


Nos pregunta si necesitamos un proxy, en nuestro caso se queda en blanco. Comienza la instalación de los programas y aplicaciones. De primeras le decimos que no queremos actualizaciones. Seleccionamos otros programas si queremos instalarlos. Lo hacemos pulsando la barra espaciadora y presionamos continuar. Para configurar el sitio de correo seleccionamos Sitio de Internet. Instala además el gestor de arranque GRUB.

Ya tenemos la instalación completa, solo debemos retirar la iso del arranque y podremos entrar a nuestro nuevo SO Ubuntu Server 14.04.


***LANNISTER***

En este servidor configuraremos el proxy. Después de buscar una solución al problema con el servidor, instalaremos un sistema en una raspberry e instalaremos squid para que haga de servidor proxy.
La raspberry utiliza Ubuntu 14.04. Crearemos el usuario llamado lannister, después añadiremos el usuario lannister al grupo sudoers y modificamos el usuario por defecto. Una vez hecho esto descargamos Squid3 y editaremos el archivo /etc/network/interfaces y asignaremos la ip fija 192.168.1.2 para la interfaz de red eth0. Editamos el archivo /etc/hostname cambiando el nombre por defecto por lannister y por último haremos un apt-get update para actualizar nuestro sistema.

Además hemos modificado la distribución del teclado de inglés a español.

## *TARGARYEN*

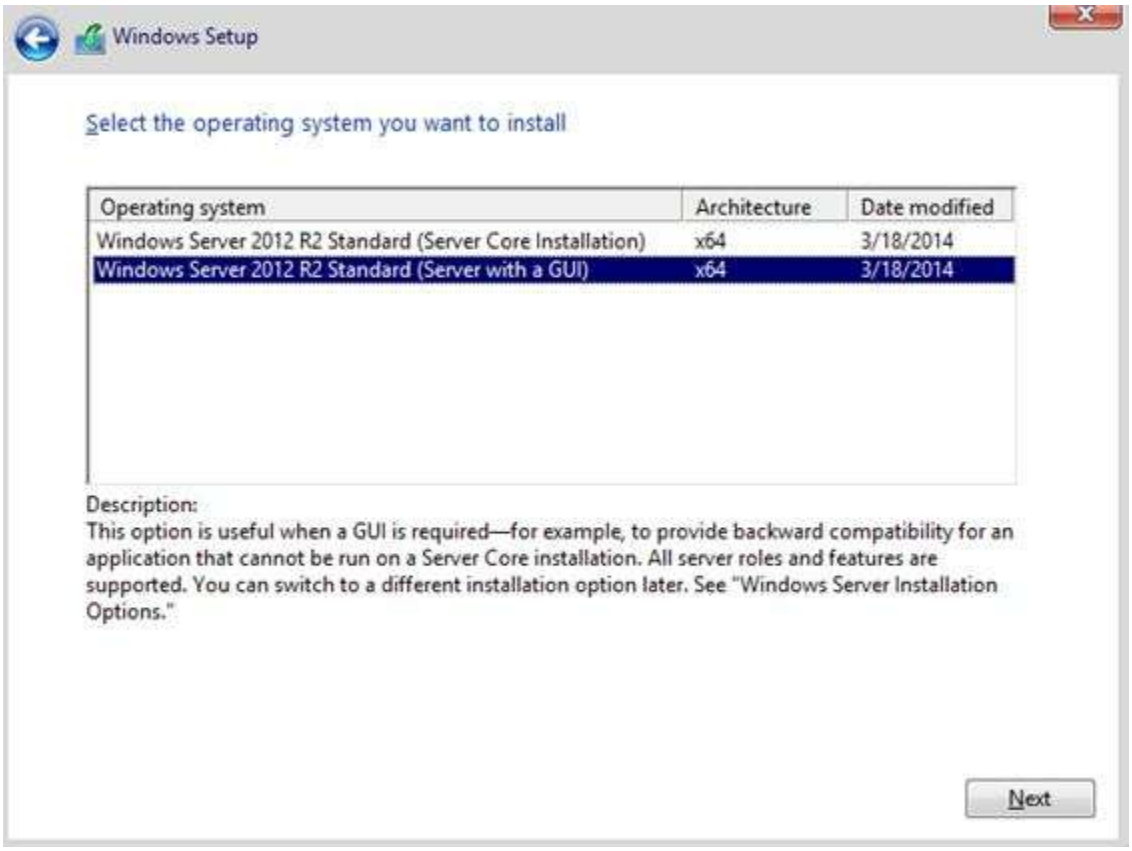
### *Instalación Windows Server 2012 en una de las máquinas compartidas en el rack para Active Directory*

Seleccionamos la imagen (iso que hemos descargado previamente) y arrancamos la máquina desde donde la tengamos.

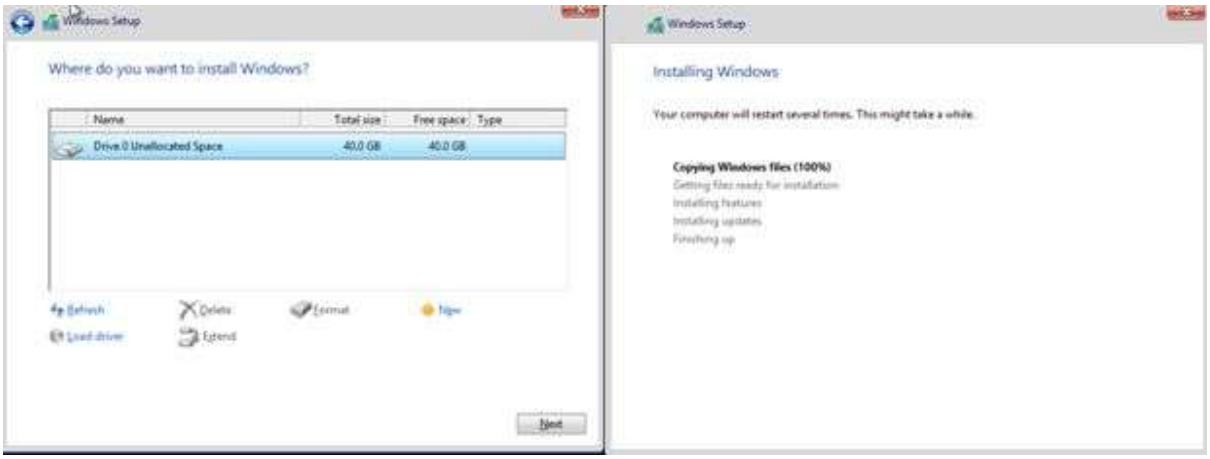

Seleccionamos leguaje, hora y teclado para posteriormente introducir la clave de activación del producto.

Elegimos Server con entorno gráfico (GUI, Graphical User Interface).

A continuación, aceptamos los términos de la licencia.



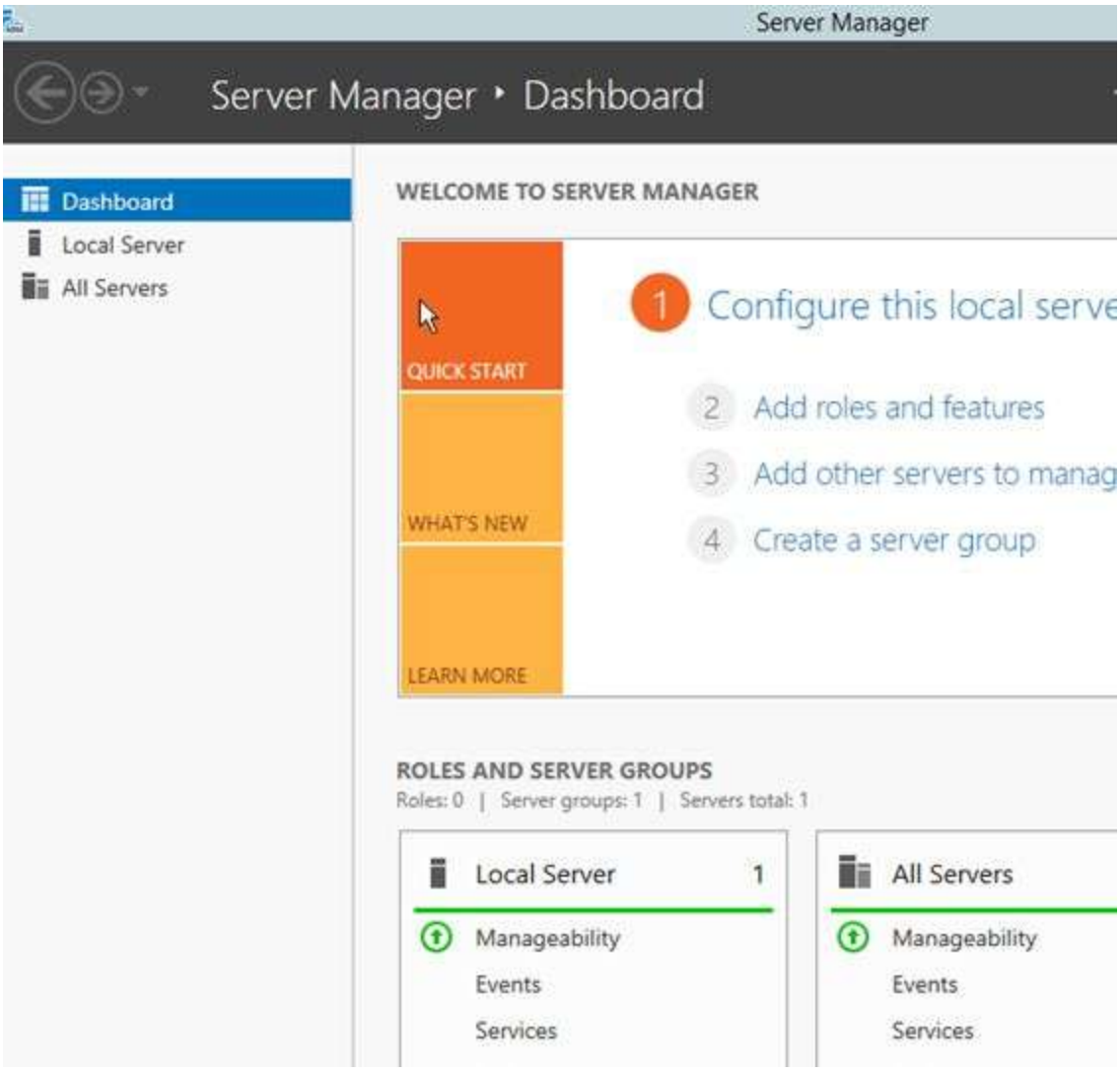

Seleccionamos dónde vamos a instalar el SO y comienza la instalación del mismo.



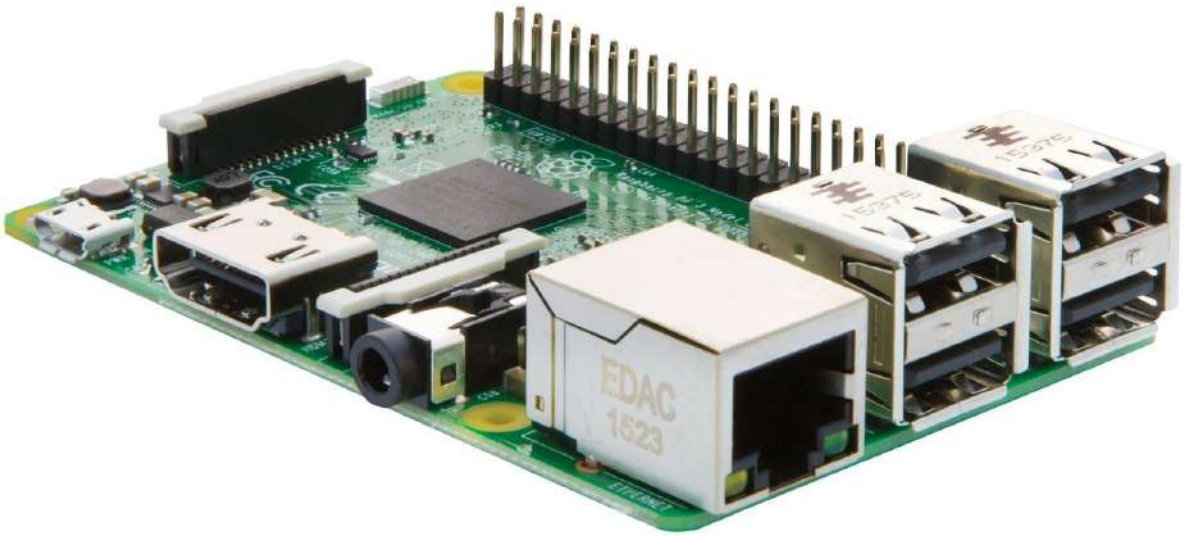

Durante la instalación tendrás que introducir una clave de Administrador (al menos 8 caracteres incluyendo números y letras).


Con esto ya tendríamos la instalación completada y nuestro SO listo para empezar a trabajar con él.

Esta es la apariencia que tiene nuestro escritorio al entrar por primera vez a Windows Server 2012 R2. Salta directamente el Server Manager.



## *RASPBERRY PI*




***Documentación Raspberry Pi Lite COD 2***

*Hardware*

Procesador:

- Chipset Broadcom BCM2387.
- 1,2 GHz de cuatro núcleos ARM Cortex-A53 GPU

Conectividad

- Ethernet
- Salidas
    - HDMI
    - RCA
    - jack de 3,5 mm de salida de audio, HDMI
    - USB 4 x Conector USB 2.0
- Ranura de tarjeta de memoria SDIO

*Sistemas*

Init

User: Pi

Password: Raspberry

Nuevo user y contraseña: “cod2”  “doc”

Damos todos los permisos al nuevo user con “usermod -g sudo cod”

***Documentación Raspberry Pi Pixel COD 1***

*Hardware*

Procesador:

- Chipset Broadcom BCM2387.

- 1,2 GHz de cuatro núcleos ARM Cortex-A53 GPU

Conectividad

- Ethernet
- Salidas
    - HDMI
    - RCA
    - jack de 3,5 mm de salida de audio, HDMI
    - USB 4 x Conector USB 2.0
- Ranura de tarjeta de memoria Micro SDIO

## *Sistemas*

Instalación de los sistemas Lite y Pixel, con los que probaremos versión TUI y GUI. Después dejaremos en las dos Raspberrys solamente el entorno en modo texto. Estos sistemas operativos están basados en la distribución Debian 8 Jessie.

Seguiremos los siguientes pasos:

1. Descargar SO en nuestro PC desde https://www.raspberrypi.org/downloads/raspbian/
2. Insertamos o conectamos la tarjeta SD en nuestro PC
3. Descomprimimos el SO que descargamos como fichero zip en la SD.
4. Una vez termina de copiarse todo, sacamos la tarjeta SD y la insertamos en la Raspberry Pi
5. Conectamos el cable HDMI, pantalla, teclado, ratón y cable de red.
6. Conectamos el cable de alimentación eléctrica y empezará a arrancar la Raspberry Pi.
7. Elegimos el idioma de instalación y el idioma del teclado.
8. Comenzamos a instalar y ya tendremos nuestras Raspberrys preparadas.

Inicialmente entraremos con estas credenciales a nuestras Raspberrys.
**User:** Pi
***Password:*** Raspberry

La primera tarea que hemos realizado es instalar los SO a las Raspberrys y posteriormente hemos creado un usuario con las iniciales de los integrantes del equipo en cada Raspberry (COD) y le hemos incluido en el grupo sudo utilizando el comando "usermod -g sudo cod"
Crearemos nuevo user y contraseña, para eliminar las que vienen por defecto: “cod” “******”
Damos todos los permisos al nuevo usuario con “usermod -g sudo cod”

Definimos dos IPs una estática y una virtual (ver ejemplo abajo) por DHCP en cada Raspberry así como DNS local en /etc/hosts asignando nombres a las dos máquinas COD1 y COD2.

```
cod1@raspberrypi:~$ cat /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'

# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

auto eth0:1
iface eth0:1 inet static
        address 192.168.1.41
        netmask 255.255.255.0

allow-hotplug wlan0
iface wlan0 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

allow-hotplug wlan1
iface wlan1 inet manual
    wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
cod1@raspberrypi:~$
```

## *Administración*

Comenzamos a trabajar de manera directa con las Raspberrys. Configuramos las pantallas, teclados y ratón a las Raspberrys para poder trabajar en ellas, instalar el sistema operativo Rasp Lite y Rasp Pixel para probar las interfaces de este sistema, TUI y GUI. Como hemos podido comprobar el sistema gráfico consume más recursos de las Raspberrys. Por lo que hemos decidido pasar las dos Raspberrys al interfaz en modo texto.

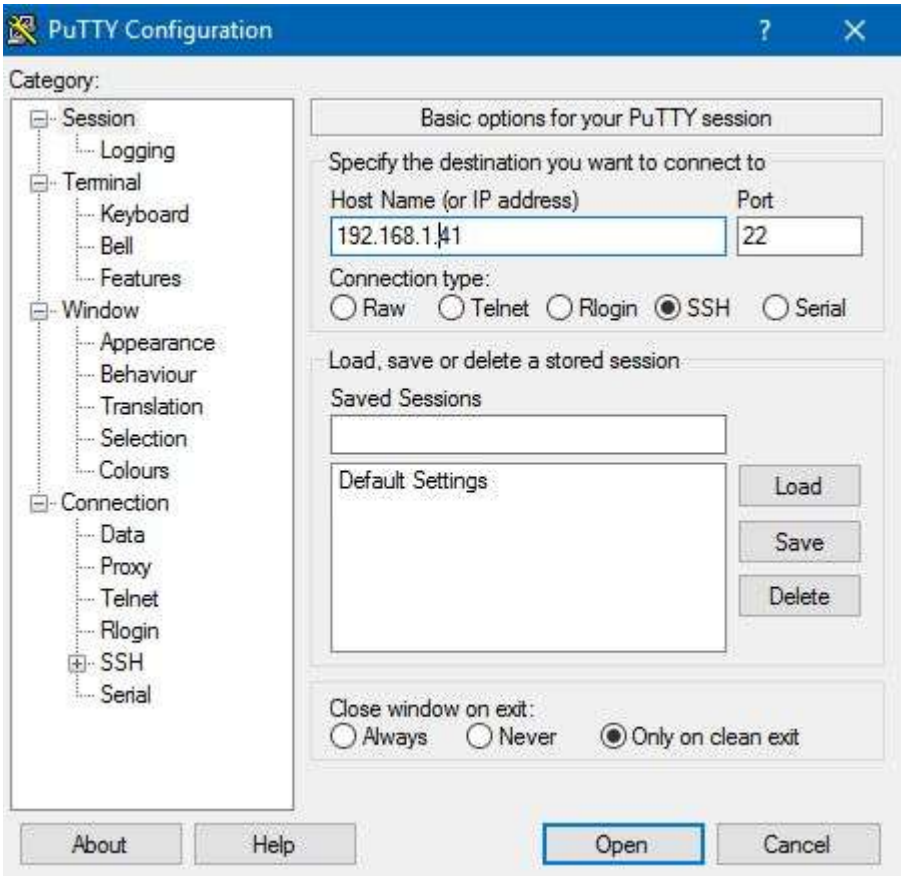
***Trabajando con las Raspberrys***

Lo primero de todo es listar el HW de los servidores y las Raspberrys con: lshw -html > invent.html
Con esto tendremos un listado del Hardware en formato HTML.

