IJO'S & COMPANY

IÑAKI SESMA | JULEN UBIRIA | OLIVER RAMAJO

1. Introducción

La transformación digital es una necesidad clave para las instituciones educativas en la era moderna. Este proyecto tiene como objetivo principal modernizar las infraestructuras digitales del instituto Plaiaundi, desarrollando un entorno tecnológico innovador y eficiente.

Nuestra propuesta incluye la creación de una página web dedicada a la **gestión académica**, donde se podrá administrar información esencial como ciclos formativos, módulos, profesores y calificaciones. Esta web estará respaldada por una base de datos normalizada y un diseño adaptativo, asegurando su accesibilidad desde cualquier dispositivo.

Además, se implementará una infraestructura de red avanzada, basada en Proxmox para la virtualización de servidores, con un servidor Ubuntu encargado de ofrecer servicios esenciales como VPN y NFS para compartir datos. El sistema de red integrará VLANs para garantizar una segmentación y seguridad adecuadas, facilitando la comunicación entre las clases de 1ASIR3 y 2ASIR3.

Este prototipo será presentado durante las jornadas de puertas abiertas, mostrando las habilidades adquiridas en el ciclo de Administración de Sistemas Informáticos en Red (ASIR), destacando como una herramienta clave para atraer a futuros estudiantes interesados en el mundo de la informática.

2. Instalación de Proxmox VE

2.1.¿Qué es Proxmox VE?

Proxmox VE es una plataforma de virtualización que usa dos tecnologías: KVM para máquinas virtuales (VM) y LXC para contenedores. Es un hipervisor tipo 1, lo que significa que corre directamente sobre el hardware, sin necesidad de un sistema operativo base.

En comparación con VirtualBox, que es un hipervisor tipo 2 (se ejecuta sobre un sistema operativo existente), Proxmox VE es más eficiente y adecuado para entornos de producción y servidores, ya que se maneja de manera más directa con el hardware y tiene características avanzadas como la gestión de contenedores.

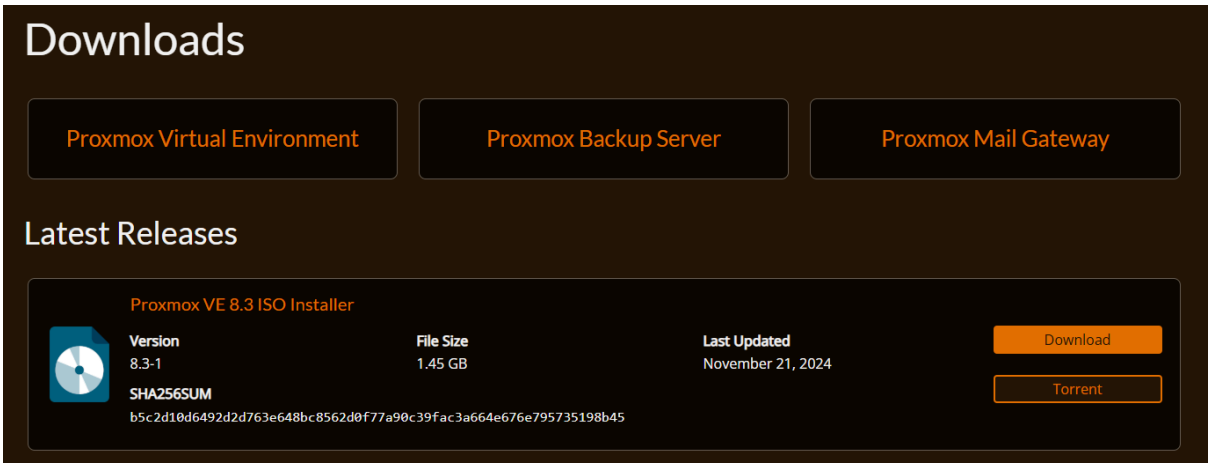
2.2. Requisitos Previos

2.2.1.Software Necesario

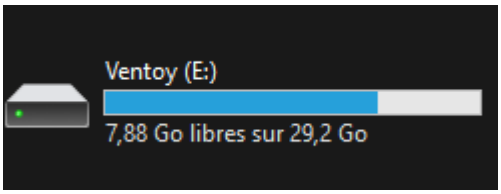
- Imagen ISO de Proxmox VE (descargada desde Proxmox).
- Software para grabar la ISO en un USB.

2.3. Crear un Medio de Instalación

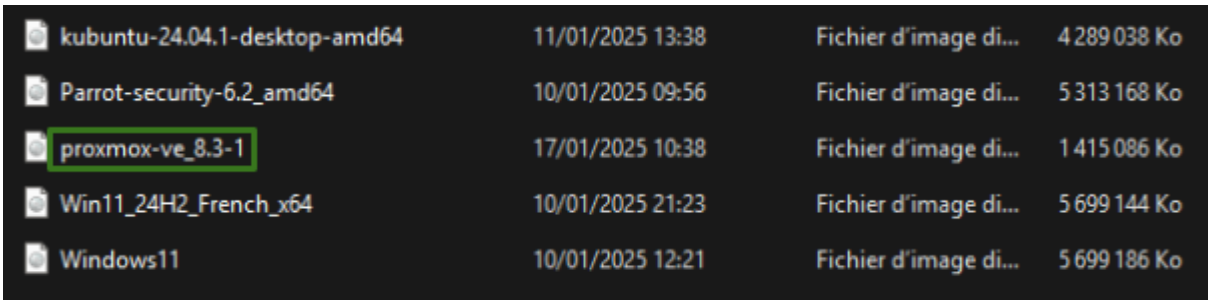
1. Descargamos la imagen ISO de Proxmox VE desde el sitio oficial.



2. Insertamos un USB en tu equipo.



3. Usamos una herramienta para grabar la ISO en el USB, en nuestro caso haremos uso de Ventoy:



   - Selecciona la imagen ISO descargada e insertala en la unidad de instalación.

2.4. Configuración de la BIOS/UEFI

1. Reiniciamos el equipo donde instalar Proxmox y accede al menú de configuración de BIOS/UEFI.

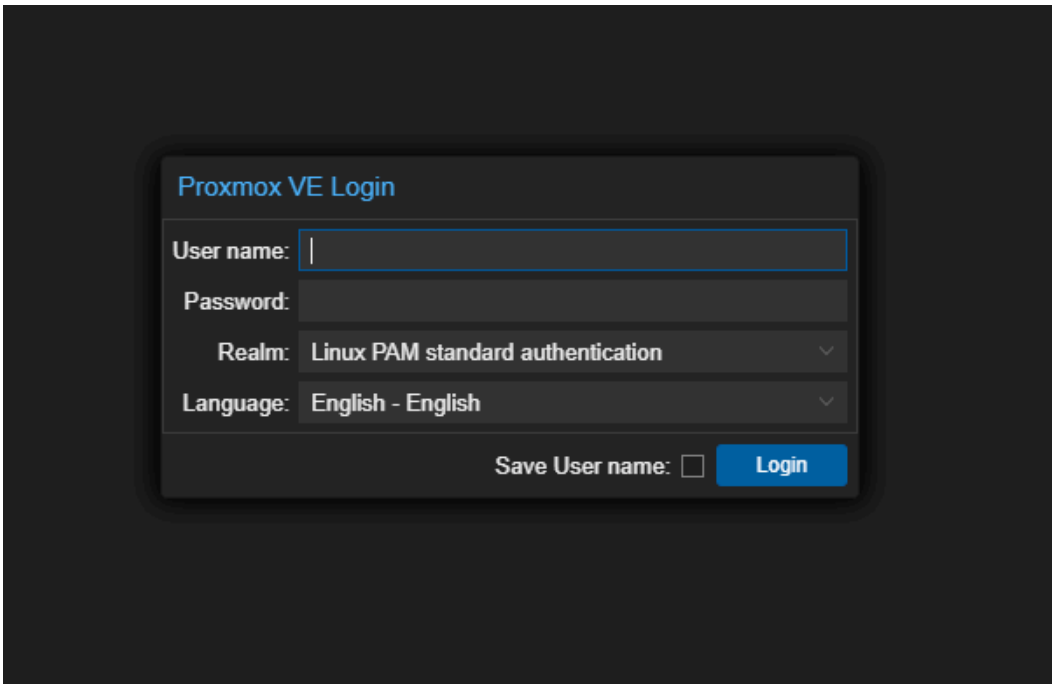
2.5. Instalación de Proxmox VE

1. Insertamos el USB y arranca desde él.
2. En el menú de inicio de Proxmox, selecciona:
   - **Install Proxmox VE**.
3. Aceptamos el acuerdo de licencia.
4. Seleccionamos el disco donde se instalará Proxmox:
5. Configuramos la región y zona horaria.
6. Introducimos una contraseña para el administrador y una dirección de correo para alertas.
7. Configuración la red:
   - Asignamos una dirección IP estática, máscara de red, puerta de enlace y DNS.

2.6. Primer Inicio y Configuración Inicial

1. Una vez completada la instalación, Proxmox se reiniciará.
2. Accedemos a la interfaz web desde un navegador o en su defecto, desde la propia máquina, en nuestro caso, accederemos desde la web:
   - Dirección: https://<IP_del_Servidor>:8006

3. Introducimos las credenciales del usuario root y la contraseña definida durante la instalación.



3. Creación de Máquinas Virtuales en Proxmox VE

3.1. Requisitos Previos

3.1.1.Infraestructura Necesaria

- **Servidor Proxmox VE**: Instalado y accesible desde la interfaz web (https://<IP_del_servidor>:8006).
- **Almacenamiento Suficiente**.
- **Imágenes ISO**: Descarga y sube al servidor las imágenes ISO necesarias:
  - **Ubuntu Server**: caso será usado para el servidor.
  - **Xubuntu**: entorno ligero y funcional para los clientes.

3.2. Máquinas Virtuales a Crear

3.2.1.Resumen

1. Servidor Ubuntu.
2. Cliente 1 para 1ASIR3.

3. Cliente 2 para 1ASIR3.
4. Cliente 1 para 2ASIR3.
5. Cliente 2 para 2ASIR3.

Cada máquina cliente debe estar conectada a su VLAN correspondiente y tener acceso al servidor. Todas las máquinas serán configuradas para el propósito del proyecto. Asimismo, en posteriores apartados, se explicará al detalle la configuración realizada para cada máquina.

3.3. Subir las Imágenes ISO a Proxmox

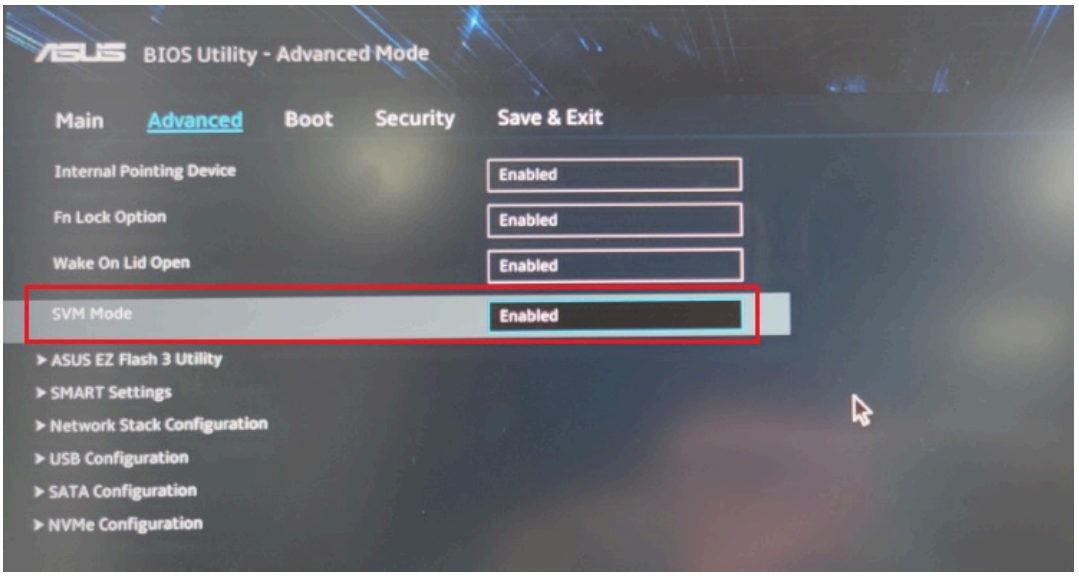
1. Accede a la interfaz web de Proxmox.
2. Ve a **Datacenter > Node > Local > ISO Images**.
3. Haz clic en **Upload**.
4. Selecciona la ISO y súbela al almacenamiento.

3.4. Crear una Máquina Virtual

3.4.1 Iniciar el Asistente de Creación

1. En el panel izquierdo, selecciona el nodo donde deseas crear la máquina.
2. Haz clic en **Create VM**.

ATENCIÓN: la virtualización deberá estar activada previamente en la BIOS en la que se aloja Proxmox.



3.4.2.Configuración Básica

1. **General**:
   - **VM ID**: Proporciona un identificador único (en este caso, 100 para el servidor, 101 para Cliente 1, etc.).
   - **Name**: Escribe un nombre descriptivo (por ejemplo, ServidorUbuntu o 1ASIR3Alum1, 2ASIR3Prof2).
2. **OS**:
   - Seleccionaremos la ISO correspondiente.
   - Configuraremos el tipo de sistema operativo (Linux, Ubuntu 64-bit).

3.4.3.Configurar el Hardware

1. **Disco Duro**:
   - Seleccionaremos el almacenamiento y asignaremos el tamaño del disco (15 GB para clientes, 50 GB para el servidor).
2. **CPU**:
   - Asignaremos 1 core para clientes y 2 cores para el servidor.
3. **Memoria RAM**:
   - Clientes: 1 GB.
   - Servidor: 2 GB.
4. **Red**:
   - Seleccionamos **VirtIO** para el adaptador de red.
   - Configuramos la red para conectarla a la VLAN correspondiente (en nuestro caso, configuraremos más tarde las VLANs).

3.5. Configurar las Máquinas Virtuales Específicas

3.5.1.Servidor Ubuntu

1. Descargamos y utilizamos una imagen de **Ubuntu Server**.
2. Durante la instalación:
    - Configura una dirección IP estática.

3.5.2.Clientes

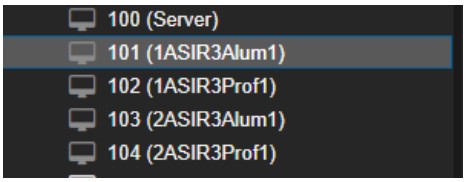
1. Descargamos una imagen de **Xubuntu**.
2. En caso de querer automatizar la instalación, usa un archivo de instalación desatendida (autoinstall.yaml).
3. Configura cada cliente para unirse al servidor:
    - Configura la IP de cada cliente según su VLAN.

