

a) Estudio de los diferentes roles de los usuarios del sistema

En el marco del sistema **SIGIE (Sistema Informático de Gestión Institucional Educativa)** se identifican los siguientes **roles de usuarios del sistema operativo** y de la **aplicación web**.

Roles del sistema operativo (infraestructura)

1. Administrador del sistema (root / superusuario)

- Tiene el control total del servidor.
- Se encarga de instalar y mantener los paquetes, gestionar los usuarios y permisos, configurar la red y realizar copias de seguridad.
- Un error en este rol puede comprometer la disponibilidad de todo el sistema.
- Medida de seguridad: acceso restringido a pocas personas y uso de **sudo** con registro.

2. Administrador de base de datos

- Gestiona el motor de base de datos (MySQL/MariaDB).
- Crea las bases de datos y los usuarios de la aplicación.
- Administra respaldos y recuperaciones en caso de pérdida de datos.

3. Administrador web

- Responsable de la configuración del servidor web (Apache/Nginx).
- Define los hosts virtuales, certificados SSL y despliegue de la aplicación SIGIE.

4. Administrador de red

- Configura y mantiene los servicios de red, cortafuegos (UFW), políticas de seguridad y monitoreo de tráfico.
- Asegura que los servicios del sistema sean accesibles de forma controlada y segura.

Roles de la aplicación (usuarios finales)

1. Administrador (app)

- Crea y gestiona usuarios especiales (coordinadores, gestores de contenido).
- Supervisa la aplicación a nivel funcional.

2. Coordinador

- Crea clases, asigna alumnos y profesores, publica eventos generales o específicos por clase.

3. Profesor

- Visualiza clases, gestiona fechas en el calendario de exámenes/tareas, accede a eventos publicados.

4. Alumno

- Visualiza calendario y eventos.
- Accede a enlaces personales y recursos asignados.

5. Gestor de contenido

- Publica materiales o actividades complementarias.
- No administra usuarios ni clases.

b) Diagrama de navegabilidad e implementación de shell script

Diagrama de navegabilidad



Script Bash: create_users_groups.sh

```
#!/usr/bin/env bash
# Script para crear usuarios y grupos de SIGIE
# Ejecutar con sudo

set -euo pipefail

GROUPS=(administrador coordinadores profesores alumnos gestor)

echo "==> Creando grupos..."
for g in "${GROUPS[@]}; do
    if getent group "$g" > /dev/null; then
        echo "Grupo $g ya existe"
    else
        groupadd "$g"
        echo "Grupo $g creado"
    fi
done

USERS=(
    "sigie_admin:administrador"
    "coord1:coordinadores"
    "prof1:profesores"
    "alum1:alumnos"
    "gest1:gestor"
)

DEFAULT_PASS="Pass1234"

echo "==> Creando usuarios..."
for entry in "${USERS[@]}; do
    IFS=":" read -r user group <<< "$entry"
    if id "$user" &>/dev/null; then
        echo "Usuario $user ya existe"
    else
        useradd -m -s /bin/bash -G "$group" "$user"
        echo "$user:$DEFAULT_PASS" | chpasswd
        chage -d 0 "$user"
        echo "Usuario $user creado en grupo $group (contraseña temporal:
$DEFAULT_PASS)"
    fi
done
```

done

echo "Usuarios y grupos configurados correctamente."

c) Relevamiento y justificación del Sistema Operativo

Para la implementación del proyecto SIGIE se evaluaron dos sistemas operativos posibles para el **servidor** y para las **terminales de usuario**:

Alternativa 1: Debian 12 (Bookworm)

- **Ventajas:**
 - Máxima estabilidad y seguridad.
 - Ciclo de soporte largo (5 años).
 - Gran disponibilidad de paquetes.
- **Desventajas:**
 - Versiones más antiguas de software, lo que puede limitar nuevas características.
 - Menos amigable para principiantes en comparación con Ubuntu.

Alternativa 2: Ubuntu 22.04 LTS

- **Ventajas:**
 - Basado en Debian pero con mayor facilidad de uso.
 - Amplia documentación y comunidad activa.
 - Versiones más actualizadas de software.
 - Integración directa con servidores web, PHP y MySQL/MariaDB.
 - Excelente soporte en VirtualBox para exportar a `.ova`.
- **Desventajas:**

- Ciclo de vida algo más corto (5 años vs. Debian 5+).
- Requiere más recursos de hardware en desktop.

