

IFA. Práctica de laboratorio 02

Hugo Fonseca Díaz
email uo258318@uniovi.es

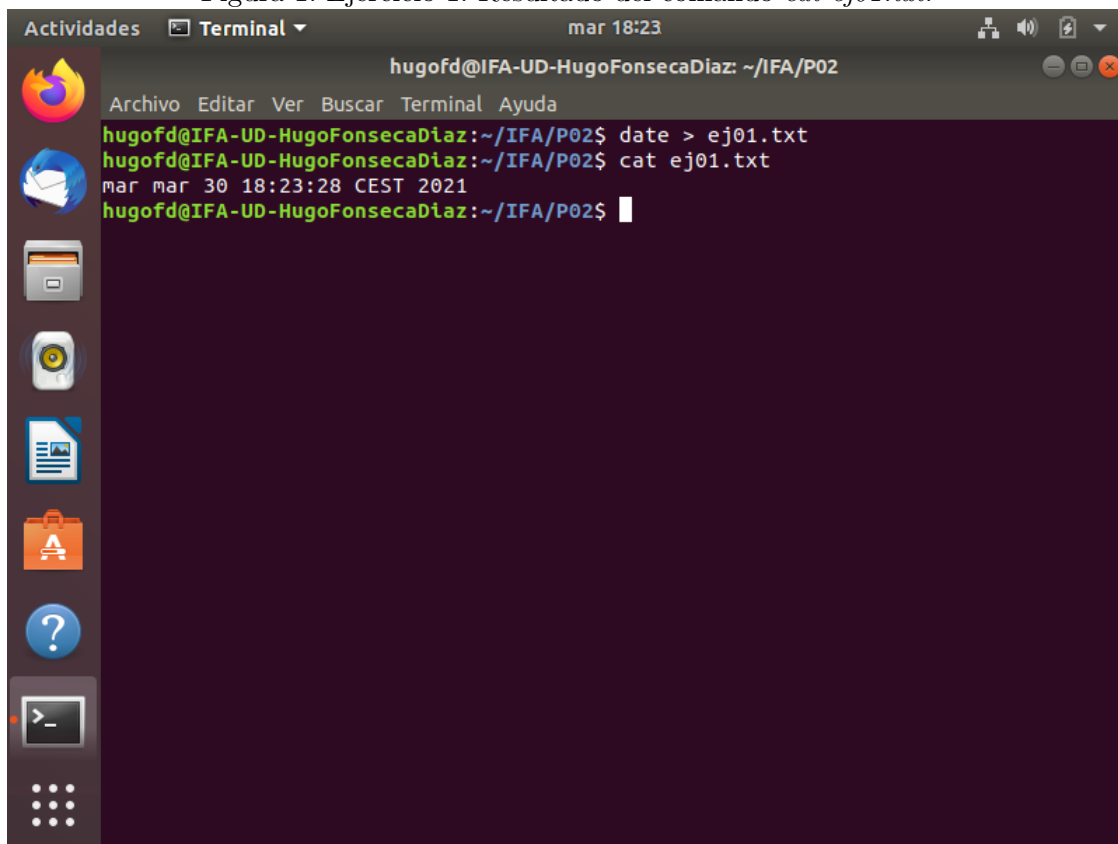
Escuela de Ingeniería Informática. Universidad de Oviedo.

21 de junio de 2021

1. Ejercicio 1

Se guarda la fecha y hora del sistema en el archivo `ej01.txt` con el comando `date > ej01.txt`. Se muestra ese archivo con el comando `cat`.

Figura 1: Ejercicio 1: Resultado del comando `cat ej01.txt`.



Se accede al sitio web <https://time.is/es/Spain> y se comprueba que la hora es la misma.

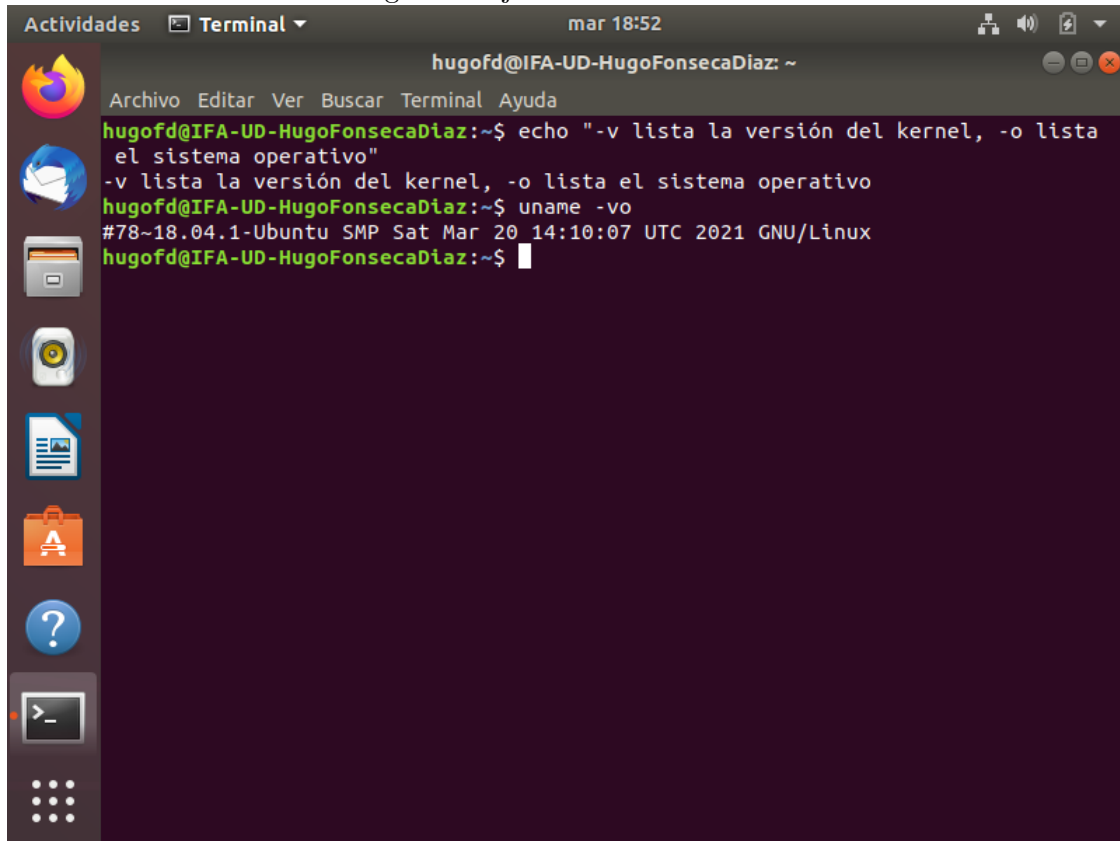
Figura 2: Ejercicio 1: Hora en el sitio web *time.is*.



2. Ejercicio 2

Se utiliza el comando `uname` con las opciones `v` (lista la versión del kernel) y `o` (lista el nombre del sistema operativo).

Figura 3: Ejercicio 2: *uname -vo*.



The image shows a terminal window titled "Terminal" with a menu bar containing "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The window title bar also shows "mar 18:52". The terminal content shows the user "hugofd@IFA-UD-HugoFonsecaDiaz" at the prompt. The user enters the command `echo "-v lista la versión del kernel, -o lista el sistema operativo"`, which outputs `-v lista la versión del kernel, -o lista el sistema operativo`. Then, the user enters `uname -vo`, which outputs `#78~18.04.1-Ubuntu SMP Sat Mar 20 14:10:07 UTC 2021 GNU/Linux`. The terminal window has a sidebar on the left with icons for various applications like Firefox, Mail, Files, and a Dash icon at the bottom.

```
hugofd@IFA-UD-HugoFonsecaDiaz:~$ echo "-v lista la versión del kernel, -o lista el sistema operativo"
-v lista la versión del kernel, -o lista el sistema operativo
hugofd@IFA-UD-HugoFonsecaDiaz:~$ uname -vo
#78~18.04.1-Ubuntu SMP Sat Mar 20 14:10:07 UTC 2021 GNU/Linux
hugofd@IFA-UD-HugoFonsecaDiaz:~$
```

3. Ejercicio 3

Se utiliza el comando `lshw`, primero con la flag `short` para encontrar el nombre de la clase de los dispositivos de red.

Figura 4: Ejercicio 3: *lshw -short*.

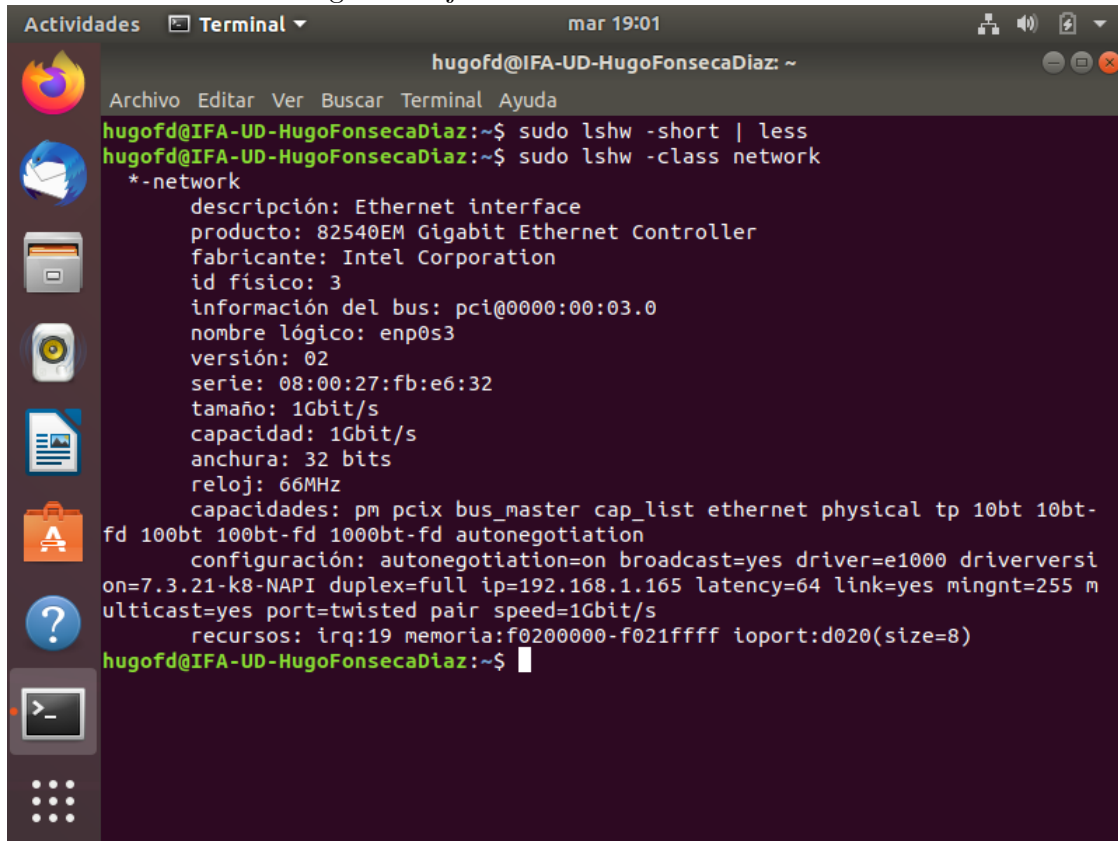
```

hugofd@IFA-UD-HugoFonsecaDiaz: ~
lshw -short
/0/0          memory      128KiB BIOS
/0/1          memory      1987MiB Memoria de sistema
/0/2          processor   Intel(R) Core(TM) i7-8550U CPU @ 1.80G
Hz
/0/100        bridge      440FX - 82441FX PMC [Natoma]
/0/100/1      bridge      82371SB PIIX3 ISA [Natoma/Triton II]
/0/100/1.1    storage     82371AB/EB/MB PIIX4 IDE
/0/100/2      display     SVGA II Adapter
/0/100/3      enp0s3     network    82540EM Gigabit Ethernet Controller
/0/100/4      generic     VirtualBox Guest Service
/0/100/5      multimedia  82801AA AC'97 Audio Controller
/0/100/6      bus          KeyLargo/Intrepid USB
/0/100/6/1    usb1         bus        OHCI PCI host controller
/0/100/6/1/1  input        USB Tablet
/0/100/7      bridge      82371AB/EB/MB PIIX4 ACPI
/0/100/d      storage     82801HM/HEM (ICH8M/ICH8M-E) SATA Contr
oller [AHCI mode]
/0/3          scsi1       storage     CD-ROM
/0/3/0.0.0    /dev/cdrom   disk
/0/4          scsi2       storage     42GB VBOX HARDDISK
/0/4/0.0.0    /dev/sda     disk
/0/4/0.0.0/1  /dev/sda1    volume     5721MiB partici3n EXT4
/0/4/0.0.0/2  /dev/sda2    volume     4768MiB partici3n EXT4
/0/4/0.0.0/3  /dev/sda3    volume     23GiB partici3n EXT4
/0/4/0.0.0/4  /dev/sda4    volume     6626MiB Extended partition
/0/4/0.0.0/4/5 /dev/sda5    volume     1906MiB partici3n EXT4
/0/4/0.0.0/4/6 /dev/sda6    volume     1904MiB partici3n EXT4

```

Una vez se sabe que el nombre de la clase de los dispositivos de red es **network**, se utiliza el comando **lshw** con la flag **-class network**.

Figura 5: Ejercicio 3: *lshw -class network*.

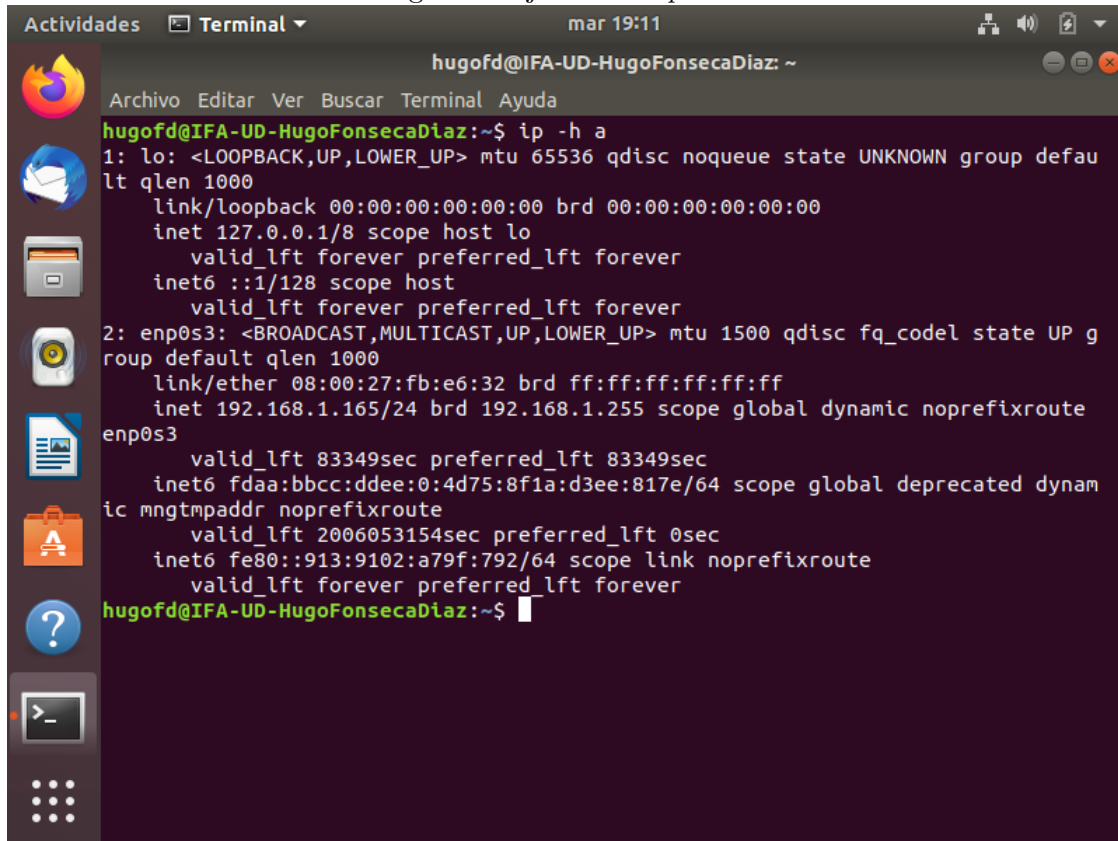


The image shows a terminal window titled "hugofd@IFA-UD-HugoFonsecaDiaz: ~" with a menu bar containing "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The terminal displays the following commands and output:

```
hugofd@IFA-UD-HugoFonsecaDiaz:~$ sudo lshw -short | less
hugofd@IFA-UD-HugoFonsecaDiaz:~$ sudo lshw -class network
*-network
    descripción: Ethernet interface
    producto: 82540EM Gigabit Ethernet Controller
    fabricante: Intel Corporation
    id físico: 3
    información del bus: pci@0000:00:03.0
    nombre lógico: enp0s3
    versión: 02
    serie: 08:00:27:fb:e6:32
    tamaño: 1Gbit/s
    capacidad: 1Gbit/s
    anchura: 32 bits
    reloj: 66MHz
    capacidades: pm pcix bus_master cap_list ethernet physical tp 10bt 10bt-
fd 100bt 100bt-fd 1000bt-fd autonegotiation
    configuración: autonegotiation=on broadcast=yes driver=e1000 driverversi
on=7.3.21-k8-NAPI duplex=full ip=192.168.1.165 latency=64 link=yes mingnt=255 m
ulticast=yes port=twisted pair speed=1Gbit/s
    recursos: irq:19 memoria:f0200000-f021ffff ioport:d020(size=8)
hugofd@IFA-UD-HugoFonsecaDiaz:~$
```

También puede utilizarse el comando `ip -h enp0s3` para mostrar más información sobre el dispositivo de red `enp0s3`.

