

Fase 1: Creación y administración de usuarios y grupos

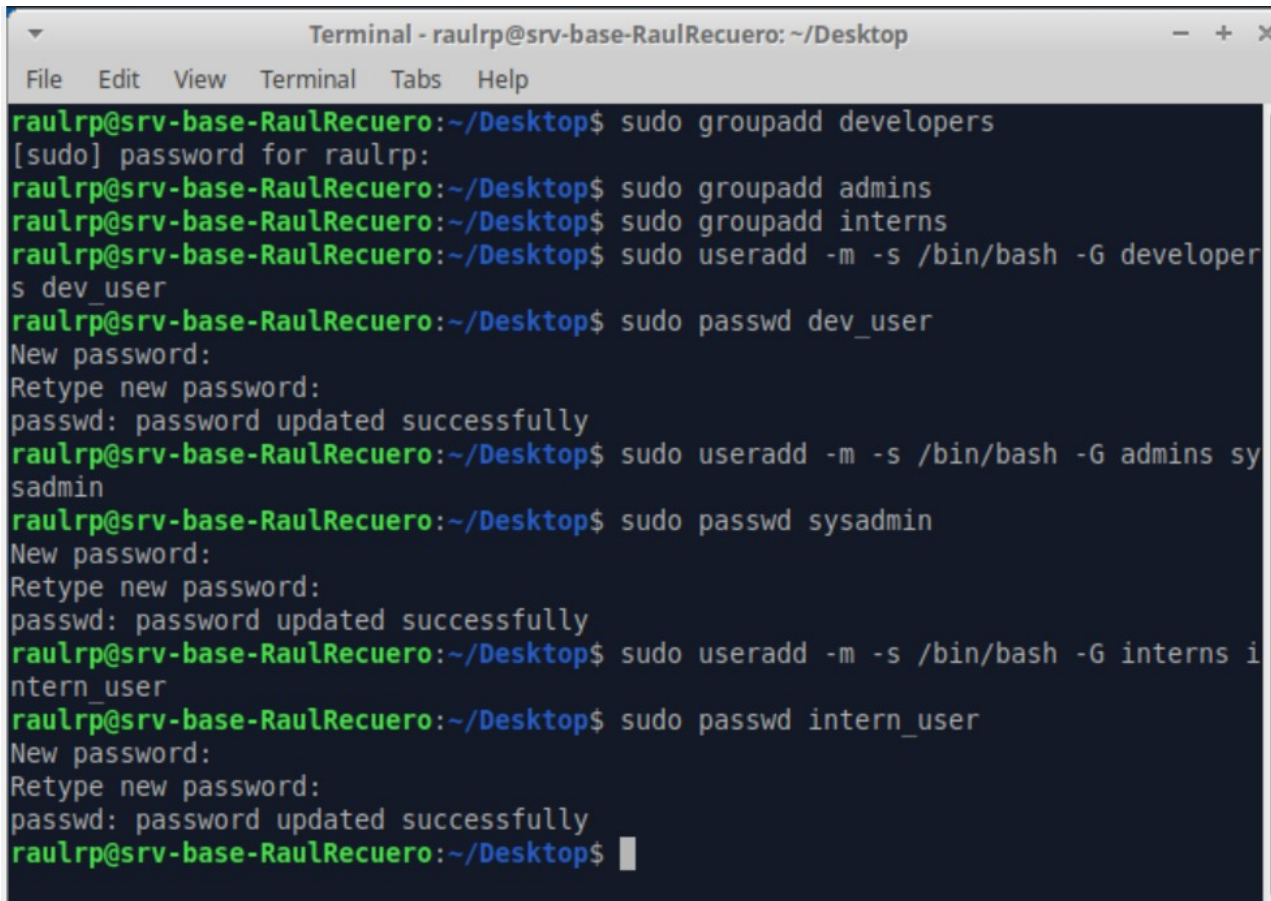
✓ Crear tres usuarios con diferentes roles:

- dev_user (Desarrollador con acceso a directorios específicos).
- sysadmin (Administrador con permisos avanzados).
- intern_user (Usuario con permisos limitados).

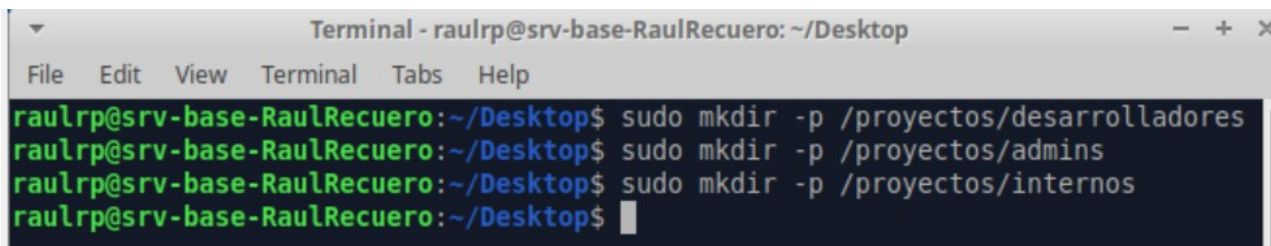
✓ Asignar cada usuario a un grupo específico (developers, admins, interns).

✓ Configurar permisos de acceso a carpetas para cada grupo.

