SÍNTESIS

Marc, Bryan y David

STUCOM

SIMIX 2, N

ÍNDICE

ÍNDICE 2
Versión 0 4
Redes ..... ¡Error! Marcador no definido.
Sistemas 4
DHCP 4
Robótica ..... ¡Error! Marcador no definido.
Versión 1 8
Redes 8
Sistemas 8
Instalación del servicio DNS 8
Robótica ..... ¡Error! Marcador no definido.
Versión 2 14
Redes 14
Implementación de VLANS 14
Sistemas 15
Windows active directory 15
Robótica 20
Buzzer + Temperatura 20
Versión 3 22
BBDD 22
Creación de base de datos con dos tablas 22
Sistemas 23
FTP 23
Robótica 30
Código de Acceso 30
Versión 4 33
BBDD 33
Creación de tabla WEB y Consultas a la base de datos 33
Sistemas 37
Pagina php 37
Robótica 43
Código de Acceso + INFORMACIÓN USUARIOS 43
Versión 5 46
BBDD 46

Creación de tabla WEB y Consultas a la base de datos 46
Sistemas 49
Active directory 49
Robótica 59
Código de Acceso con INFO y LEDS 59
Versión 6 61
Redes 61
Sistemas 64
SISTEMAS OPERATIVOS: MAIL 64
Robótica 67
Código de Acceso con cerradura 67
Versión 7 70
Redes 70
Sistemas 73
Squid 73
Comprobación 75
Robótica 76
Información LCD: 76
CÓDIGO 77
Video: 78
WOKWI: 78
Versión 8 79
Redes 79
Sistemas 81
Robótica 88
PRESENTACION 91

Versión 0

Sistemas

DHCP

Para poder empezar a convertir Ubuntu Server en un servidor necesitaremos instalar el DHCP, para eso, el primer paso necesario será upgradear y updatear Ubuntu Server.

- Sudo apt upgrade.

```
marc@marc:~$ sudo apt upgrade_
```

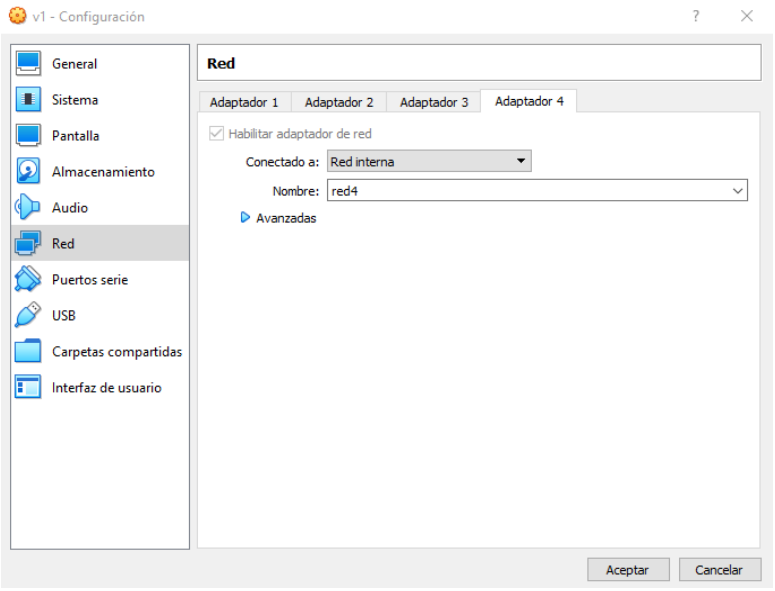
- Sudo apt update.

```
marc@marc:~$ sudo apt update_
```

Luego de ya tener Ubuntu Server preparado vamos a iniciar la instalación del DHCP con el siguiente comando.

```
marc@marc:~$ sudo apt-get install isc-dhcp-server
```

Una vez el servidor ya esté instalado agregaremos 3 tarjetas de redes internas, una vez creadas las redes internas se requerirá apagar y encender la máquina para que pueda detectar tarjetas de red.



Una vez ya estén instaladas las 3 tarjetas de red vamos a necesitar configurar las IPs de cada una de ellas, pero para empezar vamos a necesitar saber los nombres de cada red, para eso podríamos usar el siguiente comando que es muy útil en este tipo de situaciones, y lo que hace es mostrarnos los nombres de las redes disponibles.

```
marc@marc:/etc/bind$ ls /sys/class/net
enp0s10  enp0s3  enp0s8  enp0s9  lo
marc@marc:/etc/bind$ _
```

Una vez sepamos los nombres de las redes solo tocara configurarlas. Al inicio solo saldrá como predeterminada la red **'enp0s3'** y deberemos agregar las demás manualmente, pero es prácticamente copiar y pegar cambiando las redes y los nombres, por lo que será rápido y sencillo.

```
marc@marc:~$ sudo nano /etc/netplan/00-installer-config.yaml _
```

v1 [Corriendo] - Oracle VM VirtualBox

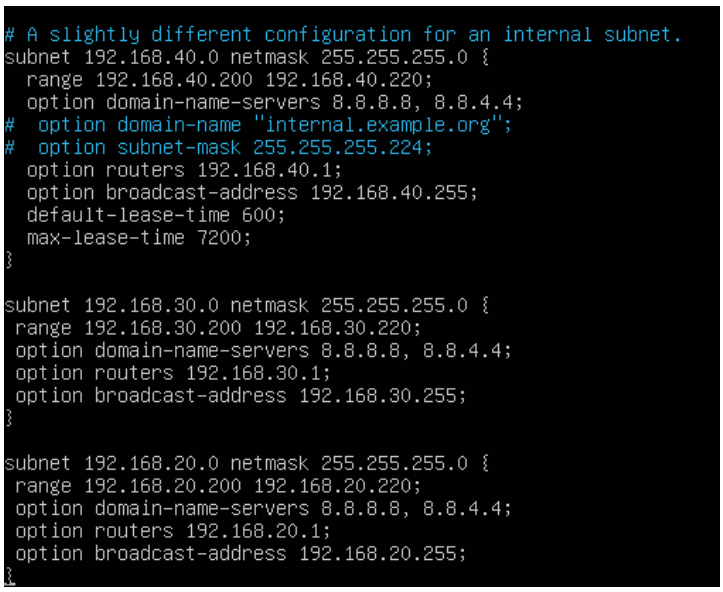
Archivo Máquina Ver Entrada Dispositivos Ayuda

```
  GNU nano 6.2                    /etc/netplan/00-installer-config.yaml
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      #dhcp4: true
      addresses: [192.168.10.2/24]
      gateway4: 192.168.10.1
    enp0s8:
      #dhcp4:
      addresses: [192.168.40.10/24]
    enp0s9:
      #dhcp4: true
      addresses: [192.168.30.10/24]
    enp0s10:
      #dhcp4: true
      addresses: [192.168.20.10/24]

  version: 2
```

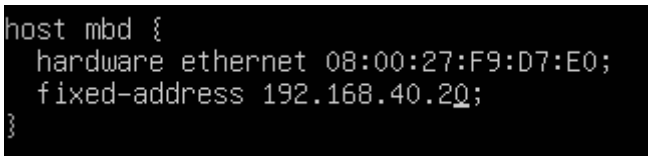
Ahora que ya deberíamos tener las 3 redes creadas y con sus IPs tocaría configurar el DHCP para las 3 redes con cada una de sus redes respectivamente.

```
marc@marc:~$ sudo nano /etc/dhcp/dhcpd.conf_
```

Empezamos con quitar los '#' para que se pudiera ejecutar la acción y cambiando los ejemplos por las IPs y redes de cada uno, al inicio solo deberíamos cambiar los ejemplos, pero las otras dos redes tocarían copiar y crear el código y comandos manualmente con cada red respectivamente para poder tener las 3, por lo que sería un trabajo más tedioso al saber que si te equivocas en una sola letra o espacio podría dejar de funcionar.



Luego deberíamos bajar más para poder configurar el host del cliente, donde pondremos su Mac y la IP reservada para poder tener la conexión con el cliente Windows 10.



Para finalizar con el servidor DHCP de Ubuntu Server tocaría guardar el progreso y reiniciar el servidor, una vez hecho eso tocaría encenderlo y comprobar que funciona con los siguientes comandos paso por paso.

```
marc@marc:~$ sudo /etc/init.d/isc-dhcp-server restart_
marc@marc:~$ sudo /etc/init.d/isc-dhcp-server start
```

```
marc@marc:~$ sudo /etc/init.d/isc-dhcp-server restart
[sudo] password for marc:
Restarting isc-dhcp-server (via systemctl): isc-dhcp-server.service.
marc@marc:~$ sudo /etc/init.d/isc-dhcp-server start
Starting isc-dhcp-server (via systemctl): isc-dhcp-server.service.
marc@marc:~$ sudo /etc/init.d/isc-dhcp-server status
● isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled
     Active: active (running) since Sun 2022-10-02 20:23:48 UTC; 17s ago
       Docs: man:dhcpd(8)
   Main PID: 997 (dhcpd)
      Tasks: 4 (limit: 1030)
     Memory: 4.8M
        CPU: 28ms
     CGroup: /system.slice/isc-dhcp-server.service
             └─997 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/d

oct 02 20:23:48 marc sh[997]: Listening on LPF/enp0s8/08:00:27:64:d0:f3/192.168.30.0/24
oct 02 20:23:48 marc dhcpd[997]: Sending on   LPF/enp0s8/08:00:27:64:d0:f3/192.168.30.0/24
oct 02 20:23:48 marc sh[997]: Sending on   LPF/enp0s8/08:00:27:64:d0:f3/192.168.30.0/24
oct 02 20:23:48 marc dhcpd[997]: Listening on LPF/enp0s3/08:00:27:df:dd:b0/192.168.40.0/24
oct 02 20:23:48 marc sh[997]: Listening on LPF/enp0s3/08:00:27:df:dd:b0/192.168.40.0/24
oct 02 20:23:48 marc dhcpd[997]: Sending on   LPF/enp0s3/08:00:27:df:dd:b0/192.168.40.0/24
oct 02 20:23:48 marc sh[997]: Sending on   LPF/enp0s3/08:00:27:df:dd:b0/192.168.40.0/24
oct 02 20:23:48 marc dhcpd[997]: Sending on   Socket/fallback/fallback-net
oct 02 20:23:48 marc sh[997]: Sending on   Socket/fallback/fallback-net
oct 02 20:23:48 marc dhcpd[997]: Server starting service.
```

Windows 10

Para finalizar encenderemos el cliente Windows 10 con la red interna ya puesta y solo quedaría comprobar con “ipconfig /all” si la IP que ha recibido es la misma que se ha programado anteriormente en el DHCP de Ubuntu Server.

```
   Sufijo DNS específico para la conexión. . : example.org
   Descripción . . . . . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Dirección física. . . . . . . . . . . . . : 08-00-27-F9-D7-E0
   DHCP habilitado . . . . . . . . . . . . . : sí
   Configuración automática habilitada . . . : sí
   Vínculo: dirección IPv6 local. . . : fe80::c945:8591:96bc:43fb%14(Preferido)
   Dirección IPv4. . . . . . . . . . . . . . : 192.168.40.20(Preferido)
   Máscara de subred . . . . . . . . . . . . : 255.255.255.0
   Concesión obtenida. . . . . . . . . . . . : miércoles, 19 de octubre de 2022 15:58:26
   La concesión expira . . . . . . . . . . . : miércoles, 19 de octubre de 2022 16:10:56
   Puerta de enlace predeterminada . . . . . : 192.168.40.1
   Servidor DHCP . . . . . . . . . . . . . . : 192.168.40.10
   IAID DHCPv6 . . . . . . . . . . . . . . . : 101187623
   DUID de cliente DHCPv6. . . . . . . . . . : 00-01-00-01-2A-E0-5F-57-08-00-27-F9-D7-E0
   Servidores DNS. . . . . . . . . . . . . . : 192.168.40.10
   NetBIOS sobre TCP/IP. . . . . . . . . . . : habilitado
```

Versión 1

Redes

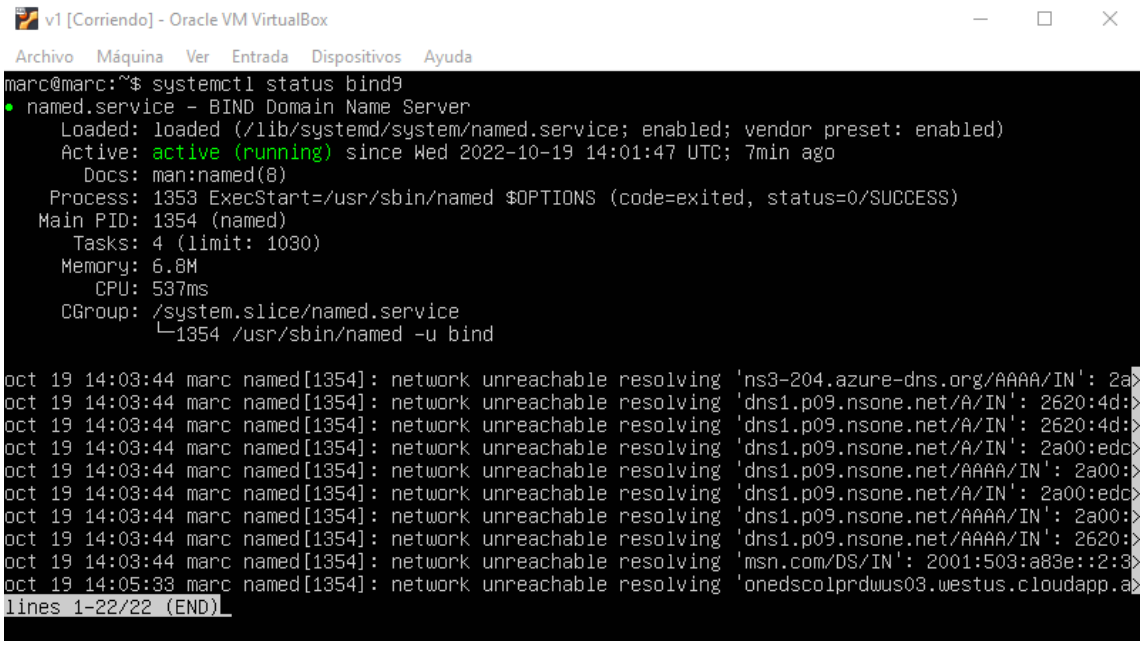
Sistemas

Instalación del servicio DNS

Para iniciar con esta parte de la práctica vamos a empezar instalando el servidor DNS, para eso solo deberemos ejecutar un comando, el cual es **"sudo apt install -y bind9"**.

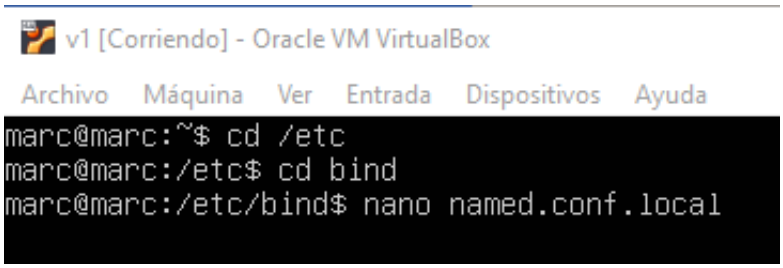
```
marc@marc:~$ sudo apt install -y bind9
```

Una vez el servidor está instalado comprobaremos si funciona y se ha instalado correctamente, para eso deberemos ejecutar el comando **"systemctl status bind9"**.



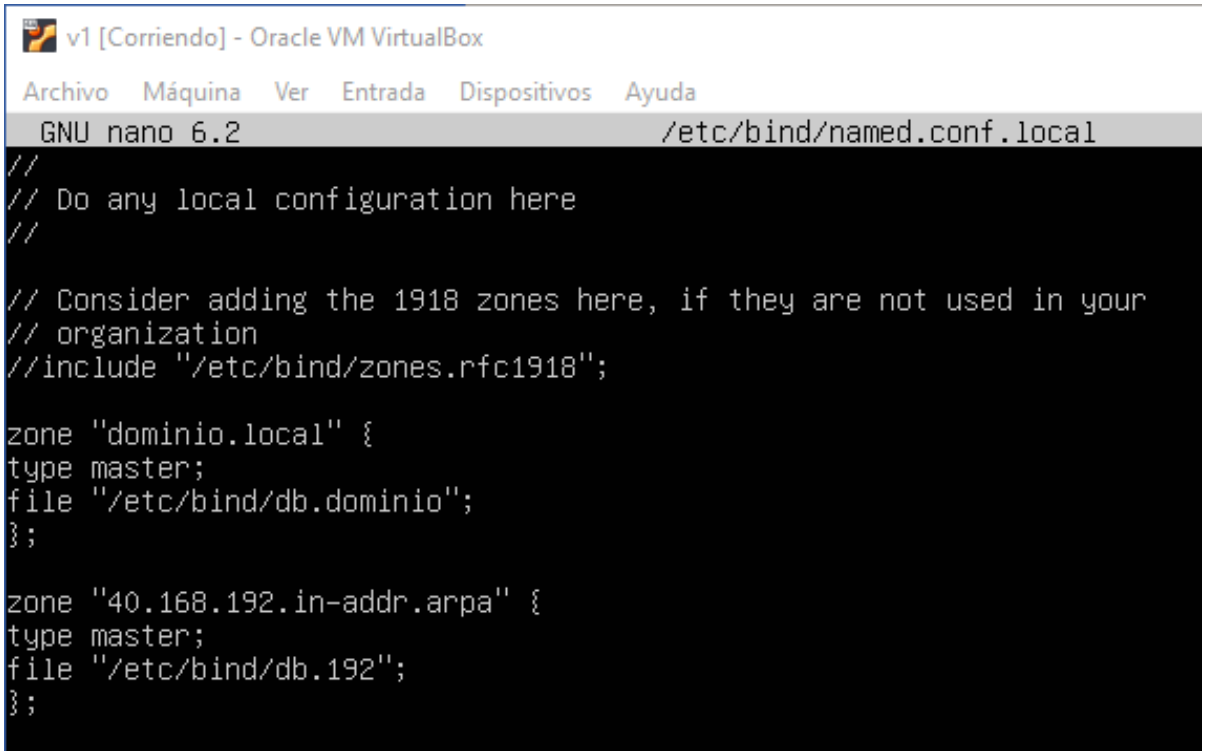
Para empezar con la creación de la zona vamos a tener que encontrar el archivo el cual configurar para añadir la zona, una vez sepamos qué fichero deberemos cambiar vamos a tener que ir hacia ese directorio y con el nano vamos a cambiar y configurar su

interior. Para ir hacia el directorio utilizaremos el comando **"cd /etc/bind"**, se puede hacer tanto individualmente (**"marc@marc:~$ cd /etc"** y luego **"marc@marc:/etc$ cd bind"**) o con el comando que he dicho anteriormente. Y a continuación haremos uso del nano para poder entrar en el fichero con **"nano named.conf.local"**.



Una vez entremos al nano de **"named.conf.local"**, vamos a tener que ir debajo del todo del fichero y crear la zona de **"dominio.local"** que va a contener registros tipo A.
Una vez creada la zona **"dominio.local"** vamos a crear otra zona, **"40.168.192.in-addr.arpa"**, que va a contener registros tipo **PTR**.

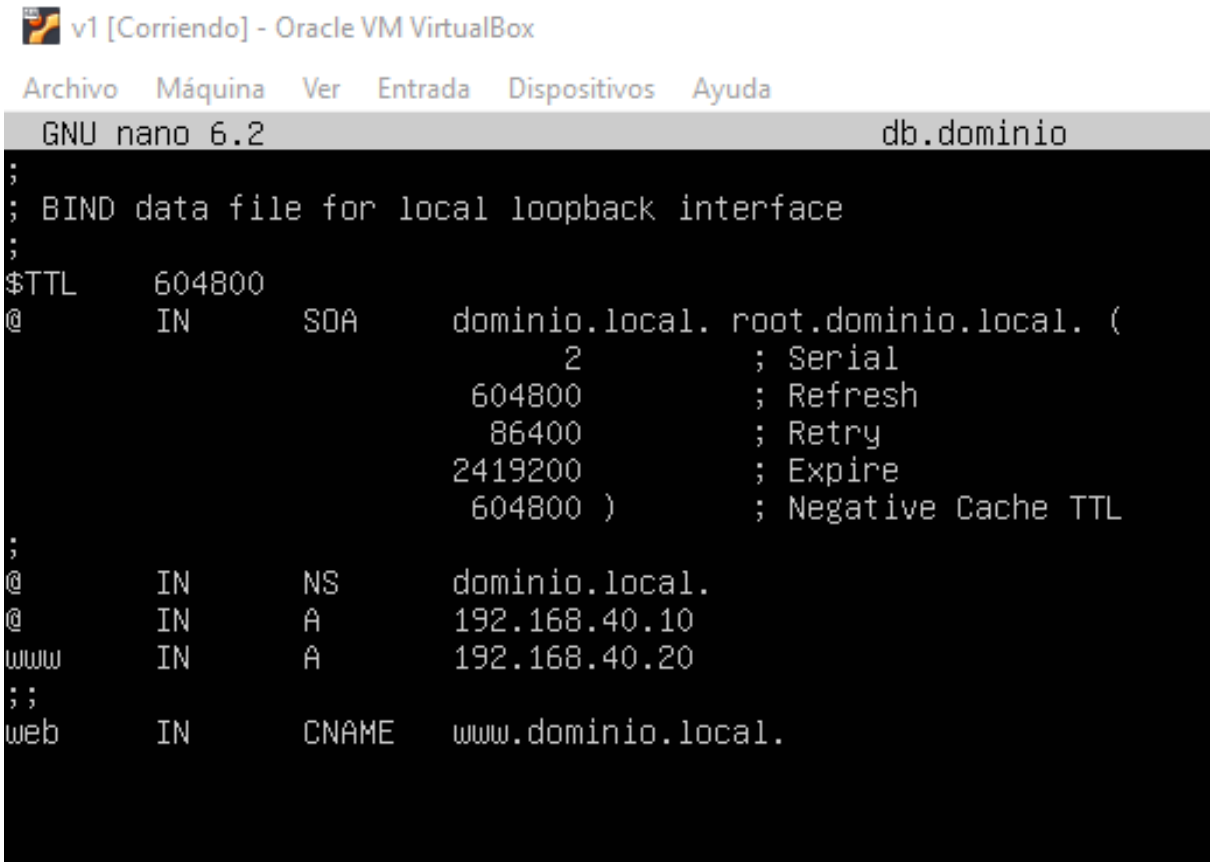
Cada una de las zonas vamos a decretar si son primarias o secundarias con el **"type master; o type slave;"**, luego de decretar si son primarias o secundarias vamos a marcar dónde están los registros definidos de cada zona, en el caso de la zona **"dominio.local"** los registros definidos van a estar en el fichero **"db.dominio"** donde los registros van a ser tipo A. En el caso de la zona **"40.168.192.in-addr.arpa"** los registros definidos van a estar en el fichero **"db.192"** el cual va a contener registros tipo **PTR**.



Luego de guardar el fichero vamos a crear los ficheros que van a contener los registros de las zonas, ya que no queremos crearlos de cero, vamos a copiar y a modificar posteriormente unos archivos que ya están en el sistema, los cuales son **"db.127"** y **"db.local"** respectivamente.

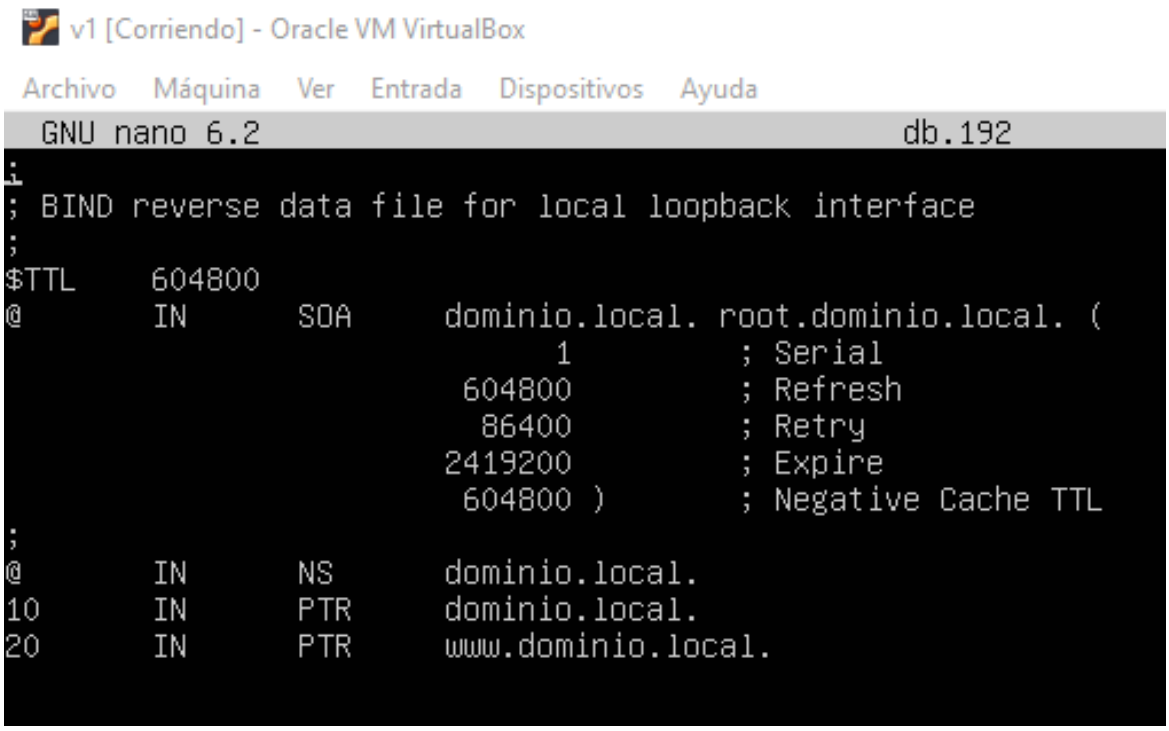
```
marc@marc:/etc/bind$ sudo cp db.127 db.192_
marc@marc:/etc/bind$ sudo cp db.local db.dominio_
```

Luego de copiar los ficheros vamos a tener que configurarlos, para entrar dentro el fichero usaremos el comando nano como anteriormente hemos hecho. En el caso **"db.dominio"** que es el que se mostrará a continuación, vamos a decretar que su zona es **"dominio.local"**. Para poder decretar cuál es su zona y que tipo de registros deben de captar, vamos a cambiar los nombres de la configuración inicial por el nombre **"dominio.local"** en **"db.dominio"**.
Luego de haber asignado su zona, vamos a crear el registro tipo A y su dirección IP asignada, para eso vamos a ir a bajo del todo del fichero y vamos a crear/modificar las 3 últimas líneas que se pueden ver en el documento.



Luego de haber configurado el fichero "db.muntane" vamos a proseguir a configurar el siguiente fichero "db.192". En el caso **"db.192"** que es el que se mostrará a continuación, vamos a decretar que su dominio es **"dominio.local"** ya que en el fichero

muntane.local figura los registros PTR para el anterior tipo A creado en db.muntane (pc1.muntane.local). De modo que al hacer un **"nslookup"** con el cliente declare que su IP es **"192.168.40.200"**.

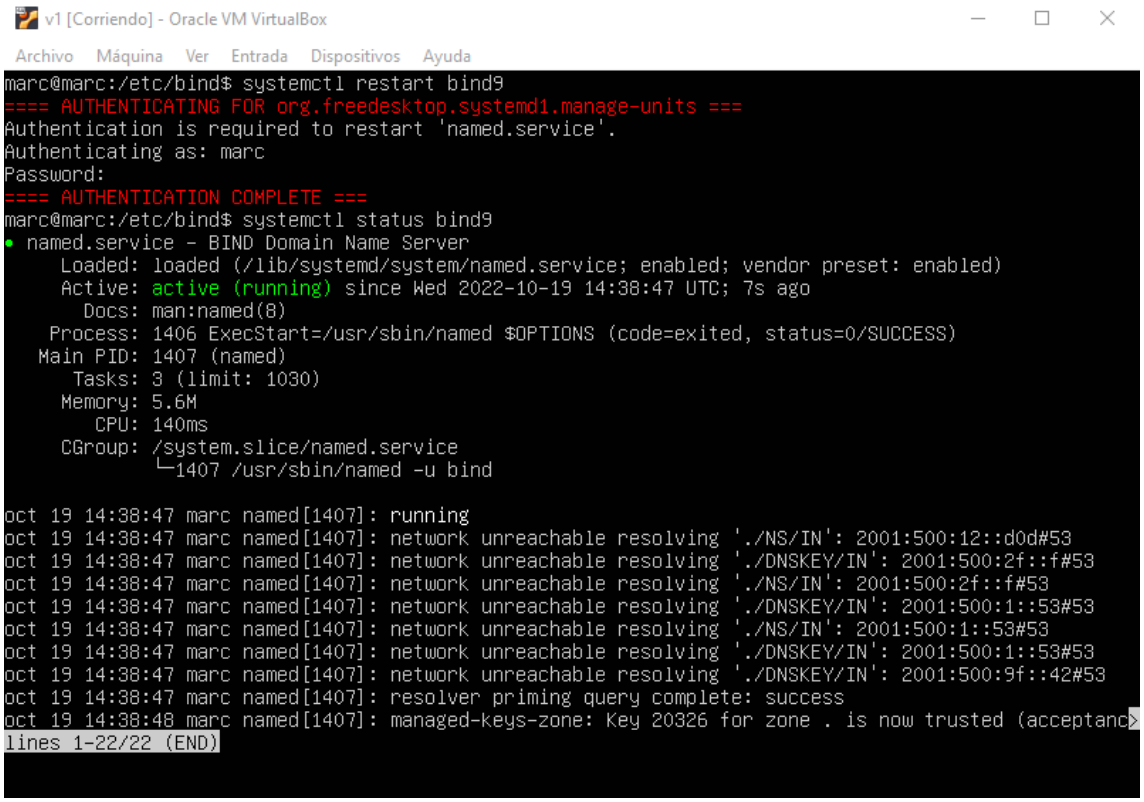


Para finalizar tocaría configurar el forward, el forward nos permite redireccionar mensajes a otra direccion, en este caso seria a google("**8.8.8.8**")

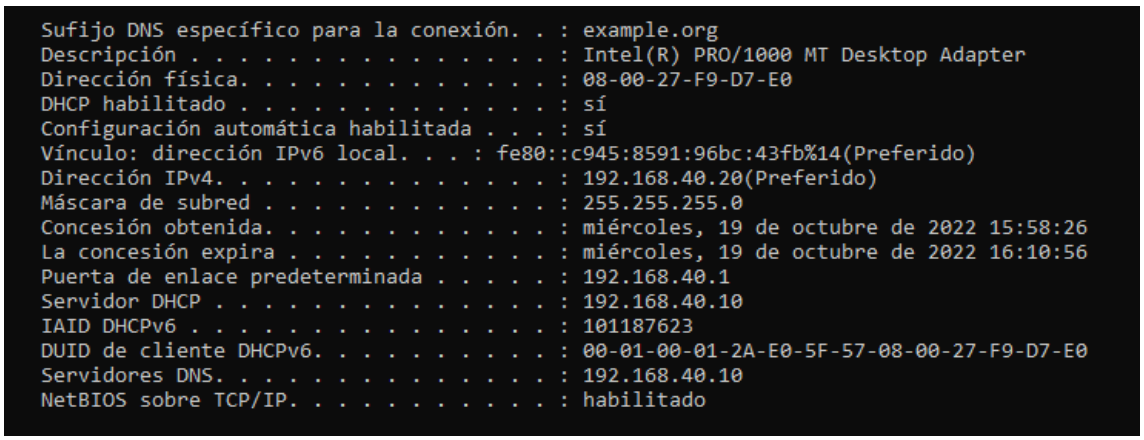
Una vez hemos configurado el archivo **"db.192"** y tenemos todo guardado y hecho vamos a reiniciar el DNS, luego de reiniciarlo lo vamos a iniciar y para finalizar vamos a comprobar si funciona, todos estos comandos se pueden ejecutar con una misma línea

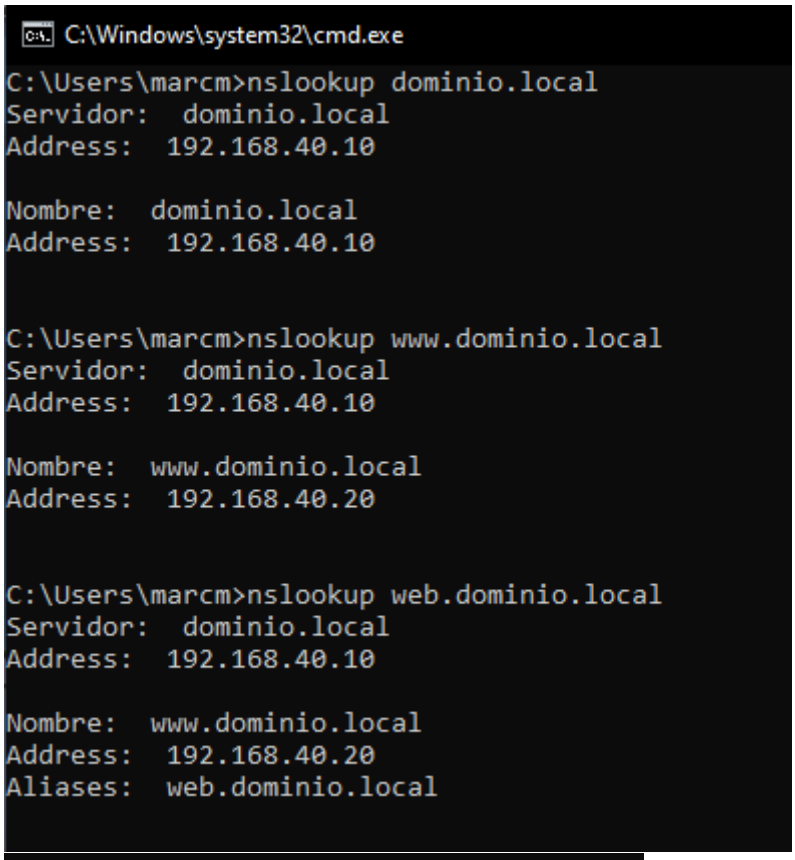
de comando **“sudo service bind9 …”**, donde los 3 puntos van a ser las 3 palabras siguientes, restart, start, o status dependiendo de lo que queramos hacer

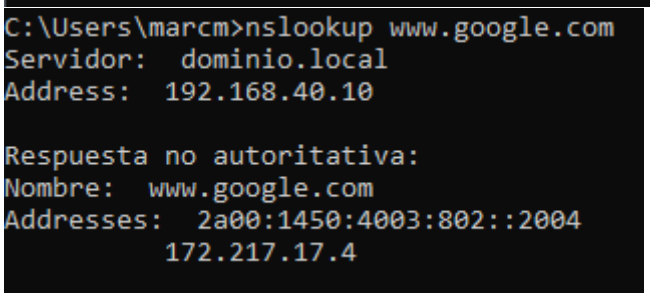
.