Figura 6: Ejercicio 3: `ip -h a`.



```
hugofd@IFA-UD-HugoFonsecaDiaz: ~$ ip -h a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
lt 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
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:fb:e6:32 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.165/24 brd 192.168.1.255 scope global dynamic noprefixroute
enp0s3
        valid_lft 83349sec preferred_lft 83349sec
    inet6 fd00::bbcc:ddee:0:4d75:8f1a:d3ee:817e/64 scope global deprecated dynam
ic mngtmpaddr noprefixroute
        valid_lft 2006053154sec preferred_lft 0sec
    inet6 fe80::913:9102:a79f:792/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
hugofd@IFA-UD-HugoFonsecaDiaz: ~$
```

4. Ejercicio 4

Se utiliza el comando `netstat` del paquete `net-tools`. Su flag `a` permite ver todos los sockets, por lo que `sudo netstat -a > ej04.txt` guarda la información de los sockets activos y no activos en un fichero de texto. También son interesantes sus flags `n` (se muestran las direcciones numéricamente), `p` (se muestran los procesos pertenecientes a los sockets), `t` (tcp) y `u` (udp).

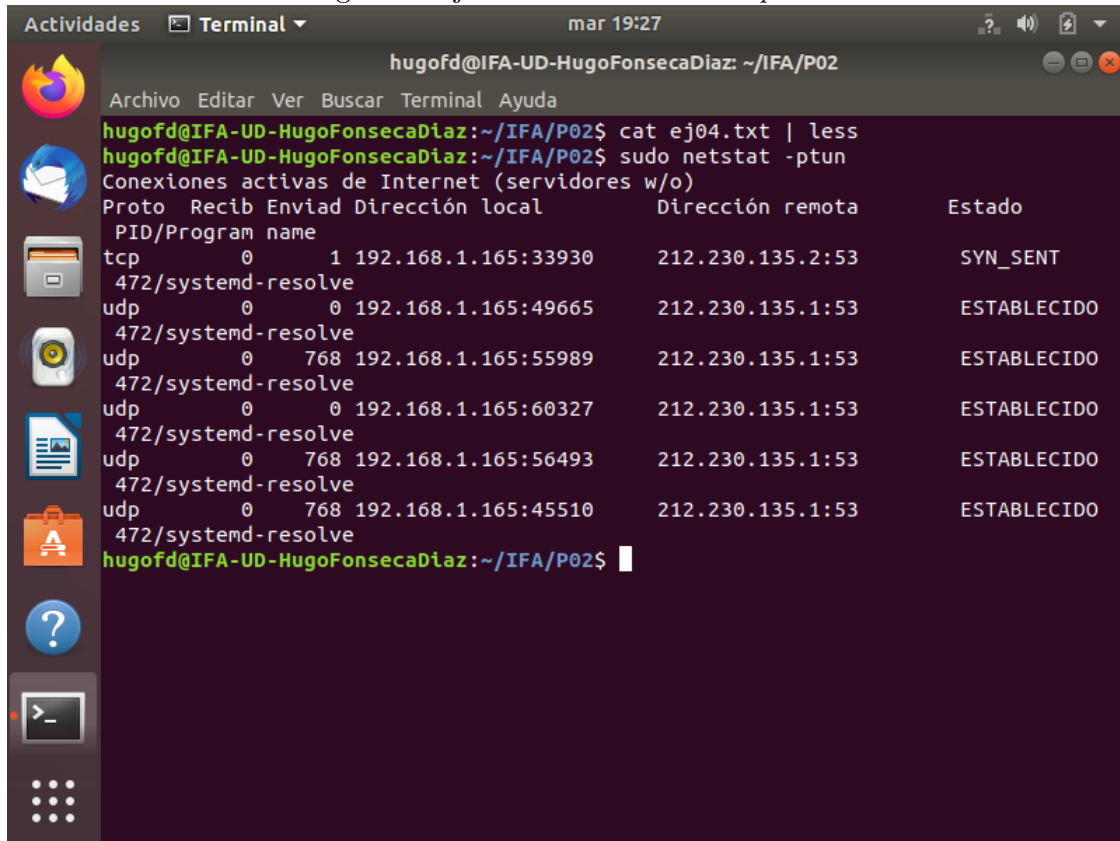
Figura 7: Ejercicio 4: *cat ej04.txt / less.*

```

hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02
raw6      0      0 [::]:ipv6-icmp      [::]:*      7
Sockets activos de dominio UNIX (servidores y establecidos)
Proto RefCnt Flags      Type      State      I-Node      Ruta
unix  2      [ ACC ]      FLUJO      ESCUCHANDO 27759      @/tmp/.ICE-unix/1484
unix  2      [ ACC ]      FLUJO      ESCUCHANDO 27310      @/tmp/dbus-q0eqbK05
unix  2      [ ]        DGRAM      ESCUCHANDO 27179      /run/user/1000/syste
md/notify
unix  2      [ ]        DGRAM      ESCUCHANDO 22206      /run/user/121/system
d/notify
unix  2      [ ACC ]      SEQPACKET  ESCUCHANDO 13206      /run/udev/control
unix  2      [ ACC ]      FLUJO      ESCUCHANDO 27182      /run/user/1000/syste
md/private
unix  2      [ ACC ]      FLUJO      ESCUCHANDO 22209      /run/user/121/system
d/private
unix  2      [ ACC ]      FLUJO      ESCUCHANDO 27186      /run/user/1000/gnupg
/S.gpg-agent.extra
unix  2      [ ACC ]      FLUJO      ESCUCHANDO 22377      /run/user/121/gnupg/
S.gpg-agent.extra
unix  2      [ ACC ]      FLUJO      ESCUCHANDO 27187      /run/user/1000/snapd
-session-agent.socket
unix  2      [ ACC ]      FLUJO      ESCUCHANDO 22378      /run/user/121/bus
unix  2      [ ACC ]      FLUJO      ESCUCHANDO 27188      /run/user/1000/gnupg
/S.gpg-agent.brower
unix  2      [ ACC ]      FLUJO      ESCUCHANDO 27189      /run/user/1000/gnupg
/S.gpg-agent
unix  2      [ ACC ]      FLUJO      ESCUCHANDO 22379      /run/user/121/pulse/
native
:

```

Figura 8: Ejercicio 4: *sudo netstat -ptun*.



```
hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02$ cat ej04.txt | less
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02$ sudo netstat -ptun
Conexiones activas de Internet (servidores w/o)
Proto Recib Enviad Dirección local Dirección remota Estado
PID/Program name
tcp 0 1 192.168.1.165:33930 212.230.135.2:53 SYN_SENT
472/systemd-resolve
udp 0 0 192.168.1.165:49665 212.230.135.1:53 ESTABLECIDO
472/systemd-resolve
udp 0 768 192.168.1.165:55989 212.230.135.1:53 ESTABLECIDO
472/systemd-resolve
udp 0 0 192.168.1.165:60327 212.230.135.1:53 ESTABLECIDO
472/systemd-resolve
udp 0 768 192.168.1.165:56493 212.230.135.1:53 ESTABLECIDO
472/systemd-resolve
udp 0 768 192.168.1.165:45510 212.230.135.1:53 ESTABLECIDO
472/systemd-resolve
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02$
```

También se puede ver información de los servicios de red en `/etc/services`.

Figura 9: Ejercicio 4: *less /etc/services*.

```
hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, officially ports have two entries
# even if the protocol doesn't support UDP operations.
#
# Updated from http://www.iana.org/assignments/port-numbers and other
# sources like http://www.freebsd.org/cgi/cvsweb.cgi/src/etc/services .
# New ports will be added on request if they have been officially assigned
# by IANA and used in the real-world or are needed by a debian package.
# If you need a huge list of used numbers please install the nmap package.

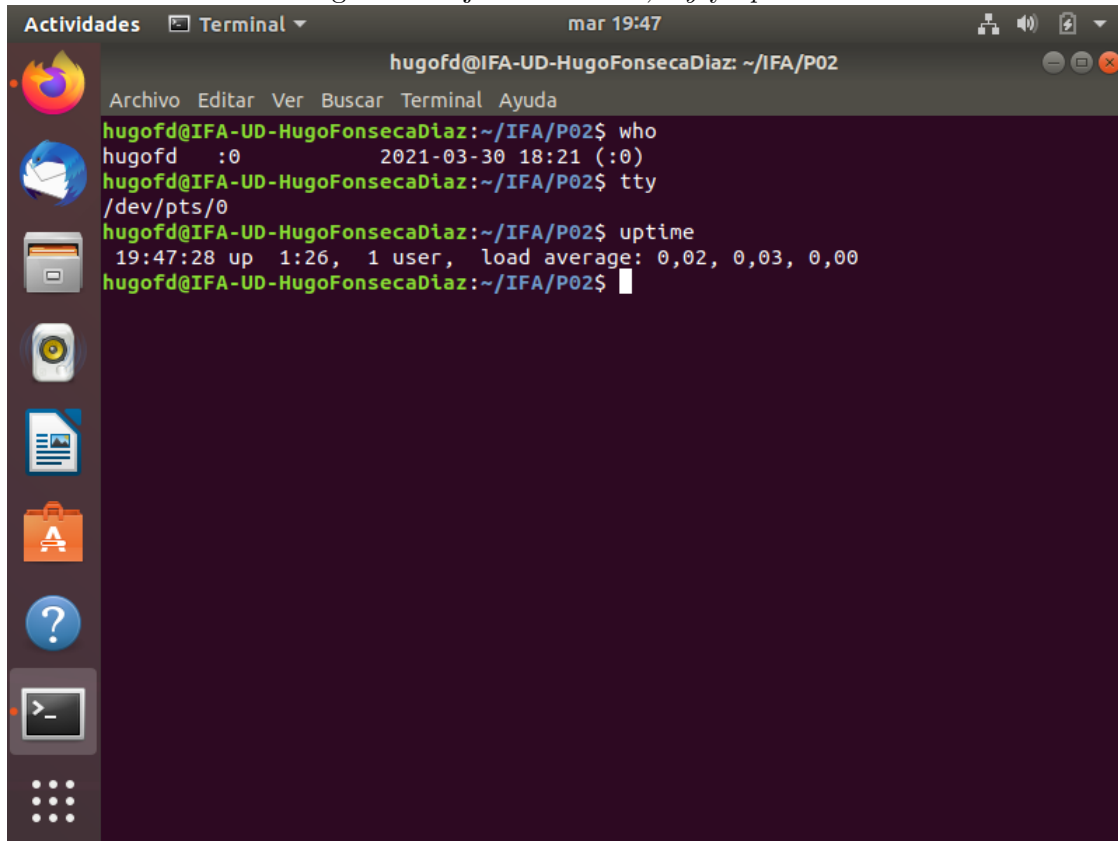
tcpmux      1/tcp                                # TCP port service multiplexer
echo        7/tcp
echo        7/udp
discard     9/tcp                                sink null
discard     9/udp                                sink null
sysstat     11/tcp                                users
daytime     13/tcp
daytime     13/udp
netstat     15/tcp
qotd        17/tcp                                quote
msp         18/tcp                                # message send protocol
msp         18/udp
chargen     19/tcp                                ttytst source
chargen     19/udp                                ttytst source
ftp-data    20/tcp

/etc/services
```

5. Ejercicio 5

Para resolver este ejercicio se usan tres comandos: **who** muestra los usuarios conectados y la terminal en la que están, **tty** muestra la terminal conectada actualmente al standard input y **uptime** muestra el tiempo que ha pasado desde el arranque del sistema.

Figura 10: Ejercicio 5: *who*, *tty* y *uptime*.



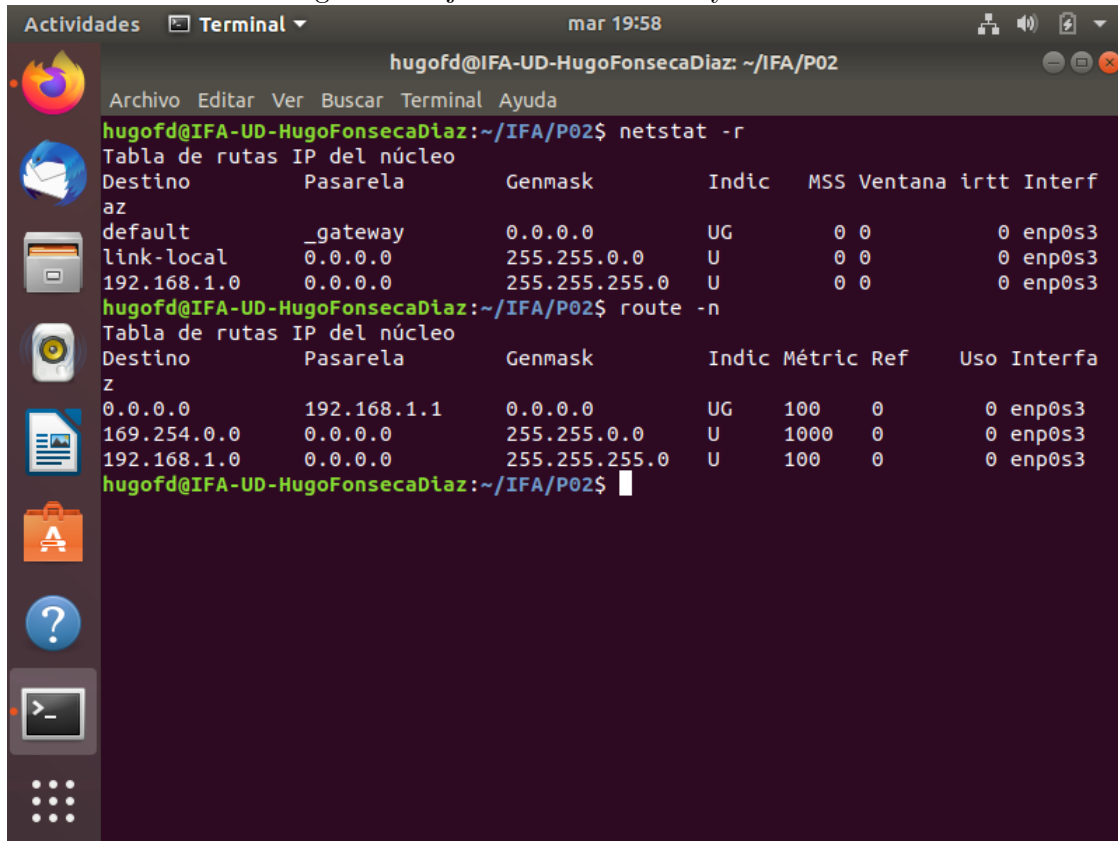
The image shows a terminal window titled "hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02". The window has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The terminal output is as follows:

```
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02$ who
hugofd  :0                2021-03-30 18:21 (:0)
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02$ tty
/dev/pts/0
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02$ uptime
19:47:28 up 1:26, 1 user, load average: 0,02, 0,03, 0,00
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02$
```

6. Ejercicio 6

Existen al menos dos opciones de mostrar la información sobre la tabla de enrutamiento: mediante el comando `netstat` con su flag `r` (que muestra la tabla de enrutamiento) o usando el comando `route` con su flag `n` (que muestra las direcciones de red de forma numérica).

