

# SEGUNDA ENTREGA - ADMINISTRACIÓN DE SISTEMAS OPERATIVOS

## Punto a) Usuarios del sistema creados de acuerdo al estudio de roles

### Procedimiento realizado:

Se eliminaron usuarios y grupos previos que no correspondían al estudio de roles actual, y se creó un script para generar los usuarios y grupos según los roles técnicos requeridos para el servidor.

Los roles técnicos definidos fueron:

- **sysadmin** → grupo **sistema** (Administrador del sistema)
- **dbadmin** → grupo **bd** (Administrador de base de datos)
- **webadmin** → grupo **web** (Administrador web)
- **netadmin** → grupo **red** (Administrador de red)

### Script utilizado (create\_roles.sh):

```
#!/bin/bash
echo "==> Creando grupos de roles..."
for group in sistema bd web red
do
    if getent group $group >/dev/null; then
        echo "Grupo $group ya existe"
    else
        sudo groupadd $group
        echo "Grupo $group creado"
    fi
done

echo "==> Creando usuarios..."
declare -A users
users=(
    [sysadmin]=sistema
    [dbadmin]=bd
    [webadmin]=web
    [netadmin]=red
)
```

```

for user in "${!users[@]}"
do
    if id "$user" &>/dev/null; then
        echo "Usuario $user ya existe"
    else
        sudo useradd -m -s /bin/bash -g "${users[$user]}" "$user"
        echo "$user:Pass1234" | sudo chpasswd
        sudo chage -d 0 "$user"
        echo "Usuario $user creado en grupo ${users[$user]} (contraseña temporal: Pass1234,
se pedirá cambiarla en el primer login)"
    fi
done

echo "Usuarios y roles creados correctamente."

```

## Comandos ejecutados:

```

nano create_roles.sh
chmod +x create_roles.sh
sudo ./create_roles.sh

```

## Verificación:

```

getent group | grep -E "sistema|bd|web|red"
cut -d: -f1 /etc/passwd | grep -E "sysadmin|dbadmin|webadmin|netadmin"

```



```

sigie-admin@sigie-server: ~
=== GRUPOS CREADOS ===
sistema:x:1001:
bd:x:1002:
web:x:1003:
red:x:1004:

=== USUARIOS CREADOS ===
webadmin
netadmin
sysadmin
dbadmin
sigie-admin@sigie-server:~$ |

```

## Punto b) Menús de usuario para los roles técnicos (1ra versión navegable y aplicable)

### Procedimiento realizado:

Se creó un único script (menu.sh) que adapta el menú según el usuario que lo ejecute, utilizando el comando `whoami` para detectar el rol actual.

### Script utilizado (menu.sh):

```
#!/bin/bash
while true; do
    clear
    echo "=====
    echo "    Menú del rol: $(whoami)"
    echo "=====

    case $(whoami) in
        sysadmin)
            echo "1) Actualizar sistema"
            echo "2) Gestionar usuarios"
            echo "3) Ver uso de disco"
            echo "4) Salir"
            ;;
        dbadmin)
            echo "1) Iniciar servicio MySQL/MariaDB"
            echo "2) Detener servicio MySQL/MariaDB"
            echo "3) Ver estado de MySQL/MariaDB"
            echo "4) Salir"
            ;;
        webadmin)
            echo "1) Iniciar Apache"
            echo "2) Detener Apache"
            echo "3) Ver estado de Apache"
            echo "4) Salir"
            ;;
        netadmin)
            echo "1) Ver interfaces de red"
            echo "2) Ver reglas de firewall (UFW)"
            echo "3) Ver conexiones activas"
            echo "4) Salir"
            ;;
    esac

    echo "=====
    read -p "Seleccione una opción: " opcion
```

```

case $(whoami) in
    sysadmin)
        case $opcion in
            1) sudo apt update && sudo apt upgrade -y ;;
            2) cut -d: -f1 /etc/passwd | less ;;
            3) df -h ;;
            4) exit ;;
            *) echo "Opción inválida";;
        esac
        ;;
    dbadmin)
        case $opcion in
            1) sudo systemctl start mysql ;;
            2) sudo systemctl stop mysql ;;
            3) sudo systemctl status mysql ;;
            4) exit ;;
            *) echo "Opción inválida";;
        esac
        ;;
    webadmin)
        case $opcion in
            1) sudo systemctl start apache2 ;;
            2) sudo systemctl stop apache2 ;;
            3) sudo systemctl status apache2 ;;
            4) exit ;;
            *) echo "Opción inválida";;
        esac
        ;;
    netadmin)
        case $opcion in
            1) ip a ;;
            2) sudo ufw status ;;
            3) ss -tulnp ;;
            4) exit ;;
            *) echo "Opción inválida";;
        esac
        ;;
esac

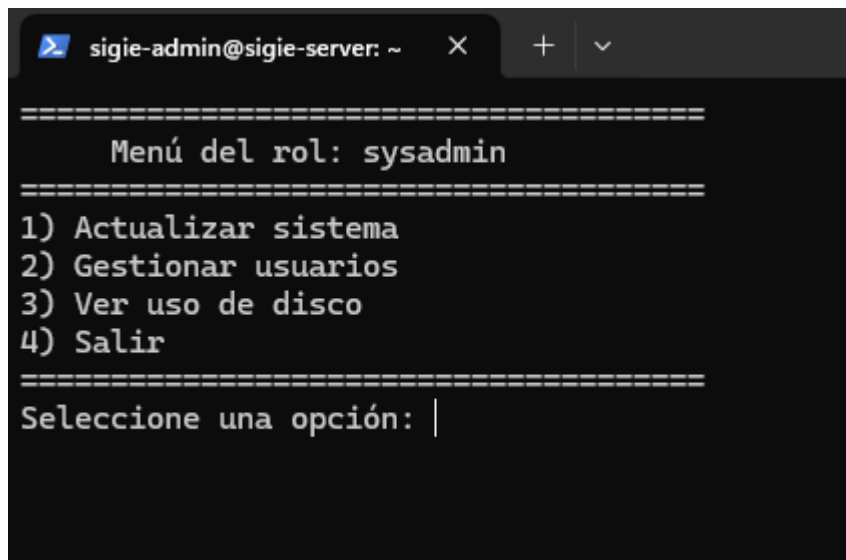
echo
read -p "Presiona Enter para continuar..."
done