Para finalizar, vamos a ver si el DHCP de la actividad anterior sigue funcionando aplicando la IP automáticamente teniendo la red en red interna, y por último, vamos a comprobar que el servidor DNS funciona correctamente.




C:\Windows\system32\cmd.exe
C:\Users\marcm>nslookup dominio.local
Servidor: dominio.local
Address: 192.168.40.10
Nombre: dominio.local
Address: 192.168.40.10
C:\Users\marcm>nslookup www.dominio.local
Servidor: dominio.local
Address: 192.168.40.10
Nombre: www.dominio.local
Address: 192.168.40.20
C:\Users\marcm>nslookup web.dominio.local
Servidor: dominio.local
Address: 192.168.40.10
Nombre: www.dominio.local
Address: 192.168.40.20
Aliases: web.dominio.local

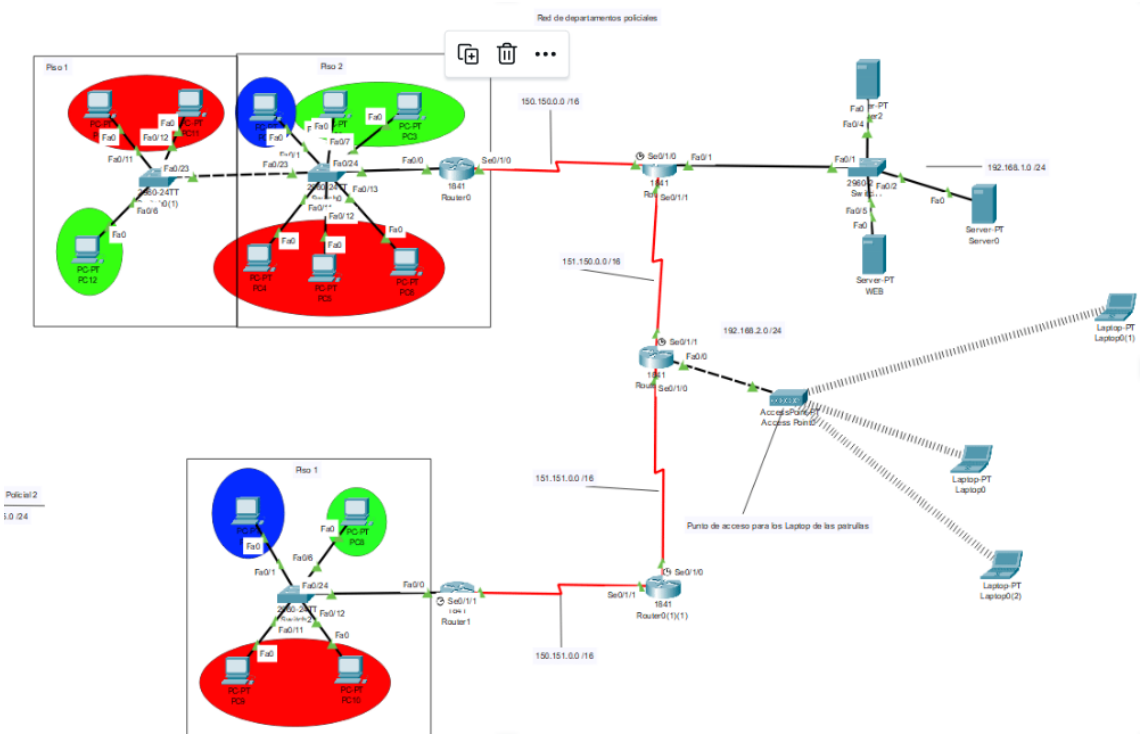

C:\Users\marcm>nslookup www.google.com
Servidor: dominio.local
Address: 192.168.40.10
Respuesta no autoritativa:
Nombre: www.google.com
Addresses: 2a00:1450:4003:802::2004
172.217.17.4

Versión 2

Redes

Implementación de VLANS

Implementar las VLAN's y los servicios de red en packet tracer. Diseñar una pequeña aproximación de la web privada y la pública, si la hubiese. Se deberá presentar el packet tracert ETIQUETADO.



Packet tracer

https://drive.google.com/file/d/1DapP-UqAkY_JAOdQ8TLXabPbDFF7wxUY/view?usp=sharing

Video

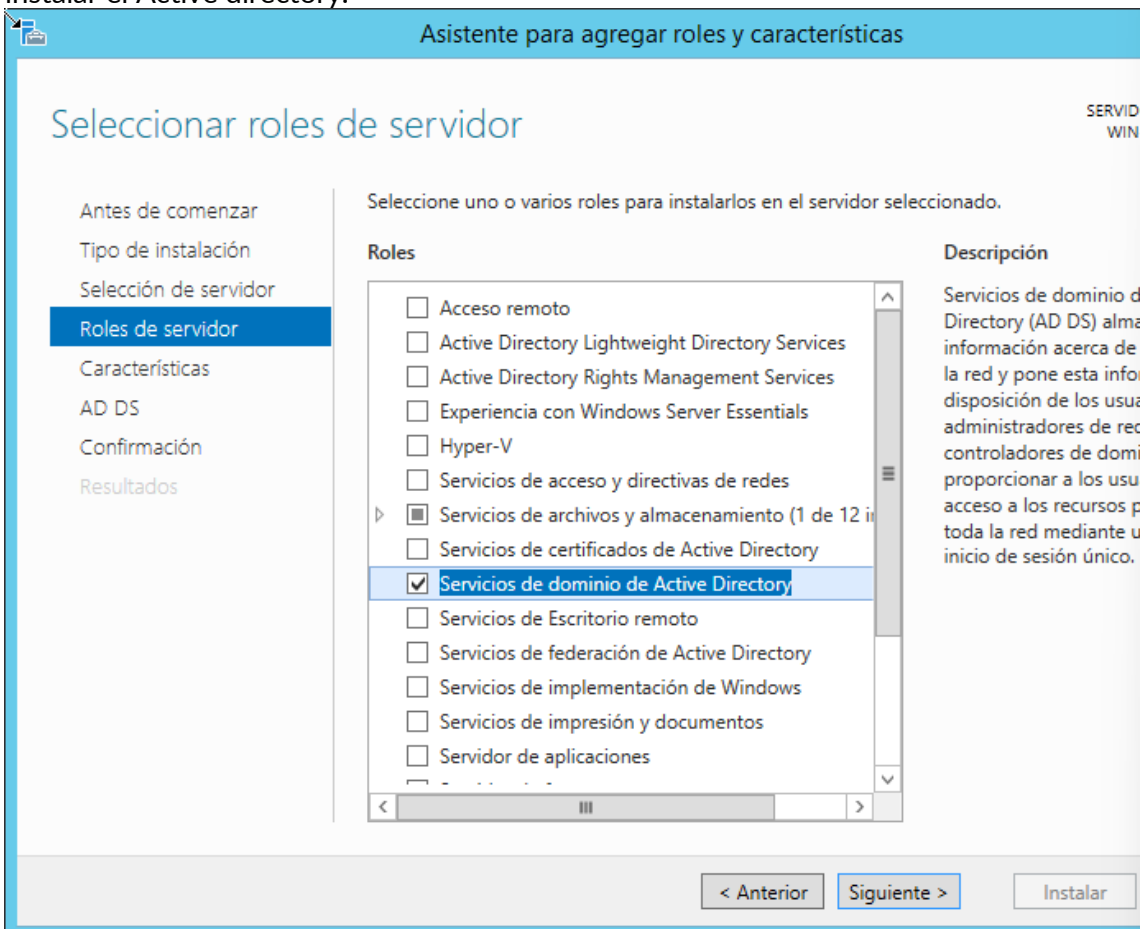
https://drive.google.com/file/d/1ksLaspbjTOv1zafcTMILGKcTtPatq36K/view?usp=sharing

(No funciona)

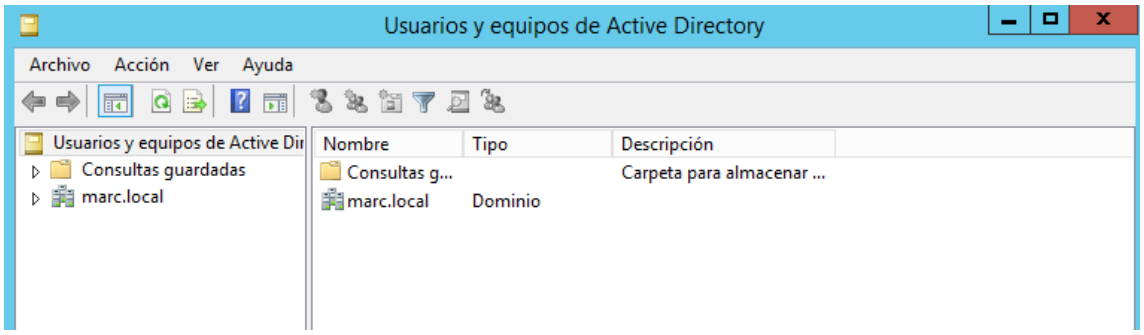
Sistemas

Windows active directory

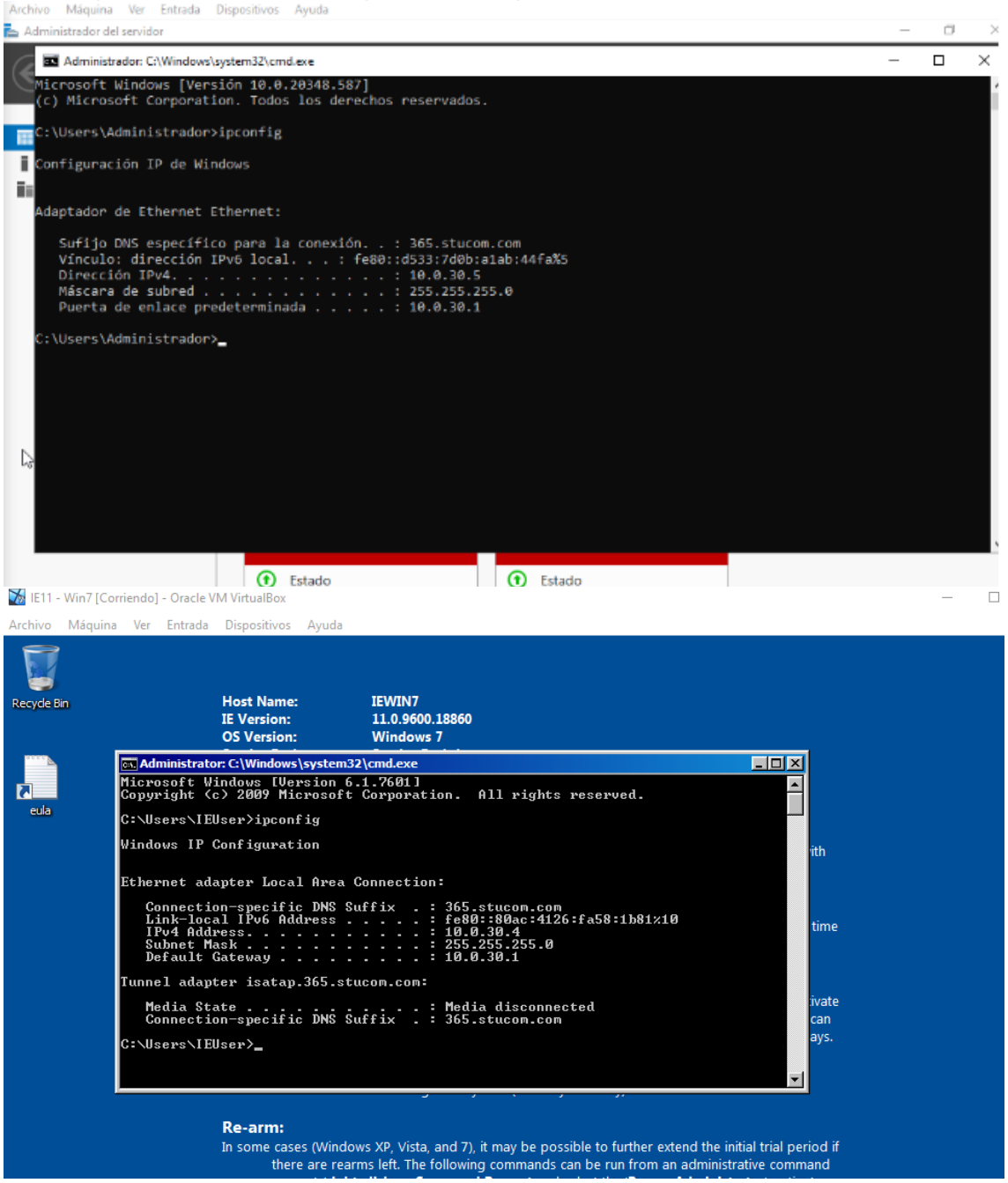
Para empezar, deberemos instalar un Windows Server el cual vamos a utilizar para instalar el Active directory.



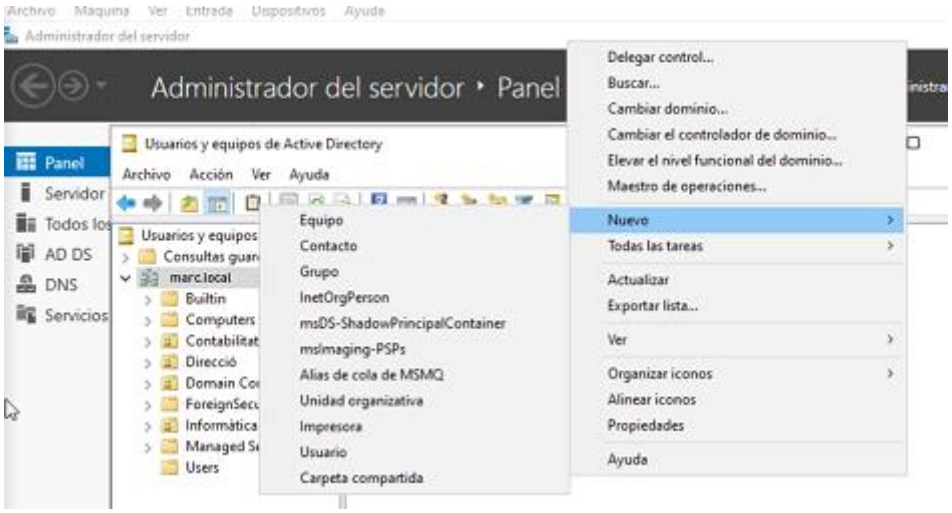
Crearemos una zona para poder empezar a utilizar el Active Directory que acabamos de instalar.

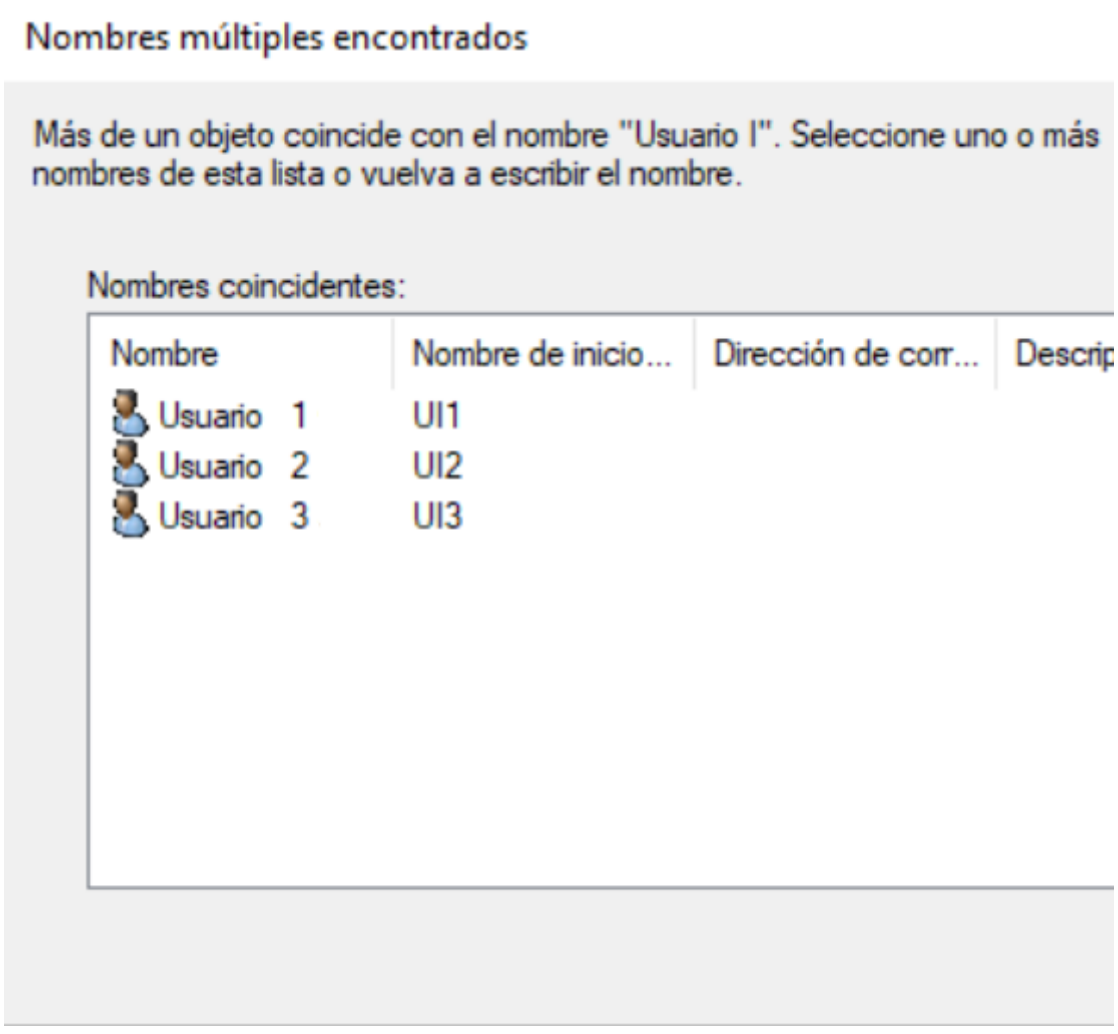


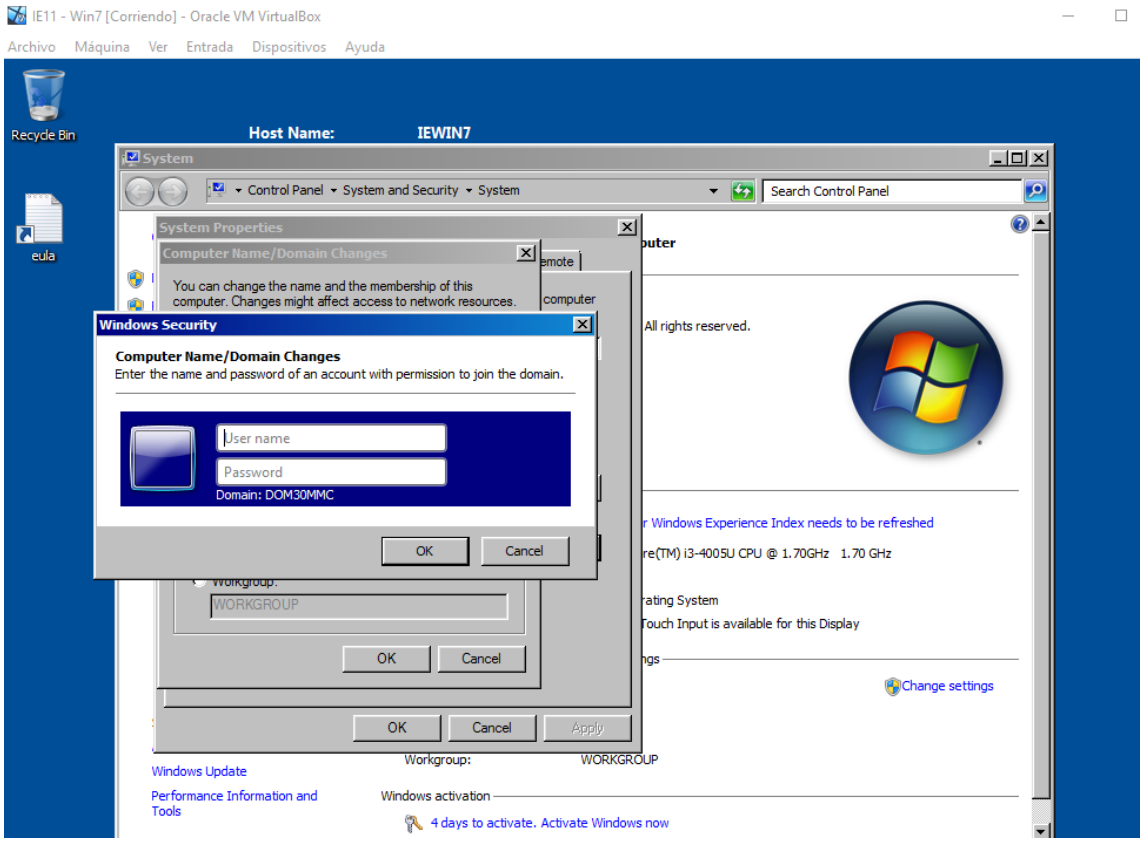
Luego de tener la zona creada y el inicio del Active Directory configurado, añadiremos una máquina cliente la cual comprobaremos que esté en la misma red.



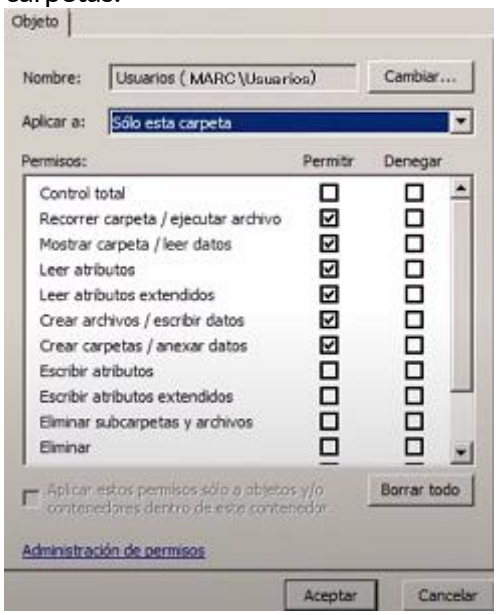
Luego de comprobar que se ven, crearemos usuarios en la zona los cuales luego vamos a conectar en la máquina cliente para que la máquina Windows Server sea su dominio.



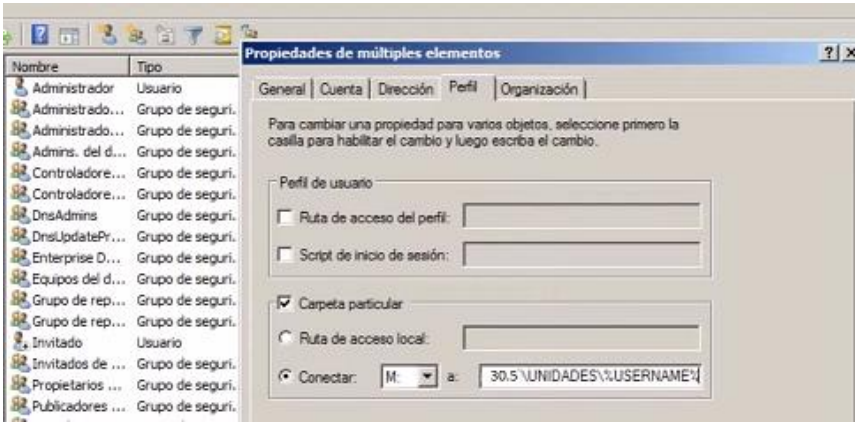




Para finalizar deberemos configurar las carpetas de los usuarios para que estén conectados al servidor, para poder ver todas las carpetas del usuario en el Windows Server, para eso, entramos a la configuración de los usuarios y configuraremos las carpetas.

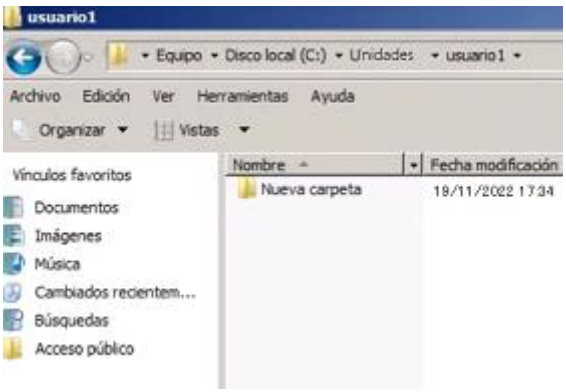


Luego, deberemos ir al perfil del usuario y configurarlo para que la ruta sea la de la IP del Servidor Windows.



Para comprobar que funciona, iniciaremos sesión en uno de los usuarios y crearemos una carpeta en la carpeta que está conectada al Servidor

Y por último, en el servidor miraremos si el usuario ha creado una carpeta, nos deberá salir el nombre de la carpeta y la fecha y la hora a la que fue creada.



Robótica

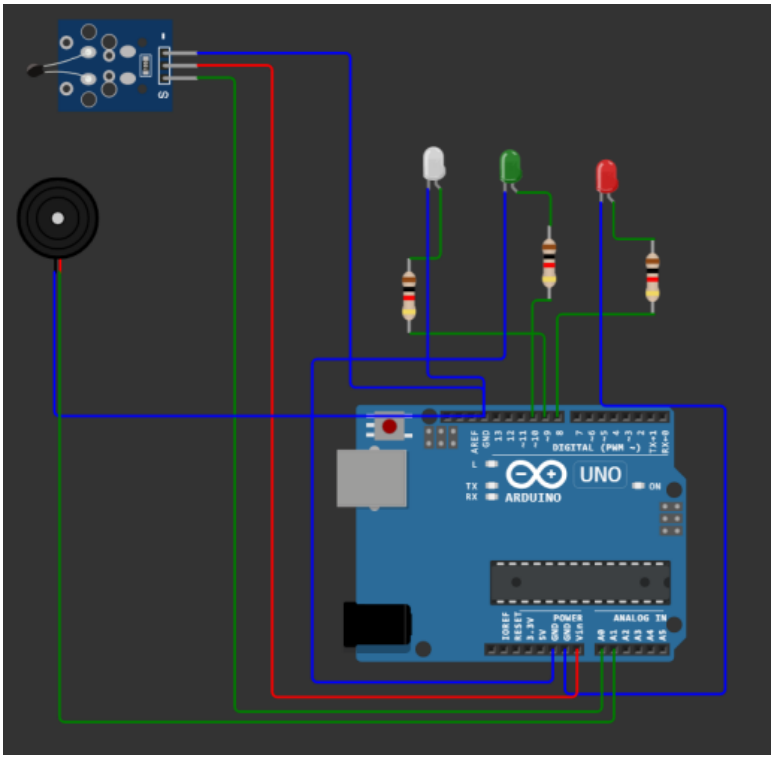
Buzzer + Temperatura

Objetivo de la práctica:

Una vez más, partiendo del circuito de la versión anterior, añadir un buzzer que genere un sonido. Este sonido ayudará a reforzar la alarma de estado crítico por exceso de temperatura que ya proporciona el diodo led rojo. El buzzer deberá emitir un sonido cuya frecuencia será de 750 Hz cuando la temperatura exceda de los 30°C. Se deberán comentar en el simulador todas las líneas de código de programación de todos los componentes empleados en esta versión.

1• Ha conectado correctamente los nuevos elementos añadidos al circuito
2• Los componentes añadidos realizan la función deseada
• Se han modificado y añadido las líneas necesarias de código para que el sistema funcione correctamente

1. Aquí podemos ver que hemos actualizado el circuito de la versión 1 y le hemos implementado un buzzer para avisar cuando el rango de la temperatura sea inestable. Podemos ver que el cable negro del buzzer está conectado al GND y El cable rojo al A1 para recibir la información.



2. El componente que hemos añadido que en este caso ha sido el buzzer, cumple su funcion necesaria. Cuando el rango de temperatura es mayor a 30 Celsius pita con una frecuencia de 750 Hz y un retrado de 100 milisegundos, en este video podemos ver su funcionamiento.

https://drive.google.com/file/d/1Rbin4w1bmtBzccSmiZnIsEAqyIv_xI1/view?usp=sharing

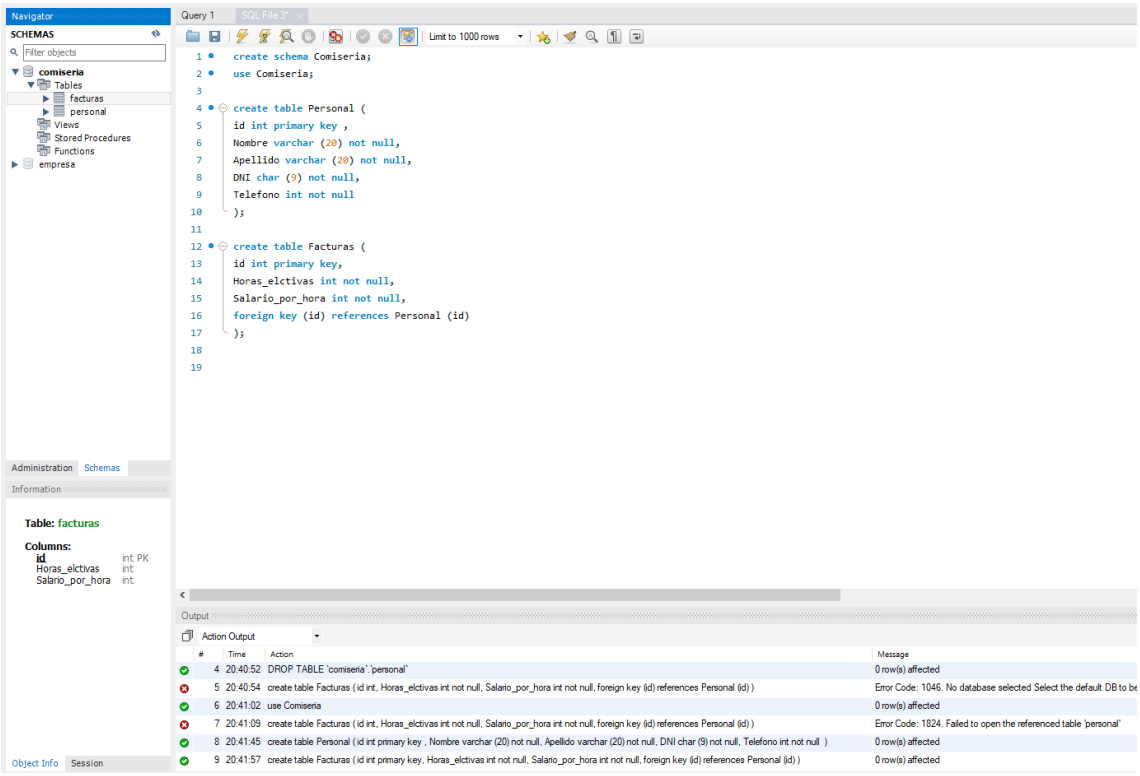
3. Aquí podemos ver el código. Para implementar el buzzer en el circuito he declarado la variable del buzzer, en el condicional de >=30.00 le he puesto para que suene con un retraso de 250 y una frecuencia de 750 Hz

```
int ledb = 9; //Declaramos el led blanco con el pin 9
int ledv = 10; //Declaramos el led verde con el pin 10
int ledr = 8;  //Declaramos el led rojo con el pin 8
const float BETA = 3950; //Declaramos el coeficiente beta del termisor
int buzzer = A1; //Declaramnos el buzzer con el pin A1


void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); // Ponemos una velocidad de transmision de 9600
  pinMode(ledb, OUTPUT); // Ponemos como salida el led blanco
  pinMode(ledv, OUTPUT);
  pinMode(ledr, OUTPUT);
  pinMode(buzzer, OUTPUT);

}

void loop() {
int analogValue = analogRead(A0);
float valorcelsius = 1 / (log(1 / (1023. / analogValue - 1)) / BETA + 1.0 / 298.15) - 273.15;

if(analogValue = analogRead(A0))
{
  digitalWrite(ledb, HIGH);
  delay(100);
  digitalWrite(ledb, LOW);
}

if (valorcelsius>= 30.00)
{
  tone(buzzer, 750);
    delay(250);
  digitalWrite(ledr, HIGH);
  noTone(buzzer);
    digitalWrite(ledv, LOW);

  Serial.print("⚠️**RANGO DE TEMPERATURA INESTABLE!!**⚠️");
}
else
{
  digitalWrite(ledv, HIGH);
  digitalWrite(ledr, LOW);
  Serial.print("✅ RANGO DE TEMPERATURA ESTABLE! ✅ ");
}

Serial.print(" TEMPERATURA: ");
Serial.print(valorcelsius);
Serial.println("°C");
delay(500);

}
```

Versión 3

BBDD

Creación de base de datos con dos tablas



Crearemos dos tablas una de ellas para tener un registro del personal la cual tendrán para acceder en cualquier pc del dominio además de hacer un seguimiento de cuándo están de servicio o no.