Decisión final

Se elige **Ubuntu 22.04 LTS** tanto para el **servidor** como para las **terminales de usuario**:

- **Servidor:** Ubuntu Server 22.04 LTS.
 - Justificación: balancea estabilidad y facilidad de configuración. La comunidad educativa tiene más acceso a documentación, lo que reduce tiempos de soporte. Compatible con LAMP/LEMP stack.
- **Terminales de usuario (alumnos y docentes):** Ubuntu Desktop 22.04 LTS.
 - Justificación: entorno gráfico intuitivo, soporte a largo plazo, disponibilidad de aplicaciones educativas.

👉 Comparado con Debian, Ubuntu ofrece la misma base estable pero con un ecosistema más amigable y actualizado, lo que lo convierte en la mejor opción para un proyecto académico como SIGIE.

Informe Técnico – Segunda Entrega SIGIE

Introducción

El proyecto **SIGIE** (Sistema de Gestión Institucional Escolar) requiere un entorno de servidor seguro, confiable y escalable. En esta entrega configuramos un **servidor en Ubuntu 24.04 dentro de VirtualBox**, implementando usuarios, red, servicio web, base de datos, backups automáticos y herramientas administrativas.

Cada paso fue documentado con scripts y configuraciones, asegurando que el sistema pueda ser mantenido y escalado en el futuro.

a) Creación de usuarios y grupos

El primer paso fue estructurar el sistema con **usuarios y grupos diferenciados**.

La idea es que cada rol tenga los permisos justos y necesarios para cumplir su función, siguiendo el principio de **mínimo privilegio**.

Se crearon los siguientes usuarios principales:


- **sigie-admin** → usuario administrador principal, gestiona todo el servidor.
- **webadmin** → gestiona únicamente el servicio web y archivos de Apache.
- **dbadmin** → gestiona la base de datos MariaDB/MySQL.
- **netadmin** → se encarga de la red y configuración de interfaces.

Grupos creados:

- **sistema** → gestión global.
- **web** → administración del servidor web.
- **db** → base de datos.
- **red** → configuración de red.

Esto se implementó con scripts (`create_users_groups.sh`) que garantizan consistencia al reinstalar el servidor.

 [Captura N°1: ejecución del script de creación de usuarios y grupos]

 [Captura N°2: comprobación en `/etc/passwd` y `/etc/group`]

b) Script de roles y menús de administración

Para facilitar la administración, se implementaron **scripts en Bash** con menús interactivos:

- `create_roles.sh` → define roles y permisos.
- `menu.sh` → menú principal con acceso a funciones de administración.
- `menu_user.sh` → menú simplificado para usuarios normales.

El objetivo es que incluso alguien sin experiencia en Linux pueda usar el sistema de manera segura. Esto también estandariza procesos y evita errores humanos.

 [Captura N°3: pantalla del menú principal en ejecución]

c) Configuración de red


El servidor fue configurado con **IP estática**, lo que permite que siempre sea accesible en la misma dirección.

Configuración usada:

- IP: `192.168.2.50`
- Máscara: `255.255.255.0`
- Gateway: `192.168.2.1`
- DNS: `8.8.8.8` y `1.1.1.1`

Esto asegura estabilidad en los servicios (Apache, SSH, MariaDB) y es fundamental para entornos de producción.

 [Captura N°4: archivo `/etc/netplan/01-netcfg.yaml`]

 [Captura N°5: prueba de conectividad con `ping 8.8.8.8`]

d) Comprobación de servicios en ejecución

Se verificó que los servicios críticos del sistema estén activos:

- **SSH** → conexión remota.
- **Apache2** → servicio web.
- **MariaDB** → motor de base de datos.

Se usaron comandos:

```
systemctl status ssh
systemctl status apache2
systemctl status mariadb
```

Esto garantiza que el sistema arranque siempre con los servicios listos.

 [Captura N°6: salida de `systemctl status apache2`]

 [Captura N°7: salida de `systemctl status mariadb`]

e) Copias de seguridad (Backups)

Uno de los aspectos más importantes es la **seguridad de la información**. Se diseñó un esquema de backups en dos niveles:


1. **Backup parcial** (`backup_sigie.sh`): copia de archivos críticos de configuración y directorios clave.
2. **Backup completo** (`backup_full.sh`): copia del sistema web, base de datos, configuraciones y scripts administrativos.

Los backups se almacenan en:

- **Dentro de la VM:** `/var/backups/sigie/`

- **En la PC host:** carpeta compartida **SIGIE_Backups** mediante VirtualBox.

Esto garantiza redundancia: incluso si la VM se rompe, la copia sigue accesible en la máquina host.

