

LOGS:

El sistema de logs de Linux (log = registro), es un mecanismo estándar que se encarga de recoger los mensajes generados por los programas, aplicaciones y demonios y enviarlos a un destino predefinido. En cada mensaje consta la fuente (el programa que generó el mensaje), la prioridad (nivel de importancia del mensaje), la fecha y la hora.

1. Cómo funciona el sistema de logs:

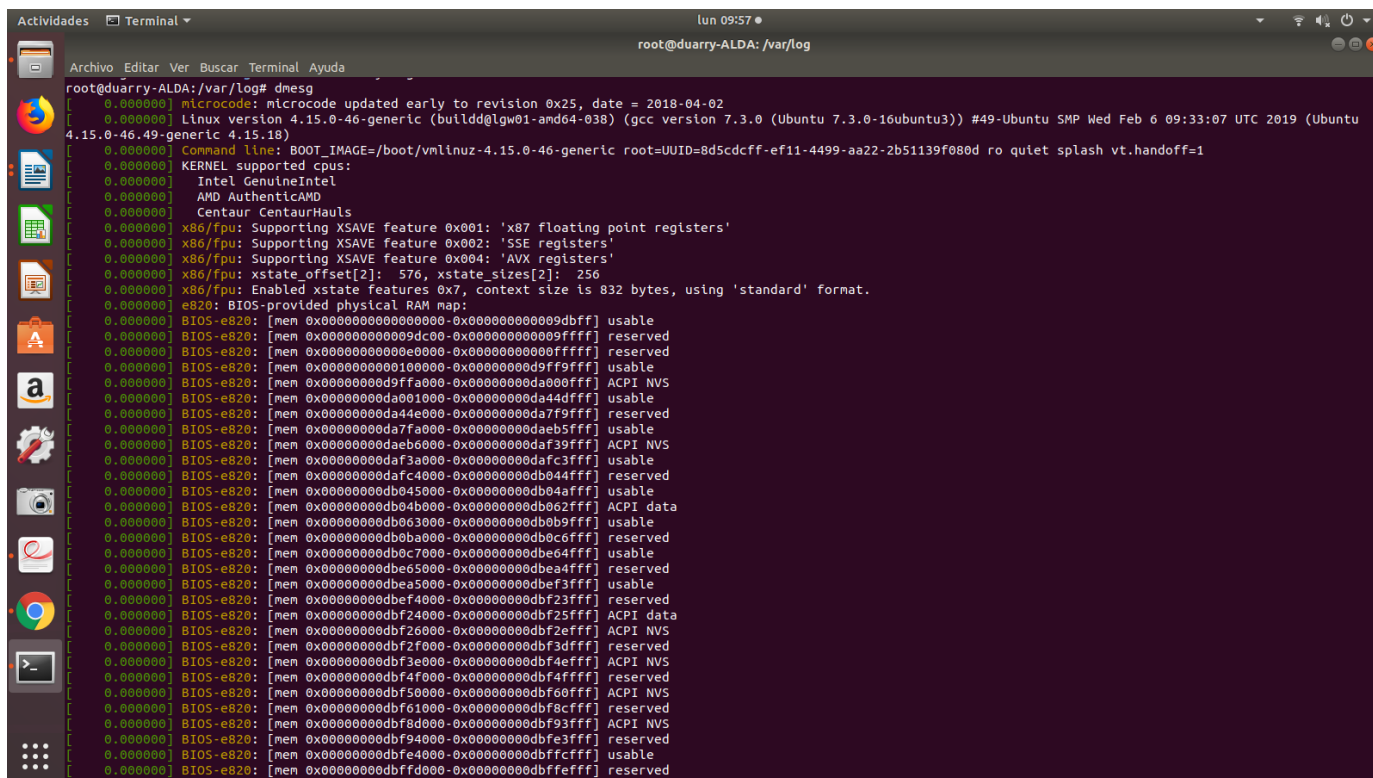
El sistema de logs arranca con el [script /etc/init.d/syslogd](#), y tiene dos demonios:

- ✓ **syslogd**: gestiona los logs del sistema. Distribuye los mensajes a archivos, tuberías, destinos remotos, terminales o usuarios, usando las indicaciones especificadas en su archivo de configuración `/etc/syslog.conf`, donde se indica qué se loguea y a dónde se envían estos logs.
- ✓ **klogd**: se encarga de los logs del kernel. Lo normal es que klogd envíe sus mensajes a syslogd pero no siempre es así, sobre todo en los eventos de alta prioridad, que salen directamente por pantalla.

Los logs se guardan en archivos ubicados en el directorio `/var/log`, aunque muchos programas manejan sus propios logs y los guardan en `/var/log/<programa>`. Además, es posible especificar múltiples destinos para un mismo mensaje.

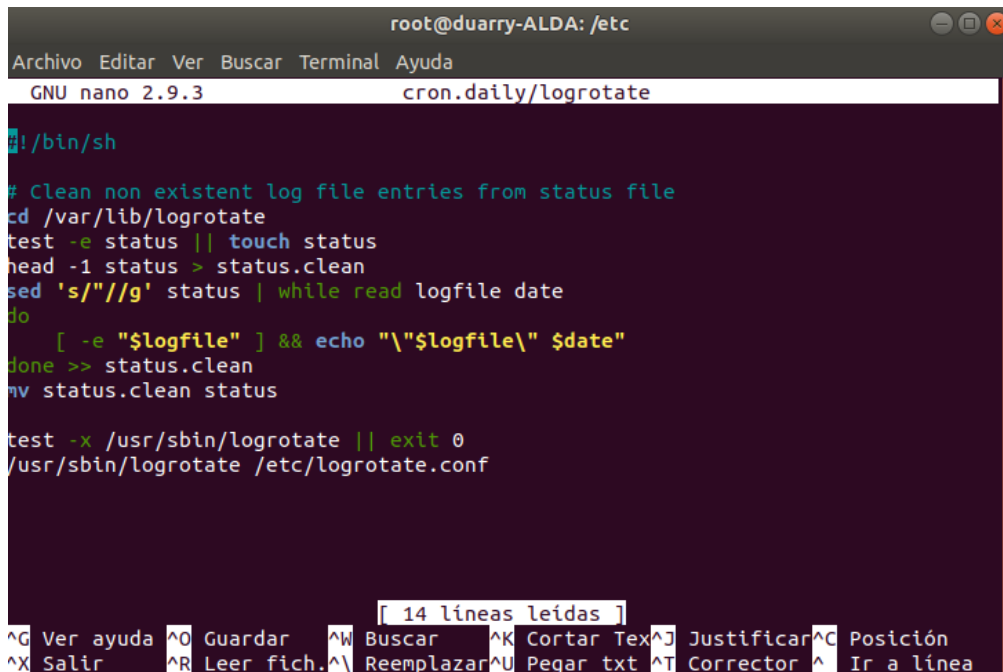
* **Ejemplo: `/var/log/dmesg`**: en este archivo se almacena la información que genera el kernel durante el arranque del sistema. Podemos ver su contenido con el comando `dmesg`:

\$ dmesg



```
root@duarry-ALDA: /var/log# dmesg
[ 0.000000] microcode: microcode updated early to revision 0x25, date = 2018-04-02
[ 0.000000] Linux version 4.15.0-46-generic (buildd@lgw01-amd64-038) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3)) #49-Ubuntu SMP Wed Feb 6 09:33:07 UTC 2019 (Ubuntu 4.15.0-46.49-generic 4.15.18)
[ 0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.15.0-46-generic root=UUID=8d5cdcff-ef11-4499-aa22-2b51139f080d ro quiet splash vt.handoff=1
[ 0.000000] KERNEL supported cpus:
[ 0.000000]   Intel GenuineIntel
[ 0.000000]   AMD AuthenticAMD
[ 0.000000]   Centaur CentaurHauls
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ 0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[ 0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard' format.
[ 0.000000] e820: BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000009dbf] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000009dc0-0x0000000000009fff] reserved
[ 0.000000] BIOS-e820: [mem 0x000000000000e000-0x000000000000ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000010000-0x0000000000009fff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000000d9ffa000-0x00000000000da00fff] ACPI NVS
[ 0.000000] BIOS-e820: [mem 0x0000000000da01000-0x0000000000da44dfff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000da44e000-0x0000000000da7f9fff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000da7fa000-0x0000000000daeb5fff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000daeb6000-0x0000000000daf39fff] ACPI NVS
[ 0.000000] BIOS-e820: [mem 0x0000000000daf3a000-0x0000000000dafc3fff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000dafc4000-0x0000000000db044fff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000db045000-0x0000000000db049fff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000db04a000-0x0000000000db062fff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x0000000000db063000-0x0000000000db069fff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