Big Data

2025/26

Ciclo	Big Data Aplicado y Sistemas de Big Data
Nombre	Carmen García Rodríguez
Correo	carmengr36@educastur.es
Nº Unidad Didáctica	1

Tabla de Contenido

1. Comandos de Información y Preparación

1. Identidad del Usuario: Abre una terminal y ejecuta un comando para saber qué usuario eres y a qué grupos perteneces.

```
carmen@servidor:~$ whoami; groups
carmen
carmen adm cdrom sudo dip plugdev lxd
```

2. Usuarios Conectados: Muestra quién está conectado actualmente al sistema. Luego, ejecuta otro comando que te dé información más detallada, como el tiempo que llevan conectados y qué están ejecutando.

```
carmen@servidor:~$ who; w
carmen   tty1         2025-10-27 09:01
carmen   pts/0        2025-10-27 09:06 (10.140.42.207)
 09:06:13 up 8 min,  2 users,  load average: 0,00, 0,00, 0,00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU  WHAT
carmen           10.140.42.207    09:06    7:57   0.00s  0.01s sshd: ca
carmen   tty1     -                09:01   47.00s  0.06s  0.03s -bash
```

3. Historial de Conexiones: Lista los últimos inicios de sesión en el sistema.

```
carmen@servidor:~$ last
carmen   pts/0        10.140.42.207    Mon Oct 27 09:06   still logged in
reboot   system boot  6.8.0-86-generic Mon Oct 27 08:58   still running
reboot   system boot  6.8.0-86-generic Mon Oct 27 08:55 - 08:56  (00:01)
reboot   system boot  6.8.0-86-generic Thu Oct 23 10:09 - 10:24  (00:14)
carmen   pts/0        10.140.42.207    Thu Oct 23 07:37 - 10:08  (02:31)
reboot   system boot  6.8.0-85-generic Thu Oct 23 07:35 - 10:08  (02:33)
carmen   pts/0        10.140.42.207    Wed Oct 22 09:50 - 10:25  (00:34)
reboot   system boot  6.8.0-85-generic Wed Oct 22 08:45 - 10:25  (01:39)
reboot   system boot  6.8.0-85-generic Mon Oct 20 06:34 - 08:44 (1+02:10)
reboot   system boot  6.8.0-85-generic Thu Oct 16 06:42 - 08:44 (5+02:02)
reboot   system boot  6.8.0-85-generic Wed Oct 15 09:46 - 10:23  (00:37)

wtmp begins Wed Oct 15 09:46:24 2025
carmen@servidor:~$ lastb
lastb: cannot open /var/log/btmp: Permission denied
carmen@servidor:~$ sudo lastb
[sudo] password for carmen:
carmen   ssh:notty    10.140.42.207    Wed Oct 22 10:21 - 10:21  (00:00)
carmen   ssh:notty    10.140.42.207    Wed Oct 22 10:21 - 10:21  (00:00)

btmp begins Wed Oct 22 10:21:29 2025
```

4. Crear Entorno de Trabajo: En tu directorio personal (/home/tu_usuario), crea una carpeta principal para todos los ejercicios llamada practicas_linux .

```
carmen@servidor:~$ mkdir /home/carmen/practicas_linux
```

5. Estructura de Directorios: Dentro de practicas_linux , crea la siguiente estructura de directorios: proyectos , documentos y scripts .

```
carmen@servidor:~/practicas_linux$ mkdir proyectos documentos scripts
carmen@servidor:~/practicas_linux$ ls -l
total 12
drwxrwxr-x 2 carmen carmen 4096 oct 27 09:12 documentos
drwxrwxr-x 2 carmen carmen 4096 oct 27 09:12 proyectos
drwxrwxr-x 2 carmen carmen 4096 oct 27 09:12 scripts
```

2. Gestión de Usuarios y Grupos

1. Crear Grupos: Crea tres nuevos grupos en el sistema: desarrolladores , analistas y becarios .

```
carmen@servidor:~/practicas_linux$ sudo groupadd desarrolladores
carmen@servidor:~/practicas_linux$ sudo groupadd analistas
carmen@servidor:~/practicas_linux$ sudo groupadd becarios
```