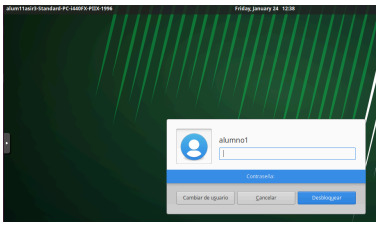
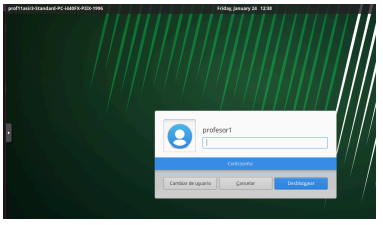
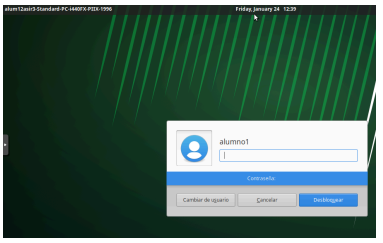
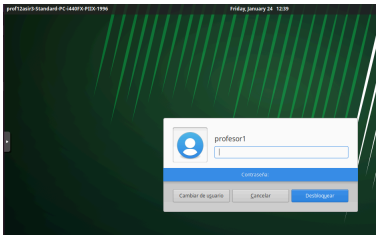
3.6. Configurar Redes y VLANs

1. En la interfaz de Proxmox:
    - En **Datacenter > Node > Network**.
    - Crearemos una interfaz de red para cada VLAN:
        - VLAN 1: Para el servidor.
        - VLAN 2: Para 1ASIR3.
        - VLAN 3: Para 2ASIR3.
2. Asociaremos cada máquina a su VLAN correspondiente en la configuración de red de Proxmox.

3.7. Finalizar Configuración y Pruebas

1. Una vez creadas las máquinas, iniciaremos cada una desde el panel de Proxmox.
2. Completaremos la instalación del sistema operativo.



Nombre	Sistema Operativo	Estado
100 *(Servidor)*	Ubuntu Server 24.04	
101 *(1ASIR3Alum1)*	Xubuntu 20.04.2 minimal	
102 *(1ASIR3Prof1)*	Xubuntu 20.04.2 minimal	
103 *(2ASIR3Alum1)*	Xubuntu 20.04.2 minimal	
104 *(2ASIR3Prof1)*	Xubuntu 20.04.2 minimal	

3.8. Creación y Uso de Plantillas en Proxmox

Las plantillas nos permiten crear máquinas virtuales de manera rápida y consistente, lo que es ideal para entornos donde necesitamos múltiples VMs con configuraciones similares.

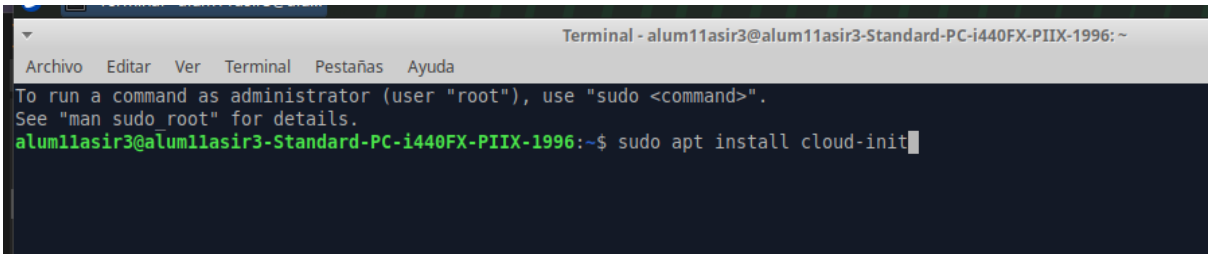
3.8.1.Crear una Plantilla

Preparamos una VM Base:

1. Creamos una VM con el sistema operativo y las configuraciones deseadas.
2. Instalamos todas las aplicaciones y actualizaciones necesarias.
3. Limpiamos la VM (por ejemplo, eliminamos historiales y cachés).

Asegurarnos de que cloud-init esté instalado:

1. Verificamos si cloud-init está instalado en la VM base. Si no lo está, lo instalamos:



Generamos un ID único para el sistema operativo:

2. Configuramos cloud-init para que, al clonarse la VM, se autogenere un ID único. Esto lo hacemos creando un archivo de configuración para cloud-init.
   Creamos el archivo /etc/cloud/cloud.cfg.d/99-final.cfg con el siguiente contenido:

```
system_info:
 default_user:
  name: root
  lock_passwd: false
  gecos: "Root User"
```

```
  sudo: ["ALL=(ALL) NOPASSWD:ALL"]

cloud_name: local
```

3. Para asegurarnos de que cada clon tenga un ID único, también añadimos la configuración para regenerar el ID de la máquina en la inicialización:

```
sudo cloud-init clean
```

3.8.2.Convertimos la VM en Plantilla:

1. Apagamos la VM.
2. Hacemos clic derecho sobre la VM y seleccionamos "Convert to Template".
3. Confirmamos la acción. La VM ahora es una plantilla y no puede iniciarse directamente.

3.8.3.Usar una Plantilla para Crear VMs

1. Seleccionamos la plantilla en el panel izquierdo.
2. Hacemos clic derecho y seleccionamos "Clone".
3. Completamos los siguientes pasos:
    - **Name**: Asignamos un nombre a la nueva VM.
    - **Target Node**: Seleccionamos el nodo donde se alojará la VM.
    - **Full Clone**: Seleccionamos esta opción si deseamos una copia independiente de la plantilla.
4. Hacemos clic en "Clone" para crear la VM.

3.8.4.Personalizar la VM Clonada

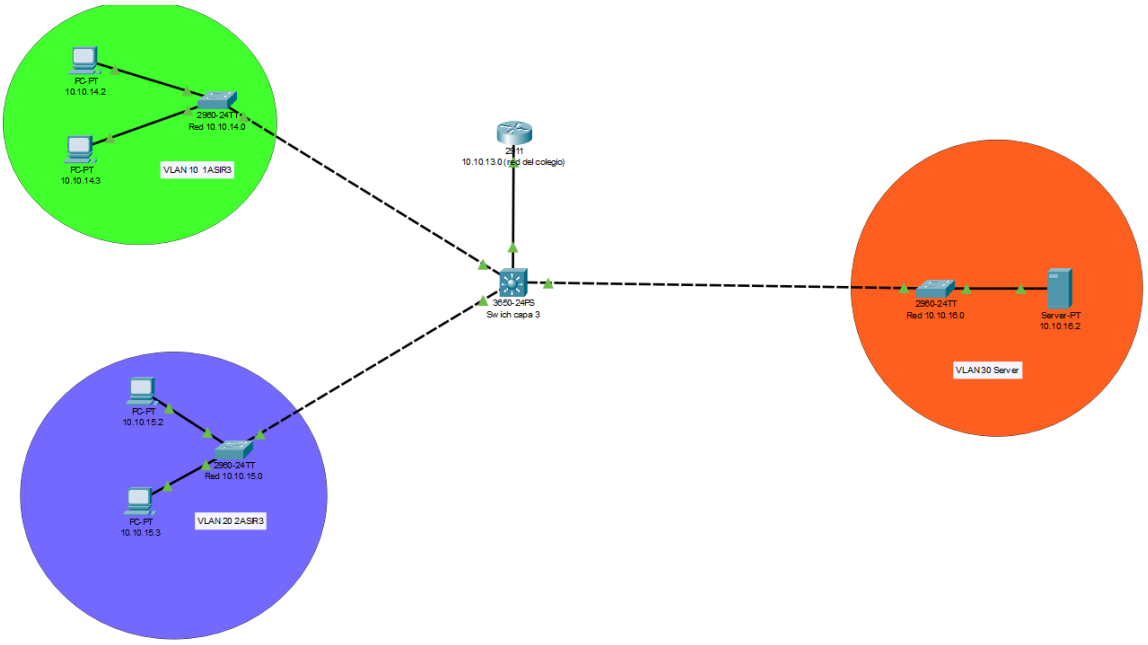
1. Iniciamos la VM clonada.
2. Accedemos a la consola y realizamos las personalizaciones necesarias (por ejemplo, cambiamos el nombre de host, configuramos la red, etc.).
    - Para cambiar el nombre de host:

```
sudo hostnamectl set-hostname nuevo-nombre
```

4. Configuración de la Red

4.1 Configurar la red de la VLAN

4.1.1.Diagrama de redes de las clases 1ASIR y 2ASIR (Packet Tracer)



La estructura debe de tener las 3 Vlan (1ASIR3, 2ASIR3, Server) en dos de ellas dos ordenadores y en la otra un servidor, además cada Vlan tendrá su switch propio conectados a un switch capa 3 que se encarga de enrutar y bloquear el tráfico además también se conectara un router simulando la red del colegio.

4.1.2.Asignar IPs

Ahora toca asignar las IPs a los dispositivos para ello yo he utilizado:

VLAN	Red	Rango	Gateway	Broadcast
VLAN 10	10.10.14.0/24	10.10.14.2-10.10.14.3	10.10.14.1	10.10.14.255
VLAN 20	10.10.15.0/24	10.10.15.2-10.10.15.3	10.10.15.1	10.10.14.255
VLAN 30	10.10.16.0/24	10.10.16.2-10.10.16.3	10.10.16.1	10.10.14.255

4.1.3.Creación de VLAN

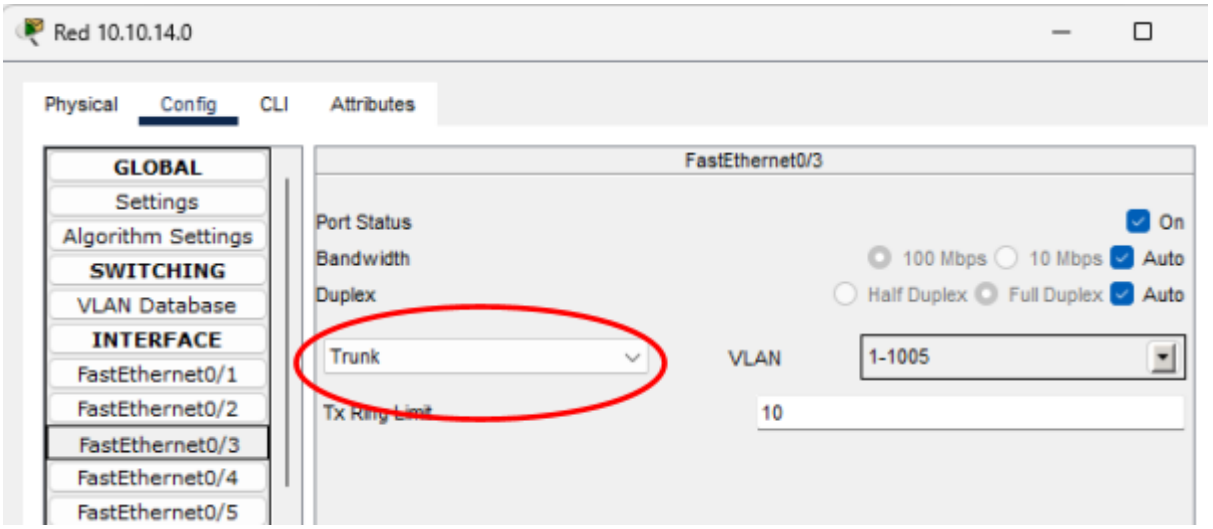
En el switch de cada Vlan usaremos este comando para crear las VLAN:

```
configure terminal
vlan <ID_VLAN>
name <NOMBRE_VLAN>
```

Y para asignar las VLAN a una interfaz usaremos:

```
configure terminal
interface <INTERFAZ>
switchport mode access
switchport access vlan <ID_VLAN>
```

Además de crear y asignar las Vlans tendremos que configurar el puerto del switch que se conectará al switch de capa 3 en modo trunk, lo haremos de manera gráfica para mayor comodidad.



4.1.4.Asignar Vlan a las interfaces del switch capa 3

Para crear las VLAN seguiremos el mismo proceso que en el paso 3.

```
configure terminal
vlan <ID_VLAN>
name <NOMBRE_VLAN>
```

Y para asignarles una ip a cada VLAN usaremos:

```
configure terminal
interface vlan <ID_VLAN>
ip address <IP> <MASCARA>
no shutdown
exit
```

Además tendremos que asignar una interfaz a las VLAN:

```
configure terminal
interface <INTERFAZ>
switchport mode access
switchport access vlan <ID_VLAN>
exit
```

Y para que todo funcione tendremos que activar el routing en el switch capa 3 para que tome las funciones de un router:

```
configure terminal
ip routing
exit
```

4.2.Bloquear el tráfico entre Vlan 10 y 20

Después de hacer los pasos anteriores habrá conectividad entre todas las Vlan, sin embargo el enunciado del ejercicio especifica que la Vlan de 1ASIR3 no puede comunicarse con 2ASIR3, para ello utilizaremos ACL dentro del switch de capa 3 .

```
configure terminal
ip access-list extended BLOQUEO_RED
deny ip <RED_1> <WILDCARD_1> <RED_2> <WILDCARD_2>
deny ip <RED_2> <WILDCARD_2> <RED_1> <WILDCARD_1>
permit ip any any
exit
```

Y aplicaremos estas reglas en sus respectivas Vlan:

```
interface vlan <ID_VLAN_1>
  ip access-group BLOQUEO_RED in
  exit
```

```
interface vlan <ID_VLAN_2>
ip access-group BLOQUEO_RED in
exit
```

Una vez hecho esto no debería haber conexión entre la Vlan 10 y 20 .

4.3.Configuración de VLAN en proxmox

4.3.1.Modificación de archivo /network/interfaces

Para crear los bridges y las 3 VLANs necesarias tendremos que modificar el archivo /etc/network/interfaces para usar el bridge por defecto (vmbr0) y crearle 3 VLANs adicionales además de asignarles las IPs.

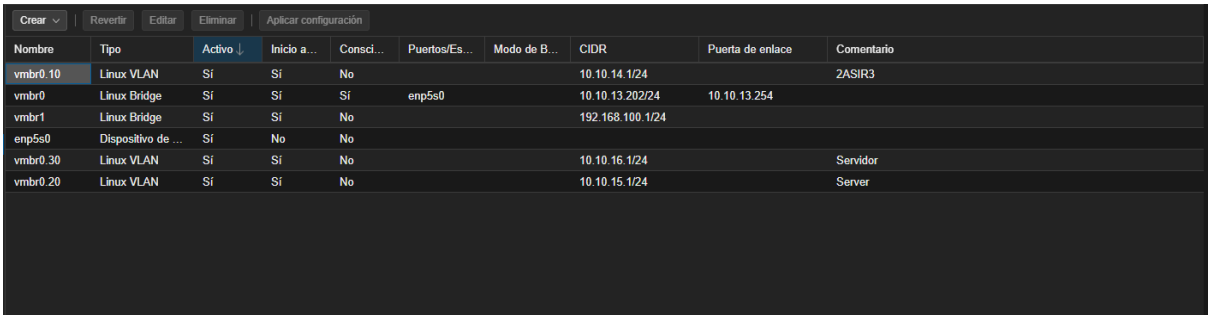
```
auto lo

iface lo inet loopback

iface enp5s0 inet manual

auto vmbr0

iface vmbr0 inet static

        address 10.10.13.202/24

        gateway 10.10.13.254

        bridge-ports enp5s0

        bridge-stp off

        bridge-fd 0

        bridge-vlan-aware yes

        bridge-vids 2-4094

#1ASIR3

auto vmbr0.10

iface vmbr0.10 inet static
```

```
        address 10.10.14.1/24

#2ASIR3

auto vmbr0.20

iface vmbr0.20 inet static

        address 10.10.15.1/24

#Server

auto vmbr0.30

iface vmbr0.30 inet static

        address 10.10.16.1/24
```

Después de modificar el archivo toca aplicarlo, para ello usaremos el comando:

```
systemctl restart networking
```

Después de aplicar la configuración podremos ver como en el apartado de red aparece el bridge y sus respectivas VLANs.