Primero creamos los grupos, los usuarios y los agregamos a sus grupos:



```
Terminal - raulrp@srv-base-RaulRecuero: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo groupadd developers
[sudo] password for raulrp:
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo groupadd admins
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo groupadd interns
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo useradd -m -s /bin/bash -G developers dev_user
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo passwd dev_user
New password:
Retype new password:
passwd: password updated successfully
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo useradd -m -s /bin/bash -G admins sysadmin
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo passwd sysadmin
New password:
Retype new password:
passwd: password updated successfully
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo useradd -m -s /bin/bash -G interns intern_user
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo passwd intern_user
New password:
Retype new password:
passwd: password updated successfully
raulrp@srv-base-RaulRecuero:~/Desktop$
```



```
Terminal - raulrp@srv-base-RaulRecuero: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo mkdir -p /proyectos/desarrolladores
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo mkdir -p /proyectos/admins
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo mkdir -p /proyectos/internos
raulrp@srv-base-RaulRecuero:~/Desktop$
```

```
Terminal - raulrp@srv-base-RaulRecuero: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chown :developers /proyectos/desarrolladores
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chown :admins /proyectos/admins
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chown :interns /proyectos/internos
raulrp@srv-base-RaulRecuero:~/Desktop$
```

Ahora los permisos y nos aseguramos que los nuevos archivos hereden el grupo:

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chmod 770 /proyectos/desarrolladores
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chmod 770 /proyectos/admins
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chmod 770 /proyectos/internos
```

```
Terminal - raulrp@srv-base-RaulRecuero: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chmod g+s /proyectos/desarrolladores
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chmod g+s /proyectos/admins
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chmod g+s /proyectos/internos
```

Fase 2: Configuración de tareas automatizadas

- ✓ Programar una tarea en cron para realizar un backup automático de un directorio.
- ✓ Establecer un script que envíe notificaciones de actividad del servidor.

```
Terminal - raulrp@srv-base-...
Terminal - raulrp@srv-base-RaulRecuero: ~/Desktop
File Edit View Terminal Tabs Help
GNU nano 7.2 /user/local/bin/backup_directorio.sh *
#!/bin/bash

ORIGEN="/proyectos/desarrolladores"
DESTINO="/backups"

mkdir -p "$DESTINO"

FECHA=$(date +%F_%H-%M)
TARFILE="$DESTINO/backup_$FECHA.tar.gz"

tar -czf "$TARFILE" "$ORIGEN"

find "$DESTINO" -type f -name "*.tar.gz" -mtime +7 -delete

sudo chmod +x /usr/local/bin/backup_directorio.sh
```

```
Terminal - raulrp@srv-base-RaulRecuero: /
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:/$ sudo chmod +x /usr/local/bin/backup_directorio.sh
```

```
Terminal - raulrp@srv-base-RaulRecuero: /
File Edit View Terminal Tabs Help
GNU nano 7.2 crontab *
0 2 * * * /usr/local/bin/backup_directorio.sh
```

```
Menú de aplicaciones Terminal - raulrp@srv-base-RaulRecuero: /
File Edit View Terminal Tabs Help
GNU nano 7.2 /usr/local/bin/notificacion_actividad.sh
#!/bin/bash

LOG="/var/log/actividad_servidor.log"
FECHA=$(date "+%F %t")

USUARIOS=$(who)
USO_DISCO=$(df -h / | tail -1)
USO_CPU=$(top -bn1 | grep "Cpu(s)" | awk '{print $2 + $4}') # uso aproximado

echo "[$FECHA] --- Actividad del servidor ---" >> $LOG
echo "Usuarios conectados: " >> $LOG
echo "$USUARIOS" >> $LOG
echo "Uso de disco: $USO_DISCO" >> $LOG
echo "Uso de CPU: $USO_CPU%" >> $LOG
echo "-----" >> $LOG

raulrp@srv-base-RaulRecuero:/$ sudo chmod +x /usr/local/bin/notificacion_actividad.sh
[sudo] password for raulrp:
```

```
Terminal - raulrp@srv-base-RaulRecuero: /
File Edit View Terminal Tabs Help
GNU nano 7.2 /tmp/crontab.gWlG6t/crontab *
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow command
0 2 * * * /usr/local/bin/backup_directorio.sh
0 * * * * /usr/local/bin/notificacion_actividad.sh
```



```
Terminal - raulrp@srv-base-...
Terminal - raulrp@srv-base-RaulRecuero: /
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:/$ sudo systemctl status cron
[sudo] password for raulrp:
● cron.service - Regular background program processing daemon
   Loaded: loaded (/usr/lib/systemd/system/cron.service; enabled; preset: ena
   Active: active (running) since Fri 2025-06-20 14:18:22 CEST; 1h 57min ago
     Docs: man:cron(8)
   Main PID: 1244 (cron)
      Tasks: 1 (limit: 4598)
     Memory: 1.0M (peak: 5.9M)
        CPU: 109ms
    CGroup: /system.slice/cron.service
            └─1244 /usr/sbin/cron -f -P

jun 20 15:55:01 srv-base-RaulRecuero CRON[2859]: pam_unix(cron:session): sessio
jun 20 15:55:01 srv-base-RaulRecuero CRON[2859]: pam_unix(cron:session): sessio
jun 20 16:00:01 srv-base-RaulRecuero CRON[2870]: pam_unix(cron:session): sessio
jun 20 16:00:01 srv-base-RaulRecuero CRON[2870]: pam_unix(cron:session): sessio
jun 20 16:00:02 srv-base-RaulRecuero CRON[2870]: pam_unix(cron:session): sessio
jun 20 16:05:01 srv-base-RaulRecuero CRON[2901]: pam_unix(cron:session): sessio
jun 20 16:05:01 srv-base-RaulRecuero CRON[2901]: pam_unix(cron:session): sessio
jun 20 16:05:01 srv-base-RaulRecuero CRON[2902]: (root) CMD (command -v debian->
jun 20 16:05:01 srv-base-RaulRecuero CRON[2901]: pam_unix(cron:session): sessio
jun 20 16:15:01 srv-base-RaulRecuero CRON[2958]: pam_unix(cron:session): sessio
jun 20 16:15:01 srv-base-RaulRecuero CRON[2958]: pam_unix(cron:session): sessio
lines 1-21/21 (END)
```

```
[2025-06-20      ] --- Actividad del servidor ---
Usuarios conectados:
raulrp  tty7      2025-06-20 14:18 (:0)
Uso de disco: /dev/mapper/ubuntu--vg-ubuntu--lv  24G  11G  12G  48% /
Uso de CPU: 0%
```

Fase 3: Monitoreo y optimización del servidor

- ✓ Identificar procesos en ejecución y ajustar su prioridad si es necesario.
- ✓ Configurar logs de auditoría para registrar accesos y acciones de usuarios.

Hay diferentes comandos para identificar procesos:

top:

```
Terminal - raulrp@srv-base-...
Terminal - raulrp@srv-base-RaulRecuero: ~/Desktop
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:~/Desktop$ top

top - 15:00:38 up 2 min, 1 user, load average: 0,30, 0,26, 0,10
Tasks: 192 total, 1 running, 191 sleeping, 0 stopped, 0 zombie
%Cpu(s):  0,0 us,  0,0 sy,  0,0 ni,100,0 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
MiB Mem :  3916,1 total,  2580,2 free,   893,1 used,   665,5 buff/cache
MiB Swap:  3916,0 total,  3916,0 free,    0,0 used.  3023,0 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
   373 root        rt   0  289116  27264  8704  S   0,3   0,7   0:00.03 multipa+
     1 root         20   0  22996  14076  9468  S   0,0   0,4   0:01.33 systemd
     2 root         20   0      0      0      0  S   0,0   0,0   0:00.00 kthreadd
     3 root         20   0      0      0      0  S   0,0   0,0   0:00.00 pool_wo+
     4 root          0 -20      0      0      0  I   0,0   0,0   0:00.00 kworker+
     5 root          0 -20      0      0      0  I   0,0   0,0   0:00.00 kworker+
     6 root          0 -20      0      0      0  I   0,0   0,0   0:00.00 kworker+
     7 root          0 -20      0      0      0  I   0,0   0,0   0:00.00 kworker+
     8 root         20   0      0      0      0  I   0,0   0,0   0:00.00 kworker+
     9 root         20   0      0      0      0  I   0,0   0,0   0:00.10 kworker+
```

htop:

```
Terminal - raulrp@srv-base-...
Terminal - raulrp@srv-base-RaulRecuero: ~/Desktop
File Edit View Terminal Tabs Help

0.7% Tasks: 103, 329 thr, 89 kthr; 1 runnin
0.7% Load average: 0.25 0.26 0.11
Mem[|||||] 648M/3.82G Uptime: 00:03:50
Swp[|] 0K/3.82G

Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
2217 raulrp 20 0 8604 4736 3584 R 2.0 0.1 0:00.33 htop
1320 root 20 0 350M 99860 54584 S 1.3 2.5 0:01.59 /usr/lib/xorg
2201 raulrp 20 0 594M 41236 32896 S 0.7 1.0 0:00.19 /usr/bin/xfce
1 root 20 0 22996 14076 9468 S 0.0 0.4 0:01.33 /sbin/init
321 root 19 -1 42508 16900 15620 S 0.0 0.4 0:00.16 /usr/lib/syst
373 root RT 0 282M 27264 8704 S 0.0 0.7 0:00.01 /sbin/multipa
387 root 20 0 282M 27264 8704 S 0.0 0.7 0:00.00 /sbin/multipa
388 root RT 0 282M 27264 8704 S 0.0 0.7 0:00.00 /sbin/multipa
389 root RT 0 282M 27264 8704 S 0.0 0.7 0:00.00 /sbin/multipa
390 root RT 0 282M 27264 8704 S 0.0 0.7 0:00.00 /sbin/multipa
391 root RT 0 282M 27264 8704 S 0.0 0.7 0:00.00 /sbin/multipa
392 root RT 0 282M 27264 8704 S 0.0 0.7 0:00.00 /sbin/multipa
```

ps:

```
Terminal - raulrp@srv-base-RaulRecuero
File Edit View Terminal Tabs Help

raulrp@srv-base-RaulRecuero:~/Desktop$ ps
  PID TTY          TIME CMD
 2235 pts/0        00:00:00 bash
 2241 pts/0        00:00:00 ps
raulrp@srv-base-RaulRecuero:~/Desktop$
```

Con el siguiente comando se pueden ver en más detalle:

```
raulrp@srv-base-RaulRecuero:~/Desktop$ ps aux --sort=-%cpu | head -10

raulrp@srv-base-RaulRecuero:~/Desktop$ ps aux --sort=-%cpu | head -10
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      1320  0.7  2.4 359232 99796 tty7     Ssl+  14:57   0:03 /usr/lib/xorg
/Xorg -core :0 -seat seat0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt7 -nov
tswitch
raulrp    1710  0.5  0.7 344364 31244 ?        Sl     14:58   0:02 /usr/libexec/
ibus-extension-gtk3
raulrp    2225  0.3  1.0 608264 41304 ?        Sl     15:02   0:00 /usr/bin/xfce
4-terminal
root       1  0.2  0.3 22996 14204 ?        Ss     14:57   0:01 /sbin/init
root      961  0.2  0.9 1848452 38404 ?        Ssl    14:57   0:01 /usr/lib/snap
d/snapd
raulrp    1826  0.1  1.1 425680 44792 ?        Sl     14:58   0:00 xfdesktop
raulrp    1763  0.1  2.6 950848 106876 ?        Sl     14:58   0:00 xfwm4 --repla
```

Para cambiar la prioridad al iniciar un proceso:

`nice -n 10 nombre_del_comando`

Y para cambiar la prioridad de un proceso que se está ejecutando:

`sudo renice -n 5 -p <PID>`

Para configurar los logs, primero debemos instalar y configurar auditd:

sudo apt update

sudo apt install auditd audispd-plugins

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo apt install auditd audispd-plugins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libauparse0t64
The following NEW packages will be installed:
  audispd-plugins auditd libauparse0t64
0 upgraded, 3 newly installed, 0 to remove and 14 not upgraded.
Need to get 312 kB of archives.
After this operation, 1.024 kB of additional disk space will be used.
```