2. Verificar Grupos: Confirma que los grupos se han creado correctamente buscando sus nombres en el archivo /etc/group .

```
carmen@servidor:~/practicas_linux$ grep -E "desarrolladores|analistas|becarios" /etc/group
desarrolladores:x:1001:
analistas:x:1002:
becarios:x:1003:
```

3. Crear un Usuario Básico: Crea un nuevo usuario llamado juan .

```
carmen@servidor:~/practicas_linux$ sudo useradd -m juan
```

4. Crear Usuario con Grupo Primario: Crea una usuaria llamada ana y asígnala directamente al grupo primario desarrolladores .

```
carmen@servidor:~/practicas_linux$ sudo useradd -m -g desarrolladores ana
```

5. Crear Usuario Completo: Crea un usuario david asignándolo al grupo primario analistas y, a la vez, como miembro de los grupos secundarios desarrolladores y becarios .

```
carmen@servidor:~/practicas_linux$ sudo useradd -m -g analistas -G desarrolladores,becarios david
```

7. Verificar Usuarios: Comprueba que los tres nuevos usuarios existen en el sistema, inspeccionando el final del archivo /etc/passwd .

```
carmen@servidor:~/practicas_linux$ sudo tail /etc/passwd
tcpdump:x:105:107::/nonexistent:/usr/sbin/nologin
tss:x:106:108:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:107:109::/var/lib/landscape:/usr/sbin/nologin
fwupd-refresh:x:989:989:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin
usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
carmen:x:1000:1000:carmen:/home/carmen:/bin/bash
sshd:x:109:65534::/run/sshd:/usr/sbin/nologin
juan:x:1001:1004::/home/juan:/bin/sh
ana:x:1002:1001::/home/ana:/bin/sh
david:x:1003:1002::/home/david:/bin/sh
```

8. Cambiar de Usuario: Conviértete en el usuario juan usando el comando su .Una vez dentro de su sesión, comprueba quién eres y en qué directorio te encuentras. Vuelve a tu sesión de usuario original.

```
carmen@servidor:~$ su - juan
Password:
$ whoami
juan
$ pwd
/home/juan
$ exit
```

9. Modificar Grupos de un Usuario: Modifica al usuario juan para que su grupo primario sea becarios y añádelo también al grupo secundario analistas .

```
carmen@servidor:~$ sudo usermod -g becarios juan
[sudo] password for carmen:
carmen@servidor:~$ sudo usermod -aG analistas juan
```

10. Verificar Modificación: Comprueba que los cambios del usuario juan se han aplicado correctamente.

```
carmen@servidor:~$ groups juan
juan : becarios analistas
```

11. Bloquear una Cuenta: Bloquea la cuenta del usuario juan para que no pueda iniciar sesión.

```
carmen@servidor:~$ sudo usermod -L juan
```

12. Intentar Cambiar a Usuario Bloqueado: Intenta convertirte en el usuario juan de nuevo. Debería fallar.

```
carmen@servidor:~$ su - juan
Password:
su: Authentication failure
```

13. Desbloquear una Cuenta: Desbloquea la cuenta del usuario juan .

```
carmen@servidor:~$ sudo usermod -U juan
carmen@servidor:~$ su - juan
Password:
$
```

14. Eliminar un Grupo: Elimina el grupo becarios . ¿Qué ocurre? (Nota: Fallará si algún usuario lo tiene como grupo primario).

```
carmen@servidor:~$ sudo groupdel becarios
groupdel: cannot remove the primary group of user 'juan'
```

15. Eliminar Usuario y su Directorio: Elimina al usuario juan y asegúrate de que su directorio personal (/home/juan) también se borre.

```
carmen@servidor:~$ sudo userdel -r juan
userdel: group juan not removed because it is not the primary group of user juan.
userdel: juan mail spool (/var/mail/juan) not found
carmen@servidor:~$ id juan
id: ‘juan’: no such user
carmen@servidor:~$ ls /home
ana  carmen  david
```

3. Permisos y Propiedad de Archivos

Realiza los siguientes ejercicios dentro de la carpeta practicas_linux de tu directorio home.