La segunda tiene un registro asociado a cada persona con su salario especificado en su contrato.

Sistemas

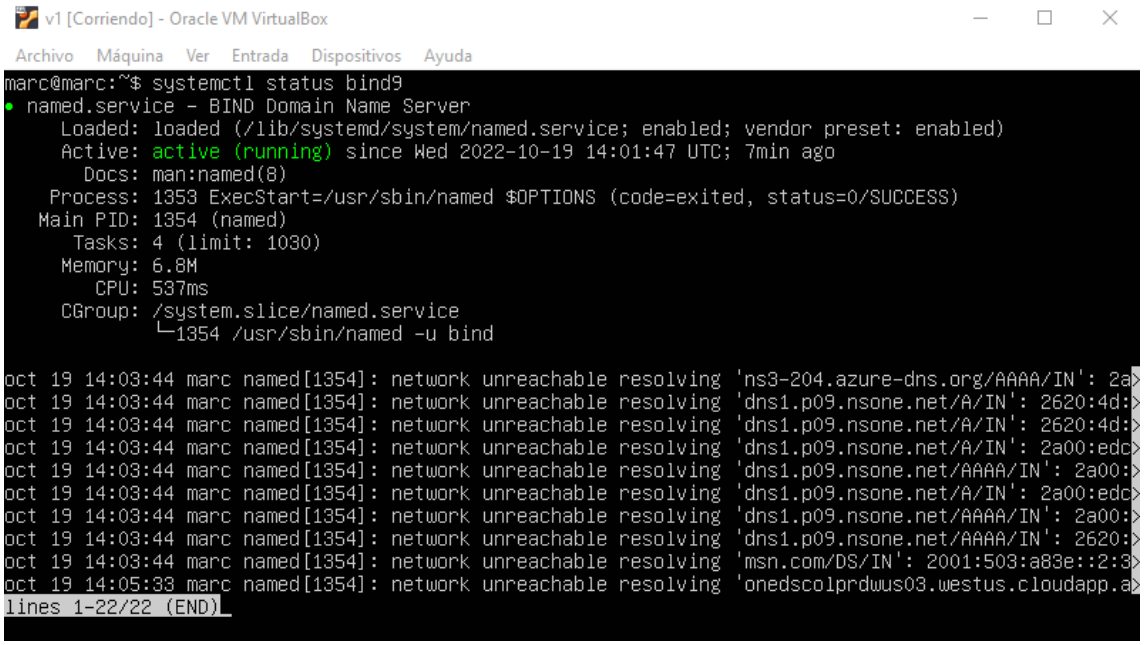
FTP

Instalación del servicio DNS

Para iniciar con esta parte de la práctica vamos a empezar instalando el servidor DNS, para eso solo deberemos ejecutar un comando, el cual es **"sudo apt install -y bind9"**.

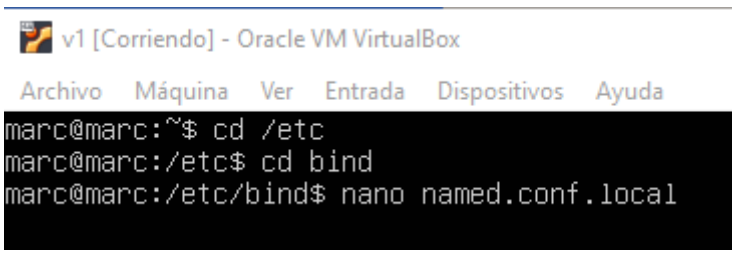
-
```
marc@marc:~$ sudo apt install -y bind9
```

Una vez el servidor está instalado comprobaremos si funciona y se ha instalado correctamente, para eso deberemos ejecutar el comando **"systemctl status bind9"**.



Creacion y configuracion de la zona **"dominio.local"**

Para empezar con la creación de la zona vamos a tener que encontrar el archivo el cual configurar para añadir la zona, una vez sepamos qué fichero deberemos cambiar vamos a tener que ir hacia ese directorio y con el nano vamos a cambiar y configurar su interior. Para ir hacia el directorio utilizaremos el comando **"cd /etc/bind"**, se puede hacer tanto individualmente (**"marc@marc:~$ cd /etc"** y luego **"marc@marc:/etc$ cd bind"**) o con el comando que he dicho anteriormente. Y a continuación haremos uso del nano para poder entrar en el fichero con **"nano named.conf.local"**.



Una vez entremos al nano de **"named.conf.local"**, vamos a tener que ir debajo del todo del fichero y crear la zona de **"grupo4.com"** que va a contener registros tipo A.
Una vez creada la zona **"grupo4.com"** vamos a crear otra zona, **"2.0.10.in-addr.arpa"**, que va a contener registros tipo **PTR**.

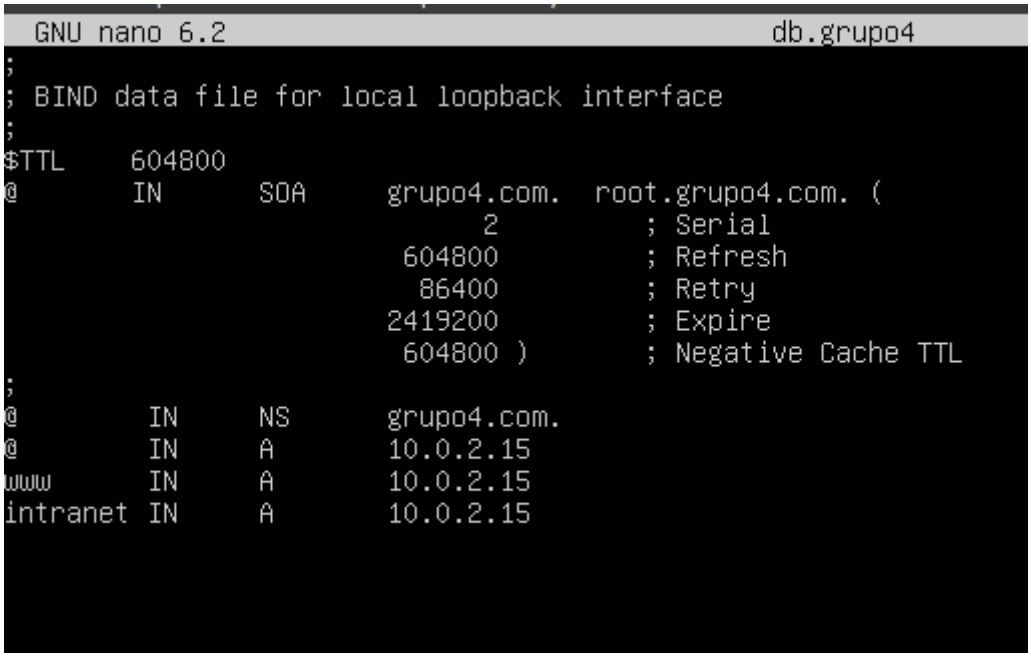
En cada una de las zonas vamos a decretar si son primarias o secundarias con el **"type master; o type slave;"**, luego de decretar si son primarias o secundarias vamos a marcar dónde están los registros definidos de cada zona, en el caso de la zona **"grupo4.com"** los registros definidos van a estar en el fichero **"db.dominio"** donde los registros van a ser tipo A. En el caso de la zona **"2.0.10.in-addr.arpa"** los registros definidos van a estar en el fichero **"db.190"** el cual va a contener registros tipo **PTR**.

```
  GNU nano 6.2                                named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "grupo4.com" {
type master;
file "/etc/bind/db.grupo4";
};

zone "2.0.10.in-addr.arpa" {
type master;
file "etc/bind/db.190";
};
```

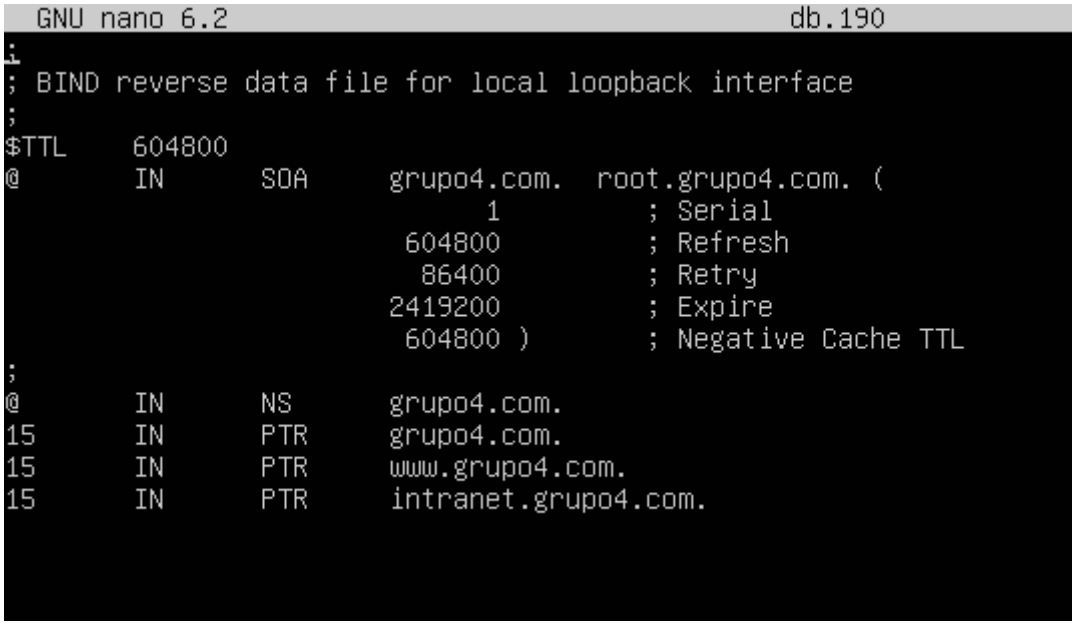
Luego de guardar el fichero vamos a crear los ficheros que van a contener los registros de las zonas, ya que no queremos crearlos de cero, vamos a copiar y a modificar posteriormente unos archivos que ya están en el sistema, los cuales son **"db.127"** y **"db.local"** respectivamente.

Luego de copiar los ficheros vamos a tener que configurarlos, para entrar dentro el fichero usaremos el comando nano como anteriormente hemos hecho. En el caso **"db.grupo4"** que es el que se mostrará a continuación, vamos a decretar que su zona es **"grupo4.com"**. Para poder decretar cuál es su zona y que tipo de registros deben de captar, vamos a cambiar los nombres de la configuración inicial por el nombre **"grupo4.com"** en **"db.grupo4"**.

Luego de haber asignado su zona, vamos a crear el registro tipo A y su dirección IP asignada, para eso vamos a ir a bajo del todo del fichero y vamos a crear/modificar las 3 últimas líneas que se pueden ver en el documento.



Luego de haber configurado el fichero "db.muntane" vamos a proseguir a configurar el siguiente fichero **"db.190"**. En el caso **"db.190"** que es el que se mostrará a continuación, vamos a decretar que su dominio es **"db.grupo4"** ya que en el fichero muntane.local figura los registros PTR para el anterior tipo A creado en db.muntane (pc1.muntane.local). De modo que al hacer un **"nslookup"** con el cliente declare que su IP es **"10.0.2.15"**.



Instalamos el servicio de web **"apache2"** y el servicio FTP **"vsftpd"** con el siguiente comando.

- ```
  marc@marc:~$ sudo apt-get install vsftpd
  ```
- ```
  marc@marc:~$ sudo apt-get install apache2_
  ```

Añadimos un nuevo usuario con **"adduser"** el cual se llama **"poffice"**

```
marc@marc:~$ sudo adduser poffice
Adding user `poffice' ...
Adding new group `poffice' (1001) ...
Adding new user `poffice' (1001) with group `poffice' ...
Creating home directory `/home/poffice' ...
Copying files from `/etc/skel' ...
New password: _
```

Con el comando **"usermod"** para cambiar la carpeta de inicio del usuario creado anteriormente

```
marc@marc:~$ sudo usermod -d /var/www/html poffice
```

Editamos el archivo para que los usuarios puedan editar y leer.c

```
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change
```

Editamos más el archivo para enjaular la carpeta del usuario definido anteriormente anteriormente

```
chroot_local_user=NO
chroot_list_enable=YES
# (default follows)
chroot_list_file=/etc/vsftpd.chroot_list
allow_writeable_chroot=YES_
#
```

Continuamos configurando para darle permisos totales a la carpeta

```
marc@marc:/etc$ cd
marc@marc:~$ sudo -i
root@marc:~# chmod -R 777 /var/www/html
root@marc:~# ll /var/www
total 12
drwxr-xr-x  3 root root 4096 Jan 22 16:41 ./
drwxr-xr-x 14 root root 4096 Jan 22 16:41 ../
drwxrwxrwx  2 root root 4096 Jan 22 16:41 html/
root@marc:~#
```

Creamos dos páginas web las cuales van a estar conectadas al DNS y serán las principales páginas web, una privada y la otra publica

```
root@marc:/etc/apache2/sites-available# ls
000-default.conf  default-ssl.conf  wwwmarc1.conf  wwwmarc2.conf
```

Creamos el HTML el cual va a a salir al ingresar el nombre de la web en google

```
  GNU nano 6.2                                        index.html
<html>
<h1> Hola, estas en el directorio publico </h1>
</html>
```

Configuramos los archivos para poder hacer que salga el respectivo HTML para su respectiva web

```
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/privado
        ServerName intranet.grupo4.com

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

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

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
        <Directory /var/www/html/grupopcx/privado>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride ALL
        Order allow,deny
        allow from all
        </Directory>
</VirtualHost>
```

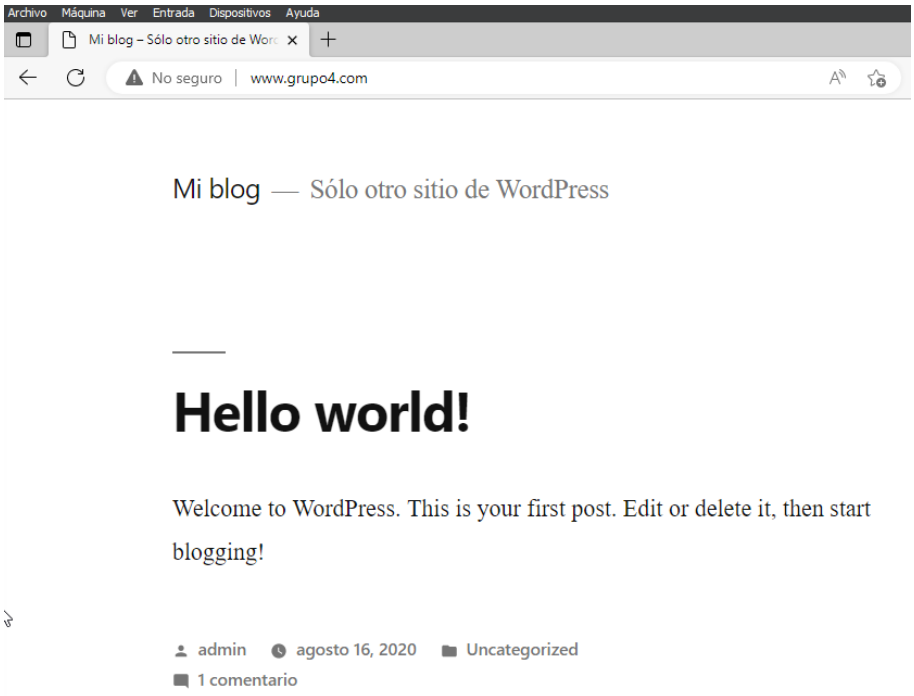
```
 GNU nano 6.2                                  wwwmarc2.conf *
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/publico
        ServerName www.grupo4.com

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

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

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

        <Directory /var/www/html/grupopcx/privado>
        Optinons Indexes FollowSymLinks MultiViews
        AllowOverride ALL
        Order aloow,deny
        allow from all
        </Directory>
</VirtualHost>
```

Comprobamos que funciona con el propio servidor de Ubuntu Server probando un ping

```
root@marc:~# ping grupo4.com
PING grupo4.com (10.0.2.15) 56(84) bytes of data.
64 bytes from marc (10.0.2.15): icmp_seq=1 ttl=64 time=0.012 ms
64 bytes from marc (10.0.2.15): icmp_seq=2 ttl=64 time=0.026 ms
^C64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.038 ms

root@marc:~# ping www.grupo4.com
PING www.grupo4.com (10.0.2.15) 56(84) bytes of data.
64 bytes from marc (10.0.2.15): icmp_seq=1 ttl=64 time=0.012 ms
^C64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=0.034 ms

root@marc:~# ping intranet.grupo4.com
PING intranet.grupo4.com (10.0.2.15) 56(84) bytes of data.
64 bytes from marc (10.0.2.15): icmp_seq=1 ttl=64 time=0.012 ms
64 bytes from marc (10.0.2.15): icmp_seq=2 ttl=64 time=0.023 ms
64 bytes from marc (10.0.2.15): icmp_seq=3 ttl=64 time=0.047 ms
```

Para finalizar comprobamos las páginas web con el usuario

En la siguiente web de “grupo4.com” podemos ver que funciona pero alguien mas ya habia creado su página web con ese nombre de host y se me sincroniza a esa pagina y no a la creada anteriormente


Archivo Máquina Ver Entrada Dispositivos Ayuda
Mi blog – Sólo otro sitio de Wor
No seguro | www.grupo4.com
Mi blog — Sólo otro sitio de WordPress
Hello world!
Welcome to WordPress. This is your first post. Edit or delete it, then start blogging!
admin agosto 16, 2020 Uncategorized
1 comentario

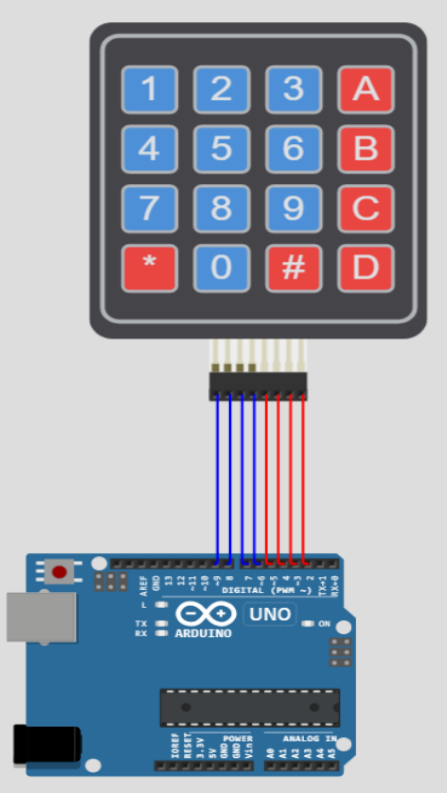
Robótica

Código de Acceso

Objetivo de la práctica:

En esta versión tenemos que diseñar un sistema de control de acceso, mediante una clave introducida en el teclado matricial 4x4, con un total de 30 contraseñas asignadas a 30 trabajadores de la empresa. Deberemos mostrar por pantalla si la clave es correcta o no.

1. **Ha conectado correctamente el teclado matricial 4x4 con la placa Arduino**
   Como podemos ver en la imagen, tenemos conectado el teclado matricial a la placa base, los cables azules son los pines para las filas y los cables rojos para las columnas.



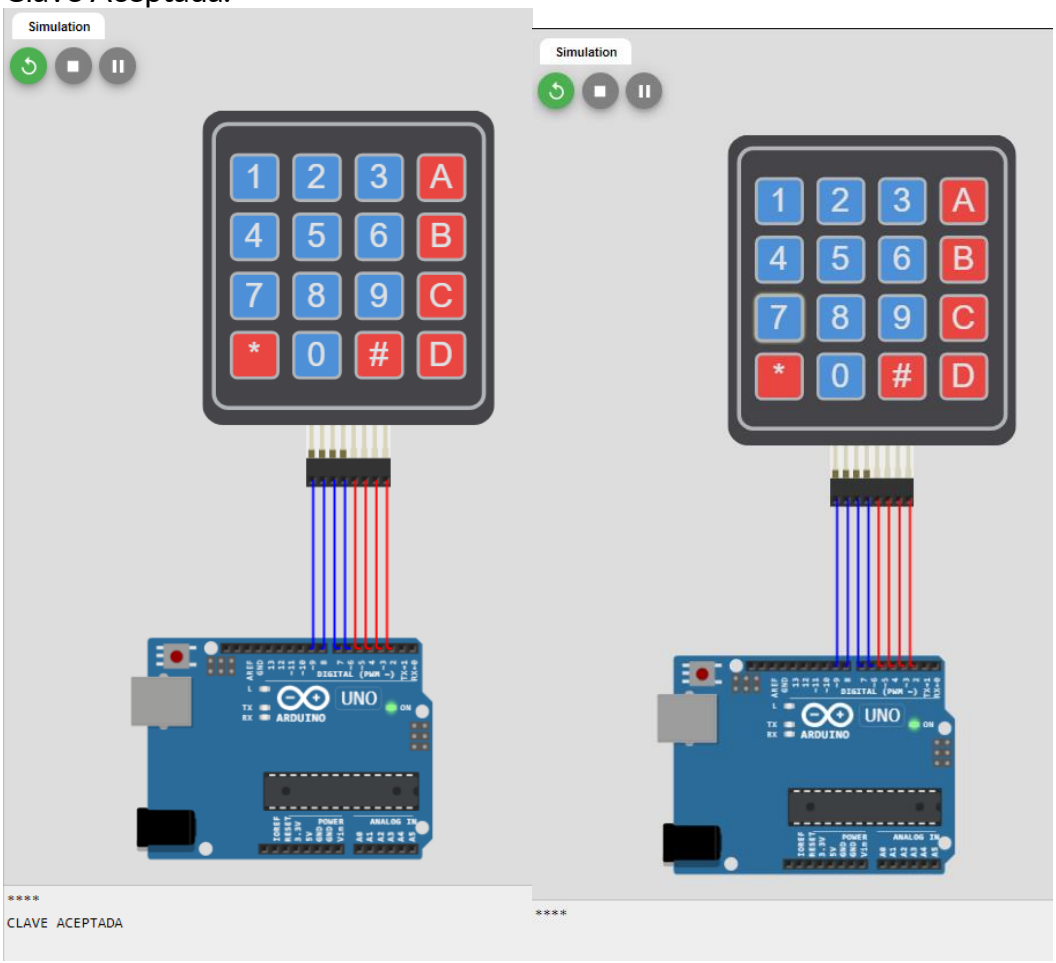
2. **Ha programado las 30 claves para los 30 trabajadores:**
   En la captura que vemos a continuación podemos ver el código comentado:

```
#include <Keypad.h> // Añadimos la libreria del teclado matricial
const byte ROWS = 4; // Numero de filas
const byte COLS = 4; // Numero de columnas

char keys[ROWS][COLS] = { //Creamos una matriz con los caracteres del teclado
  { '1', '2', '3'},
  { '4', '5', '6'},
  { '7', '8', '9'},
  { ' ', '0',' '},
};
byte rowPins[ROWS] = {9, 8, 7, 6 };  // Pines de Arduino para las filas
byte colPins[COLS] = {5, 4, 3, 2}; // Pines de Arduino para las columnas

char i[4] = {0,0,0,0}; // Declaramos una matriz de cuatro variables
char key = 0; //Variable que contendrá cada uno de lso digitos del teclado
int j = 0; // Con esta variable que el char I acumule los digitos
int cont = 0; //variable que se encarga de contar que tenemos 4 numeros

Keypad tecladomatricial = Keypad (makeKeymap (keys), rowPins, colPins, ROWS, COLS ); // Creamos un objeto teclado
// donde pasamos la matriz de las teclas, pines de las filas, numero de filas, numero de columnas

void setup(){
  Serial.begin(9600); //Iniciamos comunicación serie
}

void loop() {
  while (cont < 4) { // Mientas que el contador de numeros sea menor a 4

char key = tecladomatricial.getKey(); //La variable guarda las teclas pulsadas

  i[j] = key; //Si la variable contiene datos

  if (key) { //Si pusamlos una tecla
      Serial.print('*'); //Entonces que salga un asterisco en cada digito

      j=j+1; // Incrementamos la variable J
      cont = cont+1; // incrementamos la variable contados de numeros introducidos

    }
  }


//CONTRASEÑA USUARIO2////////////////////////////////////////////////
if(i[0] == '1' && i[1] == '2' && i[2] == '1' && i[3] == '2') // En la primera posicion de la variable array tenemos
// el 1, en la segunda un 2, en la tercera un 1 y en la cuarta un 2.

{ // Mostramos por el monitor serie
  Serial.println();
  Serial.println("CLAVE ACEPTADA ");
  Serial.println();
  analogWrite(A0, 255);

  i[0]=0; // Le ponemos al valor i
  i[1]=0; // Ponemos un cero a la variable array i a la espera de la introduccion de un digito por parte del usuario
  i[2]=0;
  i[3]=0;
  // Y asi con todos los usuarios de los trabajadores pero cambiando los digitos de la contraseña
```

Aquí muestro un video donde se ven los 30 usuarios:
https://drive.google.com/file/d/14Ah7IWv_Y9pK8Rr0D3tqNyvE09gADLO-/view?usp=sharing

3. Aparece la información sobre el acceso o no acceso de los trabajadores por el Serial Monitor

En la captura que vemos a continuación, al poner una contraseña correcta pone Clave Aceptada.

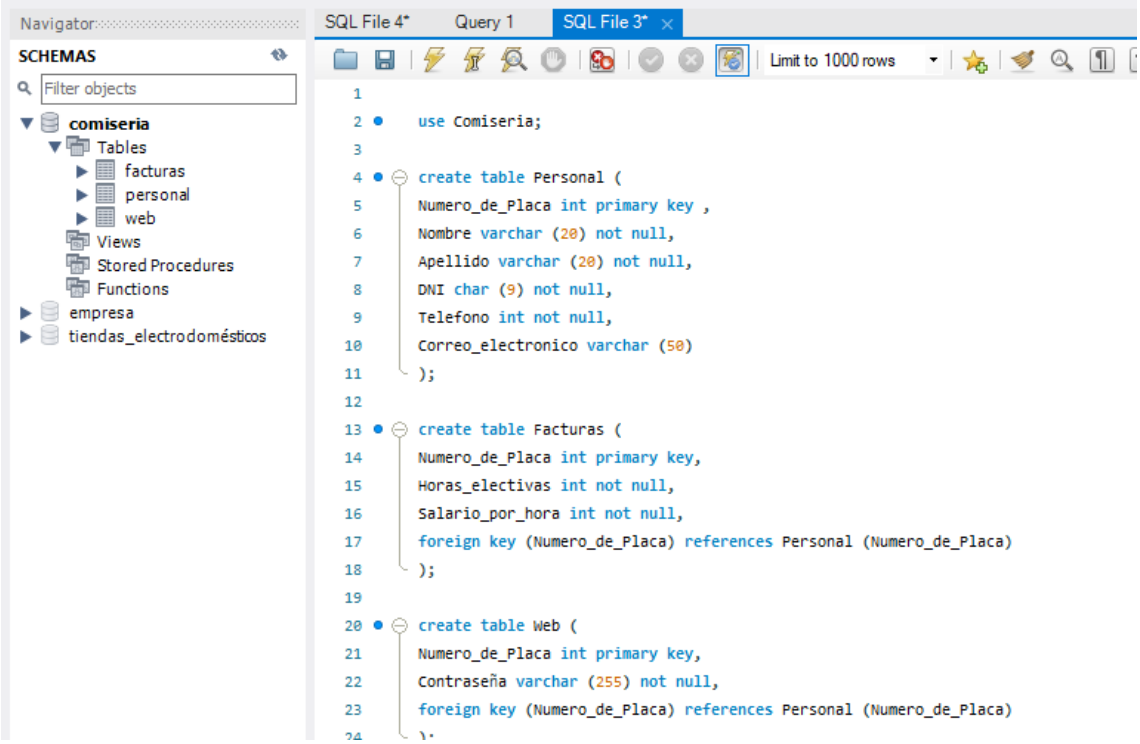


Wokwi: https://wokwi.com/projects/354294580838070273

Versión 4

BBDD

Creación de tabla WEB y Consultas a la base de datos



Ampliamos la base de datos añadiendo la tabla web además de rellenar estas tablas con datos

```
insert Personal (`Numero_de_Placa`,`Nombre`,`Apellido`,`DNI`,`Telefono`,`Correo_electronico`)
values
("7","Antonio","Corola","13456728Q","642321518","Antonio.Corola7@comiseria.com");

insert Facturas (`Numero_de_Placa`,`Horas_electivas`,`Salario_por_hora`)
values
("7","8","9");

insert Web (`Numero_de_Placa`,`Contraseña`)
values
("7","2345");



insert Personal (`Numero_de_Placa`,`Nombre`,`Apellido`,`DNI`,`Telefono`,`Correo_electronico`)
values
("25","Carlos","Pastor","12345678Q","345226645","CarlosPastor25@comiseria.com");

insert Facturas (`Numero_de_Placa`,`Horas_electivas`,`Salario_por_hora`)
values
("25","6","9");

INSERT INTO Web (Numero_de_Placa, Contraseña)
VALUES ('25', '1234');
```

```
144 •  select * from Facturas;
```

Result Grid | Filter Rows: | Edit:

Numero_de_Placa	Horas_electivas	Salario_por_hora
7	8	9
14	8	8
16	7	9
25	6	9
51	8	8
68	5	12
76	8	11

```
145 •  select * from Personal;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

Numero_de_Placa	Nombre	Apellido	DNI	Telefono	Correo_electronico
7	Antonio	Corola	13456728Q	642321518	Antonio.Corola7@comiseria.com
14	Miguel	Sanchez	34201328W	642868691	Miguel.Sanchez14@comiseria.com
16	Carla	Monte	31951328T	642868501	Carla.Monte16@comiseria.com
25	Carlos	Pastor	12345678Q	345226645	CarlosPastor25@comiseria.com
51	Paco	Sanchez	71201328S	644268691	Paco.Sanchez51@comiseria.com
68	Perkin	Merto	69924739A	644153256	Perkin.Merto68@comiseria.com
76	Alonso	Casa	31956728Q	642378508	Alonso.Casa76@comiseria.com

```
146 •  select * from web;
```

Result Grid | Filter Rows:

Numero_de_Placa	Contraseña
7	2345
14	1113
16	5631
25	1234
51	1712
68	6305
76	5576

A continuación, haremos diferentes consultas a la base de datos

Select con operación numérica.

```
122 •  SELECT Numero_de_Placa, Horas_electivas * Salario_por_hora AS Salario_dia
123    FROM Facturas;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Numero_de_Placa	Salario_dia
7	72
14	64
16	63
25	54
51	64
68	60
76	88

```
125 •  SELECT web.Numero_de_Placa, web.contraseña, Facturas.Horas_electivas * Facturas.Salario_por_hora AS Salario_dia
126    FROM web
127    JOIN Facturas
128    ON web.Numero_de_Placa = Facturas.Numero_de_Placa;
129
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Numero_de_Placa	contraseña	Salario_dia
7	2345	72
14	1113	64
16	5631	63
25	1234	54
51	1712	64
68	6305	60
76	5576	88

Select con función CONCAT.

