

Informe Final - Proyecto AdminSuite

Arquitectura general:

El menú está formado por un conjunto de scripts diseñados en Bash.

El script principal es adminsuite.sh y este abarca todos los submenús.

Cada módulo tiene una función clara e independiente.

Módulo	Función principal
usuarios_grupos.sh	Alta, baja, modificación y búsqueda de usuarios y grupos.
aplicacion_web.sh	Instalación, actualización y respaldo de WordPress utilizando Docker.
respaldos.sh	Crear respaldos locales y restauración manual o programada con cron.
logs_reportes.sh	Consulta de logs del sistema, de la app y exportación de informes.
red.sh	Herramientas de red (ping, nmap, servicios, barrido simple).
sistema.sh	Información del sistema, espacio en disco, servicios activos, actualización del SO.
ayuda.sh	Guía breve del menú.

Entorno de desarrollo y pruebas:

- Distribución utilizada: CentOS 7 (x86_64)
- Privilegios: sudo habilitado
- Dependencias instaladas: Docker, nmap, cron
- Máquina Virtual CentOS 7 con VirtualBox

Todas las pruebas las realizamos dentro de este entorno.

Decisiones técnicas:

- Lenguaje: Bash, ya que es el que hemos estado utilizando.
- Estructura modular: cada menú en archivo separado. Facilita el mantenimiento y permite agregar o modificar funciones sin afectar todo el script general.
- Logs: registro detallado en `/var/log/adminsuite.log` con fecha y acción. Permite revisar las acciones y detectar errores o problemas de manera rápida y segura.
- Aplicación web: WordPress con Docker, instalable y actualizable. Garantiza que la instalación sea más sencilla y fácil de actualizar o mover a otro sistema.
- Respaldos: uso de `tar` y `cron`. Automatiza copias de seguridad programadas, asegurando que los datos se puedan recuperar en caso de fallo o de necesitarlo.
- Seguridad: principio de mínimo privilegio y sin exponer contraseñas. Reduce riesgos de seguridad al limitar permisos y proteger credenciales que pueden ser sensibles.

Resultados de pruebas:

Menú principal:

```
*-*-*-*-* Admin Suite *-*-*-*-*  
1 - Usuarios y Grupos  
2 - Aplicación Web  
3 - RespalDOS Programados  
4 - Logs y Reportes  
5 - Herramientas de Red  
6 - Sistema  
7 - Ayuda  
0 - Salir  
*-*-*-*-*  
Ingrese una opción:
```

Herramientas de Red:

```
*-*-* Herramientas de Red *-*-*  
1 - Ping a host  
2 - Barrido simple de hosts activos  
3 - Servicios locales  
4 - Escaneo básico (nmap)  
5 - Informe de red  
0 - Volver  
Ingrese una opción:
```

Usuarios y Grupos:

```
*-*-* Usuarios y Grupos *-*-*  
1 - Crear usuario  
2 - Eliminar usuario  
3 - Modificar usuario  
4 - Agregar/Quitar a grupo  
5 - Listar usuarios y grupos  
6 - Buscar por patrón  
7 - Exportar reporte con fecha  
0 - Volver  
Ingrese una opción:
```

Sistema:

```
*-*-* Sistema *-*-*  
1 - Información del sistema  
2 - Espacio en disco  
3 - Servicios activos  
4 - Actualizar SO  
0 - Volver  
Ingrese una opción:
```

Aplicación Web:

```
*-*-* Aplicación Web (WordPress) *-*-*  
1 - Instalar (Docker)  
2 - Actualizar  
3 - Ver estado  
4 - Backup/Restore solo app  
0 - Volver  
Ingrese una opción:
```

Logs y Reportes:

```
*-*-* Logs y Reportes *-*-*  
1 - Ver logs de la app/SO  
2 - Buscar por patrón  
3 - Errores recientes  
4 - Exportar informe de errores (txt)  
0 - Volver  
Ingrese una opción:
```

Ayuda:

```
*-*-* Ayuda de AdminSuite *-*-*  
Este script automatiza tareas de administración para Linux.  
Menús disponibles:  
1. Usuarios y Grupos  
2. Aplicación Web (WordPress)  
3. RespalDOS Programados  
4. Logs y Reportes  
5. Herramientas de Red  
6. Sistema  
Las acciones quedan registradas en: /var/log/adminsuite.log  
Presione cualquier tecla para salir.
```

Riesgos y consideraciones

- La primera instalación de Docker puede demorar por la descarga de la imagen.
- El espacio disponible puede llegar a afectar a los respaldos.
- Los logs pueden crecer rápidamente: se recomienda limpieza periódica.

Mejoras futuras

- Soporte para respaldo de base de datos MySQL de WordPress.
- Automatización de actualizaciones del Sistema Operativo/Contenedor.