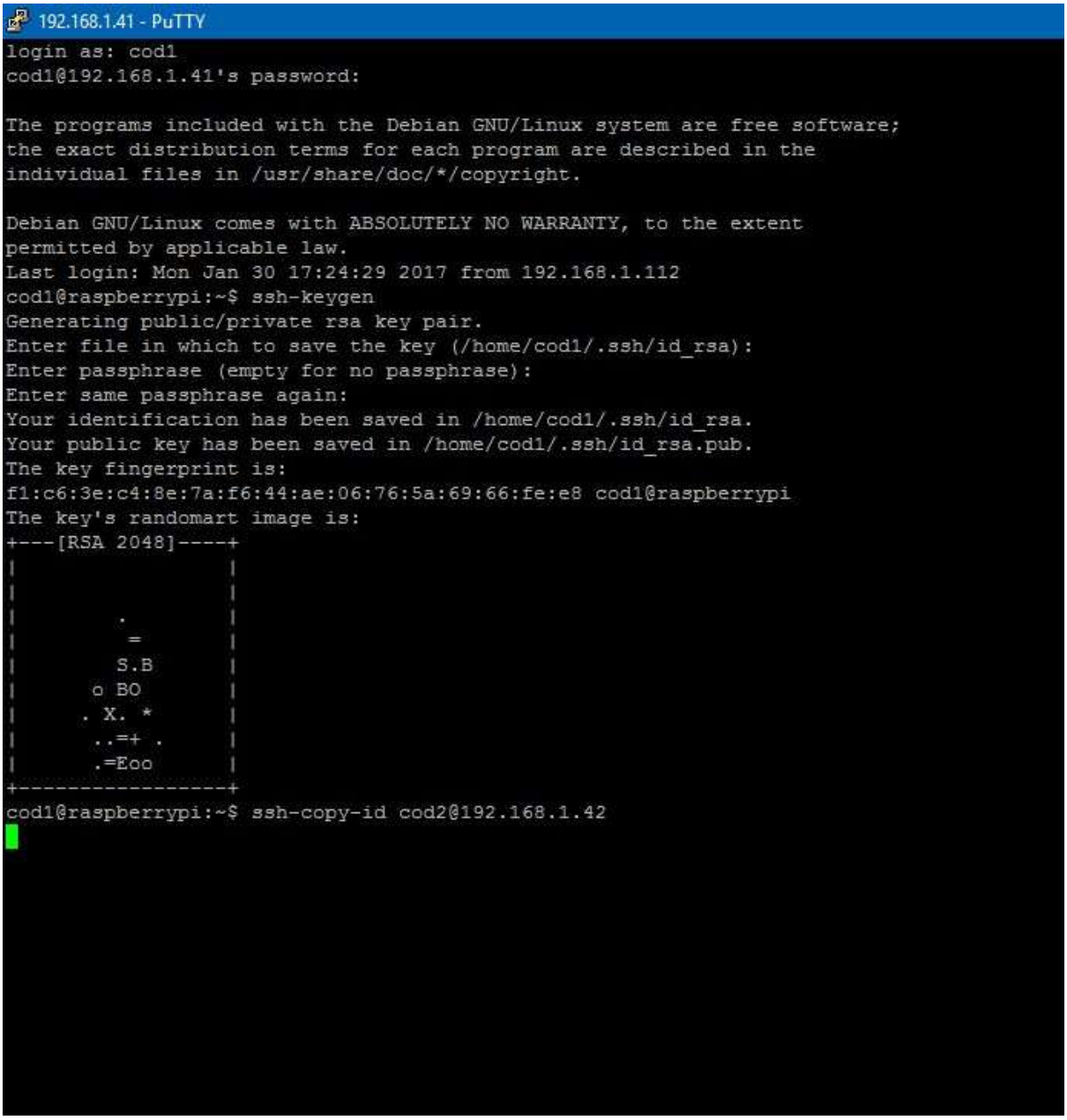
Además chequeamos ambas máquinas con los comandos: nmap,iftop, (ASOpedia).

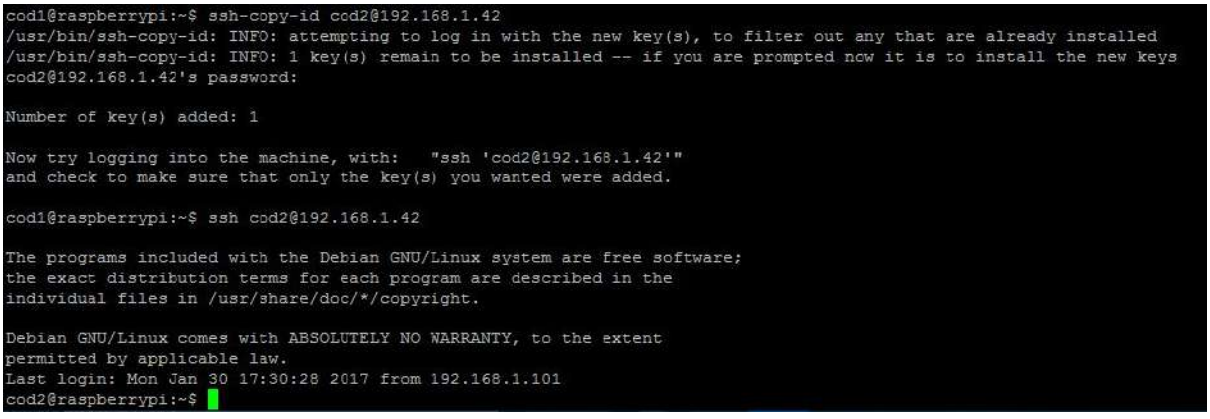
En clase hemos creado un NAS (Network Attached Storage) con un server ubuntu para dar servicio a todos los grupos y provisionalmente las Raspberrys se conectaran y almacenarán en él las copias de seguridad.

Para trabajar de forma remota cada miembro del grupo COD tenemos instalado en nuestro portátil el software Putty para conectarnos de manera remota a las Raspberrys y los servidores.
Como vemos en la siguiente captura nos vamos a conectar a la Raspberry COD1.



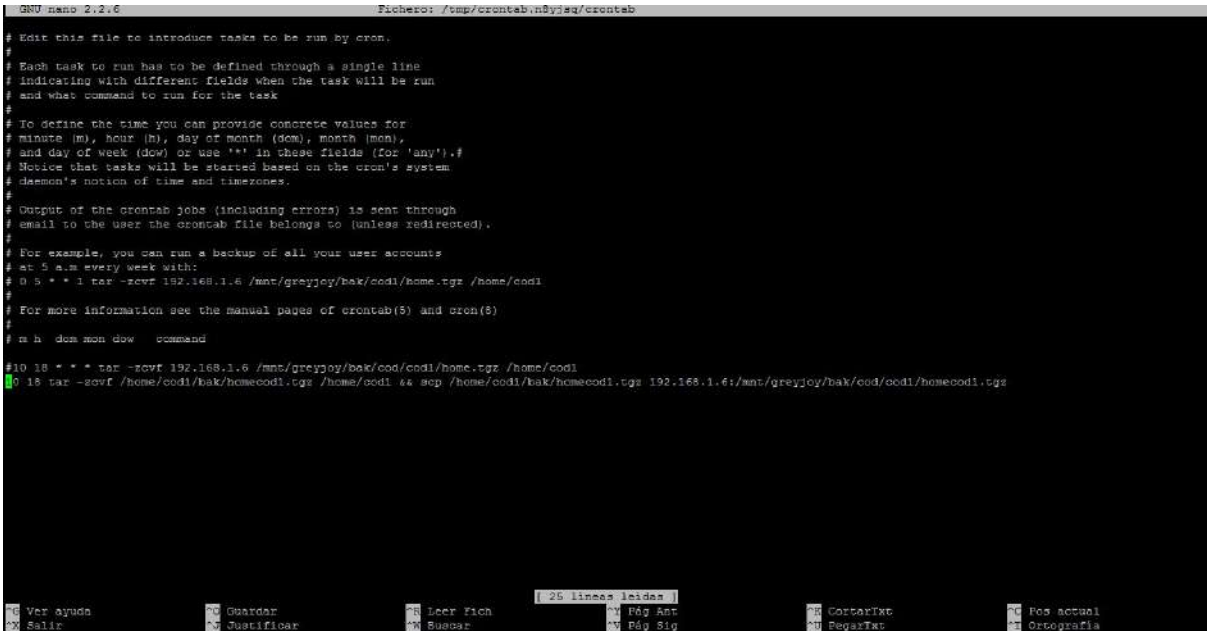

Crear par de claves para conexión sin password entre Raspberrys, para ello utilizaremos ssh keygen y vamos a sincronizar entre Raspberrys COD1 y COD2 para generar confianza entre ellas. Creamos así el par de claves para conexión sin password entre Raspberrys. Haremos los mismos pasos con el servidor NAS.


192.168.1.41 - PuTTY
login as: cod1
cod1@192.168.1.41's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jan 30 17:24:29 2017 from 192.168.1.112
cod1@raspberrypi:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cod1/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cod1/.ssh/id_rsa.
Your public key has been saved in /home/cod1/.ssh/id_rsa.pub.
The key fingerprint is:
f1:c6:3e:c4:8e:7a:f6:44:ae:06:76:5a:69:66:fe:e8 cod1@raspberrypi
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|                 |
|        .        |
|         =       |
|        S.B      |
|       o BO      |
|      . X. *     |
|       ..=+ .    |
|       .=Eoo     |
+-----------------+
cod1@raspberrypi:~$ ssh-copy-id cod2@192.168.1.42


cod1@raspberrypi:~$ ssh-copy-id cod2@192.168.1.42
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
cod2@192.168.1.42's password:
Number of key(s) added: 1
Now try logging into the machine, with:   "ssh 'cod2@192.168.1.42'"
and check to make sure that only the key(s) you wanted were added.
cod1@raspberrypi:~$ ssh cod2@192.168.1.42
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jan 30 17:30:28 2017 from 192.168.1.101
cod2@raspberrypi:~$

Haremos los mismo con los servidores, exportar claves públicas desde las Raspberrys sobre el NAS para acceso sin contraseña.

Enviar imágenes de las copias de seguridad de las Raspberrys como tarea de cron. Para ello crearemos un script dentro de Crontab que empaquetará las copias o bien con tar o con tgz.




10 18 tar -zcvf /home/cod1/bak/homecod1.tgz /home/cod1 && scp /home/cod1/bak/homecod1.tgz 192.168.1.6:/mnt/greyjoy/bak/cod/cod1/homecod1.tgz

Cambiamos los permisos de la carpeta cod del NAS donde se guardan las copias de seguridad.

```
cod1@stark:/mnt/greyjoy/bak$ chmod 774 cod
cod1@stark:/mnt/greyjoy/bak$ sudo chgrp cod cod
cod1@stark:/mnt/greyjoy/bak$ ls -l
total 20
drwxr-xr-x 2 bdj  sudo    4096 feb  3 19:54 bdj
drwxrwxr-- 4 cod1 cod     4096 feb  3 20:07 cod
drwxrwxr-- 4 dj1  djgroup 4096 feb  3 20:08 dj
drwxr-xr-x 2 jmc  sudo    4096 feb  3 20:13 jmc
drwxr-xr-x 2 root root    4096 feb  3 19:34 nd
cod1@stark:/mnt/greyjoy/bak$
```

Cambiamos el grupo y los permisos del mismo de la carpeta donde están las copias de seguridad.

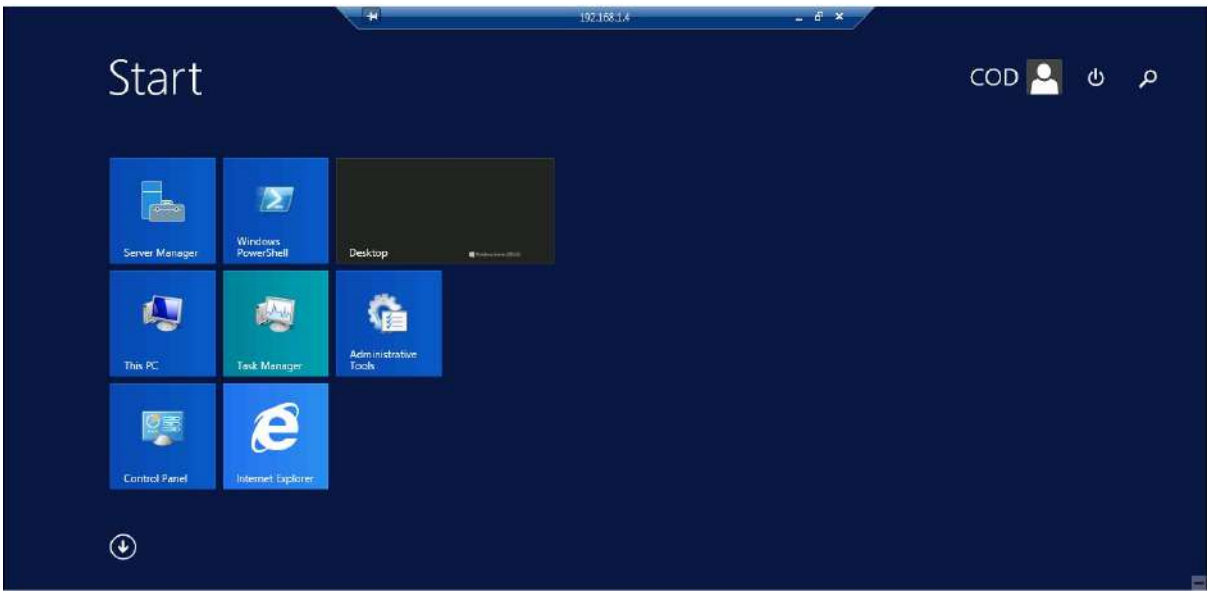
```
cod1@stark:/mnt/greyjoy/bak/cod$ chmod 774 cod1 cod2
cod1@stark:/mnt/greyjoy/bak/cod$ sudo chgrp cod cod1
cod1@stark:/mnt/greyjoy/bak/cod$ sudo chgrp cod cod2
cod1@stark:/mnt/greyjoy/bak/cod$ ls -l
total 8
drwxrwxr-- 2 cod1 cod 4096 feb  4 19:20 cod1
drwxrwxr-- 2 cod1 cod 4096 feb  4 19:20 cod2
cod1@stark:/mnt/greyjoy/bak/cod$
```

Modificamos el fichero /etc/hosts para poner nombre a cada una de las IP´s de los servidores y Raspberrys.

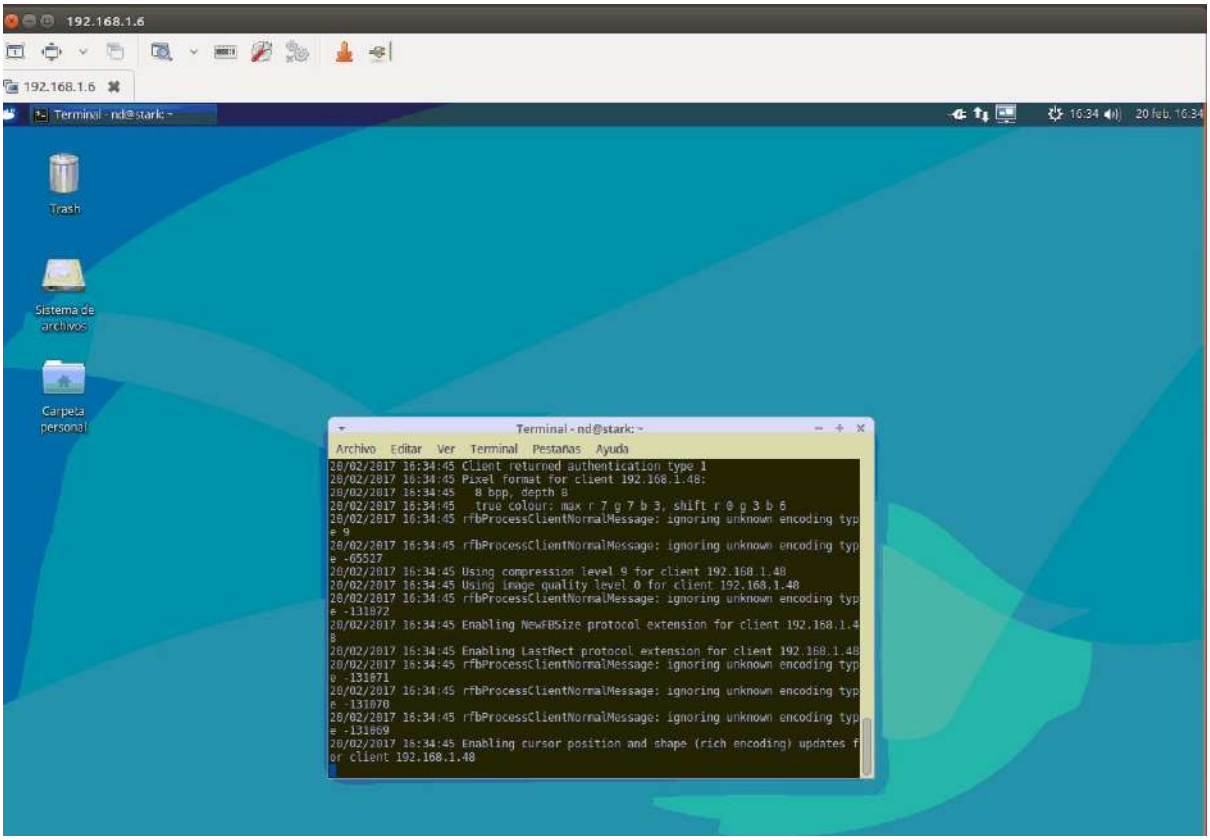

## *Conexiones remotas*

Para acceder de manera remota a los servidores utilizaremos conexión a escritorio remoto de Windows. Entraremos a través de los puntos de acceso Wi-Fi, utilizando una IP fija previamente asignada a los miembros del grupo COD e introduciendo la IP, nombre y contraseña en el programa de Conexión a Escritorio remoto.

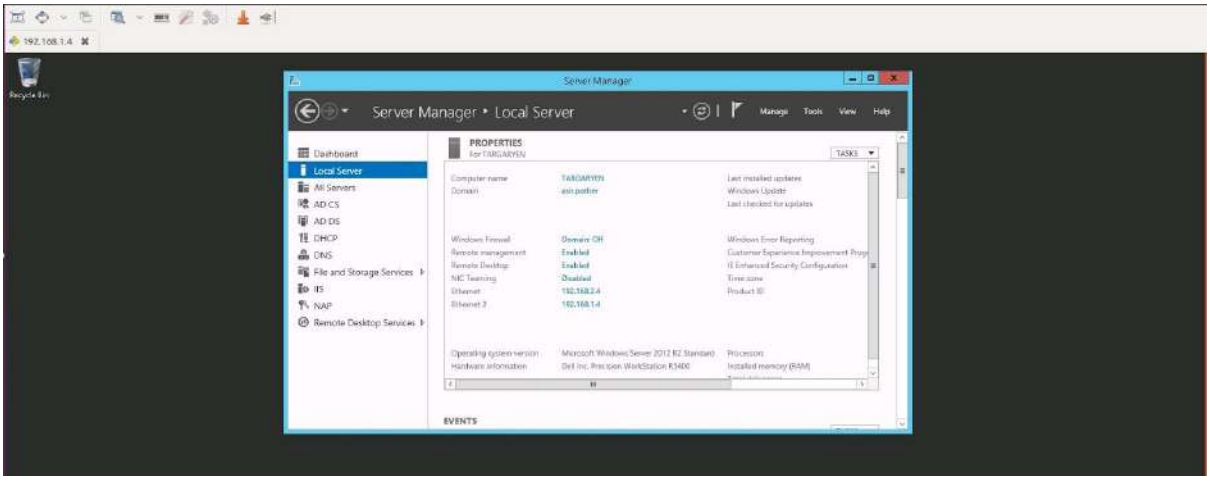

Después de esto podemos ver el escritorio (en este caso Targaryen) desde el portátil de los miembros del equipo COD.




### *Ubuntu-Ubuntu*




## *Ubuntu-Windows*

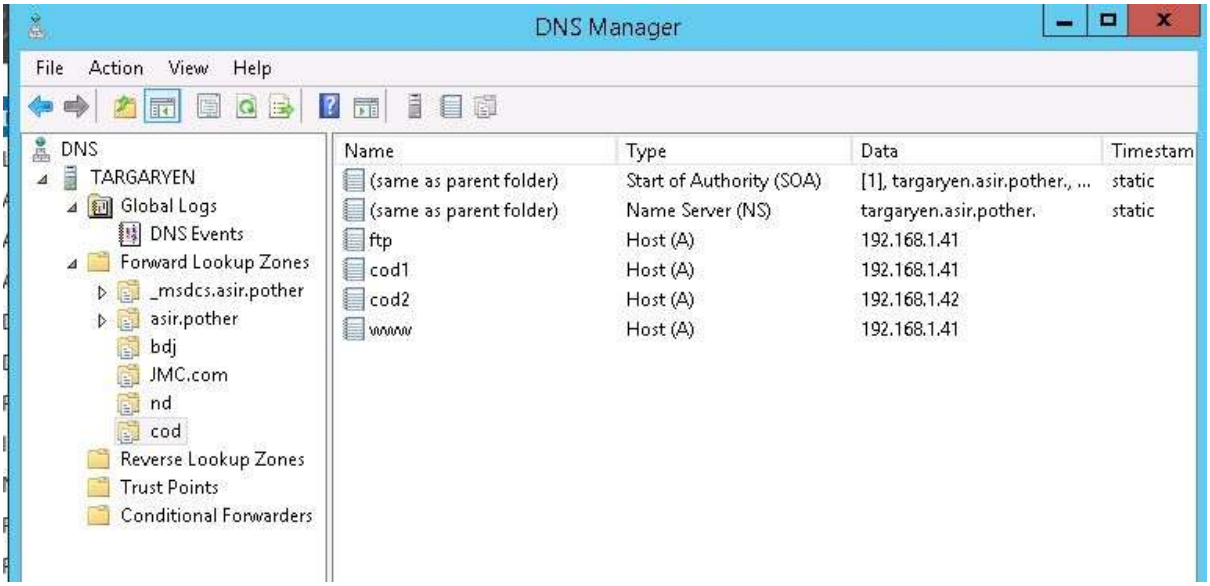



## *Administración de los servidores*

### *Active Directory*

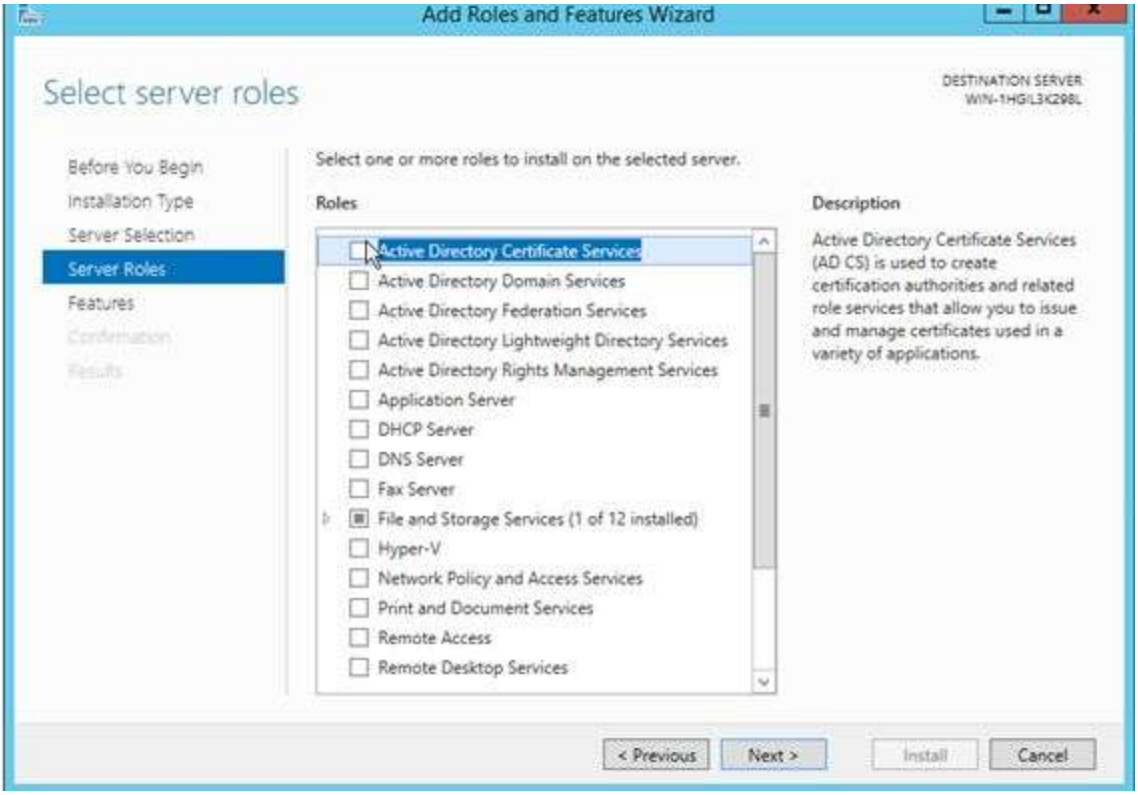
Lo primero que haremos es la instalación DNS en el servidor. Para ello entramos en Añadir roles y características y seguiremos los pasos de instalación que hemos seguido en las prácticas realizadas a lo largo del curso. Escogemos añadir DNS Server y añadimos sus características. Vemos que nos ofrece DNS y damos a next