```
0 •  SELECT CONCAT(Nombre, ' ', Apellido) AS Nombre_Completo, Numero_de_Placa, Correo_electronico
```

sult Grid | Filter Rows: | Export: | Wrap Cell Content:

Nombre_Completo	Numero_de_Placa	Correo_electronico
Antonio Corola	7	Antonio.Corola7@comiseria.com
Miguel Sanchez	14	Miguel.Sanchez14@comiseria.com
Carla Monte	16	Carla.Monte16@comiseria.com
Carlos Pastor	25	CarlosPastor25@comiseria.com
Paco Sanchez	51	Paco.Sanchez51@comiseria.com
Perkin Merto	68	Perkin.Merto68@comiseria.com
Alonso Casa	76	Alonso.Casa76@comiseria.com

select de número de placa y contraseña de la tabla web.

```
133 •  select Numero_de_Placa, Contraseña from Web;
```

Result Grid | Filter Rows: | Edit:

Numero_de_Placa	Contraseña
7	2345
14	1113
16	5631
25	1234
51	1712
68	6305
76	5576
NULL	NULL

select con where filtrando el número de placa mayor que 20.

```
135 •  SELECT *
136    FROM Personal
137    WHERE Numero_de_Placa > 20;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wra

Numero_de_Placa	Nombre	Apellido	DNI	Telefono	Correo_electronico
25	Carlos	Pastor	12345678Q	345226645	CarlosPastor25@comiseria.com
51	Paco	Sanchez	71201328S	644268691	Paco.Sanchez51@comiseria.com
68	Perkin	Merto	69924739A	644153256	Perkin.Merto68@comiseria.com
76	Alonso	Casa	31956728Q	642378508	Alonso.Casa76@comiseria.com

select numero de placa + horas electivas multiplicando salario hora mostrado el resultado como una única columna que se filtra por la columna total es mayor a 70.

```
139 •  SELECT Numero_de_Placa, Horas_electivas * Salario_por_hora AS Total
140    FROM Facturas
141    WHERE Horas_electivas * Salario_por_hora > 70;
142
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Numero_de_Placa	Total
7	72
76	88

Sistemas

Pagina php

Para empezar con la página PHP vamos a instalar el PHP con el siguiente comando

```
marc@marc:~$ sudo apt-get install php_
```

Vamos donde tenemos los html de la pagina publica y privado y creamos la carpeta php.

```
marc@marc:/var/www/html$ ls
index.html  php  privado  publico
```

En la carpeta creada anteriormente crearemos un fichero PHP que va a contener la página web para poder loguearse.

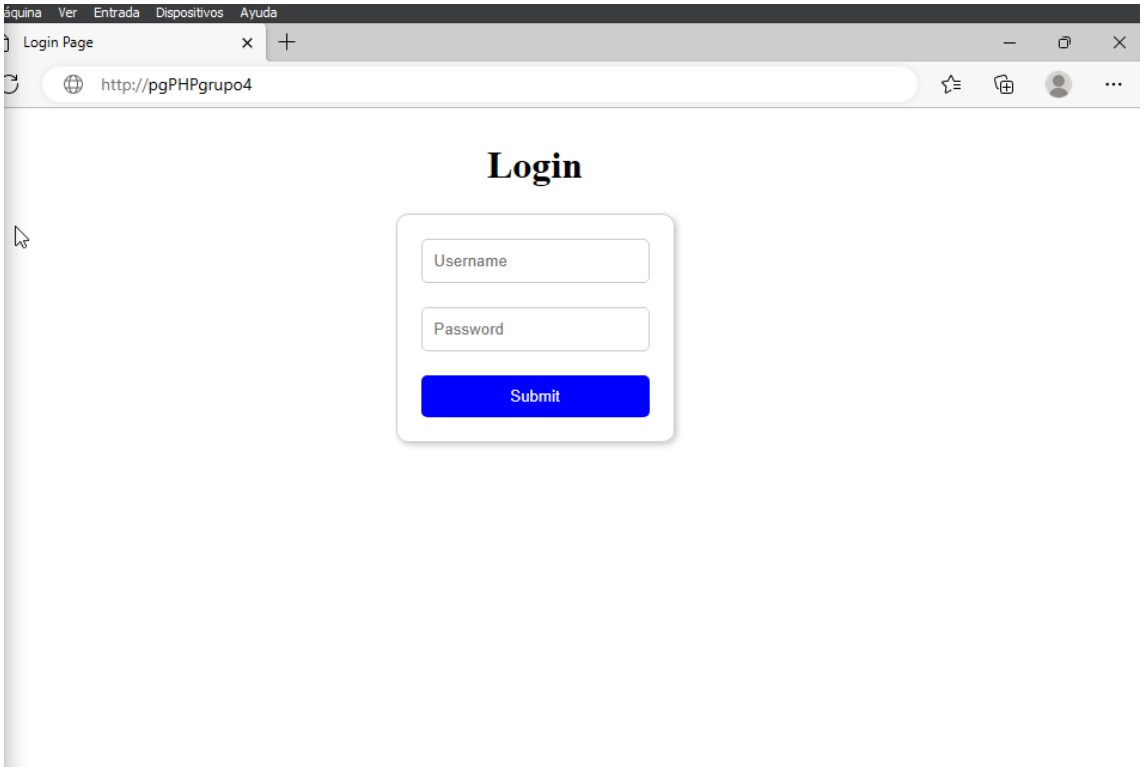
```
  GNU nano 6.2                                    index.php
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Login Page</title>
  <style type="text/css">
    .container {
      display: flex;
      flex-direction: column;
      align-items: center;
    }
    form {
      display: flex;
      flex-direction: column;
      padding: 20px;
      border: 1px solid #ccc;
      border-radius: 10px;
      box-shadow: 2px 2px 5px #ccc;
    }
    input[type="text"], input[type="password"] {
      padding: 10px;
      margin-bottom: 20px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }
    input[type="submit"] {
      padding: 10px 20px;
      background-color: blue;
      color: white;
      border: none;
      border-radius: 5px;
      cursor: pointer;
    }
```

```
 GNU nano 6.2                                    index.php
      border: none;
      border-radius: 5px;
      cursor: pointer;
    }
  </style>
</head>
<body>
  <div class="container">
    <h1>Login</h1>
    <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
      <input type="text" name="username" placeholder="Username">
      <input type="password" name="password" placeholder="Password">
      <input type="submit" value="Submit">
    </form>
    <?php
      if(isset($_POST['submit'])){
        $username = $_POST['username'];
        $password = $_POST['password'];

        if($username === 'admin' && $password === 'password'){
          session_start();
          $_SESSION['logged_in'] = true;
          header("Location: dashboard.php");
          exit;
        }else{
          echo "<p>Invalid username or password</p>";
        }
      }
    ?>
  </div>
</body>
</html>
```

Nos vamos en “/apache2/sites-available/” para crear la carpeta que conectará a la página web con el DNS.

```
 GNU nano 6.2                  /etc/apache2/sites-available/ProyectoA1.com *
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/pagina.php_
        ServerName pgPHPgrupo4

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

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

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
        <Directory /var/www/html/grupopcx/privado>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride ALL
        Order allow,deny
```

Para finalizar, vamos a ir a la máquina cliente y vamos a comprobar que el DNS y el apache2 funcionen correctamente y la página PHP funcione a la perfección.



Luego seguiremos con la parte de mysql, para empezar abriremos el mysql.

```
root@marc:~# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

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> create user 'policia'@'%' identified by 'policia123';
Query OK, 0 rows affected (0,64 sec)
```

Proseguiremos creando el usuario y le añadiremos los privilegios.

```
mysql> grant all privileges on *.* to 'policia'@'%' with grant option;
Query OK, 0 rows affected (0,08 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0,02 sec)
```

Tocará seguir con un select para comprobar que se ha creado correctamente.

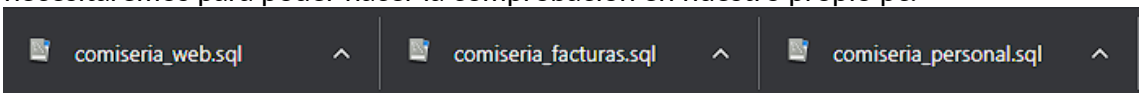
```
mysql> SELECT User, Host FROM mysql.user;
+------------------+-----------+
| User             | Host      |
+------------------+-----------+
| policia          | %         |
| debian-sys-maint | localhost |
| mysql.infoschema | localhost |
| mysql.session    | localhost |
| mysql.sys        | localhost |
| root             | localhost |
+------------------+-----------+
6 rows in set (0,00 sec)
```

Luego deberemos entrar en mysqld.conf para poder configurar la carpeta para que sea accesible para todos.

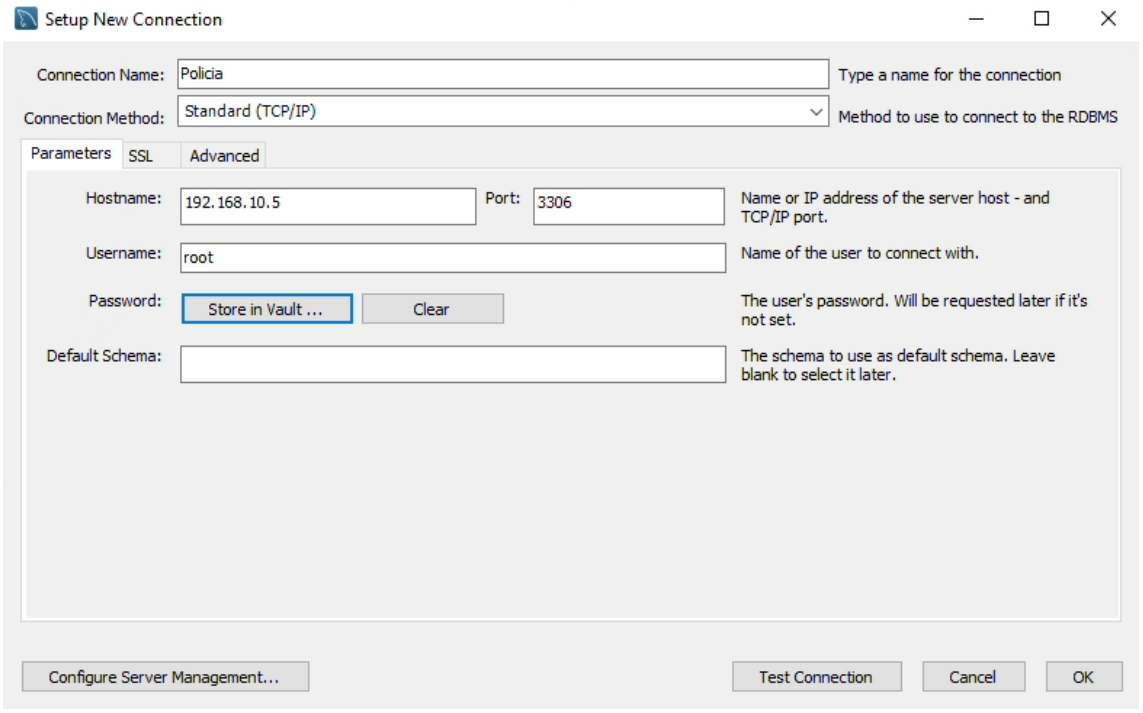
```
root@marc:/etc/mysql/mysql.conf.d# ls
mysql.cnf  mysqld.cnf
```

```
  GNU nano 6.2                                    mysqld.cnf *
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and wi
# --print-defaults to see which it would actually understand and us
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user            = mysql
# pid-file      = /var/run/mysqld/mysqld.pid
# socket        = /var/run/mysqld/mysqld.sock
# port          = 3306
# datadir       = /var/lib/mysql


# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-syste
# tmpdir                = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
mysqlx-bind-address     = 0.0.0.0
#
```

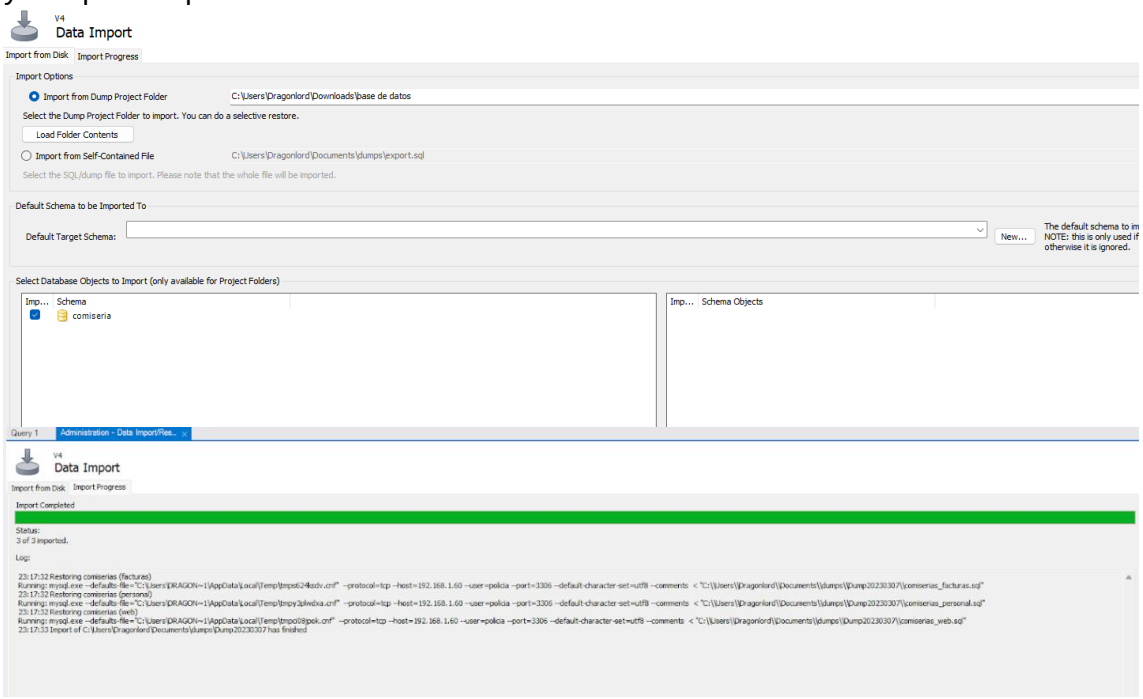
Seguiremos importando la base de datos de nuestro compañero David, la cual necesitaremos para poder hacer la comprobación en nuestro propio pc.

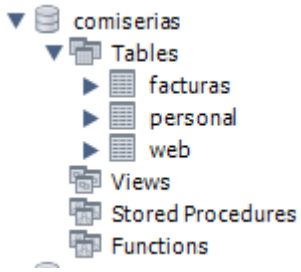


Crearemos la conexión con nuestro servidor de ubuntu.



Una vez dentro tendremos que importar los archivos que tenemos de la base de datos y comprobar que funcionan.





Luego lo único que deberemos hacer será un show database, un USE y un select para comprobar que la base de datos funciona correctamente en nuestro servidor.

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| comiserias         |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0,00 sec)
```

```
mysql> USE comiserias;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

```
mysql> SELECT * FROM comiserias.personal;
+-----------------+---------+----------+-----------+-----------+-----------------------------+
| Numero_de_Placa | Nombre  | Apellido | DNI       | Telefono  | Correo_electronico          |
+-----------------+---------+----------+-----------+-----------+-----------------------------+
|               7 | Antonio | Corola   | 13456728Q | 642321518 | Antonio.Corola7@comiseria.com |
+-----------------+---------+----------+-----------+-----------+-----------------------------+
1 row in set (0,00 sec)
```

Robótica

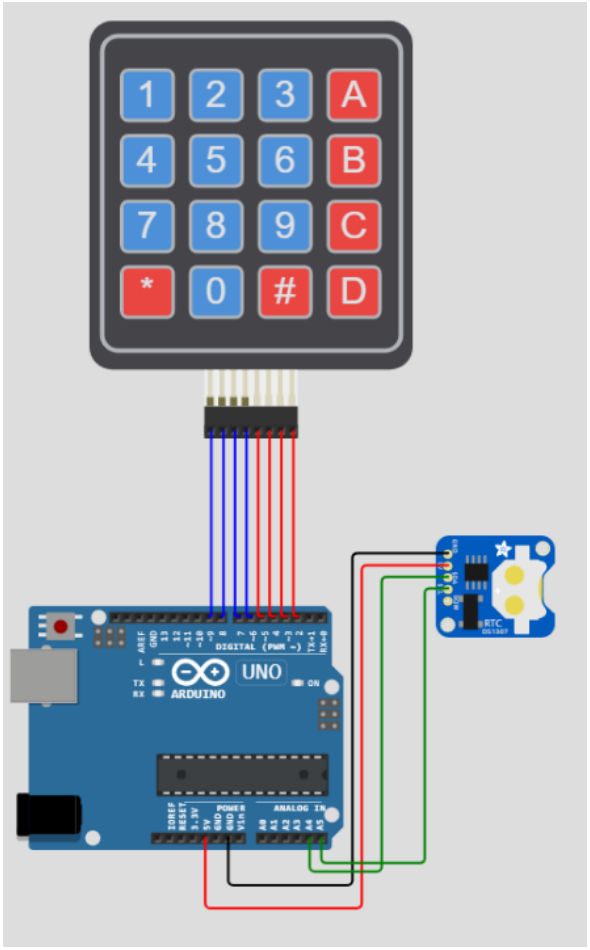
Código de Acceso + INFORMACIÓN USUARIOS

Objetivo de la práctica:

En esta versión, partiendo del circuito anterior tenemos que añadir información a cada trabajador al que le hemos asignado una contraseña. Tendremos que poner información como la ID del Trabajador, Nombre y Apellido y a que hora ha accedido. Si el código de acceso es incorrecto, pues por pantalla saldrá clave incorrecta

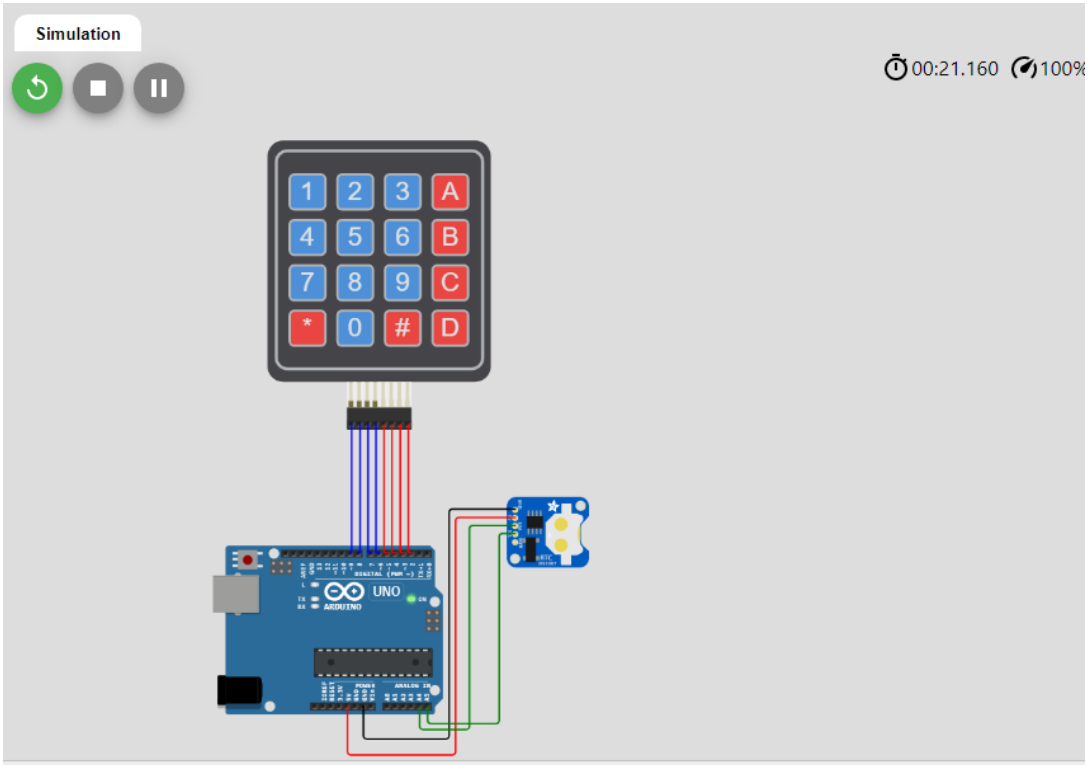
1. ***Ha conectado correctamente el RTC con la placa Arduino***

Como podemos ver en la captura hemos conectado correctamente el RTC en la placa arduino:



2. El módulo RTC añadido realizan la función deseada (muestra la fecha y hora actuales):

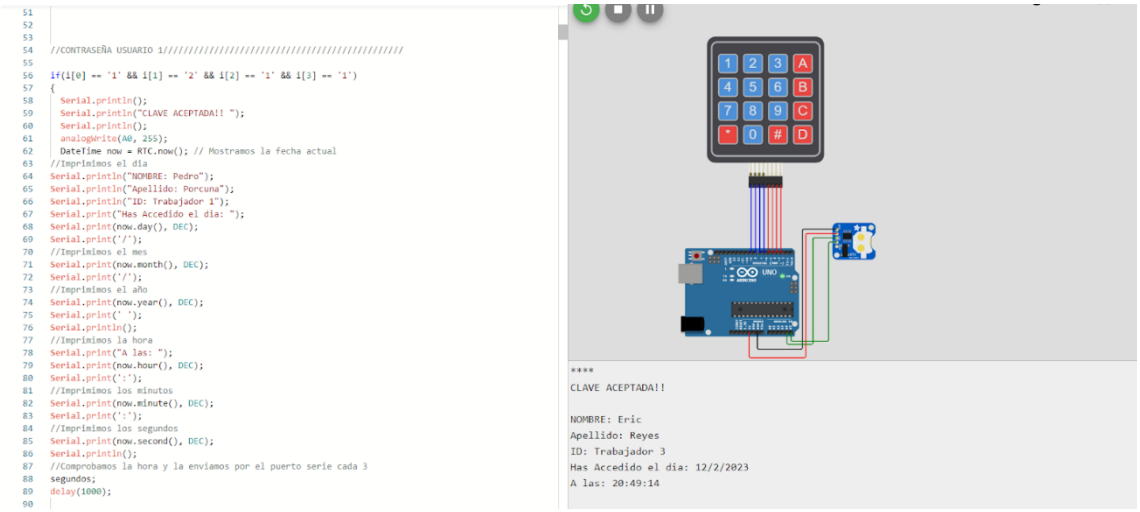
Aquí podemos ver que se muestra toda la información + la fecha de inicio de sesión:




```
****
CLAVE ACEPTADA!!

NOMBRE: Eric
Apellido: Reyes
ID: Trabajador 3
Has Accedido el dia: 12/2/2023
A las: 20:49:14
```

3. Se han modificado y añadido las líneas necesarias de código para que el sistema funcione correctamente:

En cada contraseña he modificado el código de manera que cuando cada vez que se introduzca la clave correcta mostrará la información necesaria:



```
//CONTRASEÑA USUARIO 1/////////////////////////////////////////////

if(i[0] == '1' && i[1] == '2' && i[2] == '1' && i[3] == '1')
{
  Serial.println();
  Serial.println("CLAVE ACEPTADA!! ");
  Serial.println();
  analogWrite(A0, 255);
  DateTime now = RTC.now(); // Mostramos la fecha actual
//Imprimimos el dia
Serial.println("NOMBRE: Pedro");
Serial.println("Apellido: Porcuna");
Serial.println("ID: Trabajador 1");
Serial.print("Has Accedido el dia: ");
Serial.print(now.day(), DEC);
Serial.print('/');
//Imprimimos el mes
Serial.print(now.month(), DEC);
Serial.print('/');
//Imprimimos el año
Serial.print(now.year(), DEC);
Serial.print(' ');
Serial.println();
//Imprimimos la hora
Serial.print("A las: ");
Serial.print(now.hour(), DEC);
Serial.print(':');
//Imprimimos los minutos
Serial.print(now.minute(), DEC);
Serial.print(':');
//Imprimimos los segundos
Serial.print(now.second(), DEC);
Serial.println();
//Comprobamos la hora y la enviamos por el puerto serie cada 3
segundos;
delay(1000);
```

```
****
CLAVE ACEPTADA!!

NOMBRE: Eric
Apellido: Reyes
ID: Trabajador 3
Has Accedido el dia: 12/2/2023
A las: 20:49:14
```

WOKWI: https://wokwi.com/projects/356014772429795329

En esta versión hemos tenido que hacer el circuito de manera física, aqui muestro un video donde he montado el circuito y podemos ver en el serial monitor la información de cada trabajador:

LINK VIDEO: https://youtu.be/5wACzRjAWpE

Versión 5

BBDD

Creación de tabla WEB y Consultas a la base de datos

Actualización de tabla facturas utilizando update.

```
149 •   ALTER TABLE Facturas ADD COLUMN Salario_mensual NUMERIC;
150
151 •   update Facturas
152     set `Salario_mensual` = (Horas_electivas * Salario_por_hora) * 30;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

Numero_de_Placa	Horas_electivas	Salario_por_hora	Salario_mensual
7	8	9	2160
14	8	8	1920
16	7	9	1890
25	6	9	1620
51	8	8	1920
76	8	11	2640
109	5	12	1800

Numero_de_Placa	Nombre	Apellido	DNI	Telefono	Correo_electronico
7	Antonio	Corola	13456728Q	642321518	Antonio.Corola7@comiseria.com
14	Miguel	Sanchez	34201328W	642868691	Miguel.Sanchez14@comiseria.com
16	Carla	Monte	31951328T	642868501	Carla.Monte16@comiseria.com
25	Carlos	Pastor	12345678Q	345226645	CarlosPastor25@comiseria.com
51	Paco	Sanchez	71201328S	644268691	Paco.Sanchez51@comiseria.com
68	Perkin	Merto	69924739A	644153256	Perkin.Merto68@comiseria.com
76	Alonso	Casa	31956728Q	642378508	Alonso.Casa76@comiseria.com
109	Alberto	Matos	69234539B	644166956	Alberto.Matos109@comiseria.com

Numero_de_Placa	Horas_electivas	Salario_por_hora	Salario_mensual
7	8	9	2160
14	8	8	1920
16	7	9	1890
25	6	9	1620
51	8	8	1920
76	8	11	2640
109	5	12	1800

1. Insertamos un nuevo usuario.

```
154 •  insert Personal
155    values
156    ("109","Alberto","Matos","69234539B","644166956","Alberto.Matos109@comiseria.com");
157
158 •  insert Facturas
159    values
160    ("109","5","12");
161
162 •  insert Web (`Numero_de_Placa`,`Contraseña`)
163    values
164    ("109","5555");
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Numero_de_Placa	Contraseña
7	2345
14	1113
16	5631
25	1234
51	1712
68	6305
76	5576
109	5555

2. Cambiamos el password de un usuario.

```
166 •  update web
167    set `Contraseña` = 9555 where Numero_de_Placa = 109;
```

Result Grid | Filter Rows: | Edit: | Export/I

Numero_de_Placa	Contraseña
7	2345
14	1113
16	5631
25	1234
51	1712
68	6305
76	5576
109	9555

3. Eliminamos uno de los usuarios de la base de datos.

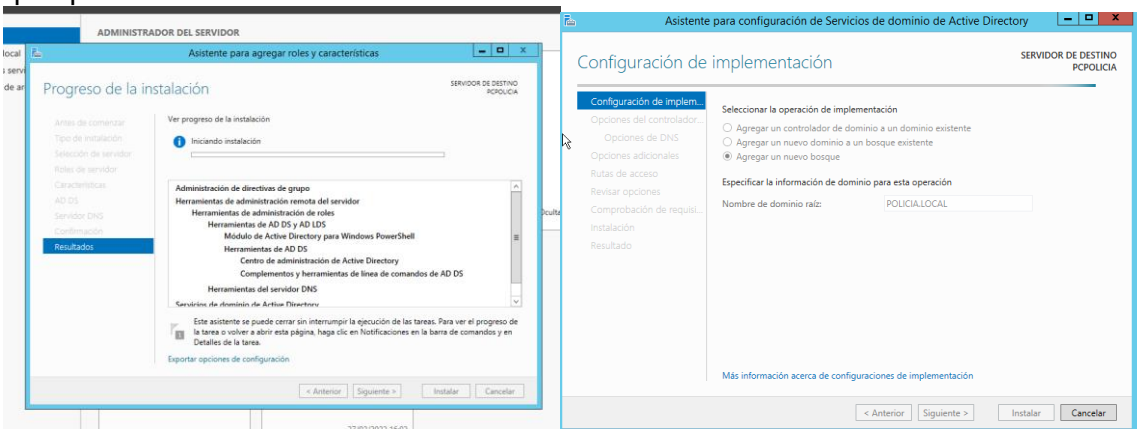
```
DELETE web, Facturas
FROM web
INNER JOIN Facturas ON web.Numero_de_Placa = Facturas.Numero_de_Placa
WHERE web.Numero_de_Placa = 68;