Figura 11: Ejercicio 6: *netstat -r* y *route -n*.



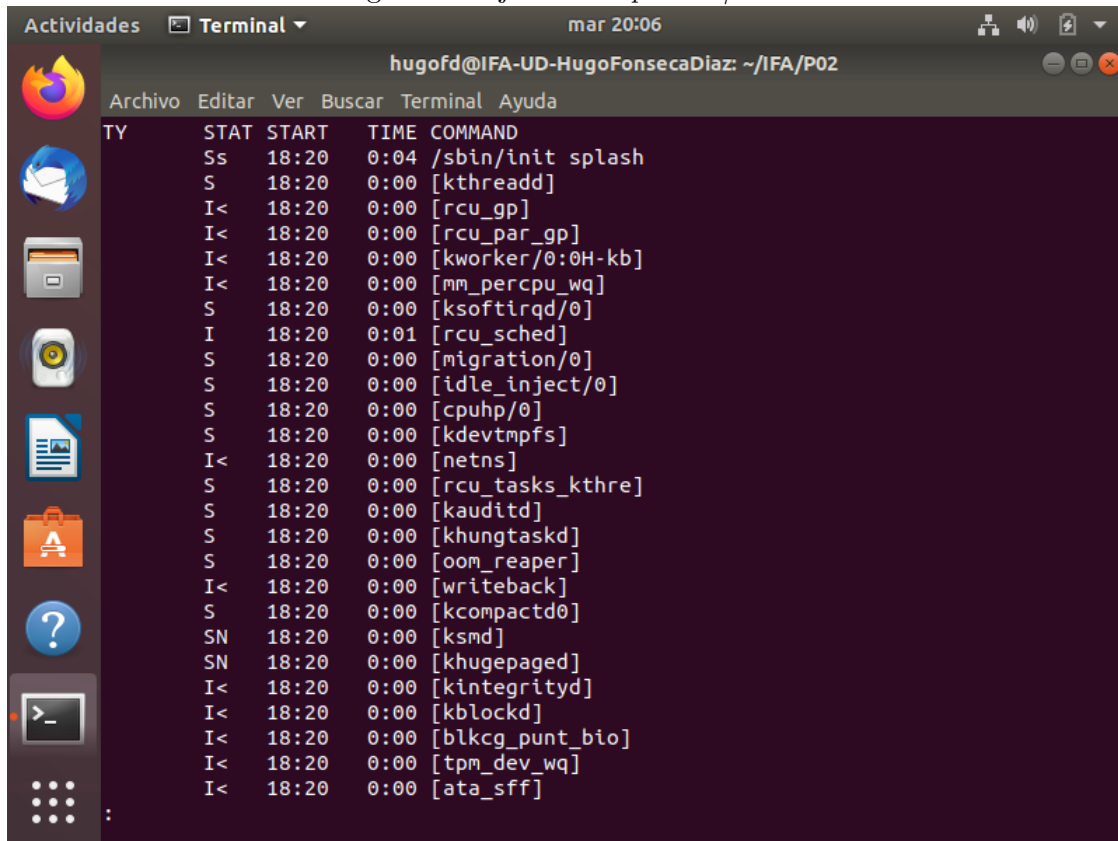
The screenshot shows a terminal window titled "hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02". The user has executed two commands: `netstat -r` and `route -n`. The output of `netstat -r` shows the kernel routing table with columns: Destino, Pasarela, Genmask, Indic, MSS, Ventana, irtt, and Interf. The output of `route -n` shows the IP routing table with columns: Destino, Pasarela, Genmask, Indic, Métric, Ref, and Uso Interfa.

```
hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02$ netstat -r
Tabla de rutas IP del núcleo
Destino      Pasarela      Genmask      Indic  MSS Ventana irtt Interf
az
default      _gateway      0.0.0.0      UG      0 0      0 enp0s3
link-local    0.0.0.0      255.255.0.0  U      0 0      0 enp0s3
192.168.1.0   0.0.0.0      255.255.255.0 U      0 0      0 enp0s3
hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02$ route -n
Tabla de rutas IP del núcleo
Destino      Pasarela      Genmask      Indic  Métric Ref  Uso Interfa
Z
0.0.0.0      192.168.1.1   0.0.0.0      UG      100  0      0 enp0s3
169.254.0.0   0.0.0.0      255.255.0.0  U      1000 0      0 enp0s3
192.168.1.0   0.0.0.0      255.255.255.0 U      100  0      0 enp0s3
hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02$
```

7. Ejercicio 7

Se usa el comando `ps`. Dicho comando puede utilizarse siguiendo tres sintaxis: la de UNIX, la de BSD o la de GNU. Para mostrar todos los procesos del sistema con sintaxis de UNIX podría usarse `ps -eF`. Con sintaxis de BSD se puede usar `ps axu`. Para que se muestre el nombre del proceso sin cortarse se puede pasar el resultado del comando `ps` al comando `less` con una pipe de UNIX.

Figura 12: Ejercicio 7: *ps aux | less*.

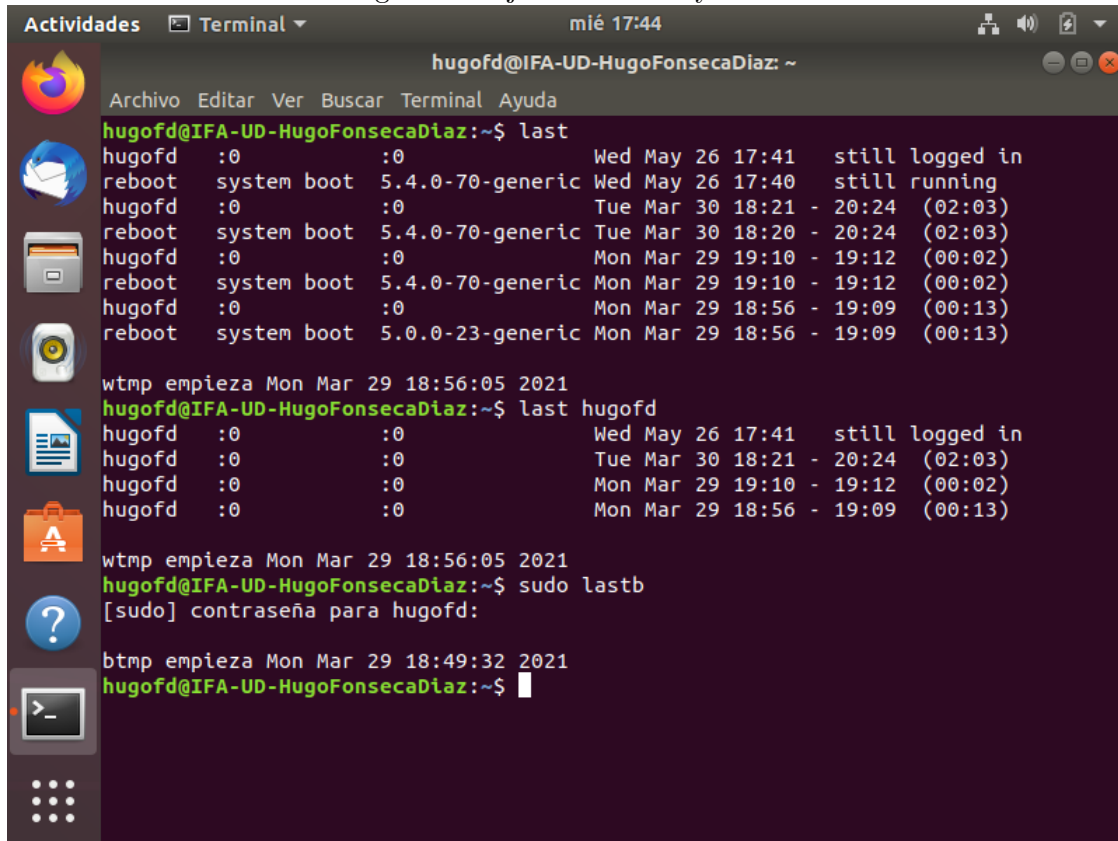


```
hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02
TY  STAT  START  TIME  COMMAND
Ss  18:20  0:04  /sbin/init splash
S   18:20  0:00  [kthreadd]
I<  18:20  0:00  [rcu_gp]
I<  18:20  0:00  [rcu_par_gp]
I<  18:20  0:00  [kworker/0:0H-kb]
I<  18:20  0:00  [mm_percpu_wq]
S   18:20  0:00  [ksoftirqd/0]
I   18:20  0:01  [rcu_sched]
S   18:20  0:00  [migration/0]
S   18:20  0:00  [idle_inject/0]
S   18:20  0:00  [cpuhp/0]
S   18:20  0:00  [kdevtmpfs]
I<  18:20  0:00  [netns]
S   18:20  0:00  [rcu_tasks_kthre]
S   18:20  0:00  [kauditd]
S   18:20  0:00  [khungtaskd]
S   18:20  0:00  [oom_reaper]
I<  18:20  0:00  [writeback]
S   18:20  0:00  [kcompactd0]
SN  18:20  0:00  [ksmd]
SN  18:20  0:00  [khugepaged]
I<  18:20  0:00  [kintegrityd]
I<  18:20  0:00  [kblockd]
I<  18:20  0:00  [blkcg_punt_bio]
I<  18:20  0:00  [tpm_dev_wq]
I<  18:20  0:00  [ata_sff]
```

8. Ejercicio 8

Se usarán los comandos `last` y `lastb`. El primero se utiliza para sacar la información de los accesos de todos los usuarios al sistema, incluyendo también un ejemplo de uso para un usuario concreto. El segundo es un comando similar pero buscando en `/var/log/btmp`, lo que muestra intentos fallidos de acceso al sistema.

Figura 13: Ejercicio 8: *last* y *lastb*.



The screenshot shows a terminal window titled "hugofd@IFA-UD-HugoFonsecaDiaz: ~" with a menu bar containing "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The terminal displays the output of the `last` command, which lists system boot events and user logins. The output is as follows:

```
hugofd@IFA-UD-HugoFonsecaDiaz:~$ last
hugofd :0 :0 Wed May 26 17:41 still logged in
reboot system boot 5.4.0-70-generic Wed May 26 17:40 still running
hugofd :0 :0 Tue Mar 30 18:21 - 20:24 (02:03)
reboot system boot 5.4.0-70-generic Tue Mar 30 18:20 - 20:24 (02:03)
hugofd :0 :0 Mon Mar 29 19:10 - 19:12 (00:02)
reboot system boot 5.4.0-70-generic Mon Mar 29 19:10 - 19:12 (00:02)
hugofd :0 :0 Mon Mar 29 18:56 - 19:09 (00:13)
reboot system boot 5.0.0-23-generic Mon Mar 29 18:56 - 19:09 (00:13)

wtmp empieza Mon Mar 29 18:56:05 2021
hugofd@IFA-UD-HugoFonsecaDiaz:~$ last hugofd
hugofd :0 :0 Wed May 26 17:41 still logged in
hugofd :0 :0 Tue Mar 30 18:21 - 20:24 (02:03)
hugofd :0 :0 Mon Mar 29 19:10 - 19:12 (00:02)
hugofd :0 :0 Mon Mar 29 18:56 - 19:09 (00:13)

wtmp empieza Mon Mar 29 18:56:05 2021
hugofd@IFA-UD-HugoFonsecaDiaz:~$ sudo lastb
[sudo] contraseña para hugofd:

btmp empieza Mon Mar 29 18:49:32 2021
hugofd@IFA-UD-HugoFonsecaDiaz:~$
```

9. Ejercicio 9

Se utiliza el comando `lsof`, cuya salida está pensada para ser la entrada de otro programa que la parsee. Se hace una pipe de Unix con el comando `less` para poder visualizar la salida del comando.

Figura 14: Ejercicio 9: *lsof* / *less*.

```

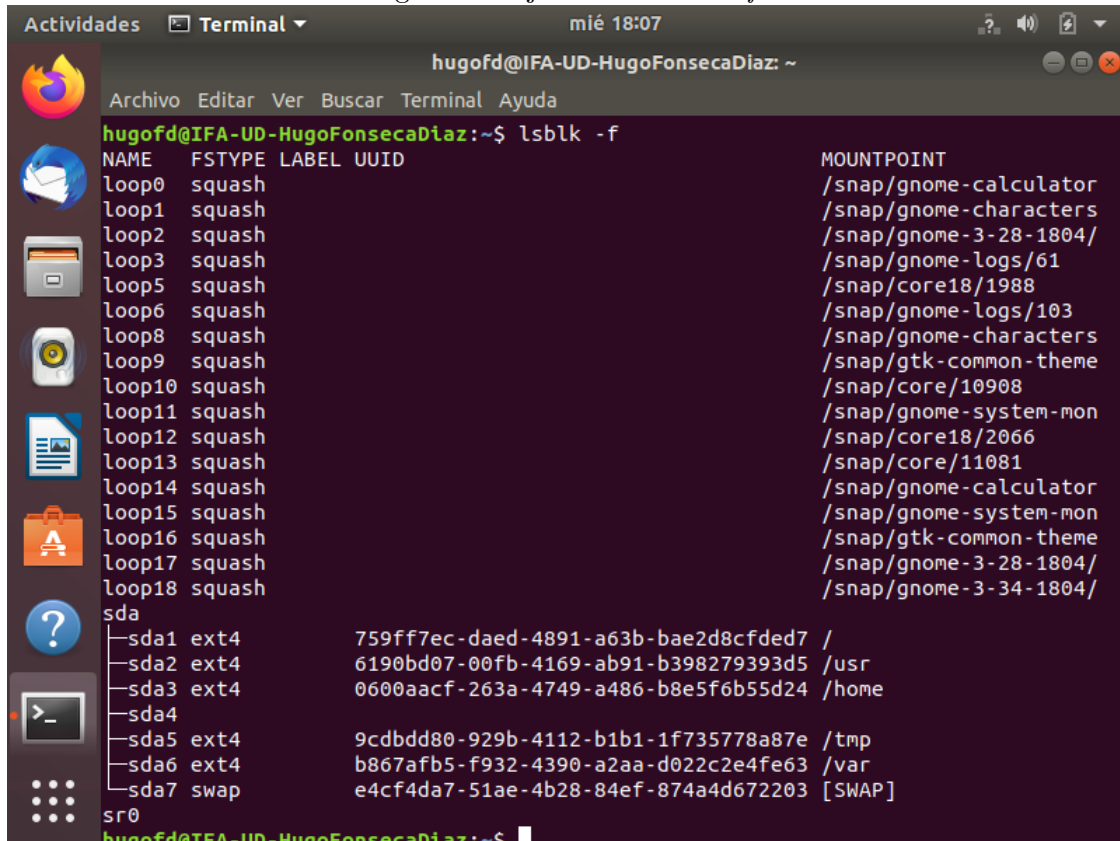
hugofd@IFA-UD-HugoFonsecaDiaz: ~
COMMAND  PID  TID      USER    FD      TYPE      DEVICE SIZE/OFF
NODE NAME
systemd   1    1        root    cwd     unknown
/proc/1/cwd (readlink: Permission denied)
systemd   1    1        root    rtd     unknown
/proc/1/root (readlink: Permission denied)
systemd   1    1        root    txt     unknown
/proc/1/exe (readlink: Permission denied)
systemd   1    1        root    NOFD
/proc/1/fd (opendir: Permission denied)
kthreadd  2    2        root    cwd     unknown
/proc/2/cwd (readlink: Permission denied)
kthreadd  2    2        root    rtd     unknown
/proc/2/root (readlink: Permission denied)
kthreadd  2    2        root    txt     unknown
/proc/2/exe (readlink: Permission denied)
kthreadd  2    2        root    NOFD
/proc/2/fd (opendir: Permission denied)
rcu_gp    3    3        root    cwd     unknown
/proc/3/cwd (readlink: Permission denied)
rcu_gp    3    3        root    rtd     unknown
/proc/3/root (readlink: Permission denied)
rcu_gp    3    3        root    txt     unknown
/proc/3/exe (readlink: Permission denied)
rcu_gp    3    3        root    NOFD
/proc/3/fd (opendir: Permission denied)
rcu_par_g 4    4        root    cwd     unknown

```

10. Ejercicio 10

Se puede usar el comando `lsblk` con la opción `f`. El comando muestra información de los dispositivos del sistema y la opción `f` muestra los sistemas de ficheros de los mismos.