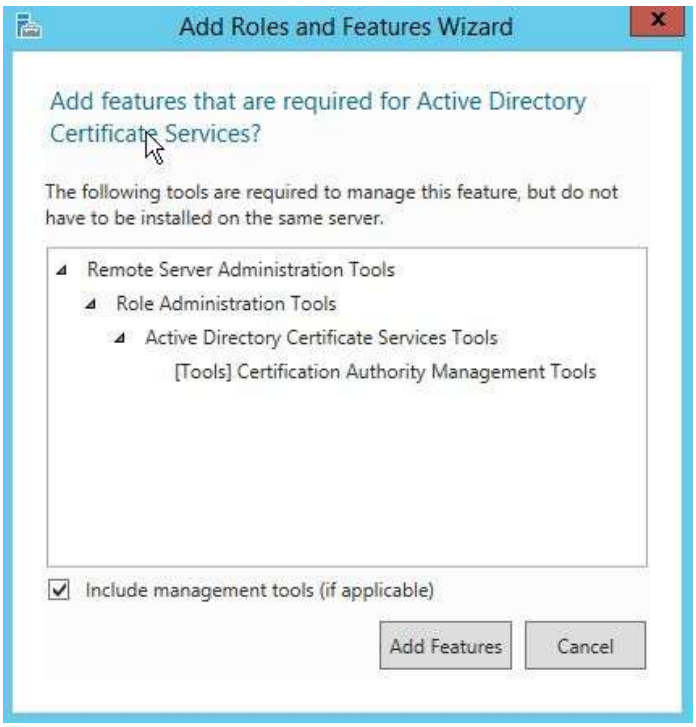
para proceder a instalar. En la siguiente imagen podemos ver el panel del DNS con los ficheros creados para la web y ftp.



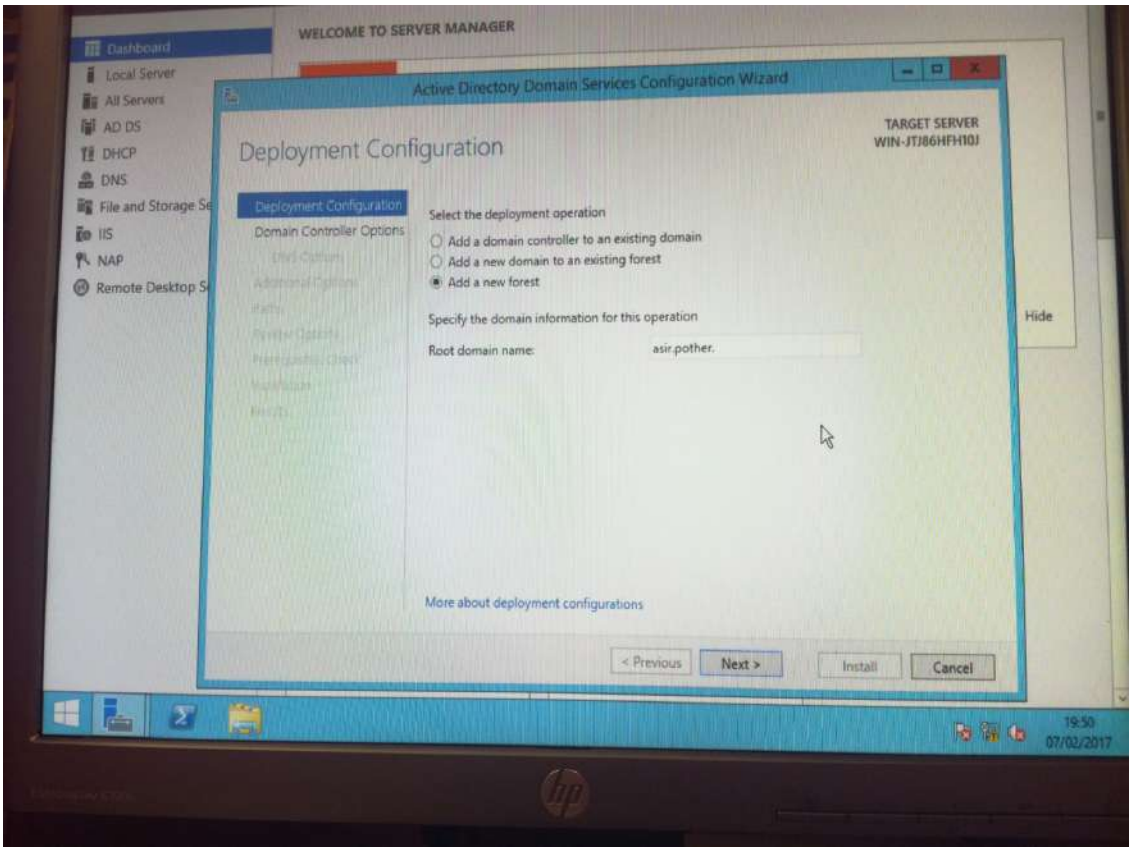


Confirmamos la instalación y ya comenzaría a instalarse el rol. Haremos lo mismo con el rol de directorio activo.

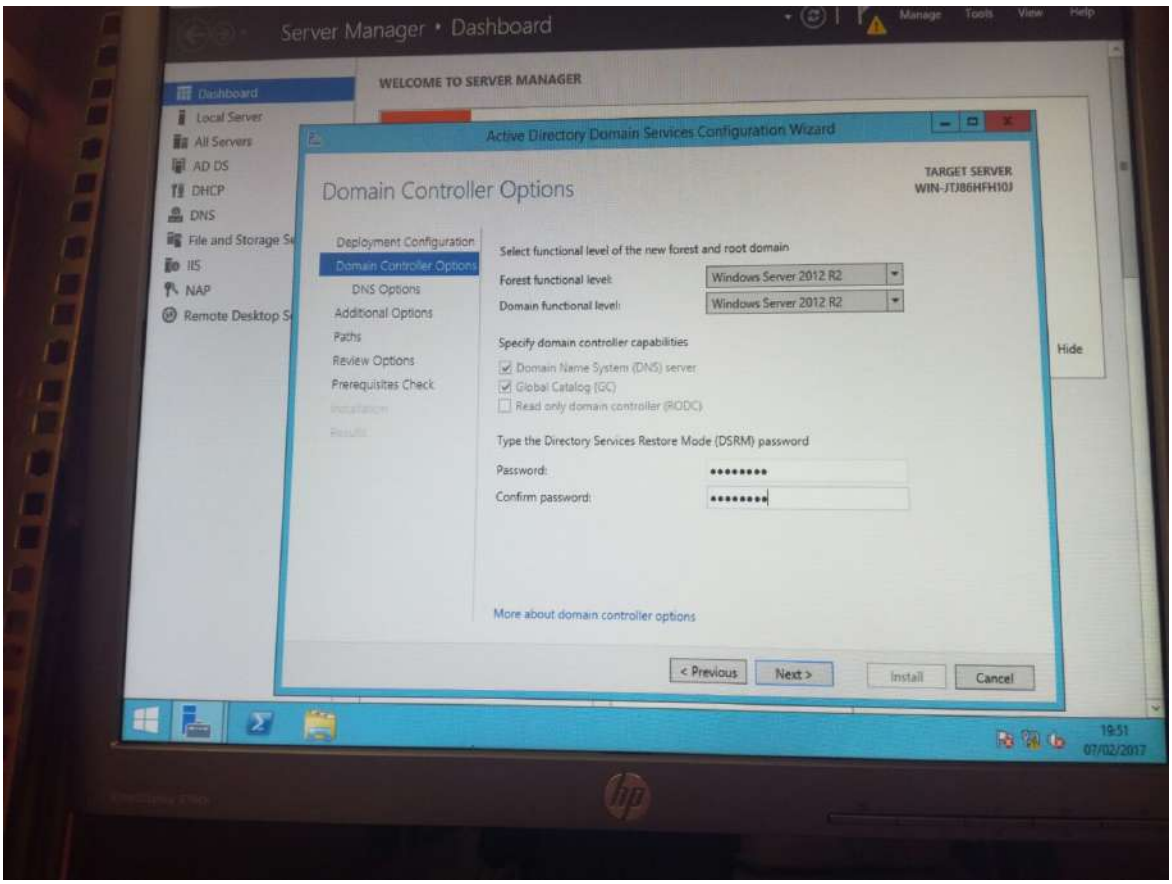




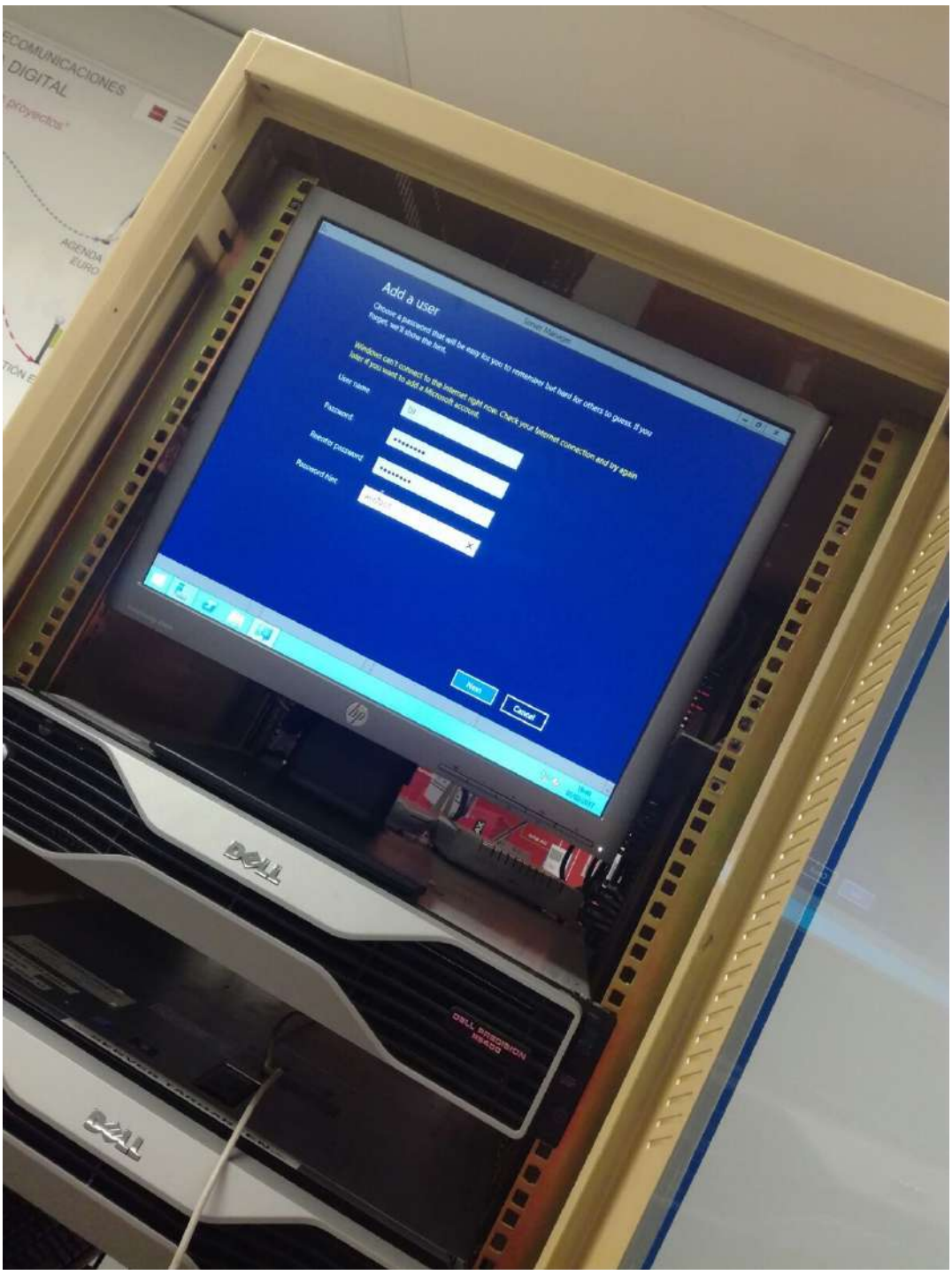

Debemos crear un nuevo bosque donde tendremos los grupos de los equipos y dentro los usuarios de los mismos. El nombre de dominio se llamará asir.pother.



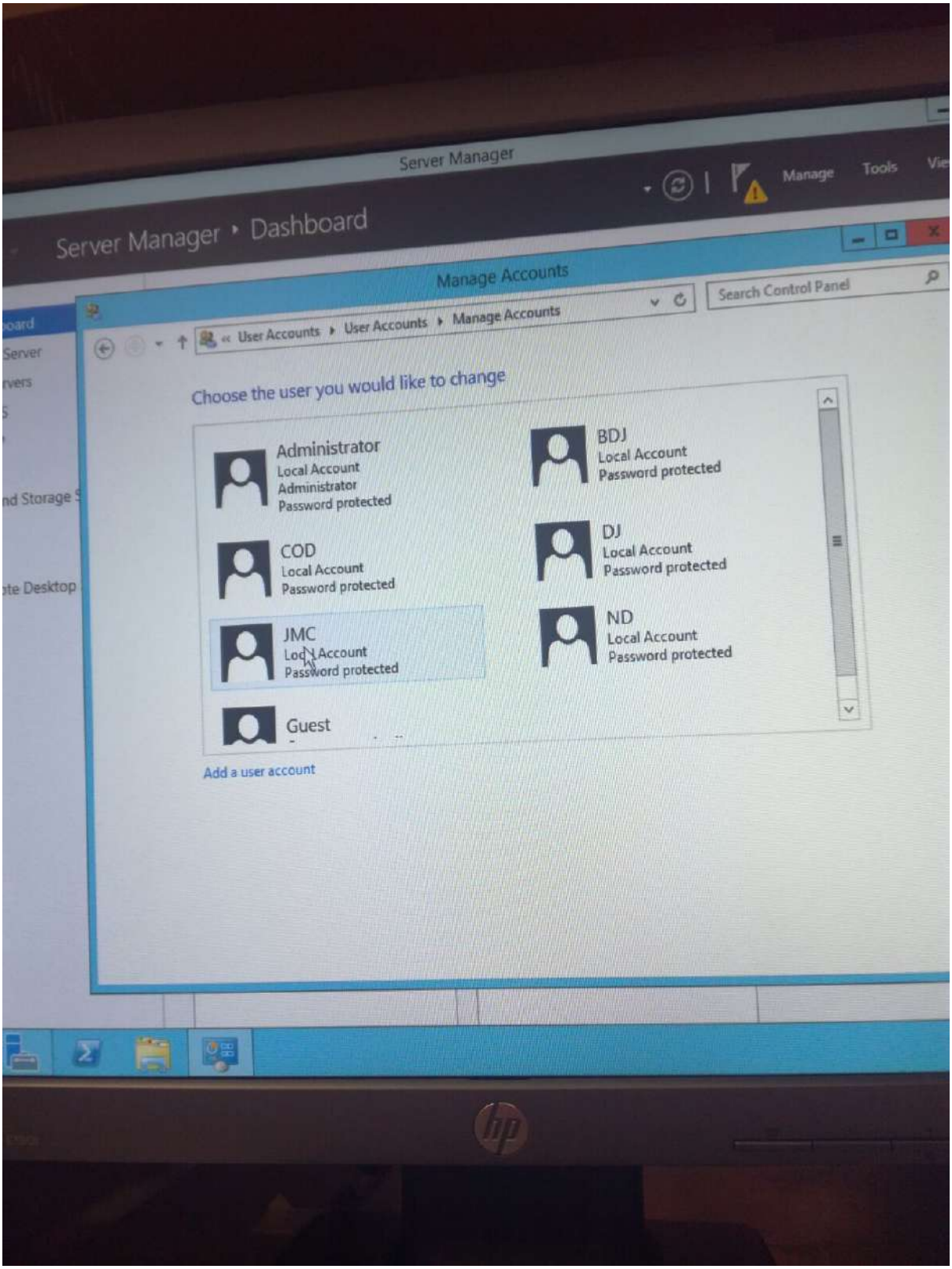

Además le pondremos una contraseña.



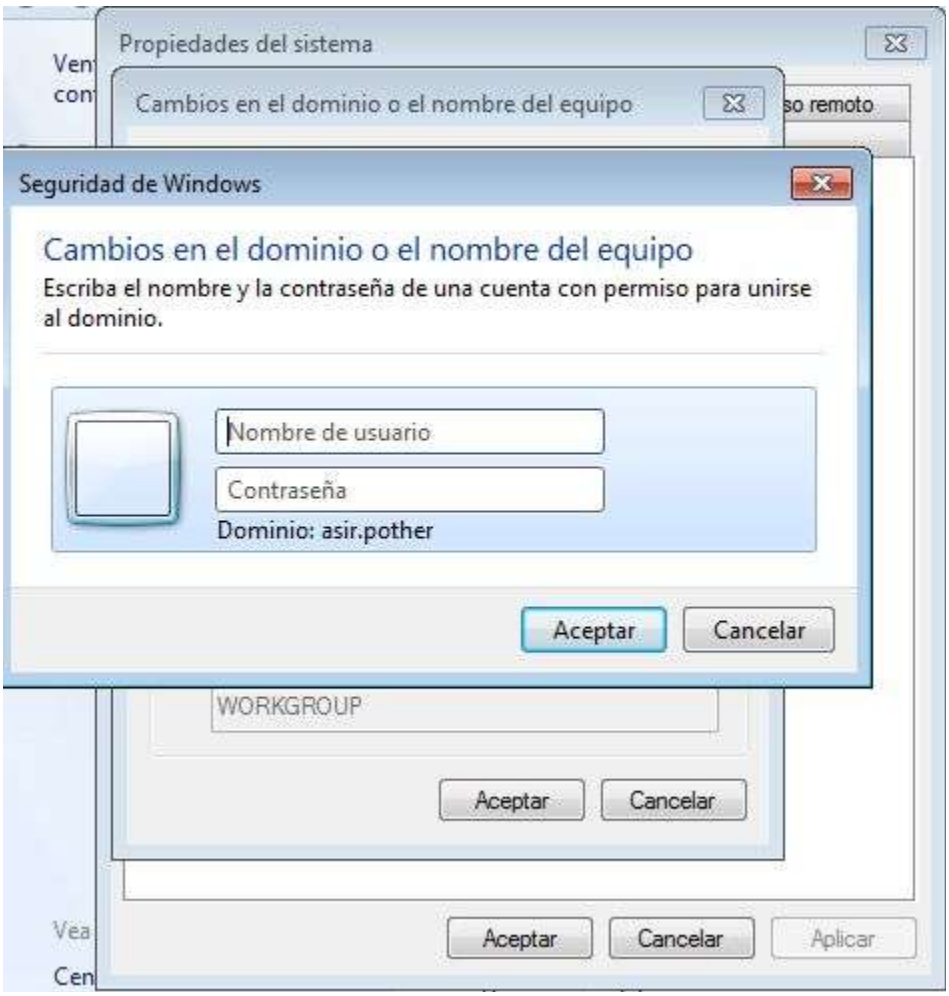

Una vez tengamos esto seleccionado podemos dar a next. La descripción que tenemos de Active Directory nos explica que tendremos unos usuarios administrados por admin (nosotros) y que les daremos una serie de permisos y accesos.