Activamos y verificamos el servicio:

sudo systemctl enable auditd

sudo systemctl start auditd

sudo systemctl status auditd

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl enable auditd
Synchronizing state of auditd.service with SysV service script with /usr/lib/sys
temd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable auditd
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl start auditd
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl status auditd
● auditd.service - Security Auditing Service
   Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled; preset: e>
   Active: active (running) since Mon 2025-06-23 15:15:35 CEST; 3min 32s ago
     Docs: man:auditd(8)
           https://github.com/linux-audit/audit-documentation
  Main PID: 2964 (auditd)
    Tasks: 2 (limit: 4598)
  Memory: 492.0K (peak: 1.9M)
     CPU: 18ms
    CGroup: /system.slice/auditd.service
            └─2964 /sbin/auditd
```

Para registrar actividades, como registrar todos los comandos de un usuario:

sudo auditctl -a always,exit -F arch=b64 -F euid=1001 -S execve -k devuser-cmd

1001 se cambia por el UID real de en este caso como ejemplo, dev_user y se puede ver con:
id dev_user

```
raulrp@srv-base-RaulRecuero:~/Desktop$ id dev_user
uid=1006(dev_user) gid=1009(dev_user) groups=1009(dev_user),1006(developers)
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo auditctl -a always,exit -F arch=b64
-F euid=1006 -S execve -k devuser-cmd
[sudo] password for raulrp:
raulrp@srv-base-RaulRecuero:~/Desktop$
```


Para auditar accesos a un archivo o directorio específico:

```
sudo auditctl -w /proyectos/desarrolladores -p rwx -k acceso-dev
```

Para ver los logs de auditoría:

```
sudo ausearch -k devuser-cmd
```

```
sudo ausearch -k acceso-dev
```

O también se pueden revisar con:

```
sudo less /var/log/audit/audit.log
```

```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo auditctl -w /proyectos/desarrolladores -p rwx -k acceso-dev
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo ausearch -k devuser-cmd
----
time->Mon Jun 23 15:24:30 2025
type=PROCTITLE msg=audit(1750685070.181:296): proctitle=617564697463746C002D6100616C776179732C65786974002D46006172636800623634002D4600657569640031303036002D5300657865637665002D6B00646576757365722D636D64
type=SYSCALL msg=audit(1750685070.181:296): arch=c000003e syscall=44 success=yes exit=1068 a0=4 a1=7ffd57bc8bd0 a2=42c a3=0 items=0 ppid=9676 pid=9677 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1 ses=4294967295 comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1750685070.181:296): auid=4294967295 ses=4294967295 subj=unconfined op=add_rule key="devuser-cmd" list=4 res=1
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo ausearch -k acceso-dev
----
time->Mon Jun 23 15:32:11 2025
type=PROCTITLE msg=audit(1750685531.632:321): proctitle=617564697463746C002D77002F70726F796563746F732F6465736172726F6C6C61646F726573002D700072777861002D6B0061636365736F2D646576
type=PATH msg=audit(1750685531.632:321): item=0 name="/proyectos/desarrolladores" inode=652804 dev=fc:00 mode=042770 ouid=0 ogid=1006 rdev=00:00 nametype=NORMAL cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1750685531.632:321): cwd="/home/raulrp/Desktop"
```

Fase 4: Documentación y presentación

- ✓ Crear un informe en Google Docs con los pasos de configuración.
- ✓ Explicar en Google Slides la importancia de la administración de usuarios y seguridad en servidores Linux.
- ✓ Justificar cómo esta configuración mejora la gestión interna de Codearts Solutions.

Justificación: Mejora de la Gestión Interna en Codearts:

La configuración realizada en las tres fases impacta directamente en la eficiencia, seguridad y organización interna de Codearts Solutions. A continuación se detallan los beneficios por fase:

Fase 1: Gestión de usuarios y permisos:

Seguridad por segmentación: Al separar usuarios por roles (desarrolladores, administradores e internos), se limita el acceso únicamente a los recursos necesarios para cada grupo. Esto reduce el riesgo de errores o accesos no autorizados.

Control granular de permisos: La configuración de permisos por grupo garantiza que la información crítica esté protegida, promoviendo una política de mínimo privilegio.

Escalabilidad organizativa: Esta estructura facilita la incorporación de nuevos empleados, asignándolos fácilmente a grupos con configuraciones predefinidas.

Fase 2: Automatización de tareas:

Backups automáticos: Garantiza la disponibilidad y recuperación de información crítica ante fallos o pérdida de datos, minimizando tiempos de inactividad.

Notificaciones periódicas: El monitoreo automatizado permite a los administradores conocer el estado del sistema en tiempo real, anticiparse a problemas y mantener trazabilidad de eventos importantes.

Fase 3: Monitoreo y optimización:

Gestión eficiente de recursos: Al identificar procesos intensivos y ajustar su prioridad, se optimiza el rendimiento del servidor, asegurando la estabilidad de servicios clave.

Auditoría de acciones: La implementación de auditd permite registrar acciones de usuarios y accesos a archivos sensibles, lo cual fortalece la trazabilidad, la seguridad y el cumplimiento de políticas internas o regulatorias.

Resultado Global:

Esta infraestructura:

Mejora la seguridad mediante control de accesos y registros.

Aumenta la eficiencia operativa gracias a tareas automatizadas y priorización de procesos.

Refuerza la confiabilidad y el control organizacional con respaldo y monitoreo constantes.

Reto Día 8: Gestión de Servicios y Automatización con Systemd en Linux

Fase 1: Análisis de servicios del sistema

- ✓ Listar todos los servicios activos del sistema usando `systemctl list-units --type=service`.
- ✓ Identificar y documentar 3 servicios activos fundamentales (ej.: `ssh`, `cron`, `networking`).
- ✓ Comprobar si el servidor web (instalado el día anterior) está activo, habilitado y funcionando.

Usamos el siguiente comando para obtener todos los servicios actualmente activos:
`systemctl list-units --type=service --state=running`


