

A thick dark blue vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the date '17-6-2020'. In the lower-left corner, there are several thin, curved lines in dark blue and light grey, resembling stylized grass or abstract brushstrokes.

17-6-2020

# Autenticación a una red Wi-Fi mediante un Servidor RADIUS

Proyecto Integrado CFGS

Jorge Rodríguez Mora  
IES GRAN CAPITÁN

## Índice

1.	Introducción.....	2
2.	Objetivos y Requisitos del proyecto.....	3
3.	Estudio Previo.....	4
3.1.	Estado actual.....	4
3.2.	Estudio de las soluciones existentes.....	5
4.	Plan de Trabajo.....	6
5.	Diseño.....	8
5.1.	Diseño general .....	8
5.2.	Diseño detallado.....	8
6.	Implementación.....	9
6.1.	Creación del escenario de pruebas .....	9
6.1.1.	Instalación del Servidor Radius. ....	9
6.1.2.	Configuración del punto de acceso.....	23
6.2.	Autenticación de clientes inalámbricos por Radius.....	25
6.2.1.	Edición del archivo clients.conf .....	25
6.2.2.	Configuración de los usuarios.....	26
6.2.3.	Conexión a la red.....	28
6.3.	FreeRadius con MySQL.....	31
6.3.1.	Instalación de MySQL .....	31
6.3.2.	Configuración de FreeRadius con MySQL .....	31
6.4.	FreeRadius con OpenLDAP .....	41
6.4.1.	Instalación y configuración de LDAP .....	41
6.4.2.	Configuración de FreeRadius con LDAP.....	44
6.5.	Autenticación de clientes inalámbricos mediante certificados. ....	47
6.5.1.	Generación de certificados.....	47
6.5.2.	Configuración de FreeRadius.....	49
6.5.3.	Autenticación de clientes inalámbricos a la red. ....	51
6.6.	Almacenamiento de claves seguras mediante funciones hash. ....	55
6.6.1.	Base de Datos interna FreeRadius. ....	56
6.6.2.	Base de datos MySQL. ....	57
6.6.3.	Base de datos LDAP. ....	58
7.	Incidencias, objetivos cumplidos y mejoras.....	60
8.	Webgrafía.....	61

## 1. Introducción

Uno de los aspectos más importantes de la Informática es la seguridad, las empresas como por ejemplo los bancos, tienen muchísimos datos sensibles (DNI, nombre, dirección teléfono) que tienen que proteger.

Cuando te acercas a una de estas empresas cualquier dispositivo móvil puede detectar su red, que en la mayoría de los casos está protegida por el protocolo WPA2 o WAP o WEP, que son los más comunes, cada uno de ellos utiliza un algoritmo que encripta la contraseña.

Pero no por ello podemos decir que la clave es segura del todo, debemos hacer todo lo posible para restringir el acceso a nuestra red. Una forma de tener mucho más control sobre el acceso a nuestra red es con un servidor Radius.

Con esta opción sólo los usuarios que estén en una base de datos (ya sea MySQL u OpenLDAP) filtrando así los usuarios que se pueden conectar a nuestra red evitando así que cualquier infiltración a nuestra red.

Una de las ventajas de usar esta opción es la encriptación de las claves, que consiste que en vez de tener la contraseña en texto plano, se use un algoritmo para ocultar dicha clave y en vez de la contraseña aparecerán una serie de caracteres alfanuméricos, mejorando así la seguridad a la hora del acceso.

Un ejemplo de ello sería:

*Sin encriptación*

```
mysql> select * from usuarios;
+-----+-----+
| nombre      | contraseña |
+-----+-----+
| no_encriptado | hola       |
+-----+-----+
1 row in set (0.00 sec)
```

*Con encriptación*

```
mysql> select * from usuarios;
+-----+-----+
| nombre      | contraseña |
+-----+-----+
| no_encriptado | hola       |
| encriptado    | 4d186321c1a7f0f354b297e8914ab240 |
+-----+-----+
2 rows in set (0.00 sec)
```

Aquí se aprecia la diferencia, según el tipo de cifrado que usemos, el del ejemplo es con md5 que te genera un total de 32 caracteres haciendo más difícil de descifrar que un simple "hola".

A continuación, podemos observar que ambas contraseñas son “hola”

```
mysql> select * from usuarios where contrasena='hola';
+-----+-----+
| nombre | contrasena |
+-----+-----+
| no_encryptado | hola |
+-----+-----+
1 row in set (0.00 sec)

mysql> select * from usuarios where contrasena=md5('hola');
+-----+-----+
| nombre | contrasena |
+-----+-----+
| encryptado | 4d186321c1a7f0f354b297e8914ab240 |
+-----+-----+
1 row in set (0.00 sec)
```

Por el contrario, las desventajas son:

- Una clave única para autenticarse.
- Configurar los WAP conllevaría un gran esfuerzo administrativo ya que se realizaría uno a uno.
- Existen herramientas automatizadas para descubrir la clave única.
- La clave se tiene que comunicar a los usuarios y pueden revelarse a través de ingeniería social.

Si se descubre la clave hay habría que reconfigurar todos los WAP y cambiar la configuración de acceso de todos los nodos inalámbricos.

Con Radius, que es un protocolo que permite mejorar la seguridad del acceso de los nodos inalámbricos a nuestra red, podemos:

- Configurar un usuario con su nombre y clave.
- Los WAP solamente se configuran para conectar con el servidor Radius y solicitar autorización.
- Si se descubren las credenciales de un usuario, sólo le afecta a él. Se da de baja y la red no queda comprometida, no hay que reconfigurar WAPs ni el resto de nodo inalámbricos.

Con todo esto conseguiremos mejorar la configuración del acceso de los nodos inalámbricos al punto de acceso.

## 2. Objetivos y Requisitos del proyecto

El objetivo de este proyecto es conseguir que los usuarios se puedan conectar a la red inalámbrica mediante autenticación con un servidor Radius.

- Aumento de la seguridad de la red inalámbrica.
- Simplificación de la gestión de usuarios y clientes inalámbricos.
- Uso de diversos tipos de base de datos de usuarios (MySQL y OpenLDAP) y clientes inalámbricos.
- Empleo de software de autenticación inalámbrica gratuito para ahorro de costes.

- Uso del método SCRUM para la gestión del proyecto que permita planificar, desarrollar y verificar los objetivos establecidos.

A continuación, nombro los requisitos del proyecto:

- Los usuarios de los nodos inalámbricos se autenticarán mediante el empleo de credenciales únicas suministradas por el administrador de la red.
- Para conseguir diferentes grados de seguridad se van a emplear diversos métodos de almacenamiento de la contraseña: texto plano y función hash.
- Empleo de diferentes tipos de base de datos de usuarios y clientes inalámbricos para comprobar las ventajas e inconvenientes de los diferentes métodos y así poder elegir el más adecuado al escenario de aplicación.
- Creación de una política de seguridad para el acceso de los nodos inalámbricos a la red.
- Desarrollo del proyecto mediante incrementos verificables.

### 3. Estudio Previo

#### 3.1. Estado actual

Hoy en día, los usuarios que se conectan a una red inalámbrica lo hacen a través de, por ejemplo, su router doméstico proporcionado por su ISP.

Dicho router trae el SSID y la clave para que los usuarios se conecten a la red desde sus dispositivos inalámbricos y puedan navegar por la red.

Para el usuario este sistema es fácil de entender, pero en un ámbito empresarial podemos añadir una capa de seguridad con la autenticación por Radius. Así podemos tener más control sobre nuestra red y de los usuarios que se conectan a ella pudiendo modificar algunos parámetros como por ejemplo el nivel de seguridad de la contraseña.

Esta seguridad viene proporcionada por diferentes protocolos que pueden ser WEP, WPA, WPA2 siendo éste último el más recomendado y es el que se encuentra en la inmensa mayoría de routers.

Con el protocolo WEP se quería conseguir que la seguridad de las redes wifi fueran como la red cableada, pero se encontraron diversas vulnerabilidades con las que se podía acceder muy fácilmente por lo que tuvieron que hacer que fuera más seguro dando lugar a WPA.

Aunque WPA tenía varias mejoras como el uso de una clave precompartida (PSK), denominada también WPA Personal, y el Protocolo de integridad de clave temporal (TKIP) para el cifrado.

Con TKIP se crea una clave temporal de 128 bits que es compartida entre los clientes y los puntos de acceso. Combina la clave temporal con la dirección MAC del cliente y agrega un vector de inicialización de 16 octetos (128 bits) para producir la clave que cifrará los datos y por último algo que mejora significativamente la seguridad de la red es el cambio de claves temporales cada 10.000 paquetes. Aún con todo esto, viendo que WPA es una gran mejora respecto a WEP, se siguieron descubriendo vulnerabilidades que comprometían la seguridad de la red.

En 2004 se adopta WPA2 como el protocolo basado en el estándar de seguridad inalámbrica 802.11i. La mejora más importante que trajo este protocolo fue el uso de Advanced Encryption Standard (AES), como dato el propio Gobierno de EEUU aprobó el uso de AES para la encriptación de los documentos de información clasificada. AES tiene un tamaño de bloque fijo de 128 bits y tamaños de llave de 128, 192 o 256 bits.

Aún con todo esto existen vulnerabilidades en este protocolo, que se descubrieron mediante el uso de Configuración de Wi-Fi Segura (WPS) pudiendo forzar el acceso entre 2 a 14 horas.

Hace relativamente poco tiempo apareció WPA3 para dar solución a los problemas con los anteriores protocolos, que explicaré en el siguiente apartado.

### 3.2. Estudio de las soluciones existentes

Una solución hoy día en la gran mayoría de routers es WPA3 que es la última versión de este protocolo.

WPA (Wi-Fi) Protected Access 3, es un estándar que evita que un tercero sin autorización pueda acceder a la información en tránsito de manera inalámbrica. Aunque la mayoría de dispositivos cuentan con WPA2 con cifrado AES ha dejado de ser seguro, ya que se encontró una vulnerabilidad en este estándar (KRACK).

Por todo ello WPA3 reemplazará al estándar WPA2 con el que se mejorarán en gran medida los mecanismos de autenticación potenciando el uso de protocolos criptográficos robustos. Entre las ventajas encontramos:

- Mayor protección.
- Se refuerza la protección en redes públicas cifrando el tráfico entre nuestro dispositivo y el punto de acceso.
- Cifrado de 192 bits.
- WPA3 Forward Secrecy. Es una característica que evita que un atacante pueda descifrar el tráfico capturado.
- Reemplaza el intercambio de claves pre-compartidas con la autenticación simultánea de iguales.

Otra solución es la que voy a desarrollar en este proyecto usando como autenticación un servidor **Radius** en Ubuntu Server 18.04 LTS (**FreeRadius**), cabe mencionar que también se puede implementar en Windows Server.

Un servidor Radius (Remote Access Dial In User Service) es un protocolo que destaca sobre todo por ofrecer un mecanismo de seguridad, flexibilidad, capacidad de expansión y una administración simplificada de las credenciales de acceso a un recurso de red, se utiliza en un esquema de cliente-servidor. Un usuario con credenciales de acceso se conecta contra un servidor que será el que se encarga de verificar la autenticidad de la información y es el encargado de determinar si el usuario puede acceder al recurso compartido o no.

## 4. Plan de Trabajo

- Creación del escenario de pruebas:
  - o Creación de la máquina virtual en VirtualBox para el servidor Radius. El sistema invitado será Ubuntu 18.04 LTS.
  - o Configuración de red del servidor. Al ser un servidor con acceso por parte de los clientes inalámbricos hay que configurar una dirección IP estática.
  - o Instalación de FreeRadius.
  - o Configuración de FreeRadius para autenticación local mediante base de datos propia con fichero de configuración de FreeRadius.
- Autenticación remota mediante clientes inalámbricos. Configuración del WAP DLINK-615 para autenticación de usuarios.
- Base de datos de clientes inalámbricos con MySQL. → Instalaremos un servidor MySQL, en el que se van a almacenar las credenciales de los usuarios.
- Base de datos de clientes inalámbricos con OpenLDAP. → Instalaremos LDAP para tener una base de datos de los usuarios que se pueden autenticar.
- Autenticación de los nodos inalámbricos con el WAP mediante certificados. → Crearemos certificados de clave asimétrica, permitiendo que los usuarios que dispongan del certificado tengan acceso a nuestra red
- Almacenamiento de claves seguras mediante función hash. → Aumentaremos la seguridad de las contraseñas mediante una función hash, que permite encriptar las contraseñas.

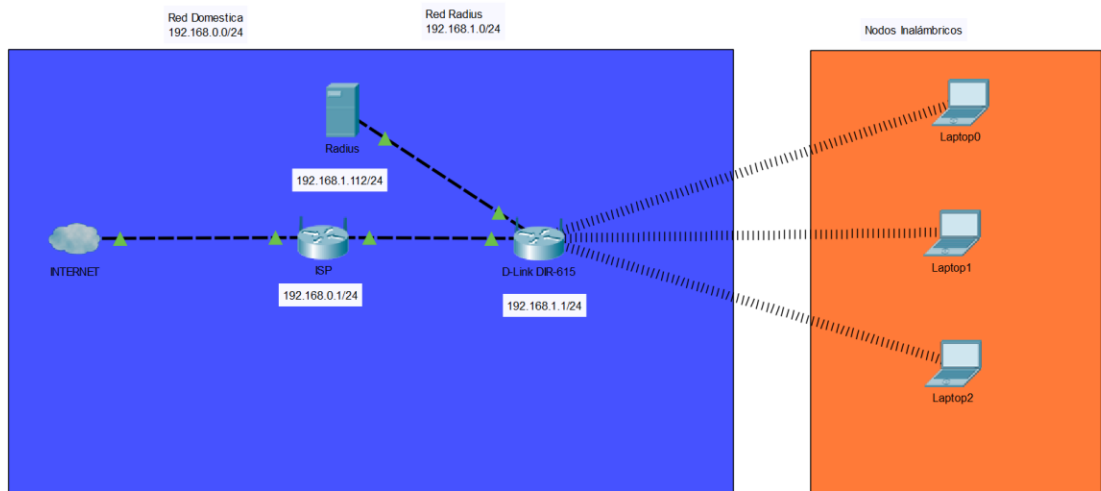
SPRINT	TIEMPO ESTIMADO	FUNCIONALIDAD	PRUEBAS A REALIZAR	FUNCIONALIDAD DEL SISTEMA
Creación del escenario de pruebas.	11 días.	<ul style="list-style-type: none"><li>- Instalación del Servidor Radius.</li><li>- Configuración del punto de acceso.</li></ul>	<ul style="list-style-type: none"><li>- El estado del servicio FreeRadius está activo y el servidor tiene salida a Internet.</li></ul>	CORRECTO
Autenticación de clientes inalámbricos por Radius.	8 días.	<ul style="list-style-type: none"><li>- Edición del archivo clients.conf para permitir la comunicación el punto de acceso.</li><li>- Configuración de los usuarios en la base local de FreeRadius (archivo users).</li></ul>	<ul style="list-style-type: none"><li>- Creación de usuarios en la base de datos interna de FreeRadius, y comprobación de conexión con el comando radtest.</li><li>- Acceso del usuario creado a la red.</li></ul>	CORRECTO

FreeRadius con MySQL.	10 días.	<ul style="list-style-type: none"> <li>- Instalación de la base de datos MySQL.</li> <li>- Importación de la base de datos a MySQL.</li> <li>- Configuración de MySQL y FreeRadius.</li> </ul>	<ul style="list-style-type: none"> <li>- Crear un usuario en la base de datos de MySQL, y comprobar con el comando radtest.</li> <li>- El usuario creado tiene acceso a Internet.</li> </ul>	CORRECTO
FreeRadius con OpenLDAP.	12 días.	<ul style="list-style-type: none"> <li>- Instalación de OpenLDAP.</li> <li>- Configuración de LDAP y FreeRadius.</li> </ul>	<ul style="list-style-type: none"> <li>- Crear y añadir un usuario a la base de datos de LDAP y comprobar con el comando radtest.</li> <li>- El usuario creado debe tener acceso a la red.</li> </ul>	CORRECTO
Almacenamiento de claves seguras mediante funciones hash.	3 días.	<ul style="list-style-type: none"> <li>- Aplicar una función al campo de la contraseña en el archivo users de FreeRadius, MySQL y LDAP.</li> </ul>	<ul style="list-style-type: none"> <li>- Comprobar que los usuarios tienen acceso a la red con el comando radtest.</li> </ul>	CORRECTO
Autenticación de clientes inalámbricos mediante certificados.	14 días.	<ul style="list-style-type: none"> <li>- Creación de certificados de la Autoridad Certificadoras, del servidor y del cliente.</li> <li>- Configurar FreeRadius para permitir el uso de certificados.</li> </ul>	<ul style="list-style-type: none"> <li>- Importar el certificado del cliente al cliente inalámbrico.</li> <li>- El cliente tiene acceso a la red a través del certificado.</li> </ul>	



## 5. Diseño

### 5.1. Diseño general



*Imagen del esquema del proyecto.*

En el esquema de la red podemos ver los componentes:

- **Radius** → Servidor Radius en Ubuntu Server 18.04 LTS en el que estará instalado FreeRadius y las bases de datos mysql y OpenLDAP, contará con una memoria RAM de 1,5GB.
- **Router ISP** → Es el router que tenemos en casa proporcionado por nuestro ISP.
- **Router D-Link DIR-615** → El router que estará configurado en modo radius y a los que se conectarán los clientes para tener acceso a Internet a través del router ISP.
- **Nodos inalámbricos** → Ya sea un móvil, portátil, Tablet...

### 5.2. Diseño detallado

El funcionamiento del flujo de la arquitectura será:

1. El cliente inalámbrico (**Laptop0**) introduce las credenciales de acceso a la red Wi-Fi.
2. El Punto de Acceso (**Router D-LINK DIR-615**) recibe la petición y la envía al servidor Radius (**Radius**) para verificar las credenciales.
3. El servidor Radius (**Radius**) recibe la petición.
4. FreeRadius busca en la base de datos las credenciales de usuario que ha recibido.

5. Al coincidir las credenciales el servidor Radius (**Radius**) manda la petición al punto de acceso, que el cliente que se intenta autenticar tiene acceso a la red.
6. El Punto de Acceso (**Router D-LINK DIR-615**) envía los datos necesarios al cliente para que pueda navegar por la red (p.e. Dirección IP) a través del router doméstico(**ISP**).