delete from Personal
where Numero_de_Placa = 68;
```

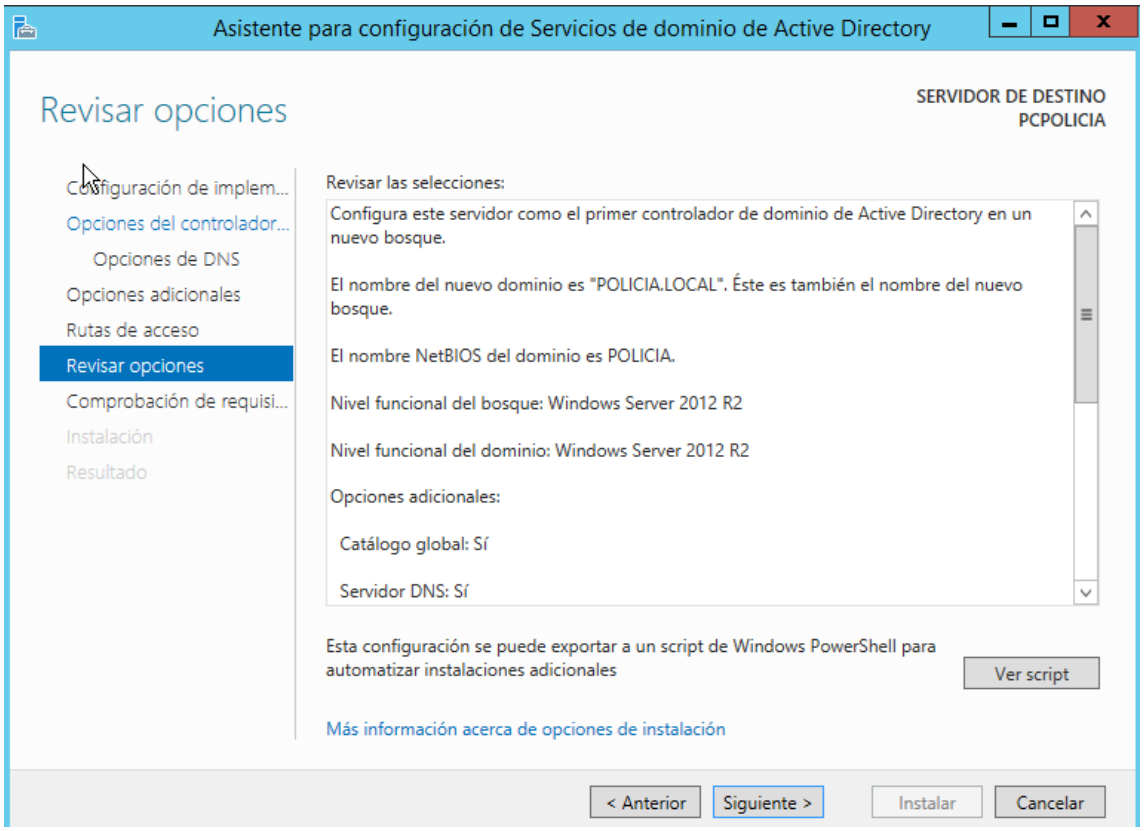
Numero_de_Placa	Contraseña
7	2345
14	1113
16	5631
25	1234
51	1712
76	5576
109	9555

Sistemas

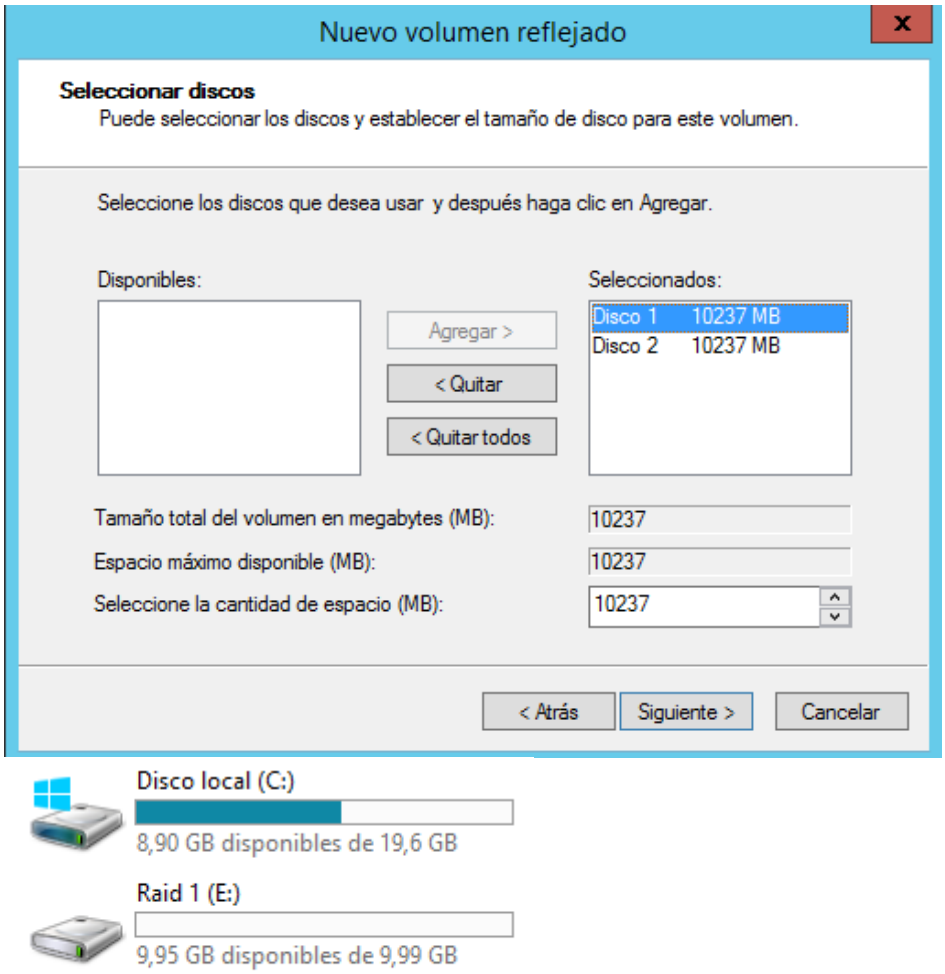
Active directory

Para empezar instalaremos el servidor active directory y crearemos el dominio de **"POLICIA.LOCAL"** que posteriormente veremos que nos sirve como base para todo lo que queremos hacer.

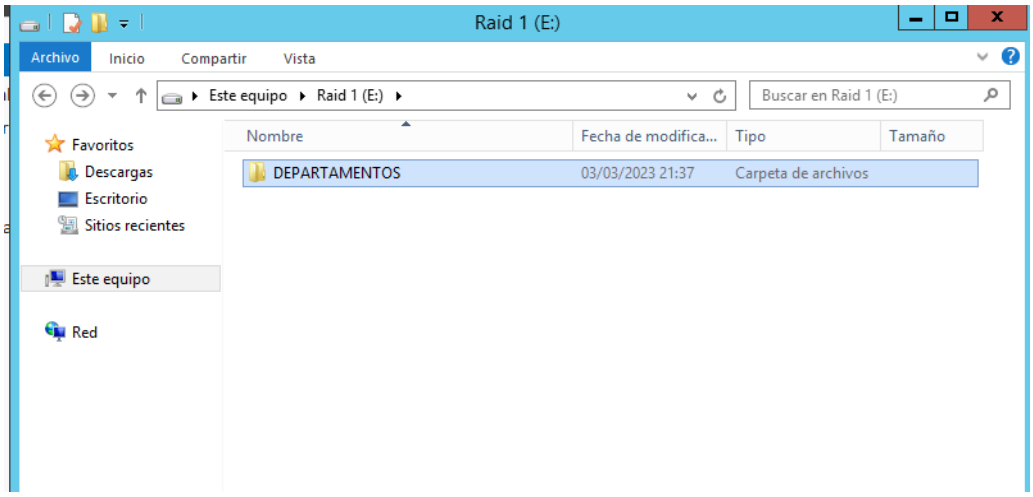


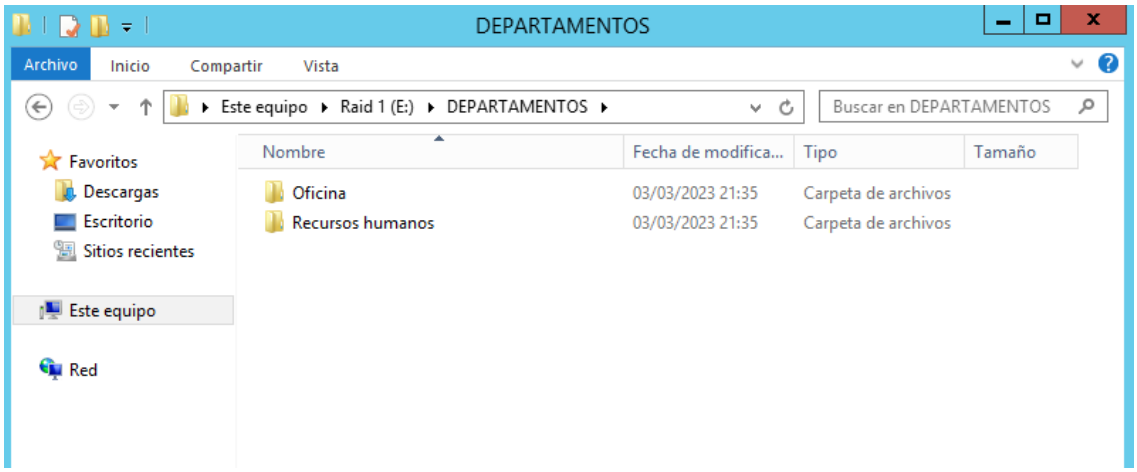


Continuaremos haciendo una nueva Raid con los dos discos sobrantes que teniamos.

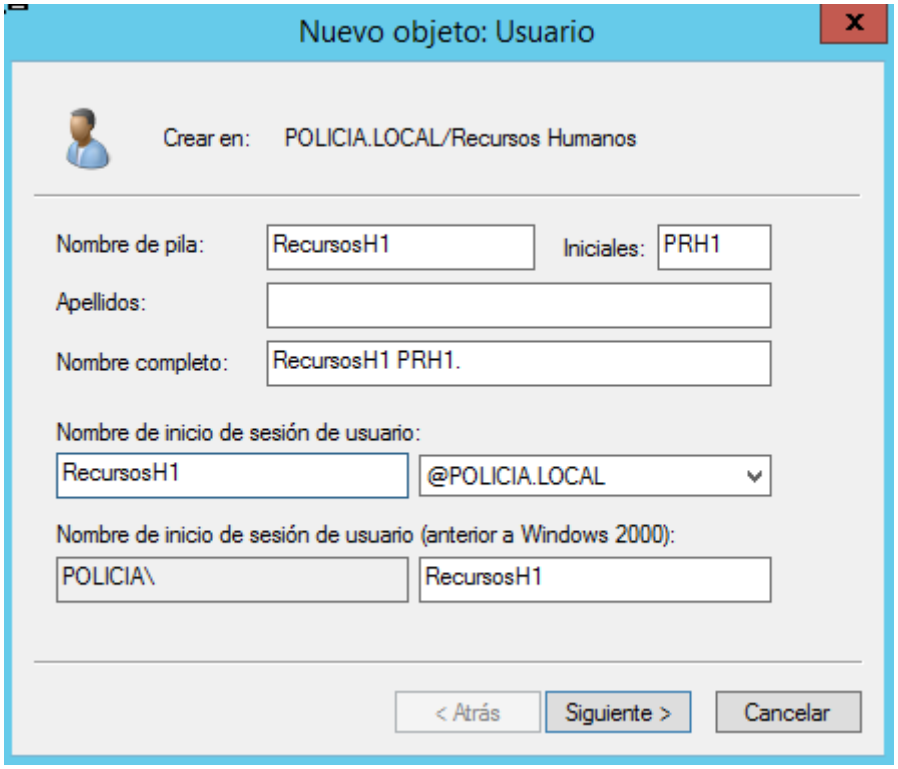


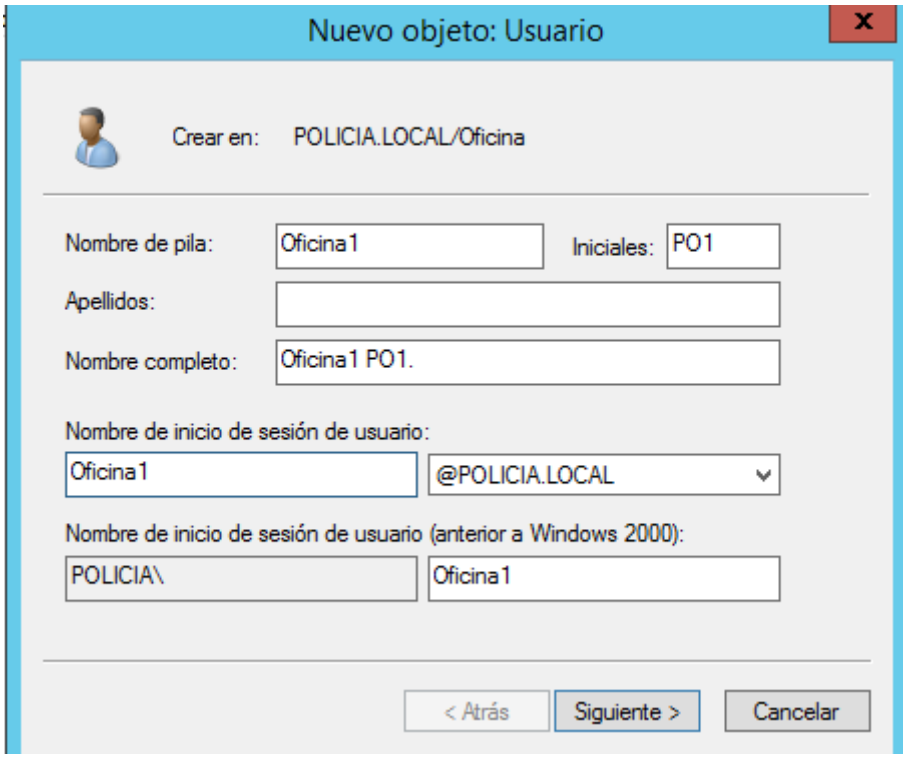
Continuaremos haciendo una carpeta para poner todos los departamentos que posteriormente deberemos utilizar para sus respectivos usuarios.



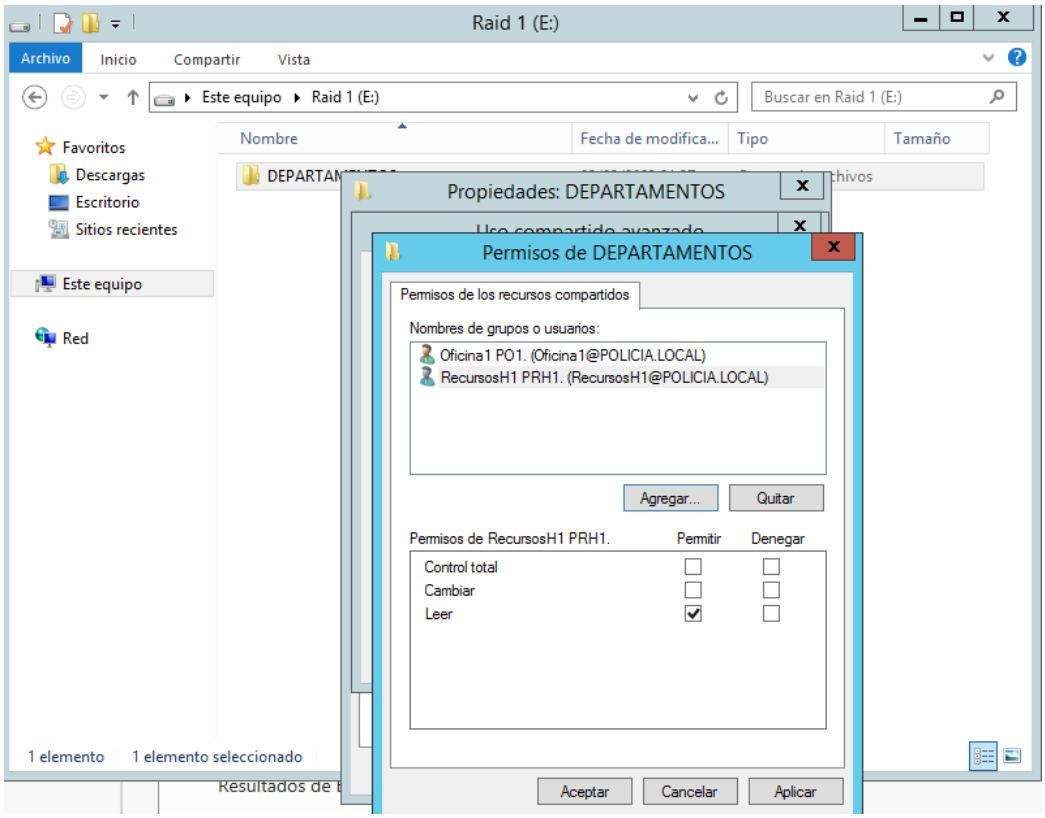


Procederemos a crear los usuarios que van a ser configurados posteriormente para entrar en sus respectivas carpetas.

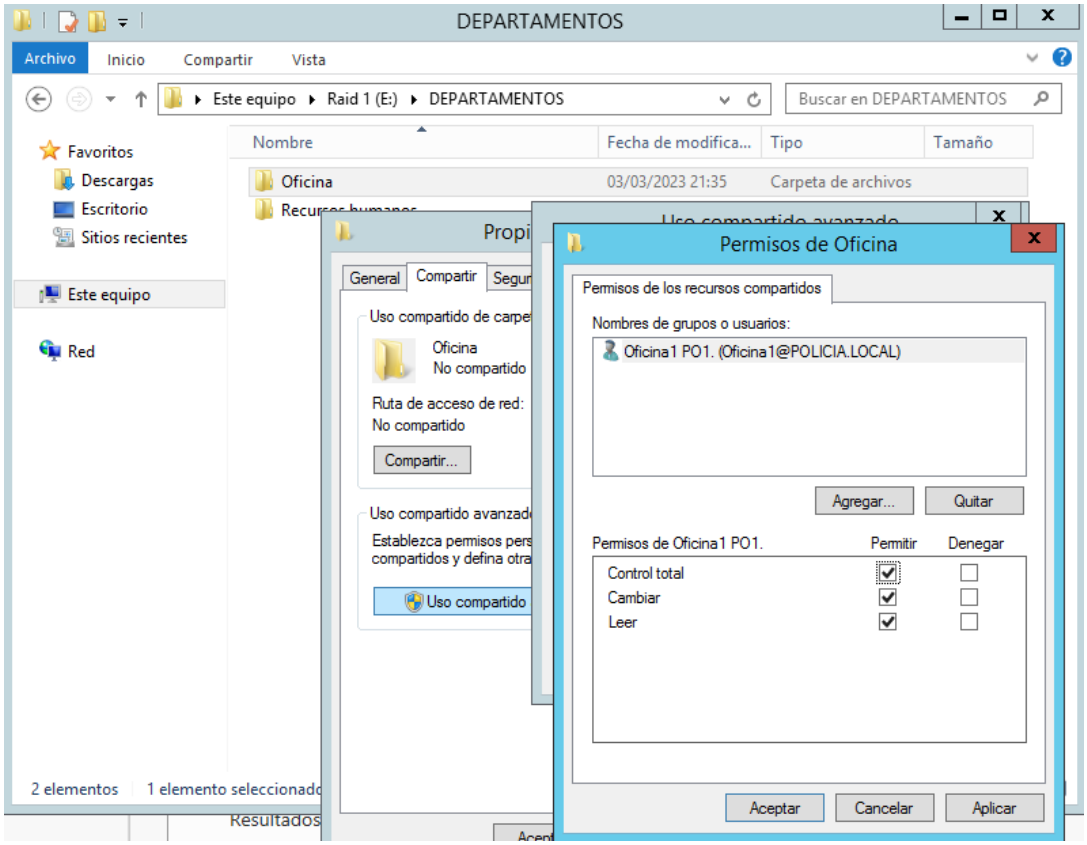




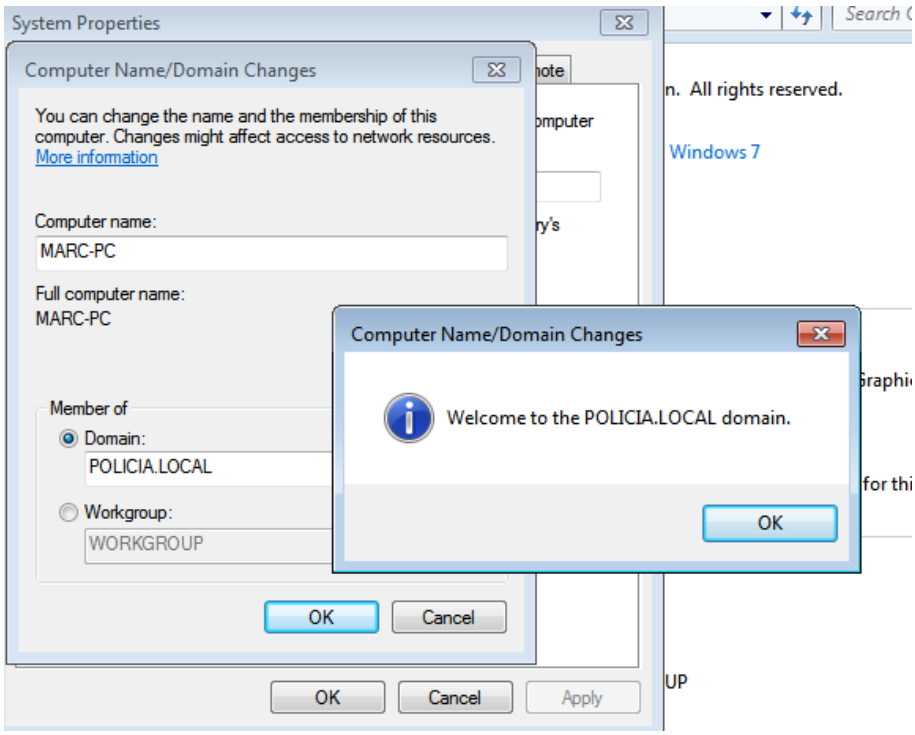
Seguiremos dando permisos a la carpeta compartida para que en departamentos puedan entrar los dos usuarios.

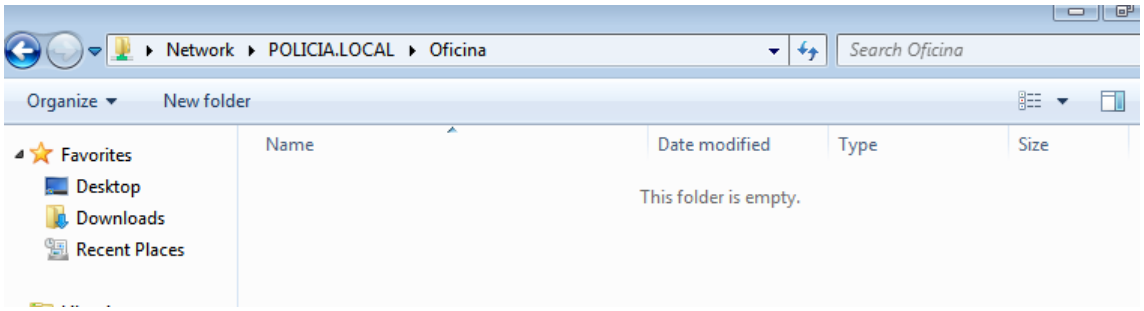


Y posteriormente configuraremos las 2 carpetas restantes para que pueda entrar solo el usuario correspondiente a cada carpeta.

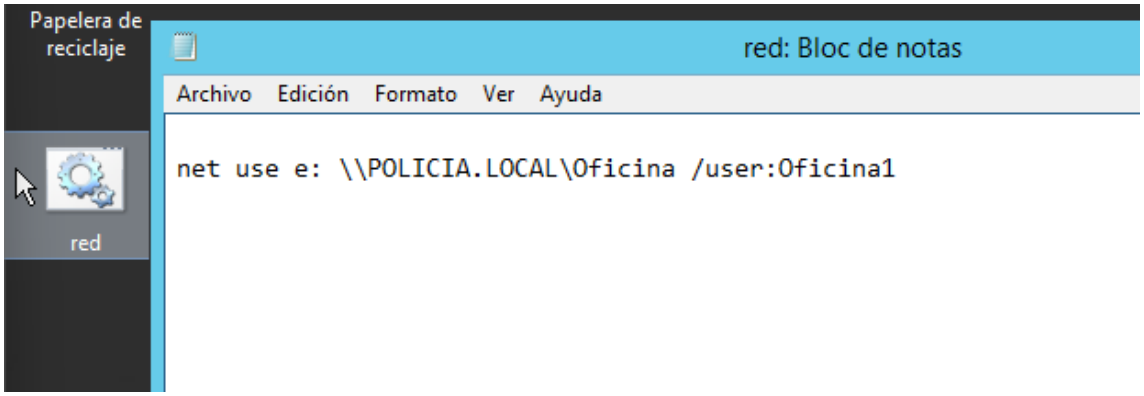


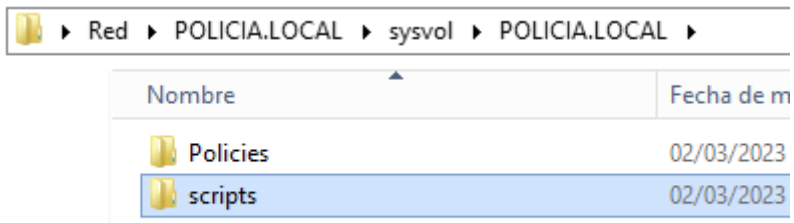
Abriremos los usuarios y los pondremos en el dominio anteriormente creado.

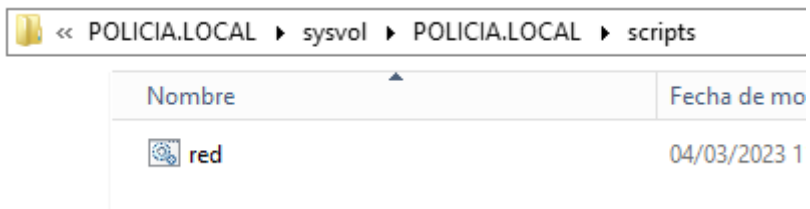


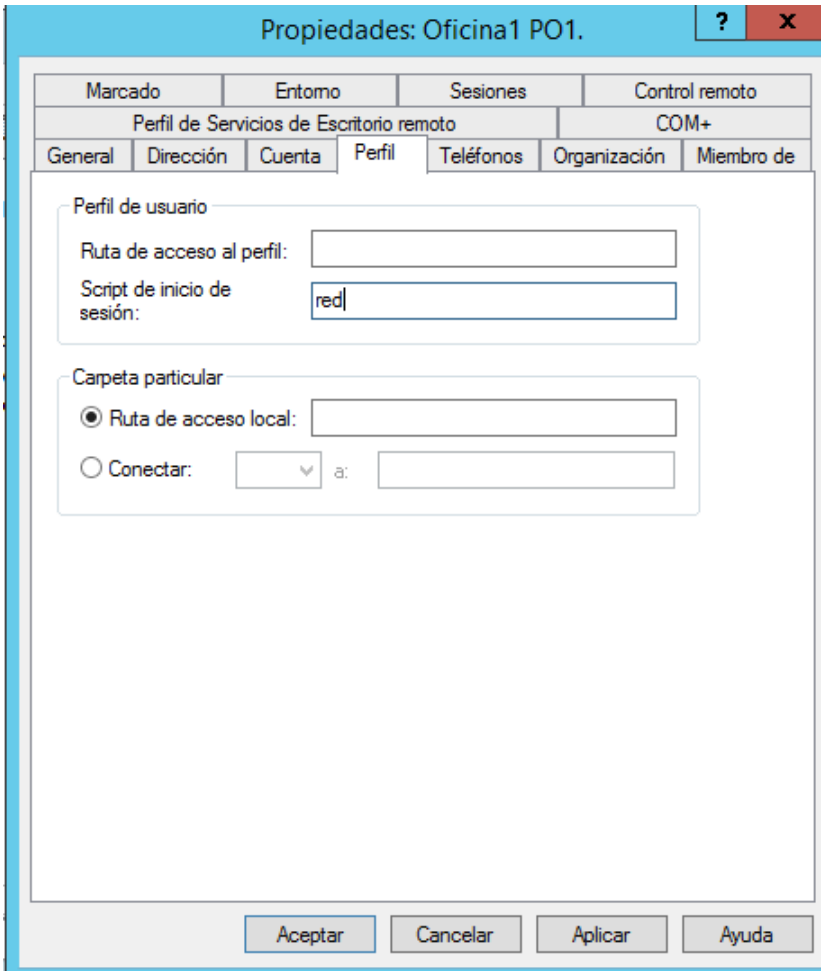


Crearemos el primer script para que una vez se inicie sesión con el usuario pueda ver su respectiva carpeta como una unidad de red en sus carpetas compartidas. Para eso crearemos el script, abriremos la carpeta sysvol dentro del servidor y la pondremos en el apartado script para que una vez configuremos el usuario pueda saber qué script iniciar al iniciar sesión.



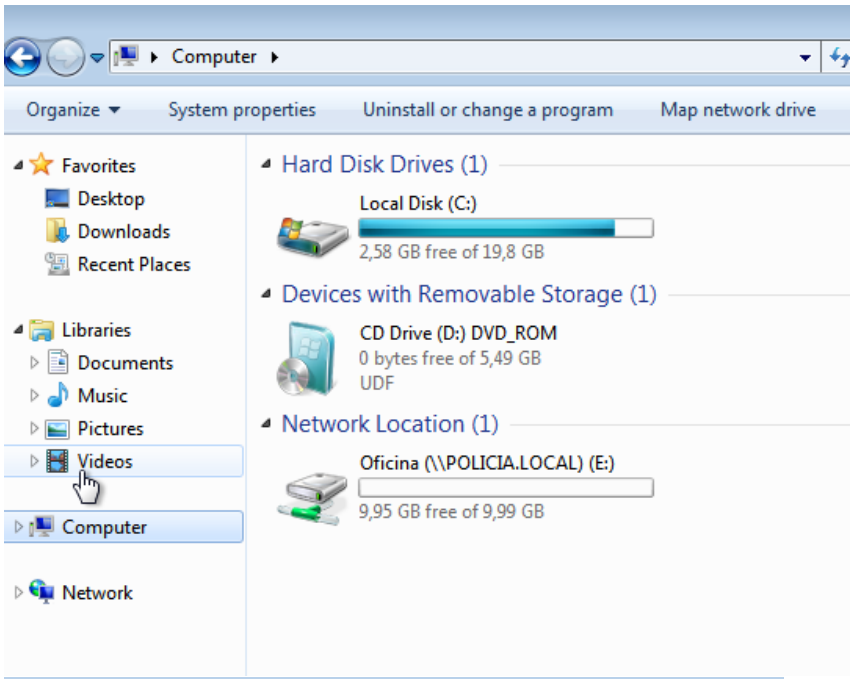


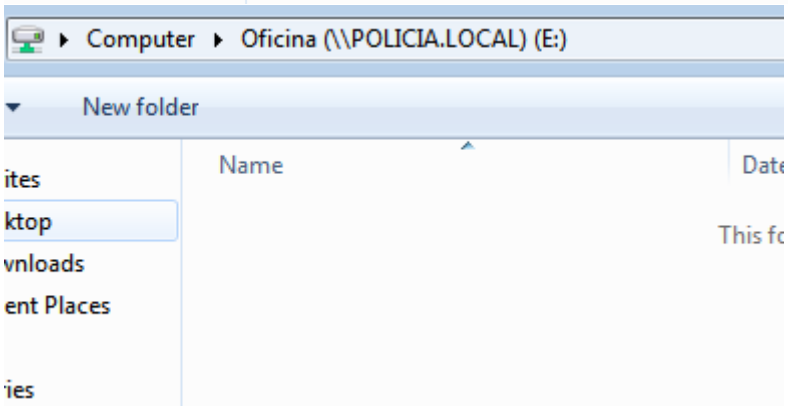




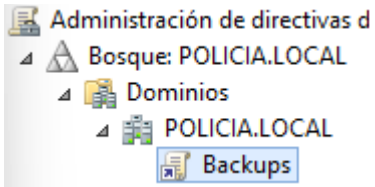
Ya solo queda iniciar sesión para verificar que funcione.

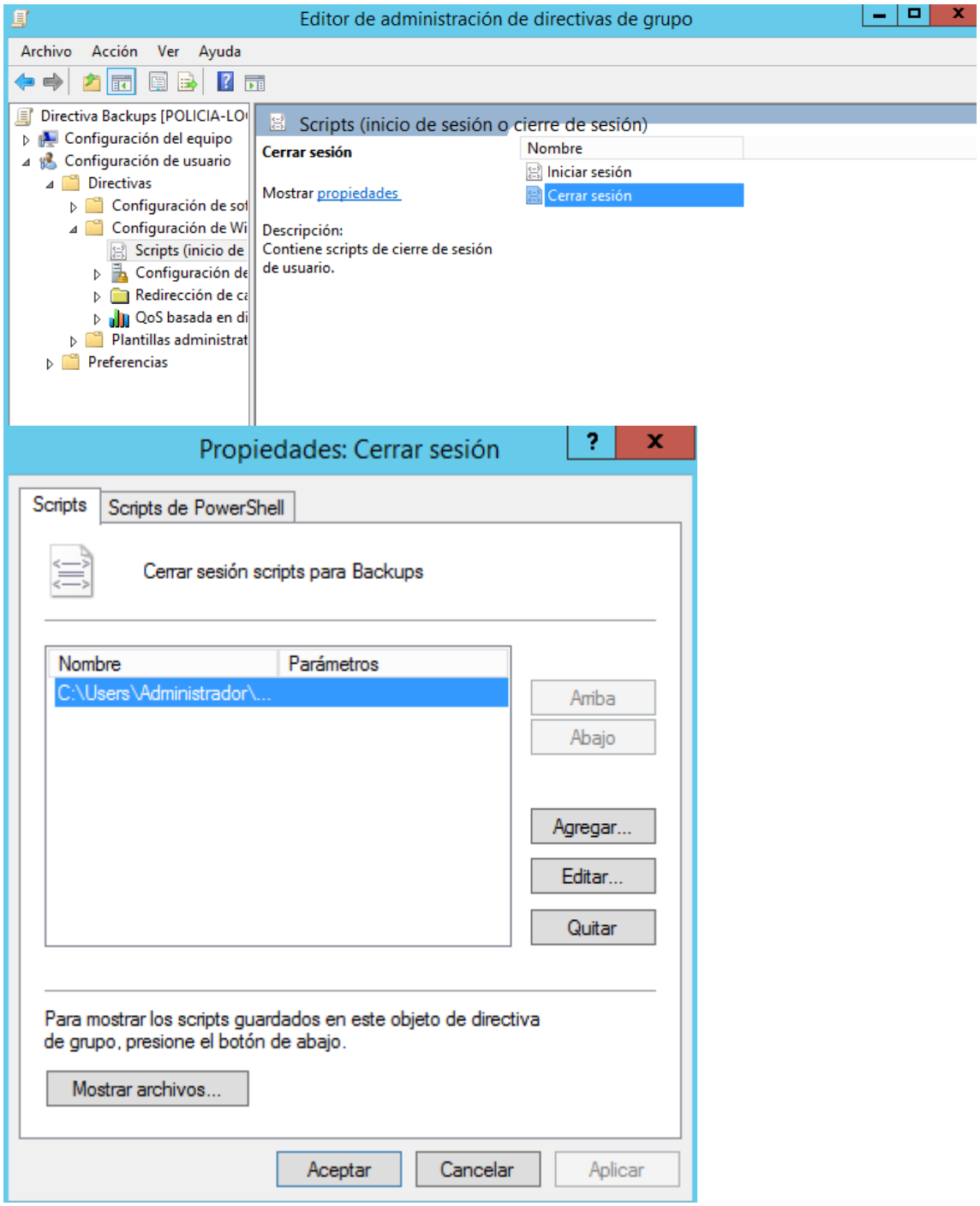
Podemos ver que funciona perfectamente.





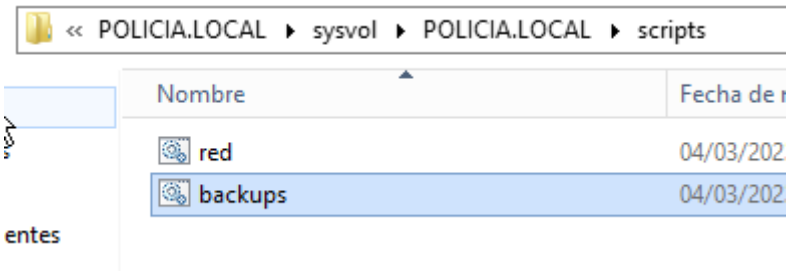
Para finalizar crearemos el script en backups, script que cree mal ya que este lo que hace es dar permisos para que copie la carpeta como debería ser, pero en vez de ser al cerrar la sesión es al iniciar sesión con el usuario.




Editor de administración de directivas de grupo
Archivo Acción Ver Ayuda
Directiva Backups [POLICIA-LO
Configuración del equipo
Configuración de usuario
Directivas
Configuración de sof
Configuración de Wi
Scripts (inicio de
Configuración de
Redirección de ca
QoS basada en di
Plantillas administrat
Preferencias
Scripts (inicio de sesión o cierre de sesión)
Cerrar sesión
Mostrar propiedades
Descripción:
Contiene scripts de cierre de sesión de usuario.
Nombre
Iniciar sesión
Cerrar sesión
Propiedades: Cerrar sesión
Scripts
Scripts de PowerShell
Cerrar sesión scripts para Backups
Nombre
Parámetros
C:\Users\Administrador\...
Arriba
Abajo
Agregar...
Editar...
Quitar
Para mostrar los scripts guardados en este objeto de directiva de grupo, presione el botón de abajo.
Mostrar archivos...
Aceptar
Cancelar
Aplicar

Aquí os dejo información extra que he utilizado para crear el siguiente script.

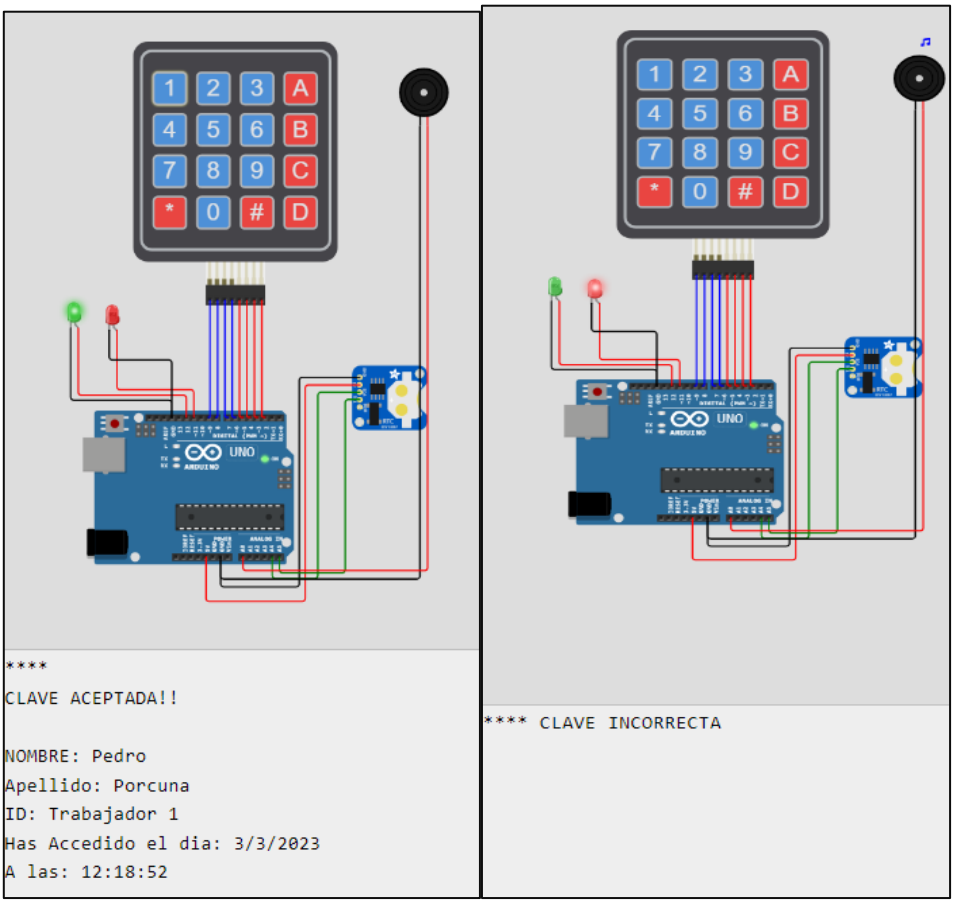
· /E: Copia las subcarpetas, incluidas las vacías.
· /Z: Copia los archivos en modo reiniciable.
· DCOPY:T: Copia las marcas de tiempo de la carpeta de origen a la carpeta de destino.
· /MT:8: Realiza la copia utilizando 8 hilos simultáneos.
· /R:10: Intenta copiar un archivo hasta 10 veces antes de fallar.
· /W:5: Espera 5 segundos entre cada intento de copia.
· /LOG+"..." Registra el resultado de la copia en un archivo de registro llamado %username% .log en la carpeta asignada. El símbolo "+" asegura que los registros se agreguen a un archivo existente en lugar de reemplazarlo.



Robótica

Código de Acceso con INFO y LEDS

Partiendo del circuito de la anterior versión, tenemos que implementar dos leds, un led de color rojo y el otro de color verde, también tendremos que implementarle un buzzer.
Cuando el trabajador sea aceptado al poner el código, se mostrará la información del trabajador y el led verde se encenderá durante dos segundos. En caso contrario, si el trabajador no es aceptado por el código de acceso, se encenderá un buzzer que sonará durante 2 segundos y se encenderá el led rojo.



En el caso de esta versión, hemos modificado estas líneas para que tanto los leds, como el buzzer, hagan la función de manera correcta:

En esta imagen podemos ver, que hemos modificado a cada usuario del código de acceso, donde dentro del IF, si pone la contraseña correcta que encienda el led verde. Y en la otra captura podemos ver que si es incorrecta que se encienda el diodo led rojo y suene el buzzer.