Figura 15: Ejercicio 10: *lsblk -f*.



```
hugofd@IFA-UD-HugoFonsecaDiaz:~$ lsblk -f
```

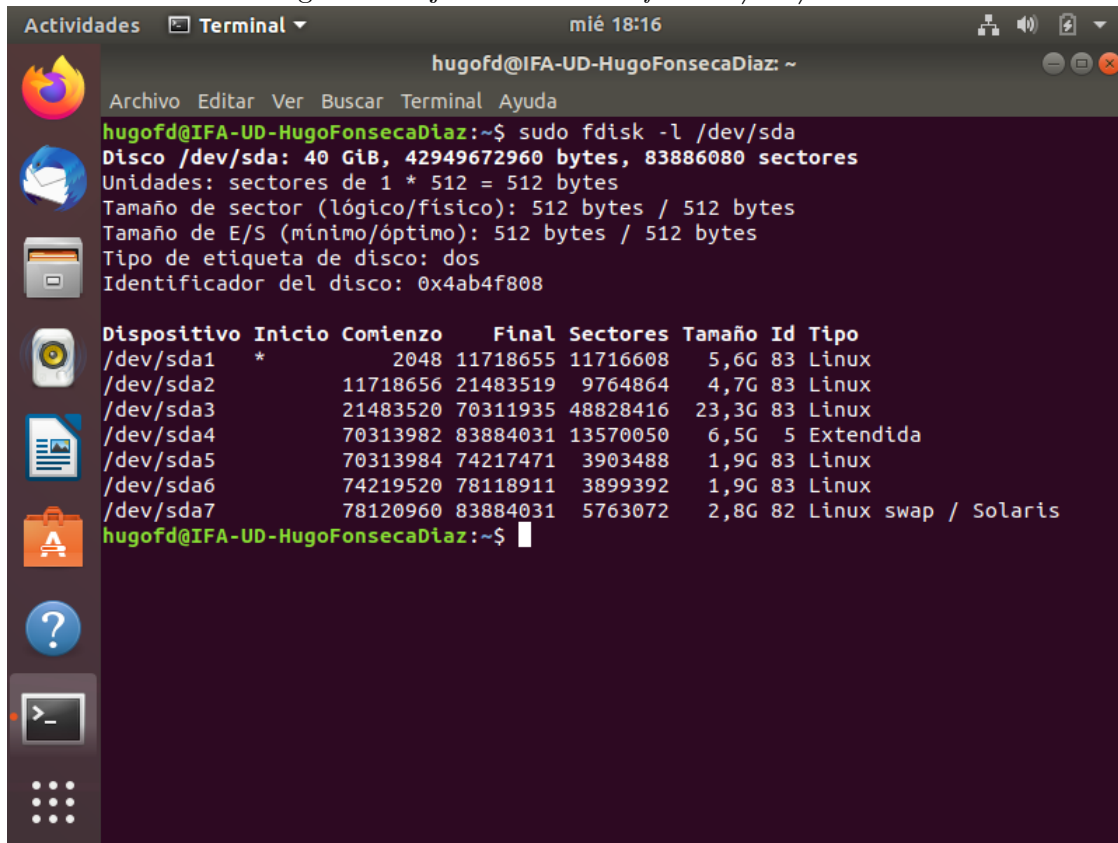
NAME	FSTYPE	LABEL	UUID	MOUNTPOINT
loop0	squash			/snap/gnome-calculator
loop1	squash			/snap/gnome-characters
loop2	squash			/snap/gnome-3-28-1804/
loop3	squash			/snap/gnome-logs/61
loop5	squash			/snap/core18/1988
loop6	squash			/snap/gnome-logs/103
loop8	squash			/snap/gnome-characters
loop9	squash			/snap/gtk-common-theme
loop10	squash			/snap/core/10908
loop11	squash			/snap/gnome-system-mon
loop12	squash			/snap/core18/2066
loop13	squash			/snap/core/11081
loop14	squash			/snap/gnome-calculator
loop15	squash			/snap/gnome-system-mon
loop16	squash			/snap/gtk-common-theme
loop17	squash			/snap/gnome-3-28-1804/
loop18	squash			/snap/gnome-3-34-1804/
sda				
sda1	ext4		759ff7ec-daed-4891-a63b-bae2d8cfded7	/
sda2	ext4		6190bd07-00fb-4169-ab91-b398279393d5	/usr
sda3	ext4		0600aacf-263a-4749-a486-b8e5f6b55d24	/home
sda4				
sda5	ext4		9cdbdd80-929b-4112-b1b1-1f735778a87e	/tmp
sda6	ext4		b867afb5-f932-4390-a2aa-d022c2e4fe63	/var
sda7	swap		e4cf4da7-51ae-4b28-84ef-874a4d672203	[SWAP]
sr0				

```
hugofd@IFA-UD-HugoFonsecaDiaz:~$
```

11. Ejercicio 11

Para mostrar las particiones del disco *sda* junto a sus sectores de inicio y fin, se utiliza el comando *fdisk* con la opción *l*, que lista dichas particiones, y pasándole como parámetro el disco que queremos inspeccionar (en este caso */dev/sda*). No es necesario especificarle que las unidades del tamaño sean sectores puesto que es el comportamiento por defecto.

Figura 16: Ejercicio 11: `sudo fdisk -l /dev/sda`.

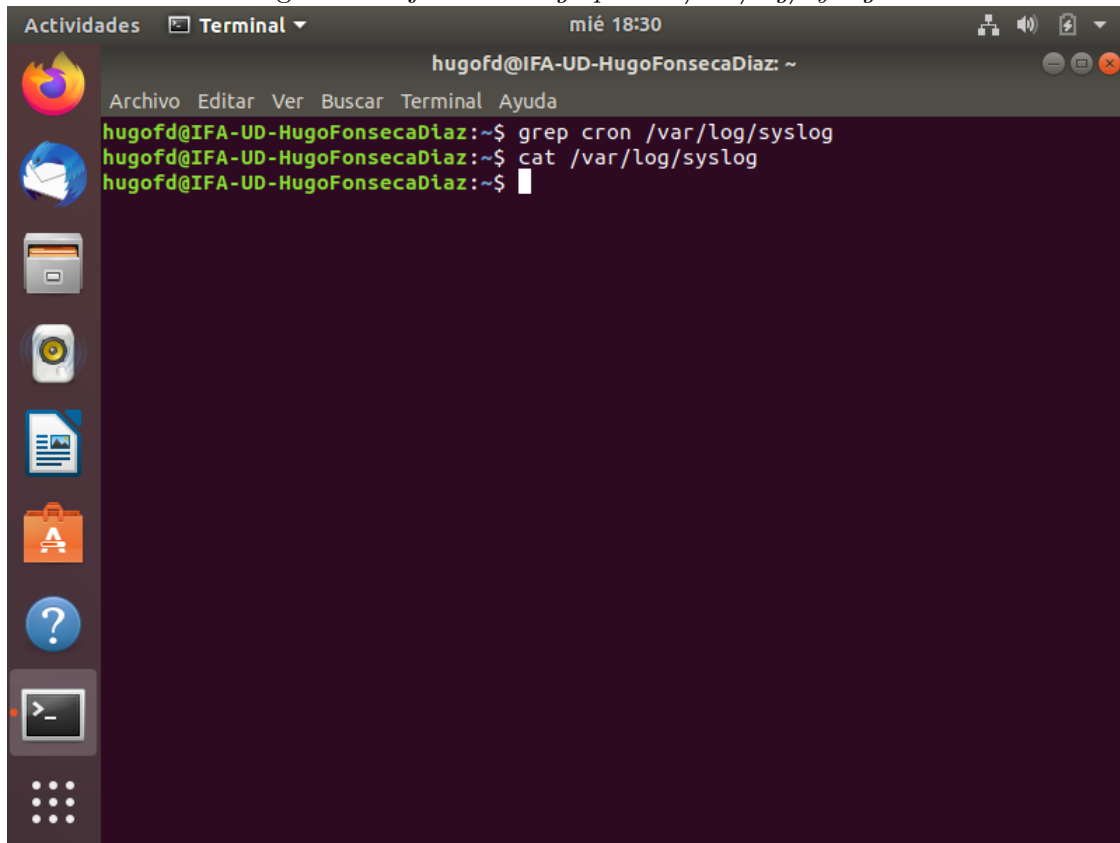


```
hugofd@IFA-UD-HugoFonsecaDiaz: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
hugofd@IFA-UD-HugoFonsecaDiaz:~$ sudo fdisk -l /dev/sda  
Disco /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectores  
Unidades: sectores de 1 * 512 = 512 bytes  
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes  
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes  
Tipo de etiqueta de disco: dos  
Identificador del disco: 0x4ab4f808  
  
Dispositivo Inicio Comienzo Final Sectores Tamaño Id Tipo  
/dev/sda1 * 2048 11718655 11716608 5,6G 83 Linux  
/dev/sda2 11718656 21483519 9764864 4,7G 83 Linux  
/dev/sda3 21483520 70311935 48828416 23,3G 83 Linux  
/dev/sda4 70313982 83884031 13570050 6,5G 5 Extendida  
/dev/sda5 70313984 74217471 3903488 1,9G 83 Linux  
/dev/sda6 74219520 78118911 3899392 1,9G 83 Linux  
/dev/sda7 78120960 83884031 5763072 2,8G 82 Linux swap / Solaris  
hugofd@IFA-UD-HugoFonsecaDiaz:~$
```

12. Ejercicio 12

El fichero donde el kernel almacena las acciones realizadas por `cron` se encuentra en `/var/log/syslog`. Puede hacerse un `grep` con la string `cron` en dicho archivo para visualizar las acciones, sin embargo, debido al poco espacio en la partición `/var`, en nuestro caso ese archivo está vacío.

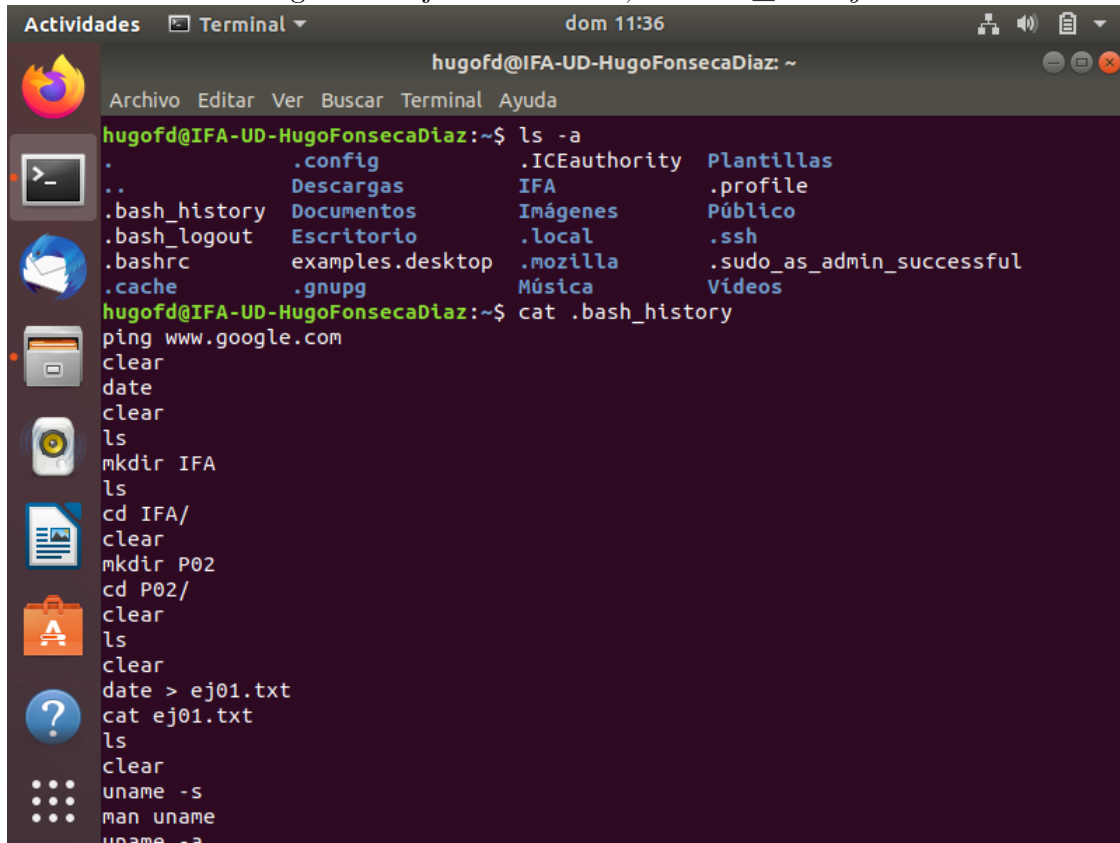
Figura 17: Ejercicio 12: *grep cron /var/log/syslog*.



13. Ejercicio 24

El historial de comandos de **bash** del usuario se encuentra en un fichero oculto de su carpeta *HOME* llamado *bash_history*. Se puede utilizar el comando **ls** con la flag *a* para listar todos los archivos, incluidos los ocultos, para comprobar que efectivamente existe el archivo del historial.

Figura 18: Ejercicio 24: `ls -a`; `cat .bash_history`.



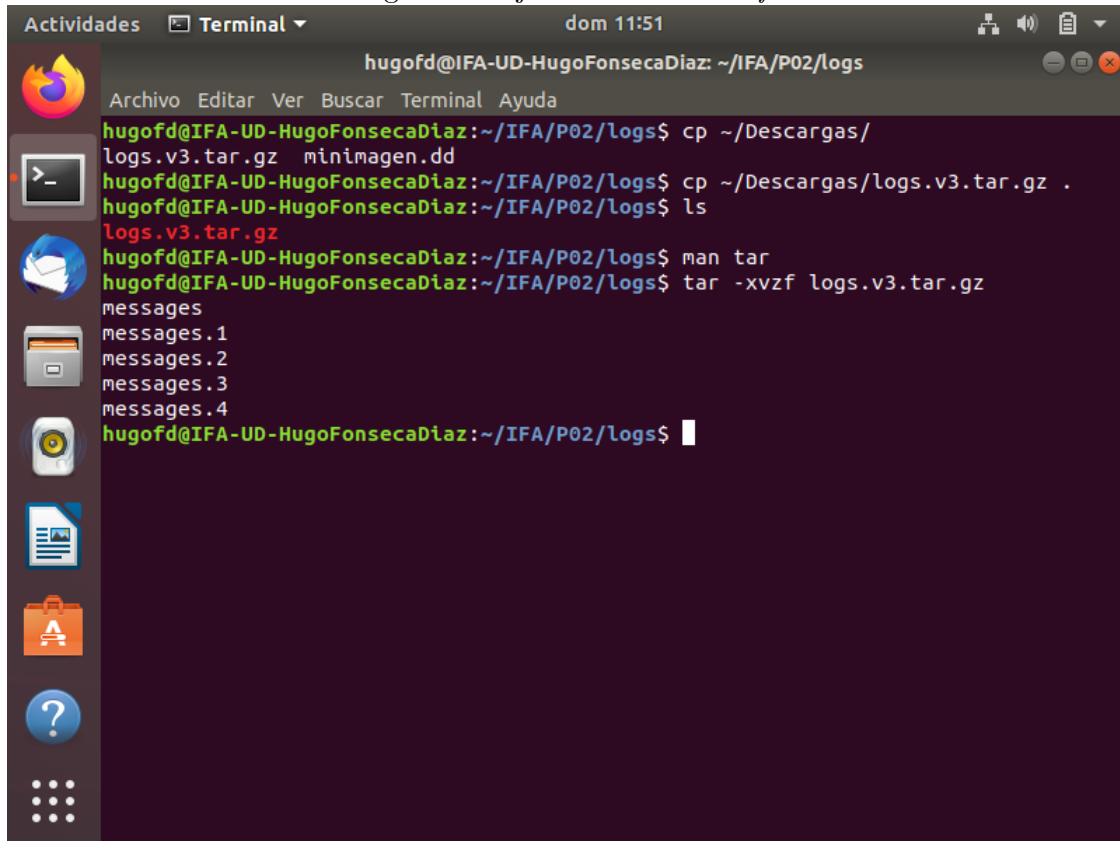
The screenshot shows a terminal window titled "hugofd@IFA-UD-HugoFonsecaDiaz: ~". The terminal displays the output of the command `ls -a`, which lists hidden files and directories. The output is as follows:

```
hugofd@IFA-UD-HugoFonsecaDiaz:~$ ls -a
.          .config      .ICEauthority  Plantillas
..         Descargas   IFA             .profile
.bash_history Documentos     Imágenes        Público
.bash_logout Escritorio     .local          .ssh
.bashrc     examples.desktop .mozilla        .sudo_as_admin_successful
.cache      .gnupg        Música          Videos
hugofd@IFA-UD-HugoFonsecaDiaz:~$ cat .bash_history
ping www.google.com
clear
date
clear
ls
mkdir IFA
ls
cd IFA/
clear
mkdir P02
cd P02/
clear
ls
clear
date > ej01.txt
cat ej01.txt
ls
clear
uname -s
man uname
uname -a
```

14. Ejercicio 27

Se descomprime el archivo con el comando `tar` y las flags `xvzf`, siendo `x` una indicación de que se quiere extraer los contenidos del archivo comprimido, `v` para que lo haga de manera verbosa, `z` para indicarle al comando que el archivo es un zip y `f` para pasarle el fichero que se desea extraer al comando.