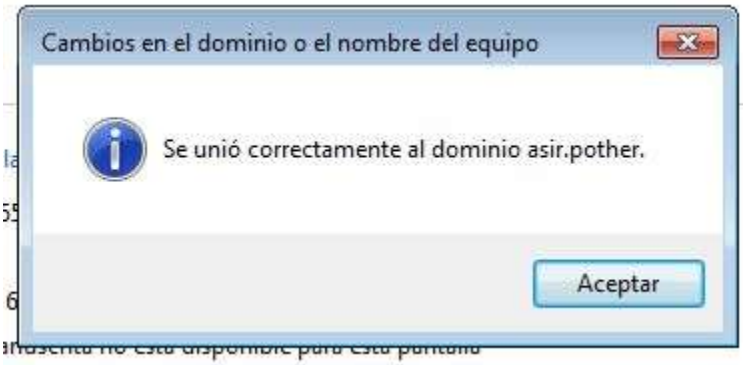

Una vez hechos estos pasos podemos ver en el Dashboard los nombres de los grupos del proyecto incluido el nuestro.



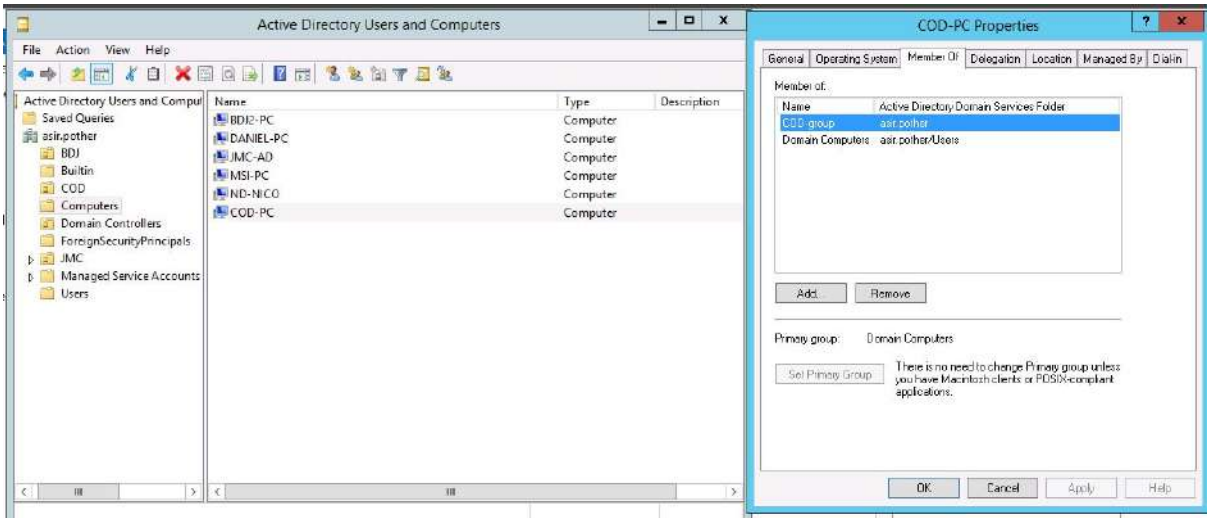

La primera vez que nos conectamos remotamente nos pedirá un usuario y contraseña del servidor. Por lo que hay que hacerlo directamente en el servidor.




Una vez realizado esto ya estaremos logueados en el dominio del proyecto.

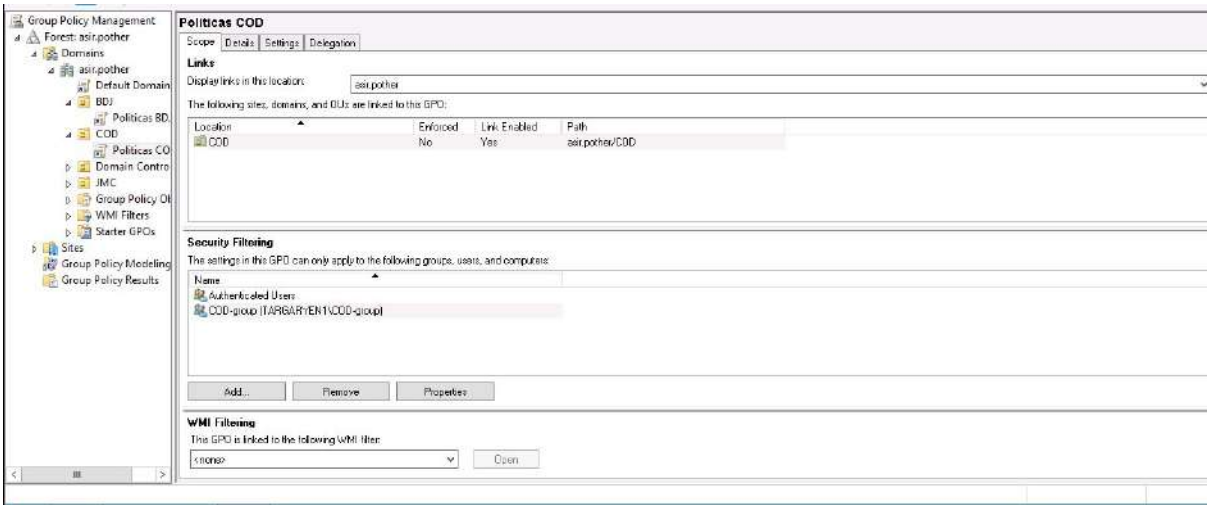



Después de tener todo esto configurado, nos conectaremos de manera remota. Desde allí podemos configurar los usuarios, los haremos miembros del grupo COD y además los podemos ver en los usuarios que están registrados en el ordenador.

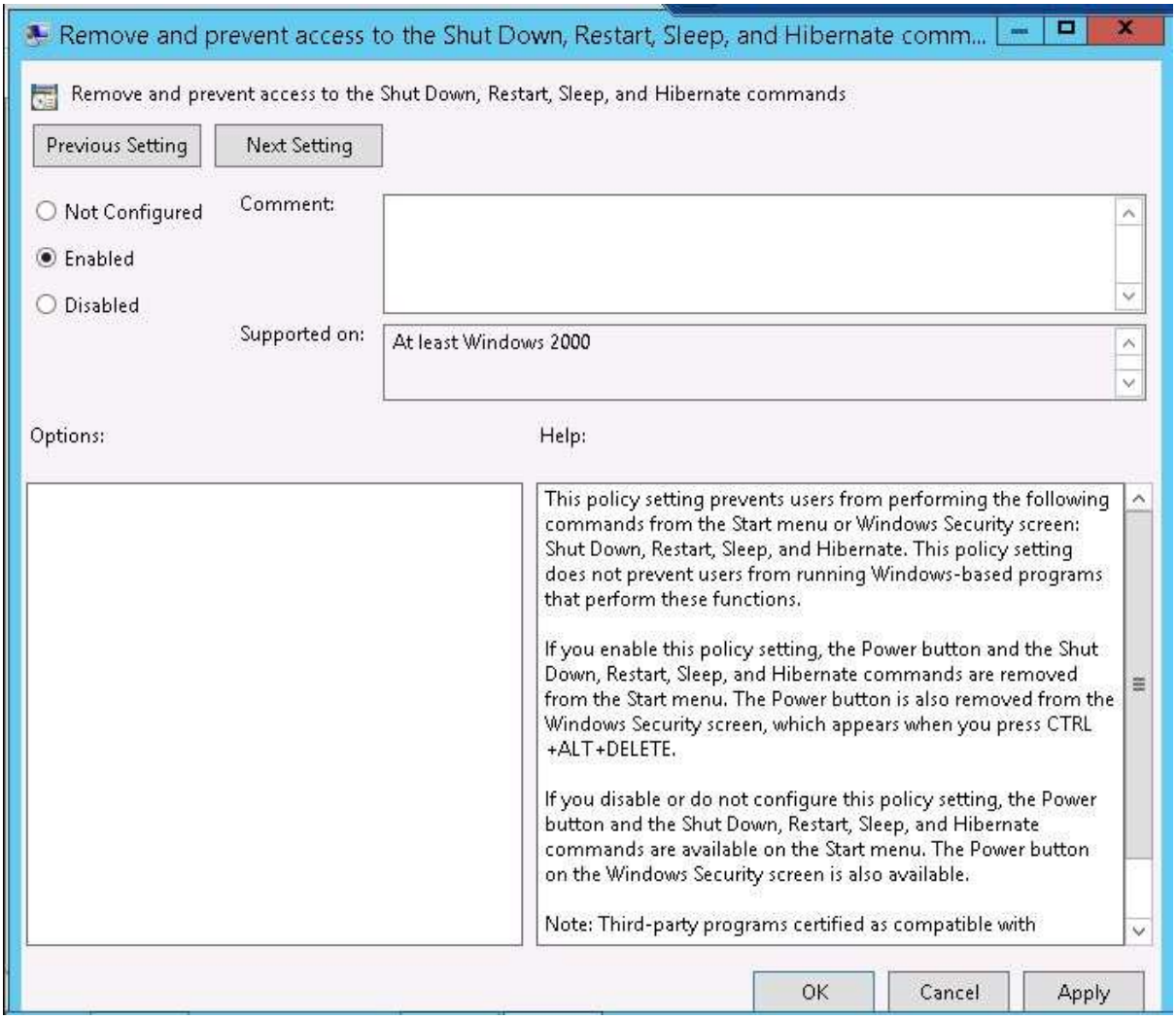



Después de tener los usuarios conectados empezaremos a crear las políticas de grupo COD.

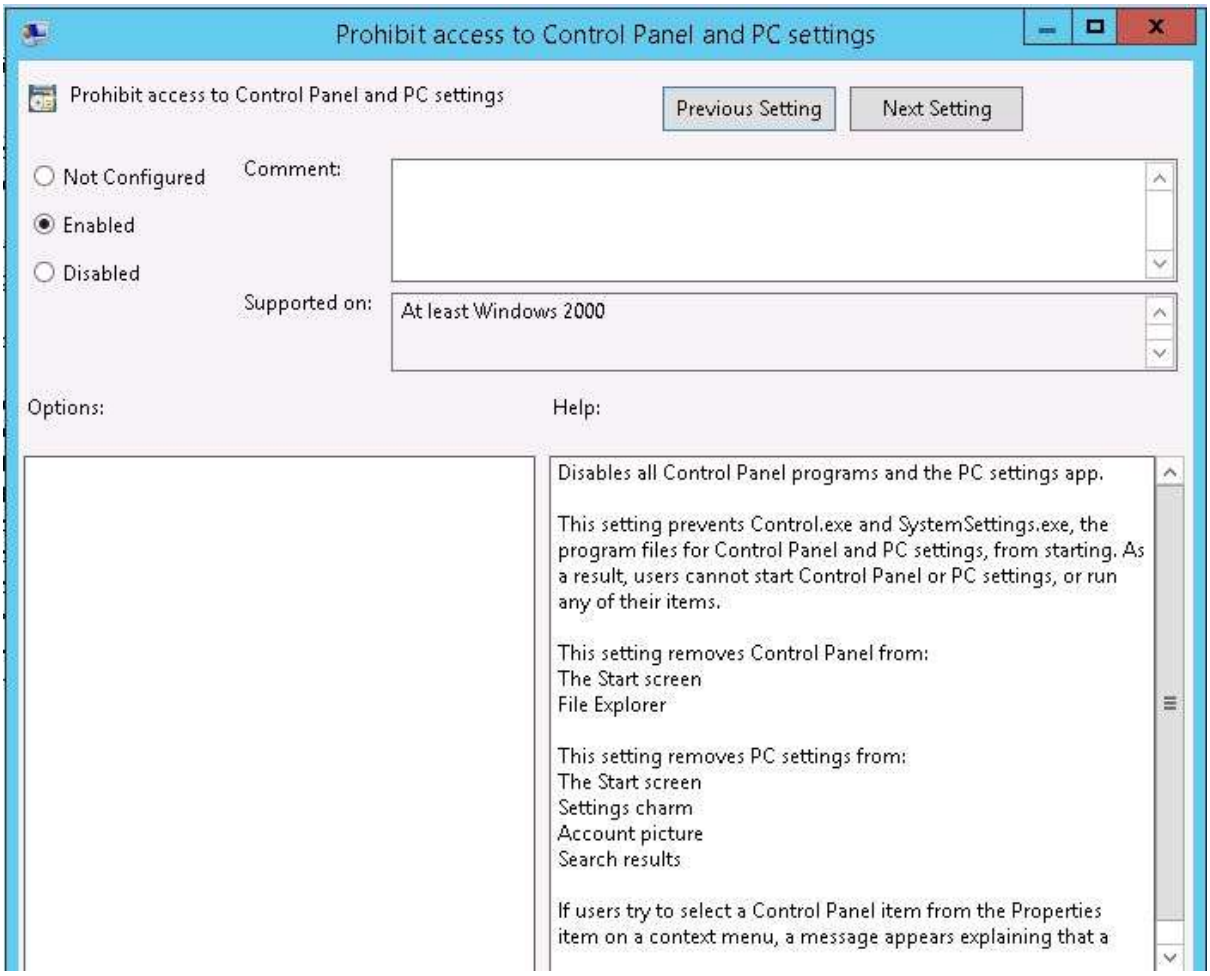
## *Creación de las GPO*



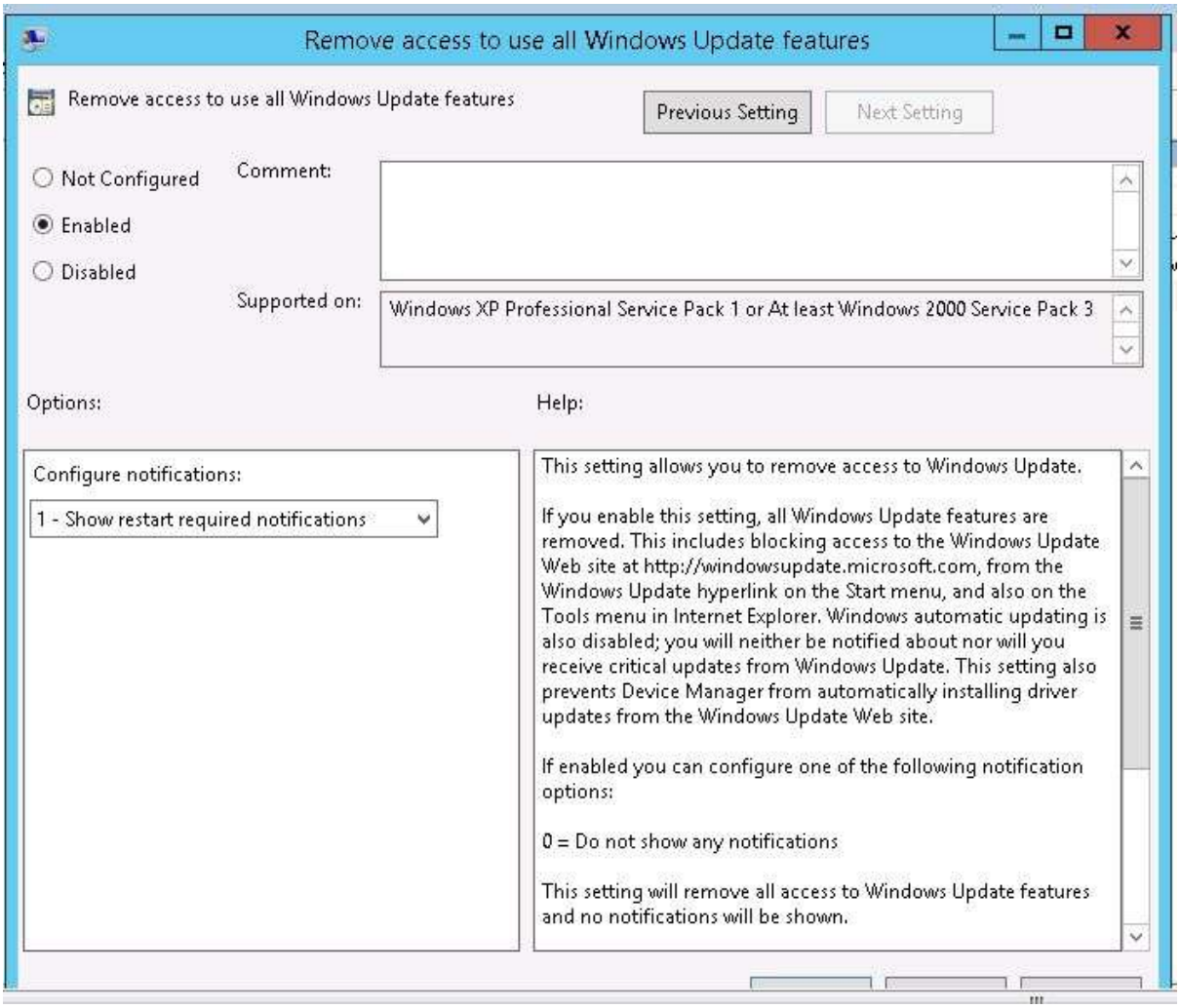

La primera política de grupo que crearemos será quitar las opciones de apagado, restart, hibernar...




La siguiente política de grupo que implementamos es la prohibición de acceso de los usuarios al panel de control.

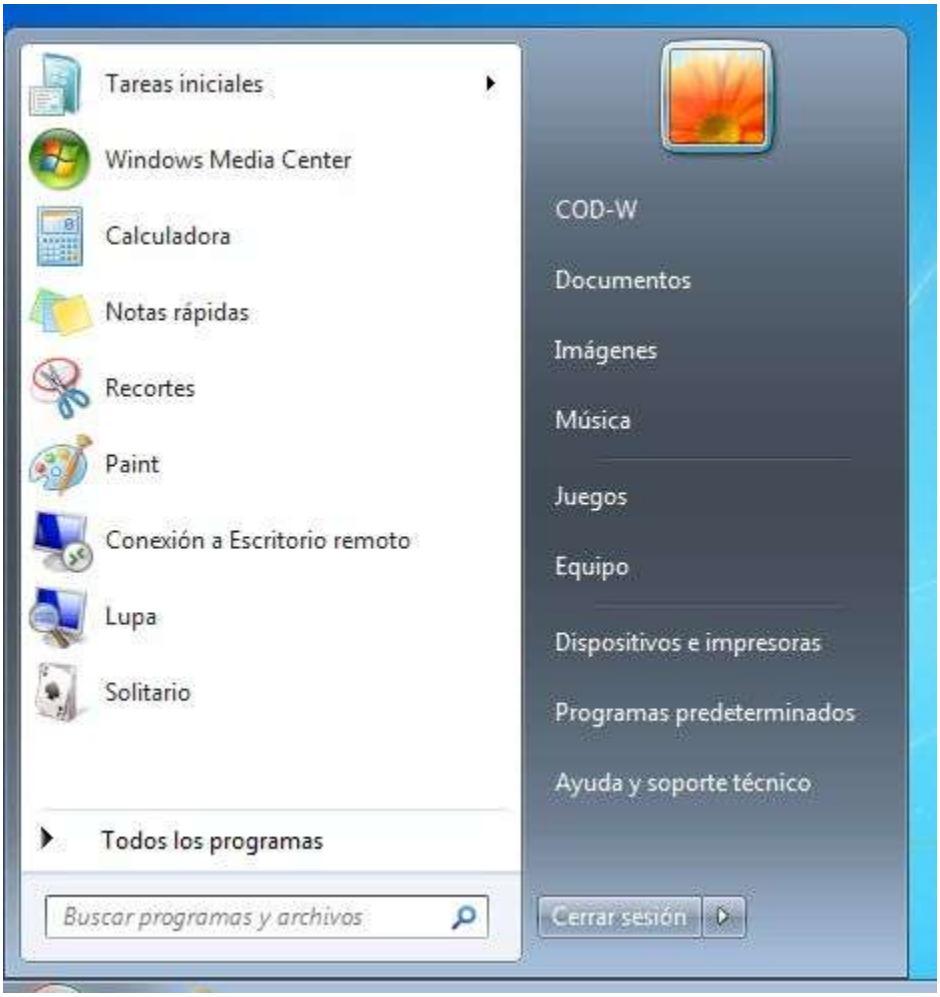


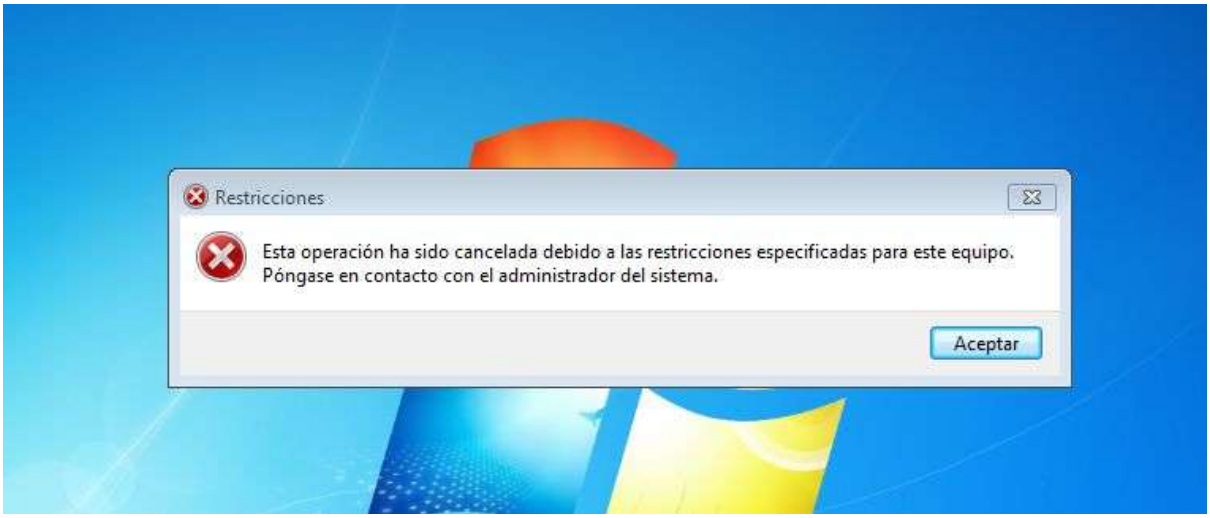

Esta política de grupo lo que hace es impedir las actualizaciones a los usuarios.