Nombre	Tipo	Activo	Inicio a...	Consci...	Puertos/Es...	Modo de B...	CIDR	Puerta de enlace	Comentario
vmbr0.10	Linux VLAN	Sí	Sí	No			10.10.14.1/24		2ASIR3
vmbr0	Linux Bridge	Sí	Sí	Sí	enp5s0		10.10.13.202/24	10.10.13.254	
vmbr1	Linux Bridge	Sí	Sí	No			192.168.100.1/24		
enp5s0	Dispositivo de ...	Sí	No	No					
vmbr0.30	Linux VLAN	Sí	Sí	No			10.10.16.1/24		Servidor
vmbr0.20	Linux VLAN	Sí	Sí	No			10.10.15.1/24		Server

4.3.2.Asignación de IPs en máquinas virtuales

Ya tenemos las VLANs creadas, ahora solo toca aplicarlas en las máquinas virtuales, para ello entraremos en las máquinas virtuales y modificaremos el archivo: /etc/netplan/01-netcfg.yaml.

```
network:
 version: 2
 ethernets:
  ens18:
   dhcp4: false
   addresses:
    - 10.10.16.2/24
   gateway4: 10.10.16.1
   nameservers:
    addresses:
     - 8.8.8.8
     - 8.8.4.4
```

En este archivo configuraremos la IP del dispositivo además de sus dns para poder conectarse a internet, tendremos que aplicar estas configuraciones en cada una de las máquinas, una vez configuradas las máquinas aplicaremos las configuraciones con el comando:

```
sudo netplan apply
```

4.3.3.Habilitar conexión a internet

Para poder conectar las máquinas a internet tendremos que aplicar el comando:

```
iptables -t nat -A POSTROUTING -o enp5s0 -j MASQUERADE
```

4.3.4.Bloquear tráfico entre Vlan 10 y Vlan 20

Para bloquear la conexión entre las Vlans usaremos unas configuraciones de ip tables:

```
iptables -A FORWARD -i vmbr0.10 -o vmbr0.20 -j DROP
```

```
iptables -A FORWARD -i vmbr0.20 -o vmbr0.10 -j DROP
```

4.4.Configurar el Redireccionamiento de Puertos con iptables

Escenario de Red

- **Subred del centro:** 10.10.13.0/24
- **IP de Proxmox:** 10.10.13.202
- **IP del Servidor (Web, NFS y VPN):** 10.10.16.2 (dentro de una VLAN en Proxmox)

4.4.1 Habilitar el reenvío de paquetes en el kernel

Ejecutamos el siguiente comando para asegurarnos de que el reenvío de paquetes está activado:

```
sudo echo "net.ipv4.ip_forward=1" >> /etc/sysctl.conf
sudo sysctl -p
```

4.4.2 Configurar iptables en Proxmox

Ejecutamos los siguientes comandos para redirigir los puertos desde Proxmox (10.10.13.202) hacia el servidor (10.10.16.2):

4.4.2.1.Redirigir HTTP (80) y HTTPS (443) para el servidor web:

```
sudo iptables -t nat -A PREROUTING -i vmbr0 -p tcp --dport 80 -j DNAT
--to-destination 10.10.16.2:80
sudo iptables -t nat -A PREROUTING -i vmbr0 -p tcp --dport 443 -j DNAT
--to-destination 10.10.16.2:443
```

4.4.2.2.Redirigir el tráfico SSH (22) hacia el servidor:

```
sudo iptables -t nat -A PREROUTING -i vmbr0 -p tcp --dport 2222 -j DNAT
--to-destination 10.10.16.2:22
```

(De esta forma, para conectarnos por SSH desde fuera usaremos el puerto 2222 en Proxmox).

4.4.2.3.Redirigir el puerto del servidor NFS (2049):

```
sudo iptables -t nat -A PREROUTING -i vmbr0 -p tcp --dport 2049 -j DNAT
--to-destination 10.10.16.2:2049
```

4.4.2.4.Redirigir el puerto de la VPN (1194 UDP, el puerto de nuestra utilidad VPN, OpenVPN):

```
sudo iptables -t nat -A PREROUTING -i vmbr0 -p udp --dport 1194 -j DNAT
--to-destination 10.10.16.2:1194
```

4.4.2.5.Hacer NAT para permitir el tráfico de vuelta:

```
sudo iptables -t nat -A POSTROUTING -s 10.10.16.2 -o vmbr0 -j MASQUERADE
```

4.4.3 Guardar las reglas de iptables

Para que los cambios sean permanentes, guardamos las reglas (opcional):

```
sudo apt install iptables-persistent
sudo netfilter-persistent save
sudo netfilter-persistent reload
```

4.5.Verificar la Configuración

Para comprobar que las reglas están aplicadas correctamente:

```
sudo iptables -t nat -L -n -v
```

Para probar la conectividad desde una máquina externa:

```
curl http://10.10.13.202
```

Para probar SSH:

```
ssh -p 2222 usuario@10.10.13.202
```

Inciso

Siguiendo estos pasos, hemos configurado el redireccionamiento de puertos en Proxmox, asegurando que los servicios del servidor sean accesibles externamente mediante la IP de Proxmox. Además, hemos implementado el soporte para VLANs y garantizado la persistencia de las reglas de iptables tras reinicios del sistema. Con esta configuración, podemos gestionar los accesos de manera eficiente y segura. Asimismo y para facilitar futuros cambios en el servidor Proxmox, hemos decidido implementar un script que se ejecutará automáticamente al encenderse el servidor con todas estas reglas, garantizando que si en algún momento se desea hacer algún cambio o hay algo que no funciona bien, podamos corregir dichos cambios e inconvenientes. He aquí el script utilizado:

```
#!/bin/bash

# Definir variables
PROXMOX_IP="10.10.13.202"
SERVER_IP="10.10.16.2"

# Habilitar el reenvío de paquetes
echo "Habilitando el reenvío de paquetes..."
echo 1 > /proc/sys/net/ipv4/ip_forward
sysctl -w net.ipv4.ip_forward=1

# Limpiar reglas previas
iptables -t nat -F
iptables -F

# Configurar NAT (masquerade) para permitir tráfico entre VLANs
iptables -t nat -A POSTROUTING -o vmbr0 -j MASQUERADE

# Redirigir puertos de Proxmox a la máquina del servidor
# HTTP (80)
iptables -t nat -A PREROUTING -p tcp --dport 80 -d $PROXMOX_IP -j DNAT
--to-destination $SERVER_IP:80
iptables -A FORWARD -p tcp -d $SERVER_IP --dport 80 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT
```

```
# HTTPS (443)
iptables -t nat -A PREROUTING -p tcp --dport 443 -d $PROXMOX_IP -j DNAT
--to-destination $SERVER_IP:443
iptables -A FORWARD -p tcp -d $SERVER_IP --dport 443 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# SSH (22)
iptables -t nat -A PREROUTING -p tcp --dport 2222 -d $PROXMOX_IP -j DNAT
--to-destination $SERVER_IP:22
iptables -A FORWARD -p tcp -d $SERVER_IP --dport 22 -m state --state
NEW,ESTABLISHED,RELATED -j ACCEPT

# OpenVPN (1194/udp)
sudo iptables -t nat -A PREROUTING -i vmbr0 -p udp --dport 1194 -j DNAT
--to-destination 10.10.16.2:1194

# Bloquear tráfico entre VLAN 10 y VLAN 20
iptables -A FORWARD -i vmbr0.10 -o vmbr0.20 -j DROP
iptables -A FORWARD -i vmbr0.20 -o vmbr0.10 -j DROP

#Habilitar puerto 3306 para MySQL
pve-firewall rule add -p tcp --dport 3306 -A ACCEPT

# Guardar reglas para que persistan tras reinicio
#apt install -y iptables-persistent
#netfilter-persistent save

# Mostrar reglas aplicadas
iptables -t nat -L -n -v
echo "Reglas aplicadas correctamente."
```

5. Instalar una VPN para Acceder desde Fuera

5.1 Introducción: ¿Qué es una VPN y por qué usarla?

¿Qué es una VPN?

Una VPN (Virtual Private Network, o Red Privada Virtual) es una tecnología que nos permite crear una conexión segura y cifrada entre un dispositivo (como un ordenador, móvil o tablet) y una red privada (como la red de nuestro servidor Ubuntu en Proxmox). Esta conexión actúa como un "túnel" seguro a través de Internet, protegiendo los datos que enviamos y recibimos.

Ventajas de usar una VPN

1. Seguridad:
    - Cifrado de datos: Toda la información que viaja a través de la VPN está cifrada, lo que la protege de posibles interceptaciones.
    - Protección contra ataques: Al acceder a la red interna a través de una VPN, reducimos el riesgo de ataques externos, ya que la conexión es privada y segura.
2. Acceso remoto:
    - Conectividad desde cualquier lugar: Nos permite acceder a los recursos de la red interna (como servidores, aplicaciones o archivos) desde fuera de la red local, como si estuviéramos físicamente en el lugar.
    - Flexibilidad: Es ideal para trabajar de manera remota o para dar acceso a usuarios externos de forma controlada.
3. Privacidad:
    - Ocultación de la IP: La VPN oculta nuestra dirección IP real, lo que aumenta la privacidad y el anonimato en Internet.
    - Evita el rastreo: El cifrado de la VPN dificulta que terceros rastreen nuestra actividad en línea.
4. Control de acceso:
    - Acceso restringido: Solo los usuarios autorizados (con los certificados y credenciales adecuados) pueden conectarse a la VPN.

- Segmentación de redes: Podemos limitar el acceso a ciertos recursos dentro de la red, lo que mejora la seguridad.

¿Por qué es una buena idea implementar una VPN en este proyecto?

En nuestro proyecto, queremos dar a conocer el ciclo de Administración de Sistemas Informáticos en Red (ASIR) y mostrar cómo se pueden implementar soluciones prácticas y seguras en un entorno real. Implementar una VPN nos permite:

- Demostrar habilidades técnicas: Configurar una VPN es una tarea avanzada que muestra nuestro dominio de redes, seguridad y sistemas.
- Proporcionar acceso seguro: Si queremos que los visitantes de las jornadas de puertas abiertas accedan a la página web o a otros recursos de forma segura, la VPN es la solución ideal.
- Fomentar la seguridad: Al usar una VPN, promovemos la importancia de la seguridad y la privacidad en el mundo digital, algo clave en la
- formación de un administrador de sistemas.

5.2 Conceptos clave: Certificados, PKI, CA, Diffie-Hellman, Tunneling, Reglas de IPTables y Autenticación y Autorización

5.2.1. Certificados Digitales

- Definición: Archivos digitales que utilizamos para verificar la identidad de un dispositivo, servidor o usuario en una red. Funcionan como una especie de "credencial digital" que garantiza que una entidad es quien dice ser.

- Importancia: Los certificados son esenciales para garantizar la autenticidad y seguridad de las conexiones VPN. Sin ellos, no podríamos confiar en que las conexiones sean seguras.

5.2.2. PKI (Infraestructura de Clave Pública)

- Definición: Un sistema que gestiona la creación, distribución y revocación de certificados digitales.
- Importancia: La PKI es la base sobre la cual se construye la seguridad de la VPN. Nos permite generar y gestionar los certificados necesarios para autenticar a los dispositivos y cifrar las comunicaciones.

5.2.3. CA (Autoridad Certificadora)

- Definición: Una entidad de confianza que emite y gestiona certificados digitales.
- Importancia: La CA es quien garantiza que los certificados son válidos y pertenecen a una entidad específica. En nuestro caso, creamos nuestra propia CA para emitir certificados para el servidor y los clientes.

5.2.4. Diffie-Hellman

- Definición: Un protocolo criptográfico que permite a dos partes generar una clave secreta compartida sobre un canal no seguro.
- Importancia: Este protocolo es crucial para el intercambio seguro de claves en la VPN. Garantiza que, incluso si alguien intercepta el tráfico, no podrá descifrarlo sin la clave compartida.

5.2.5. Tunneling

- Definición: La creación de un "túnel" seguro a través del cual viajan los datos entre el cliente y el servidor VPN.
- Importancia: El tunneling es lo que permite que los datos viajen de manera segura a través de Internet, protegiéndonos de posibles interceptaciones. En una VPN, los datos se encapsulan dentro de paquetes cifrados que viajan a través de este túnel.

5.2.6. Reglas de IPTables

- Definición: IPTables es una herramienta en Linux que nos permite configurar reglas de firewall para controlar el tráfico de red.
- Importancia: En nuestro caso, utilizamos reglas de IPTables para:
    - Permitir el tráfico de la VPN: Aseguramos que el tráfico entre el cliente y el servidor VPN no sea bloqueado.

- Habilitar el reenvío de IP: Permitimos que el tráfico de la VPN fluya entre el cliente y la red interna.
- Proteger el servidor: Bloqueamos accesos no autorizados al servidor VPN.

5.2.7. Autenticación y Autorización

- Definición:
  - Autenticación: Verificar la identidad de un usuario o dispositivo (por ejemplo, mediante certificados digitales).
  - Autorización: Determinar a qué recursos puede acceder un usuario o dispositivo autenticado.
- Importancia: En la VPN, la autenticación se realiza mediante certificados digitales, lo que garantiza que sólo los dispositivos autorizados puedan conectarse. La autorización se gestiona mediante reglas de red y permisos, asegurando que los usuarios solo accedan a los recursos permitidos.

5.3 Creación del servidor VPN

Paso 1: Preparación del servidor Ubuntu

1. Accedemos al servidor Ubuntu:
   - Nos conectamos al servidor Ubuntu (10.10.16.2) mediante SSH o directamente desde la consola de Proxmox.
2. Actualizamos el sistema:
   - Ejecutamos los siguientes comandos para asegurarnos de que el sistema esté actualizado:

```
sudo apt update
sudo apt upgrade -y
```

Paso 2: Instalación de OpenVPN

1. Instalamos OpenVPN y Easy-RSA:

- Easy-RSA es una herramienta que nos ayudará a gestionar los certificados SSL necesarios para la VPN.
- Ejecutamos el siguiente comando:

```
sudo apt install openvpn easy-rsa -y
```

- Explicación del comando:
  - **sudo apt install openvpn easy-rsa -y:** Instala los paquetes openvpn (para la VPN) y easy-rsa (para gestionar certificados).

2. Configuramos Easy-RSA:

- Copiamos la plantilla de Easy-RSA a un directorio de trabajo:

```
mkdir ~/easy-rsa
cp -r /usr/share/easy-rsa/* ~/easy-rsa/
cd ~/easy-rsa
```

- Editamos el archivo de configuración de variables:

```
nano vars
```

- Añadimos o modificamos las siguientes líneas con nuestros datos:

```
export KEY_COUNTRY="ES"
export KEY_PROVINCE="Gipuzkoa"
export KEY_CITY="Irun"
export KEY_ORG="Plaiaundi"
export KEY_EMAIL="ikdxz@plaiaundi.net"
export KEY_OU="ASIR"
export KEY_NAME="servidor"
```

3. Generamos los certificados y claves:

- Inicializamos el PKI (Public Key Infrastructure):

```
./easyrsa init-pki
```

- Explicación del comando:

- **./easyrsa init-pki:** Inicializa la infraestructura de clave pública (PKI) en el directorio actual.