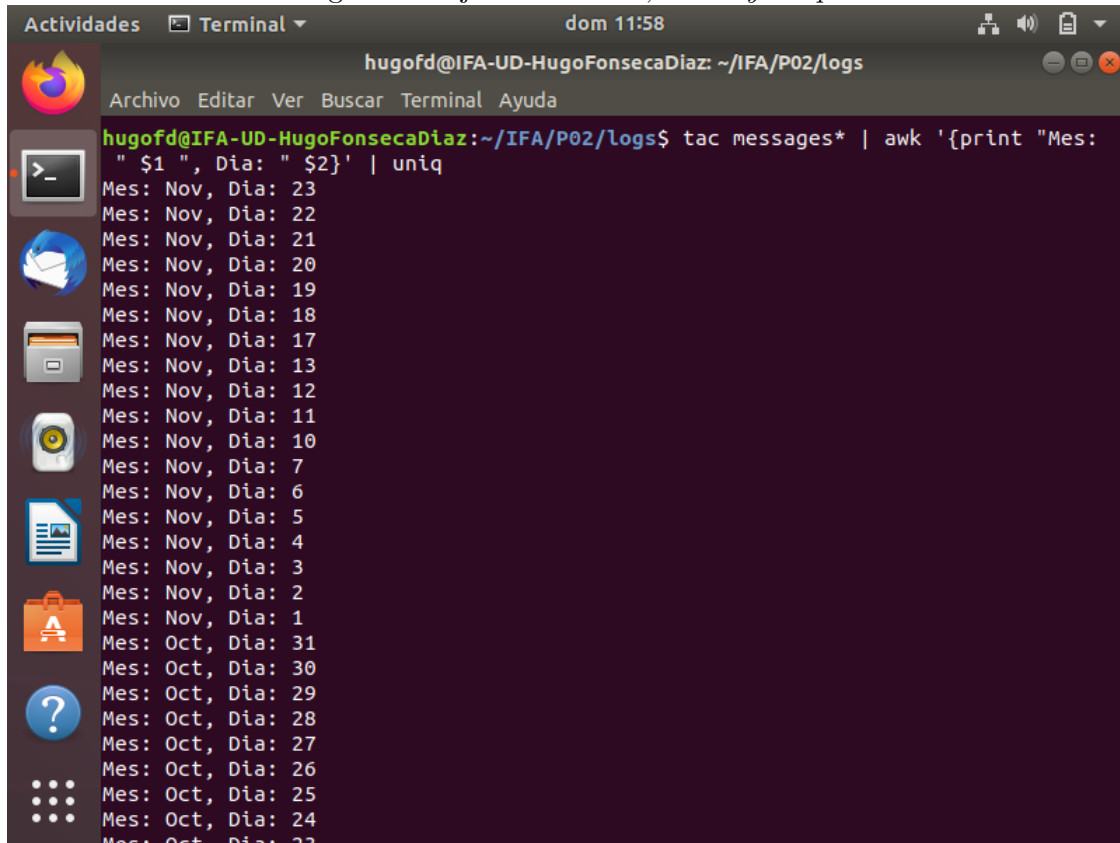
Figura 19: Ejercicio 27: *tar -xvzf*.



```
hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02/logs
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/logs$ cp ~/Descargas/
logs.v3.tar.gz minimagen.dd
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/logs$ cp ~/Descargas/logs.v3.tar.gz .
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/logs$ ls
logs.v3.tar.gz
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/logs$ man tar
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/logs$ tar -xvzf logs.v3.tar.gz
messages
messages.1
messages.2
messages.3
messages.4
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/logs$
```

Una vez descomprimidos los ficheros de texto, se procede a utilizar tres nuevas herramientas. Se usa `tac` para concatenar ficheros de forma inversa (es el comando `cat` invertido), el lenguaje de programación AWK para procesar texto y el comando `uniq` para omitir líneas repetidas.

Figura 20: Ejercicio 27: *tac*, *AWK* y *uniq*.

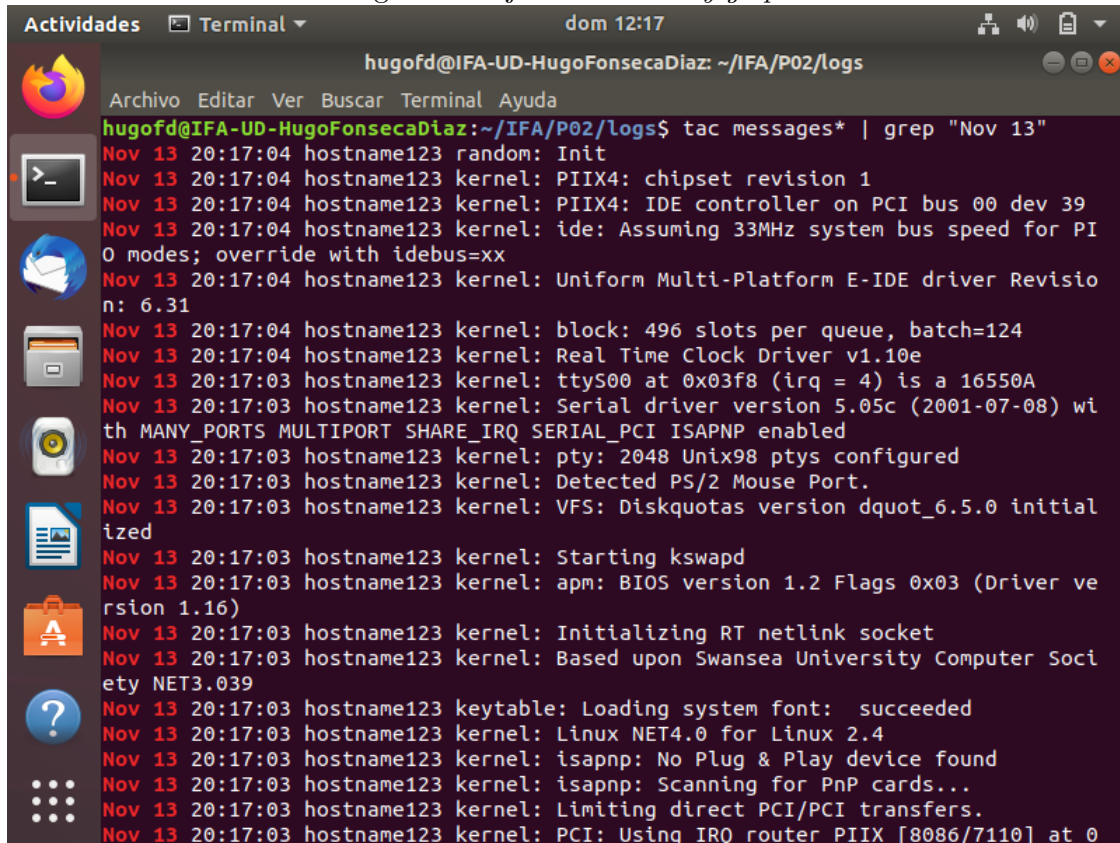


```
hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02/logs
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/logs$ tac messages* | awk '{print "Mes: " $1 " , Dia: " $2}' | uniq
Mes: Nov, Dia: 23
Mes: Nov, Dia: 22
Mes: Nov, Dia: 21
Mes: Nov, Dia: 20
Mes: Nov, Dia: 19
Mes: Nov, Dia: 18
Mes: Nov, Dia: 17
Mes: Nov, Dia: 13
Mes: Nov, Dia: 12
Mes: Nov, Dia: 11
Mes: Nov, Dia: 10
Mes: Nov, Dia: 7
Mes: Nov, Dia: 6
Mes: Nov, Dia: 5
Mes: Nov, Dia: 4
Mes: Nov, Dia: 3
Mes: Nov, Dia: 2
Mes: Nov, Dia: 1
Mes: Oct, Dia: 31
Mes: Oct, Dia: 30
Mes: Oct, Dia: 29
Mes: Oct, Dia: 28
Mes: Oct, Dia: 27
Mes: Oct, Dia: 26
Mes: Oct, Dia: 25
Mes: Oct, Dia: 24
Mes: Oct, Dia: 23
```

15. Ejercicio 28

Se usan los comandos *tac* y *grep*. El primero se usa para concatenar inversamente los ficheros de los mensajes y el segundo para buscar las líneas donde aparece la cadena de texto *"Nov 13"*.

Figura 21: Ejercicio 28: *tac* y *grep*.

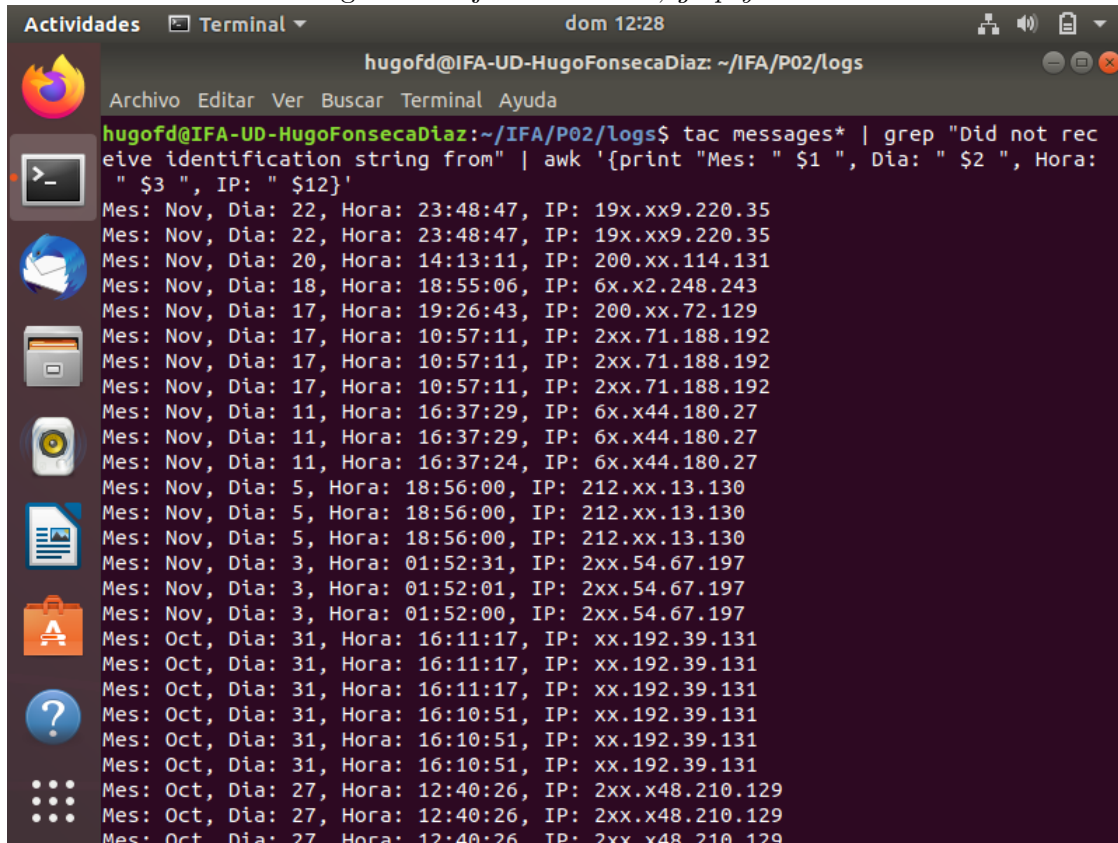


```
hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02/logs
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/logs$ tac messages* | grep "Nov 13"
Nov 13 20:17:04 hostname123 random: Init
Nov 13 20:17:04 hostname123 kernel: PIIX4: chipset revision 1
Nov 13 20:17:04 hostname123 kernel: PIIX4: IDE controller on PCI bus 00 dev 39
Nov 13 20:17:04 hostname123 kernel: ide: Assuming 33MHz system bus speed for PI
0 modes; override with idebus=xx
Nov 13 20:17:04 hostname123 kernel: Uniform Multi-Platform E-IDE driver Revisio
n: 6.31
Nov 13 20:17:04 hostname123 kernel: block: 496 slots per queue, batch=124
Nov 13 20:17:04 hostname123 kernel: Real Time Clock Driver v1.10e
Nov 13 20:17:03 hostname123 kernel: ttyS00 at 0x03f8 (irq = 4) is a 16550A
Nov 13 20:17:03 hostname123 kernel: Serial driver version 5.05c (2001-07-08) wi
th MANY_PORTS MULTIPOINT SHARE_IRQ SERIAL_PCI ISAPNP enabled
Nov 13 20:17:03 hostname123 kernel: pty: 2048 Unix98 ptys configured
Nov 13 20:17:03 hostname123 kernel: Detected PS/2 Mouse Port.
Nov 13 20:17:03 hostname123 kernel: VFS: Diskquotas version dquot_6.5.0 initial
ized
Nov 13 20:17:03 hostname123 kernel: Starting kswapd
Nov 13 20:17:03 hostname123 kernel: apm: BIOS version 1.2 Flags 0x03 (Driver ve
rsion 1.16)
Nov 13 20:17:03 hostname123 kernel: Initializing RT netlink socket
Nov 13 20:17:03 hostname123 kernel: Based upon Swansea University Computer Soci
ety NET3.039
Nov 13 20:17:03 hostname123 keytable: Loading system font: succeeded
Nov 13 20:17:03 hostname123 kernel: Linux NET4.0 for Linux 2.4
Nov 13 20:17:03 hostname123 kernel: isapnp: No Plug & Play device found
Nov 13 20:17:03 hostname123 kernel: isapnp: Scanning for PnP cards...
Nov 13 20:17:03 hostname123 kernel: Limiting direct PCI/PCI transfers.
Nov 13 20:17:03 hostname123 kernel: PCI: Using IRQ router PIIX [8086/7110] at 0
```

16. Ejercicio 29

Se usan tres herramientas. La primera es **tac**, para concatenar inversamente los ficheros de los mensajes. La segunda es **grep**, para buscar entre los mensajes aquellos con la string indicada por el enunciado. Por último, se utiliza el lenguaje de procesamiento de textos **AWK** para printear las columnas deseadas, en este caso con un título indicativo.

Figura 22: Ejercicio 29: *tac*, *grep* y *awk*.