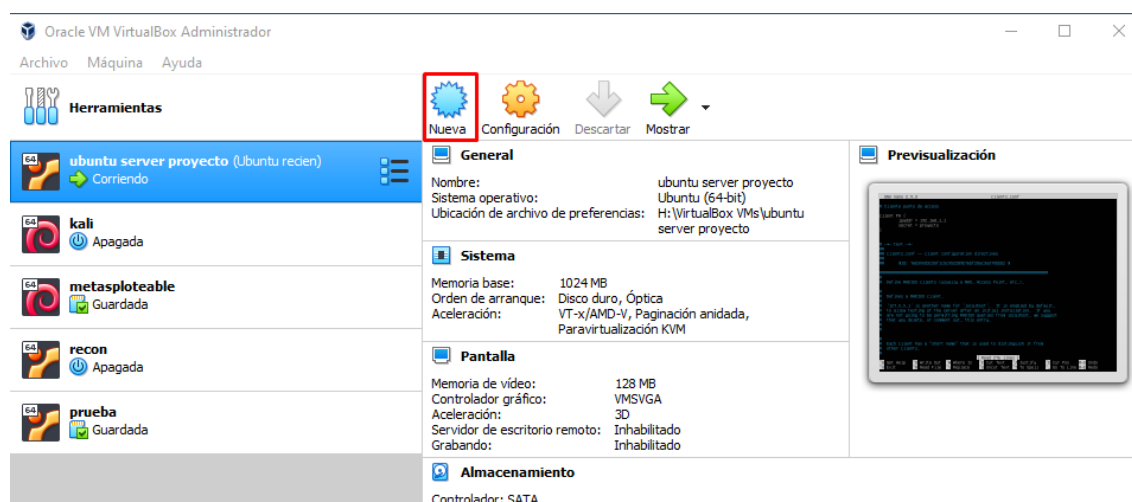
Con todo esto conseguiremos el objetivo del proyecto, permitir que los usuarios puedan conectarse a la red no por clave WPA/WEP/WEP2, sino a través de nuestro servidor Radius. Si las credenciales que introduce el usuario coinciden con las de la base de datos de nuestro servidor Radius, el usuario tendrá acceso a la red. Aplicando así una capa de seguridad a nuestra red y protegiéndola de usuarios desconocidos que no estén en la base de datos de usuarios que pueden acceder a la red.

## 6. Implementación

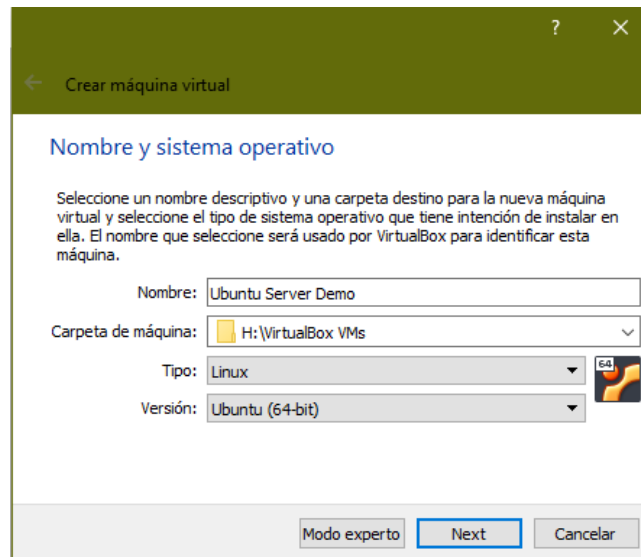
### 6.1. Creación del escenario de pruebas

#### 6.1.1. Instalación del Servidor Radius.

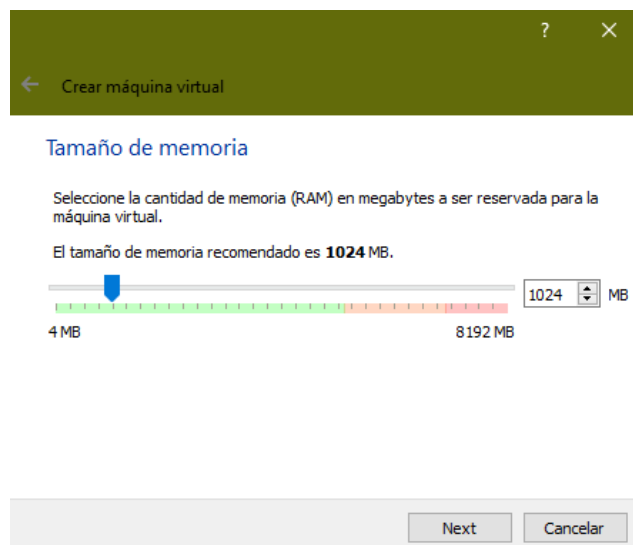
La instalación del servidor RADIUS se hará en una máquina virtual, para ello, debemos instalar el software necesario. Lo primero deberemos ir a la página web de Virtual Box y descargarnos la última versión ([enlace](#)). Una vez que tengamos instalado el software procederemos a la creación de una nueva máquina.



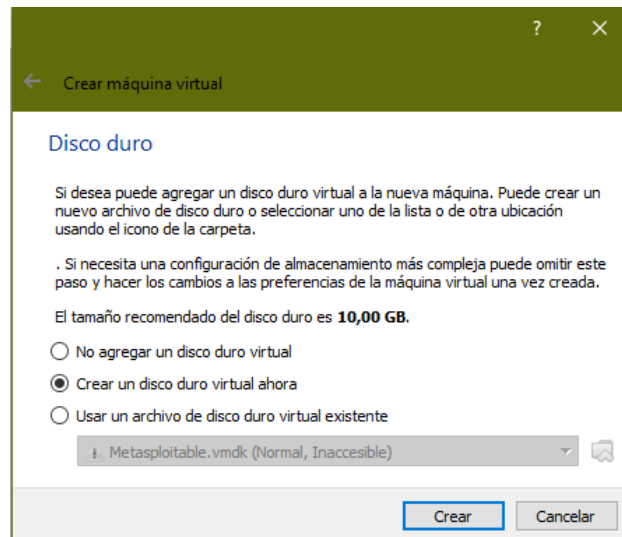
A continuación, añadimos un nombre para nuestra máquina virtual, elegimos donde se va a guardar dentro de nuestro sistema, elegimos el SO que vamos a instalar, que será **Linux**, y por último elegimos la distribución que vamos a usar **Ubuntu x64**, esto último dependerá de la arquitectura de nuestro ordenador.



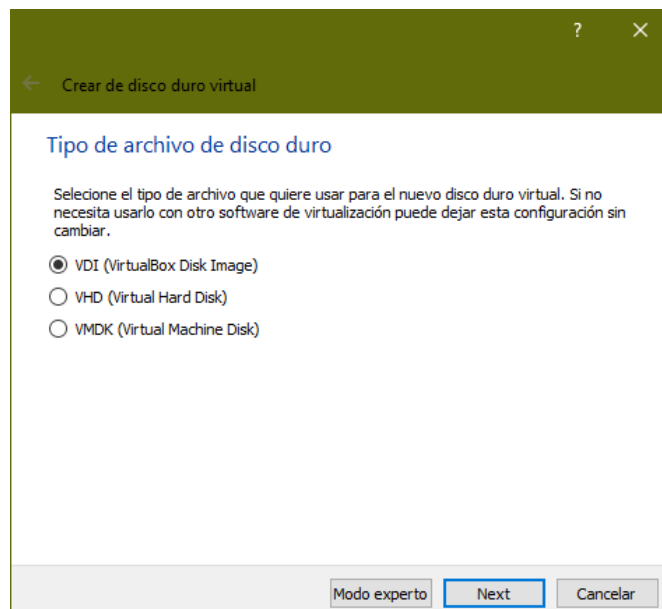
Elegimos los GB de memoria RAM que se van a usar en la máquina, esto dependerá del equipo de cada uno, pero para este caso con 1GB bastará, no obstante hay que tener cuidado de no dejar a la máquina física (nuestro ordenador) sin memoria RAM suficiente para que pueda funcionar correctamente.



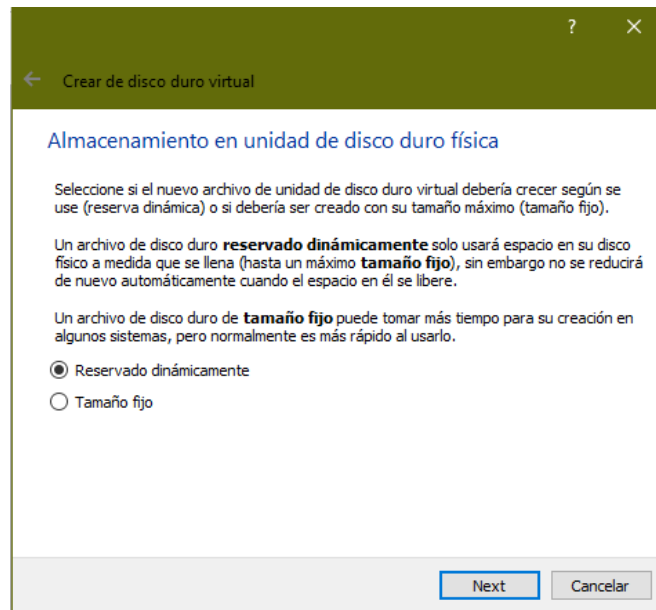
Vamos a crear el disco duro virtual donde se van a almacenar todo el sistema nuestra máquina virtual.



El tipo de disco duro será VDI, aunque esto es elección personal ya que VDI permite modificar el tamaño del disco virtual desde powershell o cmd.



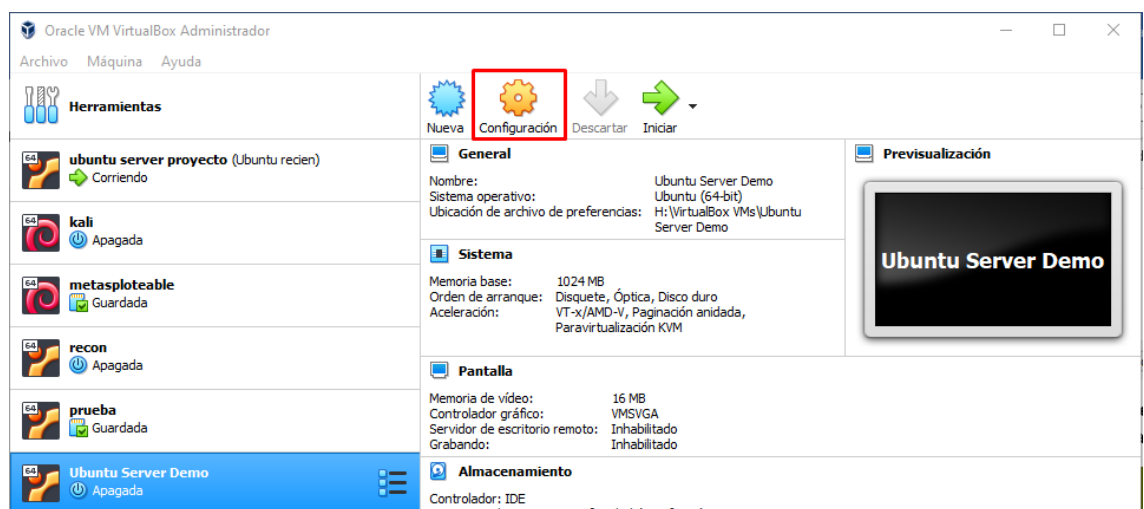
El almacenamiento va a ser **Reservado dinámicamente** para que el sistema vaya reservando espacio según se necesite.



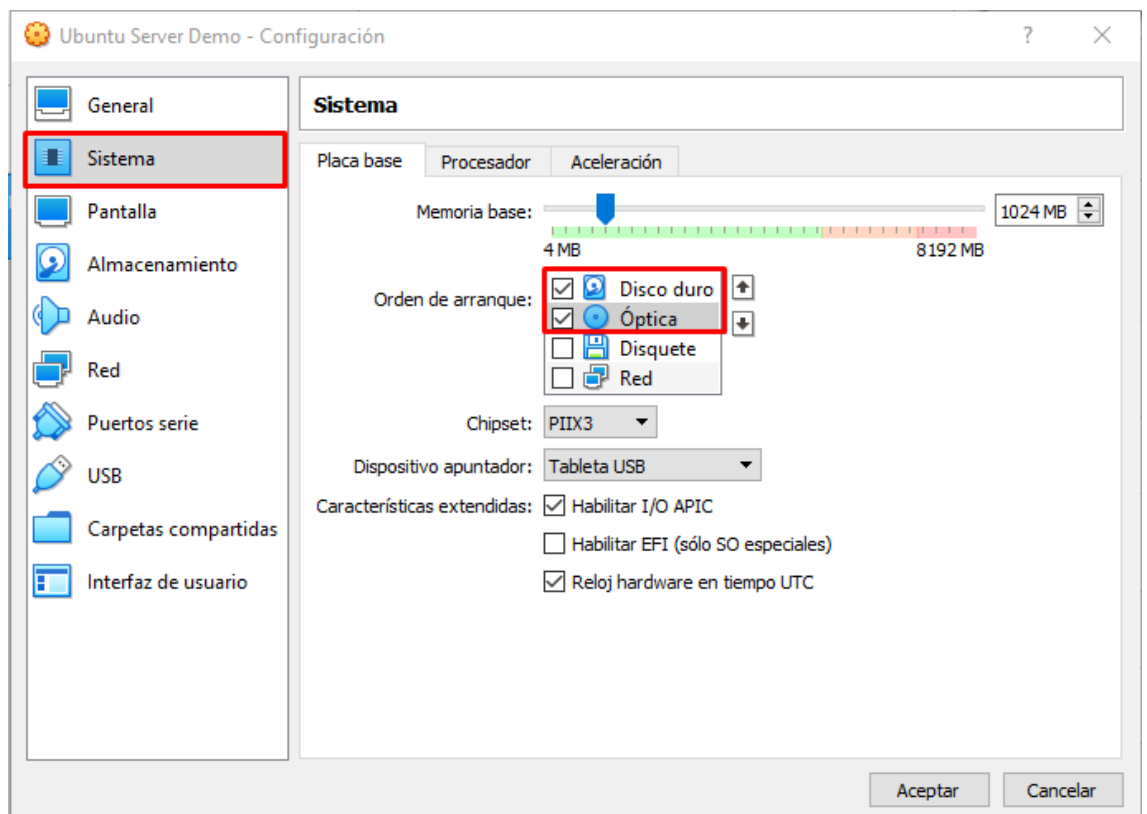
Ahora si queremos cambiar el nombre del disco lo podemos modificar, así como el espacio y ya podemos crear el disco.

Para la instalación del servidor también necesitaremos el sistema operativo, que será **Ubuntu Server 18.04 LTS**, el cual podremos descargar desde su página web ([enlace](#))

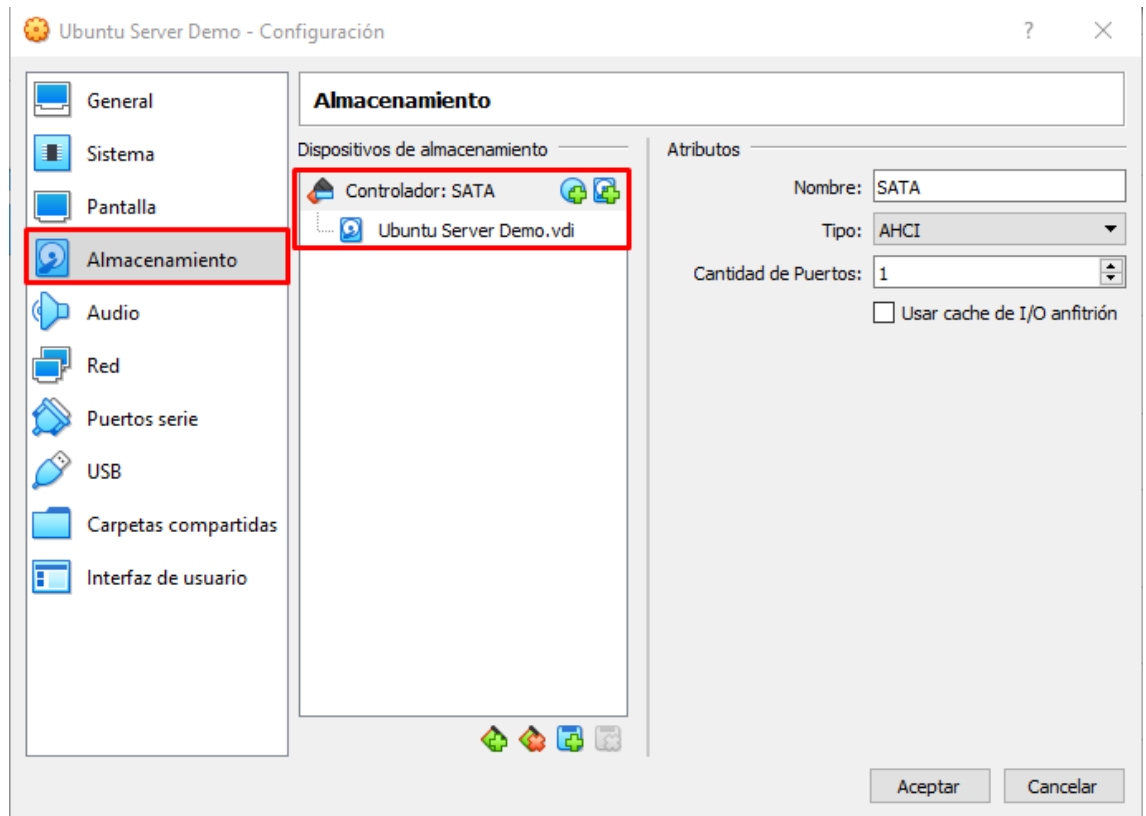
Una vez que tengamos la iso, vamos a instalarla en nuestra máquina virtual en la que antes deberemos hacer algunas configuraciones.



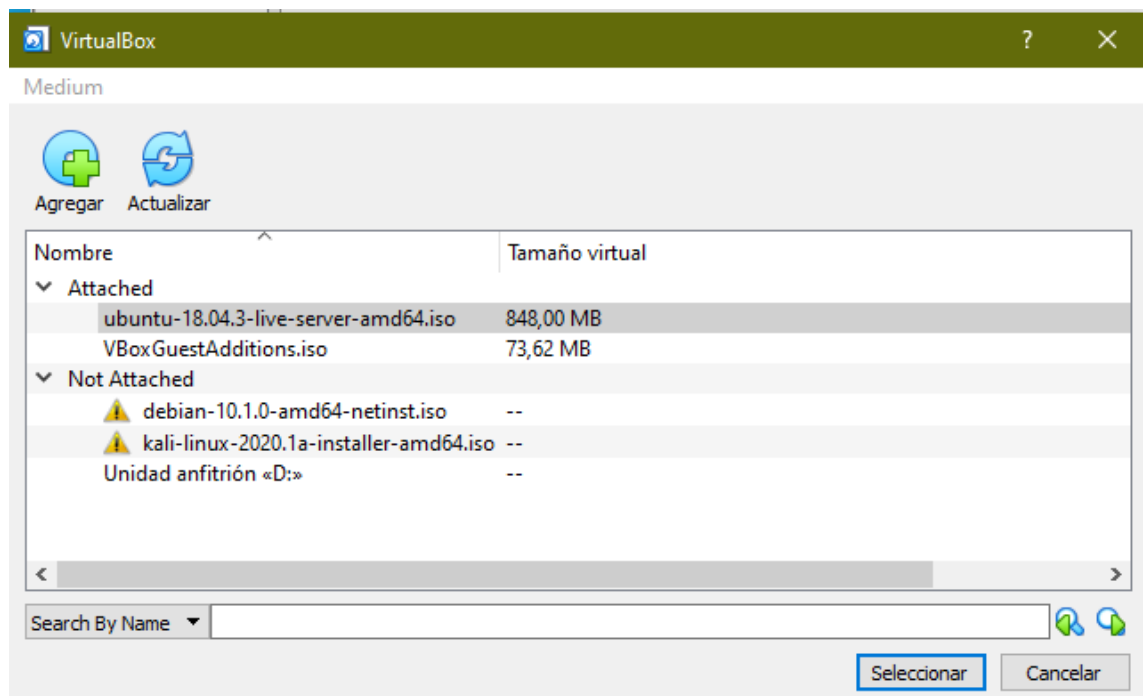
Una vez estemos en el apartado de configuración, hacemos click en sistema y dejaremos las opciones como se muestra en la siguiente imagen.



