Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

Gestiona sistemes operatius utilitzant comandes i eines gràfiques i avaluant les necessitats del sistema

0 — Introducció

Un sistema operatiu és la capa de software que permet que els recursos físics d'un ordinador puguin ser utilitzats per aplicacions i usuaris. En un context domèstic, aquesta funció sovint queda oculta: l'usuari inicia sessió i utilitza programes sense ser conscient del paper del sistema operatiu en la gestió dels recursos. En canvi, en un entorn professional, especialment vinculat al desenvolupament i desplegament d'aplicacions web, el sistema operatiu es converteix en un component crític que cal entendre i administrar.
En un servidor on s'hi allotja una aplicació web, el sistema operatiu no és només una eina d'ús final: és l'entorn d'execució de serveis essencials (nginx, Apache, MariaDB/MySQL, PostgreSQL, Docker, entre d'altres). Una configuració incorrecta d'usuaris, permisos o processos pot provocar errors en l'aplicació, filtracions de dades o interrupcions de servei. La correcta administració del sistema operatiu, per tant, forma part directa de la qualitat i la seguretat del programari desplegat.
Des d'aquesta perspectiva, l'objectiu d'aquest manual és oferir una base sòlida en:

1. la gestió d'usuaris i grups,
2. les polítiques de seguretat dels comptes,
3. la gestió de permisos i ACL,
4. el control de serveis i processos,
5. la utilització de comandes d'administració,
6. i l'anàlisi de registres (logs).

Aquesta formació és necessària abans de treballar amb eines més avançades, com ara contenidors (Docker/Podman), orquestradors (Kubernetes), sistemes CI/CD, o entorns híbrids on conviuen serveis locals i en el núvol.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

Cal remarcar que en entorns de desplegament web, la frontera entre **administració del sistema** i **desenvolupament** cada cop és més fina. Tot i que tradicionalment el rol d'"administrador de sistemes" (sysadmin) estava separat del de desenvolupador, actualment —especialment en pimes i entorns DevOps— moltes tasques estan compartides. Saber administrar el sistema és essencial perquè el desplegament sigui segur, estable i reproductible.

Aquest manual utilitza com a sistema base **Linux Debian/Ubuntu**, atès que és l'entorn més utilitzat en servidors web reals. Les referències a Windows serviran com a comparativa conceptual. La terminologia tècnica (process, service, log, daemon, filesystem, etc.) no serà traduïda quan la versió original sigui l'estàndard en la documentació internacional.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

1 — Configuració d'usuaris i grups

1.1. Identitat dins el sistema

En un sistema operatiu multiusuari, cada accés al sistema està associat a una identitat. Aquesta identitat s'anomena **usuari**, i és la base de qualsevol mecanisme de seguretat. El sistema no gestiona permisos sobre "persones", sinó sobre **usuaris registrats** que després poden correspondre a persones, serveis o aplicacions.
A Linux, cada usuari està identificat internament per un **UID** (User ID). El nom d'usuari és només una etiqueta llegible; el que realment importa és el número UID. De manera similar, els grups tenen un **GID** (Group ID). El sistema utilitza aquests identificadors per determinar l'accés als recursos.

Element	**Funció**
UID	identifica un usuari
GID	identifica un grup
HOME	directori personal
SHELL	interpret d'ordres associat a l'usuari

En un entorn web, alguns serveis tenen el seu propi usuari dedicat (p. ex. `www-data` a Debian/Ubuntu), de manera que si hi ha un problema de seguretat, l'impacte queda limitat als permisos d'aquest usuari.

1.2. Tipus d'usuaris

A Linux podem diferenciar:

- **root**: usuari administrador absolut (UID 0).
- **usuaris humans**: creats per a persones reals.
- **usuaris de sistema o de servei**: utilitzats per processos i daemons.

A Windows trobem un paral·lelisme amb:

- **Administrador**
- **Usuaris estàndard**
- **Usuaris del sistema i serveis locals**

L'existència d'usuaris de servei és especialment rellevant en entorns web, ja que cada servidor (nginx, Apache, MySQL…) s'executa amb una identitat pròpia per limitar danys en cas de vulnerabilitat.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

1.3. Rol dels grups

Els grups permeten gestionar permisos de manera col·lectiva. En lloc d'assignar permisos usuari per usuari, es defineix un grup (per exemple, `developers`) i s'hi afegeixen tots els desenvolupadors. Això facilita:

- control d'accés,
- organització del filesystem,
- i responsabilitat compartida sobre fitxers i directors.

En desplegaments web és habitual crear grups per separar rols (ex.: `devs`, `ops`, `www-admin`, `db-admin`), especialment quan diverses persones tenen accés a la mateixa màquina.

1.4. Exemple pràctic vinculat a entorn web

En un servidor web Linux, els fitxers del lloc estan habitualment ubicats a `/var/www/nom-del-projecte/`. Aquests fitxers solen pertànyer a l'usuari `www-data` (el mateix que utilitza nginx o Apache). Els desenvolupadors no haurien de tenir permisos root, però sí accés d'escriptura al directori web, sovint mitjançant un **grup compartit**.
Aquest model evita que un error humà o un exploit comprometi tot el sistema.