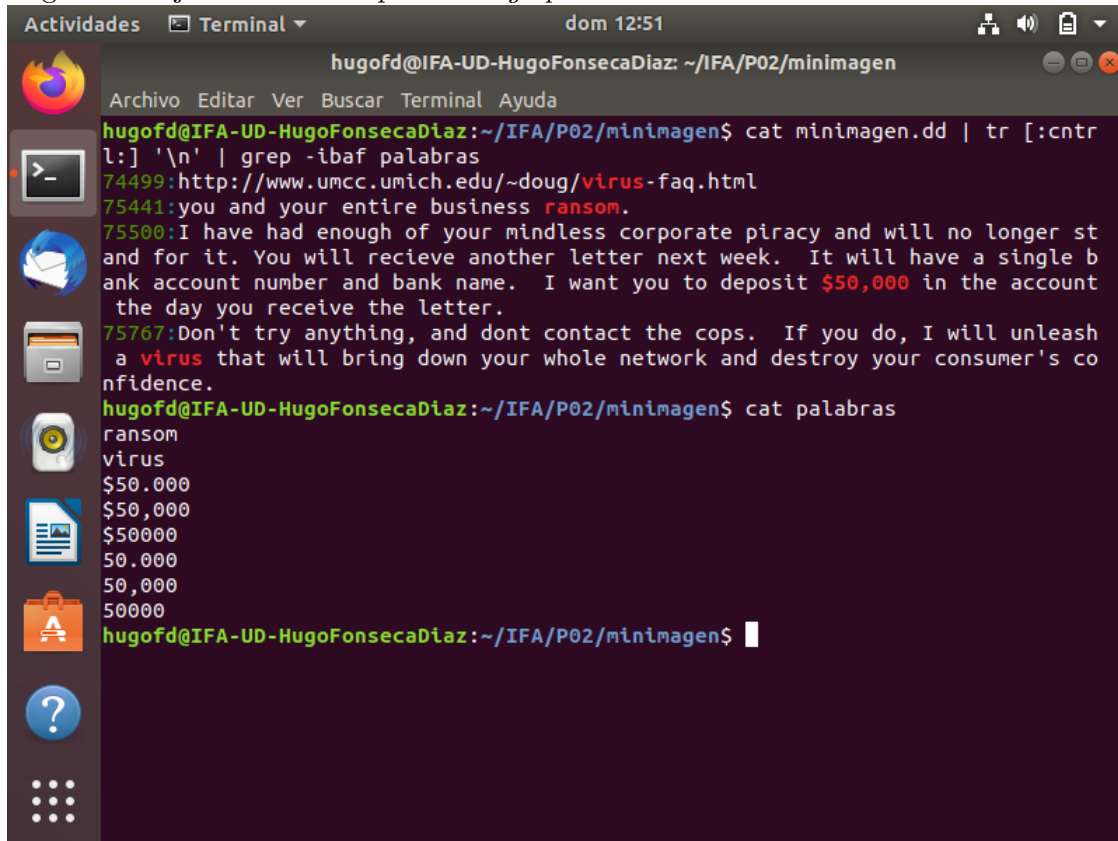


```
hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02/logs
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/logs$ tac messages* | grep "Did not receive identification string from" | awk '{print "Mes: " $1 ", Dia: " $2 ", Hora: " $3 ", IP: " $12}'
Mes: Nov, Dia: 22, Hora: 23:48:47, IP: 19x.xx9.220.35
Mes: Nov, Dia: 22, Hora: 23:48:47, IP: 19x.xx9.220.35
Mes: Nov, Dia: 20, Hora: 14:13:11, IP: 200.xx.114.131
Mes: Nov, Dia: 18, Hora: 18:55:06, IP: 6x.x2.248.243
Mes: Nov, Dia: 17, Hora: 19:26:43, IP: 200.xx.72.129
Mes: Nov, Dia: 17, Hora: 10:57:11, IP: 2xx.71.188.192
Mes: Nov, Dia: 17, Hora: 10:57:11, IP: 2xx.71.188.192
Mes: Nov, Dia: 17, Hora: 10:57:11, IP: 2xx.71.188.192
Mes: Nov, Dia: 11, Hora: 16:37:29, IP: 6x.x44.180.27
Mes: Nov, Dia: 11, Hora: 16:37:29, IP: 6x.x44.180.27
Mes: Nov, Dia: 11, Hora: 16:37:24, IP: 6x.x44.180.27
Mes: Nov, Dia: 5, Hora: 18:56:00, IP: 212.xx.13.130
Mes: Nov, Dia: 5, Hora: 18:56:00, IP: 212.xx.13.130
Mes: Nov, Dia: 5, Hora: 18:56:00, IP: 212.xx.13.130
Mes: Nov, Dia: 3, Hora: 01:52:31, IP: 2xx.54.67.197
Mes: Nov, Dia: 3, Hora: 01:52:01, IP: 2xx.54.67.197
Mes: Nov, Dia: 3, Hora: 01:52:00, IP: 2xx.54.67.197
Mes: Oct, Dia: 31, Hora: 16:11:17, IP: xx.192.39.131
Mes: Oct, Dia: 31, Hora: 16:11:17, IP: xx.192.39.131
Mes: Oct, Dia: 31, Hora: 16:11:17, IP: xx.192.39.131
Mes: Oct, Dia: 31, Hora: 16:10:51, IP: xx.192.39.131
Mes: Oct, Dia: 31, Hora: 16:10:51, IP: xx.192.39.131
Mes: Oct, Dia: 31, Hora: 16:10:51, IP: xx.192.39.131
Mes: Oct, Dia: 27, Hora: 12:40:26, IP: 2xx.x48.210.129
Mes: Oct, Dia: 27, Hora: 12:40:26, IP: 2xx.x48.210.129
Mes: Oct, Dia: 27, Hora: 12:40:26, IP: 2xx.x48.210.129
```

17. Ejercicio 30

Se define un fichero de palabras clave donde se almacenan los términos de la búsqueda. Después, se ejecuta el comando **grep** con las flags *-ibaf*, siendo *i* la opción para ignorar mayúsculas y minúsculas, *b* la que muestra el *byte offset* de la línea, *a* para especificarle al comando que trate el fichero binario como si fuera texto y *f* para obtener términos de búsqueda desde un fichero. Se utiliza el comando **tr** para traducir los caracteres de control a nueva línea.

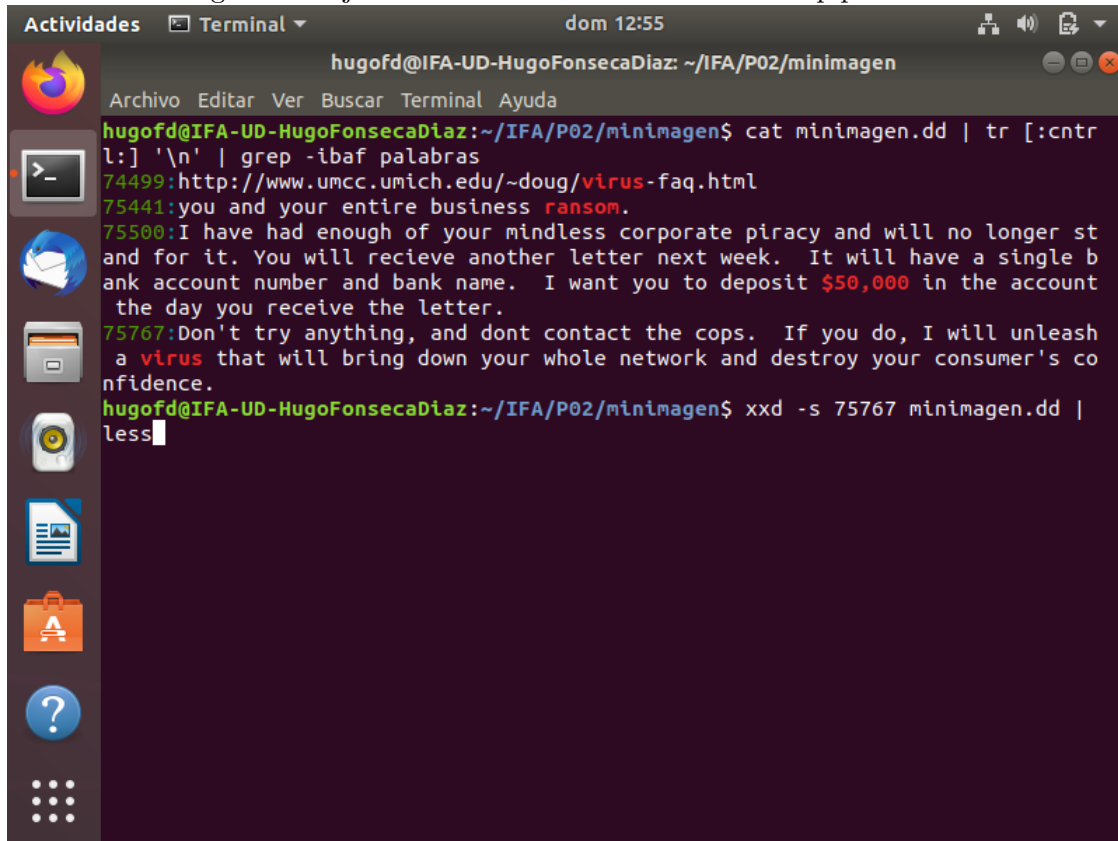
Figura 23: Ejercicio 30: Búsqueda con *grep*. Traducción de caracteres de control con *tr*



```
hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02/minimagen
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/minimagen$ cat minimagen.dd | tr [:cntrl:] '\n' | grep -ibaf palabras
74499:http://www.umcc.umich.edu/~doug/virus-faq.html
75441:you and your entire business ransom.
75500:I have had enough of your mindless corporate piracy and will no longer st
and for it. You will recieve another letter next week. It will have a single b
ank account number and bank name. I want you to deposit $50,000 in the account
the day you receive the letter.
75767:Don't try anything, and dont contact the cops. If you do, I will unleash
a virus that will bring down your whole network and destroy your consumer's co
nfidence.
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/minimagen$ cat palabras
ransom
virus
$50.000
$50,000
$50000
50.000
50,000
50000
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/minimagen$
```

Una vez encontradas las líneas donde aparecen instancias de los términos buscados, se puede pasar su offset al comando *xxd* con la flag *s*, que busca (seek) a partir del offset que recibe. Se usa además el comando *less* para poder visualizar la salida completa del comando.

Figura 24: Ejercicio 30: Comando *xxd -s* con una pipe a *less*



The screenshot shows a terminal window titled "hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02/minimagen". The terminal displays the following commands and output:

```
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/minimagen$ cat minimagen.dd | tr [:cntrl:] '\n' | grep -ibaf palabras
74499:http://www.umcc.umich.edu/~doug/virus-faq.html
75441:you and your entire business ransom.
75500:I have had enough of your mindless corporate piracy and will no longer st
and for it. You will recieve another letter next week. It will have a single b
ank account number and bank name. I want you to deposit $50,000 in the account
the day you receive the letter.
75767:Don't try anything, and dont contact the cops. If you do, I will unleash
a virus that will bring down your whole network and destroy your consumer's co
nfidence.
hugofd@IFA-UD-HugoFonsecaDiaz:~/IFA/P02/minimagen$ xxd -s 75767 minimagen.dd | less
```

The terminal window includes a sidebar with application icons (Firefox, Files, etc.) and a top bar with system information (date, time, and window controls).

Figura 25: Ejercicio 30: Visualización de la salida del comando `xxd -s` con una pipe a `less`

```

hugofd@IFA-UD-HugoFonsecaDiaz: ~/IFA/P02/minimagen
Archivo Editar Ver Buscar Terminal Ayuda
000127f7: 446f 6e27 7420 7472 7920 616e 7974 6869 Don't try anythi
00012807: 6e67 2c20 616e 6420 646f 6e74 2063 6f6e ng, and dont con
00012817: 7461 6374 2074 6865 2063 6f70 732e 2020 tact the cops.
00012827: 4966 2079 6f75 2064 6f2c 2049 2077 696c If you do, I wil
00012837: 6c20 756e 6c65 6173 6820 6120 7669 7275 l unleash a viru
00012847: 7320 7468 6174 2077 696c 6c20 6272 696e s that will brin
00012857: 6720 646f 776e 2079 6f75 7220 7768 6f6c g down your whol
00012867: 6520 6e65 7477 6f72 6b20 616e 6420 6465 e network and de
00012877: 7374 726f 7920 796f 7572 2063 6f6e 7375 stroy your consu
00012887: 6d65 7227 7320 636f 6e66 6964 656e 6365 mer's confidence
00012897: 2e20 200a 0a44 6f6e 2774 206d 6573 7320 . ..Don't mess
000128a7: 7769 7468 206d 6520 6f6e 2074 6869 7321 with me on this!
000128b7: 0a0a 0909 0979 6f75 7220 574f 7273 5420 .....your WORsT
000128c7: 4e69 6768 544d 6172 4545 450a 0a0a 0000 NighTMarEEE.....
000128d7: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000128e7: 0000 0000 0000 0000 0000 0000 0000 0000 .....
000128f7: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00012907: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00012917: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00012927: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00012937: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00012947: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00012957: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00012967: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00012977: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00012987: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00012997: 0000 0000 0000 0000 0000 0000 0000 0000 .....
:

```

18. Ejercicio 31

Se crea el caso en Autopsy con los datos solicitados.

Figura 26: Ejercicio 31: Creación del caso

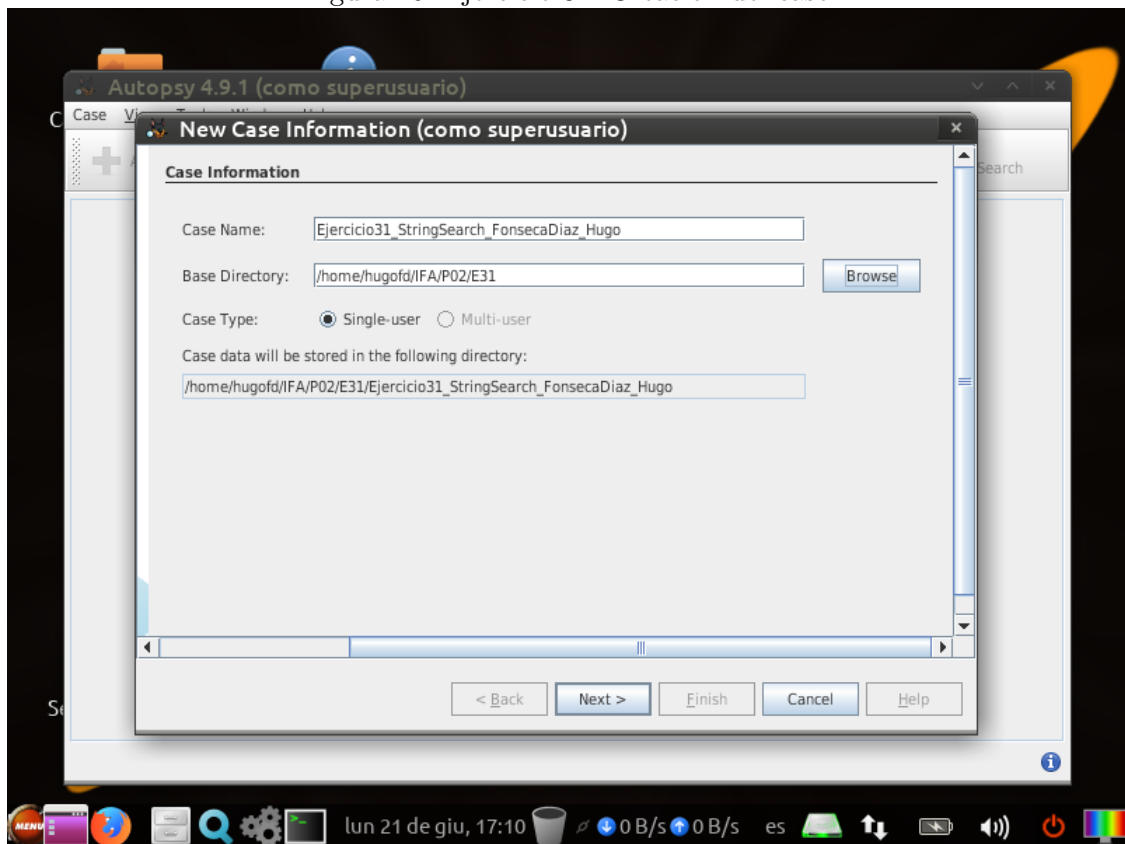
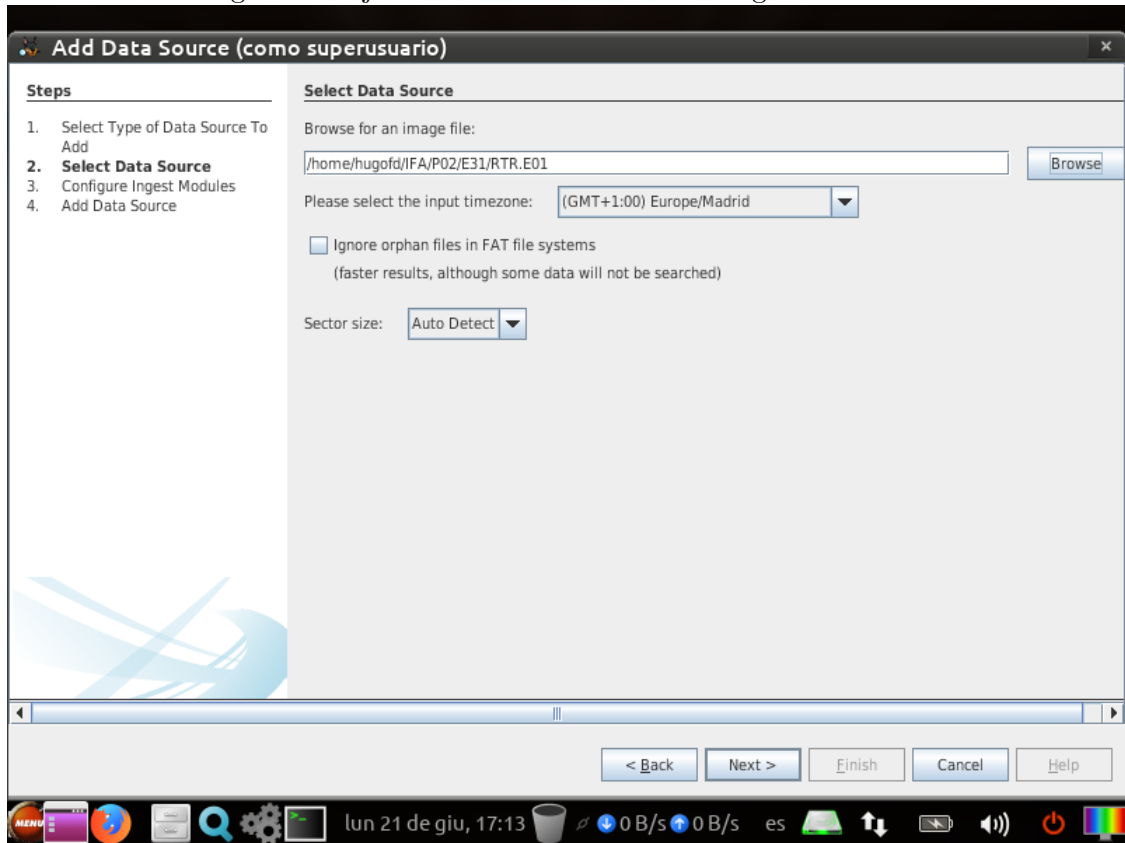