```
//CONTRASEÑA USUARIO 1/////////////////////////////////////////////////

if(i[0] == '1' && i[1] == '2' && i[2] == '1' && i[3] == '1')
{
  Serial.println();
  Serial.println("CLAVE ACEPTADA!! ");
  Serial.println();
  DateTime now = RTC.now(); // Mostramos la fecha actual
//Imprimimos el dia
Serial.println("NOMBRE: Pedro");
Serial.println("Apellido: Porcuna");
Serial.println("ID: Trabajador 1");
Serial.print("Has Accedido el dia: ");
Serial.print(now.day(), DEC);
Serial.print('/');
//Imprimimos el mes
Serial.print(now.month(), DEC);
Serial.print('/');
//Imprimimos el año
Serial.print(now.year(), DEC);
Serial.print(' ');
Serial.println();
//Imprimimos la hora
Serial.print("A las: ");
Serial.print(now.hour(), DEC);
Serial.print(':');
//Imprimimos los minutos
Serial.print(now.minute(), DEC);
Serial.print(':');
//Imprimimos los segundos
Serial.print(now.second(), DEC);
Serial.println();
//Comprobamos la hora y la enviamos por el puerto serie cada 3
segundos;
digitalWrite(ledv, HIGH);
delay(2000);
digitalWrite(ledv, LOW);
digitalWrite(ledr, LOW);
delay(1000);


  i[0]=0;
  i[1]=0;
  i[2]=0;
  i[3]=0;
}
```

Lineas modificadas

```
else{
  Serial.println(" CLAVE INCORRECTA");
  digitalWrite(ledr, HIGH);
  digitalWrite(ledv, LOW);
  tone(buzzerPin, 1000);
  delay(2000);
  noTone(buzzerPin);
  digitalWrite(ledr, LOW);


}
```

Aquí tenemos un LINK del WOKWI:
https://wokwi.com/projects/357826722976696321

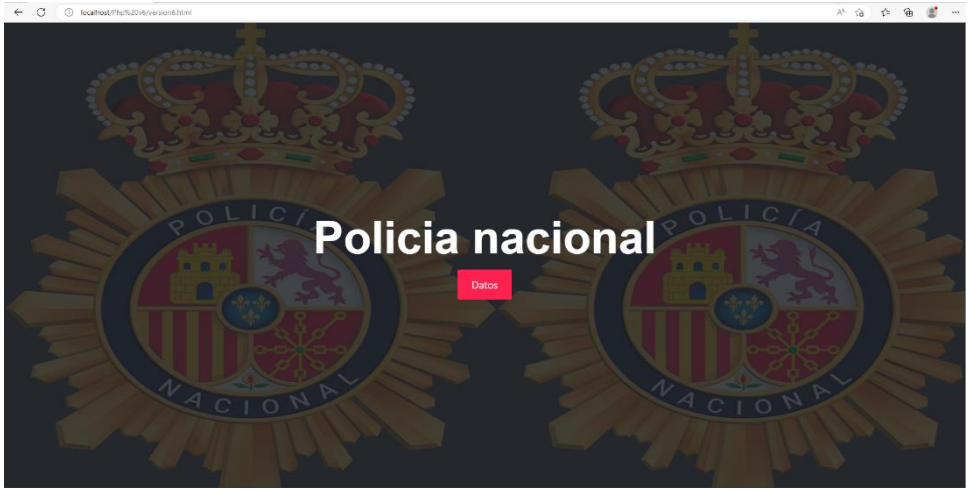
Y el video de del circuito en Físico:

https://drive.google.com/file/d/1kS4YdXgDhIl65cdo3CvDxHf-NT622w7b/view?usp=sharing

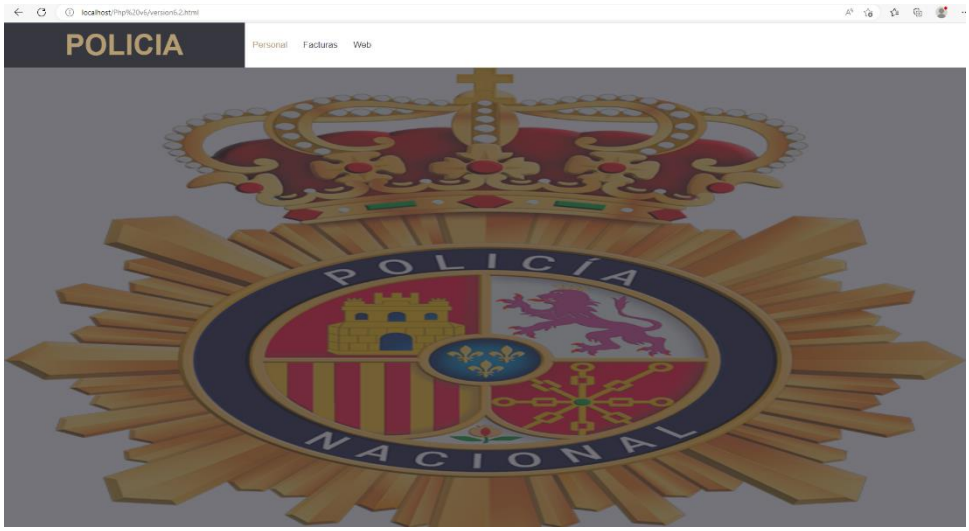
Versión 6

Redes

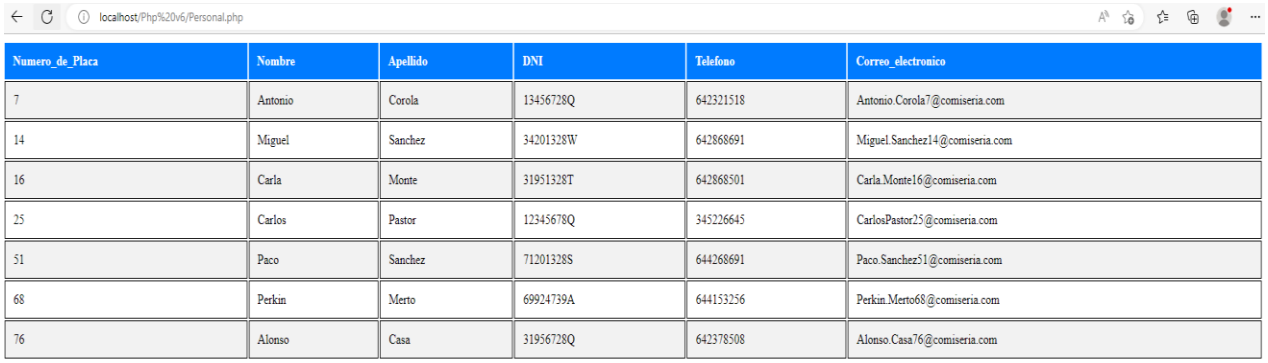
Crearemos un HTML con CSS como página principal



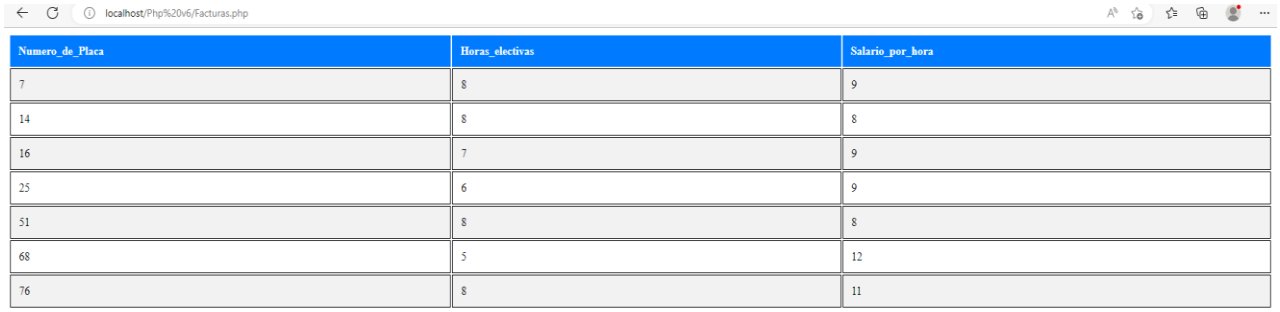
Desde la anterior página pasaremos a otra página HTML donde tendremos diferentes opciones para entrar a las varias páginas php vinculadas a la SQL



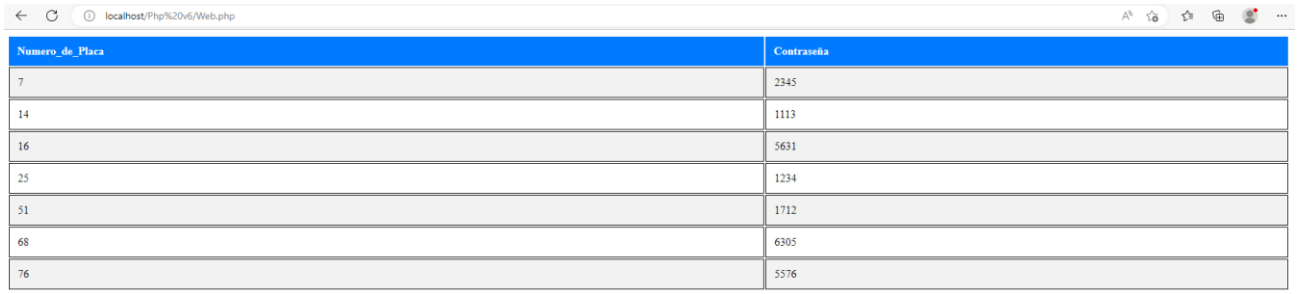
Al entrar a los diferentes enlaces tendremos toda la información que tenemos dentro de la BBDD en diferentes paginas php con sus respectivas tablas.



Numero_de_Placa	Nombre	Apellido	DNI	Telefono	Correo_electronico
7	Antonio	Corola	13456728Q	642321518	Antonio.Corola7@comiseria.com
14	Miguel	Sanchez	34201328W	642868691	Miguel.Sanchez14@comiseria.com
16	Carla	Monte	31951328T	642868501	Carla.Monte16@comiseria.com
25	Carlos	Pastor	12345678Q	345226645	CarlosPastor25@comiseria.com
51	Paco	Sanchez	71201328S	644268691	Paco.Sanchez51@comiseria.com
68	Perkin	Merto	69924739A	644153256	Perkin.Merto68@comiseria.com
76	Alonso	Casa	31956728Q	642378508	Alonso.Casa76@comiseria.com



Numero_de_Placa	Horas_electivas	Salario_por_hora
7	8	9
14	8	8
16	7	9
25	6	9
51	8	8
68	5	12
76	8	11



Numero_de_Placa	Contraseña
7	2345
14	1113
16	5631
25	1234
51	1712
68	6305
76	5576

En el código php nos conectamos a nuestra Base de Datos mediante; servername, username, password y dbname

Después indicaremos nuestra consulta Select en la línea 15

A continuación, mostraremos por pantalla los resultados de select con función echo.

Personal.php

```php
<?php
// Establece la conexión con la base de datos
$servername = "localhost";
$username = "root";
$password = "";
//$dbname = "comiseria";
$conexion = mysqli_connect($servername, $username, $password, `comiseria`);

// Comprueba si hay algún error en la conexión
if (!$conexion) {
    die('Error de conexión: ' . mysqli_connect_error());
}

// Ejecuta la consulta SELECT
$resultado = mysqli_query($conexion, "SELECT * FROM comiseria.Personal");

// Comprueba si hay algún error en la consulta
if (!$resultado) {
    die('Error en la consulta: ' . mysqli_error($conexion));
}

// Crea la tabla HTML
echo '<LINK REL="stylesheet" TYPE="text/css" HREF="Act 6.css">';
echo '<table>';
echo '<tr><th>Numero_de_Placa</th><th>Nombre</th><th>Apellido</th><th>DNI</th><th>Telefono</th><th>Correo_electronico</th></tr>';

// Itera a través de los resultados de la consulta SELECT y muestra los datos en la tabla
while ($fila = mysqli_fetch_assoc($resultado)) {
    echo '<tr>';
    echo '<td>' . $fila['Numero_de_Placa'] . '</td>';
    echo '<td>' . $fila['Nombre'] . '</td>';
    echo '<td>' . $fila['Apellido'] . '</td>';
    echo '<td>' . $fila['DNI'] . '</td>';
    echo '<td>' . $fila['Telefono'] . '</td>';
    echo '<td>' . $fila['Correo_electronico'] . '</td>';
    echo '</tr>';
}


echo '</table>';


// Cierra la conexión con la base de datos
mysqli_close($conexion);
?>
```

Con el CSS editaremos el estilo de nuestra página php donde se muestran nuestros datos.

```css
table {
  border-collapse: separate;
  border- spacing: 0;
  width: 100%;
   borde: 3px solid;
        color: black;


}

th, td {
  text-align: left;
  padding: 12px;
  border: 1px solid;
        color; black;
}

th {
  background-color: #007bff;
  color: #fff;
  border: 0px solid;
  border-color: Black;
}

tr:nth-child(even) {
  background-color: #f2f2f2;
}

                tr:hover {
                        background-color: #00d9ff;

                }
```

Sistemas

SISTEMAS OPERATIVOS: MAIL

Para empezar con la practica instalaremos el postfix y el dovecot en un solo comando que veremos a continuación.

```
marc@marc:~$ sudo apt-get install postfix dovecot-core dovecot-imapd dovecot-pop3d
```

Entraremos en la configuración del postfix e modificaremos las siguientes líneas de texto.

```
 GNU nano 6.2                                  /etc/postfix/main.cf
mydestination = localhost.$marc, localhost, $marc
```

Luego de hacer los cambios guardaremos el fichero y reiniciaremos el postfix para aplicar los cambios hechos.

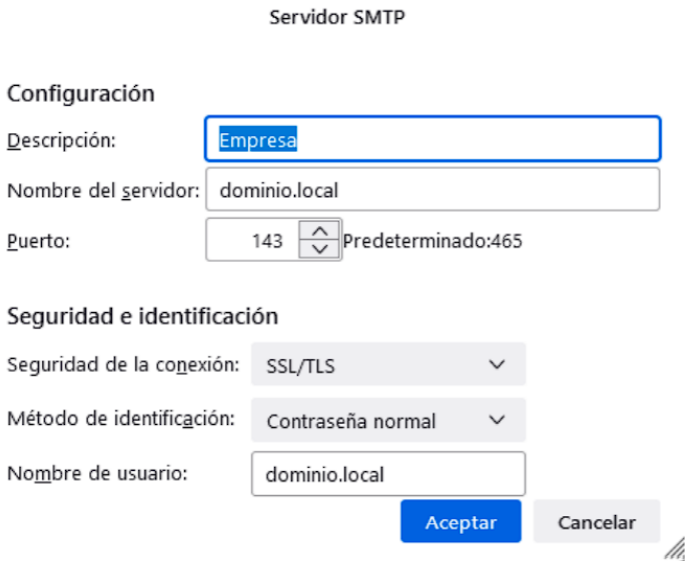
```
marc@marc:~$ sudo systemctl restart postfix
```

Comprobaremos que el postfix funcione correctamente.

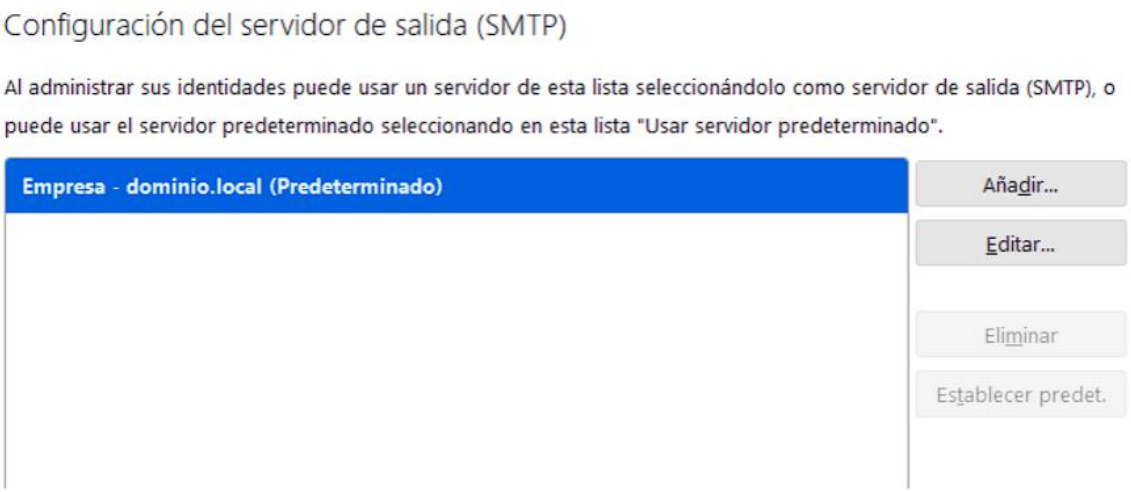
```
marc@marc:~$ sudo systemctl status postfix
● postfix.service - Postfix Mail Transport Agent
     Loaded: loaded (/lib/systemd/system/postfix.service; enabled; vendor preset: enabled)
     Active: active (exited) since Tue 2023-04-11 14:07:51 UTC; 12s ago
       Docs: man:postfix(1)
    Process: 5932 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
   Main PID: 5932 (code=exited, status=0/SUCCESS)
        CPU: 1ms

abr 11 14:07:51 marc systemd[1]: Starting Postfix Mail Transport Agent...
abr 11 14:07:51 marc systemd[1]: Finished Postfix Mail Transport Agent.
```

Luego de editar la configuración de postfix vamos a cambiar la del dovecot.

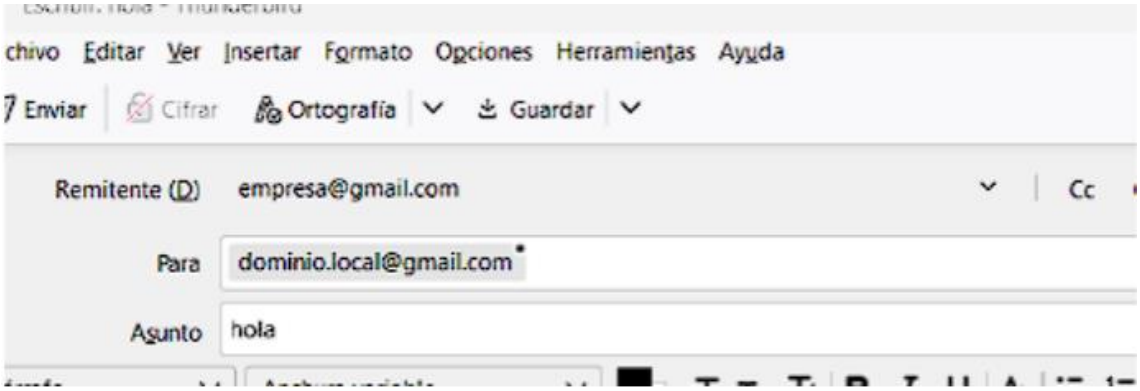
Luego de haber instalado el thunderbird añadiremos y configuraremos el servidor de salida SMTP.



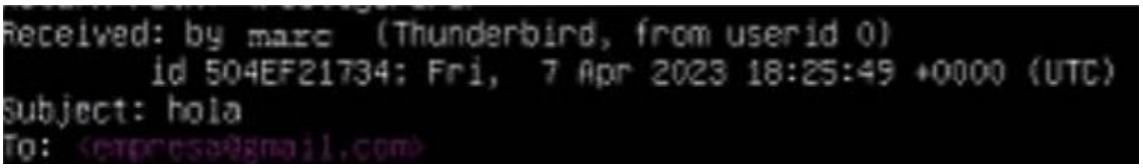
Y ponemos en predeterminado el servidor que acabamos de añadir.



Redactaremos un correo y posteriormente lo enviaremos.



Y para finalizar comprobaremos en el Ubuntu Server si se ha recibido el correo.

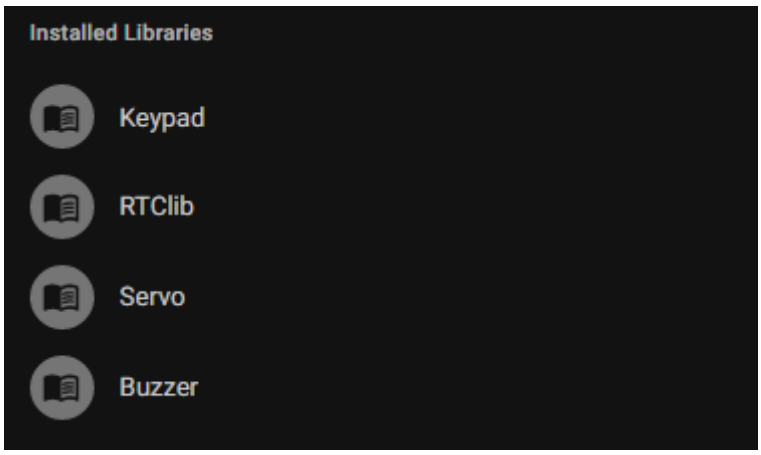


Robótica

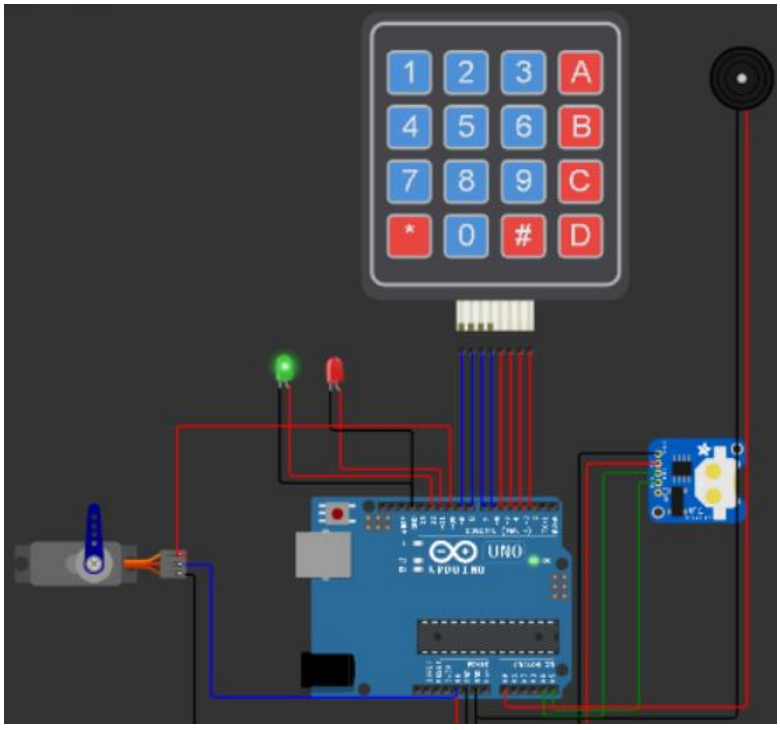
Código de Acceso con cerradura

Partiendo del circuito de la versión 5, le añadiremos un servomotor al circuito de manera que, al poner un código correcto, el servomotor hará de cerradura y se activará junto al led verde, y mostrará toda la información del trabajador, en caso de que el código sea incorrecto, el led rojo se activará y el buzzer sonará.

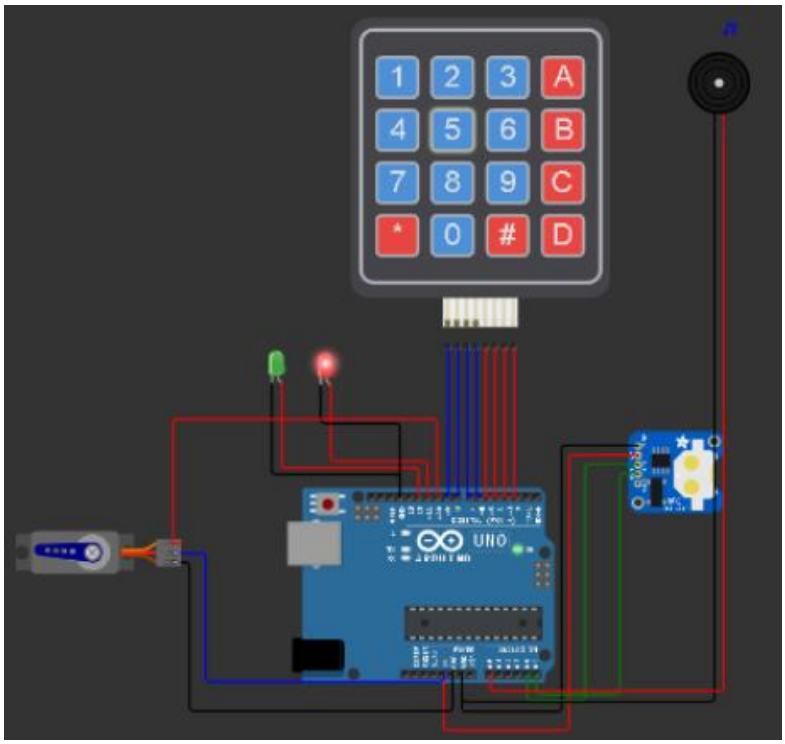
Librerías añadidas:



Contraseña correcta:



Contraseña Incorrecta:



En esta versión, hemos incluido la librería y hemos declarado el servomotor:

```
#include <Servo.h> //  incluimos la biblioteca Servo que proporciona funciones para controlar servomotores.
RTC_DS1307 RTC; // Definimos este objeto para interactura ocn el reloj en tiempo real
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"};

Servo servomotor; // Le nombramos al servo, el nombre de servomotor
```

En el void setup he iniciado el servomotor:

```
void setup(){
  Serial.begin(9600); //Iniciamos comunicación serie
  RTC.begin(); // Inisiamos el RTC para para interactuar con el reloj en tiempo real
  servomotor.attach(10); // conectamos el servo al pin 9


}
```

hemos modificado que, en cada usuario, al poner una contraseña correcta, accionemos el servomotor durante 5 segundos, en este caso he puesto un ejemplo del Usuario1:

```
//CONTRASEÑA USUARIO 1/////////////////////////////////////////////////

if(i[0] == '1' && i[1] == '2' && i[2] == '1' && i[3] == '1')
{
  Serial.println();
  Serial.println("CLAVE ACEPTADA!! ");
  Serial.println();
  DateTime now = RTC.now(); // Mostramos la fecha actual
//Imprimimos el dia
Serial.println("NOMBRE: Pedro");
Serial.println("Apellido: Porcuna");
Serial.println("ID: Trabajador 1");
Serial.print("Has Accedido el dia: ");
Serial.print(now.day(), DEC);
Serial.print('/');
//Imprimimos el mes
Serial.print(now.month(), DEC);
Serial.print('/');
//Imprimimos el año
Serial.print(now.year(), DEC);
Serial.print(' ');
Serial.println();
//Imprimimos la hora
Serial.print("A las: ");
Serial.print(now.hour(), DEC);
Serial.print(':');
//Imprimimos los minutos
Serial.print(now.minute(), DEC);
Serial.print(':');
//Imprimimos los segundos
Serial.print(now.second(), DEC);
Serial.println();



//Comprobamos la hora y la enviamos por el puerto serie cada 3
segundos;
digitalWrite(ledv, HIGH);
servomotor.write(180);
delay(2000);
digitalWrite(ledv, LOW);
digitalWrite(ledr, LOW);
servomotor.write(90);



  i[0]=0;
  i[1]=0;
  i[2]=0;
  i[3]=0;

}
```

Y en el else lo he dejado, así como en la Versión5:

```
else{
  Serial.println(" CLAVE INCORRECTA");
  digitalWrite(ledr, HIGH);
  digitalWrite(ledv, LOW);
  tone(buzzerPin, 1000);
  delay(2000);
  noTone(buzzerPin);
  digitalWrite(ledr, LOW);
```

Video de la versión:

https://drive.google.com/file/d/1Ti67xMHEZWqjWT5SZRVsH3nAiuufxvb4/view?usp=sharing

Versión 7

Redes

Agregamos las opciones altas y bajas

En la opción Altas tendremos un formulario para insertar datos en la BBDD

En la opción Bajas tendremos un formulario para eliminar datos en la BBDD

Nos conectamos a la bbdd

```php
$servername = "127.0.0.1";
$username = "root";
$password = "";
$dbname = "comisaria";
$conect = mysqli_connect($servername, $username, $password, $dbname);
```

Insertamos los datos que hemos introducido en el formulario

```php
//Recuperar datos del formulario
$Numero_de_Placa = $_POST['Numero_de_Placa'];
$Nombre = $_POST['Nombre'];
$Apellido = $_POST['Apellido'];
$DNI = $_POST['DNI'];
$Telefono = $_POST['Telefono'];
$Correo_electronico = $_POST['Correo_electronico'];

//Insertar datos en la tabla
$INSERT = "INSERT INTO Personal (Numero_de_Placa, Nombre, Apellido, DNI, Telefono, Correo_electronico) VALUES ('$Numero_de_Placa', '$Nombre', '$Apellido', '$DNI', '$Telefono', '$Correo_electr
```

En el mensaje de que nos dice que los datos han sido introducidos correctamente, clicamos para ir a la página que nos mostrara todos los datos de la bbdd

```php
if (mysqli_query($conect, $INSERT)) {
    echo '<a href= "http://localhost/Sintesis/Personal.php">Personal añadido correctamente.</a>';
} else {
    echo "Error: " . $INSERT . "<br>" . mysqli_error($conect);
}
```

Eliminamos los datos que hemos introducido en el formulario

```php
// Obtener el Numero_de_Placa del formulario
$Numero_de_Placa = $_POST['Numero_de_Placa'];

// Consulta SQL para eliminar el registro
$sql = "DELETE FROM Personal WHERE Numero_de_Placa = '$Numero_de_Placa'";
```

En el mensaje de que nos dice que los datos han sido eliminados correctamente, clicamos para ir a la página que nos mostrara todos los datos de la bbdd

```php
// Ejecutar la consulta y comprobar si se eliminó el registro
if ($conect->query($sql) === TRUE) {
    echo  '<a href= "http://localhost/Sintesis/Personal.php">El registro ha sido eliminado correctamente</a>';
} else {
    echo "Error al eliminar el registro: " . $conect->error;
}
```

Sistemas

Squid

Los usuarios de Internet usarán los servicios de un servidor proxy squid (linux server) que gestionan el acceso a internet y los sitios web prohibidos por la organización (Se debe demostrar el control).

Así que debe montarse un servidor linux server, con una sola tarjeta, no lo usaremos de gateway, que tenga la capacidad de dar acceso a un número reducido de usuarios a determinados lugares.

Por un lado se va a dar permiso a un número reducido de usuarios a la navegación en general, indicando que la autenticación que se hace va a ser válida solo por 5 horas.

Además queremos filtrar una lista de páginas para que no sean accesibles a través del proxy squid.

Para empezar, deberemos instalar el squid.