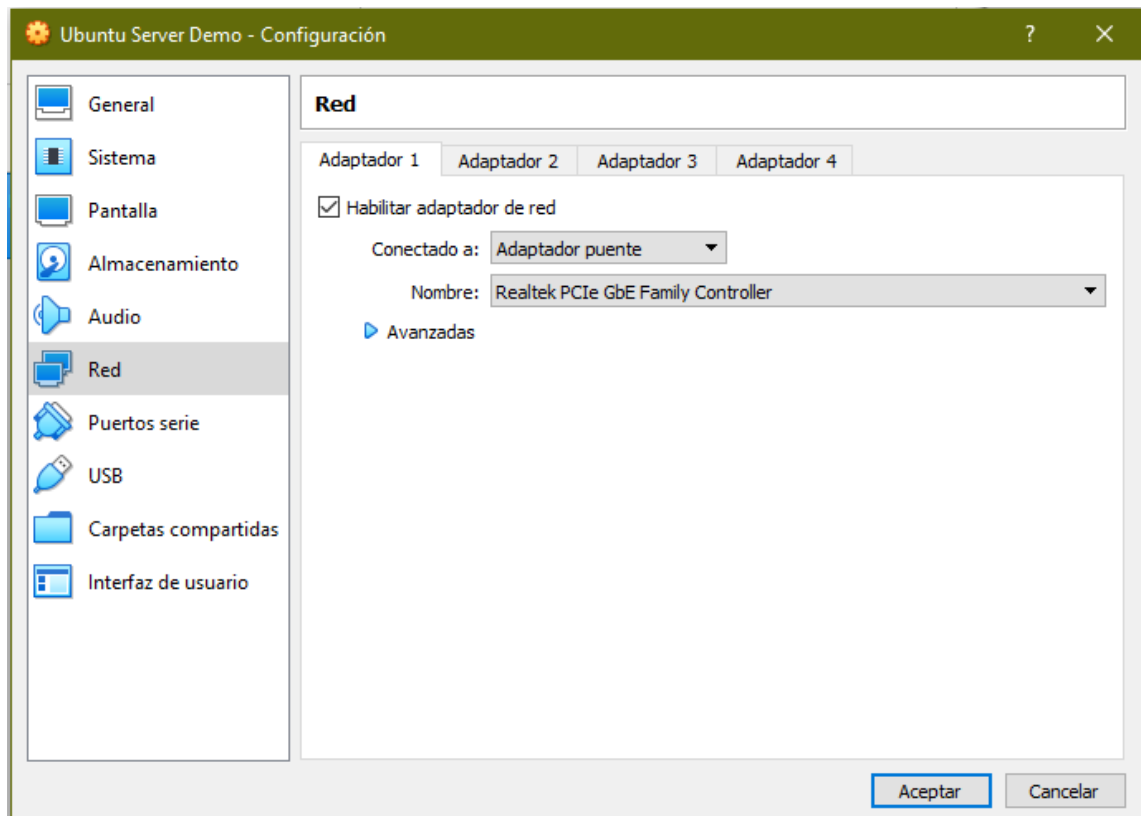
En Almacenamiento lo dejaremos así



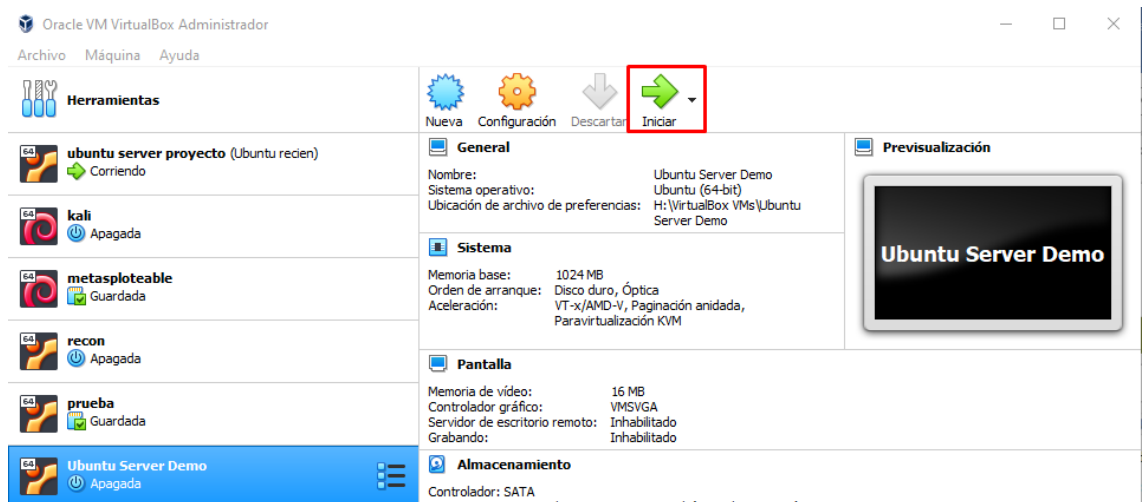
Añadimos la iso que hemos descargado previamente haciendo click en el icono de CD.



Y por último en el apartado de Red indicaremos que la conexión será Adaptador Puente, lo que permite que la máquina virtual esté en la misma red que la máquina física.

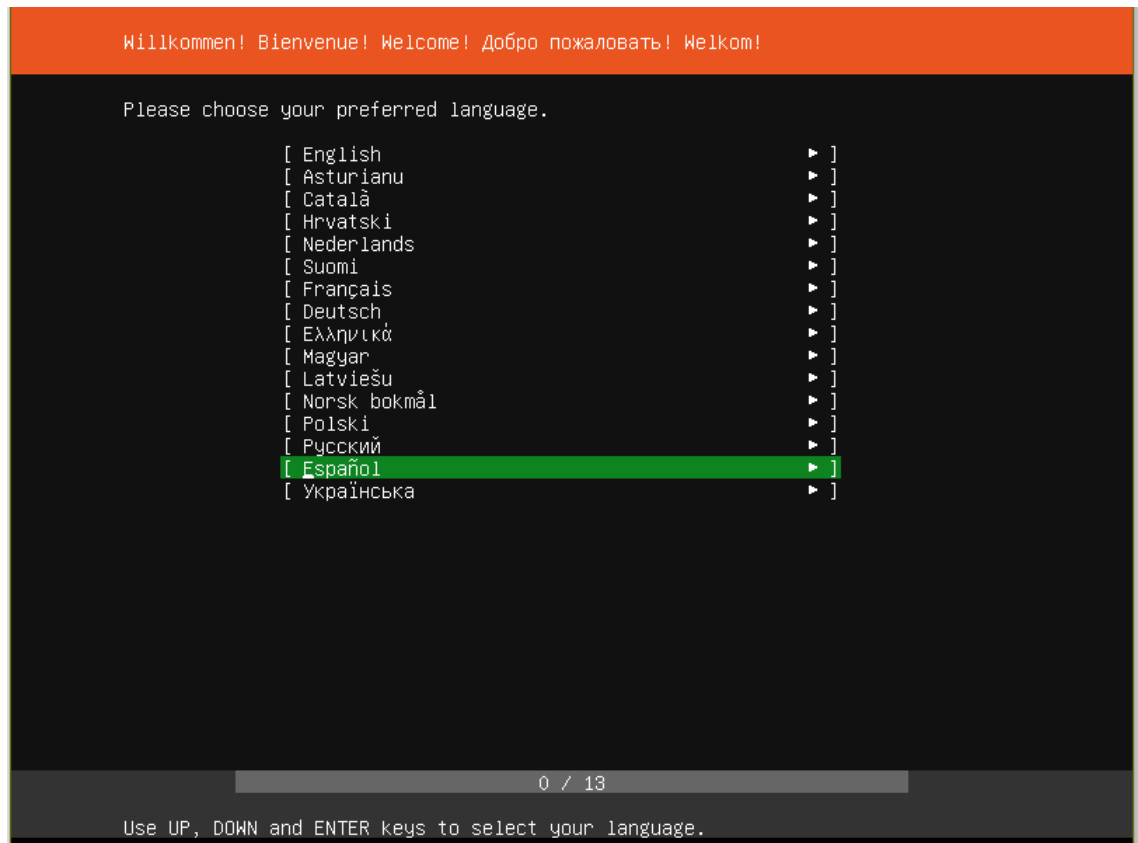


Una vez que tenemos ya configurado la máquina virtual, vamos a iniciarla para la instalación del sistema operativo.

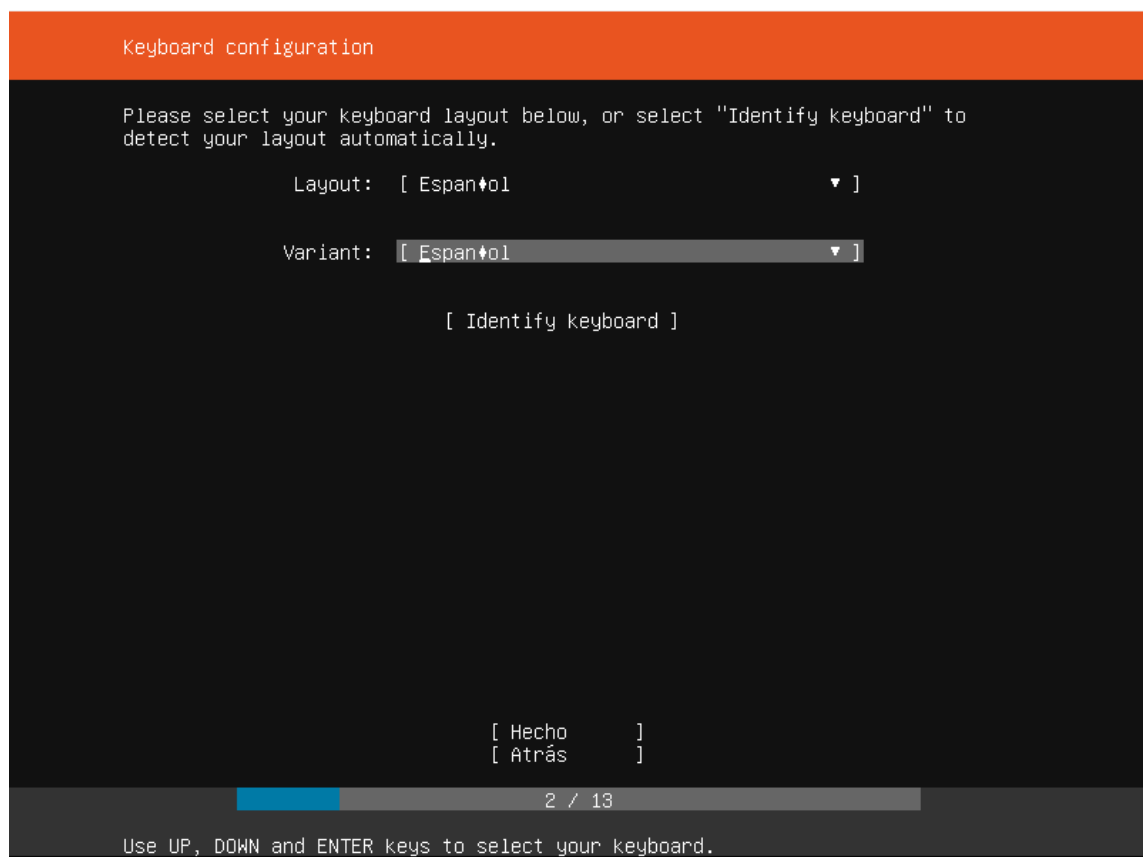


Lo primero que nos aparecerá será el menú para elegir el idioma, seleccionamos nuestro idioma.

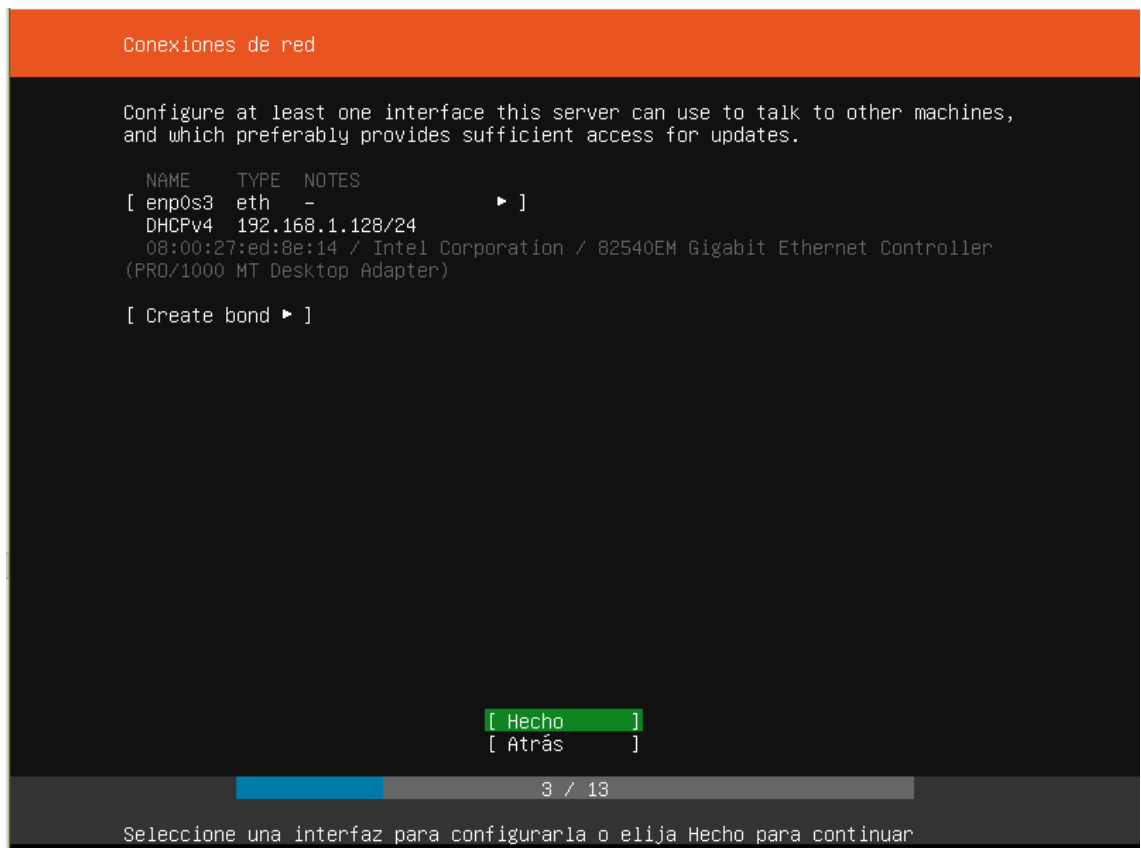




Elegimos el idioma de nuestro teclado.



En tipo de red, como hemos elegido adaptador puente se nos asignará una IP en la misma red que la máquina física.



Como no disponemos de ningún servidor proxy, le damos a siguiente.

Configure proxy

If this system requires a proxy to connect to the internet, enter its details here.

Proxy address:

If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of "http://[[user] [:pass]@]host[:port]/".

[ Hecho ]

[ Atrás ]

4 / 13

La configuración de los mirrors de Ubuntu la dejamos por defecto.

Configure Ubuntu archive mirror

If you use an alternative mirror for Ubuntu, enter its details here.

Mirror address:

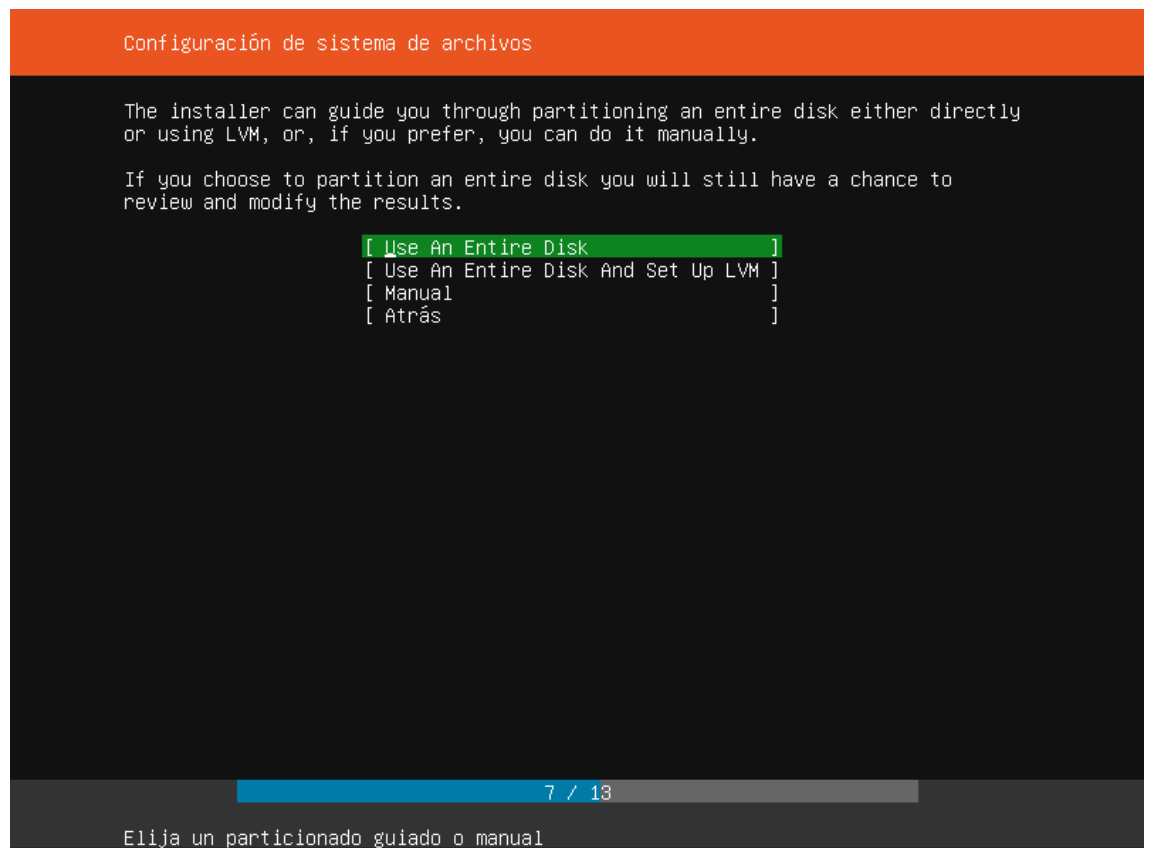
You may provide an archive mirror that will be used instead of the default.