Figura 27: Ejercicio 31: Selección de la imagen a analizar



Se seleccionan los módulos y se configura el módulo de búsqueda de palabras clave.

Figura 28: Ejercicio 31: Palabras clave

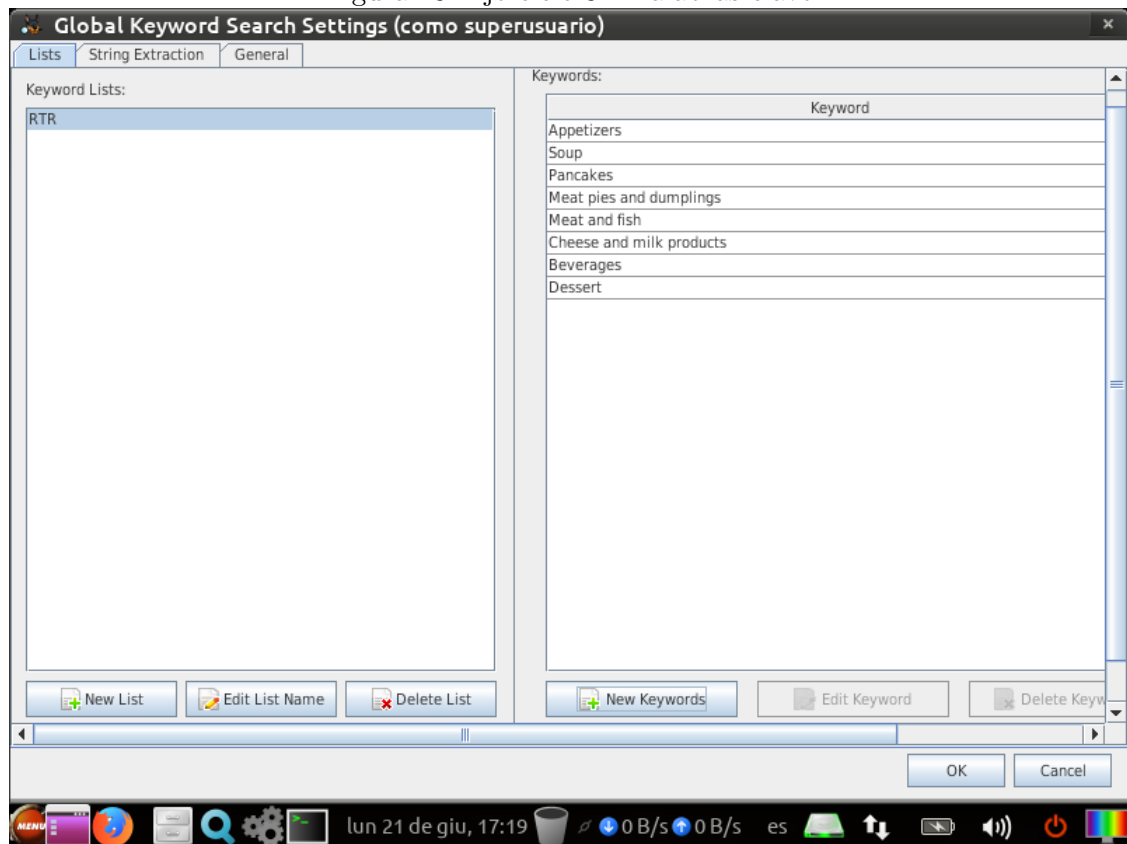


Figura 29: Ejercicio 31: Módulos seleccionados

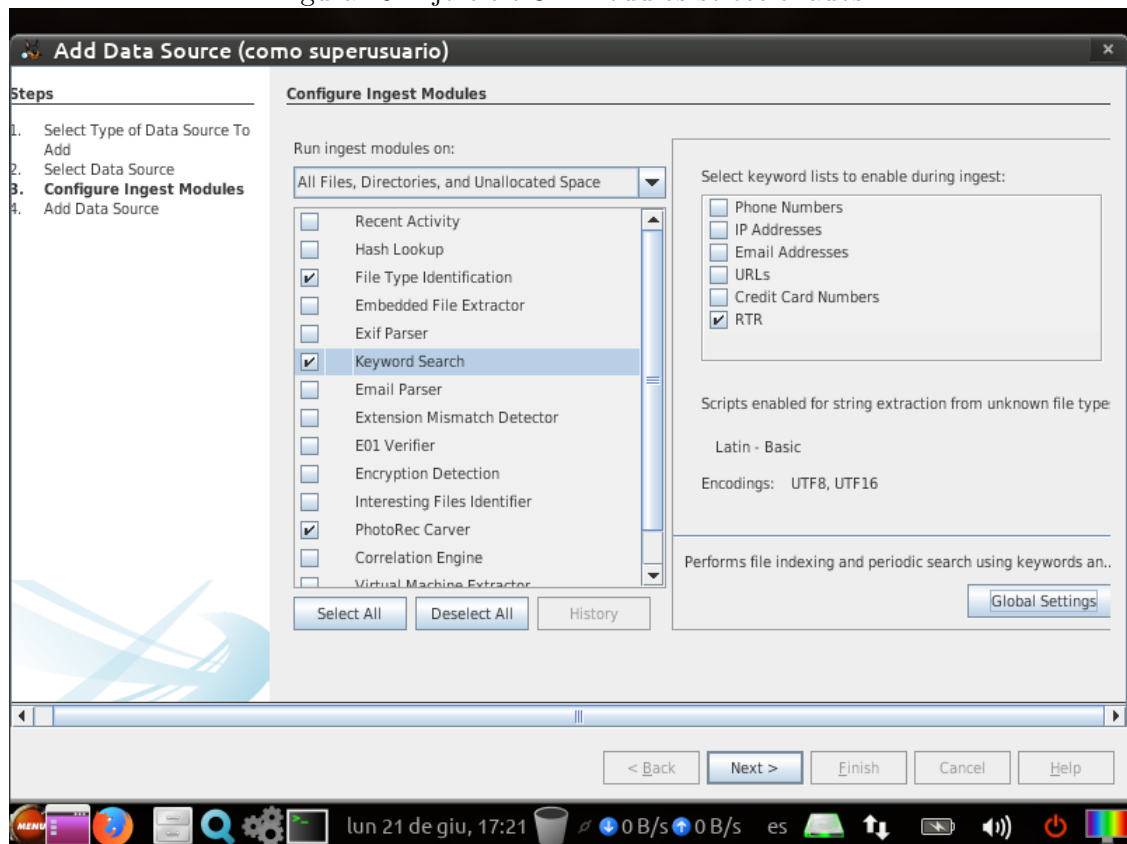
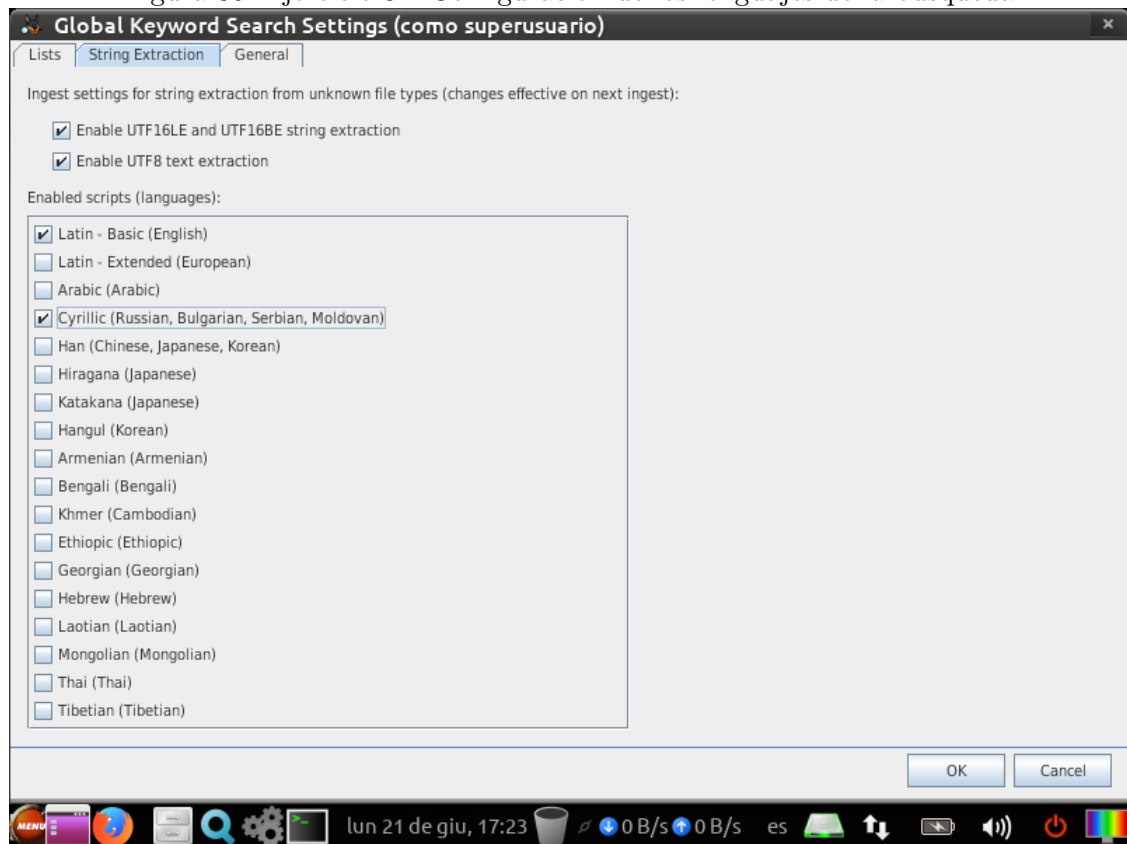
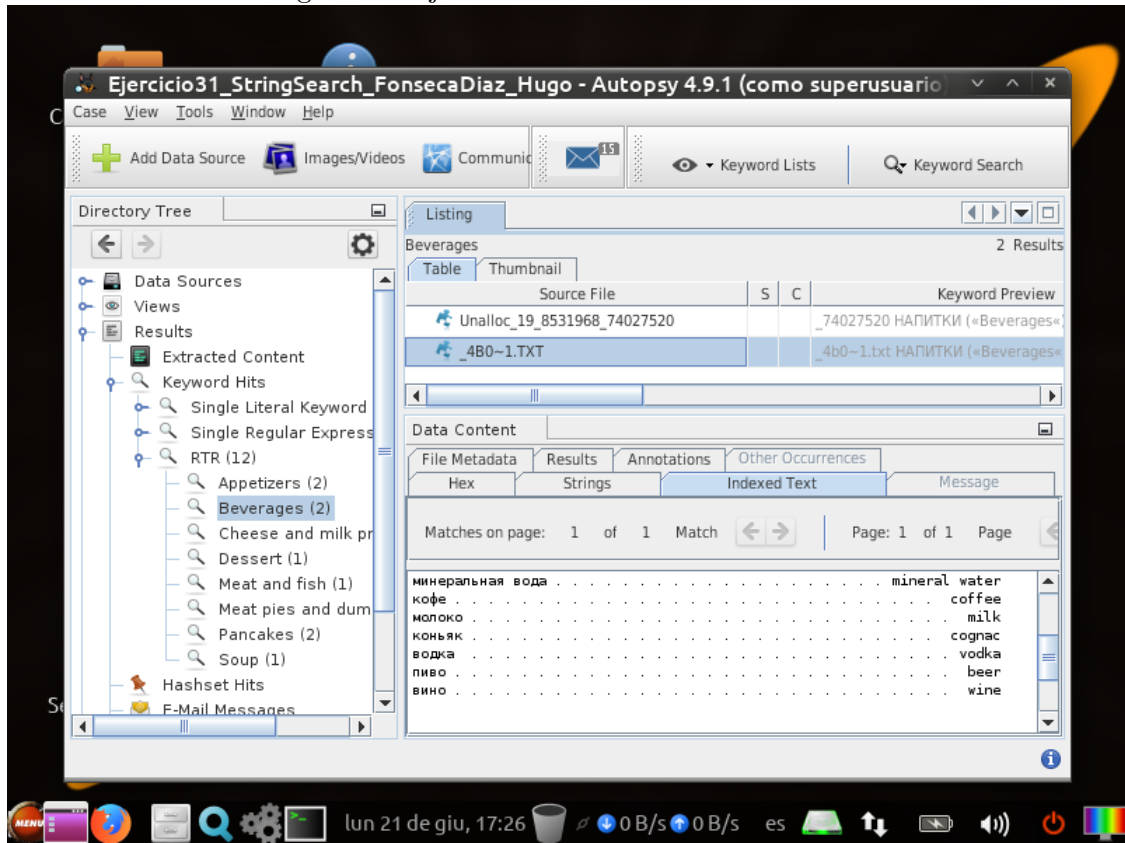