```
marc@marc:~$ sudo apt install squid_
```

Seguiremos con la configuración del squid. Lo primero que miraremos será el puerto para luego saber en que puerto se deben conectar los usuarios.

```
  GNU nano 6.2                                /etc/squid/squid.conf *
#                      easily accept requests destined to this port.
#
#       If you run Squid on a dual-homed machine with an internal
#       and an external interface we recommend you to specify the
#       internal address:port in http_port. This way Squid will only be
#       visible on the internal address.
#
#

# Squid normally listens to port 3128
http_port 3128
```

Seguiremos con el bloqueo de las paginas web no deseadas, aquí podemos ver dos ejemplos distintos para bloquear de dos formas distintas algunas páginas web, el primer ejemplo es bloqueando '.youtube.com' lo que hará que nuestro usuario no pueda entrar o acceder a esa página, lo haremos con las dos primeras líneas de comando, por otra parte, tendremos el bloqueo de todas aquellas paginas que contengan la palabra 'juegos', esto lo conseguiremos añadiendo las 2 líneas finales.

```
acl Videomarc dstdomain .facebook.com .youtube.com
http_access deny Videomarc
acl Juegosmarc url_regex -i juegos
http_access deny Juegosmarc
```

La próxima configuración será para que solo 10 usuarios se puedan conectar a nuestro servidor squid, eso lo conseguiremos haciendo que solo este permitido de la Ip 1.1 a la 1.10, y vamos a denegar el resto de las Ip's. Eso lo conseguiremos con las siguientes 3 lineas.

```
  GNU nano 6.2                                /etc/squid/squid.conf *
acl Policias src 192.168.1.1-192.168.1.10
http_access allow Policias
http_access deny all
```

Para finalizar haremos la autentificación de usuario, que solo se pueda acceder 5 horas antes de volver a autentificarse, y también añadiremos que sea capaz de hacer las 10 peticiones de autoricacion de los 10 usuarios a la vez.

Para la auntentificacion tendremos la primera línea de código, donde todos los usuarios se guardarán en la ruta '/etc/squid/passwd'.

Para que pueda hacer las 10 peticiones a la misma vez le añadiremos la segunda línea.

Y para finalizar, la última línea de código es para limitar el tiempo de acceso a 5 horas.

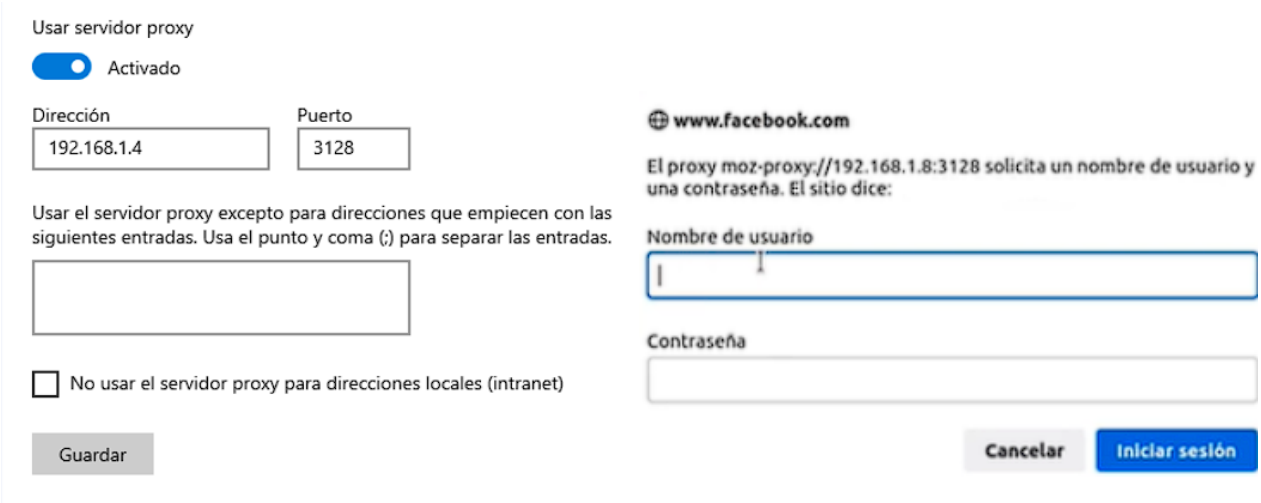
```
  GNU nano 6.2                          /etc/squid/squid.conf *
acl identificacion proxy_auth REQUIRED
auth_param basic progra /usr/lib/squid3/basic_ncsa_auth /etc/squid/passwd
auth_param basic Policias 5
auth_param basic credentialsttl 5 houts
```

Comprobación

Para realizar la comprobación necesitaremos una maquina cliente la cual pondremos en **'Red NAT'**, para que este dentro del rango de IP's que están permitidas.

```
C:\Users\Marc>ipconfig

Configuración IP de Windows


Adaptador de Ethernet Ethernet:

   Sufijo DNS específico para la conexión. . :
   Vínculo: dirección IPv6 local. . . : fe80::f848:35b8:e115:87d8%8
   Dirección IPv4. . . . . . . . . . . . : 192.168.1.8
   Máscara de subred . . . . . . . . . . : 255.255.255.0
   Puerta de enlace predeterminada . . . . . : 192.168.1.1
```

Seguiremos añadiendo el servidor proxy para que pueda aplicar la configuración en el usuario. Y aprovecharemos para ver si nos pide la identificación de usuarios.



Para comprobar que cliente no puede acceder a una de las paginas bloqueados debemos intentar acceder a ella.

Robótica

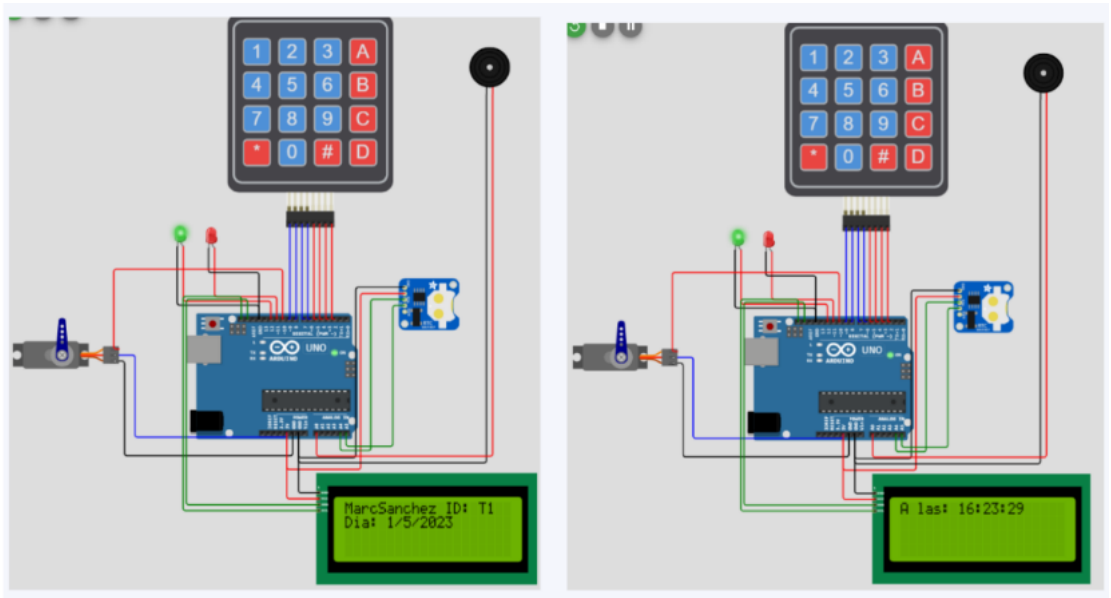
Información LCD:

Partiendo del circuito de la versión 6, añadiremos un LCD 16x2 donde toda la información que se genera al acceder al sistema de acceso se mostrará en el LCD.

Aquí podemos ver un ejemplo:

Aparte del accionamiento del LCD, cuando el código sea correcto, se encenderá el led verde, y se abrirá la cerradura del servomotor. En cambio, sí se pone un código incorrecto, el led rojo se encenderá, el servomotor no se abrirá y sonará el buzzer.

Aquí podemos ver el circuito completo:



CÓDIGO

Añadimos la librería LiquidCrystal2IC:

```
#include <Keypad.h> // Añadimos la libreria del teclado matricial
#include <LiquidCrystal_I2C.h>
const byte ROWS = 4; // Numero de filas
const byte COLS = 4; // Numero de columnas


#include "RTClib.h" // Esta libreria proporciona funciones para controlar un reloj de tiempo real (RTC) conectado a la placa.
#include <Servo.h> //  incluimos la biblioteca Servo que proporciona funciones para controlar servomotores.
RTC_DS1307 RTC; // Definimos este objeto para interactura ocn el reloj en tiempo real
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"}; // Definimos los dias
LiquidCrystal_I2C lcd(0x27,16,2);

Servo servomotor; // Le nombramos al servo, el nombre de servomotor

int segundos; // decalarmos los segundos
```

Iniciamoz y encendemos la pantalla del LCD:

```
void setup(){
  Serial.begin(9600); //Iniciamos comunicación serie
  RTC.begin(); // Inisiamos el RTC para para interactuar con el reloj en tiempo real
  servomotor.attach(10); // conectamos el servo al pin 9
  lcd.init(); // Inicializar la pantalla LCD
  lcd.backlight(); // Encender la retroiluminación de la LCD


}
```

Aquí podemos ver el ejemplo de un usuario, donde podemos ver que cada linea del código esta comentada:

```
//CONTRASEÑA USUARIO 1//////////////////////////////////////////////////

if(i[0] == '1' && i[1] == '2' && i[2] == '1' && i[3] == '1')
{
  Serial.println();
  Serial.println("CLAVE ACEPTADA!! "); //Mostramos por monitor serie ese mensaje
  Serial.println();
  // Mostramos la fecha actual
//Imprimimos el dia



lcd.setCursor(0,0); // Posicionamos el cursor en la primera linea

lcd.print("Marc"); // Imprimimos en el LCD el nombre
lcd.print("Sanchez"); // Imprimimos en el LCD el apellido
digitalWrite(ledv, HIGH); // Encendemos el led verde
servomotor.write(180); // Se mueve el servomotor
lcd.print(" ");
lcd.print("ID: T1"); // Imprimimos en el LCD el ID

lcd.setCursor(0,1); // Posicionamos el cursor en la segunda linea

DateTime now = RTC.now(); //Asignamos la fecha actual
lcd.print("Dia: "); // Imprimimos en el LCD Dia
lcd.print(now.day(), DEC); // Imprimimos en el LCD el dia actual
lcd.print('/');
//Imprimimos el mes
lcd.print(now.month(), DEC); // Imprimimos en el LCD el mes actual
lcd.print('/');
//Imprimimos el año
lcd.print(now.year(), DEC); // Imprimimos en el LCD el año actual
lcd.print(' ');
delay(2000); // Ponemos un retraso de 2 segundos
lcd.clear(); // Limpiamos el LCD


lcd.setCursor(0,0); // Posicionamos el cursor en la primera linea

//Imprimimos la hora
lcd.print("A las: "); // Imprimimos en el LCD a las:
lcd.print(now.hour(), DEC); // Imprimimos en el LCD la hora actual
lcd.print(':');
//Imprimimos los minutos
lcd.print(now.minute(), DEC); // Imprimimos en el LCD el minuto actual
lcd.print(':');
//Imprimimos los segundos
lcd.print(now.second(), DEC); // Imprimimos en el LCD el segundos actual
lcd.println();
delay(2000); // Ponemos un retraso de 2 segundos
digitalWrite(ledv, LOW); // Apagamos el led verde
digitalWrite(ledr, LOW); // Apagamos el led rojo
servomotor.write(90); //movemos el servomotor
lcd.clear(); // Limpiamos el LCD


segundos;
```

Video:

https://drive.google.com/file/d/1YQREVZnIhVHoAbOIhncJmJ0ccxhInTPu/view?usp=sharing

WOKWI:

https://wokwi.com/projects/357826722976696321

Versión 8

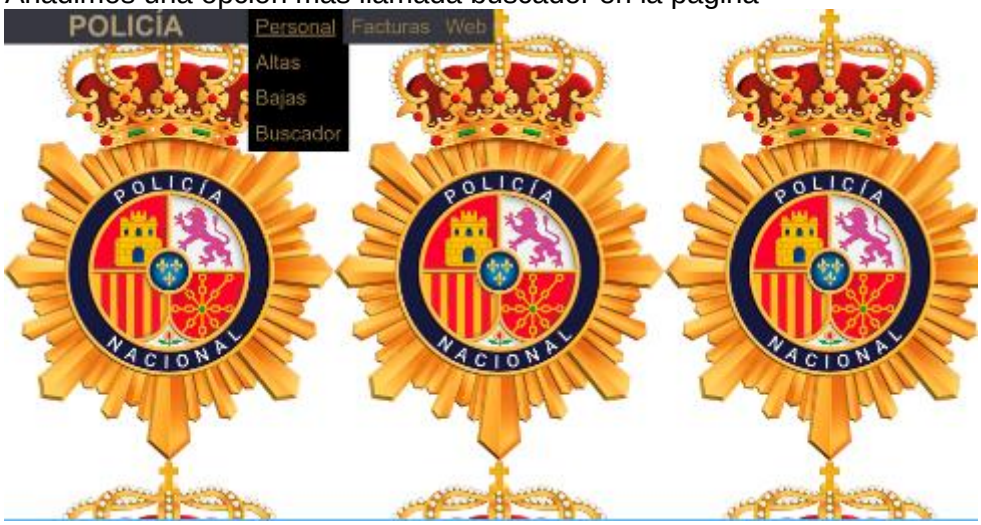
Redes

La página de login

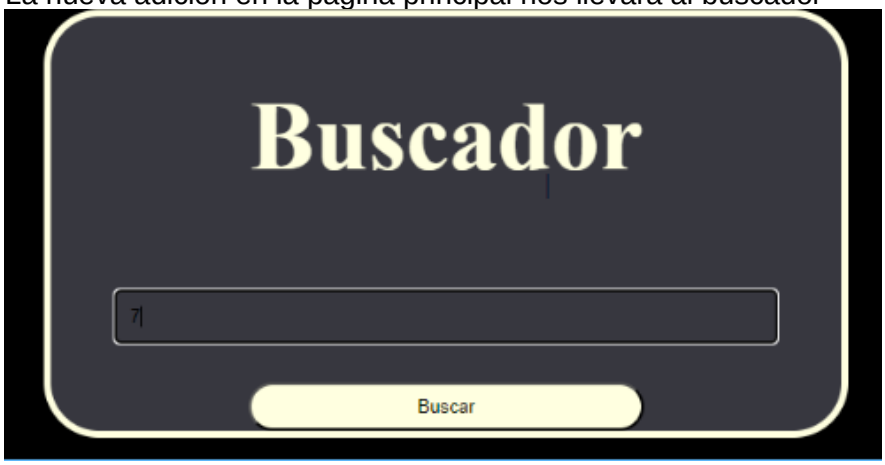
Vinculamos los campos username y password creados en el html para después desde la página php hacemos un select preguntando si el usuario puesto en el html existe en la BBDD en caso que exista nos llevará a la página que hemos creado en caso contrario no dará un mensaje de que el usuario no existe

```php
<?php
// Obtener los valores enviados desde el formulario
$Numero_de_Placa = $_POST['username'];
$Contraseña = $_POST['password'];

// Realizar la conexión a la base de datos (asumiendo MySQL en este ejemplo)
$servername = "localhost";
$db_username = "root";
$db_password = "";
$db_name = "comisaria";

$conn = new mysqli($servername, $db_username, $db_password, $db_name);

// Verificar si la conexión fue exitosa
if ($conn->connect_error) {
    die("Error en la conexión a la base de datos: " . $conn->connect_error);
}

// Consulta para verificar si el usuario existe en la base de datos
$sql = "SELECT * FROM Web WHERE Numero_de_Placa='$Numero_de_Placa' AND Contraseña='$Contraseña'";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // El usuario existe, se ha iniciado sesión correctamente
    header ("location: ../version6.2.html");
} else {
    // El usuario no existe o las credenciales son incorrectas
    echo "Nombre de usuario o contraseña incorrectos.";
}

$conn->close();
?>
```

Añadimos una opción más llamada buscador en la página



La nueva adición en la página principal nos llevará al buscador



Creamos una variable vinculada al html donde haremos un select hacia la BBDD y nos dará la información que buscamos

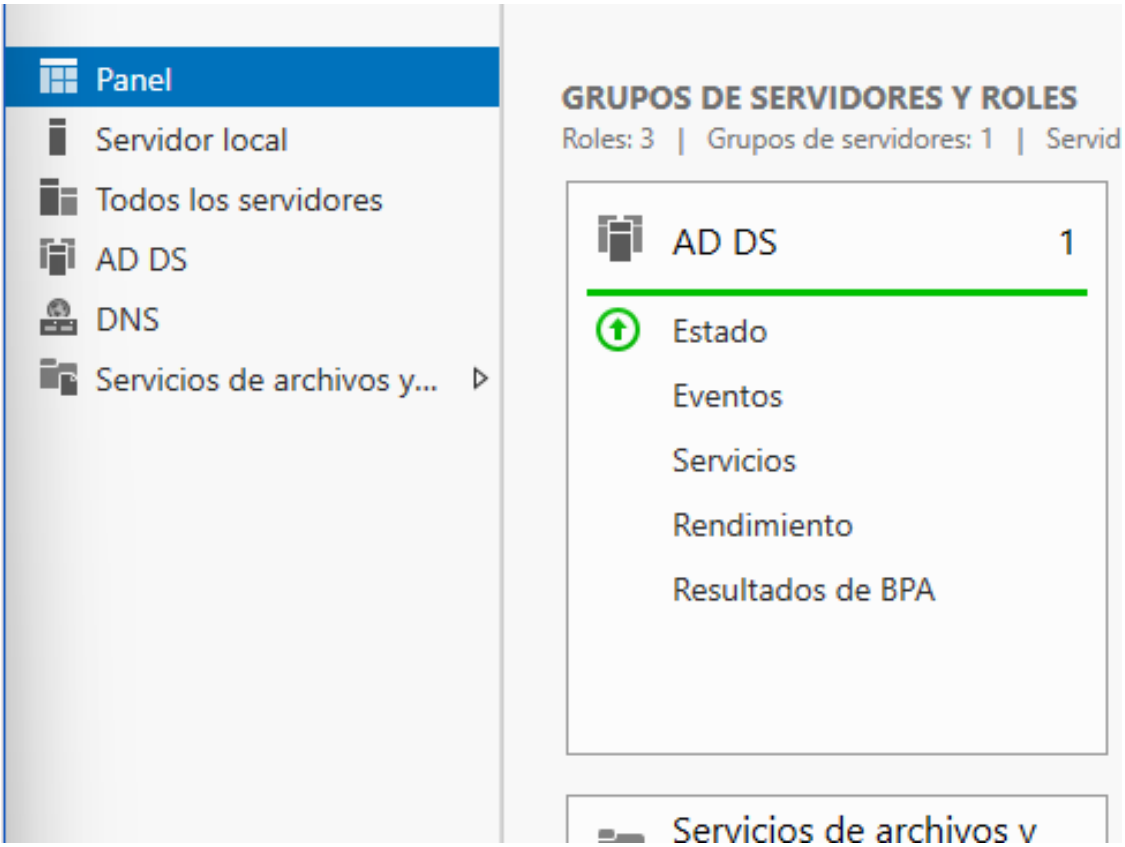
```php
$Numero_de_Placa = $_POST['Numero_de_Placa'];
// Ejecuta la consulta SELECT

$SELECT = mysqli_query($cnx, "SELECT * FROM comisaria.Personal  WHERE Numero_de_Placa = $Numero_de_Placa");
// Comprueba si hay algún error en la consulta
if (!$SELECT) {
  die('Error en la consulta: ' . mysqli_error($cnx));
}

// Crea la tabla HTML
echo '<LINK REL="stylesheet" TYPE="text/css" HREF="../Act 6.css">';
echo '<table>';
echo '<tr><th>Numero_de_Placa</th><th>Nombre</th><th>Apellido</th><th>DNI</th><th>Telefono</th><th>Correo_electronico</th></tr>';

// Itera a través de los resultados de la consulta SELECT y muestra los datos en la tabla
while ($fila = mysqli_fetch_assoc($SELECT)) {