```

raulrp@srv-base-RaulRecuero:~/Desktop$ sudo auditctl -w /proyectos/desarrolladores
es -p rwx -k acceso-dev
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo ausearch -k devuser-cmd
----
time->Mon Jun 23 15:24:30 2025
type=PROCTITLE msg=audit(1750685070.181:296): proctitle=617564697463746C002D6100
616C776179732C65786974002D46006172636800623634002D4600657569640031303036002D5300
657865637665002D6B00646576757365722D636D64
type=SYSCALL msg=audit(1750685070.181:296): arch=c000003e syscall=44 success=yes
exit=1068 a0=4 a1=7ffd57bc8bd0 a2=42c a3=0 items=0 ppid=9676 pid=9677 auid=4294
967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts1 ses=4294
967295 comm="auditctl" exe="/usr/sbin/auditctl" subj=unconfined key=(null)
type=CONFIG_CHANGE msg=audit(1750685070.181:296): auid=4294967295 ses=4294967295
subj=unconfined op=add_rule key="devuser-cmd" list=4 res=1
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo ausearch -k acceso-dev
----
time->Mon Jun 23 15:32:11 2025
type=PROCTITLE msg=audit(1750685531.632:321): proctitle=617564697463746C002D7700
2F70726F796563746F732F6465736172726F6C6C61646F726573002D700072777861002D6B006163
6365736F2D646576
type=PATH msg=audit(1750685531.632:321): item=0 name="/proyectos/desarrolladores
" inode=652804 dev=fc:00 mode=042770 ouid=0 ogid=1006 rdev=00:00 nametype=NORMAL
cap_fp=0 cap_fi=0 cap_fe=0 cap_fver=0 cap_frootid=0
type=CWD msg=audit(1750685531.632:321): cwd="/home/raulro/Desktop"

```

Identificar y documentar tres servicios activos fundamentales:

ssh.service es OpenBSD Secure Shell server y permite el acceso remoto seguro al sistema a través del protocolo SSH. Es crucial para la administración remota del servidor.

cron.service es Regular background program processing daemon y ejecuta tareas programadas automáticamente. Fundamental para la automatización de tareas.

networking.service es Raise network interfaces y administra y levanta las interfaces de red al iniciar el sistema. Esencial para la conectividad de red.

Ahora vamos a comprobar nuestro servidor web con el comando:

```
sudo systemctl status apache2.service
```

O

```
systemctl is-active apache2
```

```
systemctl is-enabled apache2
```

```

raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl status apache2.service
[sudo] password for raulrp:
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset:
   Active: active (running) since Mon 2025-06-23 15:42:05 CEST; 20min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Process: 1184 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU
 Main PID: 1218 (apache2)
    Tasks: 55 (limit: 4598)
   Memory: 7.6M (peak: 8.1M)
      CPU: 86ms
   CGroup: /system.slice/apache2.service
           └─1218 /usr/sbin/apache2 -k start
             └─1219 /usr/sbin/apache2 -k start
               └─1220 /usr/sbin/apache2 -k start

jun 23 15:42:04 srv-base-RaulRecuero systemd[1]: Starting apache2.service - The
jun 23 15:42:05 srv-base-RaulRecuero apachectl[1213]: AH00558: apache2: Could n
jun 23 15:42:05 srv-base-RaulRecuero systemd[1]: Started apache2.service - The
lines 1-17/17 (END)

```

Detener, reiniciar y habilitar al arranque el servicio web:

Detener:

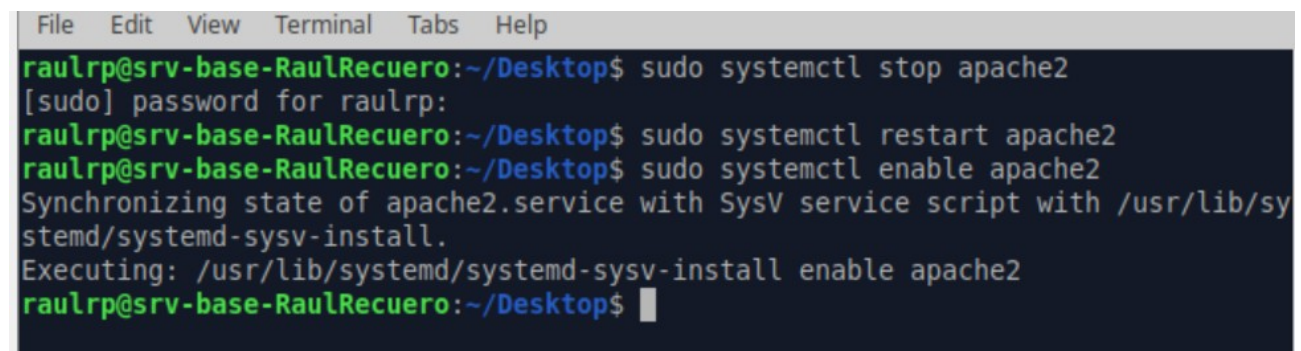
```
sudo systemctl stop apache2
```

Reiniciar:

```
sudo systemctl restart apache2
```

Habilitar al arranque:

```
sudo systemctl enable apache2
```



```
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl stop apache2
[sudo] password for raulrp:
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl restart apache2
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/sy
stemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
raulrp@srv-base-RaulRecuero:~/Desktop$
```

Para modificar la configuración para reinicio automático si falla:

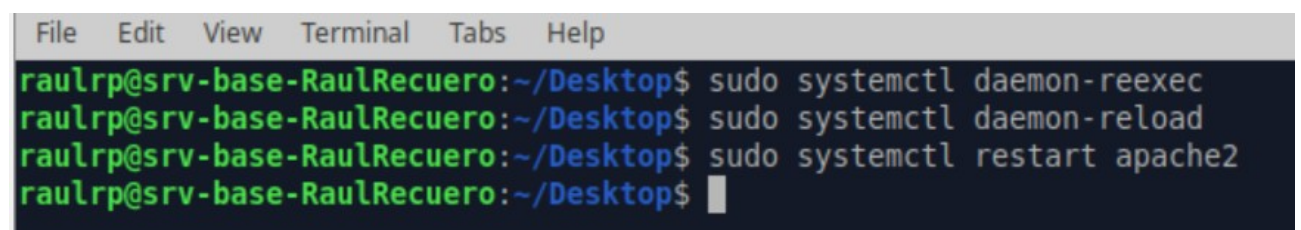
```
sudo systemctl edit apache2
```

En nano escribimos:

```
[Service]
Restart=Always
```

Guardamos y salimos y:

```
sudo systemctl daemon-reexec
sudo systemctl daemon-reload
sudo systemctl restart apache2
```



```
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl daemon-reexec
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl daemon-reload
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl restart apache2
raulrp@srv-base-RaulRecuero:~/Desktop$
```

Crear un alias para reiniciar rápidamente el servicio web:

Hay que editar el archivo .bashrc

```
nano ~/.bashrc
```

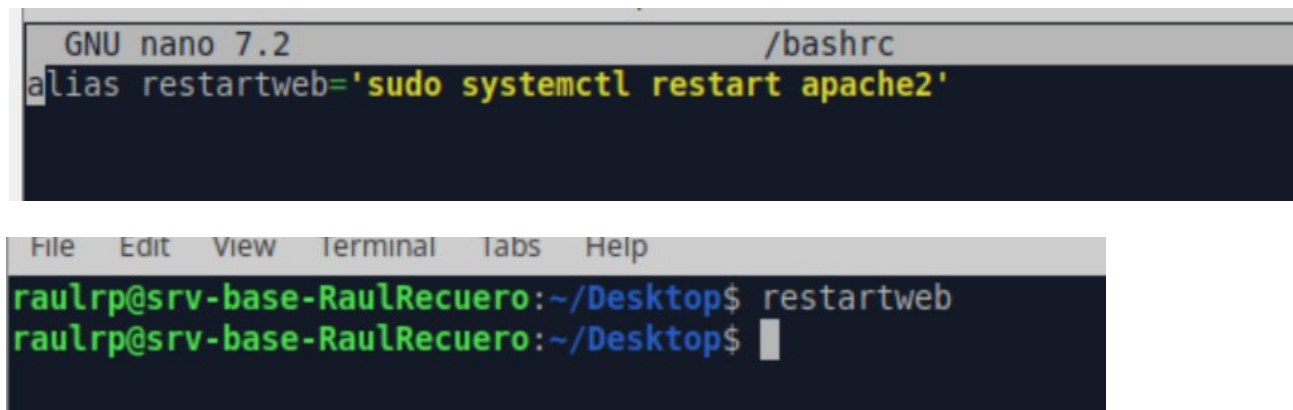
Añadir al final del archivo:

```
alias restartweb='sudo systemctl restart apache2'
```

Aplicamos los cambios en la terminal:

```
source ~/.bashrc
```

Reiniciamos con restartweb



```
GNU nano 7.2 /bashrc
alias restartweb='sudo systemctl restart apache2'
```

```
raulrp@srv-base-RaulRecuero:~/Desktop$ restartweb
raulrp@srv-base-RaulRecuero:~/Desktop$
```

Fase 3: Creación de un servicio personalizado

- ✓ Crear un script Bash llamado saludo.sh que escriba "¡Servidor iniciado correctamente!" en un archivo /var/log/saludo.log.
- ✓ Crear un nuevo servicio de systemd llamado saludo.service que ejecute ese script automáticamente al iniciar el sistema.
- ✓ Comprobar que el servicio:

- Está habilitado
- Se ejecuta al arrancar
- Crea el archivo de log correctamente

Vamos a crear el script que escribirá el mensaje en el log:

```
sudo nano /usr/local/bin/saludo.sh
```

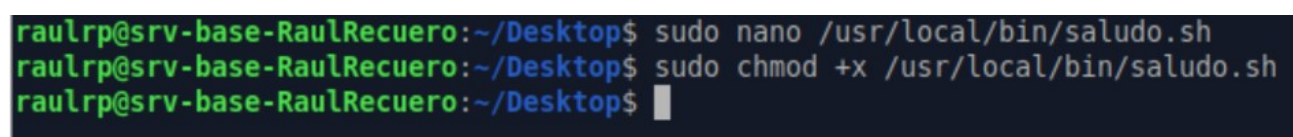
Con el siguiente contenido:

```
#!/bin/bash
```

```
echo "Servidor iniciado correctamente!" » /var/log/saludo.log
```

Damos permisos de ejecución:

```
sudo chmod +x /usr/local/bin/saludo.sh
```



```
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo nano /usr/local/bin/saludo.sh
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo chmod +x /usr/local/bin/saludo.sh
raulrp@srv-base-RaulRecuero:~/Desktop$
```



```
GNU nano 7.2 /usr/local/bin/saludo.sh
#!/bin/bash

echo "Servidor iniciado correctamente!" >> /var/log/saludo.log
```

Creamos el servicio saludo.service:

```
sudo nano /etc/systemd/system/saludo.service
```

[Unit]

Description=Servicio de saludo al iniciar el servidor

After=network.target

[Service]

Type=oneshot

ExecStart=/usr/local/bin/saludo.sh

RemainAfterExit=true

[Install]

WantedBy=multi-user.target

```
raulrp@srv-base-RaulRecuerdo:~/Desktop$ sudo nano /etc/systemd/system/saludo.serv
ice
```

```
Terminal - raulrp@srv-base-RaulRecuerdo: ~/Desktop
File Edit View Terminal Tabs Help
GNU nano 7.2 /etc/systemd/system/saludo.service
[Unit]
Description=Servicio de saludo al iniciar el servidor
After=network.target
[Service]
Type=oneshot
ExecStart=/usr/local/bin/saludo.sh
RemainAfterExit=true
[Install]
WantedBy=multi-user.target
```

Para habilitar el servicio y verificar su funcionamiento, primero recargamos systemd para que reconozca el nuevo servicio:

```
sudo systemctl daemon-reload
```

Habilitamos el servicio para que se inicie automáticamente:

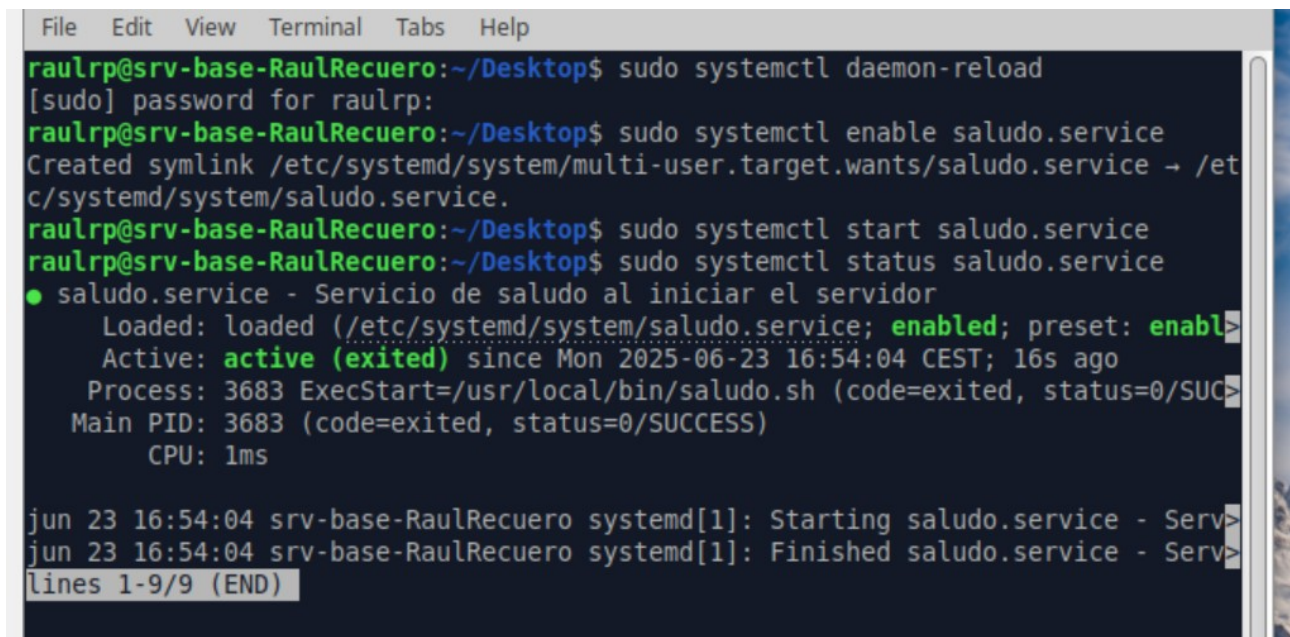
sudo systemctl enable saludo.service

Lo ejecutamos:

sudo systemctl start saludo.service

Comprobamos el estado:

sudo systemctl status saludo.service



```
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl daemon-reload
[sudo] password for raulrp:
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl enable saludo.service
Created symlink /etc/systemd/system/multi-user.target.wants/saludo.service → /etc/systemd/system/saludo.service.
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl start saludo.service
raulrp@srv-base-RaulRecuero:~/Desktop$ sudo systemctl status saludo.service
● saludo.service - Servicio de saludo al iniciar el servidor
   Loaded: loaded (/etc/systemd/system/saludo.service; enabled; preset: enabled)
   Active: active (exited) since Mon 2025-06-23 16:54:04 CEST; 16s ago
     Process: 3683 ExecStart=/usr/local/bin/saludo.sh (code=exited, status=0/SUCCESS)
    Main PID: 3683 (code=exited, status=0/SUCCESS)
       CPU: 1ms

jun 23 16:54:04 srv-base-RaulRecuero systemd[1]: Starting saludo.service - Servicio de saludo al iniciar el servidor:
jun 23 16:54:04 srv-base-RaulRecuero systemd[1]: Finished saludo.service - Servicio de saludo al iniciar el servidor:
lines 1-9/9 (END)
```

Verificación final:

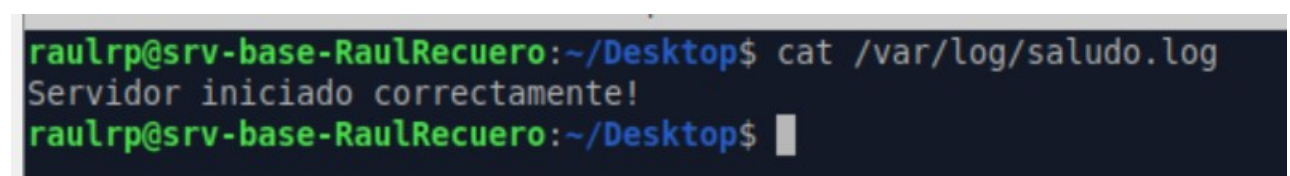
Comprobar si el log se creó:

cat /var/log/saludo.log

Reiniciar:

sudo reboot

Revisar el log nuevamente para confirmar.



```
raulrp@srv-base-RaulRecuero:~/Desktop$ cat /var/log/saludo.log
Servidor iniciado correctamente!
raulrp@srv-base-RaulRecuero:~/Desktop$
```

```
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:~/Desktop$ cat /var/log/saludo.log
Servidor iniciado correctamente!
Servidor iniciado correctamente!
raulrp@srv-base-RaulRecuero:~/Desktop$
```

Fase 4: Monitorización y logs

- ✓ Visualizar los logs de los servicios anteriores con journalctl.
- ✓ Filtrar los mensajes de error o advertencia (journalctl -p 3 -xb).
- ✓ Registrar el estado del servicio saludo y guardar una copia del log en /srv/logs/saludo_journal.log.

Para ver los logs recientes de los servicios anteriores:

```
journalctl -u saludo.service
```

```
journalctl -u apache2.service
```

```
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:~/Desktop$ journalctl -u saludo.service
jun 23 16:54:04 srv-base-RaulRecuero systemd[1]: Starting saludo.service - Serv>
jun 23 16:54:04 srv-base-RaulRecuero systemd[1]: Finished saludo.service - Serv>
jun 23 16:57:37 srv-base-RaulRecuero systemd[1]: saludo.service: Deactivated su>
jun 23 16:57:37 srv-base-RaulRecuero systemd[1]: Stopped saludo.service - Servi>
-- Boot 837b2d8567a741159b051e82d568a7db --
jun 23 16:57:50 srv-base-RaulRecuero systemd[1]: Starting saludo.service - Serv>
jun 23 16:57:50 srv-base-RaulRecuero systemd[1]: Finished saludo.service - Serv>
lines 1-7/7 (END)
```