[ Hecho ]

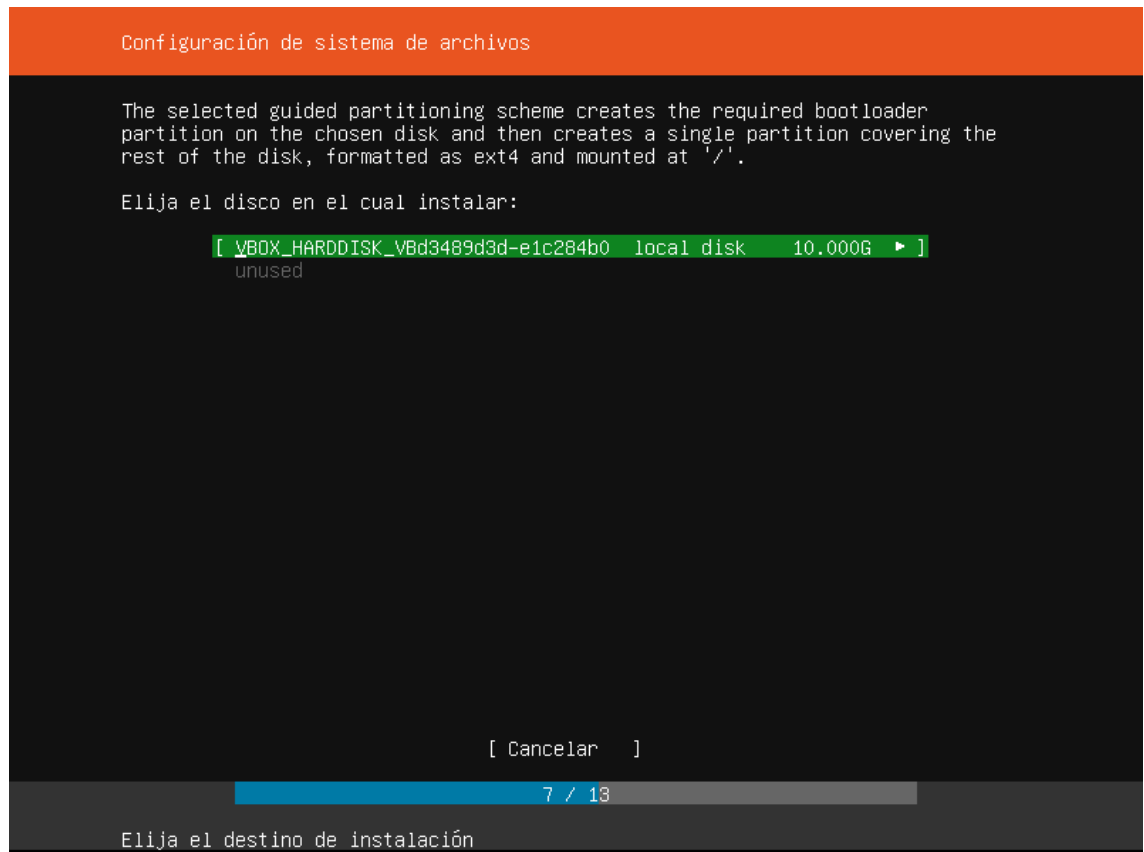
[ Atrás ]

5 / 13

A la hora de particionar el disco vamos a elegir usar el disco entero.



Seleccionamos el disco donde se va a realizar las particiones.



Una vez que veamos las particiones que nos va a crear aceptamos y confirmamos. Ahora debemos de incluir los datos del usuario para poder iniciar sesión una vez finalizada la instalación.

Configuración de perfil

Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name:

Your server's name:   
The name it uses when it talks to other computers.

Pick a username:

Choose a password:

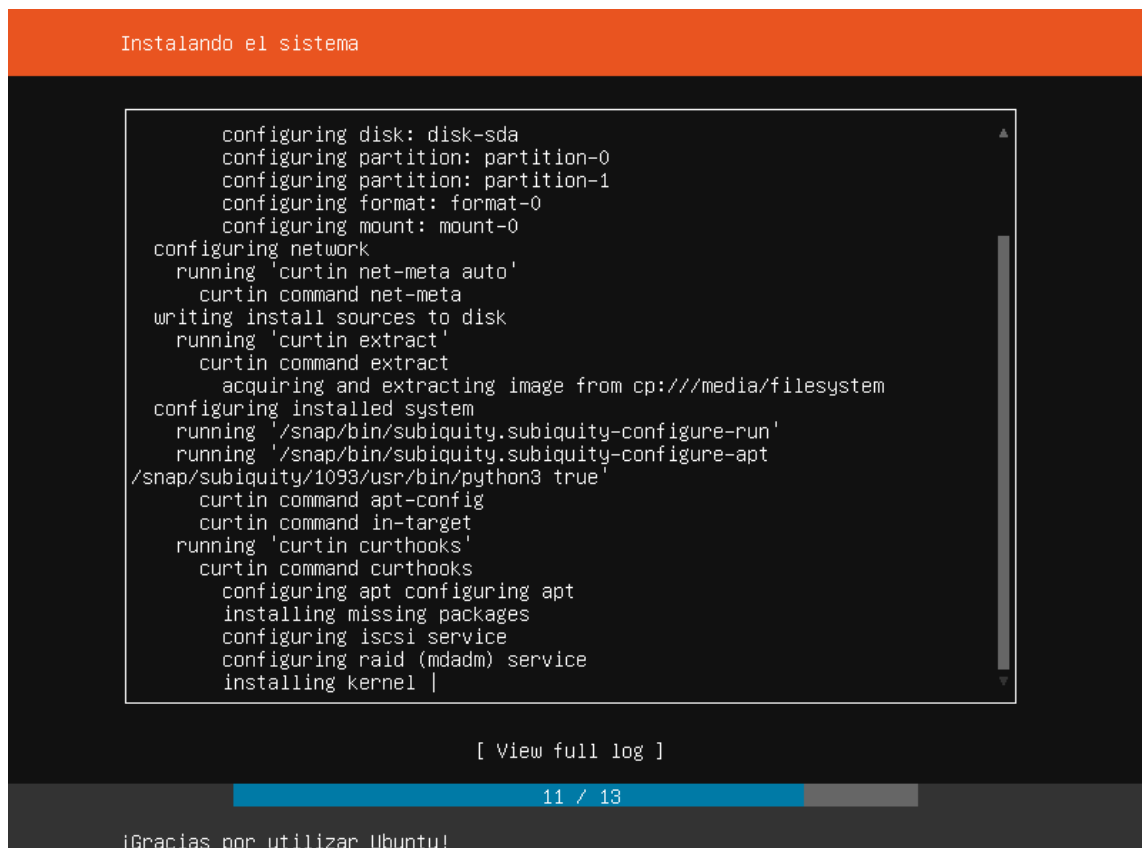
Confirm your password:

[ Hecho ]

7 / 13

Install in progress: installing kernel |

Hacemos click en hecho sin necesidad de modificar ninguna opción hasta que lleguemos a esta imagen.



Ya se está instalado Ubuntu 18.04.

Una vez haya finalizado la instalación del sistema operativo y hayamos iniciado sesión, vamos a proceder a instalar **FreeRadius**, que nos permitirá que nuestro servidor actúe como un servidor RADIUS.

Para ello vamos a ejecutar los siguientes comandos

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Con esto nos aseguramos que el equipo esté actualizado.  
Vamos a instalar **FreeRadius** con el siguiente comando.

```
sudo apt-get install freeradius
```

Una vez se haya instalado comprobamos que el servicio está activo.

```
jorge@tfc:~$ sudo systemctl status freeradius
● freeradius.service - FreeRADIUS multi-protocol policy server
   Loaded: loaded (/lib/systemd/system/freeradius.service; disabled; vendor preset: enabled)
   Active: active (running) since Sun 2020-04-19 10:12:22 UTC; 1 day 1h ago
     Docs: man:radiusd(8)
           man:radiusd.conf(5)
           http://wiki.freeradius.org/
           http://networkradius.com/doc/
  Main PID: 2812 (freeradius)
    Tasks: 6 (limit: 1108)
   CGroup: /system.slice/freeradius.service
           └─2812 /usr/sbin/freeradius
```

### 6.1.2. Configuración del punto de acceso.

Una vez tengamos instalado **FreeRadius**, vamos a configurar el punto de acceso, para ello debemos conectar nuestro ordenador al router e introducir en el navegador la ip del router. Para averiguar dicha ip debemos abrir un cmd y ejecutar el siguiente comando.

```
C:\Users\Jorge>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet 4:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de Ethernet Ethernet 2:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de Ethernet Ethernet 3:

    Sufijo DNS específico para la conexión. . :
    Vínculo: dirección IPv6 local. . . : fe80::903e:a8b9:c357:67ae%10
    Dirección IPv4. . . . . : 192.168.56.1
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . :

Adaptador de LAN inalámbrica Conexión de área local* 1:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 2:

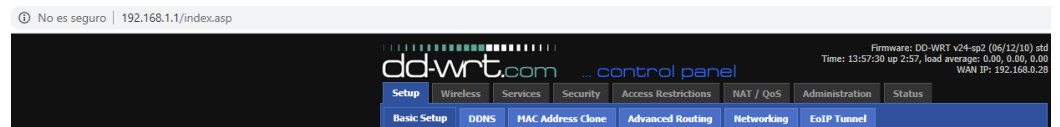
    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de Ethernet Ethernet:

    Sufijo DNS específico para la conexión. . :
    Vínculo: dirección IPv6 local. . . : fe80::74c9:8cc0:daef:f4d3%20
    Dirección IPv4. . . . . : 192.168.1.133
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . : 192.168.1.1
```



Ahora que hemos averiguado la IP la introduciremos en el navegador para entrar a la configuración del router.



Aunque cada router sea distinto se suele seguir el mismo procedimiento. Lo primero será deshabilitar el DHCP, para ello nos vamos a la pestaña

## Setup → Basic Setup

The image shows the "Basic Setup" page in the dd-wrt control panel. The top navigation bar is the same as in the previous screenshot. Below it, the "Basic Setup" sub-tab is selected. The page is divided into several sections. The "WAN Setup" section has a "WAN Connection Type" dropdown set to "Automatic Configuration - DHCP" and an "STP" section with "Enable" and "Disable" radio buttons, where "Disable" is selected. The "Optional Settings" section includes fields for "Router Name" (DD-WRT), "Host Name", "Domain Name", and an "MTU" dropdown set to "Auto" with a value of "1500". The "Network Setup" section has a "Router IP" section with input fields for "Local IP Address" (192, 168, 1, 1), "Subnet Mask" (255, 255, 255, 0), "Gateway" (0, 0, 0, 0), and "Local DNS" (0, 0, 0, 0). Below this is the "Network Address Server Settings (DHCP)" section, which includes a "DHCP Type" dropdown set to "DHCP Server", a "DHCP Server" section with "Enable" and "Disable" radio buttons ( "Disable" is selected), a "Start IP Address" field (192.168.1.100), a "Maximum DHCP Users" field (50), a "Client Lease Time" field (1440 minutes), three "Static DNS" fields (all 0.0.0.0), a "WINS" field (0.0.0.0), and three checkboxes: "Use DNSMasq for DHCP" (checked), "Use DNSMasq for DNS" (checked), and "DHCP-Authoritative" (checked). On the right side of the page, there is a "Help" section with a "more..." link. It contains several informational blocks: "Automatic Configuration - DHCP:" (This setting is most commonly used by Cable operators.), "Host Name:" (Enter the host name provided by your ISP.), "Domain Name:" (Enter the domain name provided by your ISP.), "Local IP Address:" (This is the address of the router.), "Subnet Mask:" (This is the subnet mask of the router.), "DHCP Server:" (Allows the router to manage your IP addresses.), "Start IP Address:" (The address you would like to start with.), "Maximum DHCP Users:" (You may limit the number of addresses your router hands out. 0 means only predefined static leases will be handed out.), and "Time Settings:" (Choose the time zone you are in and Summer Time (DST) period. The router can use local time or UTC time.).

Una vez que hemos deshabilitado DHCP vamos a configurar el router para que el método de autenticación sea a través de un servidor RADIUS, para ello nos dirigimos a la pestaña **Wireless** → **Wireless Security** modificaremos las opciones como las de la imagen a continuación.

The screenshot shows the 'Wireless Security w10' configuration page. The 'Security Mode' is set to 'WPA2 Enterprise'. The 'WPA Algorithms' are set to 'AES'. The 'Radius Auth Server Address' is '192.168.1.112'. The 'Radius Auth Server Port' is '1812' (Default: 1812). The 'Radius Auth Shared Secret' is masked with dots. The 'Key Renewal Interval (in seconds)' is '3600'. There are 'Save' and 'Apply Settings' buttons at the bottom. A 'Help' section on the right states: 'Security Mode: You may choose from Disable, WEP, WPA Personal, WPA Enterprise, or RADIUS. All devices on your network must use the same security mode.'

En **Security Mode** elegimos WPA2 Enterprise que es la que nos va a permitir el método de acceso mediante servidor RADIUS.

En **WPA Algorithms** elegimos AES, ya que permite claves de cifrado de hasta 256 bits.

En **Radius Auth Server Address** tenemos que especificar la ip de nuestro servidor RADIUS, al ser Ubuntu habrá que ejecutar el siguiente comando para averiguar la IP "ip address show".

En **Radius Auth Shared Secret** tendremos que escribir una clave que luego necesitaremos a la hora de configurar **FreeRadius**. Este apartado es importante ya que si ambas claves no coinciden no podríamos conectarnos a la red.

Todo lo demás lo dejamos por defecto.

## 6.2. Autenticación de clientes inalámbricos por Radius.

### 6.2.1. Edición del archivo clients.conf

Lo primero a la hora de configurar nuestro servidor RADIUS será añadir el/los clientes que van a usar nuestro servidor para la autenticación de los usuarios.

Para ello nos dirigimos al directorio donde se encuentran los archivos de configuración.

```
jorge@tfc:~$ cd /etc/freeradius/3.0/
jorge@tfc:/etc/freeradius/3.0$ ll
total 152
drwxr-xr-x 9 freerad freerad 4096 abr 20 11:24 ./
drwxr-s--x 3 freerad freerad 4096 mar 16 10:51 ../
drwxr-xr-x 2 freerad freerad 4096 mar 16 10:51 certs/
-rw-r----- 1 freerad freerad 7560 abr 19 09:53 clients.conf
-rw-r----- 1 freerad freerad 1440 abr 17 2019 dictionary
-rw-r----- 1 freerad freerad 2661 abr 17 2019 experimental.conf
lrwxrwxrwx 1 freerad freerad 28 abr 17 2019 hints -> mods-config/preprocess/hints
lrwxrwxrwx 1 freerad freerad 33 abr 17 2019 huntgroups -> mods-config/preprocess/huntgroups
drwxr-xr-x 2 freerad freerad 4096 mar 16 10:51 mods-available/
drwxr-xr-x 9 freerad freerad 4096 mar 16 10:51 mods-config/
drwxr-xr-x 2 freerad freerad 4096 mar 16 10:51 mods-enabled/
-rw-r----- 1 freerad freerad 52 abr 17 2019 panic.gdb
drwxr-xr-x 2 freerad freerad 4096 mar 16 10:51 policy.d/
-rw-r----- 1 freerad freerad 28361 abr 17 2019 proxy.conf
-rw-r----- 1 freerad freerad 26897 abr 17 2019 radiusd.conf
-rw-r----- 1 freerad freerad 20807 abr 17 2019 README.rst
drwxr-xr-x 2 freerad freerad 4096 mar 16 10:51 sites-available/
drwxr-xr-x 2 freerad freerad 4096 mar 16 10:51 sites-enabled/
-rw-r----- 1 freerad freerad 3470 abr 17 2019 templates.conf
-rw-r----- 1 freerad freerad 8536 abr 17 2019 trigger.conf
lrwxrwxrwx 1 freerad freerad 27 abr 17 2019 users -> mods-config/files/authorize
jorge@tfc:/etc/freeradius/3.0$
```

Editamos el archivo con **nano** y añadimos el cliente de la siguiente forma.

```
# Cliente punto de acceso
client PA {
    # Dirección IP del cliente (router)
    ipaddr = 192.168.1.1
    # Clave que tiene que coincidir con la "shared key" en la configuración del router
    secret = proyecto
}
```

En los archivos de configuración es muy importante la correcta escritura ya que un simple error de sintaxis hará fallar el servicio a la hora de aplicar los cambios que hemos hecho.

Una vez tengamos el cliente configurado, guardamos el archivo y reiniciamos el servicio.

```
jorge@tfc:/etc/freeradius/3.0$ sudo systemctl restart freeradius
jorge@tfc:/etc/freeradius/3.0$
```

Si el servicio se reinicia sin ningún error, la configuración que hemos añadido se ha aplicado correctamente.

### 6.2.2. Configuración de los usuarios.

A continuación, luego de haber especificado el cliente que va a usar RADIUS, deberemos crear los usuarios para que se puedan conectar. Para ello vamos a crearlos en la base de datos propia de **FreeRadius** que se encuentra en el mismo directorio al que nos dirigimos anteriormente.

```
jorge@tfc:/etc/freeradius/3.0$ ll
total 152
drwxr-xr-x 9 freerad freerad 4096 abr 20 11:57 ./
drwxr-s--x 3 freerad freerad 4096 mar 16 10:51 ../
drwxr-xr-x 2 freerad freerad 4096 mar 16 10:51 certs/
-rw-r----- 1 freerad freerad 7682 abr 20 11:57 clients.conf
-rw-r----- 1 freerad freerad 1440 abr 17 2019 dictionary
-rw-r----- 1 freerad freerad 2661 abr 17 2019 experimental.conf
lrwxrwxrwx 1 freerad freerad 28 abr 17 2019 hints -> mods-config/preprocess/hints
lrwxrwxrwx 1 freerad freerad 33 abr 17 2019 huntgroups -> mods-config/preprocess/huntgroups
drwxr-xr-x 2 freerad freerad 4096 mar 16 10:51 mods-available/
drwxr-xr-x 9 freerad freerad 4096 mar 16 10:51 mods-config/
drwxr-xr-x 2 freerad freerad 4096 mar 16 10:51 mods-enabled/
-rw-r----- 1 freerad freerad 52 abr 17 2019 panic.gdb
drwxr-xr-x 2 freerad freerad 4096 mar 16 10:51 policy.d/
-rw-r----- 1 freerad freerad 28361 abr 17 2019 proxy.conf
-rw-r----- 1 freerad freerad 26897 abr 17 2019 radiusd.conf
-rw-r----- 1 freerad freerad 20807 abr 17 2019 README.rst
drwxr-xr-x 2 freerad freerad 4096 mar 16 10:51 sites-available/
drwxr-xr-x 2 freerad freerad 4096 mar 16 10:51 sites-enabled/
-rw-r----- 1 freerad freerad 3470 abr 17 2019 templates.conf
-rw-r----- 1 freerad freerad 8536 abr 17 2019 trigger.conf
lrwxrwxrwx 1 freerad freerad 27 abr 17 2019 users -> mods-config/files/authorize
jorge@tfc:/etc/freeradius/3.0$
```

Volvemos a editar con **nano** el archivo **users**. Las opciones para crear un usuario son variadas y permiten muchas configuraciones, en este caso vamos a crear un usuario que pueda conectarse mediante RADIUS y que al conseguirlo tenga acceso a internet.

Partiendo del usuario de ejemplo “bob” vamos a añadir las siguientes líneas.

```
bob      Cleartext-Password := "hello"
        Reply-Message := "Hello, %{User-Name}",
        Framed-IP-Address = 192.168.1.10,
        Framed-IP-Netmask = 255.255.255.0,
        Framed-Routing = Broadcast-Listen
```

Como se ha mencionado anteriormente la nomenclatura es muy importante y deberemos prestar atención.