000db06a000-0x0000000000db06cfff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000db06d000-0x0000000000db06ffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000db070000-0x0000000000db074fff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000db075000-0x0000000000db079fff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000db07a000-0x0000000000db07ffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000db080000-0x0000000000db083fff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000db084000-0x0000000000db087fff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000db088000-0x0000000000db08bfff] ACPI NVS
[ 0.000000] BIOS-e820: [mem 0x0000000000db08c000-0x0000000000db08ffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000db090000-0x0000000000db093fff] ACPI NVS
[ 0.000000] BIOS-e820: [mem 0x0000000000db094000-0x0000000000db097fff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000db098000-0x0000000000db09bfff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000db09c000-0x0000000000db09ffff] usable
[ 0.000000] BIOS-e820: [mem 0x0000000000db0a0000-0x0000000000db0a3fff] reserved
```

Los archivos de log crecen y con el tiempo se pueden volver muy extensos, pero en `/etc/cron.daily` está el script `/etc/cron.daily/logrotate`, (cuyo archivo de configuración es `/etc/logrotate.conf`), que se encarga de comprimirlos, añadiéndoles la extensión `.1.gz`, `.2.gz`, etc., volviendo a crear uno vacío.



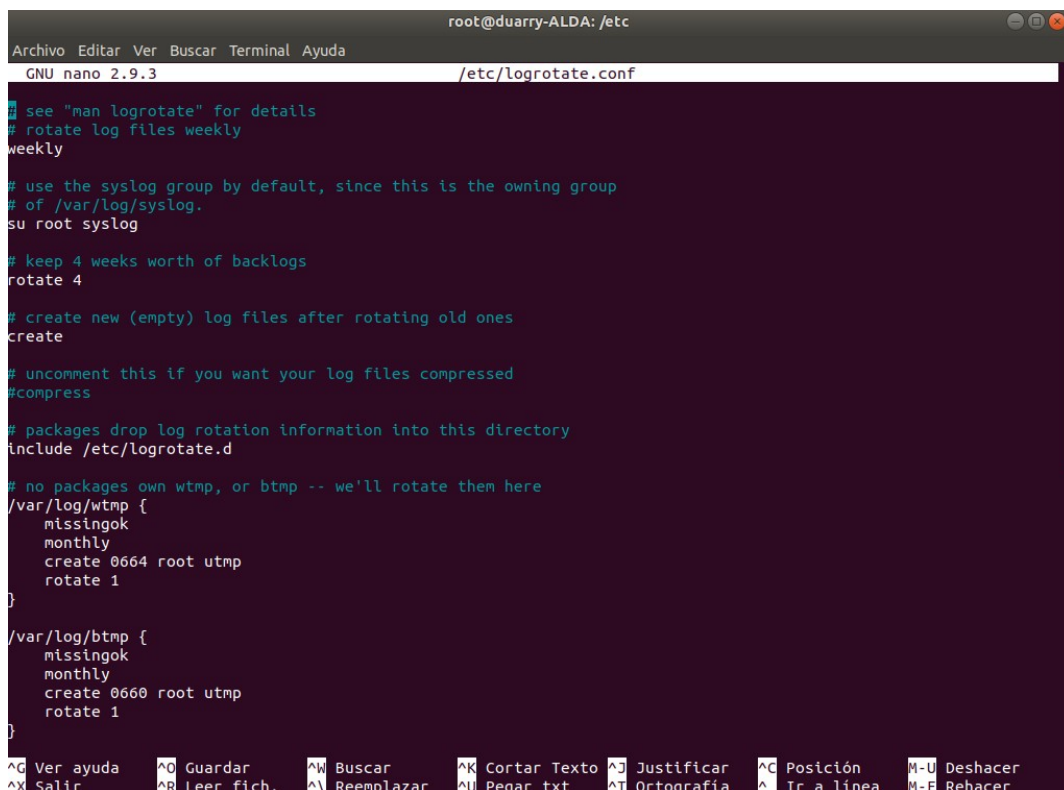
The screenshot shows a terminal window with the nano editor open to the file `/etc/cron.daily/logrotate`. The script content is as follows:

```
#!/bin/sh

# Clean non existent log file entries from status file
cd /var/lib/logrotate
test -e status || touch status
head -1 status > status.clean
sed 's/"//g' status | while read logfile date
do
    [ -e "$logfile" ] && echo "\"$logfile\" $date"
done >> status.clean
mv status.clean status

test -x /usr/sbin/logrotate || exit 0
/usr/sbin/logrotate /etc/logrotate.conf
```