- Generamos la CA (Autoridad Certificadora):

```
./easyrsa build-ca
```

- Explicación del comando:

  - **./easyrsa build-ca:** Crea la Autoridad Certificadora (CA) y genera los certificados raíz. Nos pedirá una contraseña para proteger la CA.

- Generamos el certificado y la clave para el servidor:

```
./easyrsa gen-req 10.10.16.2 nopass
./easyrsa sign-req server 10.10.16.2
```

- Explicación de los comandos:

  - **./easyrsa gen-req server nopass:** Genera una solicitud de certificado para el servidor sin contraseña (nopass).
  - **./easyrsa sign-req server server:** Firma la solicitud de certificado para el servidor.

- Generamos el certificado Diffie-Hellman (necesario para el intercambio de claves):

```
./easyrsa gen-dh
```

- Explicación del comando:

  - **./easyrsa gen-dh:** Genera el archivo de parámetros Diffie-Hellman, que se utiliza para el intercambio seguro de claves.

- Generamos la clave HMAC (para mejorar la seguridad):

```
openvpn --genkey secret ta.key
```

- Explicación del comando:

- **openvpn --genkey ta.key:** Genera una clave HMAC (Hash-based Message Authentication Code) para proteger contra ataques de repetición.

4. Movemos los archivos generados:

    - Copiamos los archivos generados al directorio de configuración de OpenVPN:

```
sudo cp pki/ca.crt pki/issued/10.10.16.2.crt pki/private/10.10.16.2.key pki/dh.pem ta.key /etc/openvpn/server/
```

Paso 3: Configuración del servidor OpenVPN

1. Creamos el archivo de configuración del servidor:

    - Copiamos la plantilla de configuración:

```
sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf /etc/openvpn/server/
```

    - Explicación de los comandos:

        - **sudo cp ... /etc/openvpn/server/:** Copia la plantilla de configuración del servidor.

    - Editamos el archivo de configuración:

```
sudo nano /etc/openvpn/server/server.conf
```

    - Aseguramos que las siguientes líneas estén configuradas correctamente:

```
port 1194
proto udp
dev tun
ca /etc/openvpn/server/ca.crt
cert /etc/openvpn/server/10.10.16.2.crt
key /etc/openvpn/server/10.10.16.2.key
dh /etc/openvpn/server/dh.pem
server 10.8.0.0 255.255.255.0
push "redirect-gateway def1 bypass-dhcp"
```

```
push "dhcp-option DNS 8.8.8.8"
keepalive 10 120
tls-auth /etc/openvpn/server/ta.key 0
cipher AES-256-GCM
auth SHA256
data-ciphers AES-256-GCM:AES-128-GCM
auth SHA256
data-ciphers AES-256-GCM:AES-128-GCM
user nobody
group nogroup
persist-key
persist-tun
status openvpn-status.log
verb 3
```

Resumen de la configuración	
Parámetro	**Descripción**
port 1194	Puerto en el que el servidor escucha conexiones.
proto udp	Protocolo de transporte (UDP o TCP).
dev tun	Tipo de dispositivo de red virtual (tun para IP, tap para Ethernet).
ca ca.crt	Certificado de la Autoridad Certificadora (CA).
cert server.crt	Certificado del servidor.
key server.key	Clave privada del servidor.
dh dh.pem	Parámetros Diffie-Hellman para el intercambio de claves.
server 10.8.0.0 255.255.255.0	Subred para asignar direcciones IP a los clientes.
push "redirect-gateway def1"	Redirige todo el tráfico del cliente a través de la VPN.
push "dhcp-option DNS 8.8.8.8"	Mecanismo para mantener la conexión activa.

keepalive 10 120	Mecanismo para mantener la conexión activa.
tls-auth ta.key 0	Autenticación TLS adicional para mayor seguridad.
cipher AES-256-GCM	Algoritmo de cifrado utilizado.
user nobody	Grupo bajo el cual se ejecuta OpenVPN.
group nogroup	Grupo bajo el cual se ejecuta OpenVPN.
persist-key	Evita la relectura de la clave privada en reinicios.
persist-key	Evita el cierre y reapertura de la interfaz TUN/TAP en reinicios.
status openvpn-status.log	Nivel de detalle de los registros (logs).
verb 3	Nivel de detalle de los registros (logs).

2. Habilitamos el reenvío de IP:
    - Editamos el archivo de configuración de red:

```
sudo nano /etc/sysctl.conf
```

    - Aseguramos que la siguiente línea esté descomentada:

```
net.ipv4.ip_forward=1
```

    - Aplicamos los cambios:

```
sudo sysctl -p
```

    - Explicación de los comandos:
        - **sudo sysctl -p:** Aplica los cambios en la configuración del kernel.

3. Configuramos las reglas de iptables:
    - Añadimos las reglas necesarias para permitir el tráfico a través de la VPN:

```
sudo iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o ens18 -j MASQUERADE
```

- Guardamos las reglas para que persistan después de un reinicio:

```
sudo apt install iptables-persistent
sudo netfilter-persistent save
```

- Explicación de los comandos:

  - **sudo iptables -t nat -A POSTROUTING ...:** Añade una regla de NAT para redirigir el tráfico de la VPN.
  - **sudo apt install iptables-persistent:** Instala el paquete para guardar las reglas de iptables.
  - **sudo netfilter-persistent save:** Guarda las reglas actuales de iptables.

4. Habilitamos y reiniciamos OpenVPN:

   - Habilitamos el servicio para que se inicie automáticamente:

```
sudo systemctl enable openvpn@server
```

   - Reiniciamos el servicio:

```
sudo systemctl restart openvpn@server
```

   - Explicación de los comandos:

     - **sudo systemctl enable openvpn@servidor:** Habilita el servicio de OpenVPN para que se inicie automáticamente.
     - **sudo systemctl restart openvpn@servidor:** Reinicia el servicio de OpenVPN.

Paso 4: Configuración de los clientes

1. Generamos los certificados para los clientes:

   - Volvemos al directorio de Easy-RSA:

```
cd ~/easy-rsa
```

   - Generamos un certificado para el cliente (por ejemplo, cliente1):

```
./easyrsa gen-req cliente1 nopass
```

```
./easyrsa sign-req client cliente1
```

- Explicación de los comandos:
  - **./easyrsa gen-req asir1 nopass:** Genera una solicitud de certificado para el cliente sin contraseña.
  - **./easyrsa sign-req client asir1:** Firma la solicitud de certificado para el cliente.

2. Creamos el archivo de configuración del cliente:
   - Copiamos la plantilla de configuración:

```
cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf
~/cliente1.ovpn
```

   - Editamos el archivo:

```
nano ~/cliente1.ovpn
```

   - Configuramos las siguientes líneas:

```
client
dev tun
proto udp
remote 10.10.13.202 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
cert cliente1.crt
key cliente1.key
tls-auth ta.key 1
cipher AES-256-GCM
auth SHA256
verb 3
```

3. Transferimos el archivo de configuración al cliente:

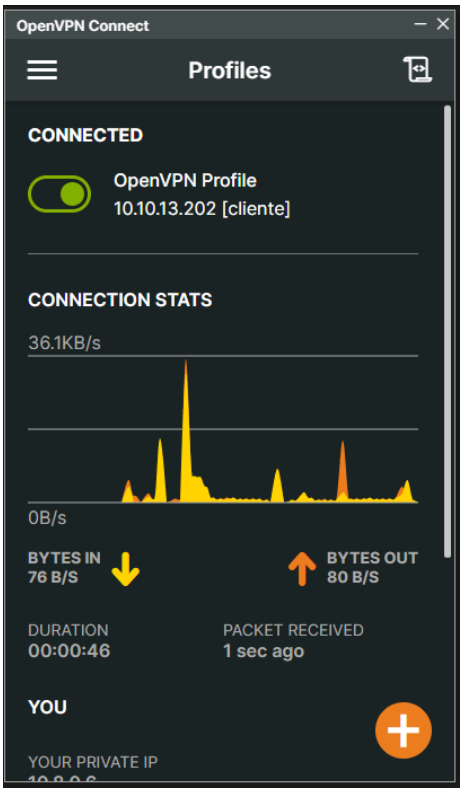
- Copiamos el archivo cliente1.ovpn y los archivos de certificados (ca.crt, cliente1.crt, cliente1.key, ta.key) al dispositivo del cliente.

Paso 5: Conexión desde el cliente

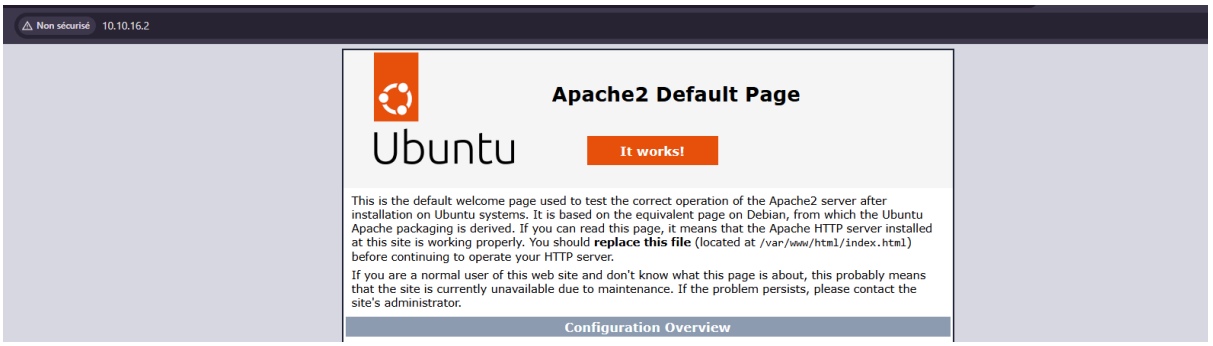
1. Instalamos OpenVPN en el cliente:
   - En el dispositivo del cliente, instalamos OpenVPN.
2. Importamos la configuración:
   - Importamos el archivo cliente1.ovpn en el cliente OpenVPN.
3. Conectamos a la VPN:
   - Iniciamos la conexión VPN desde el cliente. Si todo está bien configurado, el cliente debería conectarse al servidor y tener acceso a la red interna.

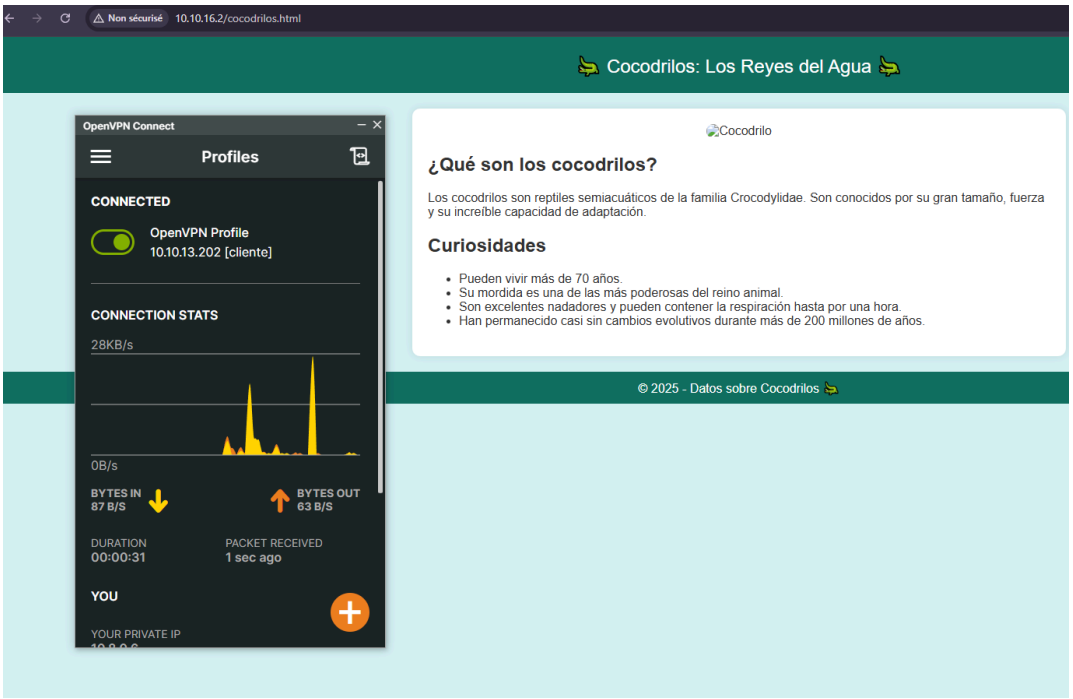
Paso 6: Verificación

1. Comprobamos la conexión:
   - Desde el cliente, intentamos acceder a recursos internos (por ejemplo, la página web en 10.10.16.2).
   - Verificamos que el tráfico esté pasando a través de la VPN.




```
Carte inconnue Conexión de área local :

   Suffixe DNS propre à la connexion. . . :
   Adresse IPv6 de liaison locale. . . . .: fe80::1007:3ed3:4936:1001%32
   Adresse IPv4. . . . . . . . . . . . . .: 10.8.0.6
   Masque de sous-réseau. . . . . . . . . : 255.255.255.252
   Passerelle par défaut. . . . . . . . . :

Carte Ethernet Ethernet :
```





6. Base de Datos y Modelado

6.1 Diagrama de Entidad-Relación

El sistema diseñado busca modelar la información académica del instituto Plaiaundi de una manera clara y estructurada. Este modelo tiene como objetivo gestionar datos relacionados con los ciclos formativos, los módulos que los componen, los profesores que los imparten, los alumnos que los cursan y las calificaciones obtenidas en los diferentes módulos. A continuación, se detalla el propósito de cada entidad, los datos que contiene y las relaciones que existen entre ellas.

En primer lugar, los ciclos formativos representan las titulaciones ofertadas en el instituto. Cada ciclo formativo está compuesto por varios módulos y contiene información como un identificador único, su nombre, la duración del ciclo y una descripción. Esto permite al sistema organizar claramente la estructura académica y relacionarla con otras entidades.