```

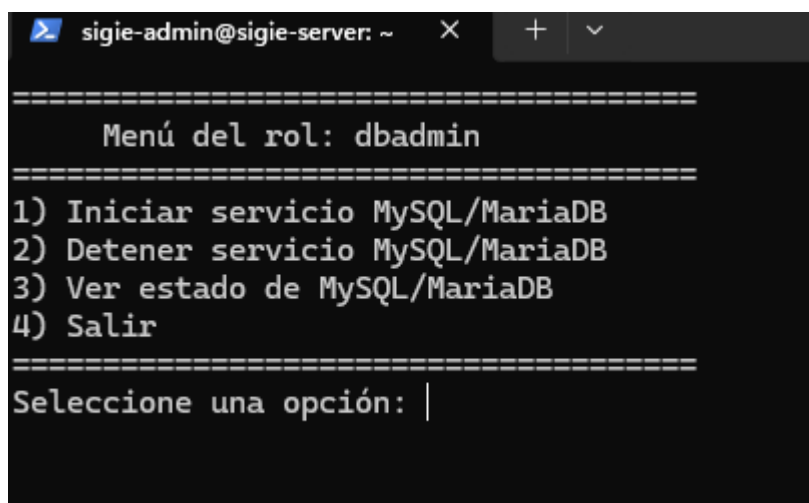
## Comandos ejecutados:

```
nano menu.sh
chmod +x menu.sh
sudo cp menu.sh /home/sysadmin/
sudo cp menu.sh /home/dbadmin/
sudo cp menu.sh /home/webadmin/
sudo cp menu.sh /home/netadmin/
```

```
# Configurar ejecución automática para cada usuario
echo "~/menu.sh" | sudo tee -a /home/sysadmin/.bashrc
echo "~/menu.sh" | sudo tee -a /home/dbadmin/.bashrc
echo "~/menu.sh" | sudo tee -a /home/webadmin/.bashrc
echo "~/menu.sh" | sudo tee -a /home/netadmin/.bashrc
```



```
sigie-admin@sigie-server: ~
=====
      Menú del rol: sysadmin
=====
1) Actualizar sistema
2) Gestionar usuarios
3) Ver uso de disco
4) Salir
=====
Seleccione una opción: |
```



```
sigie-admin@sigie-server: ~
=====
      Menú del rol: dbadmin
=====
1) Iniciar servicio MySQL/MariaDB
2) Detener servicio MySQL/MariaDB
3) Ver estado de MySQL/MariaDB
4) Salir
=====
Seleccione una opción: |
```

```
sigie-admin@sigie-server: ~  
=====   
      Menú del rol: webadmin   
=====   
1) Iniciar Apache   
2) Detener Apache   
3) Ver estado de Apache   
4) Salir   
=====   
Seleccione una opción: |
```

```
sigie-admin@sigie-server: ~  
=====   
      Menú del rol: netadmin   
=====   
1) Ver interfaces de red   
2) Ver reglas de firewall (UFW)   
3) Ver conexiones activas   
4) Salir   
=====   
Seleccione una opción: |
```

---

## Punto c) Menú del Administrador General

### Procedimiento realizado:

Se creó un script (admin\_menu.sh) exclusivo para el usuario sigie-admin con opciones más completas que reúnen las funciones de los distintos roles.

### Script utilizado (admin\_menu.sh):

```
#!/bin/bash  
while true; do  
    clear  
    echo "===== "  
    echo "  Menú del Administrador General"  
    echo "===== "  
    echo "1) Actualizar sistema"  
    echo "2) Gestionar usuarios"  
    echo "3) Ver estado de servicios (Apache, MySQL)"  
    echo "4) Gestionar red (IP y Firewall)"
```

```

echo "5) Copia de seguridad (home)"
echo "6) Ver logs del sistema"
echo "7) Salir"
echo "======"
read -p "Seleccione una opción: " opcion


case $opcion in
    1)
        echo ">>> Actualizando sistema..."
        sudo apt update && sudo apt upgrade -y
        ;;
    2)
        echo ">>> Lista de usuarios en el sistema:"
        cut -d: -f1 /etc/passwd | less
        ;;
    3)
        echo ">>> Estado de Apache:"
        sudo systemctl status apache2 --no-pager
        echo
        echo ">>> Estado de MySQL:"
        sudo systemctl status mysql --no-pager
        ;;
    4)
        echo ">>> Dirección IP del servidor:"
        ip a
        echo
        echo ">>> Estado del firewall UFW:"
        sudo ufw status
        ;;
    5)
        echo ">>> Realizando copia de seguridad de /home..."
        BACKUP_FILE="/home/sigie-admin/backup_home_$(date
+%F_%H-%M-%S).tar.gz"
        sudo tar -czvf $BACKUP_FILE /home
        echo "Backup creado en: $BACKUP_FILE"
        ;;
    6)
        echo ">>> Últimas 20 líneas de logs del sistema:"
        sudo tail -n 20 /var/log/syslog
        ;;
    7)
        echo "Saliendo del menú..."
        exit
        ;;
    *)
        echo "Opción inválida"
        ;;
esac

```

```
echo
read -p "Presiona Enter para continuar..."
done
```

### Comandos ejecutados:

```
nano admin_menu.sh
chmod +x admin_menu.sh
echo "~/admin_menu.sh" >> ~/.bashrc
```



```
Windows PowerShell
=====
Menú del Administrador General
=====
1) Actualizar sistema
2) Gestionar usuarios
3) Ver estado de servicios (Apache, MySQL)
4) Gestionar red (IP y Firewall)
5) Copia de seguridad (home)
6) Ver logs del sistema
7) Salir
=====
Seleccione una opción: |
```

---

## Punto d) Configuración de red en las terminales y el servidor

### Procedimiento realizado:

Se configuró una dirección IP fija en el servidor mediante Netplan para permitir conectividad estable desde las terminales cliente.

**Nota:** La configuración inicial se realizó en casa con IP 192.168.1.50, pero al trasladar la VM al colegio se reconfiguró para usar la red 192.168.2.50.

### Configuración aplicada en `/etc/netplan/01-netcfg.yaml`:



```
network:
  version: 2
  ethernet:
    enp0s3:
      dhcp4: no
      addresses:
        - 192.168.2.50/24
      routes:
        - to: default
          via: 192.168.2.1
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```

## Comandos ejecutados:

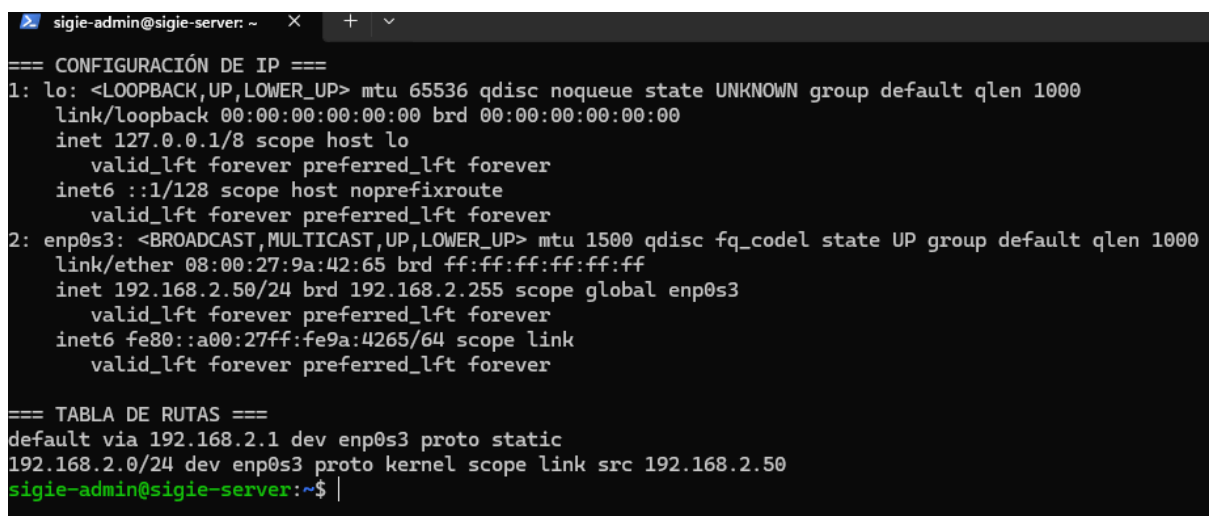
```
sudo nano /etc/netplan/01-netcfg.yaml
sudo chmod 600 /etc/netplan/*.yaml
sudo mv /etc/netplan/00-installer-config.yaml /etc/netplan/00-installer-config.yaml.bak
sudo mv /etc/netplan/50-cloud-init.yaml /etc/netplan/50-cloud-init.yaml.bak
sudo netplan apply
```

## Verificación:

```
ip a
ip r
```

## Resultado final:

- IP fija del servidor: 192.168.2.50/24
- Gateway: 192.168.2.1
- Conectividad probada desde clientes Windows via SSH



```
sigie-admin@sigie-server: ~
=== CONFIGURACIÓN DE IP ===
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:9a:42:65 brd ff:ff:ff:ff:ff:ff
   inet 192.168.2.50/24 brd 192.168.2.255 scope global enp0s3
       valid_lft forever preferred_lft forever
   inet6 fe80::a00:27ff:fe9a:4265/64 scope link
       valid_lft forever preferred_lft forever

=== TABLA DE RUTAS ===
default via 192.168.2.1 dev enp0s3 proto static
192.168.2.0/24 dev enp0s3 proto kernel scope link src 192.168.2.50
sigie-admin@sigie-server:~$
```

---

## Punto e) Configuración del servicio SSH en el cliente y el servidor

### Procedimiento realizado:

Se configuró y aseguró el servicio SSH para permitir acceso remoto seguro al servidor desde las terminales cliente.

### Comandos ejecutados en el servidor:

```
sudo systemctl status ssh  
sudo systemctl enable ssh  
sudo systemctl is-enabled ssh
```

### Configuración de seguridad aplicada:

```
sudo nano /etc/ssh/sshd_config
```

Parámetros configurados:

- `PermitRootLogin no` - Bloquea acceso directo del root por SSH
- `PasswordAuthentication yes` - Permite login por contraseña

```
sudo systemctl restart ssh
```

### Verificación desde cliente:

```
ssh sysadmin@192.168.2.50  
ssh dbadmin@192.168.2.50  
ssh webadmin@192.168.2.50  
ssh netadmin@192.168.2.50
```

```

=== ESTADO DEL SERVICIO SSH ===
[sudo] password for sigie-admin:
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-09-19 11:48:15 UTC; 51min ago
 TriggeredBy: ● ssh.socket
    Docs: man:sshd(8)
          man:sshd_config(5)
   Process: 13285 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Process: 13308 ExecReload=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Process: 13309 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
  Main PID: 13286 (sshd)
    Tasks: 1 (limit: 4605)
   Memory: 3.1M (peak: 4.0M)
      CPU: 102ms
   CGroup: /system.slice/ssh.service
           └─13286 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

sep 19 11:48:15 sigie-server systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
sep 19 11:49:45 sigie-server systemd[1]: Reloading ssh.service - OpenBSD Secure Shell server...
sep 19 11:49:45 sigie-server systemd[1]: Reloaded ssh.service - OpenBSD Secure Shell server.
sep 19 11:49:45 sigie-server sshd[13286]: Received SIGHUP; restarting.
sep 19 11:49:45 sigie-server sshd[13286]: Server listening on 0.0.0.0 port 22.
sep 19 11:49:45 sigie-server sshd[13286]: Server listening on :: port 22.
sep 19 12:30:56 sigie-server sshd[14120]: Accepted password for sigie-admin from 192.168.2.221 port 59904 ssh2
sep 19 12:30:56 sigie-server sshd[14120]: pam_unix(sshd:session): session opened for user sigie-admin(uid=1000)...n(uid=0)
sep 19 12:38:49 sigie-server sshd[14280]: Accepted password for sigie-admin from 192.168.2.221 port 50985 ssh2
sep 19 12:38:49 sigie-server sshd[14280]: pam_unix(sshd:session): session opened for user sigie-admin(uid=1000)...n(uid=0)
Hint: Some lines were ellipsized, use -l to show in full.

=== SSH HABILITADO AL INICIO ===
enabled
sigie-admin@sigie-server:~$

```

## Punto f) Archivos crontab con rutinas de backup y sus correspondientes scripts para el administrador

### Procedimiento realizado:

Se creó un sistema de backup automático administrado por el rol sysadmin, consistente en un script de respaldo y su programación via crontab.

### Preparación de la carpeta de backups:

```

# Desde sigie-admin:
sudo mkdir -p /var/backups/sigie
sudo chown sysadmin:sistema /var/backups/sigie
sudo chmod 770 /var/backups/sigie

```

### Script de backup creado (/home/sysadmin/backup\_sigie.sh):

```

#!/bin/bash
# Script de backup SIGIE

# Fecha actual
FECHA=$(date +%Y-%m-%d_%H-%M-%S)

# Carpeta de destino
DESTINO="/var/backups/sigie"

```

```
# Respaldo de configuraciones y usuarios
tar -czf $DESTINO/backup_sistema_$(date +%Y%m%d).tar.gz /etc /home
```

```
# (Más adelante, cuando tengamos MySQL/MariaDB instalado)
# mysqldump -u root -pTU_PASS_AQUI --all-databases >
$DESTINO/backup_bd_$(date +%Y%m%d).sql
```

```
echo "Backup completado en $DESTINO a las $(date +%Y%m%d %H:%M)"
```

### Comandos ejecutados como sysadmin:

```
# Entrar como sysadmin y salir del menú (opción 4)
nano /home/sysadmin/backup_sigie.sh
chmod +x /home/sysadmin/backup_sigie.sh
```

```
# Probar manualmente
/home/sysadmin/backup_sigie.sh
```

```
# Verificar backup creado
ls -lh /var/backups/sigie/
```

```
# Programar con crontab
crontab -e
# Agregar línea: 0 2 * * * /home/sysadmin/backup_sigie.sh
```

```
# Verificar programación
crontab -l
```

**Resultado:** El script genera backups automáticos todos los días a las 2:00 AM con el contenido de /etc y /home.