Por último podemos ver las políticas de grupo ya implantadas desde un usuario cliente que está dentro del grupo de política de COD. Las políticas que hemos implementado son:

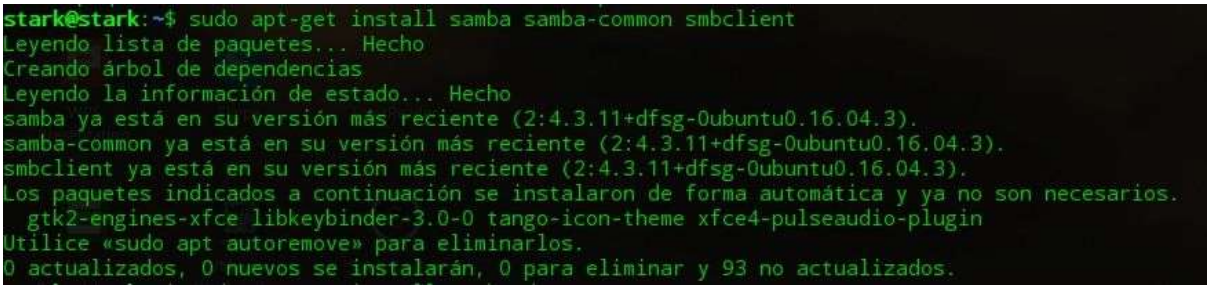
- El usuario del Directorio Activo sólo puede cerrar sesión (no tiene permisos para reiniciar, apagar o suspender).
- El usuario del Directorio Activo no puede hacer actualizaciones en el sistema.


Tareas iniciales
Windows Media Center
Calculadora
Notas rápidas
Recortes
Paint
Conexión a Escritorio remoto
Lupa
Solitario
Todos los programas
Buscar programas y archivos
COD-W
Documentos
Imágenes
Música
Juegos
Equipo
Dispositivos e impresoras
Programas predeterminados
Ayuda y soporte técnico
Cerrar sesión


Restricciones
Esta operación ha sido cancelada debido a las restricciones especificadas para este equipo.
Póngase en contacto con el administrador del sistema.
Aceptar

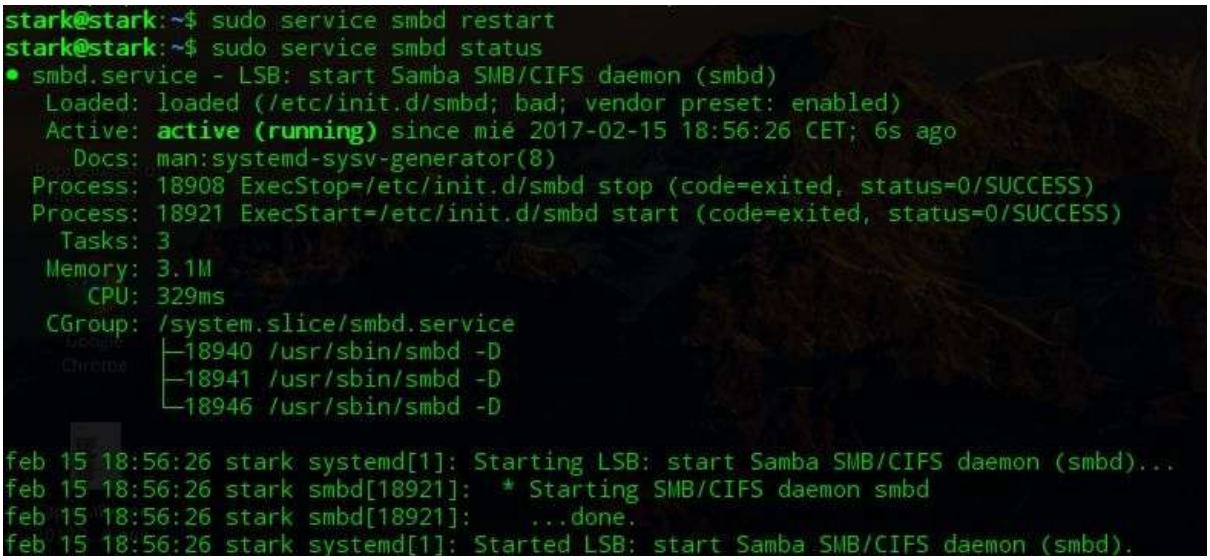
## *Samba en el servidor Stark*

En primer lugar, en UBUNTU 16.04 ya tenemos algunos paquetes SAMBA instalados (podemos comprobarlo con dpkg -l nombrepaquete) si no fuera así instalaremos: sudo apt-get install samba samba-common smbclient samba-doc samba - Servidor de archivos e impresoras samba-common - Archivos comunes de samba utilizados para clientes y servidores. smbclient - Cliente simple swat - Herramienta de administración de Samba vía web samba-doc - Documentación de Samba. smbfs - Comandos para montar y desmontar unidades de red samba system-config-samba: interfaz gráfica de configuración para samba




A continuación editamos el archivo de configuración /etc/samba/smb.conf

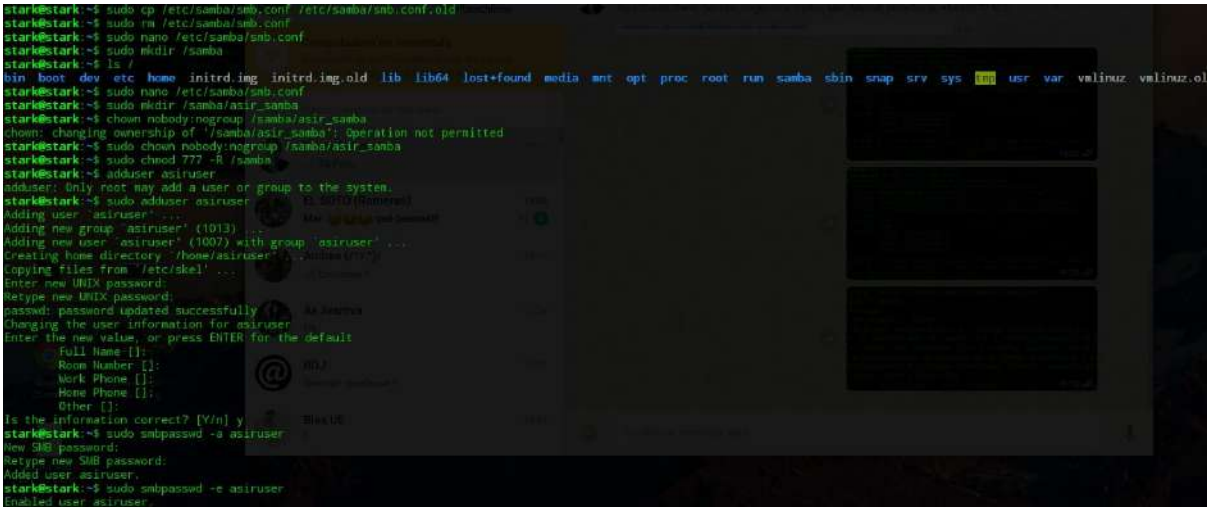

(para que los cambios surjan efecto se ha de reiniciar servicio # service smbd restart)



Necesitamos crear una carpeta en Ubuntu Server para compartir, por ejemplo, creamos

Hacemos por defecto de propiedad pública y asignamos todos los permisos en smb.conf lo restringimos ya al usuario winuser y esta configuración es predominante, se pedirán claves!)

En el sistema LINUX necesitamos convertir un usuario en usuario samba para ello primero lo creamos en el sistema: # adduser winuser Y ahora para que sea usuario samba: # smbpasswd -a winuser (facilitamos una contraseña segura para la conexión desde Windows, puede ser otra) # smbpasswd -e winuser (lo habilitamos, enable).

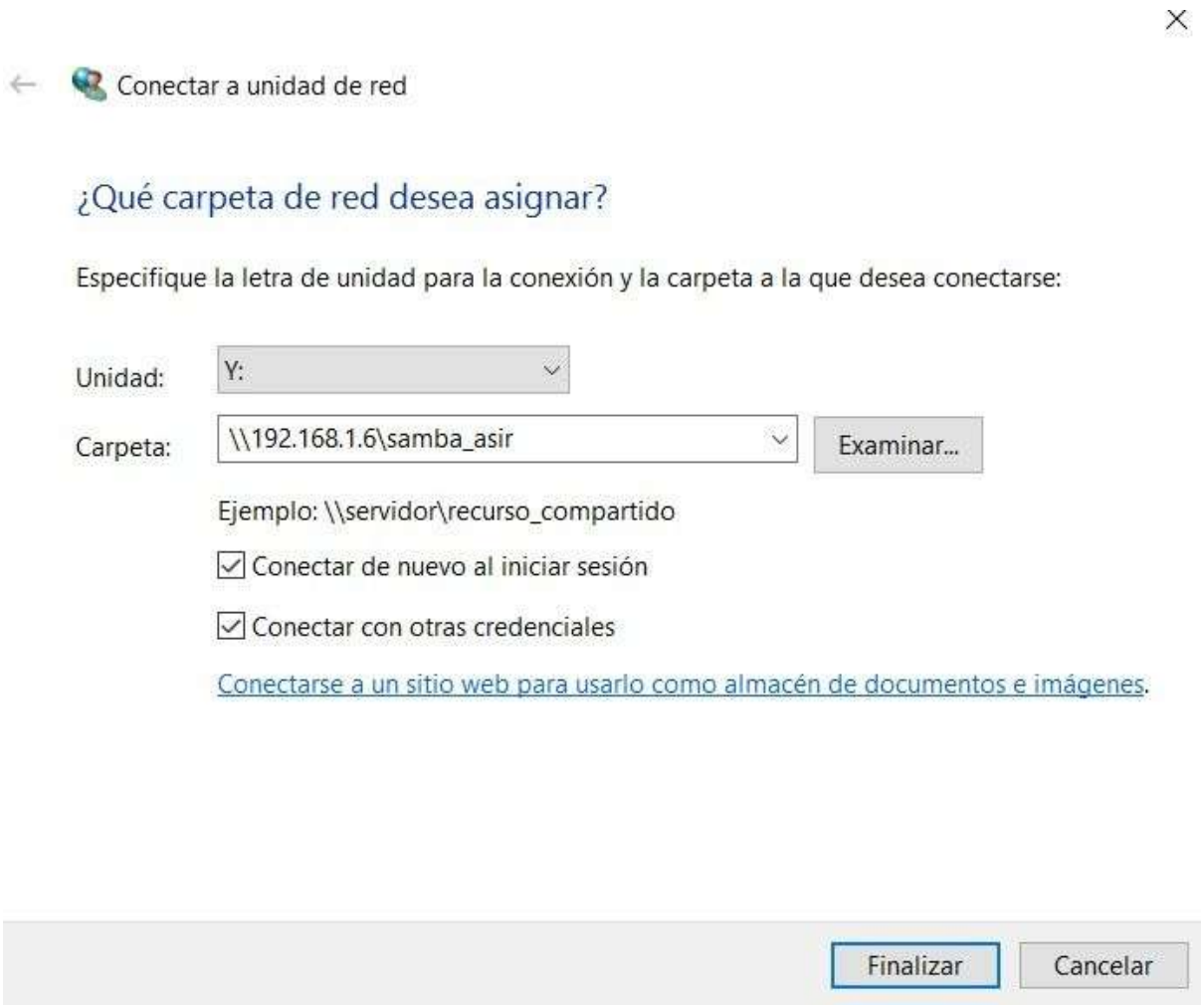



Por último hacemos un service smbd restart.

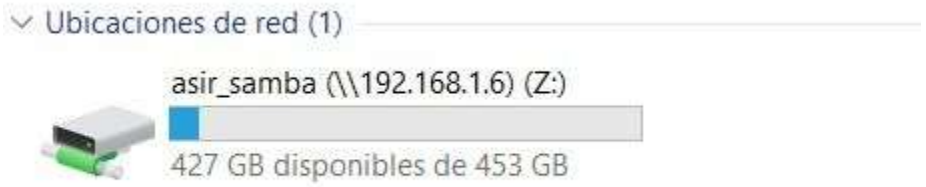
***En el cliente WINDOWS***

*Administrar Recursos Compartidos*

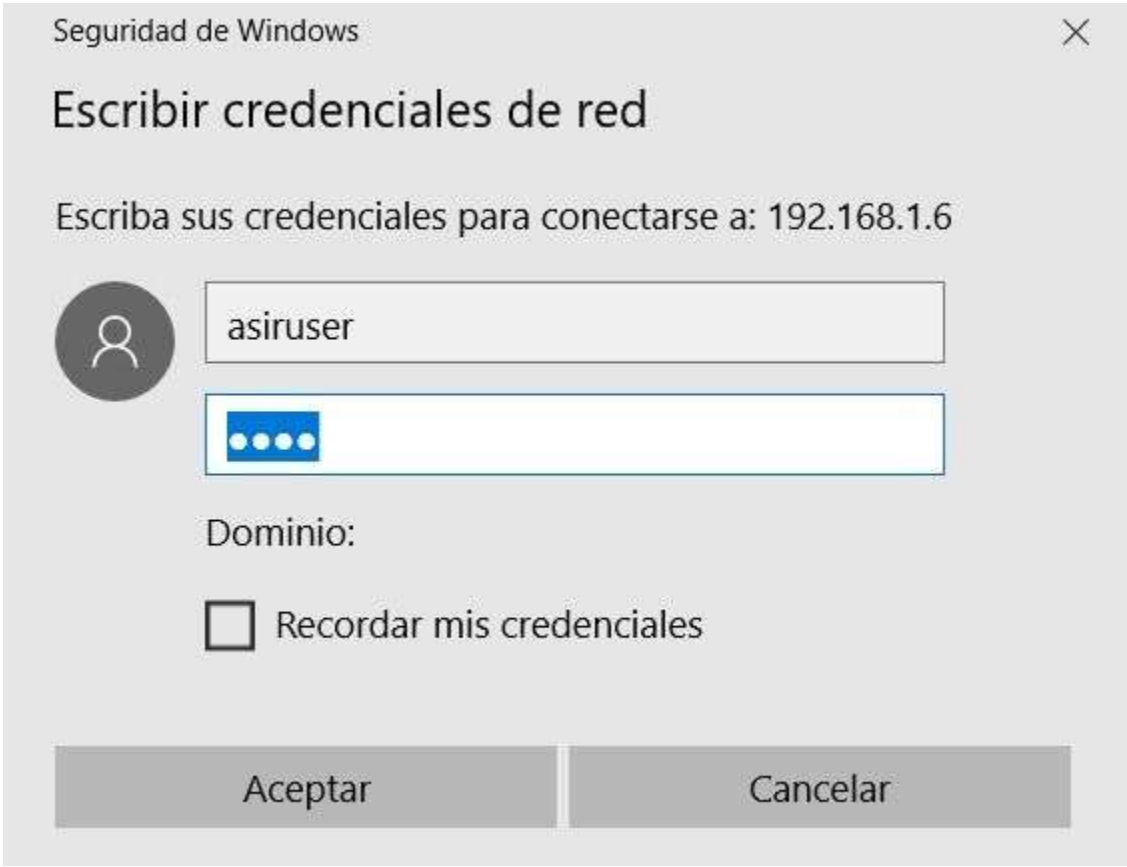
Conectar a unidad de red y facilitamos la IP o el Dominio de nuestra máquina UBUNTU en red (podemos asignarle una letra de Unidad)




Ya podemos ver el recurso compartido.



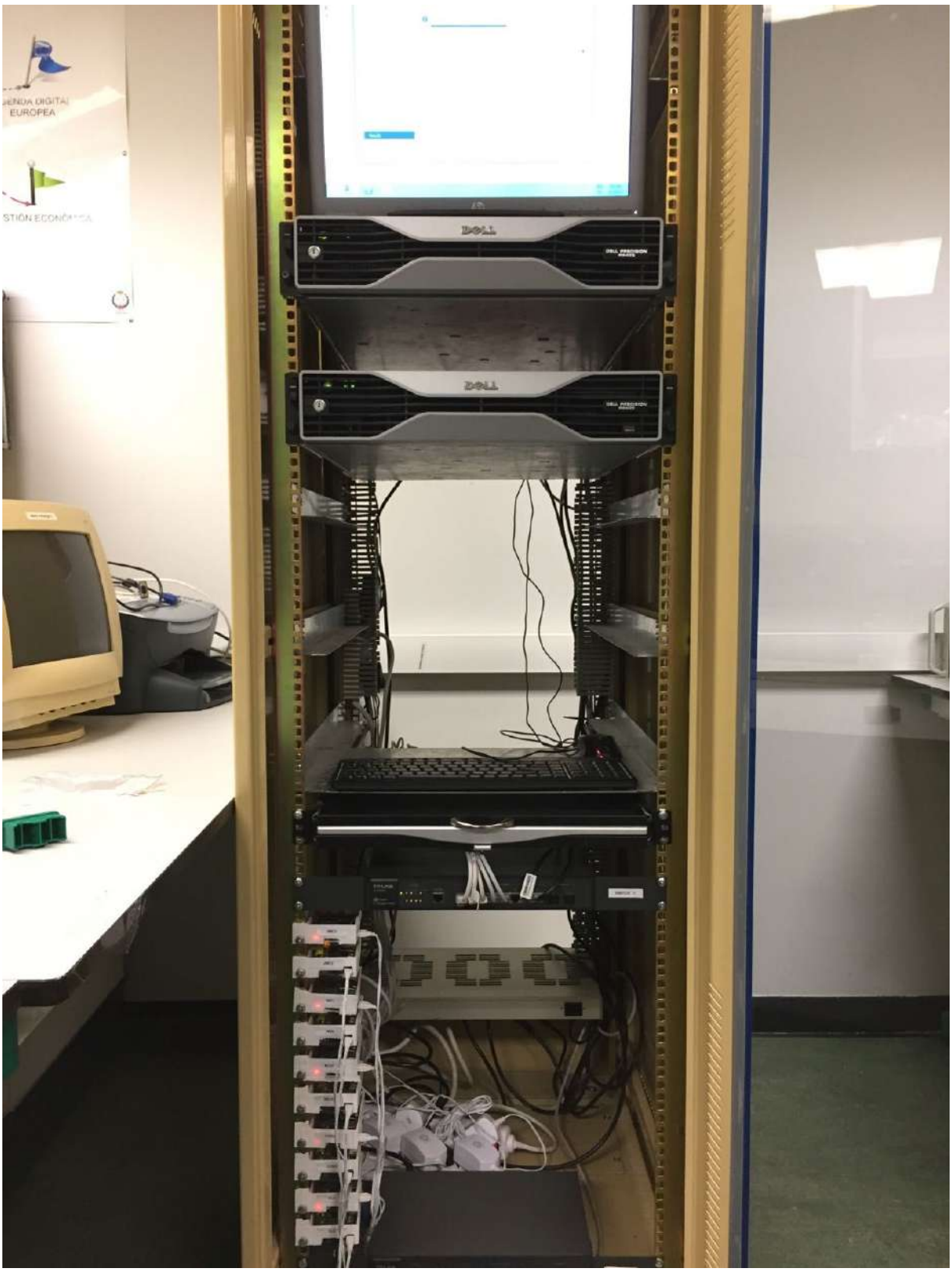

Por lo que ya podemos entrar en la carpeta compartida con nuestro usuario y contraseña.




## *Redes*

### *Montaje del Rack dentro del CPD (Centro de Proceso de Datos o aula 308)*

Lo primero que haremos será comprobar todos los elementos que tenemos. Comprobaremos que todos los cables que tenemos siguen el estándar y son cables directos que siguen la norma a ambos extremos del cable. Seguiremos la norma T-568B de cables directos utilizados para conectar routers, switches y hubs. Para realizar la comprobación utilizaremos los tester de los que disponemos en el aula.




Para el diseño y despliegue de esta red, hemos pensado en la manera más simple y sencilla. De esta manera se encontrará todo en el rack:
Tendremos el acceso de los usuarios COD a dos puntos.
El cableado se hará en el rack usando cables directos con un sistema administrativo horizontal.

***RACK***

Stark,Targaryen y Lannister(de arriba a abajo). Debajo las Raspberrys conectadas y los switches y dentro los puntos de acceso (no aparecen en la imagen).