1. Crear Archivos de Prueba: Dentro de la carpeta proyectos , crea un archivo vacío llamado informe.txt . Dentro de scripts , crea otro archivo vacío llamado lanzar_app.sh .

```
carmen@servidor:~/practicas_linux$ touch proyectos/informe.txt
carmen@servidor:~/practicas_linux$ touch scripts/lanzar_app.sh
```

2. Ver Permisos: Muestra los permisos por defecto de los archivos y directorios que has creado. Anota quién es el propietario y el grupo.

```
carmen@servidor:~/practicas_linux$ ls -lR
.:
total 12
drwxrwxr-x 2 carmen carmen 4096 oct 29 10:10 documentos
drwxrwxr-x 2 carmen carmen 4096 oct 29 10:10 proyectos
drwxrwxr-x 2 carmen carmen 4096 oct 29 10:10 scripts

./documentos:
total 0

./proyectos:
total 0
-rw-rw-r-- 1 carmen carmen 0 oct 29 10:10 informe.txt

./scripts:
total 0
-rw-rw-r-- 1 carmen carmen 0 oct 29 10:10 lanzar_app.sh
```

Propietario carmen, del grupo carmen

3. Cambiar Propietario: Cambia el propietario del archivo informe.txt para que pertenezca a la usuaria ana .

```
carmen@servidor:~/practicas_linux$ sudo chown ana proyectos/informe.txt
[sudo] password for carmen:
carmen@servidor:~/practicas_linux$ ls -l proyectos/
total 0
-rw-rw-r-- 1 ana carmen 0 oct 29 10:10 informe.txt
```

4. Cambiar Grupo: Cambia el grupo del directorio proyectos para que pertenezca al grupo desarrolladores .

```
carmen@servidor:~/practicas_linux$ sudo chgrp -R desarrolladores proyectos
carmen@servidor:~/practicas_linux$ ls -lR
.:
total 12
drwxrwxr-x 2 carmen carmen          4096 oct 29 10:10 documentos
drwxrwxr-x 2 carmen desarrolladores 4096 oct 29 10:10 proyectos
drwxrwxr-x 2 carmen carmen          4096 oct 29 10:10 scripts

./documentos:
total 0

./proyectos:
total 0
-rw-rw-r-- 1 ana desarrolladores 0 oct 29 10:10 informe.txt

./scripts:
total 0
-rw-rw-r-- 1 david analistas 0 oct 29 10:10 lanzar_app.sh
```

5. Cambiar Propietario y Grupo: Cambia el propietario y el grupo del archivo lanzar_app.sh para que pertenezcan al usuario david y al grupo analistas , respectivamente, con un solo comando.

```
carmen@servidor:~/practicas_linux$ sudo chown david:analistas scripts/lanzar_app.sh
carmen@servidor:~/practicas_linux$ ls -l scripts/
total 0
-rw-rw-r-- 1 david analistas 0 oct 29 10:10 lanzar_app.sh
```

6. Permisos con Notación Octal (Archivo): Usa la notación numérica (octal) para asignar los siguientes permisos a informe.txt : el propietario (ana) puede leer y escribir; el grupo (desarrolladores) solo puede leer; y los otros no tienen ningún permiso.

```
carmen@servidor:~/practicas_linux$ sudo chmod 640 proyectos/informe.txt
carmen@servidor:~/practicas_linux$ ls -l proyectos/
total 0
-rw-r----- 1 ana desarrolladores 0 oct 29 10:10 informe.txt
```

7. Permisos con Notación Octal (Directorio): Asigna permisos de lectura, escritura y ejecución para el propietario y solo de lectura y ejecución para los miembros del grupo al directorio documentos .

```
carmen@servidor:~/practicas_linux$ chmod 750 documentos
carmen@servidor:~/practicas_linux$ ls -l
total 12
drwxr-x--- 2 carmen carmen          4096 oct 29 10:10 documentos
```

8. Verificar Permisos: Lista el contenido de practicas_linux para verificar que todos los cambios de propietario y permisos se han aplicado correctamente.

```
carmen@servidor:~/practicas_linux$ ls -lR
.:
total 12
drwxr-x--- 2 carmen carmen          4096 oct 29 10:10 documentos
drwxrwxr-x 2 carmen desarrolladores 4096 oct 29 10:10 proyectos
drwxrwxr-x 2 carmen carmen          4096 oct 29 10:10 scripts

./documentos:
total 0

./proyectos:
total 0
-rw-r----- 1 ana desarrolladores 0 oct 29 10:10 informe.txt

./scripts:
total 0
-rw-rw-r-- 1 david analistas 0 oct 29 10:10 lanzar_app.sh
```