```
=== PROGRAMACIÓN CRON (sysadmin) ===
Password:
#
# m h dom mon dow  command
0 2 * * * /home/sysadmin/backup_sigie.sh

=== PROGRAMACIÓN CRON (sigie-admin) ===
#
# m h dom mon dow  command
0 3 * * * /home/sigie-admin/backup_full.sh
sigie-admin@sigie-server:~$ |
```

---

## Punto g) Instalación y configuración del servicio web HTTP

### Procedimiento realizado:

Se instaló y configuró Apache2 como servidor web, creando un sitio personalizado para SIGIE en lugar de usar la página por defecto.

### Comandos ejecutados como sigie-admin:

```
sudo apt update
sudo apt install -y apache2
sudo systemctl status apache2
sudo systemctl enable apache2
sudo ufw allow 'Apache Full'
```

### Configuración del sitio SIGIE:

```
sudo mkdir -p /var/www/sigie
sudo chown -R www-data:www-data /var/www/sigie
sudo chmod -R 755 /var/www/sigie
echo "<h1>SIGIE - Servidor Web Activo</h1>" | sudo tee /var/www/sigie/index.html
```

### Configuración de Virtual Host (/etc/apache2/sites-available/sigie.conf):

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName sigie.local
    DocumentRoot /var/www/sigie

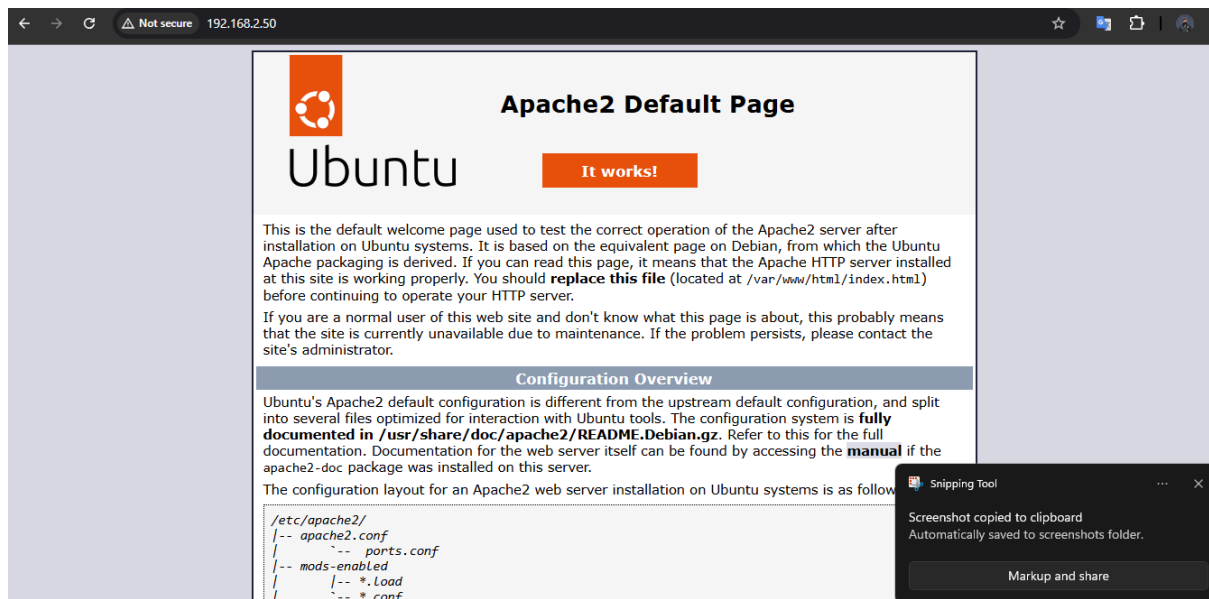
    ErrorLog ${APACHE_LOG_DIR}/sigie_error.log
    CustomLog ${APACHE_LOG_DIR}/sigie_access.log combined
</VirtualHost>
```

### Activación del sitio:

```
sudo nano /etc/apache2/sites-available/sigie.conf
sudo a2ensite sigie.conf
sudo a2dissite 000-default.conf
sudo systemctl reload apache2
```

### Verificación:

Acceso desde navegador cliente: <http://192.168.2.50> Resultado: Muestra "pache2 Ubuntu Default Page"



## Punto h) Instalación y configuración de la base de datos MySQL/MariaDB

### Procedimiento realizado:

Se instaló MariaDB como motor de base de datos, se aseguró la instalación, se importó la base de datos SIGIE existente y se creó un usuario específico para la aplicación web.

### Instalación y configuración inicial:

```
sudo apt update
sudo apt install -y mariadb-server mariadb-client
sudo systemctl status mariadb
sudo systemctl enable mariadb
```

### Asegurar la instalación:

```
sudo mysql_secure_installation
```

### Configuración aplicada:

- Switch to unix\_socket authentication: n
- Change the root password: y (se estableció contraseña)
- Remove anonymous users: y
- Disallow root login remotely: y
- Remove test database: y
- Reload privilege tables: y

## **Creación e importación de la base de datos:**

### **Transferencia del archivo .sql desde Windows:**

# Desde PowerShell Windows:

```
scp "C:\Users\trapd\OneDrive\Desktop\sigie.sql"  
sigie-admin@192.168.2.50:/home/sigie-admin/
```

### **Creación e importación en el servidor:**

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

```
CREATE DATABASE sigie CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;  
EXIT;
```

```
mysql -u root -p sigie < /home/sigie-admin/sigie.sql
```

### **Verificación de la importación:**

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

```
SHOW DATABASES;  
USE sigie;  
SHOW TABLES;
```

Tablas importadas exitosamente:

- calendario
- calendarios
- clases
- eventos
- usuarios
- usuarios\_clases

### **Creación de usuario para la aplicación:**

```
CREATE USER 'sigie_user'@'localhost' IDENTIFIED BY 'sigie2025';  
GRANT ALL PRIVILEGES ON sigie.* TO 'sigie_user'@'localhost';  
FLUSH PRIVILEGES;  
EXIT;
```

```
sigie-admin@sigie-server: ~  
+-----+  
| Database |  
+-----+  
| information_schema |  
| sigie |  
+-----+  
  
+-----+  
| Tables_in_sigie |  
+-----+  
| calendario |  
| calendarios |  
| clases |  
| eventos |  
| usuarios |  
| usuarios_clases |  
+-----+  
sigie-admin@sigie-server:~$
```

## Punto i) Shell script con simulación o conexión a la base de datos y las consultas pertinentes

### Procedimiento realizado:

Se creó un script que demuestra la conectividad a la base de datos SIGIE y ejecuta consultas de prueba usando el usuario sigie\_user.