 [Captura N°8: ejecución del script de backup]

 [Captura N°9: listado de archivos **.tar.gz** en **/var/backups/sigie**]

 [Captura N°10: carpeta **SIGIE_Backups** en Windows con los backups]

f) Automatización de backups con Cron

Los scripts de backup se integraron con **cron** para ejecutarse automáticamente:

- **backup_sigie.sh** → todos los días a las **2 AM**.
- **backup_full.sh** → todos los días a las **3 AM**.

Esto significa que **los backups se hacen solos, sin intervención manual**, lo cual es crítico en entornos de producción.

 [Captura N°11: crontab con las rutinas de backup configuradas]


g) Servicio web (Apache)

Se instaló y configuró **Apache2** como servidor web.

- Archivo principal: **/var/www/html/index.html**.
- Página personalizada con el mensaje: **“SIGIE – Servidor Web Activo”**.

Esto permite verificar que el servidor responde correctamente desde cualquier navegador de la red en:

 **http://192.168.2.50**

 [Captura N°12: navegador mostrando la página de prueba “SIGIE – Servidor Web Activo”]

h) Base de datos (MariaDB/MySQL)

El motor de base de datos seleccionado fue **MariaDB 10.11**.

Pasos realizados:

1. Instalación y activación de MariaDB.
2. Ejecución de `mysql_secure_installation` para reforzar la seguridad.
3. Creación de la base de datos `sigie`.
4. Importación del archivo `sigie.sql` (contiene tablas: `usuarios`, `clases`, `eventos`, etc.).
5. Creación del usuario dedicado `sigie_user` con contraseña `sigie2025`, que tiene permisos sobre la base.

👉 De esta forma, la aplicación SIGIE se conectará con un usuario seguro, sin usar el root.

📸 [Captura N°13: comando de creación de la base de datos]

📸 [Captura N°14: tablas listadas en `SHOW TABLES;`]

📸 [Captura N°15: usuario `sigie_user` en la base de datos]

i) Script de conexión a la base de datos

Se creó un script (`db_test.sh`) que conecta a la base y ejecuta consultas de prueba. Esto permite verificar que la conexión funciona y que los datos se almacenan correctamente.

El script:

- Muestra cantidad de usuarios.
- Lista clases creadas.
- Muestra eventos cargados.


 [Captura N°16: ejecución de `db_test.sh` mostrando resultados en consola]

j) Menú de administración

Se desarrolló un **menú integrador** que conecta todas las funcionalidades previas:

- Gestión de usuarios.
- Configuración de red.
- Backups.
- Consultas a la base de datos.

De esta manera, el administrador solo necesita ejecutar un único script y podrá acceder a todas las herramientas.


 [Captura N°17: menú general funcionando en consola]

Extras añadidos (mejoras del sistema)

Instalación de PHP y extensiones

Se instaló PHP con las extensiones necesarias (`php-mysql`, `php-xml`, `php-mbstring`, `php-curl`).

Esto asegura compatibilidad con aplicaciones modernas (como el futuro sitio SIGIE).

 [Captura N°18: `info.php` mostrando la configuración de PHP en el navegador]

Instalación de phpMyAdmin

Se instaló **phpMyAdmin** para administrar la base de datos vía web.


Acceso:


 <http://192.168.2.50/phpmyadmin>

Usuarios disponibles:

- `root` → administración total.

- **sigie_user** → administración limitada para la aplicación.

 [Captura N°19: pantalla de login de phpMyAdmin]

 [Captura N°20: vista de la base de datos SIGIE en phpMyAdmin]

Backups externos con VirtualBox Shared Folders

Se configuró la carpeta compartida **SIGIE_Backups**, montada en **/mnt/host_backups**.

Esto asegura que cada backup hecho en la VM también queda en la PC host, aumentando la resiliencia ante fallos.

 [Captura N°21: carpeta **SIGIE_Backups** vista en Windows con archivos de backup]

Seguridad y mantenimiento

- Firewall UFW con puertos abiertos solo para **22 (SSH)**, **80 (HTTP)** y más adelante **443 (HTTPS)**.
 - Backups automáticos.
 - Snapshots de VirtualBox como capa extra de recuperación.
 - Preparación para instalar certificados SSL con Let's Encrypt.
-

Conclusión

El servidor SIGIE quedó **funcional, seguro y preparado para crecer**:

- Gestión de usuarios y roles bien definida.
- Red configurada con IP estática.
- Servicios web y de base de datos activos.
- Backups automáticos y redundantes.
- Acceso a la base de datos desde consola y desde navegador (phpMyAdmin).

- Scripts y menús que simplifican la administración.

👉 En la próxima etapa, se integrará el **sitio web completo de SIGIE** con conexión a la base de datos ya instalada.