9. Permisos con Notación Simbólica (Añadir): Usa la notación simbólica para añadir el permiso de ejecución al propietario del script lanzar_app.sh .

```
carmen@servidor:~/practicas_linux$ sudo chmod u+x scripts/lanzar_app.sh
carmen@servidor:~/practicas_linux$ ls -l scripts/
total 0
-rwxrw-r-- 1 david analistas 0 oct 29 10:10 lanzar_app.sh
```

10. Permisos con Notación Simbólica (Quitar): Quita el permiso de lectura al "resto del mundo" (otros) en el directorio proyectos .

```
carmen@servidor:~/practicas_linux$ chmod o-r proyectos
carmen@servidor:~/practicas_linux$ ls -ld proyectos/
drwxrwx--x 2 carmen desarrolladores 4096 oct 29 10:10 proyectos/
```

11. Permisos Recursivos: Dentro de proyectos , crea una nueva carpeta version2 con un archivo notas.txt dentro. Luego, cambia el propietario de la carpeta proyectos y todo su contenido para que pertenezca a david con un solo comando recursivo.

```
carmen@servidor:~/practicas_linux$ mkdir proyectos/version2
```

```
carmen@servidor:~/practicas_linux$ ls -lR proyectos/
proyectos/:
total 4
-rw-r----- 1 ana    desarrolladores    0 oct 29 10:10 informe.txt
drwxrwxr-x 2 carmen carmen          4096 oct 29 10:47 version2

proyectos/version2:
total 0
-rw-rw-r-- 1 carmen carmen 0 oct 29 10:47 notas.txt
```

```
carmen@servidor:~/practicas_linux$ sudo chown -R david proyectos
```

12. Permiso Especial SGID en Directorio: Establece el permiso especial SGID en el directorio documentos . Después, cambia a ser el usuario david (su david) y crea un nuevo archivo dentro de documentos . Verifica a qué grupo pertenece el nuevo archivo (debería heredar el del directorio documentos). Vuelve a tu usuario.

```
$ ls -l proyectos
total 4
-rw-r--r-- 1 david analistas          0 oct 29 10:58 archivo_david.txt
-rw-r----- 1 david desarrolladores    0 oct 29 10:10 informe.txt
drwxrwxr-x 2 david carmen          4096 oct 29 10:47 version2
$ ls -ld proyectos
drwxrwx--x 3 david desarrolladores 4096 oct 29 10:58 proyectos
```

13. Permiso Especial SUID: Establece el permiso SUID en el script lanzar_app.sh . (Nota: Explica a tus alumnos qué implicaría esto si fuera un programa compilado).

```
carmen@servidor:~/practicas_linux$ sudo chmod u+s scripts/lanzar_app.sh
```

Permite que un programa se ejecute con los permisos del propietario del archivo. Ignorado en scripts de shell por razones de seguridad. Solo funciona en binarios compilados.

14. Comprobar umask : Muestra el valor umask actual de tu sesión.

```
carmen@servidor:~/practicas_linux$ umask
0002
```

15. Efecto de umask : Cambia temporalmente tu umask a 077 . Crea un nuevo archivo llamado privado.txt . Comprueba sus permisos por defecto. Luego, restaura el umask a su valor original.

```
carmen@servidor:~/practicas_linux$ umask 077
carmen@servidor:~/practicas_linux$ touch privado.txt
carmen@servidor:~/practicas_linux$ ls -l privado.txt
-rw------- 1 carmen carmen 0 oct 29 11:08 privado.txt
carmen@servidor:~/practicas_linux$ umask 0022
```

4. Gestión de Servicios con systemctl

Nota: Para estos ejercicios, es seguro usar un servicio como cups (impresión) o cron / crond (tareas programadas). Evita usar servicios críticos como sshd si no estás seguro.