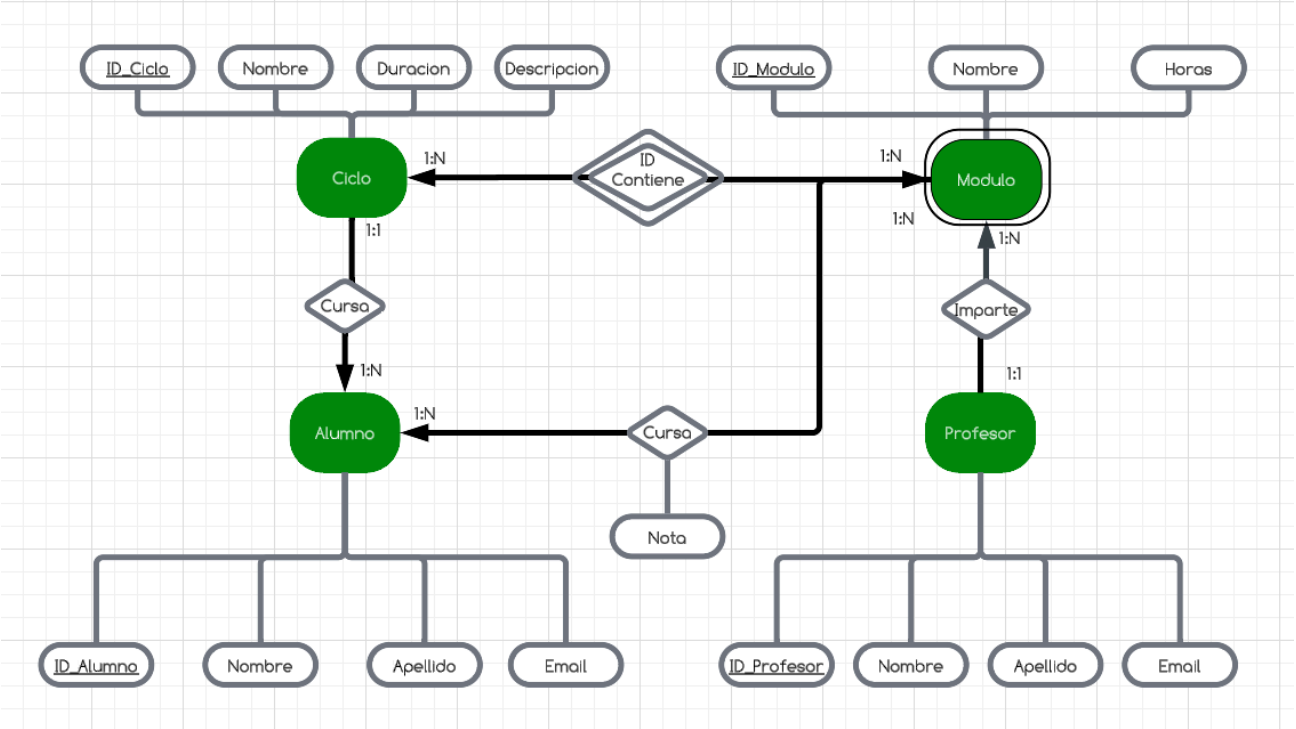
Los módulos son las asignaturas que conforman los ciclos formativos. Cada módulo tiene un identificador único, un nombre, la cantidad de horas asignadas y está vinculado a un ciclo formativo. Esto significa que cada ciclo puede tener varios módulos, y que cada módulo puede tener varios ciclos formativos distintos.

Los profesores son los responsables de impartir los módulos. En este modelo, cada profesor tiene un identificador único, su nombre y un correo electrónico de contacto. Un módulo puede ser impartido por más de un profesor, y cada profesor puede estar asociado a varios módulos, lo que refleja la flexibilidad en la asignación de docentes.

Por otro lado, los alumnos son los estudiantes matriculados en el instituto. Cada alumno tiene un identificador único, un nombre, un email y está vinculado a un ciclo formativo en el que está inscrito. Esto permite rastrear a qué ciclo pertenece cada estudiante y, a través de las relaciones del modelo, vincularlo a los módulos y calificaciones correspondientes.

Las notas representan las calificaciones obtenidas por los alumnos en los módulos que cursan.

Este diseño facilita la organización y consulta de la información académica del instituto, proporcionando una estructura clara para almacenar y acceder a datos esenciales como notas, asignaturas y vínculos entre estudiantes, profesores y ciclos formativos.



6.2 Modelo relacional normalizado

6.2.1.Aplicación de las Formas Normales:

1. Primera Forma Normal (1NF):

Todas las tablas cumplen con la 1NF:

- No hay grupos repetidos.
- Cada columna contiene valores atómicos.
- Cada tabla tiene una clave primaria.

2. **Segunda Forma Normal (2NF):**

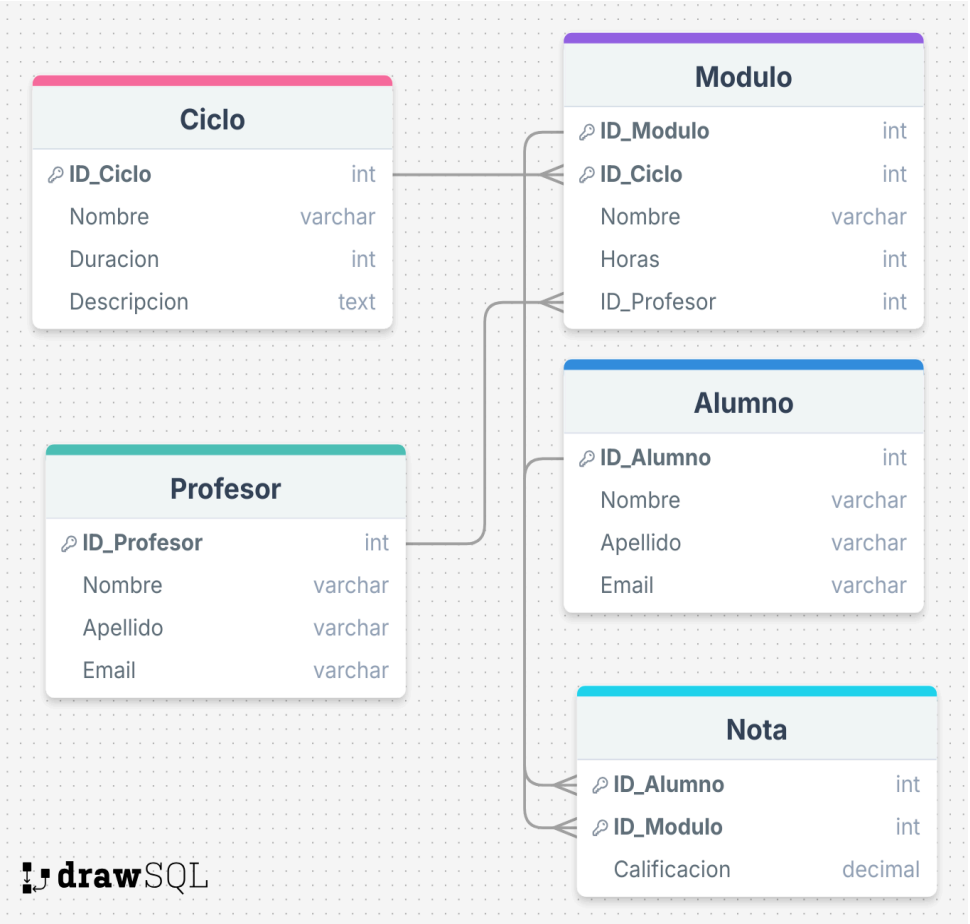
   Todas las tablas cumplen con la 2NF:

   - No hay dependencias parciales (todos los atributos no clave dependen completamente de la clave primaria).

3. **Tercera Forma Normal (3NF):**

   Todas las tablas cumplen con la 3NF:

   - No hay dependencias transitivas (los atributos no clave no dependen de otros atributos no clave).



6.3. Diseño Físico

Ciclo
En la tabla *ciclo*, hemos añadido el atributo *id_ciclo* para facilitar la identificación de los ciclos. Para ello, hemos definido este atributo como un tipo de dato *CHAR* con un máximo de 4 caracteres, dado que al tratarse de abreviaturas no es necesario asignar un tamaño mayor. En cuanto al atributo *duración*, hemos utilizado el tipo de dato *INT*, ya que es necesario almacenar un valor numérico grande para representar la duración de los ciclos.

Alumno
En la tabla *alumnos*, el identificador es de tipo *INT AUTO_INCREMENT*, lo que garantiza que al añadir nuevos alumnos a la tabla, se numerarán automáticamente, simplificando el proceso de gestión de registros.

Profesor
En la tabla *profesor*, el identificador es también de tipo *INT AUTO_INCREMENT*. Esto permite que, al añadir nuevos registros para los profesores, la clave primaria se asigne automáticamente, asegurando la consistencia de los datos.

Módulo
La tabla *módulo* combina dos claves primarias, ya que es una entidad débil. Estas claves son la de la tabla *ciclo* (ID_Ciclo) y la propia clave del módulo (ID_Modulo). El tipo de dato de *ID_Modulo* es *CHAR(6)*, lo que nos permite identificar los módulos mediante sus iniciales. Además, cuenta con el atributo *Horas*, que se ha definido como *DECIMAL(5,2)*, permitiendo almacenar valores decimales. Esto es necesario debido a que las horas pueden ser fraccionadas, especialmente cuando se asignan valores "cortos" o decimales.

Nota
La tabla *nota* tiene una clave primaria compuesta por tres claves: *ID_Alumno*, *ID_Ciclo* y *ID_Modulo*. Estas tres claves son fundamentales para identificar la nota de un alumno, ya que necesitamos conocer su identificador, el ciclo que está cursando, y el módulo correspondiente. Además, el atributo *calificación* se define como *DECIMAL*, lo que nos permite almacenar notas con valores decimales.

6.4 Copias de seguridad y exportación de datos seleccionados

6.4.1. Crear una Copia de Seguridad de Todas las Bases de Datos de MySQL

Para realizar un backup de todas las bases de datos en MySQL, usamos el siguiente comando:

```
mysqldump -u servidor -p --all-databases > /home/servidor/Copia.sql
```

6.4.2. Crear un Directorio en el Servidor Proxmox para Almacenar el Backup

Antes de transferir el archivo de respaldo, es necesario asegurarse de que exista un directorio donde almacenarlo. Para ello, utilizamos el siguiente comando:

```
mkdir -p /var/lib/vz/backups
```

6.4.3. Transferir el Archivo de Backup al Servidor Proxmox

Una vez que tenemos el archivo de la copia de seguridad, lo podemos transferir al servidor Proxmox con el siguiente comando:

```
scp /home/servidor/Copia.sql root@10.10.13.202:/var/lib/vz/backups/
```

6.5. Exportar datos selecionados

6.5.1. Generar y visualizar el archivo .csv

```
SELECT columna1, columna2
FROM nombre_tabla
INTO OUTFILE 'C:\\Users\\2asir3\\Desktop\\Archivo.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY ""
LINES TERMINATED BY '\n';
```

Ir a C:\Users\2asir3\Desktop\AlumnosYNotas.csv y abrir con Excel o Bloc de notas.

6.6. Usuarios y permisos

- **DBA**: Administradores de la base de datos con acceso completo.
- **Trabajadores**: Usuarios que pueden consultar y modificar datos (INSERT, UPDATE, DELETE).
- **Visitantes**: Usuarios con solo acceso de lectura (SELECT).

2. **Permisos**:
   - Los **DBA** tienen privilegios totales sobre la base de datos.
   - Los **trabajadores** tienen permisos para consultar y hacer cambios.
   - Los **visitantes** solo pueden consultar los datos, sin hacer modificaciones.
3. **Acceso restringido por IP**:
   - Los usuarios solo pueden acceder desde las IPs especificadas o desde localhost.

DBA Los administradores de la base de datos (dba, dba1, dba2, dba3) tienen acceso completo a la base de datos:

```
CREATE USER 'dba'@'localhost' IDENTIFIED BY 'IJOdba';
GRANT ALL PRIVILEGES ON ASIR.* TO 'dba'@'localhost';

CREATE USER 'dba1'@'10.10.13.102' IDENTIFIED BY 'IJOdba';
GRANT ALL PRIVILEGES ON ASIR.* TO 'dba1'@'10.10.13.102';

CREATE USER 'dba2'@'10.10.13.132' IDENTIFIED BY 'IJOdba';
GRANT ALL PRIVILEGES ON ASIR.* TO 'dba2'@'10.10.13.132';

CREATE USER 'dba3'@'10.10.13.143' IDENTIFIED BY 'IJOdba';
GRANT ALL PRIVILEGES ON ASIR.* TO 'dba3'@'10.10.13.143';
```

Trabajadores Los trabajadores (trabajador,trabajador1, trabajador2, trabajador3) tienen permisos para consultar, insertar, actualizar y eliminar datos:

```
CREATE USER 'trabajador1'@'10.10.13.102' IDENTIFIED BY 'IJOtrabajador1';
GRANT SELECT, INSERT, UPDATE, DELETE ON ASIR.* TO 'trabajador1'@'10.10.13.102';

CREATE USER 'trabajador2'@'10.10.13.132' IDENTIFIED BY 'IJOtrabajador2';
GRANT SELECT, INSERT, UPDATE, DELETE ON ASIR.* TO 'trabajador2'@'10.10.13.132';
```

```
CREATE USER 'trabajador3'@'10.10.13.143' IDENTIFIED BY 'IJOtrabajador3';
GRANT SELECT, INSERT, UPDATE, DELETE ON ASIR.* TO 'trabajador3'@'10.10.13.143';

CREATE USER 'trabajador'@'localhost' IDENTIFIED BY 'IJOtrabajador';
GRANT SELECT, INSERT, UPDATE, DELETE ON ASIR.* TO 'trabajador'@'localhost';
```

Visitante Los visitantes (visitante,visitante1, visitante2, visitante3) solo tienen permisos de consulta, sin poder realizar modificaciones:

```
CREATE USER 'visitante1'@'10.10.13.102' IDENTIFIED BY 'IJOvisitante1';
GRANT SELECT ON ASIR.* TO 'visitante1'@'10.10.13.102';

CREATE USER 'visitante2'@'10.10.13.132' IDENTIFIED BY 'IJOvisitante2';
GRANT SELECT ON ASIR.* TO 'visitante2'@'10.10.13.132';

CREATE USER 'visitante3'@'10.10.13.143' IDENTIFIED BY 'IJOvisitante3';
GRANT SELECT ON ASIR.* TO 'visitante3'@'10.10.13.143';

CREATE USER 'visitante'@'localhost' IDENTIFIED BY 'IJOvisitante';
GRANT SELECT ON ASIR.* TO 'visitante'@'localhost';
```

7. Desarrollo de la Página Web

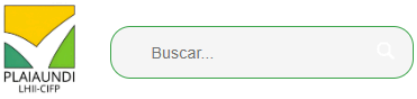
7.1. Estructura General del HTML

La página web está estructurada en lenguaje HTML y organiza el contenido en diversas secciones. La estructura incluye etiquetas de apertura y cierre, y un encabezado que contiene el título de la página, entre otros elementos.

7.2. Encabezado

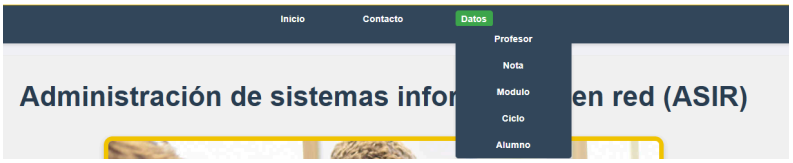
El encabezado de la página contiene los siguientes elementos:

- **LOGO:** Se encuentra dentro de un con la clase "logo" y se incluye una imagen que sirve como el logotipo del sitio.
- **BUSCADOR:** Es un campo de búsqueda que permite a los usuarios escribir para buscar contenido dentro del sitio web, acompañado de un pequeño icono de búsqueda.



7.3. Navegación

El menú de navegación permite al usuario acceder a diferentes secciones de la página. Este menú se encuentra dentro de un con la clase "nav-container". Se utiliza una lista no ordenada con enlaces a distintas secciones del sitio, además de un menú desplegable (Datos) que ofrece más opciones, como Profesor, Nota, Módulo, Ciclo y Alumno.



7.4. Contenido Principal

El contenido principal de la página se organiza en varias secciones:

- FRASE DESTACADA: Contiene el título "Administración de Sistemas Informáticos en Red (ASIR)".

- ASIGNATURAS: Se presentan varias asignaturas del curso. Cada asignatura se estructura con una imagen en la parte frontal y una

descripción en la parte posterior, utilizando un “flip” con las clases "asignatura" y "asignatura-inner".