***Unidades del rack***

Stark y Targaryen → 2U (1,75" x 2)
Lannister y dos switches → 1U (1,75" x 1)
Raspberrys → (0,625")

En el rack, a parte de los servidores, también tenemos pantalla y teclado para la configuración, además de las regletas para alimentar todos los dispositivos.

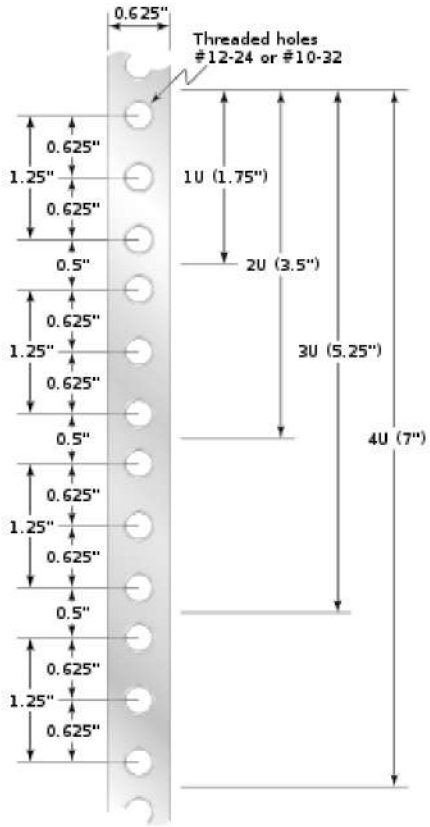
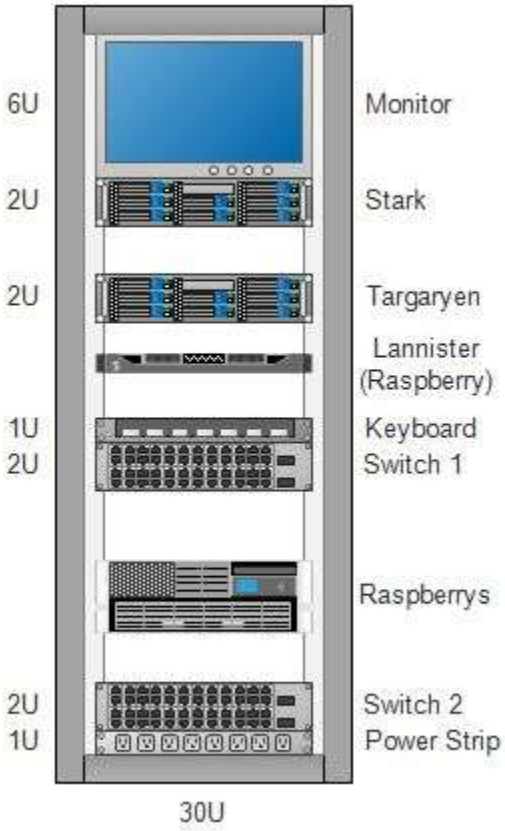



***Diagrama del Rack***

Como podemos ver tenemos todos los dispositivos en el rack, además tenemos los dos puntos de acceso que no están visibles en este diagrama.




Definimos IPs estáticas flotantes en Raspberry Lite y Pixel.

COD1 192.168.1.41

```
eth0:1    Link encap:Ethernet  HWaddr b8:27:eb:8c:45:0a
          inet addr:192.168.1.41  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

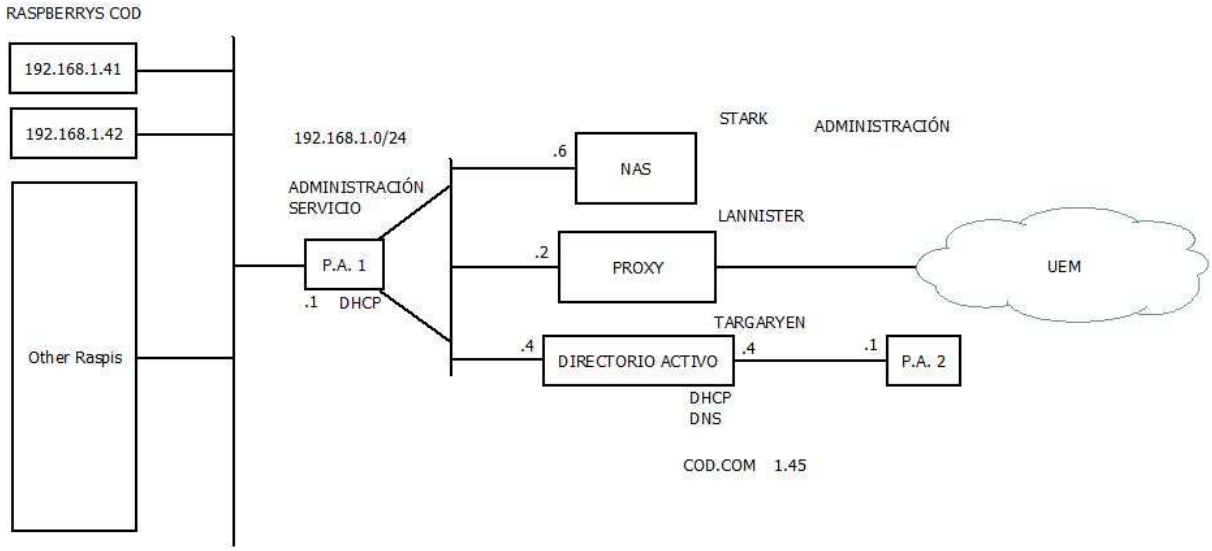
COD2 192.168.1.42

```
eth0:1    Link encap:Ethernet  HWaddr b8:27:eb:0c:f4:43
          inet addr:192.168.1.42  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

***Configuración de la red***

Elegimos los elementos que vamos a utilizar para crear la red de Pother. Una vez seleccionados creamos un esquema que nos muestre cómo debemos conectar los elementos.

# *Esquema de red*




Configuramos dos puntos de acceso uno para el frontal que dará los servicios (Raspberrys) y otro para configuración. Los puntos de acceso tendrán las direcciones de clase C 192.168.1.1 y 192.168.4.1. Desde este punto de acceso podremos entrar a configurar los servidores y las Raspberrys de manera remota. Cogemos los routers que utilizaremos y los configuramos, de tal manera que nos de una IP dentro de los rangos de nuestro equipo. Además introducimos nuestras MAC Address para que solamente tengamos acceso los miembros del grupo.

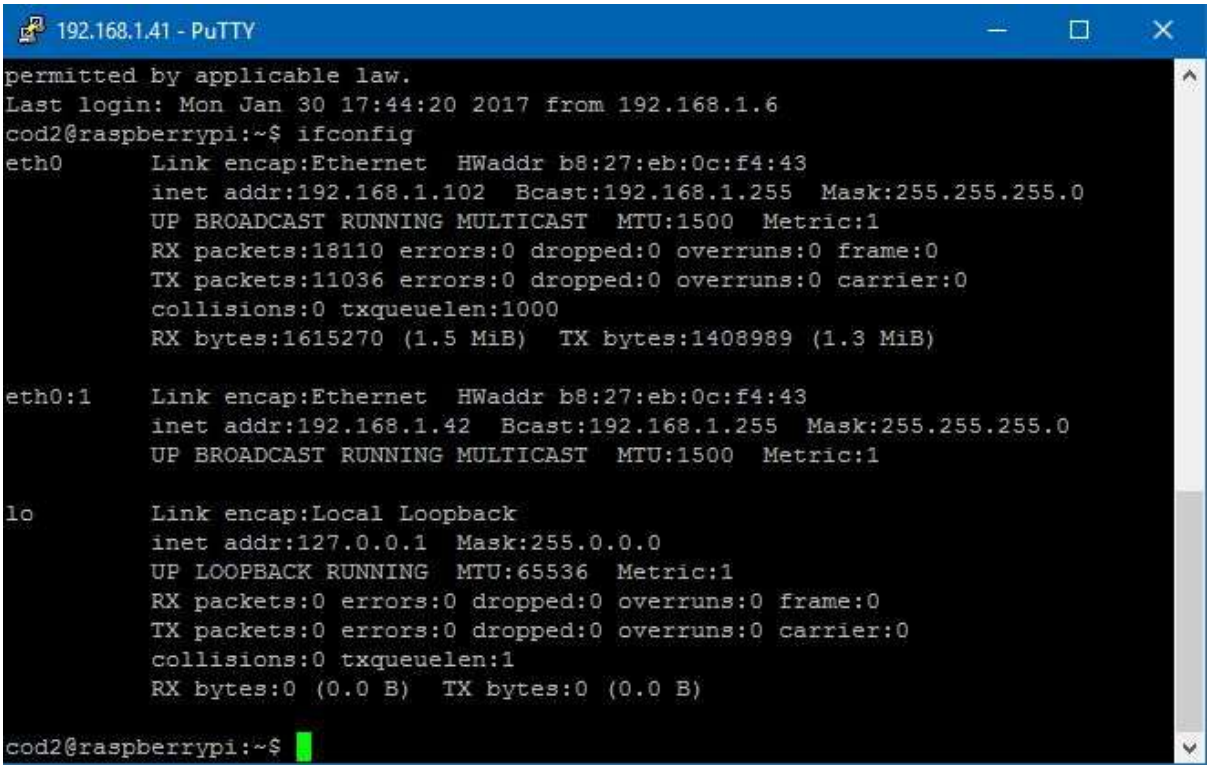
***Prueba de acceso Wi-Fi a la red***

Buscamos el SSID de los puntos de acceso que hemos creado y a los que hemos nombrado como ASIR-P1 y ASIR-P2. Previamente hemos realizado un filtrado de MAC-Adress para que sólo puedan acceder nuestros dispositivos.


En el punto de acceso ASIR-P1 la IP ha de ser estática, en el punto de acceso ASIR-P2 se asigna por DHCP.


***Configurar en LAN con la IP 192.168.1.6 para el NAS***

Desde nuestros equipos personales podremos entrar con Putty estableciendo una IP estática en nuestra tarjeta de red.




## *Instalación de servicios*

Instalación de Apache2

```
login as: cod1
cod1@192.168.1.41's password:

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb  6 17:15:36 2017 from 192.168.1.45
cod1@raspberrypi:~$ sudo apt-get install apache2
[sudo] password for cod1:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine apache2-suexec-custom openssl-blacklist
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 124 no actualizados.
```

El fichero de apache se almacena en /etc/apache2

```
cod1@raspberrypi:/etc/apache2$ ls -lat
total 88
drwxr-xr-x 111 root root  4096 feb  7 18:56 ..
drwxr-xr-x   2 root root  4096 feb  7 18:52 sites-enabled
drwxr-xr-x   2 root root  4096 feb  7 18:52 conf-enabled
drwxr-xr-x   2 root root  4096 feb  7 18:51 mods-enabled
drwxr-xr-x   2 root root  4096 feb  7 18:51 sites-available
drwxr-xr-x   8 root root  4096 feb  7 18:51 .
drwxr-xr-x   2 root root 12288 feb  7 18:51 mods-available
drwxr-xr-x   2 root root  4096 feb  7 18:51 conf-available
-rw-r--r--   1 root root  7115 ago  7  2016 apache2.conf
-rw-r--r--   1 root root  1782 jul  5  2016 envvars
-rw-r--r--   1 root root 31063 jul  5  2016 magic
-rw-r--r--   1 root root   320 jul  5  2016 ports.conf
cod1@raspberrypi:/etc/apache2$
```

Creamos los enlaces simbólicos

```
cod1@raspberrypi:/etc/apache2/mods-enabled$ sudo ln -s ../mods-available/ssl.conf
cod1@raspberrypi:/etc/apache2/mods-enabled$ sudo ln -s ../mods-available/ssl.load
cod1@raspberrypi:/etc/apache2/mods-enabled$
```

Activamos el puerto 8080 para escuchar a través de él.

```
 GNU nano 2.2.6                Fichero: ports.conf                          Modifi

# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 80
Listen 8080
<IfModule ssl_module>
        Listen 443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Modificamos los ficheros de sites-enabled

```
 GNU nano 2.2.6                        Fichero: cod.es.conf

<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        ServerName www.cod.es

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/es
        DirectoryIndex index.php

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

```
 GNU nano 2.2.6                    Fichero: cod.com.conf

<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        ServerName www.cod.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/com
        DirectoryIndex index.php

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
</VirtualHost>
```

Creamos un index dentro de la carpeta cod para que sea la página a la que accedemos. Tendrá dos dominios .com y .es y dentro la página principal o index.

```
/var/www/html/
├── com
│   ├── alta.php
│   ├── baja2.php
│   ├── baja.php
│   ├── conexion.php
│   ├── desconexion.php
│   ├── index.html
│   ├── index.php
│   ├── listado.php
│   ├── login.php
│   ├── modificar2.php
│   ├── modificar.php
│   ├── procesaLogin.php
│   └── registroUsuarios.php
└── es
    ├── alta.php
    ├── baja2.php
    ├── baja.php
    ├── conexion.php
    ├── conexion.php.bueno
    ├── conexion.php.old
    ├── desconexion.php
    ├── index.html.old
    ├── index.php
    ├── listado.php
    ├── login.php
    ├── modificar2.php
    ├── modificar.php
    ├── procesaLogin.php
    └── registroUsuarios.php
```

En /etc/apache2/sites-enabled se crean enlaces simbólicos de los ficheros de configuración creados en sites-avaliable ejecutando desde sites-enabled “sudo ln -s ../sites-available/cod.es.conf cod.es.conf” y lo mismo con el .com.

```
lrwxrwxrwx 1 root root 35 feb 15 18:07 000-default.conf -> ../sites-available/000-default.conf
lrwxrwxrwx 1 root root 31 mar 10 17:22 cod.com.conf -> ../sites-available/cod.com.conf
lrwxrwxrwx 1 root root 30 mar 10 17:15 cod.es.conf -> ../sites-available/cod.es.conf
```

Reiniciamos el servicio Apache2 restart

```
cod1@raspberrypi:/etc/apache2$ sudo service apache2 start
cod1@raspberrypi:/etc/apache2$
```

## *Instalación de vsftpd*

```
cod1@raspberrypi:~$ sudo apt-get install vsftpd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  dialog
Se instalarán los siguientes paquetes NUEVOS:
  dialog vsftpd
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 124 no actualizados.
Se necesita descargar 373 kB de archivos.
Se utilizarán 1.001 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Configuración de vsftpd.conf
-anonymous_enable=NO
-local_enable=YES
-write_enable=YES
-chroot_local_user=YES
-allow_writeable_chroot=YES

Lo único que hemos modificado para su correcto funcionamiento son estas 5 características de la configuración. Hecho esto ya funciona el ftp correctamente.

## *Seguridad en el proyecto*

### *Explicación ¿por qué es importante la seguridad en el aula del proyecto?*

La seguridad en un aula donde tendremos los equipos del proyecto es muy importante ya que la entrada de alumnos y profesores de diferentes ámbitos pueden afectar a la misma (puede suceder que entre gente no autorizada). Debido a esto cada usuario tiene una forma de tratar los equipos, diferentes cuentas y formas de acceso a los equipos. Además puede haber material sensible como software, aplicaciones, bbdd del proyecto y se debe cuidar la E/S a Internet. Todo esto es lo más habitual en el aula 308 y puede haber otro tipo de riesgos que veremos.

***Seguridad física***

- Control de acceso y seguridad de los edificios.

- Control de acceso al aula 308 donde trabajamos: Personas y equipos autorizados tendrán llave para entrar a trabajar en la clase.
- Extintor anti incendios homologado, señales luminosas y acústicas, manguera y activación de incendios

- Seguridad de equipos y periféricos que utilizaremos en el proyecto. Las Raspberrys, cables y demás hardware a utilizar están guardados con llave en los armarios.
- Seguridad de la red interna, conexiones y cableado (switches, patch panel, regletas, cables Ethernet, etc.).
- Paneles indicativos y de emergencia.




- Doble filtro para controlar la electricidad del aula con dos cuadros.
- Implementación del SAI (finalmente no aplicado en este proyecto).
- Copias de Seguridad en almacenamiento o soporte físico con los servidores (uso de NAS, Almacenamiento Conectado en Red). Volcamos las copias de seguridad en un periodo de tiempo al servidor.

***Monitorización HW de los servidores que utilizamos***

***Planificación de la Seguridad lógica***

- NAS (Ubuntu Server 16.04. Almacenamiento de copias de seguridad)
    - Contraseña de login de nuestro equipo
    - Actualizaciones automáticas
    - Comprobar la integridad del sistema con rkhunter
    - Configuraciones DNS, IP.
    - Comprobación de la disponibilidad del sistema (comprobación de puertos, escuchas... con nmap).
    - Cifrar copias de seguridad y guardarlas
    - Auditoría de seguridad
- Servidor proxy (Ubuntu 16.04) + Firewall → squid 3
    - Contraseña de login de nuestro equipo
    - Actualizaciones automáticas
    - Comprobar la integridad del sistema con rkhunter
    - Configuraciones DNS, IP

        - Comprobación de la disponibilidad del sistema (comprobación de puertos, escuchas... con nmap).
        - Auditoría de seguridad
            - Firewall: IPTABLES o UFW
- Active Directory (Windows Server 2012, Dominio) → sfc
    - Políticas y directivas de seguridad
    - Contraseña de login de nuestro equipo
    - Actualizaciones automáticas
    - Comprobar la integridad del sistema con rkhunter
    - Configuraciones DNS, IP
    - Comprobación de la disponibilidad del sistema (comprobación de puertos, escuchas... con nmap).
    - Auditoría de seguridad
- Raspberry (Pixel y Lite) + Servidor web
    - Seguridad Web
    - Certificado SSL
    - Apache 2
    - Contraseña de login en BBDD (mysql)
    - Permisos FTP (no permitir anónimos)
    - Actualizaciones automáticas
    - Comprobar la integridad del sistema con rkhunter
    - Configuraciones DNS, IP
    - Comprobación de la disponibilidad del sistema (comprobación de puertos, escuchas... con nmap).
    - Auditoría de seguridad

- ***Seguridad con las contraseñas y datos del equipo COD***
    - Una de esas técnicas que podemos utilizar es el uso adecuado de contraseñas que estén compuestas por números, mayúsculas, minúsculas y símbolos.
    - Mínimo 8 caracteres
    - El uso de software de seguridad informática como firewall, proxy…
    - Encriptación de los datos.

***Problemas encontrados***

Se podía acceder a las Raspberrys debido a que el usuario que viene por defecto “pi” no fue eliminado y cualquiera que conociera o usara Raspberrya el software podía conectarse por Wi-fi, probar las contraseñas y acceder a ellas fácilmente.

- ***Normas para los componentes de COD en el proyecto integrador***

- Respetar normas en todo momento (no permitir comer, beber o fumar en el aula…).


- Limpieza de los equipos (paños y productos adecuados).
- Sólo acceso de personal autorizado en el proyecto.
- Evitar entrar a páginas de internet poco seguras (el centro puede tener un proxy o utilizar uno externo).
- De cara al usuario de los equipos: Prevención de riesgos laborales (Posición correcta, protector de pantalla, cuidar que los cables estén en buen estado).

***Plan de Contingencias de copia de seguridad***

*Objetivos del Plan de Contingencias*

Los objetivos principales de un Plan de Contingencias son los de planificar y describir la capacidad para respuestas rápidas. Su finalidad es la de permitir el funcionamiento del sistema Pother implementado, aun cuando alguna de sus funciones deje de hacerlo por culpa de algún incidente tanto interno como ajeno a la organización. Los principales objetivos específicos son:

- Establecer un procedimiento formal y por escrito que indique las acciones a seguir frente a determinados riesgos.
- Optimizar el uso de recursos humanos y materiales.
- Un control adecuado para cumplir con las normas y procedimientos establecidos.

Las copias de seguridad de las Raspberrys se harán mediante un script en crontab a diario. *La copia de seguridad se guardará donde hayamos elegido al principio de la copia de seguridad indicado en el anterior punto como un servidor o un disco extraíble o en la red. Dicha parte se establece en las configuraciones de las Raspberrys (ver apartado configuración de las Raspberrys).*

***Copias de seguridad y restauración***

- Copia Total (diaria): 10 18 tar -zcvf /home/cod1/bak/homecod1.tgz /home/cod1 && scp /home/cod1/bak/homecod1.tgz 192.168.1.6:/mnt/greyjoy/bak/cod/cod1/homecod1.tgz

(Este apartado está dentro de crontab en ASO).

Ubicación /mnt/greyjoy/bak/cod

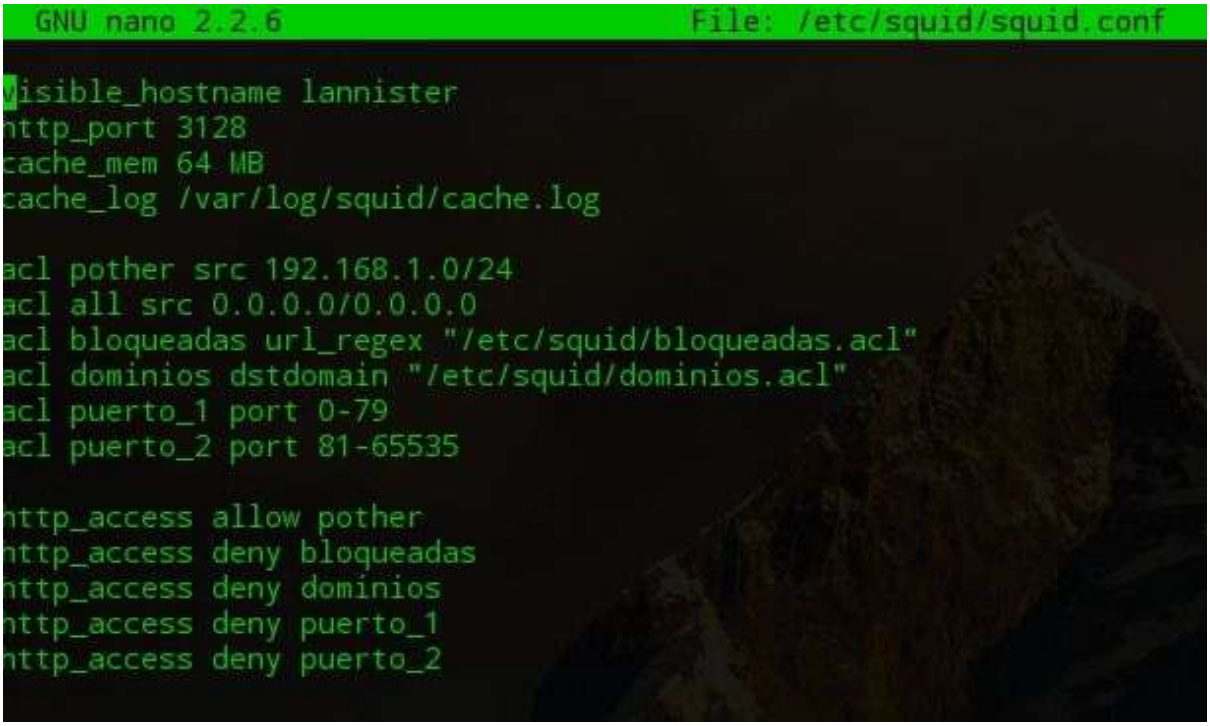
***Lugar donde se ha realizado la maqueta Pother***

Ubicación CPD en la clase 308.
Colocación: Dentro del rack del proyecto.
Servidores: Stark, Targaryen y Lannister.
Raspberrys de los equipos y COD1, COD2.
2 Switches y 2 puntos de acceso.

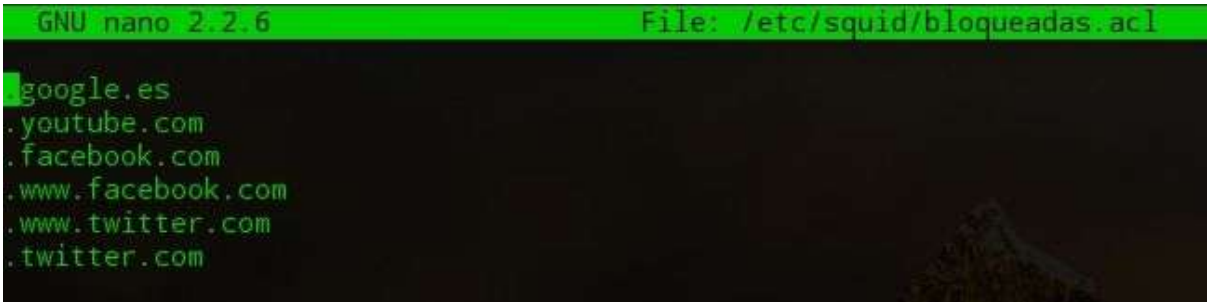
## *Configuración del proxy*

## *Configuración en /etc/squid/squid.conf*

Nombre, puerto, caché, log, listas acl y accesos al proxy.



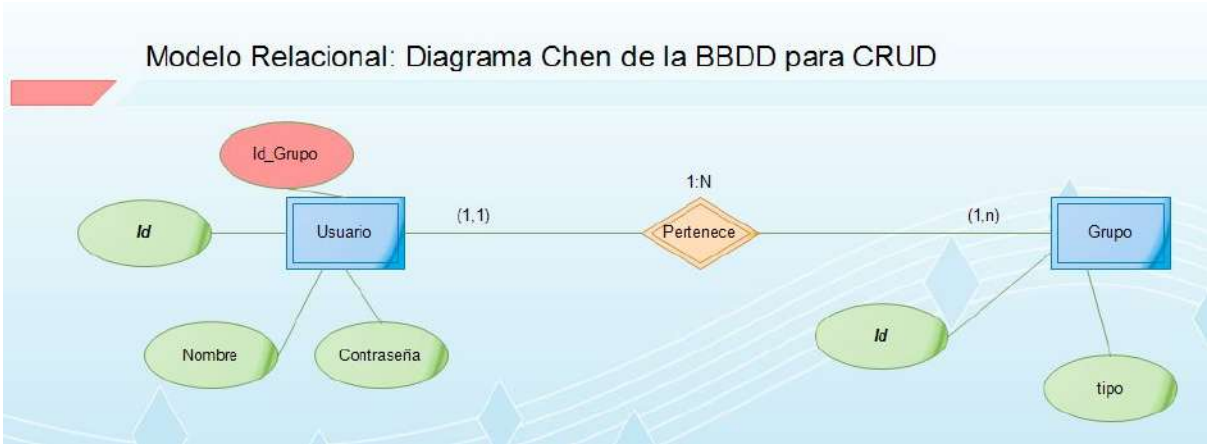

Páginas bloqueadas por el proxy




Palabras de búsqueda bloqueadas


## *Implantación de las BBDD*

Creación de la Base de Datos para el CRUD. Lo primero que haremos será crear el diagrama con el modelo Chen de la BBDD que queremos implementar. De esta forma nos hacemos una idea de lo que queremos implementar a la hora de hacer el script que implemente nuestra solución.




USUARIO {id_usuario, nombre, contrasena, id_grupo} grupo_id hace referencia a id_grupo en GRUPO.
GRUPO {id_grupo, tipo}

Después de crear esto implementaremos en un script .sql este modelo que contendrá a los usuarios dentro de un grupo que nos dirá si son administradores o son usuarios normales de dicha base de datos.