“Bob” es el nombre del usuario y seguido de este con la directiva **Cleartext-Password** := establecemos la contraseña del usuario que estemos definiendo.

**Reply-Message** es un mensaje que aparecerá al iniciar sesión exitosamente.

Para que un dispositivo tenga acceso a la red, éste debe tener una dirección IP, en este caso se la vamos a asignar de forma estática ya que en apartados anteriores hemos deshabilitado DHCP. Como tenemos una máscara de red de 24 bits, nos sobran 8 bits para la parte de host, lo que nos da un total de 254 posibles clientes, contando con el router (192.168.1.1) y la dirección IP del servidor RADIUS (192.168.1.112).

Viendo lo mencionado anteriormente deberemos usar la directiva **Framed-IP-Address** para establecer la dirección IP cuando el usuario acceda a la red. Cabe decir que como en una red no puede haber dos direcciones IP iguales, el usuario solo podrá tener acceso a un dispositivo.

Con las directivas **Framed-IP-Netmask** y **Framed-Routing** las dejamos como en la imagen anterior.

Una vez hayamos añadido el usuario, guardamos el archivo de configuración y reiniciamos el servicio para aplicar los cambios.

Si no hemos tenido ningún error y el servicio está activo, ya estaría todo configurado para que el usuario intentara conectarse a la red.

Para comprobar que el usuario está bien configurado usaremos el comando **radtest**, con el que podemos saber que usuario puede tener acceso si está bien configurado.

```
jorge@tfc:/etc/freeradius/3.0$ radtest bob hello localhost 0 testing123
Sent Access-Request Id 158 from 0.0.0.0:42986 to 127.0.0.1:1812 length 73
  User-Name = "bob"
  User-Password = "hello"
  NAS-IP-Address = 127.0.0.1
  NAS-Port = 0
  Message-Authenticator = 0x00
  Cleartext-Password = "hello"
Received Access-Accept Id 158 from 127.0.0.1:1812 to 0.0.0.0:0 length 50
  Reply-Message = "Hello, bob"
  Framed-IP-Address = 192.168.1.10
  Framed-IP-Netmask = 255.255.255.0
  Framed-Routing = Broadcast-Listen
jorge@tfc:/etc/freeradius/3.0$ _
```

Si recibimos **Access-Accept** el usuario está bien configurado.

### 6.2.3. Conexión a la red.

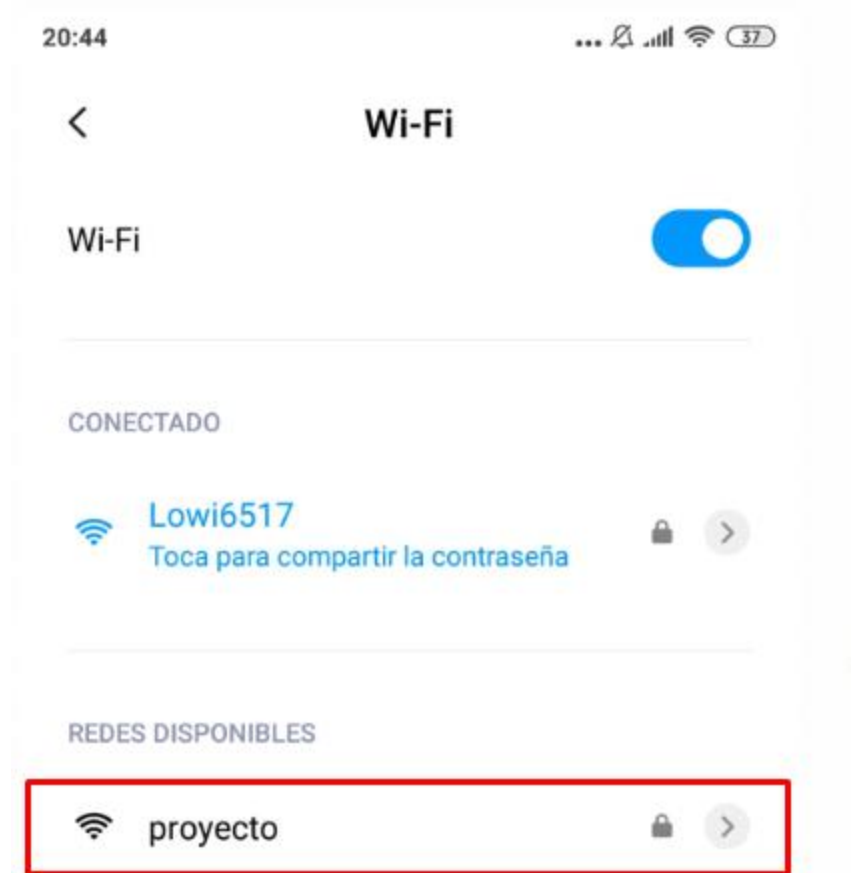
Lo primero saber cómo se llama nuestra red que se va a usar para autentificarnos con el servidor RADIUS, para ello nos dirigimos a la pestaña **Wireless → Basic Settings**.

Como se indica en la siguiente imagen, deberemos cambiar el SSID por el nombre que queramos y que será el que aparezca cuando los clientes busquen el nombre de la red.

The screenshot shows the Mikrotik WinBox interface for configuring wireless settings. The 'Basic Settings' tab is selected. Under 'Physical Interface ra0 - SSID [proyecto] HWAddr [F0:7D:68:8E:1D:40]', the 'Wireless Network Name (SSID)' field is highlighted with a red box and contains the text 'proyecto'. Other settings include 'Wireless Mode' set to 'AP', 'Wireless Network Mode' set to 'Mixed', 'Wireless Channel' set to '6 - 2.437 GHz', 'Channel Width' set to '20 MHz', 'Wireless SSID Broadcast' set to 'Enable', and 'Network Configuration' set to 'Bridged'. A 'Virtual Interfaces' section with an 'Add' button is visible below. At the bottom are 'Save', 'Apply Settings', and 'Cancel Changes' buttons.

Ya que sabemos cuál es nuestra SSID, vamos a usar un dispositivo inalámbrico para conectarnos a nuestra red.


En mi caso voy a usar mi móvil android para conectarme. Lo primero será buscar la red que hemos modificado en la configuración del router.



Una vez que tengamos ya la red vamos a conectarnos a ella, si tenemos bien configurado todo nos deberá aparecer un usuario y contraseña que están almacenados en el archivo **users** del servidor RADIUS.

**proyecto**

Identidad


Contraseña 

Opciones avanzadas **Conectar**

Introduciremos las credenciales del usuario “bob” que es el que incluimos en el archivo de configuración.

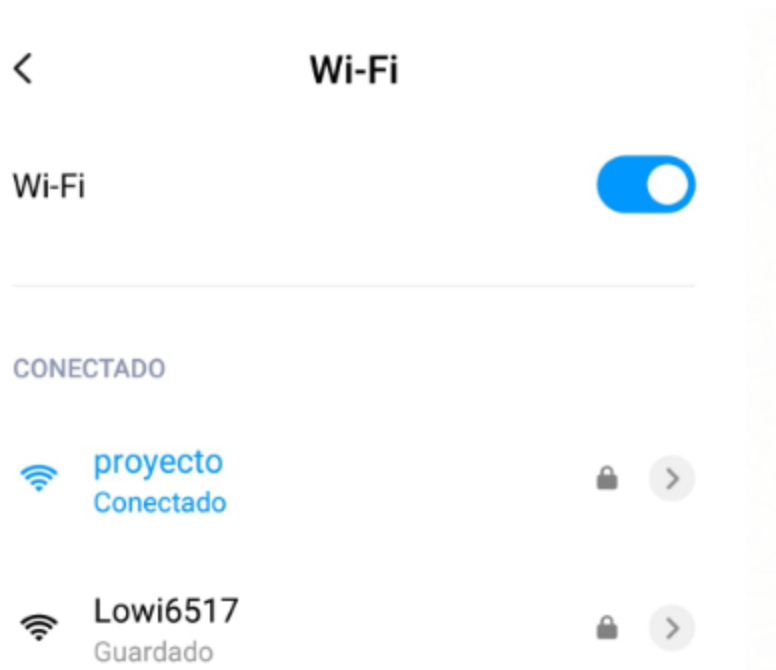
**proyecto**

bob

hello 

Opciones avanzadas **Conectar**

Al darle a conectar nos deberá dar una dirección IP y el usuario se habrá conectado correctamente a la red y tendrá acceso a Internet.



### 6.3. FreeRadius con MySQL

Una de las ventajas de usar FreeRadius es que permite usar una Base de Datos como MySQL o PostgreSQL, para almacenar las credenciales de los usuarios, esto nos permite incrementar la seguridad de las credenciales de nuestros clientes, ya que permite unas opciones de seguridad que FreeRadius no tiene. Asimismo, también se podría dar el caso de tener un servidor exclusivo de la base de datos, evitando así tener todo en el mismo servidor, mejorando el rendimiento de ambos servicios.

#### 6.3.1. Instalación de MySQL

Lo primero de todo será la instalación de la Base de Datos, en este caso se va a usar MySQL. Para ello ejecutaremos el siguiente comando:

```
root@tfc:~# apt-get install mysql-server
```

Una vez haya finalizado la instalación necesitaremos instalar el siguiente módulo:

```
root@tfc:~# apt-get install freeradius-mysql
```

Este módulo permitirá a FreeRadius comunicarse con nuestro servidor de MySQL mediante las siguientes configuraciones.

#### 6.3.2. Configuración de FreeRadius con MySQL

Por defecto, FreeRadius trabaja con su propia base de datos, pero como se ha explicado anteriormente no es la mejor opción.



A continuación, vamos a configurar tanto FreeRadius como MySQL.

Lo primero será crear una Base de Datos de MySQL, que es donde se van a almacenar las tablas con las credenciales de los usuarios.

Accedemos a MySQL con el usuario root y la clave de root:

```
root@tfc:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 59
Server version: 5.7.30-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, 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> _

mysql> create database fct_radius;
ERROR 1007 (HY000): Can't create database 'fct_radius'; database exists
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| fct_radius |
| mysql |
| performance_schema |
| sys |
| usuarios |
+-----+
6 rows in set (0.00 sec)
```

En mi caso ya tenía creada la base de datos y por eso MySQL nos dice que la base de datos ya existía.

A continuación, crearemos un usuario que tenga todos los permisos para manipular todas las tablas dentro de la base de datos “fct\_radius”.

```
mysql> create user 'radius'@'localhost' identified by 'radius';
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on fct_radius.* to 'radius'@'localhost';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

Una vez tengamos el usuario con los permisos necesarios, importaremos a nuestra base de datos un esquema sql, que nos creará las tablas necesarias para el almacenaje de las credenciales de los usuarios.

El archivo que contiene los esquemas a importar se encuentra en **/etc/freeradius/3.0/mods-config/sql/main/mysql/**

```
root@tfc:/etc/freeradius/3.0/mods-config/sql/main/mysql# ll
total 40
drwxr-xr-x 3 freerad freerad 4096 mar 16 10:51 ./
drwxr-xr-x 8 freerad freerad 4096 mar 16 10:51 ../
drwxr-xr-x 3 freerad freerad 4096 mar 16 10:51 extras/
-rw-r----- 1 freerad freerad 13738 abr 17 2019 queries.conf
-rw-r----- 1 freerad freerad 4800 abr 17 2019 schema.sql
-rw-r----- 1 freerad freerad 703 abr 17 2019 setup.sql
root@tfc:/etc/freeradius/3.0/mods-config/sql/main/mysql#
```

En este caso usaremos este archivo porque estamos usando una base de datos MySQL, esto dependerá de cuál usemos.

Si vemos el contenido del fichero

```
CREATE TABLE radacct (
  radacctid bigint(21) NOT NULL auto_increment,
  acctsessionid varchar(64) NOT NULL default '',
  acctuniqueid varchar(32) NOT NULL default '',
  username varchar(64) NOT NULL default '',
  realm varchar(64) default '',
  nasipaddress varchar(15) NOT NULL default '',
  nasportid varchar(15) default NULL,
  nasporttype varchar(32) default NULL,
  acctstarttime datetime NULL default NULL,
  acctupdatetime datetime NULL default NULL,
  acctstoptime datetime NULL default NULL,
  acctinterval int(12) default NULL,
  acctsessiontime int(12) unsigned default NULL,
  acctauthentic varchar(32) default NULL,
  connectinfo_start varchar(50) default NULL,
  connectinfo_stop varchar(50) default NULL,
  acctinputoctets bigint(20) default NULL,
  acctoutputoctets bigint(20) default NULL,
  calledstationid varchar(50) NOT NULL default '',
  --More--
```

Su contenido es la creación las tablas que necesitaremos en nuestra base de datos. Para importar el fichero a nuestra base de datos ejecutaremos el siguiente comando:

```
mysql -u root -p fct_radius < schema.sql
```

Si mostramos todas las tablas de nuestra base de datos nos debería aparecer lo siguiente:

```
mysql> show tables;
+-----+
| Tables_in_fct_radius |
+-----+
| nas                   |
| radacct               |
| radcheck              |
| radgroupcheck         |
| radgroupreply         |
| radpostauth           |
| radreply              |
| radusergroup          |
+-----+
8 rows in set (0.00 sec)
```

Ya tenemos creada nuestra base de datos con sus tablas correspondientes, ahora configuraremos FreeRadius para obtenga los datos nuestra base de datos.

Comprobaremos que en nuestro archivo **radiusd.conf** tenemos descomentada la línea **\$INCLUDE clients.conf** que es el fichero donde teníamos configurado la dirección ip de nuestro router para la autenticación por radius.

```
# CLIENTS CONFIGURATION
#
# Client configuration is defined in "clients.conf".
#
# The 'clients.conf' file contains all of the information from the old
# 'clients' and 'naslist' configuration files. We recommend that you
# do NOT use 'client's or 'naslist', although they are still
# supported.
#
# Anything listed in 'clients.conf' will take precedence over the
# information from the old-style configuration files.
#
$INCLUDE clients.conf
```

Editaremos el archivo donde se configuran los datos del servidor de la base de datos, para ellos modificaremos el archivo **/etc/freeradius/3.0/mods-available/sql**, en concreto las directivas:

**Drivers**→ Debemos elegir la correspondiente a nuestra base de datos, en nuestro caso es MySQL.

```
driver = "r1m_sql_mysql"
```

**Dialect**→ Especificamos el lenguaje de nuestra base de datos.

```
dialect = "mysql"
```

En la información de conexión tendremos que incluir los datos de nuestro servidor MySQL, dado que en este caso va a ser el mismo servidor que Radius, mi configuración será la siguiente:

```
# Connection info:
#
server = "localhost"
port = 3306
login = "radius"
password = "radius"
```

**Login** y **password** hacen referencia al usuario que se va a conectar a la base de datos y al que le dimos permiso.

Y por último en este archivo, tendremos que indicar la base de datos que creamos y sobre la que importamos las tablas.

```
radius_db = "fct_radius"
```

Guardamos los cambios. Ahora necesitamos activar este módulo, para ello utilizaremos un enlace simbólico, lo primero será desplazarse hasta la carpeta **mods-enabled**

```
root@tfc:/etc/freeradius/3.0/mods-available# cd ../mods-enabled/
root@tfc:/etc/freeradius/3.0/mods-enabled#
```

Desde este directorio ejecutaremos el siguiente comando:

```
ln -s ../mods-available/sql sql
```

Si comprobamos ahora el directorio **mods-enabled**:

```
root@tfc:/etc/freeradius/3.0/mods-enabled# ll
total 8
drwxr-xr-x 2 freerad freerad 4096 may 14 09:53 ./
drwxr-xr-x 9 freerad freerad 4096 may 14 22:18 ../
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 always -> ../mods-available/always
lrwxrwxrwx 1 freerad freerad 29 mar 16 10:51 attr_filter -> ../mods-available/attr_filter
lrwxrwxrwx 1 freerad freerad 27 mar 16 10:51 cache_eap -> ../mods-available/cache_eap
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 chap -> ../mods-available/chap
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 detail -> ../mods-available/detail
lrwxrwxrwx 1 freerad freerad 28 mar 16 10:51 detail.log -> ../mods-available/detail.log
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 digest -> ../mods-available/digest
lrwxrwxrwx 1 freerad freerad 33 mar 16 10:51 dynamic_clients -> ../mods-available/dynamic_clients
lrwxrwxrwx 1 freerad freerad 21 mar 16 10:51 eap -> ../mods-available/eap
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 echo -> ../mods-available/echo
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 exec -> ../mods-available/exec
lrwxrwxrwx 1 freerad freerad 28 mar 16 10:51 expiration -> ../mods-available/expiration
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 expr -> ../mods-available/expr
lrwxrwxrwx 1 freerad freerad 23 mar 16 10:51 files -> ../mods-available/files
lrwxrwxrwx 1 freerad freerad 25 mar 16 10:51 lineelog -> ../mods-available/lineelog
lrwxrwxrwx 1 freerad freerad 27 mar 16 10:51 logintime -> ../mods-available/logintime
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 mschap -> ../mods-available/mschap
lrwxrwxrwx 1 freerad freerad 27 mar 16 10:51 ntlm_auth -> ../mods-available/ntlm_auth
lrwxrwxrwx 1 freerad freerad 21 mar 16 10:51 pap -> ../mods-available/pap
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 passwd -> ../mods-available/passwd
lrwxrwxrwx 1 freerad freerad 28 mar 16 10:51 preprocess -> ../mods-available/preprocess
lrwxrwxrwx 1 freerad freerad 25 mar 16 10:51 radutmp -> ../mods-available/radutmp
lrwxrwxrwx 1 freerad freerad 23 mar 16 10:51 realm -> ../mods-available/realm
lrwxrwxrwx 1 freerad freerad 27 mar 16 10:51 replicate -> ../mods-available/replicate
lrwxrwxrwx 1 freerad freerad 21 mar 16 10:51 soh -> ../mods-available/soh
lrwxrwxrwx 1 freerad freerad 21 may 14 09:53 sql -> ../mods-available/sql
lrwxrwxrwx 1 freerad freerad 26 mar 16 10:51 sradutmp -> ../mods-available/sradutmp
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 unix -> ../mods-available/unix
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 unpack -> ../mods-available/unpack
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 utf8 -> ../mods-available/utf8
root@tfc:/etc/freeradius/3.0/mods-enabled#
```

Si necesitamos cambiar el usuario y grupo ejecutaremos el siguiente comando:

```
chown -h freerad: sql
```

Con esto ya tendremos el módulo activado, pero los cambios no se ejecutarán hasta que el servicio no haya sido reiniciado, pero antes de ello vamos a seguir con las configuraciones de varios archivos.

En el directorio **sites-available** modificaremos los siguientes archivos.