### Script creado (db\_test.sh):

```
#!/bin/bash  
# Script de prueba de conexión a la base de datos SIGIE  
  
DB_USER="sigie_user"  
DB_PASS="sigie2025"  
DB_NAME="sigie"  
  
echo "=====  
echo " Prueba de conexión a la BD SIGIE"  
echo "=====
```



```
# Consultar cantidad de usuarios
mysql -u $DB_USER -p$DB_PASS -D $DB_NAME -e "SELECT COUNT(*) AS
total_usuarios FROM usuarios;"

# Listar las primeras 5 clases
mysql -u $DB_USER -p$DB_PASS -D $DB_NAME -e "SELECT * FROM clases LIMIT 5;"

# Mostrar eventos registrados
mysql -u $DB_USER -p$DB_PASS -D $DB_NAME -e "SELECT * FROM eventos LIMIT 5;"

echo "=====
echo " Consulta finalizada correctamente"
echo "=====
```

## Comandos ejecutados como sigie-admin:

```
nano ~/db_test.sh
chmod +x ~/db_test.sh
./db_test.sh
```

## Resultado de la ejecución:

- Total usuarios: 6
- Se mostraron 5 clases (artístico, bt, Bt, informatica, linguistico)
- Se mostraron 2 eventos (footbal, the flash)

```
sigie-admin@sigie-server: ~
=====
Prueba de conexión a la BD SIGIE
=====
+-----+
| total_usuarios |
+-----+
|          6     |
+-----+

+-----+-----+-----+
| id_clase | nombre | año |
+-----+-----+-----+
|    16    | artístico | 2 |
|    15    | bt       | 1 |
|    10    | Bt       | 3 |
|    17    | informatica | 2 |
|    13    | linguistico | 3 |
+-----+-----+-----+

+-----+-----+-----+-----+-----+-----+
| id_evento | titulo | fecha | tipo | id_clase | creado_por | imagen |
+-----+-----+-----+-----+-----+-----+
|    2     | futbol | 2025-10-11 | general | NULL | 9 | 1755693142_paris-2024-olympics-soccer.jpg |
|    3     | the flash | 2025-01-02 | general | NULL | 9 | 1755694096_aleksandra-svyripa-sWRZvuCFp8E-unsplash.jpg |
+-----+-----+-----+-----+-----+-----+

=====
Consulta finalizada correctamente
=====
sigie-admin@sigie-server:~$ |
```

## Punto j) Menú que permita conectar todos los anteriores scripts para facilitar su uso - primera versión

### Procedimiento realizado:

Se creó un menú maestro que centraliza el acceso a todos los scripts desarrollados en los puntos anteriores, funcionando como una interfaz de administración integrada.

### Script creado (menu\_principal.sh):

```
#!/bin/bash
# Menú principal de administración SIGIE

while true; do
    clear
    echo "=====
    echo "    Menú Principal de SIGIE"
    echo "=====
    echo "1) Crear usuarios y roles del sistema"
    echo "2) Ver menú de Sysadmin (backup, usuarios, etc.)"
    echo "3) Ver menú de conexión a la BD (db_test.sh)"
    echo "4) Ver configuraciones de red actuales"
    echo "5) Ejecutar backup manual"
    echo "6) Salir"
    echo "=====
    read -p "Seleccione una opción: " opcion

    case $opcion in
        1) bash ~/create_roles.sh ;;
        2) su - sysadmin -c "/home/sysadmin/menu.sh" ;;
        3) bash ~/db_test.sh ;;
        4) ip a; ip r; read -p 'Presione ENTER para continuar...' ;;
        5) su - sysadmin -c "/home/sysadmin/backup_sigie.sh"; read -p 'Presione ENTER para
continuar...' ;;
        6) echo "Saliendo..."; exit 0 ;;
        *) echo "Opción inválida"; sleep 2 ;;
    esac
done
```

### Comandos ejecutados como sigie-admin:

```
nano ~/menu_principal.sh
chmod +x ~/menu_principal.sh
```

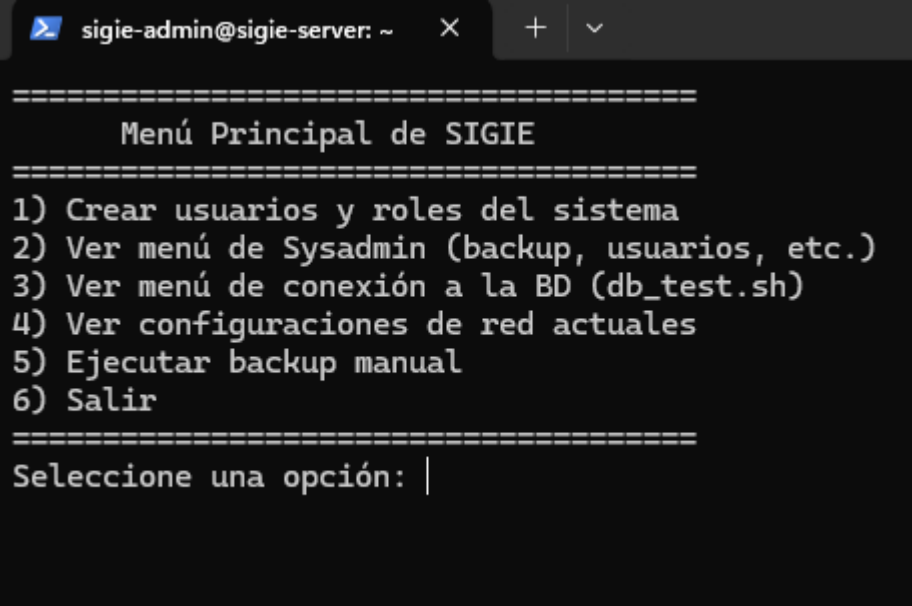
## Funcionalidad:

El menú permite ejecutar:

- Script de creación de usuarios y roles (punto a)
- Acceso al menú de sysadmin (punto b-f)
- Prueba de conexión a base de datos (punto i)
- Visualización de configuración de red (punto d)
- Ejecución manual de backup (punto f)

Este menú se ejecuta manualmente cuando sigie-admin lo requiere con:

```
./menu_principal.sh
```

A screenshot of a terminal window with a dark background. The window title bar shows 'sigie-admin@sigie-server: ~' and standard window controls. The terminal output displays a menu titled 'Menú Principal de SIGIE' enclosed in a box of equals signs. The menu lists six options: 1) Crear usuarios y roles del sistema, 2) Ver menú de Sysadmin (backup, usuarios, etc.), 3) Ver menú de conexión a la BD (db\_test.sh), 4) Ver configuraciones de red actuales, 5) Ejecutar backup manual, and 6) Salir. Below the menu, it prompts 'Seleccione una opción: |' with a cursor.

```
sigie-admin@sigie-server: ~  
=====   
          Menú Principal de SIGIE   
=====   
1) Crear usuarios y roles del sistema   
2) Ver menú de Sysadmin (backup, usuarios, etc.)   
3) Ver menú de conexión a la BD (db_test.sh)   
4) Ver configuraciones de red actuales   
5) Ejecutar backup manual   
6) Salir   
=====   
Seleccione una opción: |
```

---

## PUNTOS EXTRA:

### Punto k) Servicios adicionales para el despliegue de aplicaciones (PHP y phpMyAdmin)

#### Procedimiento realizado:

Se instalaron servicios adicionales necesarios para que el servidor pueda ejecutar la aplicación SIGIE desarrollada en PHP y facilitar la administración de la base de datos.

#### Objetivo:

Preparar completamente el entorno de producción para que la aplicación web pueda funcionar sin problemas adicionales en futuras entregas.