```
File Edit View Terminal Tabs Help
raulrp@srv-base-RaulRecuero:~/Desktop$ journalctl -u apache2.service
jun 19 16:05:22 srv-base-RaulRecuero systemd[1]: Starting apache2.service - The>
jun 19 16:05:22 srv-base-RaulRecuero apachectl[8714]: AH00558: apache2: Could n>
jun 19 16:05:22 srv-base-RaulRecuero systemd[1]: Started apache2.service - The>
jun 19 16:25:02 srv-base-RaulRecuero systemd[1]: Reloading apache2.service - Th>
jun 19 16:25:02 srv-base-RaulRecuero apachectl[9278]: AH00558: apache2: Could n>
jun 19 16:25:02 srv-base-RaulRecuero systemd[1]: Reloaded apache2.service - The>
jun 19 17:25:28 srv-base-RaulRecuero systemd[1]: Stopping apache2.service - The>
jun 19 17:25:28 srv-base-RaulRecuero apachectl[11606]: AH00558: apache2: Could >
jun 19 17:25:29 srv-base-RaulRecuero systemd[1]: apache2.service: Deactivated s>
jun 19 17:25:29 srv-base-RaulRecuero systemd[1]: Stopped apache2.service - The>
-- Boot e75bdb5c48ae418d92fba1382931b807 --
jun 19 17:25:53 srv-base-RaulRecuero systemd[1]: Starting apache2.service - The>
jun 19 17:25:53 srv-base-RaulRecuero apachectl[1191]: AH00558: apache2: Could n>
jun 19 17:25:53 srv-base-RaulRecuero systemd[1]: Started apache2.service - The>
jun 19 17:26:57 srv-base-RaulRecuero systemd[1]: Reloading apache2.service - Th>
jun 19 17:26:57 srv-base-RaulRecuero apachectl[2142]: AH00558: apache2: Could n>
jun 19 17:26:57 srv-base-RaulRecuero systemd[1]: Reloaded apache2.service - The>
jun 19 17:29:19 srv-base-RaulRecuero systemd[1]: Reloading apache2.service - Th>
jun 19 17:29:19 srv-base-RaulRecuero apachectl[2234]: AH00558: apache2: Could n>
jun 19 17:29:19 srv-base-RaulRecuero systemd[1]: Reloaded apache2.service - The>
-- Boot 2a31c5f9130440a7ab9084ed45f8f782 --
jun 20 14:18:22 srv-base-RaulRecuero systemd[1]: Starting apache2.service - The>
jun 20 14:18:22 srv-base-RaulRecuero apachectl[1261]: AH00558: apache2: Could n>
```


Para ver errores:

```
journalctl -p 3 -xb
```

Para ver advertencias:

```
journalctl -p 4 -xb
```

```
raulrp@srv-base-RaulRecuero:~/Desktop$ journalctl -p 3 -xb
jun 23 16:57:46 srv-base-RaulRecuero systemd[1]: Invalid DMI field header.
jun 23 16:57:47 srv-base-RaulRecuero kernel: vmwgfx 0000:00:02.0: [drm] *ERROR*>
jun 23 16:57:47 srv-base-RaulRecuero kernel: vmwgfx 0000:00:02.0: [drm] *ERROR*>
jun 23 16:57:47 srv-base-RaulRecuero kernel: vmwgfx 0000:00:02.0: [drm] *ERROR*>
jun 23 16:57:51 srv-base-RaulRecuero lightdm[1362]: gkr-pam: couldn't unlock th>
jun 23 16:57:56 srv-base-RaulRecuero lightdm[1493]: gkr-pam: unable to locate d>
lines 1-6/6 (END)
```

```
raulrp@srv-base-RaulRecuero:~/Desktop$ journalctl -p 4 -xb
jun 23 16:57:46 srv-base-RaulRecuero kernel: [Firmware Bug]: TSC doesn't count >
jun 23 16:57:46 srv-base-RaulRecuero kernel: Speculative Return Stack Overflow:>
jun 23 16:57:46 srv-base-RaulRecuero kernel: Speculative Return Stack Overflow:>
jun 23 16:57:46 srv-base-RaulRecuero kernel: APIC calibration not consistent wi>
jun 23 16:57:46 srv-base-RaulRecuero kernel: device-mapper: core: CONFIG_IMA_DI>
jun 23 16:57:46 srv-base-RaulRecuero kernel: platform eisa.0: EISA: Cannot allo>
jun 23 16:57:46 srv-base-RaulRecuero kernel: platform eisa.0: Cannot allocate r>
jun 23 16:57:46 srv-base-RaulRecuero kernel: platform eisa.0: Cannot allocate r>
jun 23 16:57:46 srv-base-RaulRecuero kernel: platform eisa.0: Cannot allocate r>
jun 23 16:57:46 srv-base-RaulRecuero kernel: platform eisa.0: Cannot allocate r>
jun 23 16:57:46 srv-base-RaulRecuero kernel: platform eisa.0: Cannot allocate r>
jun 23 16:57:46 srv-base-RaulRecuero kernel: platform eisa.0: Cannot allocate r>
jun 23 16:57:46 srv-base-RaulRecuero kernel: platform eisa.0: Cannot allocate r>
jun 23 16:57:46 srv-base-RaulRecuero kernel: platform eisa.0: Cannot allocate r>
jun 23 16:57:46 srv-base-RaulRecuero kernel: amd_pstate: the _CPC object is not>
```

Registrar el estado el servicio saludo y guardar copia del log:

```
sudo mkdir -p /srv/logs
```

Exportar el log:

```
sudo journalctl -u saludo.service > /srv/logs/saludo_journal.log
```

*Al ejecutar el comando anterior me dice permission denied, he estado una hora buscando el error y no soy capaz de dar con el.

Esto guarda todos los logs del servicio saludo.service en el archivo /srv/logs/saludo_journal.log

Verificar:

```
cat /srv/logs/saludo_journal.log
```