```
root@tfc:/etc/freeradius/3.0/sites-available# ll
total 192
drwxr-xr-x 2 freerad freerad 4096 may 14 10:14 ./
drwxr-xr-x 9 freerad freerad 4096 may 14 22:18 ../
-rw-r----- 1 freerad freerad 2167 abr 17 2019 abfab-tls
-rw-r----- 1 freerad freerad 5114 abr 17 2019 abfab-tr-idp
-rw-r----- 1 freerad freerad 5180 abr 17 2019 buffered-sql
-rw-r----- 1 freerad freerad 1359 abr 17 2019 challenge
-rw-r----- 1 freerad freerad 486 abr 17 2019 channel_bindings
-rw-r----- 1 freerad freerad 3599 abr 17 2019 check-eap-tls
-rw-r----- 1 freerad freerad 1334 abr 17 2019 coa
-rw-r----- 1 freerad freerad 2632 abr 17 2019 control-socket
-rw-r----- 1 freerad freerad 5765 abr 17 2019 copy-acct-to-home-server
-rw-r----- 1 freerad freerad 3466 abr 17 2019 decoupled-accounting
-rw-r----- 1 freerad freerad 28070 may 14 10:13 default
-rw-r----- 1 freerad freerad 9294 abr 17 2019 dhcp
-rw-r----- 1 freerad freerad 1033 abr 17 2019 dhcp.relay
-rw-r----- 1 freerad freerad 7091 abr 17 2019 dynamic-clients
-rw-r----- 1 freerad freerad 3382 abr 17 2019 example
-rw-r----- 1 freerad freerad 12216 may 14 10:14 inner-tunnel
-rw-r----- 1 freerad freerad 4943 abr 17 2019 originate-coa
-rw-r----- 1 freerad freerad 1026 abr 17 2019 proxy-inner-tunnel
-rw-r----- 1 freerad freerad 8543 abr 17 2019 README
-rw-r----- 1 freerad freerad 4718 abr 17 2019 robust-proxy-accounting
-rw-r----- 1 freerad freerad 820 abr 17 2019 soh
-rw-r----- 1 freerad freerad 4079 abr 17 2019 status
-rw-r----- 1 freerad freerad 15796 abr 17 2019 tls
-rw-r----- 1 freerad freerad 877 abr 17 2019 virtual.example.com
-rw-r----- 1 freerad freerad 2571 abr 17 2019 vmps
root@tfc:/etc/freeradius/3.0/sites-available#
```

En ambos archivos se van a realizar las mismas modificaciones, en la sección **authorize** vamos a descomentar las líneas de sql.

```
# Look in an SQL database. The schema of the database
# is meant to mirror the "users" file.
#
# See "Authorization Queries" in mods-available/sql
sql
```

Y realizamos lo mismo en la sección **accounting**.

```
# Log traffic to an SQL database.
#
# See "Accounting queries" in mods-available/sql
-sql
```

Si no se encuentra en alguno de los dos archivos la sección **accounting**, solo realizaremos el cambio en la que esté. También cabe añadir que si la línea de **ldap**, está descomentada, es recomendable comentarla.

También descomantaremos la siguiente línea.

```
# Set to 'yes' to read radius clients from the database ('nas' table)
# Clients will ONLY be read on server startup.
read_clients = yes

# Table to keep radius client info
client_table = "nas"
```

Con esto vamos a permitir que freeradius use los datos de la tabla **nas** para conectar con el servidor evitando así usar el archivo **clients.conf** que habíamos configurado anteriormente, aumentando así la seguridad de nuestro entorno.

```
mysql> select * from nas;
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | nasname | shortname | type | ports | secret | server | community | description |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 192.168.1.1 | router | other | NULL | proyecto | NULL | NULL | RADIUS Client |
+----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Vamos a reiniciar el servicio freeradius, para aplicar los cambios en los archivos de configuración. Si no saliese ningún error es que los cambios se han aplicado correctamente.

Procederemos a la inserción de las credenciales de un usuario ficticio, para ello, con la base de datos de freeradius, los usuarios se introducían en el archivo **users** con una sintaxis muy concreta, ahora con MySQL simplemente tendremos que introducir los datos en la tabla correspondiente.

Lo primero será acceder a MySQL, la tabla que almacena las credenciales es **radcheck**,

```
Tables_in_fct_radius
+-----+
| nas |
| radacct |
| radcheck |
| radgroupcheck |
| radgroupreply |
| radpostauth |
| radreply |
| radusergroup |
+-----+
3 rows in set (0.00 sec)
```

Si vemos las columnas veremos que tiene 4

```
mysql> show columns from radcheck;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id | int(11) unsigned | NO | PRI | NULL | auto_increment |
| username | varchar(64) | NO | MUL | | |
| attribute | varchar(64) | NO | | | |
| op | char(2) | NO | | == | |
| value | varchar(253) | NO | | | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Para introducir un usuario correctamente deberemos seguir la misma sintaxis que usa freeradius que es:

### **Username Cleartext-Password := Password**

Introduciremos en cada campo su valor correspondiente,

```
mysql> insert into radcheck values(null,"prueba1","Cleartext-Password",":=", "prueba1");
Query OK, 1 row affected (0.92 sec)
```

El primer dato es **null**, porque es un autoincrementable, por cada inserción ese campo sumará uno.

Comprobamos que los datos han sido insertados correctamente como en la siguiente imagen.

```
mysql> select * from radcheck where username="prueba1";
+-----+-----+-----+-----+-----+
| id | username | attribute          | op | value  |
+-----+-----+-----+-----+-----+
| 6 | prueba1 | Cleartext-Password | := | prueba1 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Para comprobar que freeradius usa MySQL para obtener las credenciales de los usuarios, ejecutaremos el comando **radtest** de la siguiente forma:

```
root@tfc:~# radtest prueba1 prueba1 localhost 10 testing123
Sent Access-Request Id 135 from 0.0.0.0:34567 to 127.0.0.1:1812 length 77
  User-Name = "prueba1"
  User-Password = "prueba1"
  NAS-IP-Address = 127.0.1.1
  NAS-Port = 10
  Message-Authenticator = 0x00
  Cleartext-Password = "prueba1"
Received Access-Accept Id 135 from 127.0.0.1:1812 to 0.0.0.0:0 length 20
root@tfc:~# _
```

Con esto, hemos comprobado que freeradius está usando MySQL, si obtenemos **Received Access-Accept**, el usuario tendría acceso a nuestra red a través de radius.

Pero para que pueda navegar por la red necesitamos especificar algunos parámetros como por ejemplo una dirección IP. Nuestro usuario va a obtener una dirección IP de forma estática, por lo que deberemos añadir los siguientes datos en las tablas de nuestra base de datos.

En la tabla radcheck debemos tener las credenciales del usuario para conectarse a nuestra red.

```
mysql> select * from radcheck;
```

id	username	attribute	op	value
1	usu1	usu1	:=	grupo1
2	jorge	Cleartext-Password	:=	proyecto
3	jorge2	Cleartext-Password	:=	jorge2
4	jorge3	Cleartext-Password	:=	jorge3
5	yolanda	Cleartext-Password	:=	yolanda
6	prueba1	Cleartext-Password	:=	prueba1
8	treilly	Cleartext-Password	:=	chipi

```
7 rows in set (0.00 sec)
```

Vamos a usar por ejemplo el usuario “jorge”. A continuación, vamos a especificar a qué grupo pertenece este usuario, para ello introduciremos los datos en la tabla **radusergroup**.

```
mysql> select * from radusergroup;
```

username	groupname	priority
jorge	estatico	1

```
1 row in set (0.00 sec)
```

El usuario “jorge” pertenece al grupo estático y vamos a especificar qué dirección IP tendrá el usuario al conectarse al introducir los datos correspondientes en la tabla **radreply**.

```
mysql> select * from radreply;
```

id	username	attribute	op	value
1	jorge	Framed-IP-Address	:=	192.168.1.131

```
1 row in set (0.00 sec)
```

Por último, debemos añadir los siguientes parámetros que afectan al grupo del usuario, en este caso **estático**.

```
mysql> select * from radgroupreply;
```

id	groupname	attribute	op	value
1	estatico	Framed-Protocol	:=	PPP
2	estatico	Service-Type	:=	Framed-User
3	estatico	Framed-Compression	:=	Van-Jacobson-TCP-IP

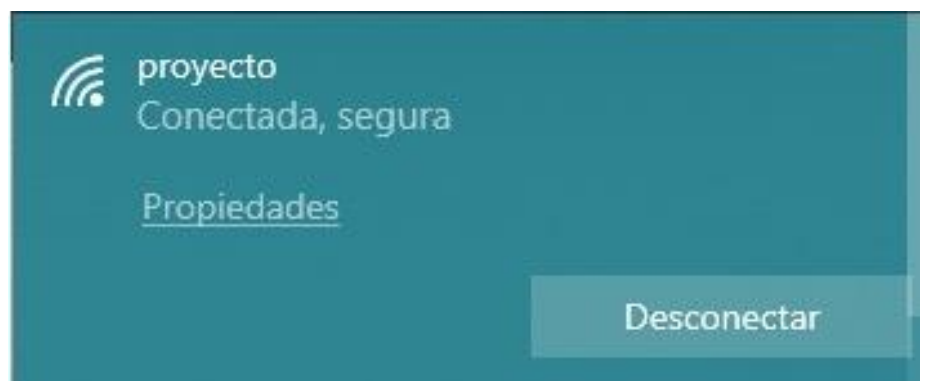
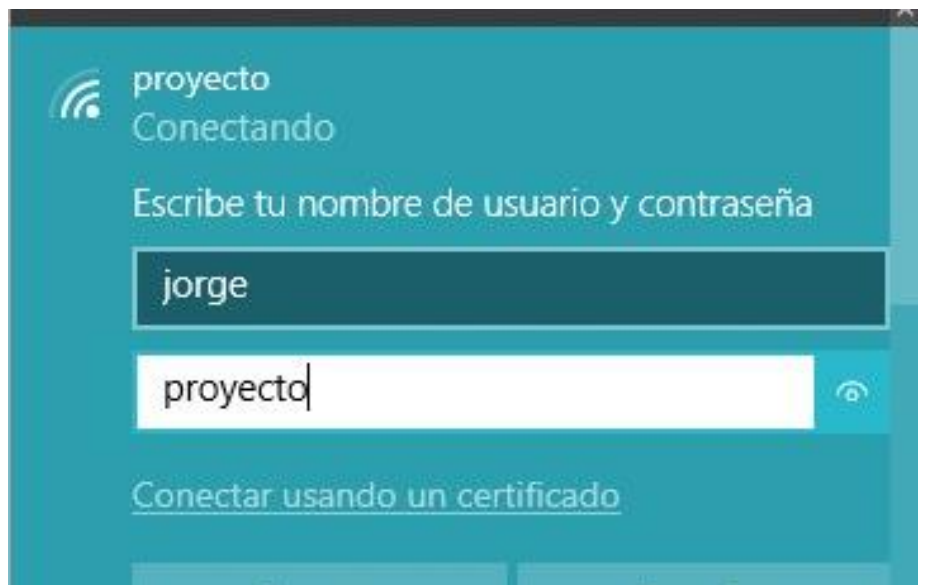
```
3 rows in set (0.00 sec)
```



Con esto, el usuario ya está listo para navegar por la red. Antes de nada, vamos a comprobar que el usuario está bien configurado.

```
root@tfc:/etc/freeradius/3.0# radtest jorge proyecto localhost 10 testing123
Sent Access-Request Id 76 from 0.0.0.0:41006 to 127.0.0.1:1812 length 75
  User-Name = "jorge"
  User-Password = "proyecto"
  NAS-IP-Address = 127.0.0.1
  NAS-Port = 10
  Message-Authenticator = 0x00
  Cleartext-Password = "proyecto"
Received Access-Accept Id 76 from 127.0.0.1:1812 to 0.0.0.0:0 length 44
  Framed-IP-Address = 192.168.1.131
  Framed-Protocol = PPP
  Service-Type = Framed-User
  Framed-Compression = Van-Jacobson-TCP-IP
root@tfc:/etc/freeradius/3.0#
```

Nos conectamos a la red **proyecto**, donde se pedirá usuario y contraseña. Una vez introducidas tendremos acceso a la red y podremos navegar por Internet.



```

C:\Users\Santiago>ping 192.168.1.1

Haciendo ping a 192.168.1.1 con 32 bytes de datos:
Respuesta desde 192.168.1.1: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.1.1: bytes=32 tiempo=2ms TTL=64
Respuesta desde 192.168.1.1: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.1.1: bytes=32 tiempo=16ms TTL=64

Estadísticas de ping para 192.168.1.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 1ms, Máximo = 16ms, Media = 5ms

C:\Users\Santiago>ping 8.88.8

Haciendo ping a 8.88.0.8 con 32 bytes de datos:
Tiempo de espera agotado para esta solicitud.

Estadísticas de ping para 8.88.0.8:
    Paquetes: enviados = 1, recibidos = 0, perdidos = 1
    (100% perdidos),
Control-C
^C
C:\Users\Santiago>ping 8.8.8.8

Haciendo ping a 8.8.8.8 con 32 bytes de datos:
Respuesta desde 8.8.8.8: bytes=32 tiempo=23ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=25ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=23ms TTL=52
Respuesta desde 8.8.8.8: bytes=32 tiempo=23ms TTL=52

Estadísticas de ping para 8.8.8.8:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 23ms, Máximo = 25ms, Media = 23ms

C:\Users\Santiago>

```

## 6.4. FreeRadius con OpenLDAP

### 6.4.1. Instalación y configuración de LDAP

Vamos a configurar FreeRadius para que los usuarios se puedan autenticar por LDAP.

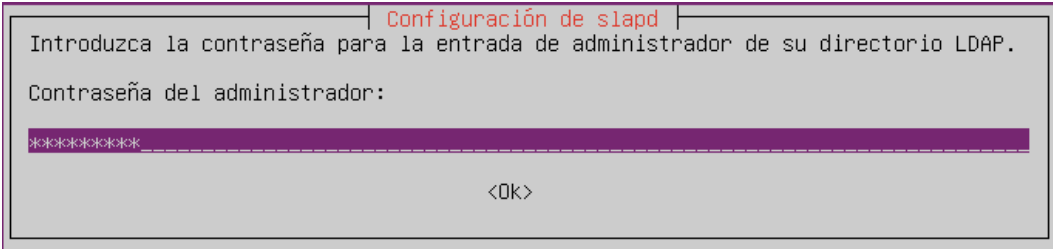
LDAP (Lightweight Directory Access Protocol), hace referencia a un protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido.

Al contrario que MySQL, deberemos habilitar el archivo de configuración **clients.conf**, con los datos del punto de acceso.

Para instalar LDAP deberemos ejecutar el siguiente comando:

```
root@tfc:/etc/freeradius/3.0# apt-get install slapd ldap-utils
```

Nos pedirá la contraseña que va a usar el administrador del directorio.



Configuración de slapd

Introduzca la contraseña para la entrada de administrador de su directorio LDAP.

Contraseña del administrador:

XXXXXXXXXX

<Ok>

Una vez que se haya instalado ejecutaremos el comando **slapcat** para ver las entradas que tendremos en nuestra base de datos ldap.

```
root@tfc:/etc/freeradius/3.0# slapcat
dn: dc=nodomain
objectClass: top
objectClass: dcObject
objectClass: organization
o: nodomain
dc: nodomain
structuralObjectClass: organization
entryUUID: 27716ff0-383f-103a-9cce-2bd6f720ce46
creatorsName: cn=admin,dc=nodomain
createTimestamp: 20200601103402Z
entryCSN: 20200601103402.005810Z#000000#000#000000
modifiersName: cn=admin,dc=nodomain
modifyTimestamp: 20200601103402Z

dn: cn=admin,dc=nodomain
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9c0xuN1V4em9sUmVBczlPbz1SdHR3RDBtdi9NUjJuOXk=
structuralObjectClass: organizationalRole
entryUUID: 27732ad4-383f-103a-9ccf-2bd6f720ce46
creatorsName: cn=admin,dc=nodomain
createTimestamp: 20200601103402Z
entryCSN: 20200601103402.017205Z#000000#000#000000
modifiersName: cn=admin,dc=nodomain
modifyTimestamp: 20200601103402Z

root@tfc:/etc/freeradius/3.0# _
```

Cada entrada empieza por el **Distinguish Name (dn)**, la primera entrada concierne a nuestro dominio, y el segundo al usuario **administrador** de ldap.

En nuestro caso, nos aparece **nodomain**, vamos a cambiarlo usando el siguiente comando:

```
root@tfc:/etc/freeradius/3.0# dpkg-reconfigure slapd
```

Cambiaremos nuestro nombre DNS,

Introduzca el nombre de dominio DNS:

jorge.proyecto

<Ok>

Configuración de slapd

Introduzca el nombre de la organización a utilizar en el DN base del directorio LDAP.

Nombre de la organización:

TFC

<Ok>

Dejamos las demás opciones por defecto, y volvemos a ejecutar el comando `slapcat`, para ver los cambios.

```
root@tfc:/etc/freeradius/3.0# slapcat
dn: dc=jorge,dc=proyecto
objectClass: top
objectClass: dcObject
objectClass: organization
o: TFC
dc: jorge
structuralObjectClass: organization
entryUUID: a3fb2cd6-3840-103a-8f03-83a840b24ebb
creatorsName: cn=admin,dc=jorge,dc=proyecto
createTimestamp: 20200601104440Z
entryCSN: 20200601104440.442583Z#000000#000#000000
modifiersName: cn=admin,dc=jorge,dc=proyecto
modifyTimestamp: 20200601104440Z

dn: cn=admin,dc=jorge,dc=proyecto
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9T1pYcDRERnc3QkpQbUhURUhWc1JNMzhNZFhM22JLa0Q=
structuralObjectClass: organizationalRole
entryUUID: a3fbd8f2-3840-103a-8f04-83a840b24ebb
creatorsName: cn=admin,dc=jorge,dc=proyecto
createTimestamp: 20200601104440Z
entryCSN: 20200601104440.447164Z#000000#000#000000
modifiersName: cn=admin,dc=jorge,dc=proyecto
modifyTimestamp: 20200601104440Z

root@tfc:/etc/freeradius/3.0#
```

Una vez que ya tenemos nuestro nombre de dominio bien configurado, vamos a crear un usuario en LDAP que usaremos más adelante para la autenticación por radius. Para ello deberemos crear un archivo de texto con la configuración del usuario:

```
GNU nano 2.9.3 usuarios_ldap.ldif
#Usuario ldap
dn: uid=usuario_ldap,dc=jorge,dc=proyecto
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: usuario_ldap
uid: usuario_ldap
uidNumber: 1052
gidNumber: 1052
homeDirectory: /home/usuario_ldap
loginShell: /bin/bash
userPassword: usuario_ldap_
```

A continuación, añadimos el usuario a nuestra base de datos LDAP con el siguiente comando:

```
root@tfc:~# ldapadd -x -W -D "cn=admin,dc=jorge,dc=proyecto" -f usuarios_ldap.ldif
Enter LDAP Password:
adding new entry "uid=usuario_ldap,dc=jorge,dc=proyecto"
```

Comprobamos con slapcat que el usuario ha sido añadido con éxito.

```
dn: uid=usuario_ldap,dc=jorge,dc=proyecto
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: usuario_ldap
uid: usuario_ldap
uidNumber: 1052
gidNumber: 1052
homeDirectory: /home/usuario_ldap
loginShell: /bin/bash
userPassword:: dXN1YXJpb19sZGFw
structuralObjectClass: account
entryUUID: 1540dfae-3890-103a-9f42-6ff0ad71fa6d
creatorsName: cn=admin,dc=jorge,dc=proyecto
createTimestamp: 20200601201320Z
entryCSN: 20200601201320.723681Z#000000#000#000000
modifiersName: cn=admin,dc=jorge,dc=proyecto
modifyTimestamp: 20200601201320Z
```

#### 6.4.2. Configuración de FreeRadius con LDAP

Una vez que ya tenemos LDAP con un usuario en la base de datos vamos a configurar, para que los usuarios en la base de datos, puedan acceder a la red por radius.