## Por qué es necesario:

- La aplicación SIGIE está desarrollada en PHP y requiere un intérprete con extensiones específicas
- phpMyAdmin proporciona una interfaz gráfica intuitiva para administrar la base de datos
- Estas herramientas facilitan el desarrollo, pruebas y mantenimiento del sistema

## 1. Instalación y configuración de PHP

### Instalación de PHP 8.3 y extensiones esenciales:

```
sudo apt update
```

```
sudo apt install -y php libapache2-mod-php php-mysql php-xml php-mbstring php-curl
```

### Componentes instalados:

- **php**: Intérprete principal de PHP 8.3
- **libapache2-mod-php**: Módulo para integrar PHP con Apache2
- **php-mysql**: Extensión para conectividad con MariaDB/MySQL
- **php-xml**: Procesamiento de documentos XML
- **php-mbstring**: Manejo de cadenas multibyte
- **php-curl**: Cliente para transferencia de datos HTTP/HTTPS

### Verificación de la instalación:

```
php -v
```

```
sudo systemctl restart apache2
```

### Creación de archivo de prueba:

```
echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php
```

## 2. Resolución de problemas de configuración Apache

Durante la implementación se detectó que Apache no estaba sirviendo correctamente los archivos PHP debido a una configuración incompleta del VirtualHost por defecto.

### Problema identificado:

El archivo `/etc/apache2/sites-enabled/000-default.conf` no existía, causando que Apache no tuviera un DocumentRoot definido para servir archivos desde `/var/www/html/`.

### **Solución implementada:**

# Recreación del VirtualHost por defecto

```
echo '<VirtualHost *:80>
```

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

```
    <Directory /var/www/html>
```

```
        Options Indexes FollowSymLinks
```

```
        AllowOverride All
```

```
        Require all granted
```

```
    </Directory>
```

```
</VirtualHost>' | sudo tee /etc/apache2/sites-available/000-default.conf
```

# Activación del sitio

```
sudo a2ensite 000-default.conf
```

```
sudo systemctl reload apache2
```

## **3. Instalación y configuración de phpMyAdmin**

### **Proceso de instalación:**

```
sudo apt install -y phpmyadmin
```

### **Configuración durante la instalación:**

- Servidor web seleccionado: **apache2** ✓
- Configuración automática de base de datos: **No** (para mantener control sobre la configuración existente)

### **Resolución de problemas de acceso:**

**Problema inicial:** phpMyAdmin no se cargaba correctamente desde el navegador

**Causa:** Falta de archivo de configuración de Apache para phpMyAdmin

### **Solución implementada:**

# Creación manual del archivo de configuración

```
sudo nano /etc/apache2/conf-available/phpmyadmin.conf
```

Contenido del archivo de configuración:

```
# phpMyAdmin default Apache configuration
```

```
Alias /phpmyadmin /usr/share/phpmyadmin
```

```
<Directory /usr/share/phpmyadmin>
```

```
    Options SymLinksIfOwnerMatch
```

```
    DirectoryIndex index.php
```

```

<IfModule mod_php8.c>
    AddType application/x-httpd-php .php
    php_flag magic_quotes_gpc Off
    php_flag track_vars On
    php_flag register_globals Off
    php_value include_path .
</IfModule>
</Directory>

```

### Activación de la configuración:

# Limpieza de configuraciones previas conflictivas  
 sudo rm /etc/apache2/conf-enabled/phpmyadmin.conf

# Activación correcta de la configuración  
 sudo a2enconf phpmyadmin.conf  
 sudo systemctl restart apache2

## 4. Pruebas y verificación

### Acceso a PHP:

URL: <http://192.168.2.50/info.php> Resultado: Página de información de PHP 8.3 con todas las extensiones cargadas

PHP Version 8.3.6	
System	Linux sigie-server 6.8.0-83-generic #83-Ubuntu SMP PREEMPT_DYNAMIC Fri Sep 5 21:46:54 UTC 2025 x86_64
Build Date	Jul 14 2025 18:30:55
Build System	Linux
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/8.3/apache2
Loaded Configuration File	/etc/php/8.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/8.3/apache2/conf.d
Additional .ini files parsed	/etc/php/8.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/8.3/apache2/conf.d/10-opcache.ini, /etc/php/8.3/apache2/conf.d/10-pdo.ini, /etc/php/8.3/apache2/conf.d/15-xml.ini, /etc/php/8.3/apache2/conf.d/20-bz2.ini, /etc/php/8.3/apache2/conf.d/20-calendar.ini, /etc/php/8.3/apache2/conf.d/20-ctype.ini, /etc/php/8.3/apache2/conf.d/20-curl.ini, /etc/php/8.3/apache2/conf.d/20-dom.ini, /etc/php/8.3/apache2/conf.d/20-exif.ini, /etc/php/8.3/apache2/conf.d/20-ffi.ini, /etc/php/8.3/apache2/conf.d/20-fileinfo.ini, /etc/php/8.3/apache2/conf.d/20-ftp.ini, /etc/php/8.3/apache2/conf.d/20-gd.ini, /etc/php/8.3/apache2/conf.d/20-gettext.ini, /etc/php/8.3/apache2/conf.d/20-iconv.ini, /etc/php/8.3/apache2/conf.d/20-imagick.ini, /etc/php/8.3/apache2/conf.d/20-ldap.ini, /etc/php/8.3/apache2/conf.d/20-ldap_sasl.ini, /etc/php/8.3/apache2/conf.d/20-mbstring.ini, /etc/php/8.3/apache2/conf.d/20-mcrypt.ini, /etc/php/8.3/apache2/conf.d/20-mysql.ini, /etc/php/8.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/8.3/apache2/conf.d/20-phar.ini, /etc/php/8.3/apache2/conf.d/20-posix.ini, /etc/php/8.3/apache2/conf.d/20-readline.ini, /etc/php/8.3/apache2/conf.d/20-shmop.ini, /etc/php/8.3/apache2/conf.d/20-simplexml.ini, /etc/php/8.3/apache2/conf.d/20-sockets.ini, /etc/php/8.3/apache2/conf.d/20-sysmsg.ini, /etc/php/8.3/apache2/conf.d/20-syssem.ini, /etc/php/8.3/apache2/conf.d/20-sysvshm.ini, /etc/php/8.3/apache2/conf.d/20-tokenizer.ini, /etc/php/8.3/apache2/conf.d/20-xmlreader.ini, /etc/php/8.3/apache2/conf.d/20-xmlwriter.ini, /etc/php/8.3/apache2/conf.d/20-xsl.ini, /etc/php/8.3/apache2/conf.d/20-zip.ini
PHP API	20230831
PHP Extension	20230831
Zend Extension	420230831
Zend Extension Build	API420230831.NTS
PHP Extension Build	API20230831.NTS

## Acceso a phpMyAdmin:

URL: <http://192.168.2.50/phpmyadmin> Resultado: Interfaz de login funcional



## Credenciales de acceso a phpMyAdmin:

- **Usuario recomendado para la aplicación:** `sigie_user`
- **Contraseña:** `sigie2025`
- **Base de datos accesible:** `sigie` (con todas las tablas: usuarios, clases, eventos, calendarios, etc.)