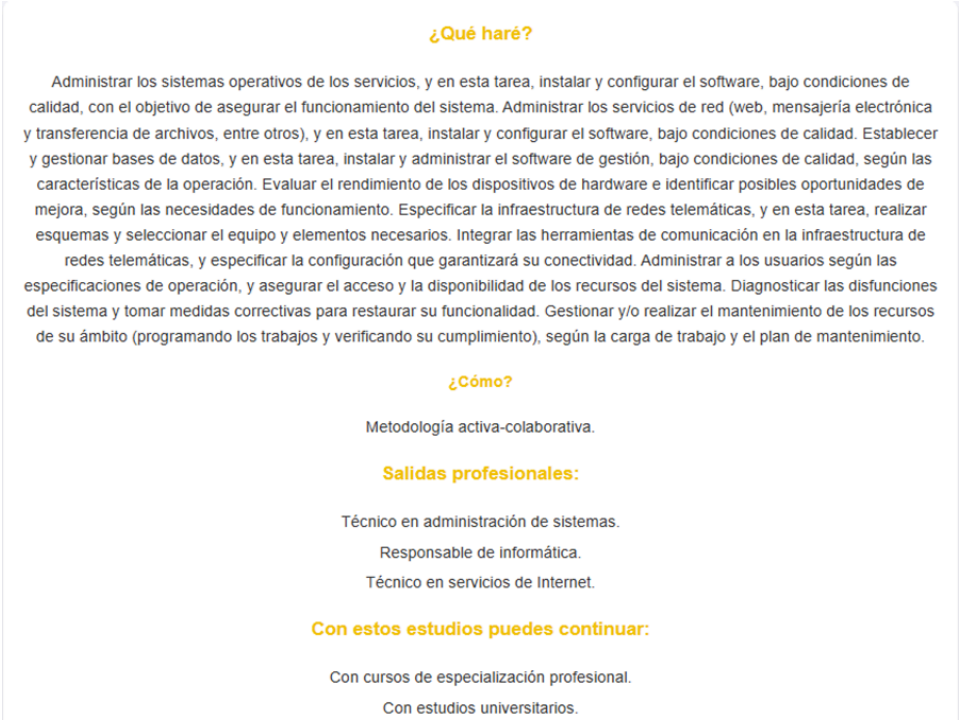
- TABLAS: Se muestran dos tablas con el plan de estudios, divididas en módulos y horas semanales de cada uno. Cada tabla está contenida en un con la clase "tablas-contenedor".

Plan de estudios (2000 Horas, 2 Cursos)

SEMANAL	1. MODULOS DEL CURSO
8h	Implantación de sistemas operativos
6h	Planificación de redes
6h	Administración de bases de datos
4h	Lenguaje de marcas y sistemas de información
4h	Itinerario personal para la empleabilidad

SEMANAL	2. MODULOS DEL CURSO
6h	Administración de sistemas operativos
6h	Servicios de red e Internet
5h	Implantación de aplicaciones web
3h	Administración de sistemas gestores de bases de datos
5h	Seguridad y alta disponibilidad
2h	Inglés Técnico
3h	Empresa e Iniciativa Emprendedora
5h	Proyecto de administración de sistemas informáticos en red. (Febrero a Junio, 50h)
35h	Formación en Centros de Trabajo. (Febrero a Junio, 360h)

- **TEXTO DESCRIPTIVO:** Un bloque con información adicional sobre qué se aprenderá, cómo se aprenderá, cómo se llevará a cabo el curso y las posibles salidas profesionales.



¿Qué haré?

Administrar los sistemas operativos de los servicios, y en esta tarea, instalar y configurar el software, bajo condiciones de calidad, con el objetivo de asegurar el funcionamiento del sistema. Administrar los servicios de red (web, mensajería electrónica y transferencia de archivos, entre otros), y en esta tarea, instalar y configurar el software, bajo condiciones de calidad. Establecer y gestionar bases de datos, y en esta tarea, instalar y administrar el software de gestión, bajo condiciones de calidad, según las características de la operación. Evaluar el rendimiento de los dispositivos de hardware e identificar posibles oportunidades de mejora, según las necesidades de funcionamiento. Especificar la infraestructura de redes telemáticas, y en esta tarea, realizar esquemas y seleccionar el equipo y elementos necesarios. Integrar las herramientas de comunicación en la infraestructura de redes telemáticas, y especificar la configuración que garantizará su conectividad. Administrar a los usuarios según las especificaciones de operación, y asegurar el acceso y la disponibilidad de los recursos del sistema. Diagnosticar las disfunciones del sistema y tomar medidas correctivas para restaurar su funcionalidad. Gestionar y/o realizar el mantenimiento de los recursos de su ámbito (programando los trabajos y verificando su cumplimiento), según la carga de trabajo y el plan de mantenimiento.

¿Cómo?

Metodología activa-colaborativa.

Salidas profesionales:

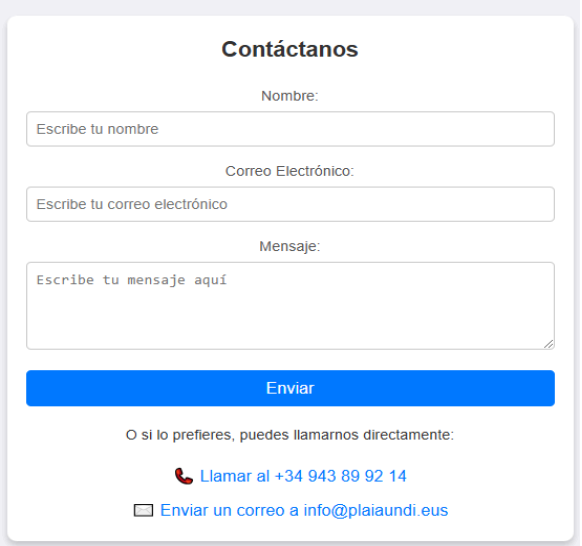
Técnico en administración de sistemas.
Responsable de informática.
Técnico en servicios de Internet.

Con estos estudios puedes continuar:

Con cursos de especialización profesional.
Con estudios universitarios.

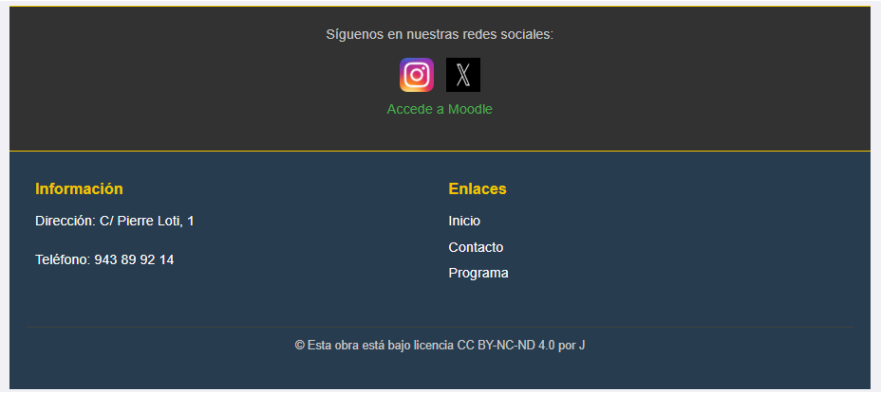
7.5. Formulario de contacto

Esta sección está diseñada para que los usuarios puedan ponerse en contacto con el sitio web. El formulario incluye varios campos, como nombre, correo electrónico y mensaje (para que el usuario pueda escribir sus inquietudes). Además, se proporcionan datos adicionales, como el número de teléfono y una dirección de correo electrónico.



7.6 Pie de página

El pie de página contiene información sobre cómo contactar con el sitio, enlaces a redes sociales y un enlace a Moodle para acceder a recursos adicionales. También incluye la dirección de la institución, el número de teléfono y otros enlaces importantes, como "Inicio" y "Programa".



8. Exportación y Validación de Datos

8.1. Exportación a XML

Una vez hemos terminado nuestra base de datos, tendremos que exportar cada una de las tablas que hemos creado (Profesor, Alumno, Nota, Módulo y Ciclo) a XML. Este formato estructurado nos permite

almacenar la información de manera organizada y legible para otras aplicaciones.

```
<?xml version="1.0" encoding="UTF-8"?>
<database name="asir">
  <table name="profesor">
    <column name="ID_Profesor" type="int" primaryKey="true"/>
    <column name="Nombre" type="varchar(30)"/>
    <column name="Apellido" type="varchar(30)"/>
    <column name="Email" type="varchar(30)"/>
    <data>
      <row>
        <ID_Profesor>1</ID_Profesor>
        <Nombre>JUAN</Nombre>
        <Apellido>CUESTA</Apellido>
        <Email>J.CUESTA@PLAIAUNDI.COM</Email>
      </row>
      <row>
        <ID_Profesor>2</ID_Profesor>
        <Nombre>SAMUEL</Nombre>
        <Apellido>DE LUQUE</Apellido>
        <Email>S.DELUQUE@PLAIAUNDI.COM</Email>
      </row>
      <row>
        <ID_Profesor>3</ID_Profesor>
        <Nombre>GUILLERMO</Nombre>
        <Apellido>GARCIA</Apellido>
        <Email>G.GARCIA@PLAIAUNDI.COM</Email>
      </row>
      <row>
        <ID_Profesor>4</ID_Profesor>
        <Nombre>ANTONIO</Nombre>
        <Apellido>ORTERA</Apellido>
        <Email>A.ORTERA@PLAIAUNDI.COM</Email>
      </row>
      <row>
        <ID_Profesor>5</ID_Profesor>
        <Nombre>LUCIA</Nombre>
        <Apellido>ALVAREZ</Apellido>
        <Email>L.ALVAREZ@PLAIAUNDI.COM</Email>
```

```
        </row>
      </data>
    </table>
</database>
```

Este XML es el de profesor pero tendremos que repetir el mismo proceso con todas las tablas que tenemos creadas.

8.2.Validación con XSD

Después de generar el XML, tendremos que usar el XSD para validar los datos. El XSD sirve para definir las reglas y la estructura del XML y se asegura de que los datos cumplan con un formato en específico. También evita errores como etiquetas mal escritas o datos incorrectos. Para ver si tenemos bien validado el XSD podremos usar la web FreeFormatter que nos ayudara de forma rapida a saber si esta bien.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
 <xs:element name="database">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="table">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="column" maxOccurs="unbounded">
        <xs:complexType>
         <xs:sequence>
          <xs:element name="name" type="xs:string"/>
          <xs:element name="type" type="xs:string"/>
          <xs:element name="primaryKey" type="xs:boolean" minOccurs="0"/>
         </xs:sequence>
        </xs:complexType>
       </xs:element>
       <xs:element name="data">
        <xs:complexType>
         <xs:sequence>
          <xs:element name="row" maxOccurs="unbounded">
           <xs:complexType>
            <xs:sequence>
```

```
                <xs:element name="ID_Profesor" type="xs:int"/>
                <xs:element name="Nombre" type="xs:string"/>
                <xs:element name="Apellido" type="xs:string"/>
                <xs:element name="Email" type="xs:string"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
 </xs:complexType>
</xs:element>
</xs:schema>
```

Este sería el XSD de la tabla profesor, también habría que hacer lo mismo con las demás tablas.

8.3.Transformación con XSLT

Hemos utilizado XSLT para convertir los datos XML en HTML. El XSLT extrae la información del XML y la reestructura en otro formato. Permite generar documentos en HTML a partir de los datos. Esto facilita la automatización para mostrar los datos en la web sin escribir código HTML manualmente

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
  <xsl:output method="html" indent="yes"/>
  <xsl:template match="/">
    <html>
      <head>
        <title>Tabla de Profesores</title>
        <style>
          table {
            border-collapse: collapse;
            width: 100%;
```

```
            font-family: Arial, sans-serif;
        }
        th, td {
            border: 1px solid #ddd;
            padding: 8px;
            text-align: left;
        }
        th {
            background-color: #f2f2f2;
            font-weight: bold;
        }
        tr:nth-child(even) {
            background-color: #f9f9f9;
        }
    </style>
</head>
<body>
    <h1>Profesores</h1>
    <table>
        <thead>
            <tr>
                <th>ID Profesor</th>
                <th>Nombre</th>
                <th>Apellido</th>
                <th>Email</th>
            </tr>
        </thead>
        <tbody>
            <xsl:for-each select="database/table/data/row">
                <tr>
                    <td><xsl:value-of select="ID_Profesor"/></td>
                    <td><xsl:value-of select="Nombre"/></td>
                    <td><xsl:value-of select="Apellido"/></td>
                    <td><xsl:value-of select="Email"/></td>
                </tr>
            </xsl:for-each>
        </tbody>
    </table>
</body>
</html>
```

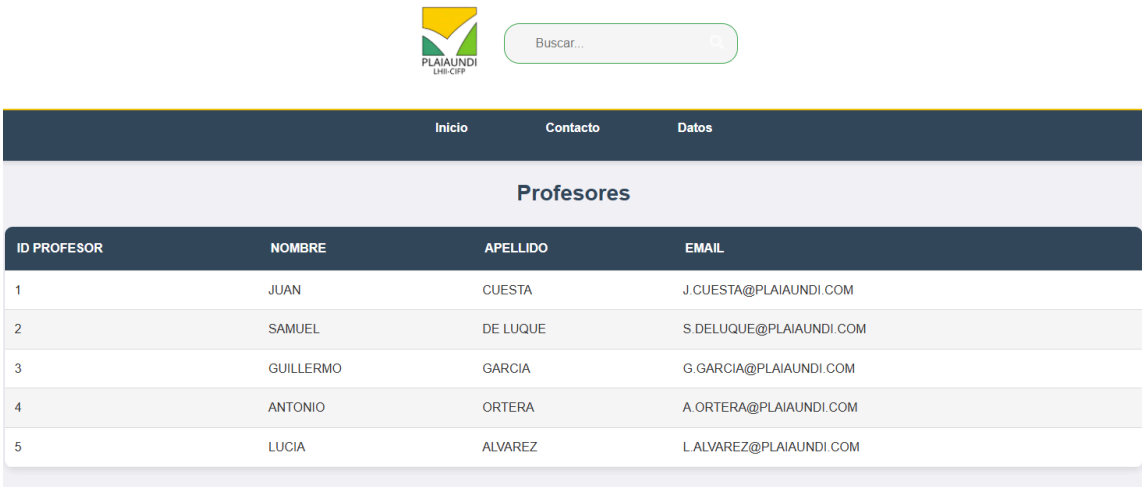
```
  </xsl:template>
</xsl:stylesheet>
```

Este sería el XSL de la tabla profesor también.

8.4. Generación del HTML final

Al final del proceso, los datos que estaban en MySQL ahora estarán en HTML y ya estarán listos para mostrarse en la página web. Para poder sacar el HTML de cada una de las tablas lo que haremos es copiar el XML y el XSL y los pegaremos en la página web FreeFormatter y nos generará el HTML fácilmente.

- Para terminar nos debería de quedar de la siguiente manera en la página web:



ID PROFESOR	NOMBRE	APELLIDO	EMAIL
1	JUAN	CUESTA	J.CUESTA@PLAIAUNDI.COM
2	SAMUEL	DE LUQUE	S.DELUQUE@PLAIAUNDI.COM
3	GUILLERMO	GARCIA	G.GARCIA@PLAIAUNDI.COM
4	ANTONIO	ORTERA	A.ORTERA@PLAIAUNDI.COM
5	LUCIA	ALVAREZ	L.ALVAREZ@PLAIAUNDI.COM

9. Configuración del NFS

9.1.Introducción a NFS (Network File System)

El **Network File System (NFS)** es un protocolo que permite el acceso remoto a sistemas de archivos a través de una red. Su principal objetivo es permitir que un cliente acceda a archivos en un servidor como si fueran locales, proporcionando una forma transparente de compartir directorios y archivos entre múltiples sistemas. NFS se utiliza principalmente en sistemas operativos

basados en Unix y Linux, aunque también hay implementaciones disponibles para otros sistemas como Windows.