Lo primero será instalar el paquete **freeradius-ldap**:

```
root@tfc:/etc/freeradius/3.0# apt-get install freeradius-ldap
```

Ahora modificaremos los archivos de configuración de freeradius. Deberemos modificar el archivo de configuración del módulo de LDAP, que

se encuentra en **/etc/freeradius/3.0/mods-available/ldap**, aquí deberemos de especificar los datos de nuestro servidor ldap.

En la directiva **ldap{}**, deberemos modificar lo siguiente:

La dirección IP o nombre de nuestro servidor.

```
server = 'localhost'
```

Credenciales del administrador de ldap.

```
identity = 'cn=admin,dc=jorge,dc=proyecto'
password = admin1234
```

En la directiva **user{}**, debemos modificar:

```
base_dn = "dc=jorge,dc=proyecto"
```

Con esto vamos a indicar cuál es el **Distinguish Name (dn)** a partir del cual se van a realizar las búsquedas de los usuarios de nuestra base de datos.

Guardamos el archivo y realizamos el enlace simbólico a la carpeta **/etc/freeradius/3.0/mods-enabled/** y cambiamos el propietario y el grupo del enlace simbólico.

```
root@tfc:/etc/freeradius/3.0/mods-enabled# ln -s ../mods-available/ldap ldap \
> && \
> chown -h freerad: ldap
```

```
root@tfc:/etc/freeradius/3.0/mods-enabled# ll
total 8
drwxr-xr-x 2 freerad freerad 4096 jun  1 19:22 ./
drwxr-xr-x 9 freerad freerad 4096 jun  1 09:53 ../
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 always -> ../mods-available/always
lrwxrwxrwx 1 freerad freerad 29 mar 16 10:51 attr_filter -> ../mods-available/attr_filter
lrwxrwxrwx 1 freerad freerad 27 mar 16 10:51 cache_eap -> ../mods-available/cache_eap
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 chap -> ../mods-available/chap
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 detail -> ../mods-available/detail
lrwxrwxrwx 1 freerad freerad 28 mar 16 10:51 detail.log -> ../mods-available/detail.log
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 digest -> ../mods-available/digest
lrwxrwxrwx 1 freerad freerad 33 mar 16 10:51 dynamic_clients -> ../mods-available/dynamic_clients
lrwxrwxrwx 1 freerad freerad 21 mar 16 10:51 eap -> ../mods-available/eap
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 echo -> ../mods-available/echo
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 exec -> ../mods-available/exec
lrwxrwxrwx 1 freerad freerad 28 mar 16 10:51 expiration -> ../mods-available/expiration
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 expr -> ../mods-available/expr
lrwxrwxrwx 1 freerad freerad 23 mar 16 10:51 files -> ../mods-available/files
lrwxrwxrwx 1 freerad freerad 22 jun  1 19:22 ldap -> ../mods-available/ldap
lrwxrwxrwx 1 freerad freerad 25 mar 16 10:51 linelog -> ../mods-available/linelog
lrwxrwxrwx 1 freerad freerad 27 mar 16 10:51 logintime -> ../mods-available/logintime
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 mschap -> ../mods-available/mschap
lrwxrwxrwx 1 freerad freerad 27 mar 16 10:51 ntlm_auth -> ../mods-available/ntlm_auth
lrwxrwxrwx 1 freerad freerad 21 mar 16 10:51 pap -> ../mods-available/pap
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 passwd -> ../mods-available/passwd
lrwxrwxrwx 1 freerad freerad 28 mar 16 10:51 preprocess -> ../mods-available/preprocess
lrwxrwxrwx 1 freerad freerad 25 mar 16 10:51 radutmp -> ../mods-available/radutmp
lrwxrwxrwx 1 freerad freerad 23 mar 16 10:51 realm -> ../mods-available/realm
lrwxrwxrwx 1 freerad freerad 27 mar 16 10:51 replicate -> ../mods-available/replicate
lrwxrwxrwx 1 freerad freerad 21 mar 16 10:51 soh -> ../mods-available/soh
lrwxrwxrwx 1 freerad freerad 21 may 14 09:53 sql -> ../mods-available/sql
lrwxrwxrwx 1 freerad freerad 26 mar 16 10:51 sradutmp -> ../mods-available/sradutmp
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 unix -> ../mods-available/unix
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 unpack -> ../mods-available/unpack
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 utf8 -> ../mods-available/utf8
root@tfc:/etc/freeradius/3.0/mods-enabled#
```

A continuación, en el directorio **/etc/freeradius/3.0/sites-enabled** vamos a modificar dos archivos. El primero será el archivo default, en el que vamos

a descomentar las siguientes líneas para indicar a **FreeRadius**, que vamos a usar nuestra base de datos de LDAP.

En la directiva **authorize{}**, descomentamos la siguiente línea.

```
#
# The ldap module reads passwords from the LDAP database.
-ldap
```

Y comentamos la siguiente línea.

```
#
# Read the 'users' file. In v3, this is located in
# raddb/mods-config/files/authorize
# files
```

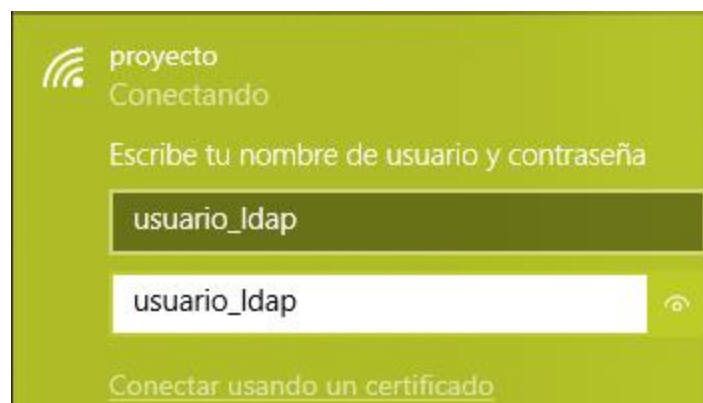
Realizamos lo mismo en el archivo **inner-tunnel** y descomentamos la siguiente línea.

```
#
Auth-Type LDAP {
    ldap
}
```

Con estos cambios ya tenemos configurado **FreeRadius**, reiniciamos el servicio para aplicar los cambios y ya tendremos todo configurado.

Comprobamos que el usuario de ldap tiene acceso a la red.

```
root@tfc:/etc/freeradius/3.0# radtest usuario_ldap usuario_ldap localhost 0 testing123
Sent Access-Request Id 65 from 0.0.0.0:53169 to 127.0.0.1:1812 length 82
  User-Name = "usuario_ldap"
  User-Password = "usuario_ldap"
  NAS-IP-Address = 127.0.1.1
  NAS-Port = 0
  Message-Authenticator = 0x00
  Cleartext-Password = "usuario_ldap"
Received Access-Accept Id 65 from 127.0.0.1:1812 to 0.0.0.0:0 length 20
root@tfc:/etc/freeradius/3.0# _
```





## 6.5. Autenticación de clientes inalámbricos mediante certificados.

FreeRadius permite el uso de certificados para la conexión de los clientes inalámbricos. Con estos certificados el usuario puede acceder a la red sin necesidad de ingresar sus credenciales. Para ello FreeRadius nos provee de varios scripts para la generación de estos certificados, en total serán tres certificados:

- Autoridad Certificadora.
- Certificado del servidor.
- Certificado del cliente.

### 6.5.1. Generación de certificados.

Los scripts para la generación de los certificados se encuentra en la ruta **/etc/freeradius/3.0/certs**

Primero tendremos que modificar los archivos de configuración de los certificados. Empezaremos por el de la Autoridad certificadora:

```
[ req ]
prompt                        = no
distinguished_name            = certificate_authority
default_bits                   = 2048
input_password                 = certificado
output_password                = certificado
x509_extensions               = v3_ca

[certificate_authority]
countryName                    = ES
stateOrProvinceName            = Andalucia
localityName                    = Cordoba
organizationName                = Autoridad SA
emailAddress                    = admin@autoridad.org
commonName                     = "Autoridad Certificadora"
```

Deberemos cambiar de **[req]** los campos **input\_password** y **output\_password**, y por último modificaremos según nuestros datos los campos de **[certificate\_authority]**.

Realizaremos los cambios tanto para el **server.cnf** y el **client.cnf**. El servidor lo configuraremos tal que así:



```
[ req ]
prompt                        = no
distinguished_name            = server
default_bits                  = 2048
input_password                = certificado
output_password               = certificado

[server]
countryName                   = ES
stateOrProvinceName           = Andalucia
localityName                   = Cordoba
organizationName               = Autoridad SA
emailAddress                   = admin@servidor.org
commonName                     = "Servidor Radius"
```

Y el cliente:

```
[ req ]
prompt                        = no
distinguished_name            = client
default_bits                  = 2048
input_password                = certificado
output_password               = certificado

[client]
countryName                   = ES
stateOrProvinceName           = Andalucia
localityName                   = Cordoba
organizationName               = Autoridad SA
emailAddress                   = jorge@proyecto.es
commonName                     = Jorge Rodriguez Mora
```

Antes de nada es muy importante que el campo **organizationName** sea IGUAL en los tres archivos de configuración ya que de lo contrario no saldrá un error. Una vez que tengamos los tres archivos configurados, ejecutaremos el comando **make**, con lo que se generarán los certificados como se muestran en la imagen.

```

-rwxr-xr-x 1 freerad freerad 4409 jun  5 11:02 01.pem*
-rwxr-xr-x 1 freerad freerad 4417 jun  5 11:06 02.pem*
-rwxr-xr-x 1 freerad freerad 2706 abr 17 2019 bootstrap*
-rwxr-xr-x 1 freerad freerad 1435 jun  5 10:55 ca.cnf*
-rwxr-xr-x 1 freerad freerad 1269 jun  5 10:55 ca.der*
-rwxr-xr-x 1 freerad freerad 1854 jun  5 10:55 ca.key*
-rwxr-xr-x 1 freerad freerad 1773 jun  5 10:55 ca.pem*
drwxr-xr-x 2 freerad freerad 4096 jun  5 10:27 certs_radius/
-rwxr-xr-x 1 freerad freerad 1115 jun  5 11:06 client.cnf*
-rwxr-xr-x 1 freerad freerad 4417 jun  5 11:06 client.crt*
-rwxr-xr-x 1 freerad freerad 1054 jun  5 11:06 client.csr*
-rwxr-xr-x 1 freerad freerad 1854 jun  5 11:06 client.key*
-rwxr-xr-x 1 freerad freerad 2589 jun  5 11:06 client.p12*
-rwxr-xr-x 1 freerad freerad 3696 jun  5 11:06 client.pem*
-rwxr-xr-x 1 freerad freerad  424 jun  5 12:22 dh*
-rwxr-xr-x 1 freerad freerad  230 jun  5 11:06 index.txt*
-rwxr-xr-x 1 freerad freerad   21 jun  5 11:06 index.txt.attr*
-rwxr-xr-x 1 freerad freerad   21 jun  5 11:02 index.txt.attr.old*
-rwxr-xr-x 1 freerad freerad  113 jun  5 11:02 index.txt.old*
-rwxr-xr-x 1 freerad freerad 1131 abr 17 2019 inner-server.cnf*
-rwxr-xr-x 1 freerad freerad 3696 jun  5 11:06 'jorge@proyecto.es.pem'*
-rwxr-xr-x 1 freerad freerad 6155 abr 17 2019 Makefile*
-rwxr-xr-x 1 freerad freerad  176 jun  5 11:06 passwords.mk*
-rwxr-xr-x 1 freerad freerad 8714 abr 17 2019 README*
-rwxr-xr-x 1 freerad freerad    3 jun  5 11:06 serial*
-rwxr-xr-x 1 freerad freerad    3 jun  5 11:02 serial.old*
-rwxr-xr-x 1 freerad freerad 1122 jun  5 11:02 server.cnf*
-rwxr-xr-x 1 freerad freerad 4409 jun  5 11:02 server.crt*
-rwxr-xr-x 1 freerad freerad 1050 jun  5 11:02 server.csr*
-rwxr-xr-x 1 freerad freerad 1854 jun  5 11:02 server.key*
-rwxr-xr-x 1 freerad freerad 2581 jun  5 11:02 server.p12*
-rwxr-xr-x 1 freerad freerad 3688 jun  5 11:02 server.pem*
-rwxr-xr-x 1 freerad freerad  708 abr 17 2019 xpextensions*
root@tfc:/etc/freeradius/3.0/certs#

```

### 6.5.2. Configuración de FreeRadius.

Ahora debemos configurar FreeRadius para poder usar los certificados, para ello iremos al archivo de configuración del módulo EAP, que se encuentra en la ruta **/etc/freeradius/3.0/mods-enabled/eap**

```

root@tfc:/etc/freeradius/3.0/certs# cd ../mods-enabled/
root@tfc:/etc/freeradius/3.0/mods-enabled# ll
total 8
drwxr-xr-x 2 freerad freerad 4096 jun  5 21:21 ./
drwxr-xr-x 9 freerad freerad 4096 jun  5 15:41 ../
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 always -> ../mods-available/always
lrwxrwxrwx 1 freerad freerad 29 mar 16 10:51 attr_filter -> ../mods-available/attr_filter
lrwxrwxrwx 1 freerad freerad 27 mar 16 10:51 cache_eap -> ../mods-available/cache_eap
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 chap -> ../mods-available/chap
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 detail -> ../mods-available/detail
lrwxrwxrwx 1 freerad freerad 28 mar 16 10:51 detail.log -> ../mods-available/detail.log
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 digest -> ../mods-available/digest
lrwxrwxrwx 1 freerad freerad 33 mar 16 10:51 dynamic_clients -> ../mods-available/dynamic_clients
lrwxrwxrwx 1 freerad freerad 21 mar 16 10:51 eap -> ../mods-available/eap
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 echo -> ../mods-available/echo
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 exec -> ../mods-available/exec
lrwxrwxrwx 1 freerad freerad 28 mar 16 10:51 expiration -> ../mods-available/expiration
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 expr -> ../mods-available/expr
lrwxrwxrwx 1 freerad freerad 23 mar 16 10:51 files -> ../mods-available/files
lrwxrwxrwx 1 freerad freerad 22 jun  5 13:35 ldap -> ../mods-available/ldap
lrwxrwxrwx 1 freerad freerad 25 mar 16 10:51 linelog -> ../mods-available/linelog
lrwxrwxrwx 1 freerad freerad 27 mar 16 10:51 logintime -> ../mods-available/logintime
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 mschap -> ../mods-available/mschap
lrwxrwxrwx 1 freerad freerad 27 mar 16 10:51 ntlm_auth -> ../mods-available/ntlm_auth
lrwxrwxrwx 1 freerad freerad 21 mar 16 10:51 pap -> ../mods-available/pap
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 passwd -> ../mods-available/passwd
lrwxrwxrwx 1 freerad freerad 28 mar 16 10:51 preprocess -> ../mods-available/preprocess
lrwxrwxrwx 1 freerad freerad 25 mar 16 10:51 radutmp -> ../mods-available/radutmp
lrwxrwxrwx 1 freerad freerad 23 mar 16 10:51 realm -> ../mods-available/realm
lrwxrwxrwx 1 freerad freerad 27 mar 16 10:51 replicate -> ../mods-available/replicate
lrwxrwxrwx 1 freerad freerad 21 mar 16 10:51 soh -> ../mods-available/soh
lrwxrwxrwx 1 freerad freerad 21 jun  5 16:00 sql -> ../mods-available/sql
lrwxrwxrwx 1 freerad freerad 26 mar 16 10:51 sradutmp -> ../mods-available/sradutmp
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 unix -> ../mods-available/unix
lrwxrwxrwx 1 freerad freerad 24 mar 16 10:51 unpack -> ../mods-available/unpack
lrwxrwxrwx 1 freerad freerad 22 mar 16 10:51 utf8 -> ../mods-available/utf8
root@tfc:/etc/freeradius/3.0/mods-enabled#

```

Lo primero será modificar el tipo de autenticación que está usando FreeRadius, para ello añadiremos o modificaremos la siguiente línea.

```
#
default_eap_type = tls
```

Ahora deberemos especificar los certificados que vamos a usar:

```

tls-config tls-common {
    private_key_password = certificado
                        #whatever
    #private_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
    private_key_file = /etc/freeradius/3.0/certs/server.key

    # If Private key & Certificate are located in
    # the same file, then private_key_file &
    # certificate_file must contain the same file
    # name.
    #
    # If ca_file (below) is not used, then the
    # certificate_file below MUST include not
    # only the server certificate, but ALSO all
    # of the CA certificates used to sign the
    # server certificate.
    #certificate_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
    certificate_file = /etc/freeradius/3.0/certs/server.pem

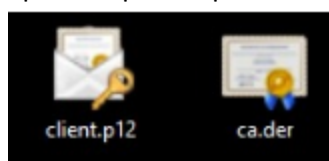
    #
    ca_file = /etc/ssl/certs/
    ca_file = /etc/freeradius/3.0/certs/ca.pem

```

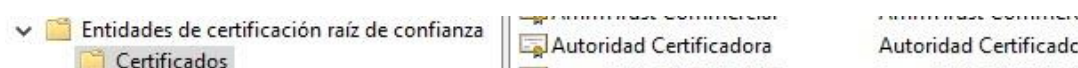
En **private\_key\_password** deberemos especificar la contraseña que añadimos en los archivos de configuración. Con esto guardamos los cambios y reiniciamos el servicio, si reinicia no hemos tenido errores en la configuración.

### 6.5.3. Autenticación de clientes inalámbricos a la red.

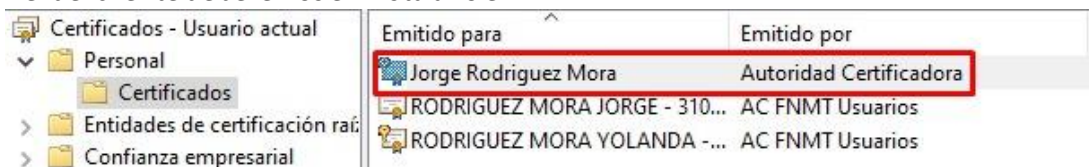
Importamos los certificados de la Autoridad Certificadora y del cliente, al nodo inalámbrico que va a usar el certificado para tener acceso a la red. Una de las opciones para importar certificado es usar **FileZilla**.