  echo '<tr>';
  echo '<td>' . $fila['Numero_de_Placa'] . '</td>';
  echo '<td>' . $fila['Nombre'] . '</td>';
  echo '<td>' . $fila['Apellido'] . '</td>';
  echo '<td>' . $fila['DNI'] . '</td>';
  echo '<td>' . $fila['Telefono'] . '</td>';
  echo '<td>' . $fila['Correo_electronico'] . '</td>';
  echo '</tr>';
}
```

Sistemas

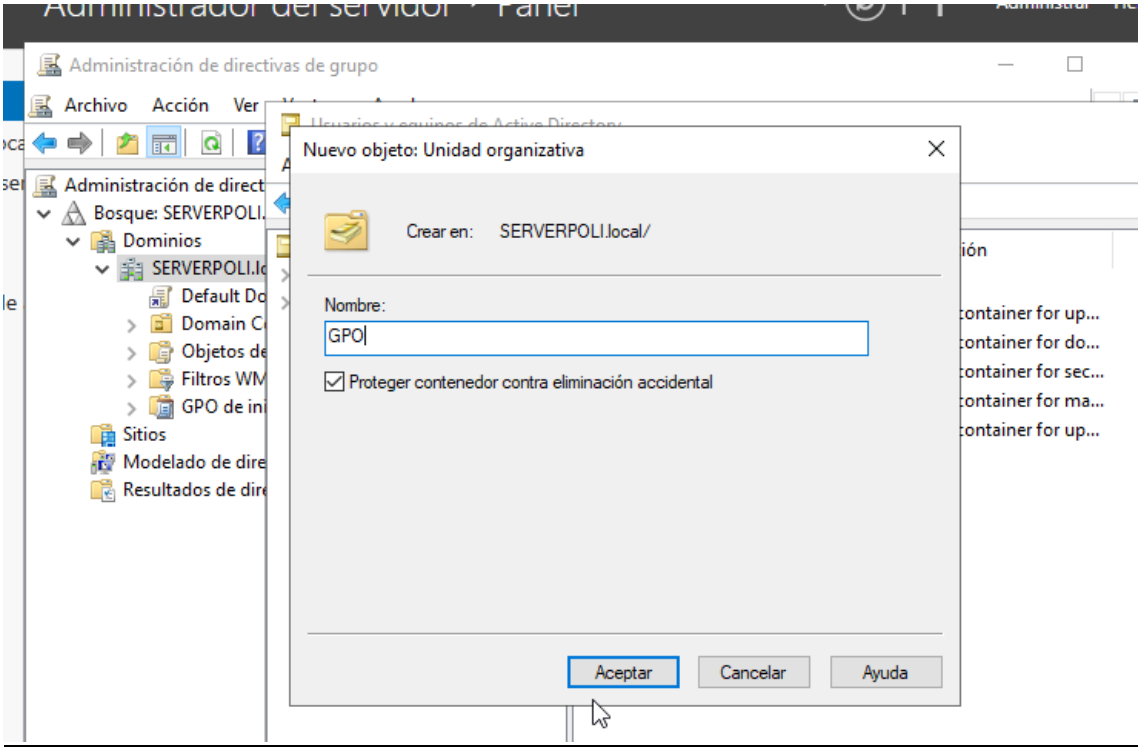
PROXY + GPO

Para empezar, vamos a necesitar 3 máquinas, un Ubuntu Server con PROXY, un Windows Server con AD donde configuraremos las GPO, y una maquina cliente que este en el dominio de la maquina Windows Server.

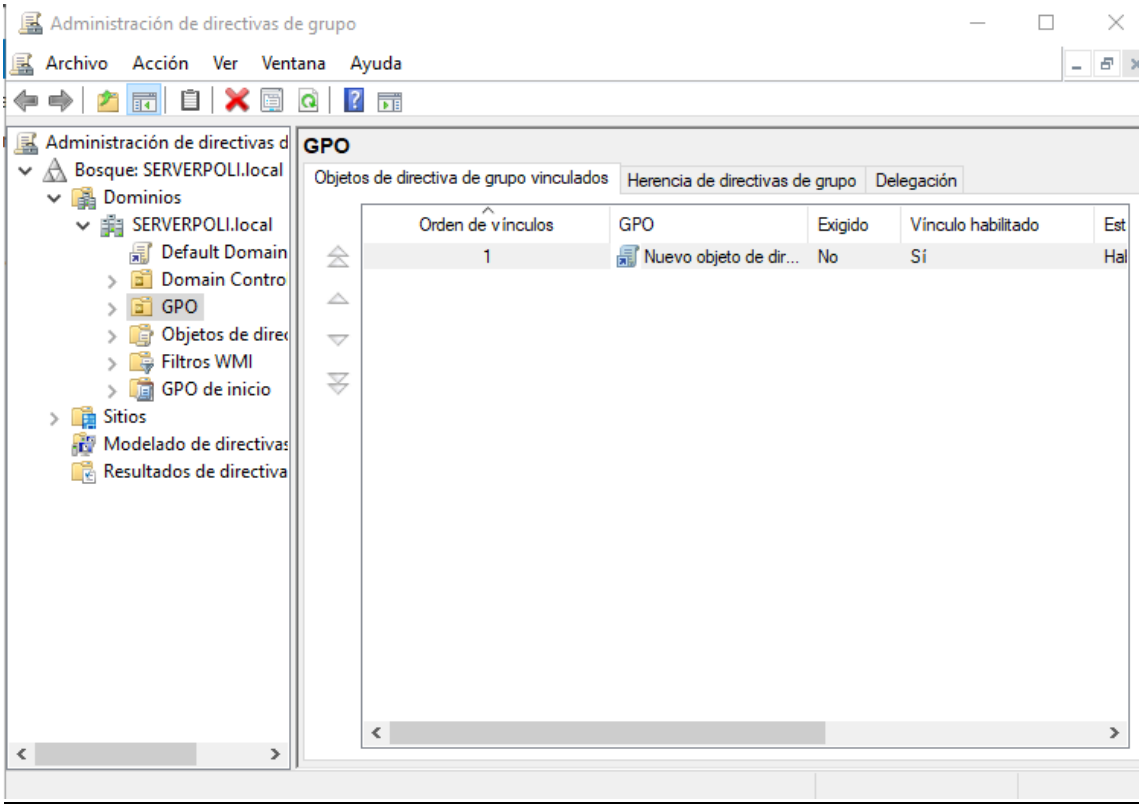
Empezaremos con la Windows Server ya que la Ubuntu Server ya la tenemos hecha de la versión anterior. Para empezar, instalamos el AD.



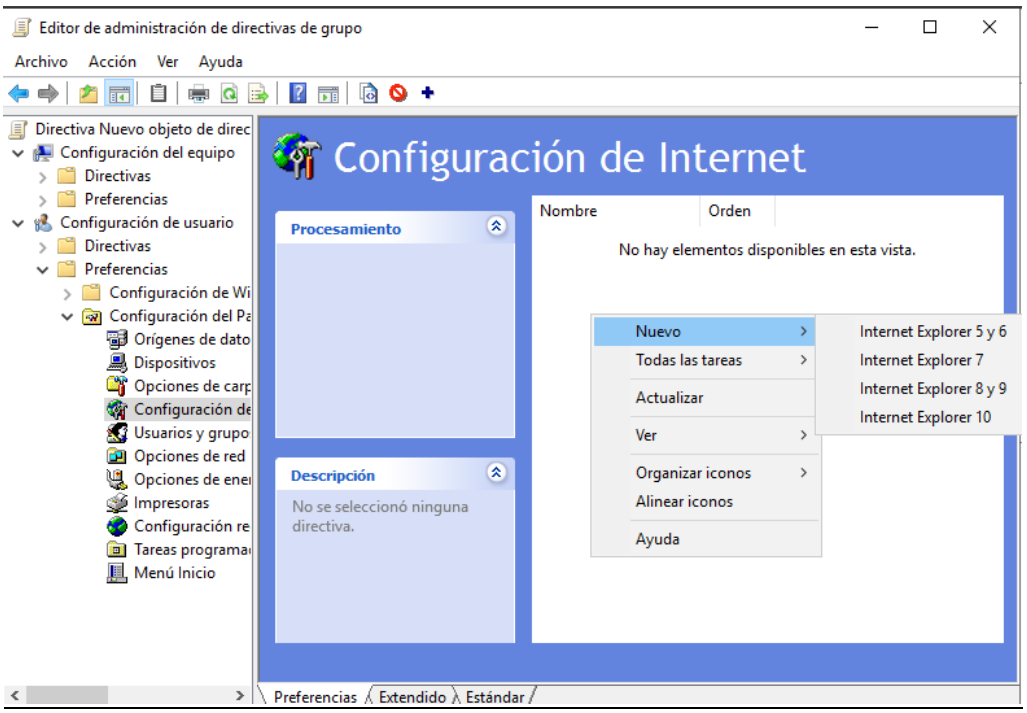
Una vez configurado el AD, continuaremos entrando al administrador de directivas de grupo, y crearemos una nueva unidad organizativa a la cual llamaremos GPO.



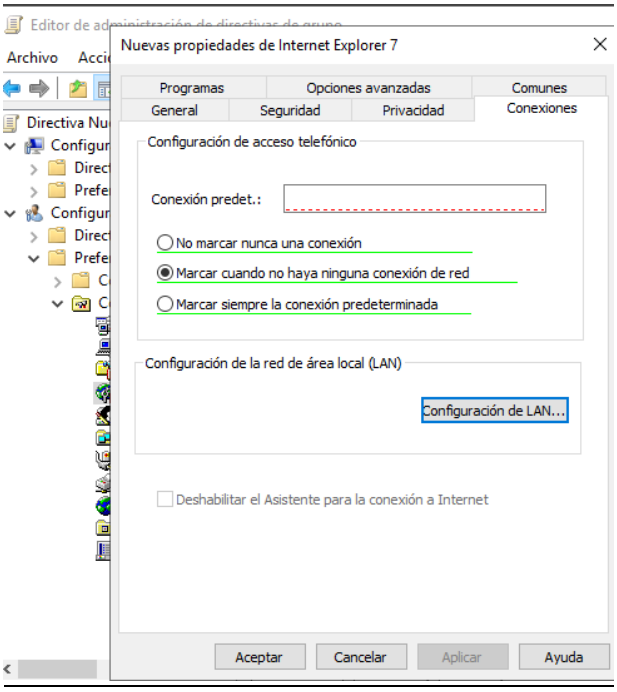
Una vez creado, crearemos dentro una GPO para configurarla posteriormente.



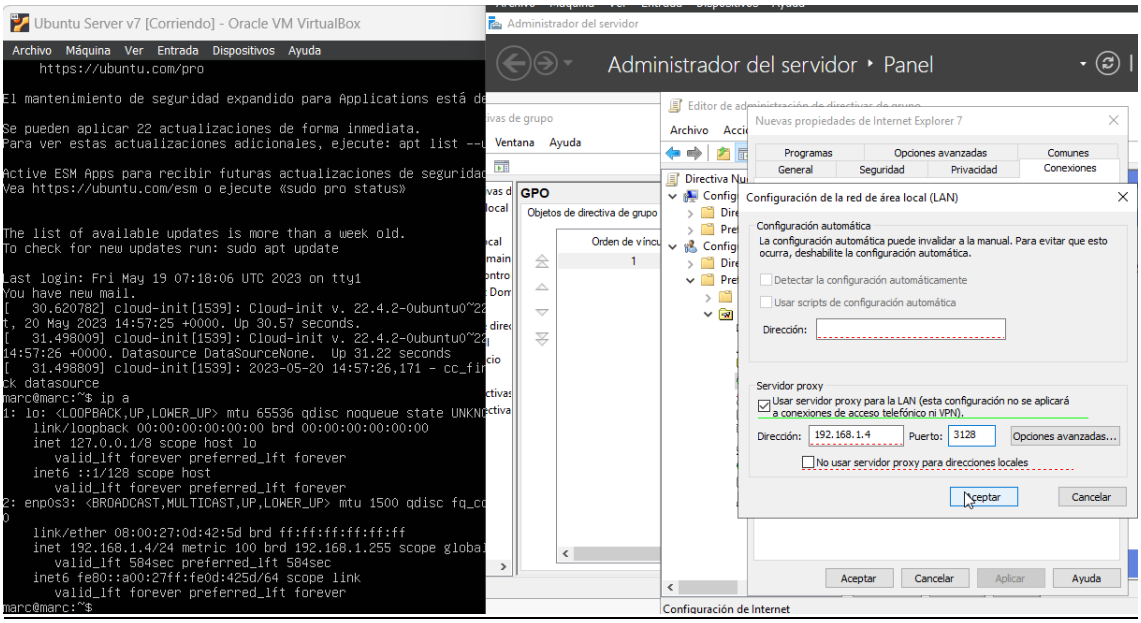
Una vez creada, entraremos para editarla. Dentro de Configuración de Usuarios, en el apartado Preferencias, en Configuración del pa..., Iremos a Configuración de internet, y añadiremos un Internet Explorer 8 y 9.



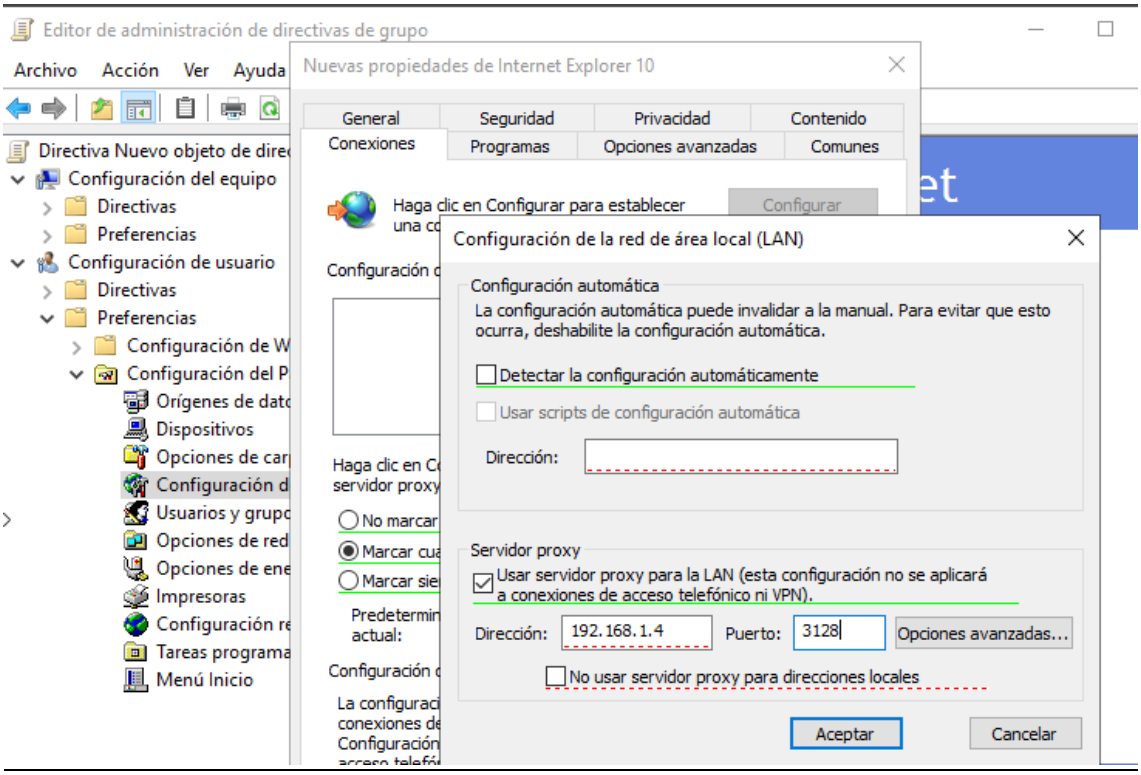
Buscaremos el apartado de conexiones, en el apartado Configuración de LAN.



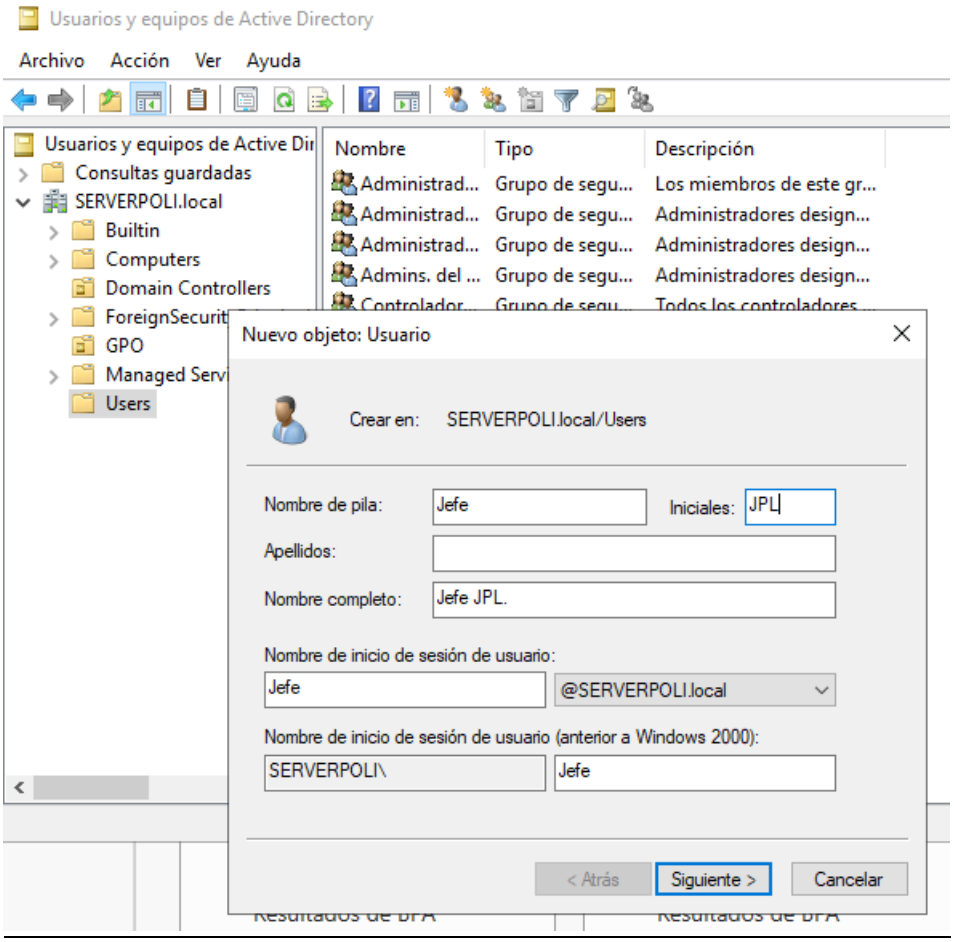
Deberemos abrir la maquina Ubuntu Server y mirar que IP teníamos (No hace falta decir que las 3 maquinas van a tener que estar en la misma red). I en opciones avanzadas, dentro de PROXY, añadiremos la IP de la maquina Ubuntu Server, y el puerto 3128, que es el que va predeterminado.

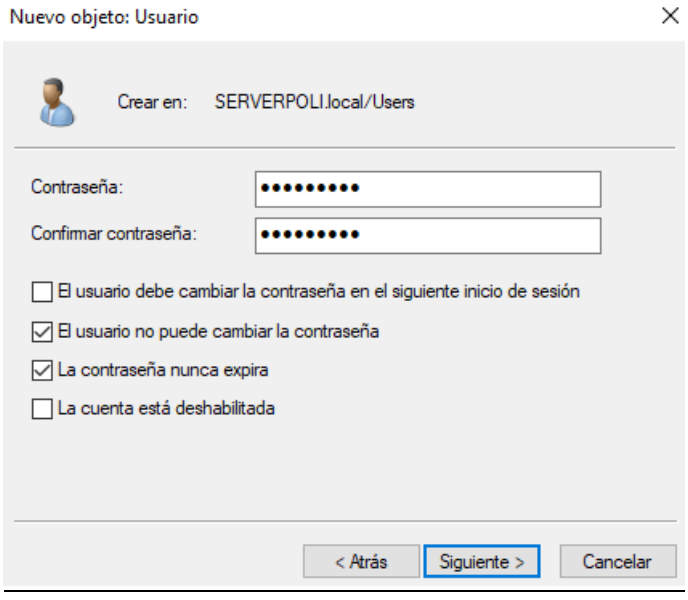


Repetiremos el mismo proceso con Internet Explorer 10, ya que no sabemos con exactitud que versión van a tener los usuarios de nuestra empresa.

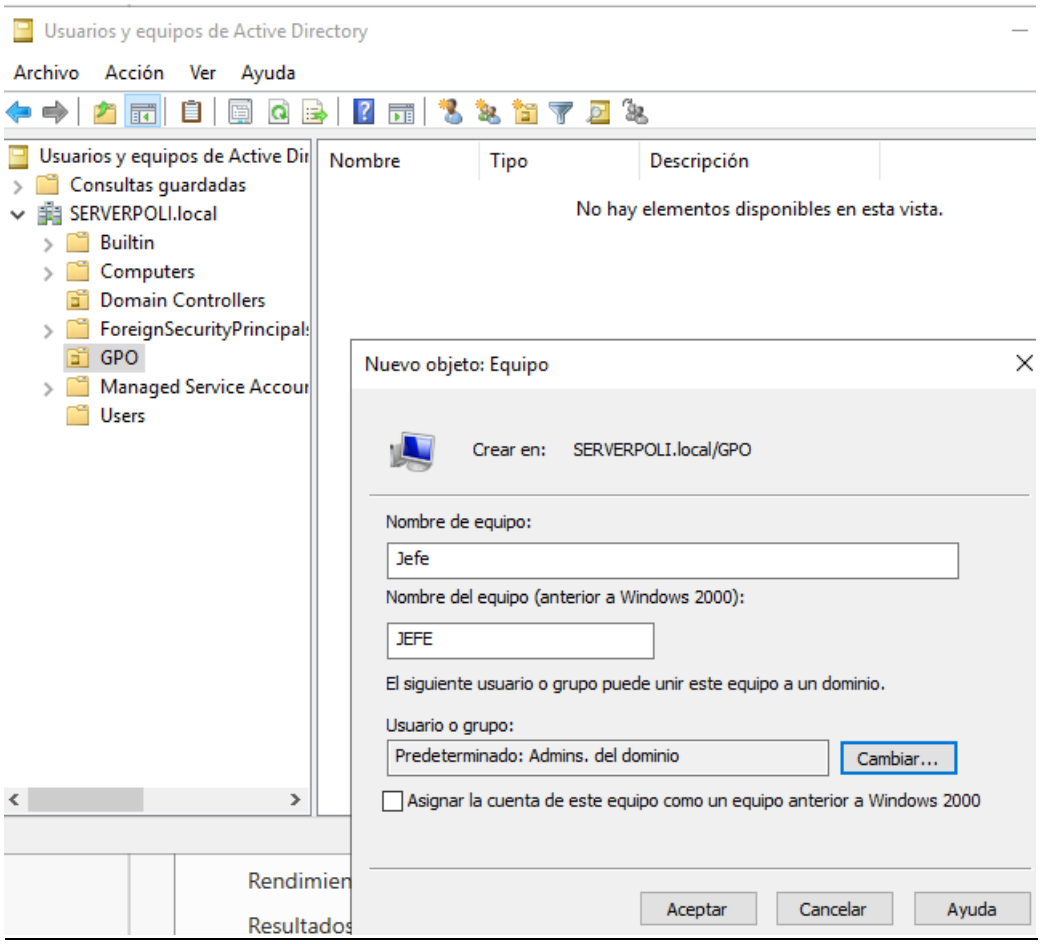


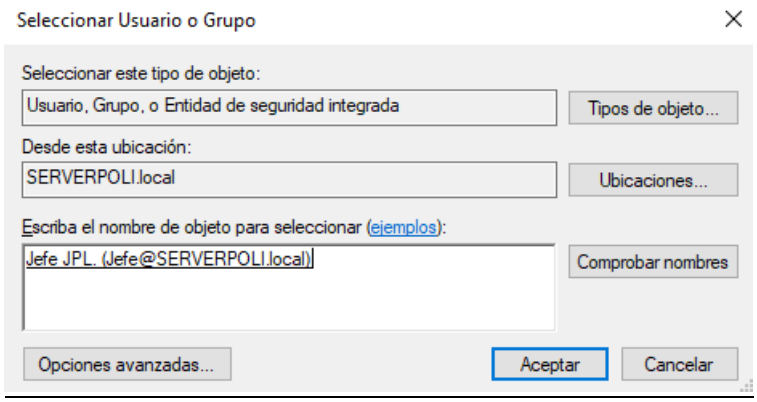
Para finalizar, solo quedara crear un nuevo usuario en AD, el cual se va a llamar Jefe.

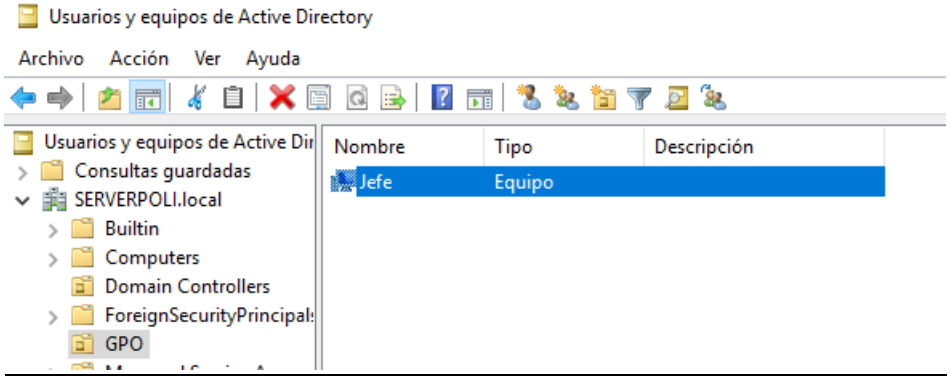




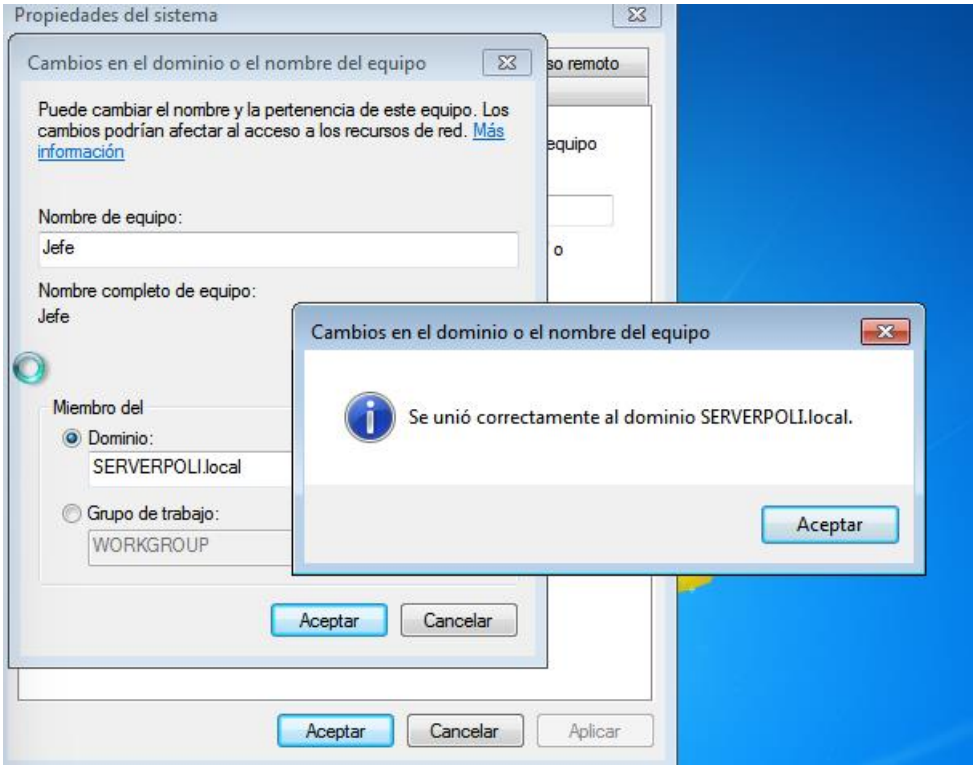
Una vez creado el usuario, haremos que su equipo este vinculado directamente a la GPO que hemos configurado anteriormente.



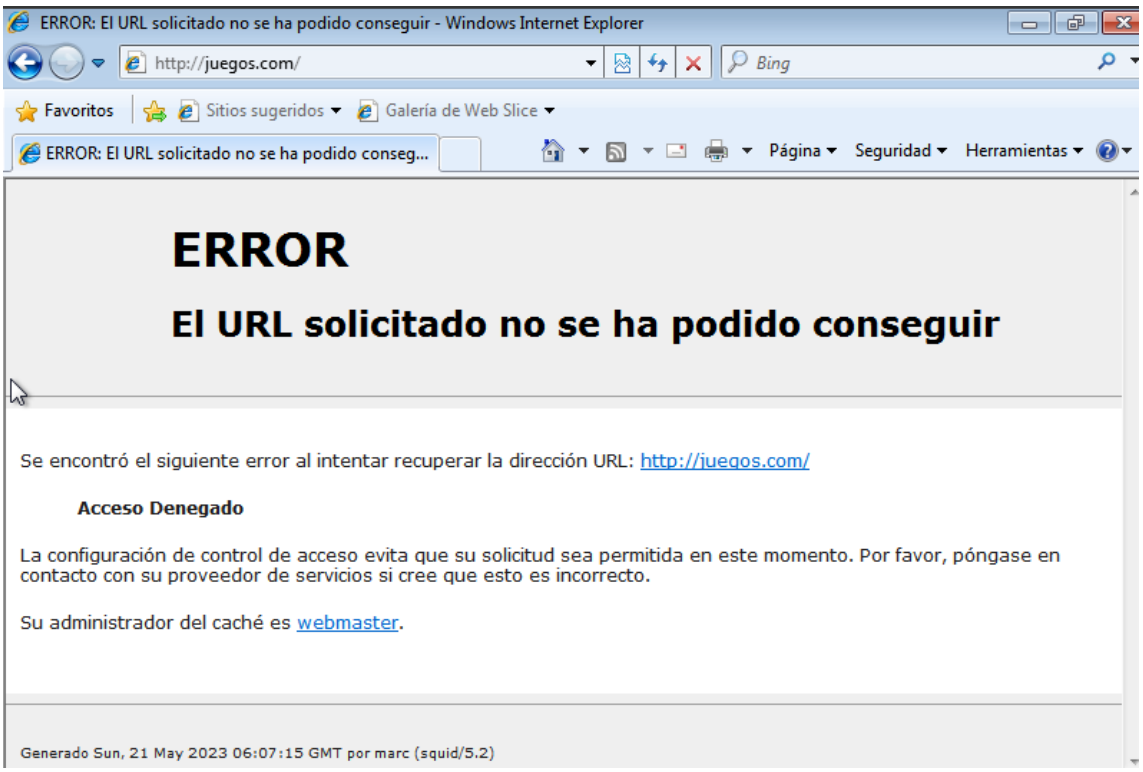




Una vez entrado en el cliente, lo uniremos al dominio del servidor Windows.

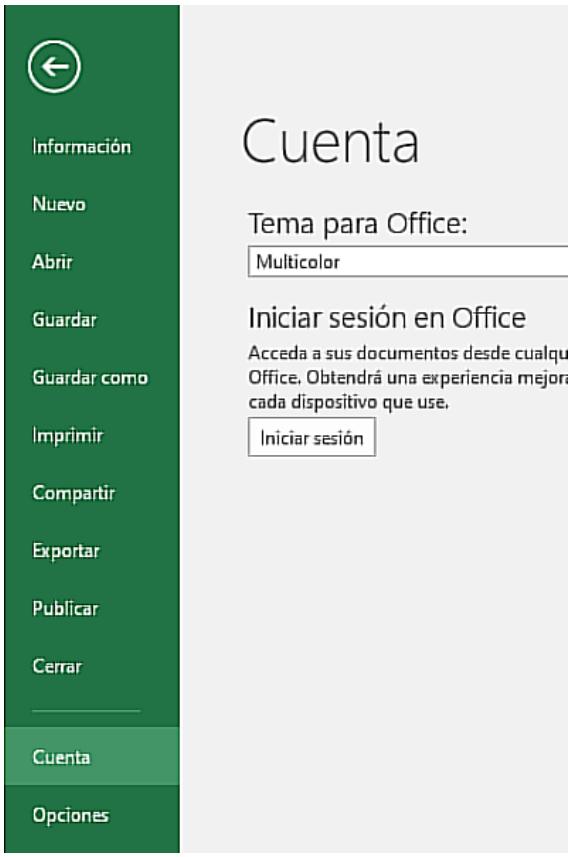


Para finalizar, entraremos dentro del usuario jefe, y buscaremos una de las cosas que están restringidas por nuestro PROXY en Ubuntu server, y comprobaremos que aquí también lo restrinja.

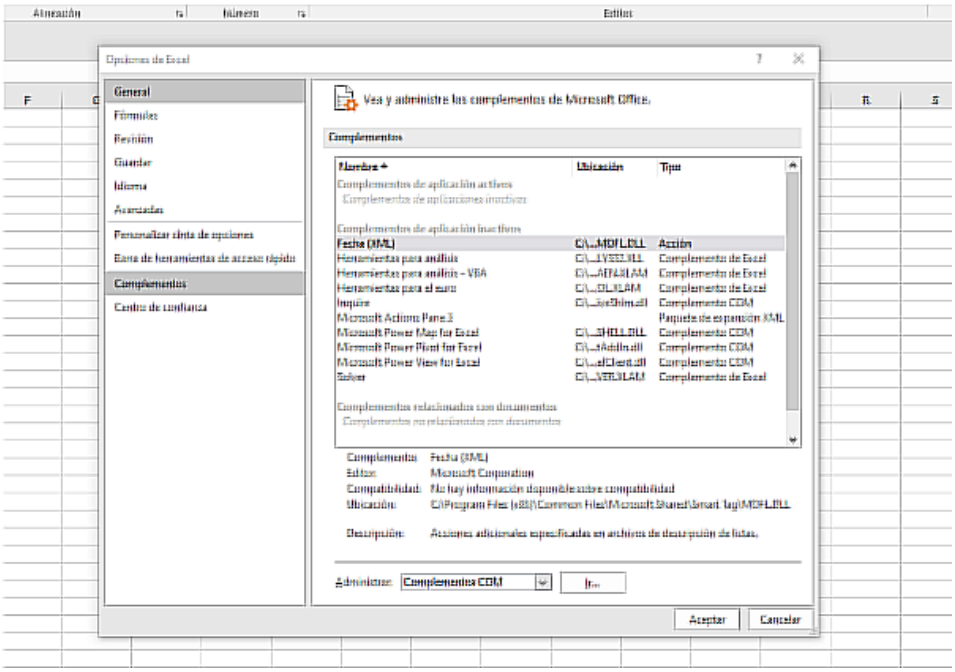


Robótica

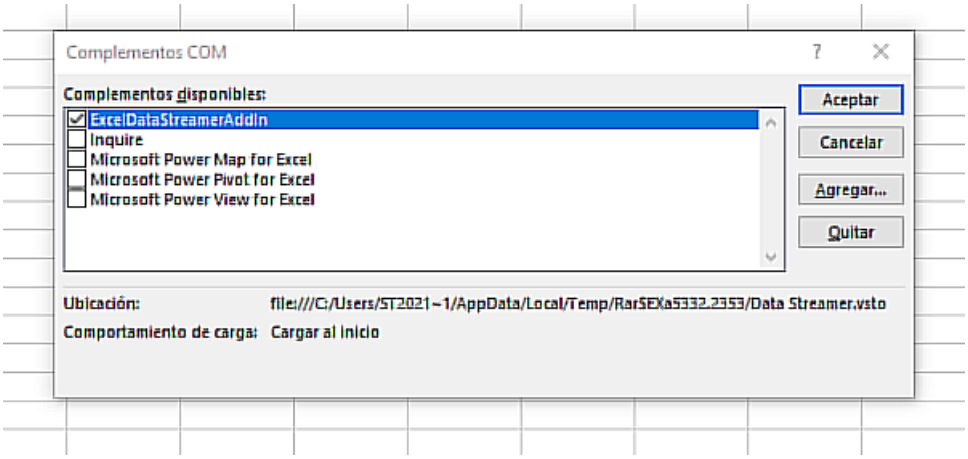
Vamos a inicio, opciones:



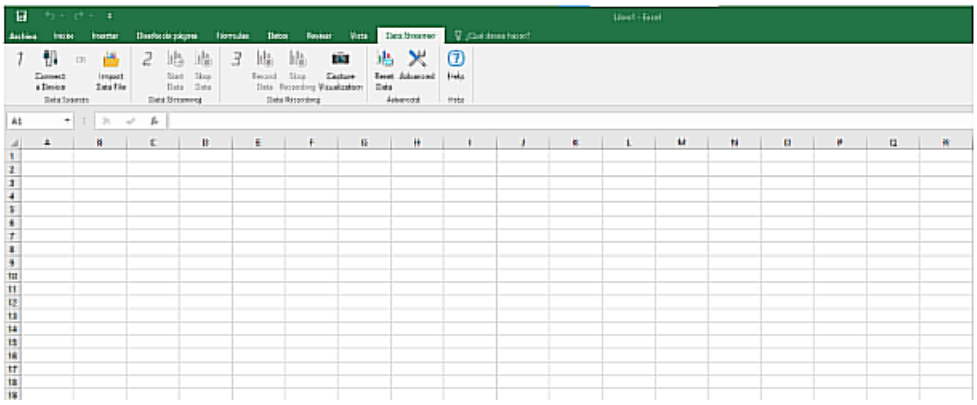
Vamos a complementos y ponemos “Complementos COM”:



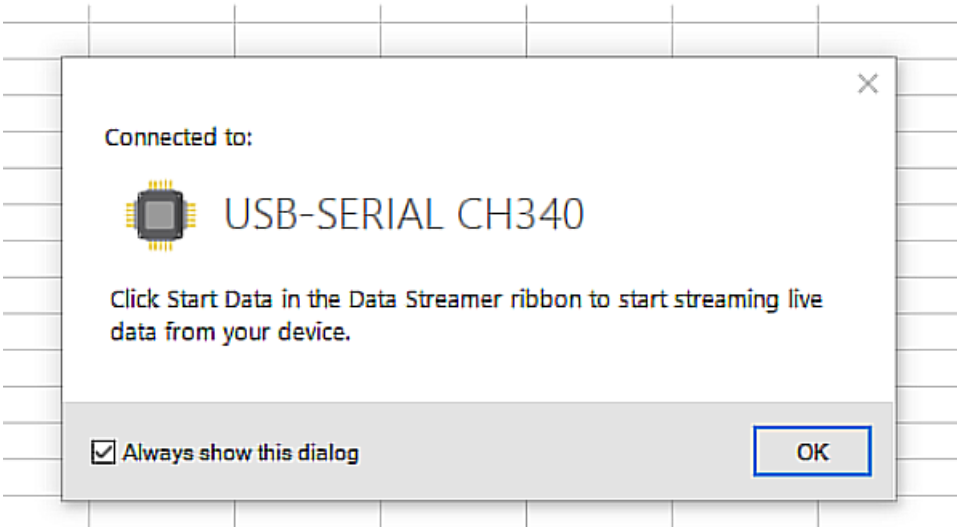
Y luego le damos a la derecha que sale “IR...” y seleccionamos la opcion de ExcelDataStreamAddin:



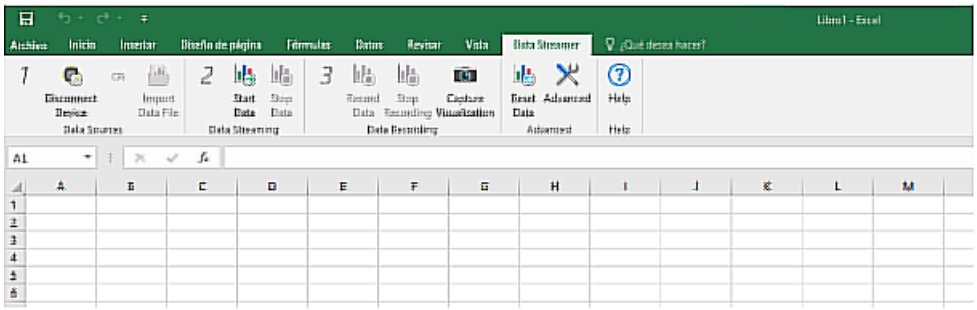
Le damos a “connect a Device” y seleccionamos nuestra placa:



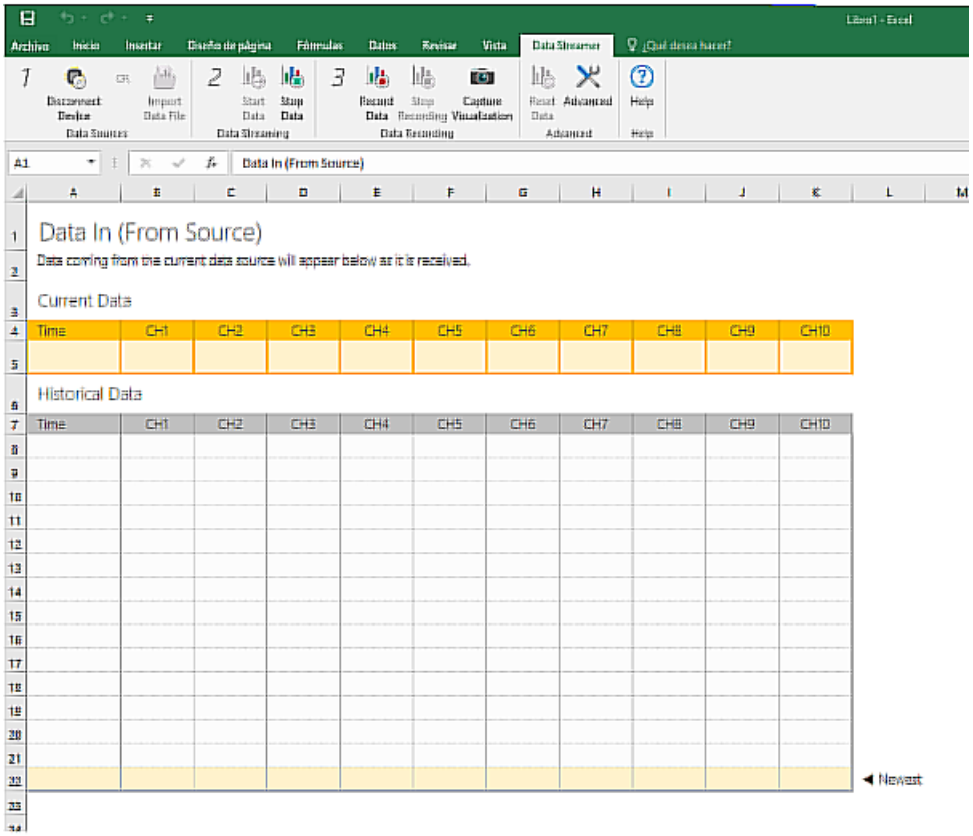
Aquí podemos ver que se ha conectado la placa al Excel:



Y ahora le daremos a “Start Data” para posteriormente guardar la info que genera arduino:



Y podemos ver aquí el Excel:



Video:

https://drive.google.com/file/d/1gcC1rQ_WZ05QyHiG2HihPXmshlHDFrbf/view?usp=sharing

PRESENTACION