Figura 30: Ejercicio 31: Configuración de los lenguajes de la búsqueda



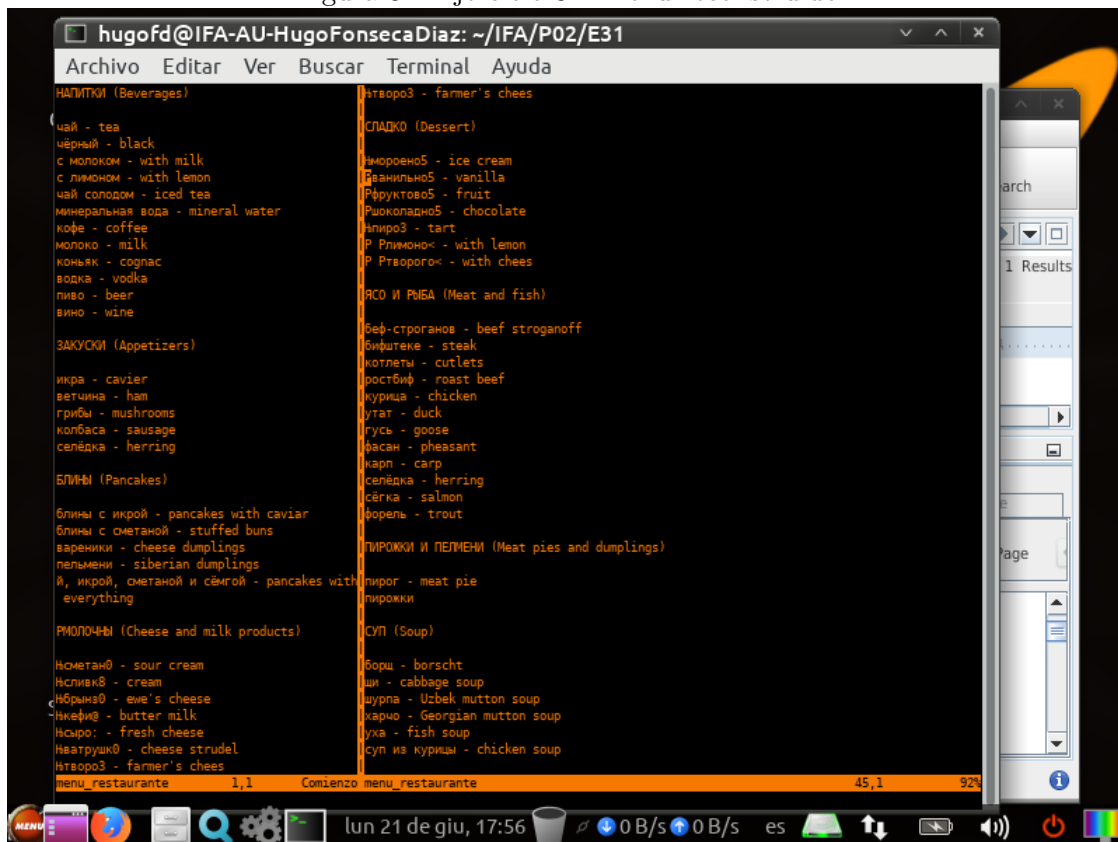
Una vez finalizado el análisis, se pueden observar los ficheros encontrados.

Figura 31: Ejercicio 31: Resultados del análisis



Se reconstruye el menú del restaurante, creado inicialmente el 3 de noviembre de 2004.

Figura 32: Ejercicio 31: Menú reconstruido



19. Ejercicio 32

Se crea el caso en Autopsy con los datos solicitados.

Figura 33: Ejercicio 32: Creación del caso

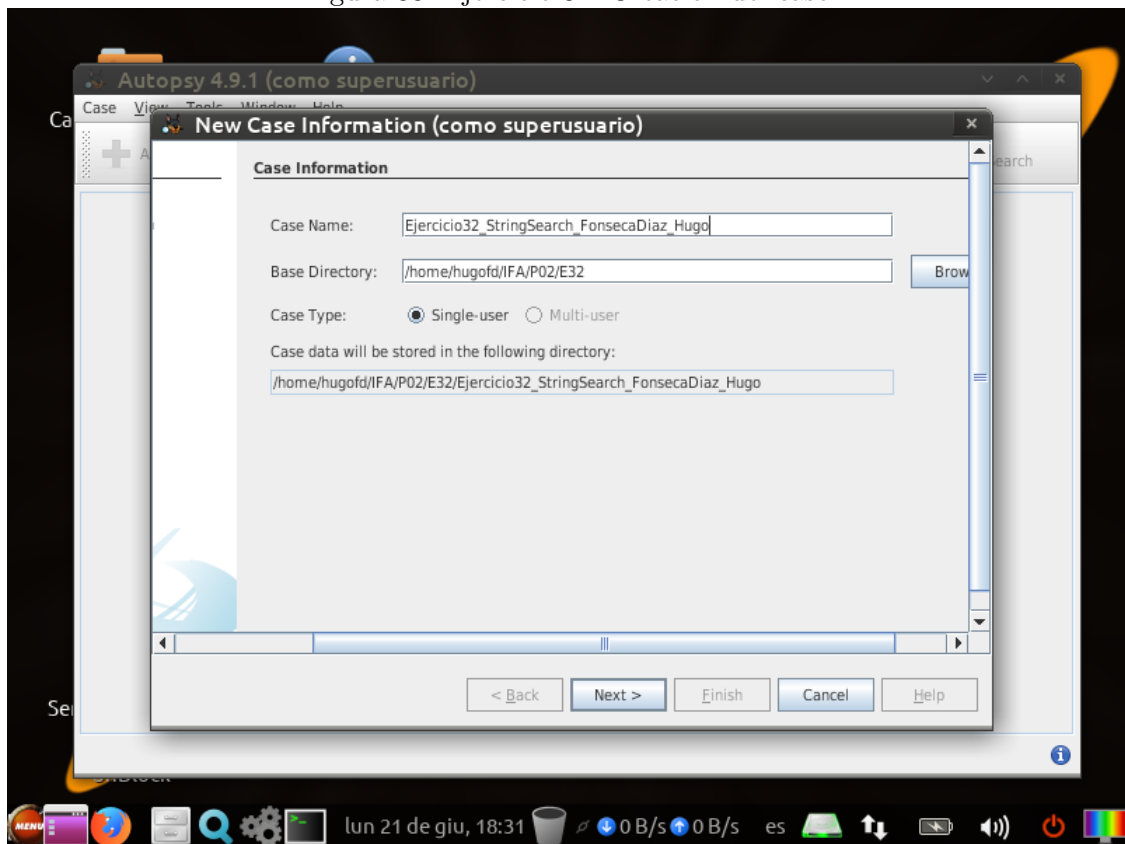
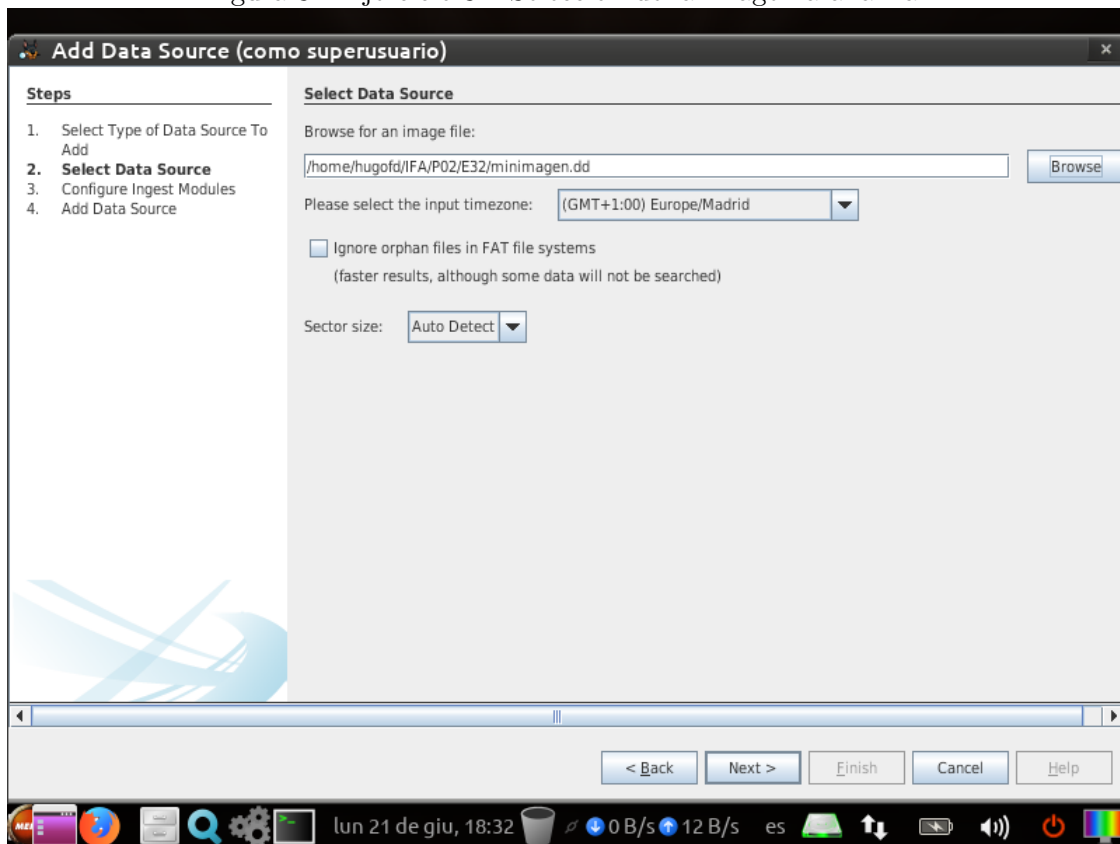
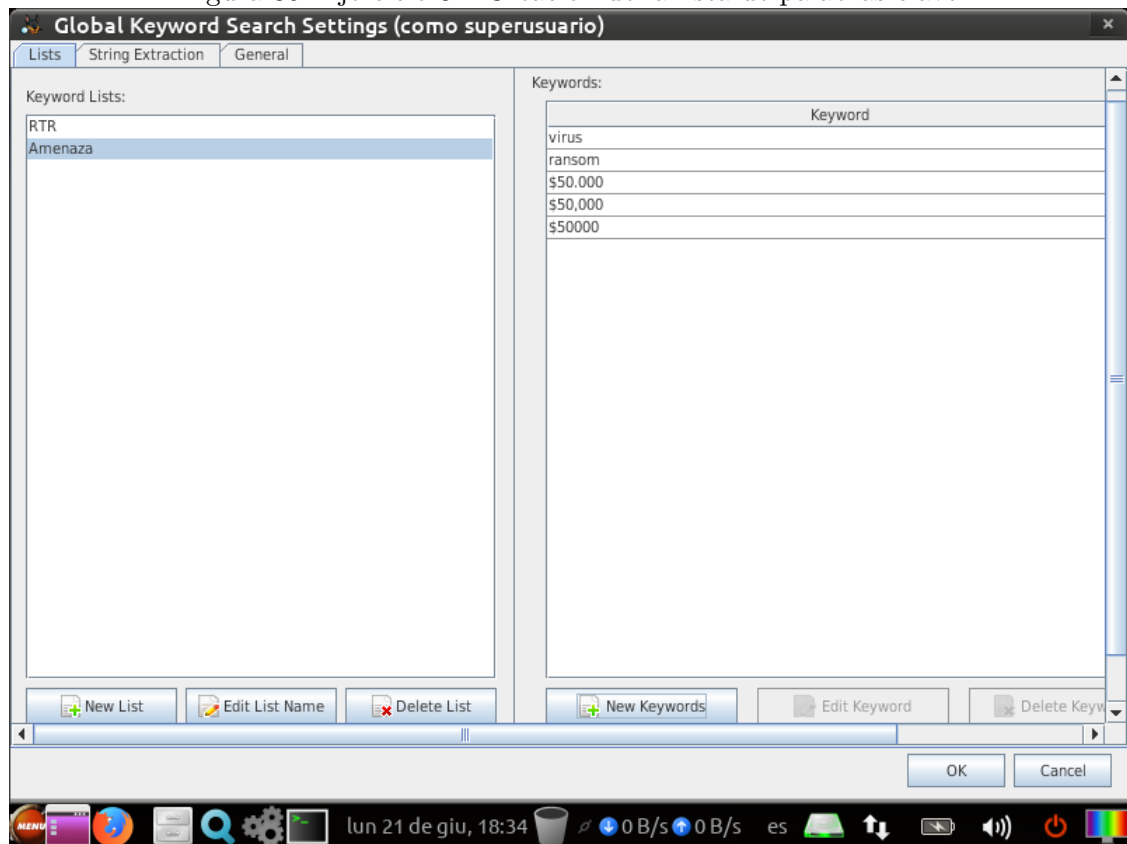


Figura 34: Ejercicio 32: Selección de la imagen a analizar



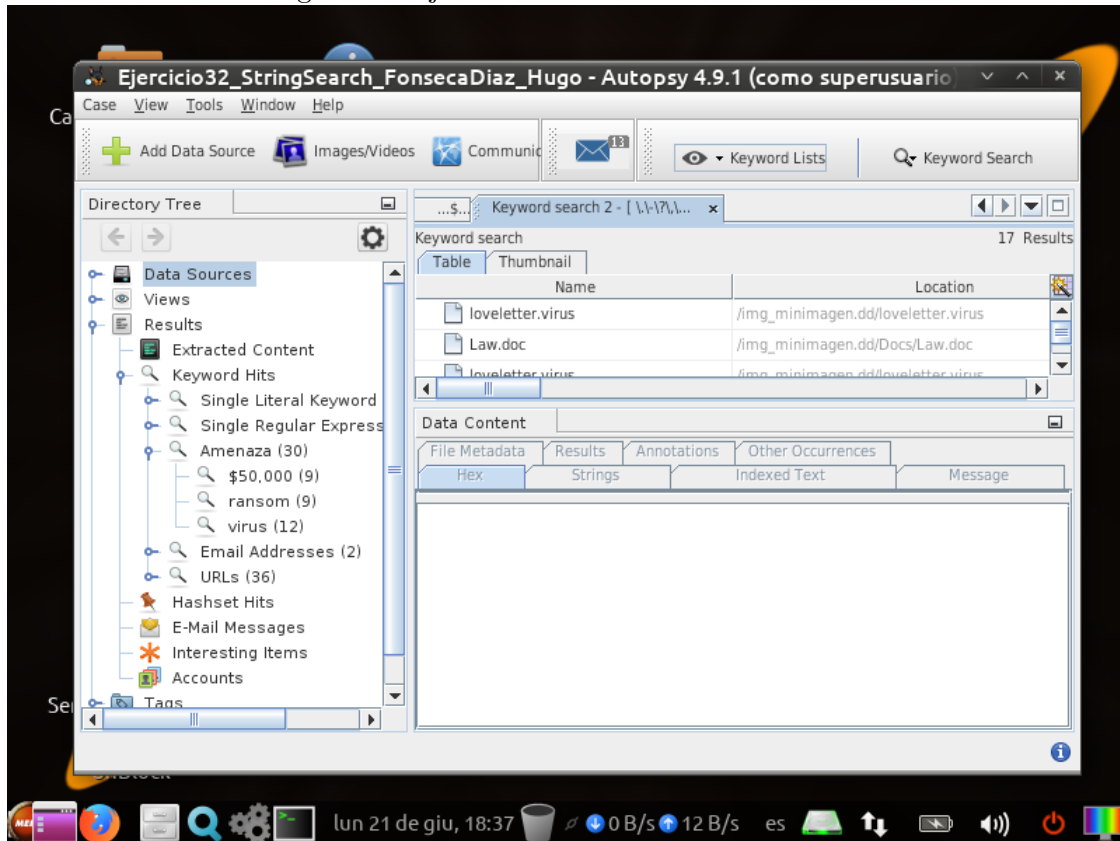
Se crea la lista de palabras clave y se seleccionan los módulos de identificación de tipo de fichero, búsqueda de palabras clave y *PhotoRec Carver*. A parte de la lista de palabras propia, se utilizan las autogeneradas por el módulo de búsqueda referentes a emails, IP y urls.

Figura 35: Ejercicio 32: Creación de la lista de palabras clave



Se produce el análisis de la imagen, cuyos resultados serán necesarios para responder a las preguntas del ejercicio.

Figura 36: Ejercicio 32: Resultados del análisis



19.1. Respuestas a las preguntas

La información a estas preguntas se encuentra en los resultados del análisis y en los metadatos de los ficheros.

- a) Hay 36 URLs con referencias a los ficheros *whyhack* y *loveletter.virus*.
- b) ReyHalif.doc
- c) 725 bytes.
- d) 147.
- e) 2.
- f) Reynolds-Halifax.
- g) your WOrsT NighTMarEEE.
- h) Content Encoding: ISO-8859-1. Content Type: text/plain; charset=ISO-8859-1.
- i) 23 de septiembre del 2000.

Referencias