1. Estado Detallado de un Servicio: Comprueba el estado completo del servicio cups .

```
carmen@servidor:~$ systemctl status cups
● cups.service - CUPS Scheduler
     Loaded: loaded (/usr/lib/systemd/system/cups.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-10-30 09:06:25 UTC; 59s ago
TriggeredBy: ● cups.socket
             ● cups.path
       Docs: man:cupsd(8)
   Main PID: 8796 (cupsd)
     Status: "Scheduler is running..."
      Tasks: 2 (limit: 2265)
     Memory: 5.1M (peak: 17.4M)
        CPU: 795ms
     CGroup: /system.slice/cups.service
             ├─8796 /usr/sbin/cupsd -l
             └─9035 /usr/lib/cups/notifier/dbus dbus://

oct 30 09:06:25 servidor systemd[1]: Starting cups.service - CUPS Scheduler...
oct 30 09:06:25 servidor systemd[1]: Started cups.service - CUPS Scheduler.
```

2. Comprobación Rápida: Utiliza un comando más directo para verificar si el servicio cups está actualmente en ejecución (activo). La salida de este comando debería ser simplemente active o inactive .

```
carmen@servidor:~$ systemctl is-active cups
active
```

3. Ver Archivo de Unidad: servicio cups (cu

```
carmen@servidor:~$ systemctl cat cups.service
# /usr/lib/systemd/system/cups.service
[Unit]
Description=CUPS Scheduler
Documentation=man:cupsd(8)
After=network.target nss-user-lookup.target nslcd.service
Requires=cups.socket

[Service]
ExecStart=/usr/sbin/cupsd -l
Type=notify
Restart=on-failure


[Install]
Also=cups.socket cups.path
WantedBy=printer.target multi-user.target
```

4. Detener un Servicio: Detén la ejecución del servicio cups . Comprueba su estado de nuevo para confirmar que está inactive (dead) .

```
carmen@servidor:~$ sudo systemctl stop cups
[sudo] password for carmen:
Stopping 'cups.service', but its triggering units are still active:
cups.socket, cups.path
carmen@servidor:~$ systemctl is-active cups
inactive
```

5. Iniciar un Servicio: Vuelve a iniciar el servicio cups . Verifica una vez más que ha vuelto al estado active (running) .

```
carmen@servidor:~$ sudo systemctl start cups
carmen@servidor:~$ systemctl is-active cups
active
```

6. Reiniciar un Servicio: El comando restart es muy común tras un cambio de configuración. Ejecútalo para el servicio cups .

```
carmen@servidor:~$ sudo systemctl restart cups
```

7. Habilitar para el Arranque: Asegúrate de que el servicio cups esté configurado para iniciarse automáticamente cada vez que el sistema arranque.

```
carmen@servidor:~$ sudo systemctl enable cups
Synchronizing state of cups.service with SysV service script with /usr/lib/systemd/systemd-sysv-
l.
Executing: /usr/lib/systemd/systemd-sysv-install enable cups
carmen@servidor:~$ systemctl is-enabled cups
enabled
```

8. Verificar si está Habilitado: Usa un comando específico para preguntar si cups está habilitado. La salida debería ser enabled o disabled .

```
carmen@servidor:~$ systemctl is-enabled cups
enabled
```

9. Deshabilitar para el Arranque: Ahora, desactiva el servicio cups para que no se inicie automáticamente.

```
carmen@servidor:~$ sudo systemctl disable cups
Synchronizing state of cups.service with SysV service script with /usr/lib/systemd/systemd-sysv-instal
l.
Executing: /usr/lib/systemd/systemd-sysv-install disable cups
Removed "/etc/systemd/system/multi-user.target.wants/cups.service".
Removed "/etc/systemd/system/multi-user.target.wants/cups.path".
Removed "/etc/systemd/system/sockets.target.wants/cups.socket".
Removed "/etc/systemd/system/printer.target.wants/cups.service".
Disabling 'cups.service', but its triggering units are still active:
cups.socket
```