The bottom of the window shows the nano editor's command palette with options like Ver ayuda, Guardar, Buscar, Cortar Text, Justificar, Posición, Salir, Leer fich., Reemplazar, Pegar txt, Corrector, and Ir a línea.



The screenshot shows a terminal window with the nano editor open to the file `/etc/logrotate.conf`. The configuration content is as follows:

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# use the syslog group by default, since this is the owning group
# of /var/log/syslog.
su root syslog

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0660 root utmp
    rotate 1
}
```

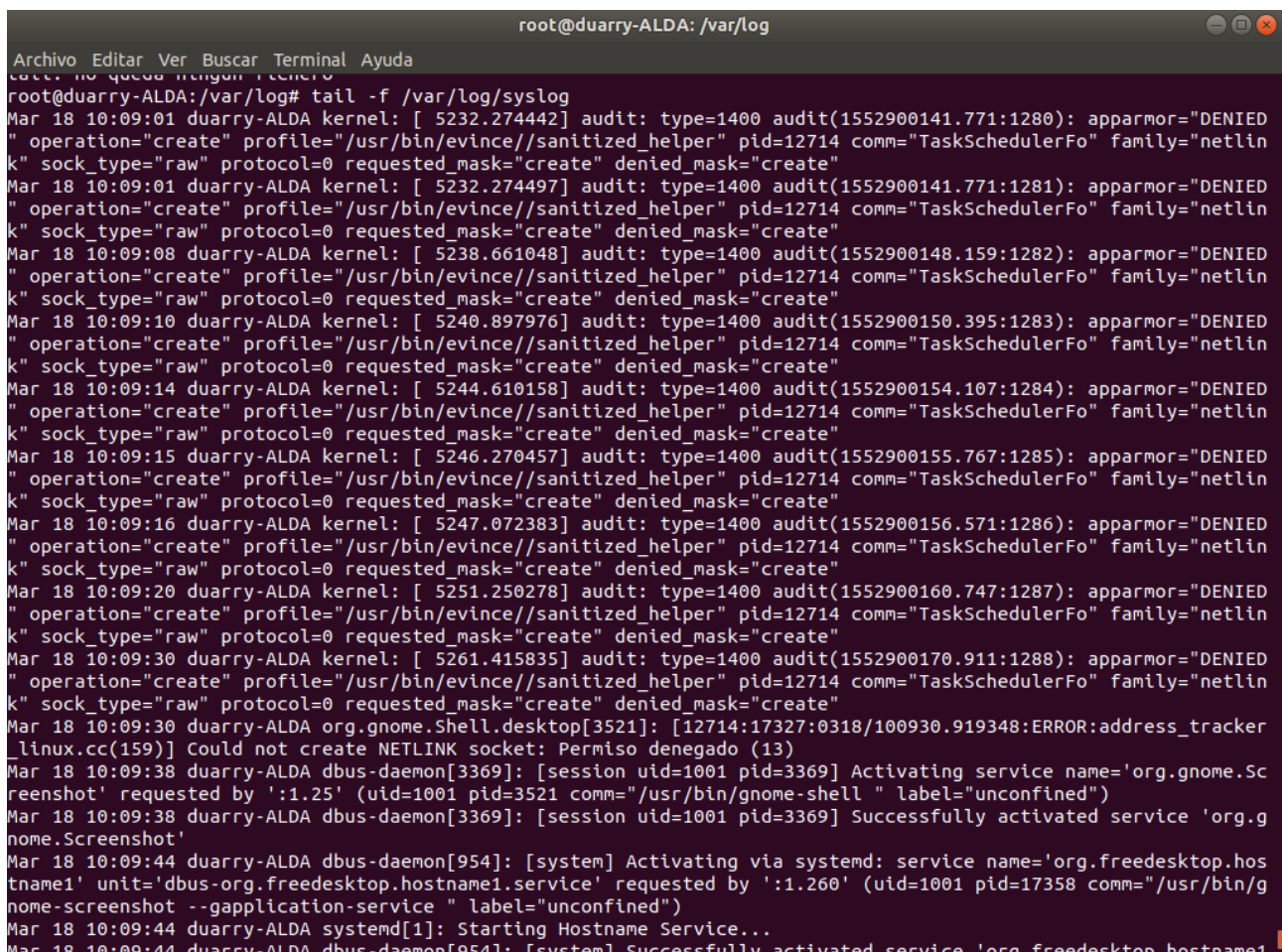
The bottom of the window shows the nano editor's command palette with options like Ver ayuda, Guardar, Buscar, Cortar Text, Justificar, Posición, Deshacer, Salir, Leer fich., Reemplazar, Pegar txt, Ortografía, Ir a línea, and Rehacer.

2. Monitorizar los logs en la consola:

Para monitorizar los logs en la consola, utilizaremos el comando **tail**. Este comando muestra las últimas líneas de uno o más archivos de texto, pero con la opción **-f**, en lugar de mostrar las últimas diez líneas y terminar, tail seguirá activo y conforme se añadan nuevas líneas al fichero las imprimirá, lo que es muy útil para monitorizar archivos.

Para monitorizar los logs en la consola, por ejemplo el archivo **/var/log/syslog**, haremos lo siguiente:

```
$ tail -f /var/log/syslog
```



```
root@duarry-ALDA: /var/log
Archivo Editar Ver Buscar Terminal Ayuda
root@duarry-ALDA:/var/log# tail -f /var/log/syslog
Mar 18 10:09:01 duarry-ALDA kernel: [ 5232.274442] audit: type=1400 audit(1552900141.771:1280): apparmor="DENIED
" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlin
k" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:01 duarry-ALDA kernel: [ 5232.274497] audit: type=1400 audit(1552900141.771:1281): apparmor="DENIED
" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlin
k" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:08 duarry-ALDA kernel: [ 5238.661048] audit: type=1400 audit(1552900148.159:1282): apparmor="DENIED
" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlin
k" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:10 duarry-ALDA kernel: [ 5240.897976] audit: type=1400 audit(1552900150.395:1283): apparmor="DENIED
" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlin
k" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:14 duarry-ALDA kernel: [ 5244.610158] audit: type=1400 audit(1552900154.107:1284): apparmor="DENIED
" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlin
k" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:15 duarry-ALDA kernel: [ 5246.270457] audit: type=1400 audit(1552900155.767:1285): apparmor="DENIED
" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlin
k" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:16 duarry-ALDA kernel: [ 5247.072383] audit: type=1400 audit(1552900156.571:1286): apparmor="DENIED
" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlin
k" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:20 duarry-ALDA kernel: [ 5251.250278] audit: type=1400 audit(1552900160.747:1287): apparmor="DENIED
" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlin
k" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:30 duarry-ALDA kernel: [ 5261.415835] audit: type=1400 audit(1552900170.911:1288): apparmor="DENIED
" operation="create" profile="/usr/bin/evince//sanitized_helper" pid=12714 comm="TaskSchedulerFo" family="netlin
k" sock_type="raw" protocol=0 requested_mask="create" denied_mask="create"
Mar 18 10:09:30 duarry-ALDA org.gnome.Shell.desktop[3521]: [12714:17327:0318/100930.919348:ERROR:address_tracker
_linux.cc(159)] Could not create NETLINK socket: Permiso denegado (13)
Mar 18 10:09:38 duarry-ALDA dbus-daemon[3369]: [session uid=1001 pid=3369] Activating service name='org.gnome.Sc
reenshot' requested by '1.25' (uid=1001 pid=3521 comm="/usr/bin/gnome-shell " label="unconfined")
Mar 18 10:09:38 duarry-ALDA dbus-daemon[3369]: [session uid=1001 pid=3369] Successfully activated service 'org.g
nome.Screenshot'
Mar 18 10:09:44 duarry-ALDA dbus-daemon[954]: [system] Activating via systemd: service name='org.freedesktop.hos
tname1' unit='dbus-org.freedesktop.hostname1.service' requested by '1.260' (uid=1001 pid=17358 comm="/usr/bin/g
nome-screenshot --gapplication-service " label="unconfined")
Mar 18 10:09:44 duarry-ALDA systemd[1]: Starting Hostname Service...
```

3. Cómo enviar todos los logs a un archivo:

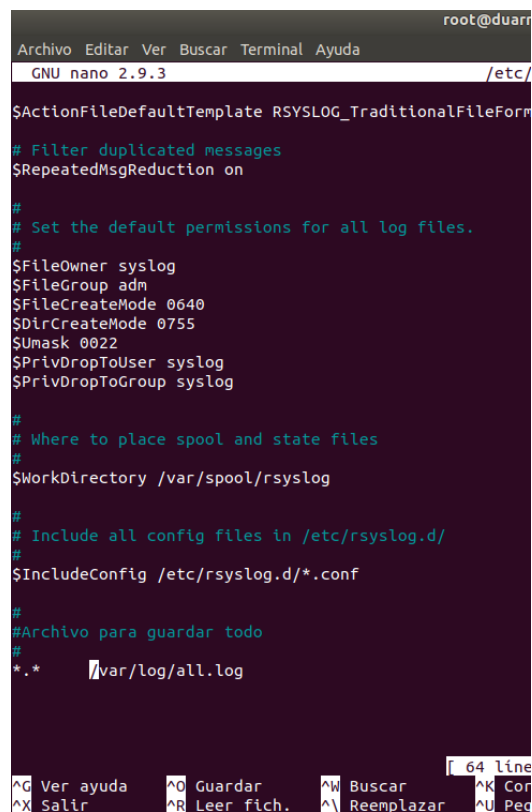
Muchas veces los logs no se miran, por lo que es buena idea configurar el sistema para enviar todos los logs a un archivo y visualizarlo en un terminal, para así poder ver en tiempo real todo lo que pasa en nuestra máquina. Para ello:

- a. Creamos el archivo vacío **/var/log/all.log**:

```
$ touch /var/log/all.log
```

b. Añadimos al archivo `/etc/rsyslog.conf` la línea:

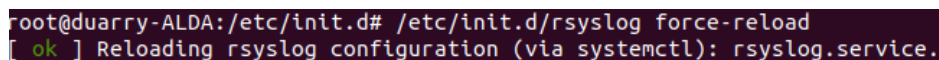
```
*.* /var/log/all.log
```