```
  GNU nano 2.2.6              Fichero: bbdd.sql

create database cod;

use cod;

create table usuario
(id_usuario int(10) not null auto_increment,
nombre varchar(255) not null,
contrasena varchar(255) not null,
grupo_id int(10) not null,
primary key (id_usuario)
);

create table grupo
(id_grupo int(10) not null auto_increment,
tipo varchar(255) not null,
primary key (id_grupo)
);

alter table usuario add foreign key (grupo_id) references grupo (id_grupo);

insert into grupo (id_grupo, tipo) values (1,"administrador"), (2,"usuario");

create user admin@localhost identified by 'doc';
create user cod1@localhost identified by 'doc';

grant select,insert,update,delete on cod.* to admin@localhost;
grant select on cod.* to cod1@localhost;
```

Configuración mysql en las Raspberrys. Primeramente entramos en mysql:

```
cod1@raspberrypi:~$ mysql -u root -p
Enter password:
```

Después de esto cargamos el script de la BBDD en las Raspberrys y comprobamos que se ha cargado.

```
mysql> source bbdd.sql
Query OK, 1 row affected (0.20 sec)

Database changed
Query OK, 0 rows affected (0.98 sec)

Query OK, 0 rows affected (0.07 sec)

Query OK, 0 rows affected (0.19 sec)
Records: 0  Duplicates: 0  Warnings: 0

Query OK, 2 rows affected (0.04 sec)
Records: 2  Duplicates: 0  Warnings: 0

Query OK, 1 row affected (0.03 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

mysql>
```

Comprobamos que está la BD en MySQL.

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| cod                |
| mysql              |
| performance_schema |
+--------------------+
4 rows in set (0.00 sec)

mysql>
```

Y después vemos las tablas que tenemos.

```
mysql> show tables;
+---------------+
| Tables_in_cod |
+---------------+
| grupo         |
| usuario       |
+---------------+
2 rows in set (0.00 sec)

mysql>
```

Mostramos los atributos de la tabla grupo y de usuario de manera descendente.

```
mysql> desc grupo;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id_grupo | int(10)      | NO   | PRI | NULL    | auto_increment |
| tipo     | varchar(255) | NO   |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
2 rows in set (0.01 sec)
```

```
mysql> desc usuario;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| id_usuario | int(10)      | NO   | PRI | NULL    | auto_increment |
| nombre     | varchar(255) | NO   |     | NULL    |                |
| contrasena | varchar(255) | NO   |     | NULL    |                |
| grupo_id   | int(10)      | NO   | MUL | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Vemos los grupos que hemos creado para esta BD, en este caso un grupo de administradores que tendrán todos los privilegios y otro para usuario normales.

```
mysql> select * from grupo;
+----------+---------------+
| id_grupo | tipo          |
+----------+---------------+
|        1 | administrador |
|        2 | usuario       |
+----------+---------------+
2 rows in set (0.01 sec)
```

***Primera idea para acceder a una web creada con CMS***

El trabajo está desarrollado con Wordpress, que es un CMS, del inglés Content Management System. Este CMS está enfocado a la creación de blogs y portales web. Está desarrollado en PHP y MySQL, bajo licencia GPL y código modificable.

La creación de nuestros CMS fue utilizando muchas herramientas que nos ofrece Wordpress para plantillas, crear una tienda, SEO, Cookies, Seguridad, copia de seguridad...

Para crear el sitio web principal utilizaremos los lenguajes aprendidos HTML, CSS, Javascript, PHP.

PHP es, junto con mysql, el complemento ideal del servidor web apache ya que dota al servidor de un lenguaje script de ejecución en el servidor lo que facilita la creación de aplicaciones web y sitios web dinámicos.

Lo primero para poder trabajar con PHP debemos instalar en las Raspberrys el módulo PHP. Para ello seguiremos los siguientes pasos:

## Instalación de PHP

Para instalar PHP en nuestra Raspberry utilizaremos apt-get. El paquete a instalar depende de la versión que deseemos instalar y la versión de apache que tengamos, en nuestro caso utilizamos la versión 2 de apache e instalamos la versión 5 de php con → sudo apt-get install libapache2-mod-php5

```
cod1@raspberrypi:~$ sudo apt-get install libapache2-mod-php5
[sudo] password for cod1:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libonig2 libperl4-corelibs-perl libqdbm14 lsof php5-cli php5-common php5-json php5-readline
Paquetes sugeridos:
  php-pear php5-user-cache
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php5 libonig2 libperl4-corelibs-perl libqdbm14 lsof php5-cli php5-common php5-json php5-readline
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 120 no actualizados.
Se necesita descargar 5.084 kB de archivos.
Se utilizarán 18,8 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] S
```

Al instalar libapache2-mod-php5 mediante apt-get, automáticamente se configura para integrarse perfectamente en apache, creando los archivos necesarios en la carpeta (/etc/apache2/mods-available) y creando los enlaces necesarios para habilitarlos en (/etc/apache2/mods-enabled).

Si vamos a conectar a bases de datos mysql desde php, necesitamos instalar el módulo php5-mysql:

sudo apt-get install php5-mysql

```
C:\Windows\system32>gpupdate
Updating policy...

Computer Policy update has completed successfully.
User Policy update has completed successfully.


C:\Windows\system32>_
```

Además, tendremos que editar el archivo /etc/php5/apache2/php.ini y añadir la línea **extension=mysql.so** como veremos en el siguiente apartado.

```
;
  extension=msql.so
```

## Configuración de PHP

El archivo de configuración de php5 es el archivo:

// Archivo de configuración de php5
/etc/php5/apache2/php.ini

```
cod1@raspberrypi:/etc/php5/apache2$ ls
conf.d  php.ini
cod1@raspberrypi:/etc/php5/apache2$ sudo nano php.ini
```

Los parámetros más destacables a configurar son:

- **Safe Mode = Off** (Modo Seguro. Si el Modo seguro está desactivado, se habilitan todas las funciones del PHP. Para un uso educativo es mejor ser funcional y no activar el modo seguro. Si el Modo seguro está activado, se deshabilitan todas las funciones del PHP consideradas peligrosas. Para servicios de hosting se recomienda activar el modo seguro)

- **Display errors = On** (Mostrar Errores. Muestra los errores en las mismas páginas, cuando les haya. Cuando hay errores en los scritps, es más fácil encontrarlos si se muestran en las páginas)
- **max_execution_time=30** (Tiempo máximo en segundos, de ejecución de un script. Si dejamos que un script se ejecute indefinidamente, podría colapsar el sistema)

```
; Maximum execution time of each script, in seconds
; http://php.net/max-execution-time
; Note: This directive is hardcoded to 0 for the CLI SAPI
max_execution_time = 30
```

- **post_max_size=8M** (Tamaño máximo de datos que se pueden enviar al servidor mediante POST)
- **upload_max_filesize = 8M** (Tamaño máximo de archivo que se puede subir al servidor. Si vamos a trabajar con archivos grandes, debemos subir este parámetro)
- **extension=mysql.so** (Activa el acceso a bases de datos MySQL desde PHP)

## Probando PHP

Una vez instalado y configurado, antes de probar debemos reiniciar el servidor web apache:

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

Ahora crearemos una página php que utilice la función phpinfo que además de comprobar que apache y php están funcionando, nos mostrará una información de la versión. Crearemos el siguiente archivo:

```
// Probando PHP. Crear archivo /var/www/phpinfo.php - permisos 644

<HTML>
<H1>Probando PHP</H1>
Salida del comando phpinfo:

<?php
        phpinfo();
?>

</HTML>
```

Ahora tan solo necesitamos arrancar el navegador e ir a la URL: http://ip-del-servidor/phpinfo.php. Si nos aparece la información de la versión de PHP significa que está correctamente instalado.

En el siguiente ejemplo vemos un programa escrito en PHP que, mediante un **bucle for** que va desde 1 hasta 10, muestra la tabla de multiplicar del 7. Si después observamos la página desde el cliente, no vemos más que la respuesta del programa PHP pero nunca el programa.

```
// Programa PHP para generar la tabla del 7. Se almacena en el servidor. Se mezcla el HTML con el código PHP
<html>
<body>
<?
// Tabla de multiplicar del 7
echo "<h2>Tabla del 7</h2>";
```

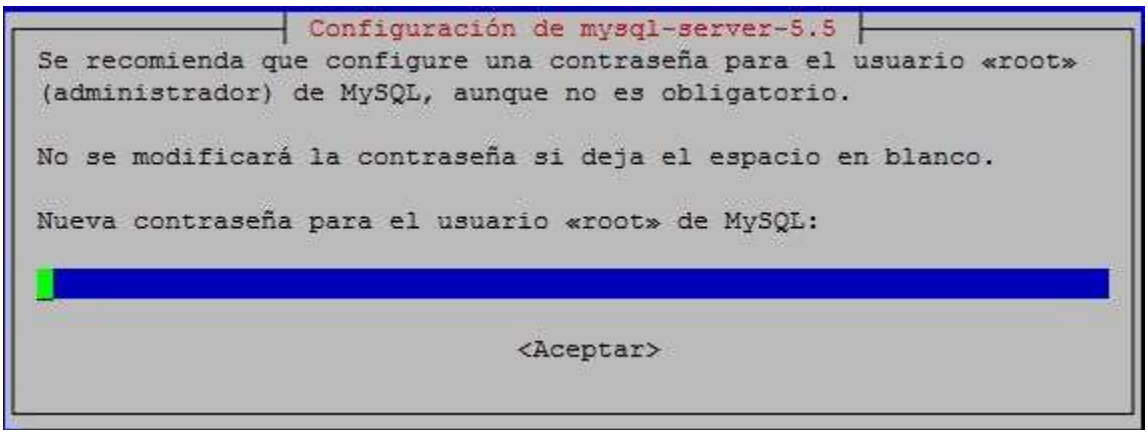
```
// Bucle de 1 a 10
for($i=1; $i<11; $i++)
echo "7 x $i = ".(7*$i)."<br>\n";
?>
</body>
</html>

// Lo que ve el cliente, una vez ejecutado el programa en el servidor: HTML puro
<html>
<body>
<h2>Tabla del 7</h2>7 x 1 = 7<br>
7 x 2 = 14<br>
7 x 3 = 21<br>
7 x 4 = 28<br>
7 x 5 = 35<br>
7 x 6 = 42<br>
7 x 7 = 49<br>
7 x 8 = 56<br>
7 x 9 = 63<br>
7 x 10 = 70<br>
</body>
</html>
```

### *Instalación MySQL Server*

Lo primero que haremos es instalar mysql-server

```
cod1@raspberrypi:~$ sudo apt-get install mysql-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  libaio1 libdbd-mysql-perl libdbi-perl libhtml-template-perl libmysqlclient18
  libterm-readkey-perl mysql-client-5.5 mysql-common mysql-server-5.5
  mysql-server-core-5.5
Paquetes sugeridos:
  libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl
  libipc-sharedcache-perl mailx tinyca
Se instalarán los siguientes paquetes NUEVOS:
```

Durante la instalación nos pedirá una contraseña para el servidor mySQL.




Una vez instalado actualizaremos la lista de paquetes disponibles.

```
cod1@raspberrypi:~$ sudo apt-get update
```

Verificamos que esté realizada la instalación y vemos que está escuchando y el puerto.

```
cod2@raspberrypi:~$ sudo netstat -tap | grep mysql
tcp        0      0 localhost:mysql         *:*                     LISTEN      82
99/mysqld
cod2@raspberrypi:~$
```

Comprobamos el fichero de configuración que esté haciendo el bind a la propia máquina

```
cod1@raspberrypi:/$ cd etc/mysql
cod1@raspberrypi:/etc/mysql$ ls -l
total 16
drwxr-xr-x 2 root root 4096 feb 24 17:26 conf.d
-rw------- 1 root root  333 feb 24 17:26 debian.cnf
-rwxr-xr-x 1 root root 1220 ene 20 04:36 debian-start
-rw-r--r-- 1 root root 3533 ene 17 12:04 my.cnf
cod1@raspberrypi:/etc/mysql$ sudo nano my.cnf
```

```
  GNU nano 2.2.6               Fichero: my.cnf

#
# The MySQL database server configuration file.
#
```

Hacemos un restart a mySQL

```
cod1@raspberrypi:/$ sudo /etc/init.d/mysql restart
[sudo] password for cod1:
[....] Restarting mysql (via systemctl): mysql.service
```

Y por último accedemos a la consola de mySQL.

```
cod1@raspberrypi:/$ mysql -u root -p
Enter password:
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37
Server version: 5.5.54-0+deb8u1 (Raspbian)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Estos pasos anteriores de instalación de Apache y MySQL, los podemos hacer también instalando solamente LAMP dónde viene todo esto integrado.

Para ofrecer este servicio hemos programado un CRUD (altas, bajas, modificaciones y consultas). Este CRUD dará acceso a la página web mediante un registro previo (o un login si ya estamos registrados). El usuario administrador tendrá acceso al panel CRUD donde podrá gestionar y administrar a los usuarios. Un usuario registrado cualquiera accederá directamente al servicio de Wordpress.

Nuestro Wordpress es un sitio con un pequeño curso de SEO (con lo que se cumplirá otra parte del curso de IAW) en el que se subirán noticias mediante un blog en el que se actuará con los usuarios y se publicarán las últimas novedades sobre SEO. Además contiene una tienda creada con WooCommerce.

***Publicaciones y descripción de todos los apartados del portal***

***Inicio:*** Página principal de mi portal web con un menú con las diferentes entradas,tienda y aspectos legales. En él se puede ver el tema utilizado. Encontramos una descripción de lo que nos vamos a encontrar, una serie de enlaces a laspáginas del sitio web, un formulario de contacto y un pie de página. Incluye elapartado blog donde se incluirán nuevas noticias sobre posicionamiento.


***SEO, SEM, Móviles:***
Entradas sobre los temas de SEO, SEM y móviles en modo blog. Estas páginas tienen submenús, que son:

***SEO:*** Escribir bien, Primeros pasos y Consejos, Metatags, Robots.
***SEM:*** Marketing, Publicidad y Anuncios y e-commerce.
***Móviles:*** SEO en Móviles y ASO.
Además tenemos la Tienda que tiene una cuenta, un carrito y finalizar compra.Y que veremos en el dossier de Tienda.
La última página es sobre nosotros donde están además las páginas fijas de Condiciones de envío, Política de Privacidad y Aviso legal.

En este sitio se podrá navegar de manera cómoda, en un entorno amigable y atendiendo a la usabilidad en un navegador web.

***Idea final para acceder a una web creada con Blogger***

Al final el acceso que daremos desde el CRUD a los usuarios será a un blog creado con Blogger. En nuestro caso utilizaremos el blog:

http://cetme77.tk/

El blog es del grupo de música de uno de los miembros COD, Carlos. El blog CETME 77, incluye:
Menú, Archivo PDF, Sonido Soundcloud, Pop-Up a Youtube, Slide & FanBox Twitter y Facebook, Script, Favicon, Adsense, Analytics, Subscripción, Cursor, Template modificado, Pie de página con Copyrigth, Contador de visitas, Enlaces a otros blogs, buscador, vídeo, archivo del blog, tags y revista. Hay varios post subidos actualmente y tiene un diseño responsive.