10. Enmascarar un Servicio: El enmascaramiento es una forma más contundente de deshabilitar, ya que impide cualquier tipo de inicio (manual o automático). Enmascara el servicio cups . Intenta iniciarlo después. Debería fallar. No olvides desenmascararlo (unmask) al terminar el ejercicio.

```
carmen@servidor:~$ sudo systemctl mask cups
Created symlink /etc/systemd/system/cups.service → /dev/null.
Masking 'cups.service', but its triggering units are still active:
cups.socket
carmen@servidor:~$ sudo systemctl start cups
Failed to start cups.service: Unit cups.service is masked.
carmen@servidor:~$ sudo systemctl unmask cups
Removed "/etc/systemd/system/cups.service".
carmen@servidor:~$ systemctl status cups
● cups.service - CUPS Scheduler
     Loaded: loaded (/usr/lib/systemd/system/cups.service; disabled; preset: enabled)
     Active: active (running) since Thu 2025-10-30 09:44:40 UTC; 7min ago
TriggeredBy: ● cups.socket
       Docs: man:cupsd(8)
   Main PID: 9199 (cupsd)
     Status: "Scheduler is running..."
      Tasks: 1 (limit: 2265)
     Memory: 1.7M (peak: 1.9M)
        CPU: 15ms
     CGroup: /system.slice/cups.service
             └─9199 /usr/sbin/cupsd -l

oct 30 09:44:40 servidor systemd[1]: Starting cups.service - CUPS Scheduler...
oct 30 09:44:40 servidor systemd[1]: Started cups.service - CUPS Scheduler.
carmen@servidor:~$
```

5. Gestión de ufw

1. Comprobar Estado y Activar UFW:
*** Primero, ejecuta un comando para verificar el estado actual del firewall. Probablemente estará inactivo.**
*** A continuación, activa UFW. Presta atención al mensaje de advertencia, especialmente si estás conectado por SSH.**

```
carmen@servidor:~$ sudo ufw status
Status: inactive
carmen@servidor:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)?
```

```
carmen@servidor:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
carmen@servidor:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
```

2. Permitir un Servicio Web (HTTP):
*** Imagina que tu servidor necesita alojar una página web. Añade una regla para permitir todas las conexiones entrantes para el servicio http .**
*** Verifica el estado del firewall de nuevo para confirmar que la regla (y el puerto 80) se ha añadido correctamente.**

```
carmen@servidor:~$ sudo ufw allow http
Rule added
Rule added (v6)
carmen@servidor:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
80/tcp                     ALLOW IN    Anywhere
22/tcp (v6)                ALLOW IN    Anywhere (v6)
80/tcp (v6)                ALLOW IN    Anywhere (v6)
```

3. Abrir un Puerto Específico: 8080

```
carmen@servidor:~$ sudo ufw allow 8080/tcp
Rule added
Rule added (v6)
carmen@servidor:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
8080/tcp                   ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
8080/tcp (v6)              ALLOW       Anywhere (v6)
```

4. Permitir un Rango de Puertos: desde el 3000 al 3100.

```
carmen@servidor:~$ sudo ufw allow 3000:3100/tcp
Rule added
Rule added (v6)
carmen@servidor:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
8080/tcp                   ALLOW       Anywhere
3000:3100/tcp              ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
8080/tcp (v6)              ALLOW       Anywhere (v6)
3000:3100/tcp (v6)         ALLOW       Anywhere (v6)
```

5. Bloquear una Dirección IP:

```
carmen@servidor:~$ sudo ufw deny from 192.168.100.50
Rule added
carmen@servidor:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere
8080/tcp                   ALLOW       Anywhere
3000:3100/tcp              ALLOW       Anywhere
Anywhere                   DENY        192.168.100.50
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)
8080/tcp (v6)              ALLOW       Anywhere (v6)
3000:3100/tcp (v6)         ALLOW       Anywhere (v6)
```

6. Listar Reglas para Borrar:

```
carmen@servidor:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere
[ 3] 8080/tcp                   ALLOW IN    Anywhere
[ 4] 3000:3100/tcp              ALLOW IN    Anywhere
[ 5] Anywhere                   DENY IN     192.168.100.50
[ 6] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 7] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 8] 8080/tcp (v6)              ALLOW IN    Anywhere (v6)
[ 9] 3000:3100/tcp (v6)         ALLOW IN    Anywhere (v6)
```

7. Eliminar una Regla: 8080

```
carmen@servidor:~$ sudo ufw delete 8
Deleting:
 allow 8080/tcp
Proceed with operation (y|n)? y
Rule deleted (v6)
carmen@servidor:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere
[ 3] 8080/tcp                   ALLOW IN    Anywhere
[ 4] 3000:3100/tcp              ALLOW IN    Anywhere
[ 5] Anywhere                   DENY IN     192.168.100.50
[ 6] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 7] 80/tcp (v6)                ALLOW IN    Anywhere (v6)
[ 8] 3000:3100/tcp (v6)         ALLOW IN    Anywhere (v6)
```