9.1.1..Sistemas Operativos Comunes para NFS

NFS es ampliamente utilizado en **Linux** y **Unix**, especialmente en entornos de servidores y estaciones de trabajo que requieren compartir grandes volúmenes de datos de manera eficiente. Algunas implementaciones de NFS también están disponibles en **macOS** y **Windows**, pero su uso es menos frecuente. En entornos de servidores, NFS es una opción popular para compartir archivos debido a su robustez, flexibilidad y facilidad de configuración.

9.2.Diferencia entre Permisos de Red y Permisos del Sistema de Archivos

9.2.1.Permisos del Sistema de Archivos

Los **permisos del sistema de archivos** son los controles de acceso tradicionales en un sistema operativo. En sistemas Unix y Linux, estos permisos se gestionan a través de las propiedades de los archivos y directorios, como los **permisos de lectura (r)**, **escritura (w)** y **ejecución (x)**, asignados a los **propietarios**, **grupos** y **otros usuarios**. Estos permisos son gestionados por el propio sistema de archivos (ext4, xfs, etc.), y definen quién puede acceder a los archivos, modificarlos o ejecutarlos.

9.2.2.Permisos de Red

Por otro lado, los **permisos de red** se refieren a las restricciones que se aplican al acceso a los recursos compartidos a través de la red, como en el caso de NFS. Los permisos de red se gestionan mediante la configuración de exportaciones en el servidor NFS, que especifican qué clientes pueden acceder a qué directorios y con qué tipo de acceso (lectura, escritura o ambos). Estos permisos no se aplican de forma local sobre los archivos, sino que permiten o deniegan el acceso a las carpetas compartidas a través de la red.

9.3.Interacción entre los Permisos de Red y los Permisos del Sistema de Archivos

Cuando un cliente accede a un sistema de archivos compartido mediante NFS, **ambos conjuntos de permisos interactúan** para determinar el acceso final:

1. **Permisos del Sistema de Archivos**: El servidor NFS controla qué usuarios y grupos pueden acceder a los archivos, modificarlos o ejecutarlos, basándose en los permisos tradicionales del sistema de archivos.
2. **Permisos de Red (NFS)**: El servidor NFS también controla qué máquinas pueden montar el sistema de archivos y qué tipo de acceso tienen (lectura o escritura). Estos permisos son definidos en el archivo de configuración de NFS (/etc/exports en Linux).

Cuando un cliente intenta acceder a un directorio NFS:

- Si el **cliente tiene permisos de red** para acceder al recurso, se permite la conexión.
- Sin embargo, **los permisos del sistema de archivos** del servidor también deben permitir la operación solicitada.

A continuación se mostrará el uso y aplicación de un servidor NFS para este proyecto:

10. Grupos y Usuarios

10.1. Grupos

Se han creado dos grupos principales para organizar a los usuarios:

- **profesores**: Grupo para los profesores con permisos elevados.
- **alumnos**: Grupo para los alumnos con permisos restringidos.

Comandos para crear los grupos:

```
sudo groupadd -g 2000 profesores
sudo groupadd -g 2001 alumnos
```

10.2. Usuarios

Se han creado los siguientes usuarios:

- **profesor1 y profesor2**: Usuarios para los profesores.
- **alumno1_1ASIR3** y **alumno2_1ASIR3**: Usuarios para los alumnos de 1ASIR3.
- **alumno1_2ASIR3** y **alumno2_2ASIR3**: Usuarios para los alumnos de 2ASIR3.

Comandos para crear los usuarios:

```
sudo useradd -u 2000 -g profesores profesor1
sudo useradd -u 2001 -g profesores profesor2
sudo useradd -u 2002 -g alumnos alumno1_1ASIR3
sudo useradd -u 2003 -g alumnos alumno2_1ASIR3
sudo useradd -u 2004 -g alumnos alumno1_2ASIR3
sudo useradd -u 2005 -g alumnos alumno2_2ASIR3
```

Asignación de usuarios a grupos:

```
sudo usermod -aG profesores profesor1
sudo usermod -aG profesores profesor2
sudo usermod -aG alumnos alumno1_1ASIR3
sudo usermod -aG alumnos alumno2_1ASIR3
sudo usermod -aG alumnos alumno1_2ASIR3
sudo usermod -aG alumnos alumno2_2ASIR3
```

11. Configuración de Permisos en el Servidor NFS

11.1 Estructura de Carpetas

Se ha configurado la siguiente estructura de carpetas en el servidor NFS:

```
/mnt/nfs/
├── 1ASIR3/
│   ├── profesores/
│   ├── alumnos/
│       ├── apuntes/
│       ├── alumno1/
│       ├── alumno2/
├── 2ASIR3/
│   ├── profesores/
│   ├── alumnos/
│       ├── apuntes/
│       ├── alumno1/
│       ├── alumno2/
```

Comandos para crear las carpetas:

```
sudo mkdir -p /mnt/nfs/1ASIR3/profesores
sudo mkdir -p /mnt/nfs/1ASIR3/alumnos/apuntes
sudo mkdir -p /mnt/nfs/1ASIR3/alumnos/alumno1
sudo mkdir -p /mnt/nfs/1ASIR3/alumnos/alumno2

sudo mkdir -p /mnt/nfs/2ASIR3/profesores
sudo mkdir -p /mnt/nfs/2ASIR3/alumnos/apuntes
sudo mkdir -p /mnt/nfs/2ASIR3/alumnos/alumno1
sudo mkdir -p /mnt/nfs/2ASIR3/alumnos/alumno2
```

11.2 Asignación de Permisos

11.2.1.Carpeta profesores

- **Propietario**: profesor1
- **Grupo**: profesores
- **Permisos**: Lectura, escritura y ejecución para el propietario y el grupo.

Comandos:

```
sudo chown -R profesor1:profesores /mnt/nfs/1ASIR3/profesores
sudo chown -R profesor1:profesores /mnt/nfs/2ASIR3/profesores
sudo chmod 770 /mnt/nfs/1ASIR3/profesores
sudo chmod 770 /mnt/nfs/2ASIR3/profesores
```

11.2.2.Carpeta apuntes

- **Propietario**: profesor1
- **Grupo**: profesores
- **Permisos**: Lectura, escritura y ejecución para el propietario y el grupo; lectura y ejecución para alumnos mediante una ACL.

Comandos:

```
sudo chown -R profesor1:profesores /mnt/nfs/1ASIR3/alumnos/apuntes
sudo chown -R profesor1:profesores /mnt/nfs/2ASIR3/alumnos/apuntes
```

```
sudo chmod 770 /mnt/nfs/1ASIR3/alumnos/apuntes
sudo chmod 770 /mnt/nfs/2ASIR3/alumnos/apuntes
sudo setfacl -m g:alumnos:rx /mnt/nfs/1ASIR3/alumnos/apuntes
sudo setfacl -m g:alumnos:rx /mnt/nfs/2ASIR3/alumnos/apuntes
```

11.2.3.Carpetas personales de alumnos

- **Propietario**: Cada alumno
- **Grupo**: alumnos
- **Permisos**: Lectura, escritura y ejecución solo para el propietario.

Comandos:

```
sudo chown -R alumno1_1ASIR3:profesores /mnt/nfs/1ASIR3/alumnos/alumno1
sudo chown -R alumno2_1ASIR3:profesores /mnt/nfs/1ASIR3/alumnos/alumno2
sudo chown -R alumno1_2ASIR3:profesores /mnt/nfs/2ASIR3/alumnos/alumno1
sudo chown -R alumno2_2ASIR3:profesores /mnt/nfs/2ASIR3/alumnos/alumno2
sudo chmod 700 /mnt/nfs/1ASIR3/alumnos/alumno1
sudo chmod 700 /mnt/nfs/1ASIR3/alumnos/alumno2
sudo chmod 700 /mnt/nfs/2ASIR3/alumnos/alumno1
sudo chmod 700 /mnt/nfs/2ASIR3/alumnos/alumno2
```

12.Configuración del Servidor NFS

12.1. Instalación del Servidor NFS

Para compartir carpetas a través de la red, instalamos el paquete necesario para el servidor NFS en el servidor principal. Utilizamos los siguientes comandos:

```
sudo apt update
sudo apt install nfs-kernel-server
sudo apt install acl
```

12.1.1 Configuración del Archivo /etc/exports

Editamos el archivo /etc/exports para definir qué directorios se compartirán y con qué permisos. Usamos el siguiente comando:

```
sudo nano /etc/exports
```

Añadimos las siguientes líneas para compartir las carpetas de las clases 1ASIR3 y 2ASIR3:

```
/mnt/nfs/1ASIR3 10.10.16.2/24(rw,sync,no_subtree_check,all_squash)
/mnt/nfs/2ASIR3 10.10.16.2/24(rw,sync,no_subtree_check,all_squash)
```

- **/mnt/nfs/1ASIR3 y /mnt/nfs/2ASIR3**: Directorios compartidos.
- **10.10.16.2/24**: Acceso restringido a la subred local.
- **rw**: Permisos de lectura y escritura.
- **sync**: Sincroniza los cambios en el servidor para mayor seguridad.
- **no_subtree_check**: Mejora el rendimiento al desactivar verificaciones de subárbol.
- **all_squash**: Mapea todos los usuarios a uno anónimo.

12.1.2. Reiniciar el Servicio NFS

Para aplicar los cambios, reiniciamos el servicio con:

```
sudo systemctl restart nfs-kernel-server
```

12.1.3. Verificación de la Configuración

Verificamos los directorios exportados correctamente con:

```
sudo exportfs -v
```

El resultado esperado:

```
/mnt/nfs/1ASIR3     10.10.14.0/24(rw,sync,no_subtree_check,all_squash)
/mnt/nfs/2ASIR3      10.10.15.0/24(rw,sync,no_subtree_check,all_squash)
```

12.2 Montaje de los Directorios Compartidos en los Clientes

12.2.1 Instalación del Cliente NFS

En los equipos cliente, instalamos el paquete necesario:

```
sudo apt update
sudo apt install nfs-common
```

```
sudo apt install acl
```

12.2.2 Creación de Puntos de Montaje

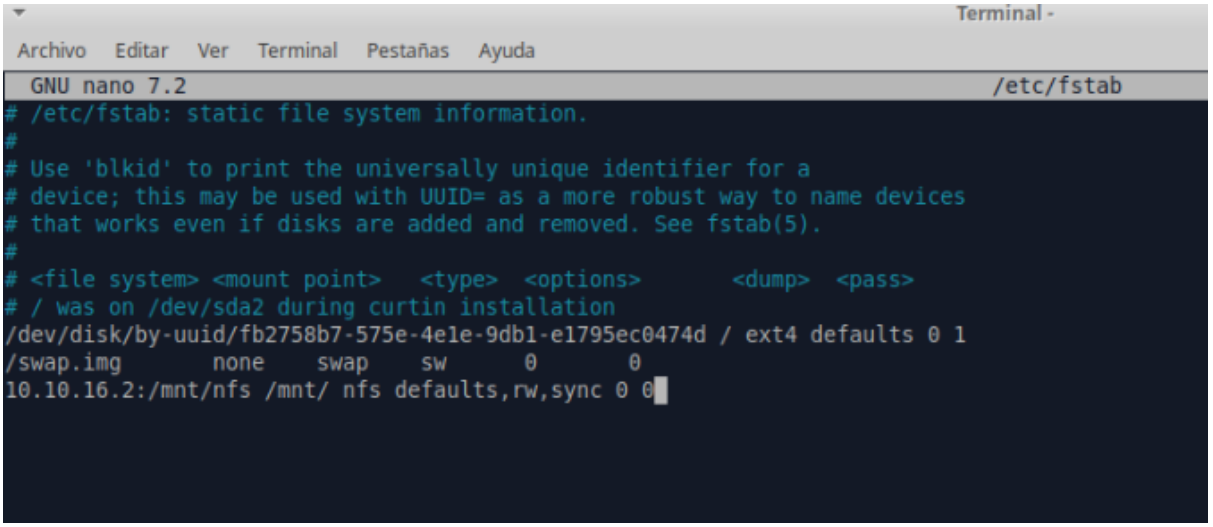
Creamos directorios locales para montar los recursos compartidos:

```
sudo mkdir -p /mnt/nfs_cliente/1ASIR3
sudo mkdir -p /mnt/nfs_cliente/2ASIR3
```

12.2.3 Montaje de los Directorios Compartidos

Montamos los directorios compartidos desde el servidor:

```
sudo mount 10.10.16.2:/mnt/nfs/1ASIR3 /mnt/nfs_cliente/1ASIR3
sudo mount 10.10.16.2:/mnt/nfs/2ASIR3 /mnt/nfs_cliente/2ASIR3
```



(Configuración en fstab para que se monte por defecto al iniciar el sistema)

12.2.4 Verificación del Montaje

Comprobamos que los recursos compartidos se montaron correctamente con:

```
df -h
```

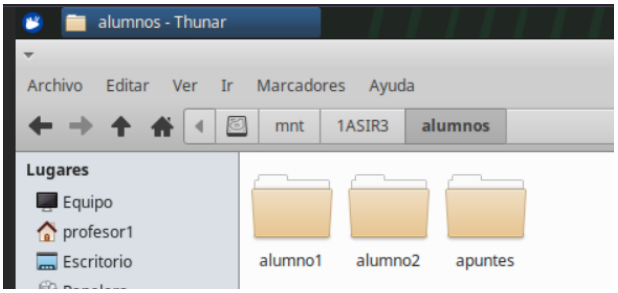
Resultado de ejemplo:

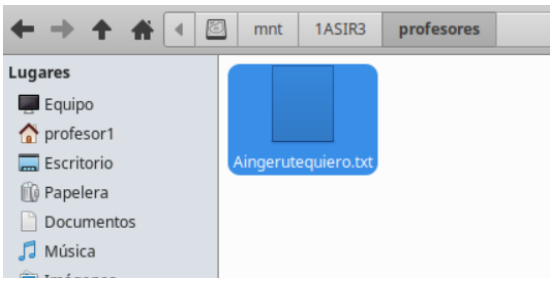
```
10.10.16.2:/mnt/nfs/1ASIR3  50G  10G  40G  20%  /mnt/nfs_cliente/1ASIR3
10.10.16.2:/mnt/nfs/2ASIR3  50G  10G  40G  20%  /mnt/nfs_cliente/2ASIR3
```

12.2.5 Pruebas de Acceso y Funcionalidad

Acceso como Profesor

Verificamos que los profesores tienen acceso completo creando un archivo.