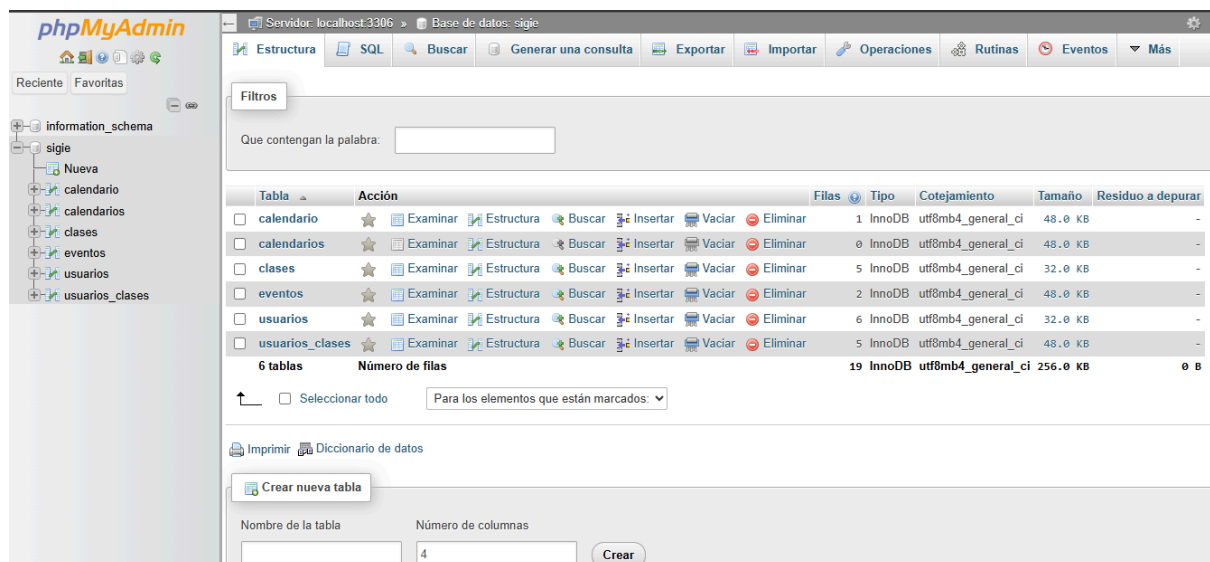


Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> calendario		1	InnoDB	utf8mb4_general_ci	48.0 KB	-
<input type="checkbox"/> calendarios		0	InnoDB	utf8mb4_general_ci	48.0 KB	-
<input type="checkbox"/> clases		5	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> eventos		2	InnoDB	utf8mb4_general_ci	48.0 KB	-
<input type="checkbox"/> usuarios		6	InnoDB	utf8mb4_general_ci	32.0 KB	-
<input type="checkbox"/> usuarios_clases		5	InnoDB	utf8mb4_general_ci	48.0 KB	-
<b>6 tablas</b>	<b>Número de filas</b>	<b>19</b>	<b>InnoDB</b>	<b>utf8mb4_general_ci</b>	<b>256.0 KB</b>	<b>0 B</b>

## Prueba de funcionalidad:

Se verificó el acceso completo a la base de datos sigie desde phpMyAdmin, confirmando que:

- Todas las tablas están presentes y accesibles
- El usuario `sigie_user` tiene permisos completos sobre la base

- La interfaz permite realizar consultas, exportaciones e importaciones

---

## Punto I) Sistema de backups avanzado y recuperación ante fallos

### Procedimiento realizado:

Se implementó un sistema robusto de respaldo que protege tanto los datos como la configuración del servidor, incluyendo replicación automática a la máquina host.

### Objetivo:

Garantizar la continuidad del servicio y la recuperación rápida ante fallos mediante múltiples capas de protección de datos.

### 1. Configuración de carpeta compartida con VirtualBox

#### Preparación en el host (VirtualBox):

1. Apagar la máquina virtual
2. Configuración → Carpetas compartidas → Agregar nueva
3. Configuración aplicada:
  - **Ruta de la carpeta:** C:\SIGIE\_Backups (en el host Windows)
  - **Nombre de la carpeta:** SIGIE\_Backups
  - **Opciones:** Automontar ✓, Solo lectura ✗

#### Configuración en la máquina virtual:

# Instalación de herramientas para carpetas compartidas

```
sudo apt install -y virtualbox-guest-utils
```

# Creación del punto de montaje

```
sudo mkdir -p /mnt/host_backups
```

# Montaje de la carpeta compartida

```
sudo mount -t vboxsf SIGIE_Backups /mnt/host_backups
```

# Verificación del montaje

```
mount | grep vboxsf
```

```
ls /mnt/host_backups
```



## 2. Script de backup completo del sistema

### Creación del script avanzado (backup\_full.sh):

```
#!/bin/bash
# backup_full.sh - Backup completo de configuración y datos SIGIE

FECHA=$(date +%Y%m%d_%H%M%S)
DESTINO="/var/backups/sigie"

# Crear carpeta si no existe
sudo mkdir -p $DESTINO

# Crear backup comprimido con todos los componentes críticos
sudo tar -czf $DESTINO/full_backup_$FECHA.tar.gz \
    /etc/apache2 \
    /etc/mysql \
    /var/www/html \
    /home/sigie-admin \
    --exclude=$DESTINO

echo "✅ Backup completo generado en $DESTINO/full_backup_$FECHA.tar.gz"

# Copia automática al host para protección externa
sudo cp "$DESTINO/full_backup_$FECHA.tar.gz" /mnt/host_backups/
```

### Componentes respaldados:

- **/etc/apache2:** Configuración completa del servidor web
- **/etc/mysql:** Configuración de MariaDB
- **/var/www/html:** Sitio web y aplicaciones
- **/home/sigie-admin:** Scripts de administración y configuraciones personalizadas

## 3. Automatización con crontab

### Programación del backup completo:

```
crontab -e
# Línea agregada:
0 3 * * * /home/sigie-admin/backup_full.sh
```

### Resultado de la automatización:

- **2:00 AM:** Backup de base de datos (sysadmin)
- **3:00 AM:** Backup completo del sistema (sigie-admin)
- **Automático:** Copia a la carpeta compartida del host

## 4. Script de restauración del sistema

### Creación del script de recuperación (restore\_full.sh):

```
#!/bin/bash
# restore_full.sh - Restaura un backup completo de SIGIE

BACKUP_DIR="/var/backups/sigie"

echo "=====
echo "  Restauración de backup SIGIE"
echo "=====

# Listar backups disponibles
echo "Backups disponibles:"
ls -lh $BACKUP_DIR/*.tar.gz
echo

# Solicitar archivo a restaurar
read -p "Ingrese el nombre del backup a restaurar (ej:
full_backup_20250918_121637.tar.gz): " BACKUP_FILE

# Verificar existencia del archivo
if [ ! -f "$BACKUP_DIR/$BACKUP_FILE" ]; then
    echo "❌ El archivo $BACKUP_FILE no existe en $BACKUP_DIR"
    exit 1
fi

# Proceso de restauración
echo "📦 Restaurando desde $BACKUP_FILE..."
sudo tar -xzf "$BACKUP_DIR/$BACKUP_FILE" -C /

# Restaurar base de datos si existe dump
if [ -f "/home/sigie-admin/sigie.sql" ]; then
    echo "📄 Restaurando base de datos SIGIE..."
    mysql -u root -p sigie < /home/sigie-admin/sigie.sql
fi

echo "=====
echo "✅ Restauración completada"
echo "=====
```