```
root@duarry
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /etc/
$ActionFileDefaultTemplate RSYSLOG_TraditionalFileForm
# Filter duplicated messages
$RepeatedMsgReduction on
#
# Set the default permissions for all log files.
#
$FileOwner syslog
$FileGroup adm
$FileCreateMode 0640
$DirCreateMode 0755
$UMask 0022
$PrivDropToUser syslog
$PrivDropToGroup syslog
#
# Where to place spool and state files
#
$WorkDirectory /var/spool/rsyslog
#
# Include all config files in /etc/rsyslog.d/
#
$IncludeConfig /etc/rsyslog.d/*.conf
#
# Archivo para guardar todo
#
*.* /var/log/all.log
r 64 line
^G Ver ayuda ^O Guardar ^W Buscar ^K Cor
^X Salir ^R Leer fich. ^_ Reemplazar ^U Pec
```

c. Volvemos a cargar la configuración de rsyslogd:

```
$ /etc/init.d/rsyslog force-reload
```



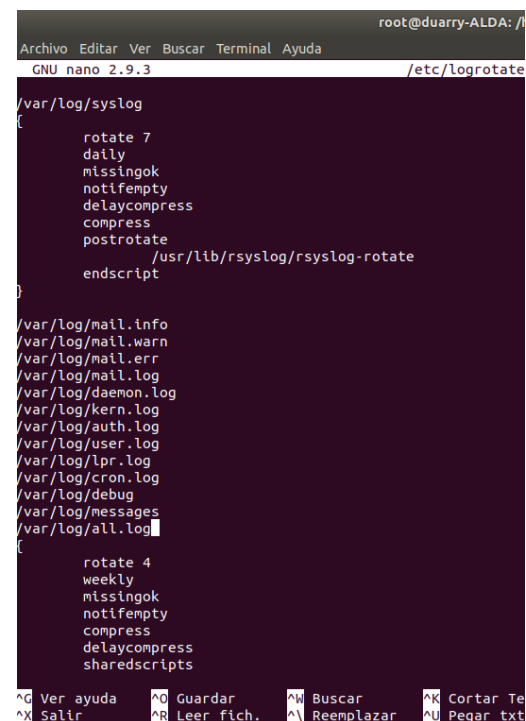
```
root@duarry-ALDA:/etc/init.d# /etc/init.d/rsyslog force-reload
[ ok ] Reloading rsyslog configuration (via systemctl): rsyslog.service.
```

d. Añadimos al archivo `/etc/logrotate.d/rsyslog` la línea:

```
/var/log/all.log
```

e. Por último, abrimos una consola y ejecutamos el comando:

```
$ tail -f /var/log/all.log
```



```
root@duarry-ALDA: /
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /etc/logrotate
/var/log/syslog
{
    rotate 7
    daily
    missingok
    notifempty
    delaycompress
    compress
    postrotate
        /usr/lib/rsyslog/rsyslog-rotate
    endsript
}
/var/log/mail.info
/var/log/mail.warn
/var/log/mail.err
/var/log/mail.log
/var/log/daemon.log
/var/log/kern.log
/var/log/auth.log
/var/log/user.log
/var/log/lpr.log
/var/log/cron.log
/var/log/debug
/var/log/messages
/var/log/all.log
{
    rotate 4
    weekly
    missingok
    notifempty
    compress
    delaycompress
    sharedscripts
}
```

4. Scripts que generan logs:

El comando logger permite enviar eventos al demonio syslogd, por lo que se utiliza en scripts para registrar mensajes vía syslogd. Por ejemplo, si hacemos:

```
$ logger -t mi_programa -f /var/log/messages "Mensaje ejemplo"
```

veremos que se ha añadido a `/var/log/messages` la línea:

```
May 14 23:10:13 pc450 mi_programa: Mensaje ejemplo
```