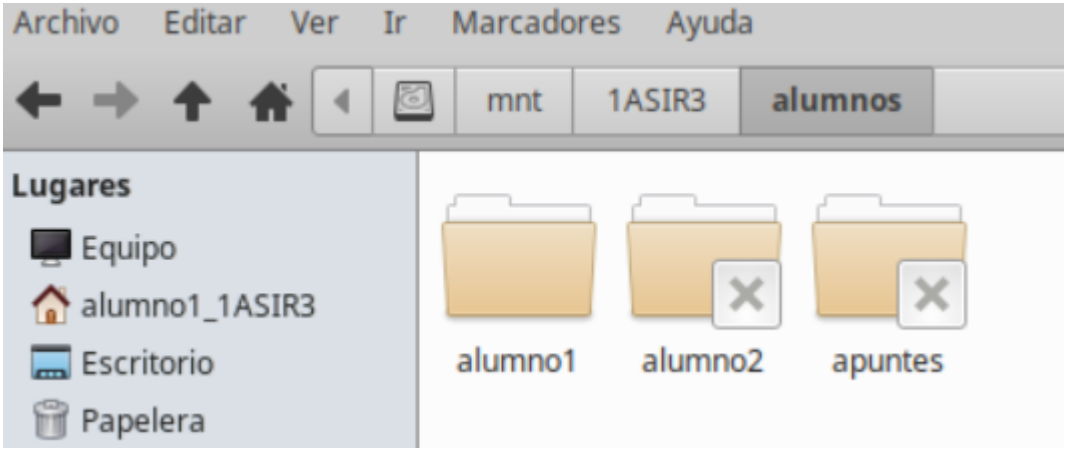


Acceso como Alumno

Probamos que los alumnos solo tienen permisos limitados en la carpeta de apuntes:

```
sudo touch /mnt/nfs_cliente/1ASIR3/alumnos/apuntes/archivo_alumno.txt
```

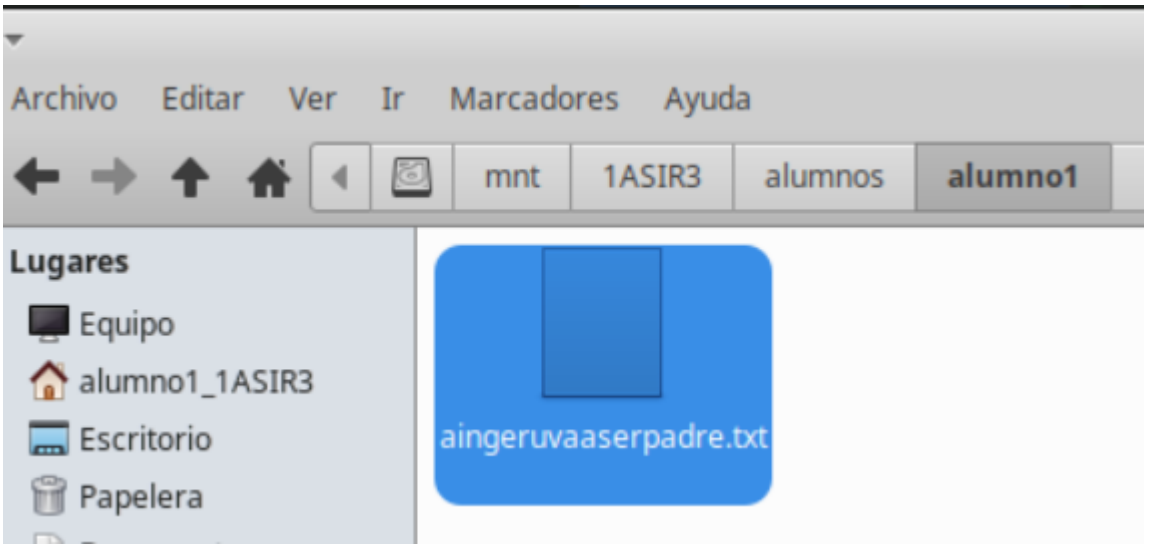
El sistema debe restringir la creación según los permisos configurados.



Acceso a Carpetas Personales

Cada alumno verifica que puede acceder y modificar su carpeta personal:

```
sudo touch /mnt/nfs_cliente/1ASIR3/alumnos/alumno1/aingeruvaaserpadre.txt
```



12.3 Consideraciones de Seguridad

12.3.1 Restricción por IP

Limitamos el acceso a la subred local 10.10.14.0/24 y 10.10.15.0/24 para evitar accesos externos.

12.3.2 Configuración del Firewall

Abrimos el puerto NFS (2049) en el servidor para los clientes y comprobamos que el firewall está activo:

```
sudo ufw allow 2049
sudo systemctl status ufw
```

12.4 Script de Creación de Grupos, Usuarios y Permisos

Para facilitar el proceso y agrupar todo lo realizado en este documento, haremos uso de un ejecutable. Este script automatiza la creación de grupos, usuarios, estructura de carpetas y permisos en el servidor NFS. También configura el archivo /etc/exports y reinicia el servicio NFS para aplicar los cambios. Paralelamente, en los PCs cliente, se ha creado otro script que crea todos los grupos y usuarios así como monta y deja montado de inicio las carpetas del servidor NFS.

1. Script para el servidor NFS:

```
#!/bin/bash

#Creacion de grupos con GID específicos
echo "Creando grupos..."
sudo groupadd -g 2000 profesores
sudo groupadd -g 2001 alumnos

# Creacion de usuarios con UID específicos y asignarlos a los grupos
echo "Creando usuarios..."
sudo useradd -u 2000 -g profesores profesor1
sudo useradd -u 2001 -g profesores profesor2
sudo useradd -u 2002 -g alumnos alumno1_1ASIR3
sudo useradd -u 2003 -g alumnos alumno2_1ASIR3
sudo useradd -u 2004 -g alumnos alumno1_2ASIR3
sudo useradd -u 2005 -g alumnos alumno2_2ASIR3
# Asignar contraseñas a los usuarios
echo "Asignando contraseñas..."
echo "profesor1:profesor1" | sudo chpasswd
echo "profesor2:profesor2" | sudo chpasswd
echo "alumno1_1ASIR3:alumno1" | sudo chpasswd
echo "alumno2_1ASIR3:alumno2" | sudo chpasswd
echo "alumno1_2ASIR3:alumno3" | sudo chpasswd
echo "alumno2_2ASIR3:alumno4" | sudo chpasswd


# Creacion estructura de carpetas
echo "Creando estructura de carpetas..."
sudo mkdir -p /mnt/nfs/1ASIR3/profesores
sudo mkdir -p /mnt/nfs/1ASIR3/alumnos/apuntes
sudo mkdir -p /mnt/nfs/1ASIR3/alumnos/alumno1
sudo mkdir -p /mnt/nfs/1ASIR3/alumnos/alumno2


sudo mkdir -p /mnt/nfs/2ASIR3/profesores
sudo mkdir -p /mnt/nfs/2ASIR3/alumnos/apuntes
sudo mkdir -p /mnt/nfs/2ASIR3/alumnos/alumno1
sudo mkdir -p /mnt/nfs/2ASIR3/alumnos/alumno2
```

```
# Asignar permisos a las carpetas
echo "Asignando permisos..."

# Carpeta profesores
sudo chown -R profesor1:profesores /mnt/nfs/1ASIR3/profesores
sudo chown -R profesor2:profesores /mnt/nfs/2ASIR3/profesores
sudo chmod 770 /mnt/nfs/1ASIR3/profesores
sudo chmod 770 /mnt/nfs/2ASIR3/profesores

# Carpeta apuntes
sudo chown -R profesor1:profesores /mnt/nfs/1ASIR3/alumnos/apuntes
sudo chown -R profesor2:profesores /mnt/nfs/2ASIR3/alumnos/apuntes
sudo chmod 770 /mnt/nfs/1ASIR3/alumnos/apuntes
sudo chmod 770 /mnt/nfs/2ASIR3/alumnos/apuntes
sudo setfacl -m g:alumnos:rx /mnt/nfs/1ASIR3/alumnos/apuntes
sudo setfacl -m g:alumnos:rx /mnt/nfs/2ASIR3/alumnos/apuntes

# Carpetas personales de alumnos
sudo chown -R alumno1_1ASIR3:profesores /mnt/nfs/1ASIR3/alumnos/alumno1
sudo chown -R alumno2_1ASIR3:profesores /mnt/nfs/1ASIR3/alumnos/alumno2
sudo chown -R alumno1_2ASIR3:profesores /mnt/nfs/2ASIR3/alumnos/alumno1
sudo chown -R alumno2_2ASIR3:profesores /mnt/nfs/2ASIR3/alumnos/alumno2
sudo chmod 770 /mnt/nfs/1ASIR3/alumnos/alumno1
sudo chmod 770 /mnt/nfs/1ASIR3/alumnos/alumno2
sudo chmod 770 /mnt/nfs/2ASIR3/alumnos/alumno1
sudo chmod 770 /mnt/nfs/2ASIR3/alumnos/alumno2

# Configurar NFS
echo "Configurando NFS..."
sudo bash -c 'echo "/mnt/nfs/1ASIR3
10.10.14.0/24(rw,sync,no_subtree_checkno_,no_all_squash)" >> /etc/exports'
sudo bash -c 'echo "/mnt/nfs/2ASIR3
10.10.15.0/24(rw,sync,no_subtree_check,no_all_squash)" >> /etc/exports'


# Reiniciar el servicio NFS
echo "Reiniciando el servicio NFS..."
sudo systemctl restart nfs-kernel-server
```

```
# Verificar la configuración
echo "Verificando la configuración..."
sudo exportfs -v

echo "¡Yeah buddy, configuración completada!"
```

2. Script para los clientes

```
#!/bin/bash

# Crear grupos
groupadd -g 2000 profesores
groupadd -g 2001 alumnos

# Crear los usuarios
useradd -u 2001 -g 2000 -m -s /bin/bash profesor1
useradd -u 2001 -g 2000 -m -s /bin/bash profesor2
useradd -u 2002 -g 2001 -m -s /bin/bash alumno1_1ASIR3
useradd -u 2003 -g 2001 -m -s /bin/bash alumno2_1ASIR3
useradd -u 2004 -g 2001 -m -s /bin/bash alumno1_2ASIR3
useradd -u 2005 -g 2001 -m -s /bin/bash alumno2_2ASIR3

# Establecer contraseñas para todos los usuarios
echo "contrasena123" | passwd --stdin profesor1
echo "contrasena123" | passwd --stdin profesor2
echo "contrasena123" | passwd --stdin alumno1_1ASIR3
echo "contrasena123" | passwd --stdin alumno2_1ASIR3
echo "contrasena123" | passwd --stdin alumno1_2ASIR3
echo "contrasena123" | passwd --stdin alumno2_2ASIR3

# Crear carpetas en /home
mkdir -p /home/profesor1
mkdir -p /home/profesor2
mkdir -p /home/alumno1_1ASIR3
mkdir -p /home/alumno2_1ASIR3
mkdir -p /home/alumno1_2ASIR3
mkdir -p /home/alumno2_2ASIR3
```

```
# Agregar entradas al fstab para montaje automático
echo "10.10.16.2:/mnt/nfs/ /mnt/profesor1 nfs defaults 0 0" >> /etc/fstab
echo "10.10.16.2:/mnt/nfs/ /mnt/profesor2 nfs defaults 0 0" >> /etc/fstab
echo "10.10.16.2:/mnt/nfs/ /mnt/alumno1_1ASIR3 nfs defaults 0 0" >> /etc/fstab
echo "10.10.16.2:/mnt/nfs/ /mnt/alumno2_1ASIR3 nfs defaults 0 0" >> /etc/fstab
echo "10.10.16.2:/mnt/nfs/ /mnt/alumno1_2ASIR3 nfs defaults 0 0" >> /etc/fstab
echo "10.10.16.2:/mnt/nfs/ /mnt/alumno2_2ASIR3 nfs defaults 0 0" >> /etc/fstab

# Montar todos los sistemas de archivos configurados en fstab
mount -a

echo "Yeah baby. Los usuarios y grupos han sido creados correctamente y el montaje automático ha sido configurado."
```

13. Conclusiones

En primer lugar, se pueden enumerar una serie de puntos, que a grandes rasgos, determinan qué y cómo se han implementado en el proyecto y lo que aportan a futuro.

1. Integración de tecnologías avanzadas: El proyecto ha logrado integrar diversas tecnologías y herramientas modernas, como Proxmox, servidores Ubuntu, VLANs, NFS, y bases de datos normalizadas. Esto no solo optimiza el rendimiento y la seguridad de la infraestructura, sino que también facilita la gestión y el acceso a la información por parte de los usuarios.

2. Mejora en la accesibilidad: Gracias al redireccionamiento de puertos y la implementación de un servidor Ubuntu accesible desde fuera de la red del proxmox, se ha logrado una mayor accesibilidad, lo que permite a estudiantes y profesores interactuar con los sistemas y bases de datos desde cualquier ubicación, garantizando flexibilidad en el trabajo y el aprendizaje.

3. Optimización de la gestión académica: La creación de una página web adaptativa para la gestión académica, que incluye información sobre ciclos, módulos, notas y más, permite a la comunidad educativa tener acceso a información actualizada de forma sencilla y directa. Esto no solo agiliza los procesos administrativos, sino que también mejora la transparencia y comunicación entre profesores y alumnos.

4. Impulso a la formación práctica: La base de datos para la bolsa de trabajo y la gestión de prácticas ofrece a los estudiantes la posibilidad de interactuar con el mundo laboral, lo que enriquece su experiencia educativa y facilita su inserción en el mercado laboral mediante prácticas profesionales o DUALES.

5. Diseño colaborativo y trabajo en equipo: El uso de herramientas como Trello para gestionar el proyecto ha fomentado la colaboración y coordinación efectiva entre los miembros del equipo, permitiendo avances constantes en todas las áreas del proyecto de manera sincronizada.

6. Escalabilidad y futuro: El diseño modular y flexible de la infraestructura permite que el sistema se pueda expandir fácilmente en el futuro. Las tecnologías empleadas garantizan que el sistema sea escalable y adaptable a posibles necesidades adicionales, como el incremento en el número de usuarios o la integración de nuevas herramientas.

7. Prototipo para jornadas de puertas abiertas: El proyecto ha cumplido con el objetivo de servir como prototipo para las jornadas de puertas abiertas del instituto, demostrando la capacidad de los estudiantes para implementar soluciones tecnológicas de alta calidad, y mostrando lo aprendido en el ciclo de ASIR.

Este proyecto no solo ha sido una oportunidad para aplicar lo aprendido, sino también para demostrar el poder de la tecnología como una herramienta de transformación. Cada línea de código, cada diseño adaptativo y cada componente de la infraestructura creada es testamento del esfuerzo, la dedicación y el espíritu colaborativo que han impulsado este proyecto desde su inicio.

Al ver cómo nuestras ideas han tomado forma, creando un entorno donde estudiantes y profesores pueden conectar, aprender y crecer, recordamos que la tecnología, cuando se utiliza con propósito, tiene la capacidad de cambiar el mundo. Este proyecto no solo moderniza una institución educativa; ha sembrado las semillas de un futuro más prometedor para todos los que formamos parte de este camino.

Es un reflejo de lo que somos capaces de lograr cuando unimos nuestras fuerzas, nuestros conocimientos y nuestra pasión por aprender. Cada uno de nosotros ha dejado una huella en este proyecto, y esa huella será el faro que guíe a los estudiantes del mañana hacia nuevas oportunidades, nuevos horizontes, y un mundo lleno de posibilidades.

Este es solo el principio. Lo que hemos construido aquí no es solo una infraestructura digital, es un puente hacia el futuro, donde la innovación y el aprendizaje no tienen límites. ¡Que este sea solo el primero de muchos pasos hacia un cambio significativo y duradero en nuestra comunidad educativa y más allá!