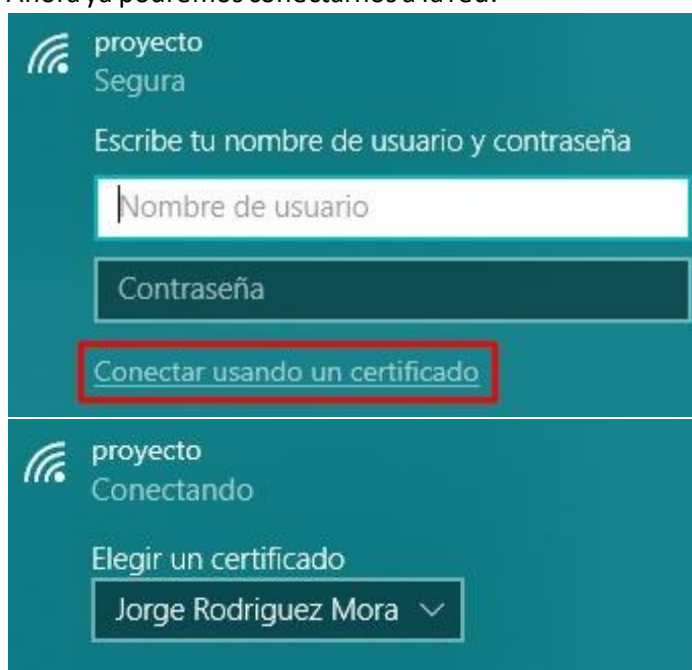
Una vez que tengamos los certificados en el cliente deberemos instalarlos. El certificado de la Autoridad Certificadora, deberemos instalarlo en:



Y el del cliente deberemos en instalarlo en:



Ahora ya podremos conectarnos a la red:



Existe un problema a la hora de usar los certificados con Windows 10, por lo que voy a demostrar el funcionamiento de los certificados a través de Android. Realizaremos el mismo procedimiento, deberemos importar los certificados a nuestro móvil, por ejemplo a través de WhatsApp. Una vez importados procederemos a instalarlos, es importante recordar la clave de los certificados.

Iremos a **Ajustes de red -> Wi-Fi**, seleccionaremos la red **proyecto** y modificaremos sus ajustes.

Cabe recordar que el protocolo que se usa para los certificados es **EAP-TLS**, sabiendo esto modificaremos los ajustes de nuestra red Wi-Fi en el móvil, de la siguiente manera:

0:35

proyecto Detalles de red

Intensidad de la señal Excelente

proyecto

Método EAP TLS >

Autenticación de fase 2 Ninguno >

Certificado de CA ca >

Dominio

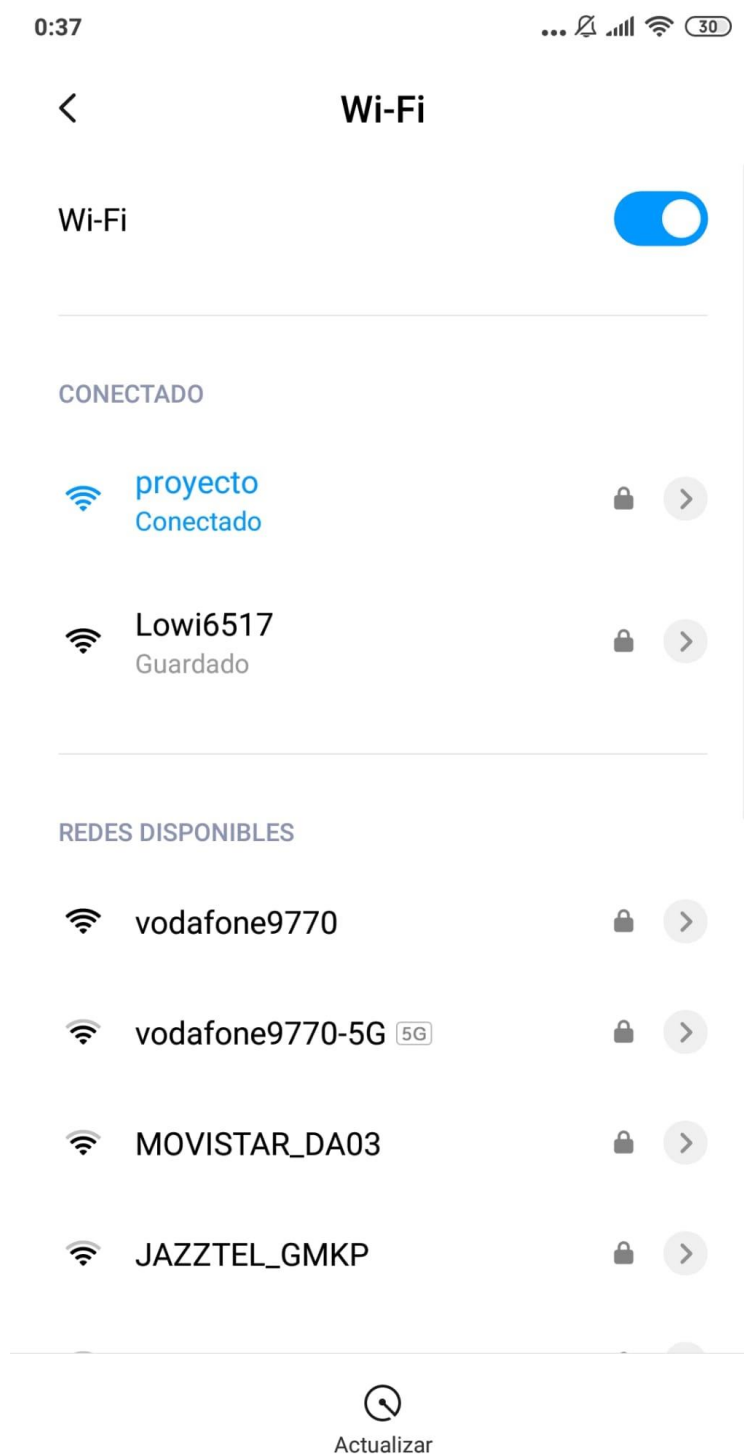
Certificado de usuario f1a031b9-0a0e2abbe695d.. >

Nombre de usuario Jorge

Identidad anónima

Opciones sencillas Guardar

Una vez que hayamos modificado la conexión, guardamos los ajustes y probamos conexión.



Con esto, hemos conseguido acceder a la red, sin uso de las credenciales de usuario y contraseña, ya que la conexión se ha realizado a través de los datos contenidos en los certificados.

Por último vamos a realizar la autenticación en un nodo Ubuntu 18.04 LTS. Los pasos a realizar son los mismos que hemos realizado anteriormente. Lo primero será importar los certificados, por ejemplo, con un pen drive).

Una vez con los certificados importados, vamos a configurar la red para modificar el protocolo de acceso a la red para el uso de los certificados. Aquí no hace falta la instalación de los certificados, simplemente indicar la ruta.

**Visible Networks**

Network Name	Security	Authentication	Identity	User certificate	CA certificate	Private key	Private key password
Lowi6517	WPA & WPA2 Enterprise	TLS	Jorge	(None)	ca.der	client.p12	.....
proyecto	WPA & WPA2 Enterprise	TLS	Jorge	(None)	ca.der	client.p12	.....
Sweex LW050v2	WPA & WPA2 Enterprise	TLS	Jorge	(None)	ca.der	client.p12	.....
vodafone9770	WPA & WPA2 Enterprise	TLS	Jorge	(None)	ca.der	client.p12	.....

Cancel **proyecto** Apply

Details Identity IPv4 IPv6 **Security**

Security: WPA & WPA2 Enterprise

Authentication: TLS

Identity: Jorge

User certificate: (None)

CA certificate: ca.der

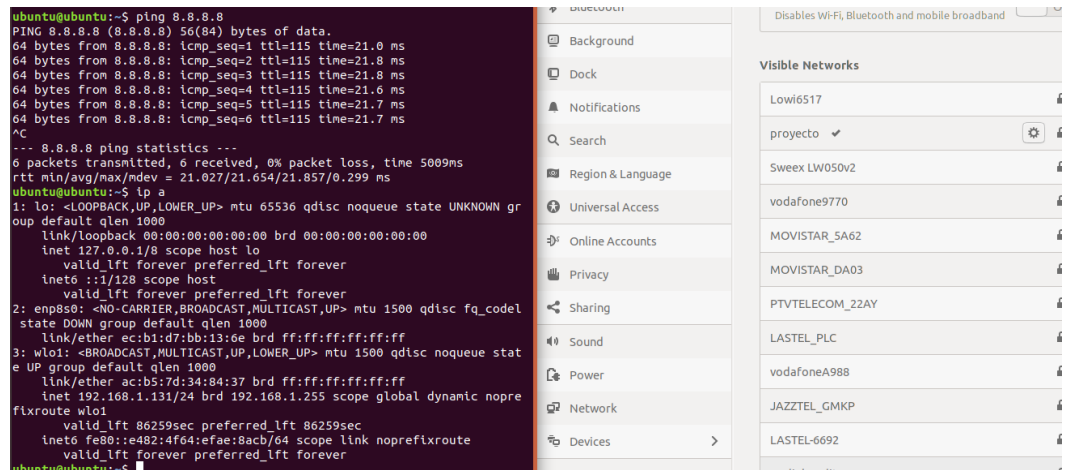
☐ No CA certificate is required

Private key: client.p12

Private key password: .....

☐ Show password

Guardamos los cambios y probamos la conexión a la red.



## 6.6. Almacenamiento de claves seguras mediante funciones hash.

Una función hash es un algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija. La longitud del hash es independiente de la longitud del valor de entrada.

Una de las funciones más conocidas son MD5 y SHA (SHA1,SHA128,SHA256,SHA512), estas funciones nos permiten encriptar nuestras contraseñas impidiendo así que las puedan obtener sin nuestro consentimiento.

A continuación, vamos a aplicar estas funciones a nuestro servidor Radius, añadiendo así una capa de seguridad. Lo primero será saber los atributos que nos permiten que Radius trabaje con funciones hash, para ello ejecutaremos el siguiente comando **man rlm\_pap**, con esto, nos saldrá el manual del módulo que permite el cifrado de contraseñas en Radius.



Header	Attribute	Description
{clear}	Cleartext-Password	Clear-text passwords
{cleartext}	Cleartext-Password	Clear-text passwords
{crypt}	Crypt-Password	Unix-style "crypt"ed passwords
{md5}	MD5-Password	MD5 hashed passwords
{base64_md5}	MD5-Password	MD5 hashed passwords
{smd5}	SMD5-Password	MD5 hashed passwords, with a salt
{sha}	SHA1-Password	SHA1 hashed passwords
	SHA1-Password	SHA1 hashed passwords
{ssha}	SSHA-Password	SHA1 hashed passwords, with a salt
{sha2}	SHA2-Password	SHA2 hashed passwords
{sha224}	SHA2-Password	SHA2 hashed passwords
{sha256}	SHA2-Password	SHA2 hashed passwords
{sha384}	SHA2-Password	SHA2 hashed passwords
{sha512}	SHA2-Password	SHA2 hashed passwords
{ssha224}	SSHA2-224-Password	SHA2 hashed passwords, with a salt
{ssha256}	SSHA2-256-Password	SHA2 hashed passwords, with a salt
{ssha384}	SSHA2-384-Password	SHA2 hashed passwords, with a salt
{ssha512}	SSHA2-512-Password	SHA2 hashed passwords, with a salt
{nt}	NT-Password	Windows NT hashed passwords
{nthash}	NT-Password	Windows NT hashed passwords
{md4}	NT-Password	Windows NT hashed passwords
{x-nthash}	NT-Password	Windows NT hashed passwords
{ns-mta-md5}	NS-MTA-MD5-Password	Netscape MTA MD5 hashed passwords
{x- orcllmv}	LM-Password	Windows LANMAN hashed passwords
{X- orclntv}	NT-Password	Windows NT hashed passwords

Según el tipo de cifrado que queramos aplicar a nuestras contraseñas deberemos aplicar uno de los atributos de la imagen.

#### 6.6.1. Base de Datos interna FreeRadius.

Vamos a aplicar MD5 a un usuario de nuestro archivo USERS. Por ejemplo al usuario **bob**, cuya contraseña no está encriptada

```
bob      Cleartext-Password := "hello"
         Reply-Message := "Hello, %{User-Name}",
         Framed-IP-Address = 192.168.1.10,
         Framed-IP-Netmask = 255.255.255.0,
         Framed-Routing = Broadcast-Listen
```

Si ejecutamos el comando radtest veremos que el usuario puede acceder a la red.

```
root@tfc:/etc/freeradius/3.0# radtest bob hello localhost 0 testing123
Sent Access-Request Id 144 from 0.0.0.0:34860 to 127.0.0.1:1812 length 73
  User-Name = "bob"
  User-Password = "hello"
  NAS-IP-Address = 127.0.1.1
  NAS-Port = 0
  Message-Authenticator = 0x00
  Cleartext-Password = "hello"
Received Access-Accept Id 144 from 127.0.0.1:1812 to 0.0.0.0:0 length 50
```

Para encriptar una contraseña vamos a usar el comando **md5sum** de la siguiente manera **echo -n password | md5sum**:

```
root@tfc:/etc/freeradius/3.0# echo -n hola | md5sum
4d186321c1a7f0f354b297e8914ab240 -
```

Con esto ya tenemos la función hash MD5 de hola. En el archivo user vamos a cambiar **hello** por la función hash.

```
bob      Cleartext-Password := "4d186321c1a7f0f354b297e8914ab240"
```

Ahora debemos modificar el atributo **Cleartext-Password**, por el de la función hash correspondiente **MD5-Password**.

```
bob      MD5-Password := "4d186321c1a7f0f354b297e8914ab240"
```

Reiniciamos el servicio freeradius, y comprobamos con radtest:

```
root@tfc:/etc/freeradius/3.0# radtest bob hola localhost 0 testing123
Sent Access-Request Id 248 from 0.0.0.0:33609 to 127.0.0.1:1812 length 73
  User-Name = "bob"
  User-Password = "hola"
  NAS-IP-Address = 127.0.1.1
  NAS-Port = 0
  Message-Authenticator = 0x00
  Cleartext-Password = "hola"
Received Access-Accept Id 248 from 127.0.0.1:1812 to 0.0.0.0:0 length 50
```

Comprobamos que el usuario bob tiene acceso.

### 6.6.2. Base de datos MySQL.

Para realizar esta función en MySQL deberemos realizar estos cambios en la base de datos.

En la tabla **radcheck** es donde se almacenan los atributos de la contraseña.

```
mysql> select * from radcheck;
+----+-----+-----+-----+-----+
| id | username | attribute | op | value |
+----+-----+-----+-----+-----+
| 1 | usu1 | usu1 | := | grupo1 |
| 2 | jorge | Cleartext-Password | := | proyecto |
| 3 | jorge2 | Cleartext-Password | := | jorge2 |
| 4 | jorge3 | Cleartext-Password | := | jorge3 |
| 5 | yolanda | Cleartext-Password | := | yolanda |
| 6 | prueba1 | Cleartext-Password | := | prueba1 |
| 8 | trelly | Cleartext-Password | := | chipi |
+----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Deberemos cambiar **Cleartext-Password** por el atributo de cifrado que queramos, y en el campo **value** deberemos añadir la función hash.

```
mysql> insert into radcheck values(null,"cifrado","MD5-Password",":=",md5("hola"));

mysql> select * from radcheck;
+----+-----+-----+-----+-----+
| id | username | attribute | op | value |
+----+-----+-----+-----+-----+
| 1 | usu1 | usu1 | := | grupo1 |
| 2 | jorge | Cleartext-Password | := | proyecto |
| 3 | jorge2 | Cleartext-Password | := | jorge2 |
| 4 | jorge3 | Cleartext-Password | := | jorge3 |
| 5 | yolanda | Cleartext-Password | := | yolanda |
| 6 | prueba1 | Cleartext-Password | := | prueba1 |
| 8 | trelly | Cleartext-Password | := | chipi |
| 9 | cifrado | MD5-Password | := | 4d186321c1a7f0f354b297e8914ab240 |
+----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Comprobamos que el nuevo usuario tiene acceso a la red.

```
root@tfc:/etc/freeradius/3.0/mods-enabled# radtest cifrado hola localhost 0 testing123
Sent Access-Request Id 80 from 0.0.0.0:38614 to 127.0.0.1:1812 length 77
  User-Name = "cifrado"
  User-Password = "hola"
  NAS-IP-Address = 127.0.0.1
  NAS-Port = 0
  Message-Authenticator = 0x00
  Cleartext-Password = "hola"
Received Access-Accept Id 80 from 127.0.0.1:1812 to 0.0.0.0:0 length 20
```

### 6.6.3. Base de datos LDAP.

Para las funciones hash de LDAP, al introducir la clave del usuario en el fichero de text, cuando se añade este usuario a la base de datos, se genera una función hash de forma automática.

En el archivo de texto para crear un usuario tenemos lo siguiente:

```
#Usuario ldap
dn: uid=usuario_ldap,dc=jorge,dc=proyecto
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: usuario_ldap
uid: usuario_ldap
uidNumber: 1050
gidNumber: 1050
homeDirectory: /home/usuario_ldap
loginShell: /bin/bash
userPassword: usuario_ldap
```

El campo **userPassword** es el que va a ser cifrado cuando el usuario se añada a la base de datos. Una vez añadido, si vemos los usuarios de LDAP veremos cómo se ha cifrado la contraseña. Ejecutamos el comando **slapcat**:

```
dn: uid=usuario_ldap,dc=jorge,dc=proyecto
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: usuario_ldap
uid: usuario_ldap
uidNumber: 1050
gidNumber: 1050
homeDirectory: /home/usuario_ldap
loginShell: /bin/bash
userPassword:: dXN1YXJpb19sZGFw
structuralObjectClass: account
entryUUID: 591e19ba-37c0-103a-8114-85a5587a14d2
creatorsName: cn=admin,dc=jorge,dc=proyecto
createTimestamp: 20200531192619Z
entryCSN: 20200531192619.261122Z#000000#000#000000
modifiersName: cn=admin,dc=jorge,dc=proyecto
modifyTimestamp: 20200531192619Z
```

## 7. Incidencias, objetivos cumplidos y mejoras.

SPRINTS	RESULTADO
Creación del entorno de pruebas.	✓
Autenticación remota de clientes inalámbricos.	✓
FreeRadius con MySQL	✓
FreeRadius con LDAP	✓
Almacenamiento de claves seguras mediante funciones hash.	✓
Autenticación mediante certificados.	×

Entre las próximas mejoras del proyecto:

- Automatización de varios procesos, como la creación de usuarios en MySQL y LDAP mediante scripts.
- Servidor de MySQL independiente en remoto.
- Realización de una arquitectura maestro-esclavo MySQL para añadir alta disponibilidad a las credenciales de los usuarios.
- Uso de herramientas gráficas para el mantenimiento de las bases de datos como phpmyadmin y phpldapadmin.

Entre las incidencias encontradas durante el desarrollo del proyecto:

- Fallos de autenticación de clientes mediante certificados, debido al sistema operativo Windows.
- Escasa documentación actualizada de FreeRadius, con todos los cambios y mejoras de la versión 3.0.

## 8. Webgrafía.

*Criptografía: Funciones Hash.* (2020). Obtenido de  
[https://en.wikipedia.org/wiki/Cryptographic\\_hash\\_function](https://en.wikipedia.org/wiki/Cryptographic_hash_function)

*Documentación de FreeRadius 3.0.* (2020). Obtenido de FreeRadius.org:  
<https://www.freeradius.org/documentation/>

*Documentación MySQL.* (2020). Obtenido de MySQL.com: <https://dev.mysql.com/doc/>

*Manual de comandos de Ubuntu 18.04 LTS.* (2020). Obtenido de Ubuntu Manuals:  
<http://manpages.ubuntu.com/>

*Manual OpenLDAP 2.4.* (2020). Obtenido de OpenLDAP.org:  
<https://www.openldap.org/doc/admin24/>

*Módulos FreeRadius 3.0.* (2020). Obtenido de Network Radius:  
<https://networkradius.com/doc/current/raddb/mods-available/home.html>