## *Auditoría de las máquinas*

Realizaremos la auditoría a las máquinas que tenemos. Empezaremos con las máquinas que llevan Linux y Debian Jessie. Para realizar la auditoría en estas máquinas utilizaremos Lynix. Lo primero de todo será crear un directorio dentro de /etc . Una vez hecho esto nos cambiamos a este directorio. Lo siguiente será conseguir de la web el paquete Lynis que viene comprimido.

```
cod2@raspberrypi:/$ sudo mkdir /etc/lynis/
mkdir: no se puede crear el directorio «/etc/lynis/»: El fichero ya existe
cod2@raspberrypi:/$ cd /etc/lynis
cod2@raspberrypi:/etc/lynis$ sudo wget http://cisofy.com/files/lynis-1.3.7.tar.gz
--2017-03-09 15:55:13--  http://cisofy.com/files/lynis-1.3.7.tar.gz
Resolviendo cisofy.com (cisofy.com)... 37.97.224.115, 2a01:7c8:aac4:309::1
Conectando con cisofy.com (cisofy.com)[37.97.224.115]:80... conectado.
Petición HTTP enviada, esperando respuesta... 301 Moved Permanently
Localización: https://cisofy.com/files/lynis-1.3.7.tar.gz [siguiendo]
--2017-03-09 15:55:13--  https://cisofy.com/files/lynis-1.3.7.tar.gz
Conectando con cisofy.com (cisofy.com)[37.97.224.115]:443... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 134683 (132K) [application/octet-stream]
Grabando a: "lynis-1.3.7.tar.gz"

lynis-1.3.7.tar.gz   100%[=====================>] 131,53K  --.-KB/s   en 0,1s

2017-03-09 15:55:13 (1,31 MB/s) - "lynis-1.3.7.tar.gz" guardado [134683/134683]
```

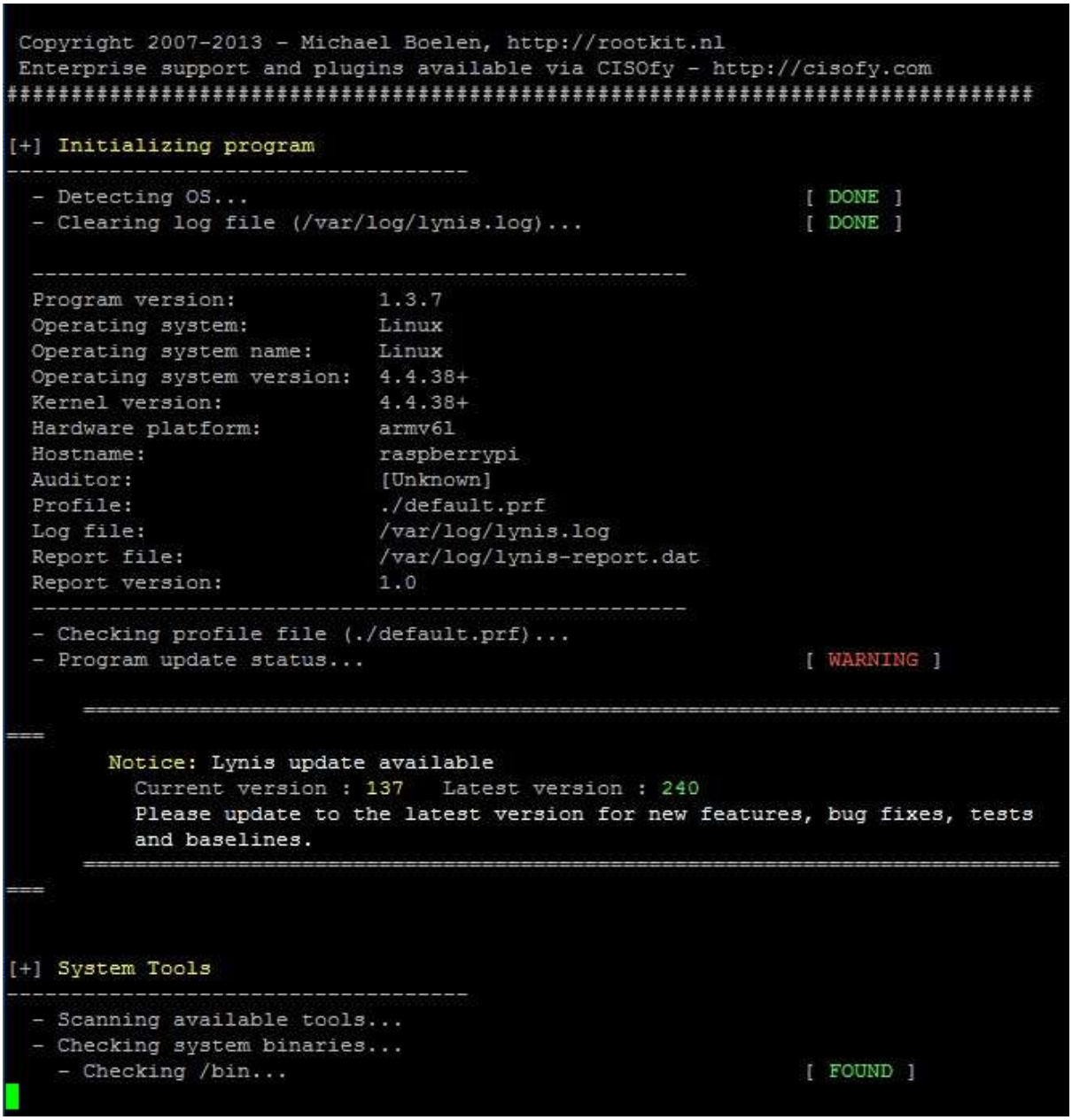
Después de esto desempaquetamos y descomprimimos

```
cod2@raspberrypi:/etc/lynis$ sudo tar xvfz lynis-1.3.7.tar.gz
lynis-1.3.7/CHANGELOG
lynis-1.3.7/FAQ
lynis-1.3.7/INSTALL
lynis-1.3.7/LICENSE
lynis-1.3.7/README
lynis-1.3.7/db/
lynis-1.3.7/db/integrity.db
```

Nos movemos a la carpeta Lynis y ejecutamos un escaneo rápido de nuestro sistema

```
cod2@raspberrypi:/etc/lynis$ cd /etc/lynis/lynis-1.3.7
cod2@raspberrypi:/etc/lynis/lynis-1.3.7$ sudo ./lynis -c -Q
```

Una vez realizado el escaneo ya podemos ver como se encuentran nuestros equipos y los warnings que nos muestra.




En nuestro caso hemos detectado errores en las actualizaciones del kernel de Linux en las Raspberrys. Todo lo demás lo encuentra y está actualizado dentro de nuestros sistemas Linux.
Este error de actualización puede que sea el causante de no haber podido realizar con éxito tareas como el cluster de alta disponibilidad y otras tareas parecidas.

```
[+] Logging and files
------------------------------------
  - Checking for a running log daemon...                    [ OK ]
    - Checking Syslog-NG status                             [ NOT FOUND ]
    - Checking Metalog status                               [ NOT FOUND ]
    - Checking RSyslog status                               [ FOUND ]
    - Checking RFC 3195 daemon status                       [ NOT FOUND ]
    - Checking minilogd instances                           [ NONE ]
  - Checking logrotate presence                             [ OK ]
  - Checking log directories (static list)                  [ DONE ]
  - Checking open log files                                 [ DONE ]
  - Checking deleted files in use                           [ FILES FOUND ]

[+] Insecure services
------------------------------------
  - Checking inetd status...                                [ NOT ACTIVE ]

[+] Banners and identification
------------------------------------
  - /etc/motd...                                            [ FOUND ]
    - /etc/motd permissions...                              [ OK ]
    - /etc/motd contents...                                 [ WEAK ]
  - /etc/issue...                                           [ FOUND ]
    - /etc/issue contents...                                [ WEAK ]
  - /etc/issue.net...                                       [ FOUND ]
    - /etc/issue.net contents...                            [ WEAK ]

[+] Scheduled tasks
------------------------------------
  - Checking crontab/cronjob                                [ DONE ]
  - Checking atd status                                     [ NOT RUNNING ]

[+] Accounting
------------------------------------
  - Checking accounting information...                      [ NOT FOUND ]
  - Checking auditd                                         [ NOT FOUND ]

[+] Time and Synchronization
------------------------------------
  - Checking running NTP daemon (ntpd)...                   [ FOUND ]
  - Checking running NTP daemon (timed)...                  [ NOT FOUND ]
```

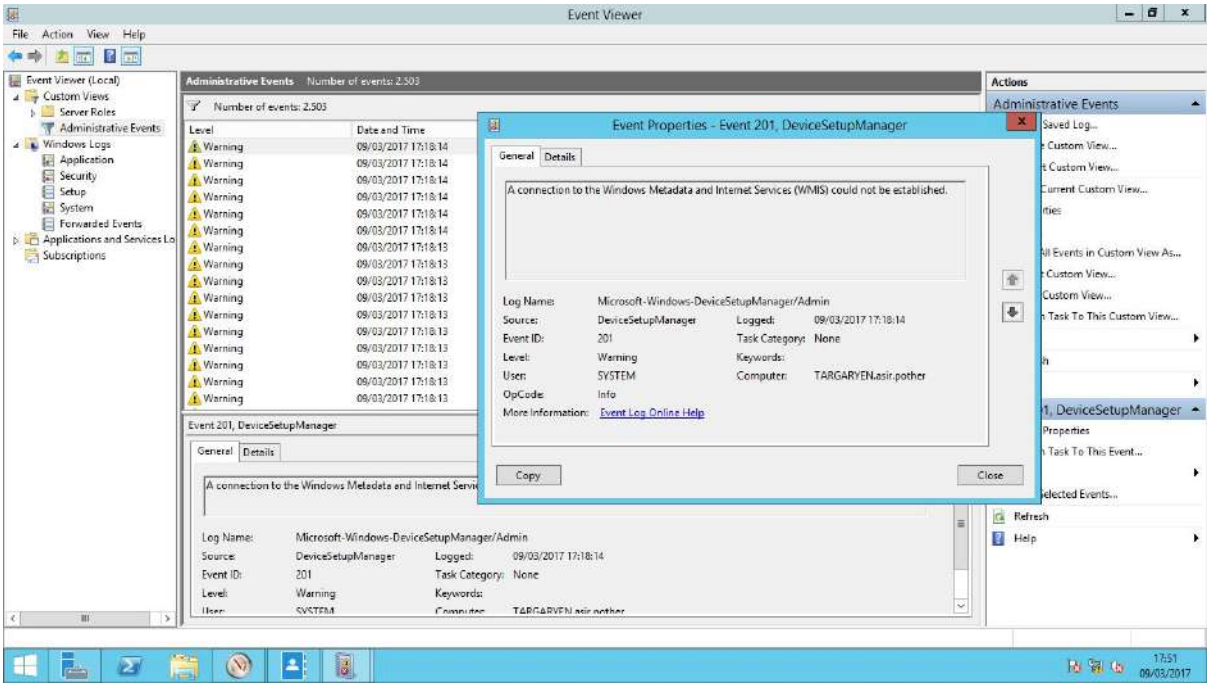
```
================================================================================
  Files:
  - Test and debug information      : /var/log/lynis.log
  - Report data                     : /var/log/lynis-report.dat
================================================================================
  Notice: Lynis update available
  Current version : 137    Latest version : 240
================================================================================
  Hardening index : [57]     [###########         ]

  Enterprise support and plugins available via CISOfy - http://cisofy.com
================================================================================
  Tip: Disable all tests which are not relevant or are too strict for the
       purpose of this particular machine. This will remove unwanted suggestions
       and also boost the hardening index. Each test should be properly analyzed
       to see if the related risks can be accepted, before disabling the test.
================================================================================
  Lynis 1.3.7
  Copyright 2007-2013 - Michael Boelen, http://rootkit.nl
================================================================================


cod2@raspberrypi:/etc/lynis/lynis-1.3.7$
```

Después de hacer estos test esta es la puntuación que nos da Lynix.

Los mismos pasos los hemos seguido con el servidor Stark

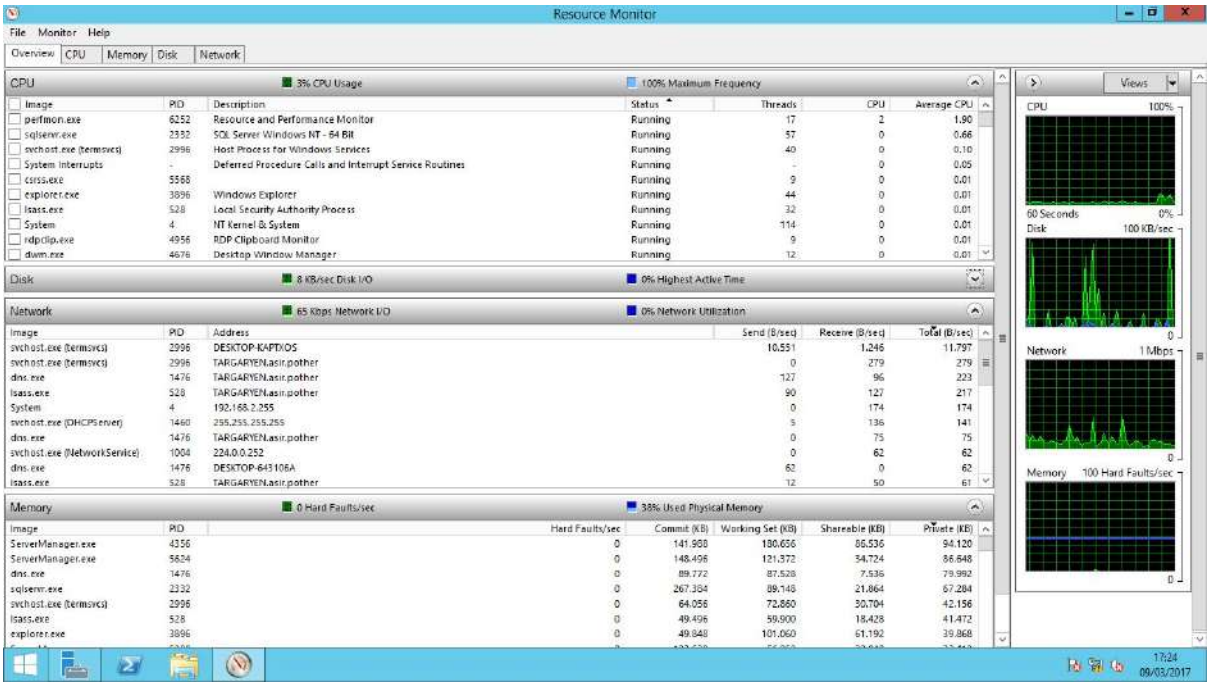
```
stark@stark:/etc/lynis/lynis-1.3.7$ sudo ./lynis -c -Q
```


Por último entramos en el servidor Targaryen y le hacemos un checkeo viendo los eventos, del sistema los logs que tenemos, los warnings que se han producido...

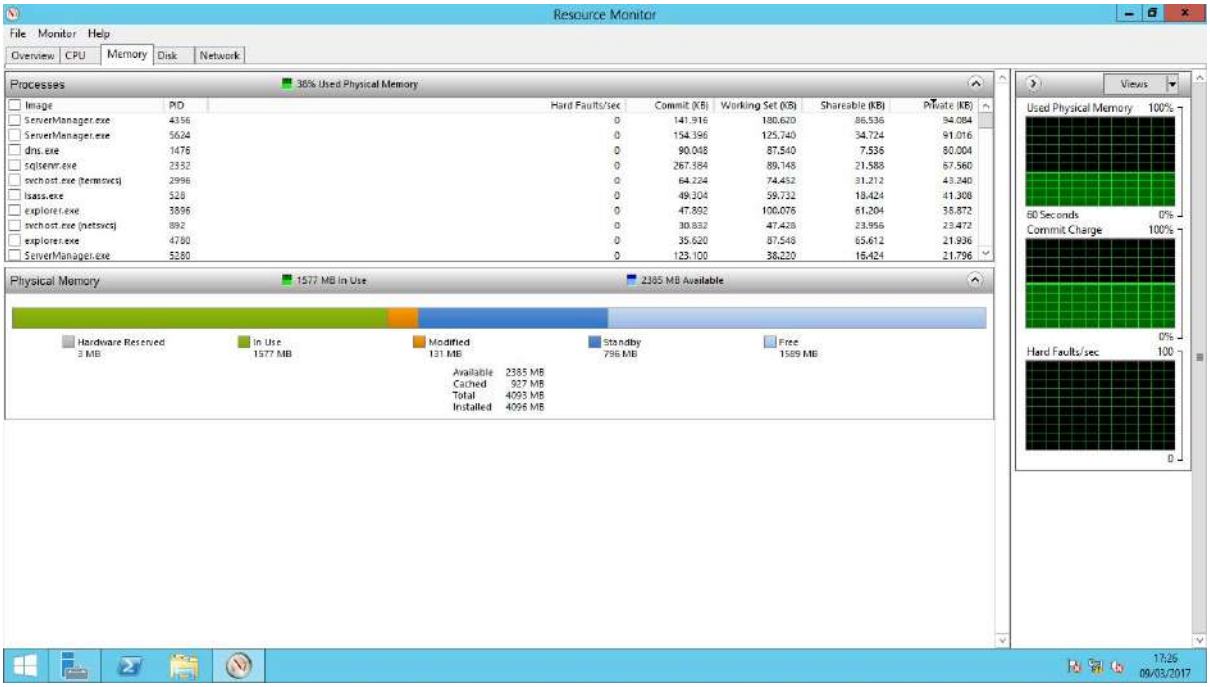


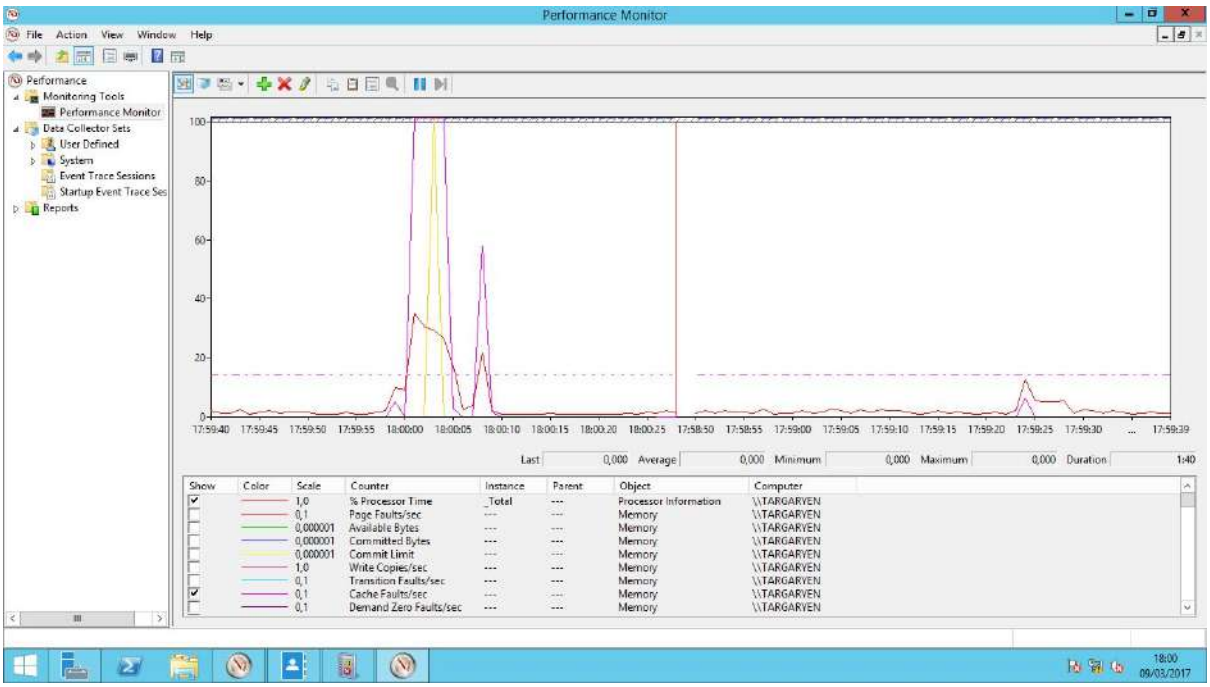

El visor de eventos nos muestra errores que se han producido o determinadas alertas.

Además, vemos los recursos que utiliza o consume nuestro sistema tanto de memoria como CPU




Como podemos ver tenemos suficientes herramientas y muy visuales en Windows para monitorear el sistema, ver los recursos que consumimos y como afecta esto a nuestro servidor.




## *Auditoría web de nuestro sitio COD*

# *Bibliografía*

Apuntes y documentación de las asignaturas del proyecto
http://kanbantool.com/es/metodologia-kanban
https://www.raspberrypi.org/downloads/raspbian/
Edrawsoft → Diagramas

***Método Kanban***
https://github.com/CODasir16/COD
https://waffle.io/CODasir16/COD
https://subefotos.com/

***Video animado sobre el proyecto Pothers***

https://www.youtube.com/watch?v=x4Zl7UOZu9U&feature=youtu.be

***Creado con:*** http://wideo.co/ →
http://www.wideo.co/view/18880501486307347686?utm_source=CopyPaste&utm_medium=share&utm_campaign=sharebox
MusicMaker
Imágenes y gifs (Google)

***Música:*** Code Or Digit - Digital Pop, MusicMaker, Carlos del Cerro