## 5. Protección mediante snapshots de VirtualBox

### Ventajas de los snapshots:

- Captura instantánea del estado completo de la máquina virtual
- Recuperación en minutos ante fallos catastróficos
- Preservación de memoria, disco y configuración del sistema

### Recomendaciones de uso:

- Crear snapshots antes de cambios importantes (instalaciones, actualizaciones)
- Mantener snapshots de estados conocidos y estables
- Combinar con backups de archivos para protección integral

## 6. Estrategia de recuperación ante fallos

### Niveles de protección implementados:

#### Nivel 1 - Archivos individuales:

- Backups diarios de base de datos (2:00 AM)
- Respaldo en `/var/backups/sigie/`

#### Nivel 2 - Sistema completo:

- Backup completo diario (3:00 AM)
- Incluye configuraciones de Apache, MariaDB y aplicaciones
- Replicación automática al host Windows

#### Nivel 3 - Máquina virtual completa:

- Snapshots manuales de VirtualBox
- Restauración completa del sistema operativo

### Procedimiento de recuperación:

#### Fallo menor (archivos específicos):

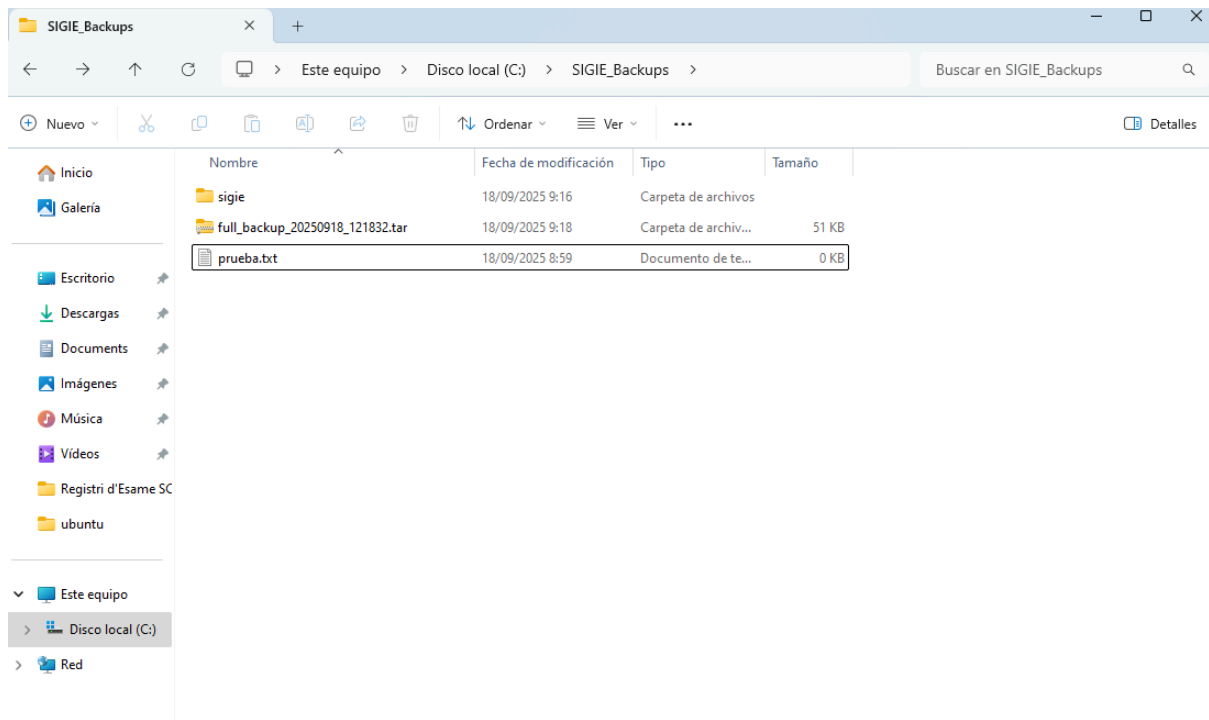
```
# Restaurar desde backup específico  
./restore_full.sh
```

#### Fallo mayor (sistema corrupto):

1. Restaurar snapshot de VirtualBox más reciente
2. Aplicar backups de archivos posteriores al snapshot
3. Verificar integridad de servicios

### Fallo catastrófico (VM destruida):

1. Crear nueva máquina virtual con Ubuntu Server
2. Configurar red y servicios básicos
3. Restaurar desde backup del host Windows
4. Reimportar base de datos



## Resumen de la Segunda Entrega

### Estado final del sistema:

- ☒ **4 usuarios técnicos** creados con sus roles específicos (sysadmin, dbadmin, webadmin, netadmin)
- ☒ **Menús personalizados** para cada rol que se ejecutan automáticamente al login
- ☒ **Menú de administrador general** para sigie-admin con opciones avanzadas
- ☒ **Red configurada** con IP fija 192.168.2.50 en la red del colegio
- ☒ **SSH configurado** y seguro para acceso remoto
- ☒ **Sistema de backup automático** programado diariamente a las 2:00 AM
- ☒ **Apache2 funcionando** sirviendo sitio web con soporte PHP
- ☒ **PHP 8.3 instalado** con todas las extensiones necesarias
- ☒ **MariaDB instalado** con base de datos SIGIE importada y usuario de aplicación
- ☒ **phpMyAdmin funcional** para administración gráfica de base de datos
- ☒ **Script de prueba de BD** verificando conectividad y consultas
- ☒ **Menú maestro** integrando todos los scripts desarrollados
- ☒ **Sistema de backup avanzado** con replicación al host y recuperación automática

## Usuarios y credenciales del sistema:

- **sigie-admin:** Usuario administrador general con sudo
- **sysadmin:** Contraseña inicial Pass1234 (cambio obligatorio) - Grupo sistema
- **dbadmin:** Contraseña inicial Pass1234 (cambio obligatorio) - Grupo bd
- **webadmin:** Contraseña inicial Pass1234 (cambio obligatorio) - Grupo web
- **netadmin:** Contraseña inicial Pass1234 (cambio obligatorio) - Grupo red
- **Usuario de BD:** sigie\_user / sigie2025 - Permisos completos en base sigie

## IP y conectividad:

- **Servidor:** 192.168.2.50/24
- **Gateway:** 192.168.2.1
- **Acceso web:** http://192.168.2.50
- **Acceso phpMyAdmin:** http://192.168.2.50/phpmyadmin
- **Acceso SSH:** ssh usuario@192.168.2.50

## Servicios automatizados:

- **Backup de base de datos:** Diariamente a las 2:00 AM (sysadmin)
- **Backup completo del sistema:** Diariamente a las 3:00 AM (sigie-admin)
- **Replicación de backups:** Automática al host Windows vía carpeta compartida
- **Arranque automático:** Todos los servicios habilitados (Apache, MariaDB, SSH)

La segunda entrega proporciona una base sólida y segura para el despliegue posterior de la aplicación SIGIE completa, incluyendo un sistema robusto de protección de datos y recuperación ante fallos.