1.5. Conseqüències d'una configuració incorrecta

Una mala gestió d'usuaris i grups pot comportar:
exposició de dades sensibles,
modificació accidental o maliciosa de fitxers web,
processos que s'executen amb permisos massa elevats,
dificultat per auditar qui ha fet què.
Aquest és el punt de partida per a la resta dels capítols: els permisos, processos i logs només tenen sentit si hi ha una base correcta d'identitats.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

2 — Seguretat dels comptes d'usuari

La seguretat d'un sistema operatiu comença en el control de qui pot accedir-hi i amb quins privilegis. Un compte d'usuari no és només un nom: és una possible porta d'entrada al sistema. Per aquest motiu, la configuració correcta dels comptes és essencial per prevenir accessos no autoritzats, especulatius o fraudulents.

2.1. Riscos habituals associats als comptes

Els comptes d'usuari poden convertir-se en un vector d'atac si no es gestionen adequadament. Els riscos més comuns són:

- **Usuaris innecessaris o temporals no eliminats**
  Sovint després d'una intervenció tècnica o una pràctica, queden comptes actius que ja no calen.
- **Ús compartit d'un mateix compte**
  Impedeix identificar accions individuals i fa impossible auditar.
- **Privilegis excessius**
  Un usuari que només necessita llegir fitxers web no hauria de tenir permisos d'escriptura ni molt menys accés root.
- **Absència de polítiques de bloqueig**
  Un atac per força bruta pot provar milers de contrasenyes sense que el sistema actuï.
- **Autenticacions automatitzades amb claus sense control**
  Per exemple, *claus SSH abandonades* a `~/.ssh/authorized_keys`.

En entorns de desplegament web, aquests riscos sovint passa desapercebuts perquè el focus es posa en el codi i el servidor web, però un sistema compromès per un compte mal gestionat és igual (o més) greu que una vulnerabilitat en l'aplicació.

2.2. Polítiques d'assignació i control d'usuaris

Per garantir la seguretat, un sistema ben administrat segueix un principi fonamental:
Principi de privilegi mínim
Cada usuari ha de tenir només els permisos estrictament necessaris per dur a terme la seva tasca.
A partir d'aquest principi es deriven pràctiques habituals:

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

Mesura	**Finalitat**
Creació d'usuaris amb rols acotats	Evita privilegis innecessaris
Separació d'usuaris humans i de servei	Millora traçabilitat
Gestió per grups, no per individu	Facilita administració
Revocació d'usuaris inactius	Redueix superfície d'atac
Registre i auditoria	Permet investigar incidències

En entorns d'equip —com ho seria un laboratori web o un servidor compartit per alumnes— és especialment útil la segmentació per grups: un grup per desenvolupadors, un altre per administradors, etc.

2.3. Polítiques de bloqueig

Les polítiques de bloqueig determinen què passa quan un usuari introdueix una contrasenya incorrecta repetidament. Sense aquestes mesures, un atacant pot provar combinacions de contrasenyes indefinidament. Debian/Ubuntu permet configurar aquest comportament mitjançant el mòdul PAM (`/etc/pam.d/common-auth`) o utilitats com `faillog` o `pam_faillock` en sistemes més moderns.
Exemples de política habitual:

- Bloqueig temporal després de diversos intents fallits
- Increment progressiu del temps d'espera
- Notificació a l'administrador
- Requeriment de canvi de contrasenya després del desbloqueig

A Windows, l'equivalent es troba a: `secpol.msc → Polítiques de comptes → Política de bloqueig`.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

2.4. Els comptes de servei en entorns web

Els serveis com nginx, Apache, MySQL, redis, etc. s'executen amb comptes dedicats precisament per reduir l'impacte en cas de vulnerabilitat. Un error habitual en entorns poc professionals és donar permisos root a aquests serveis o fer desplegaments amb comptes humans, la qual cosa redueix la seguretat de forma dràstica.
Exemples:

Servei	**Usuari habitual**	**Motiu**
nginx	`www-data`	Només necessita accedir a fitxers web
mysql	`mysql`	Aïlla la base de dades del sistema
docker	`root` + grup `docker`	Controla containers i xarxa
gitlab-runner	`gitlab-runner`	Execució aïllada d'scripts

Un bon disseny de comptes impedeix que una vulnerabilitat en un servei comprometi la resta del sistema.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

3 — Seguretat de contrasenyes

La contrasenya és el mecanisme d'autenticació més utilitzat en sistemes informàtics. Tanmateix, és també el més vulnerable quan no s'aplica una bona política de seguretat. La gestió correcta de contrasenyes implica entendre tant *com* es creen i s'emmagatzemen, com els mecanismes que es poden utilitzar per reforçar-les o substituir-les.

3.1. Contrasenyes i autenticació

Quan un usuari inicia sessió en un sistema Linux, el que realment comprova el sistema no és la contrasenya en text pla, sinó un **hash** emmagatzemat en el fitxer `/etc/shadow`.
Un hash és una transformació criptogràfica que no permet obtenir la contrasenya original. Això significa que el sistema no "sap" quina és la contrasenya original, només pot comparar el resultat d'aplicar la mateixa funció hash a la que introdueix l'usuari.
Aquesta separació impedeix que qui accedeixi al sistema pugui llegir directament les contrasenyes dels usuaris.

3.2. Hashing, salts i dificultat computacional

Per evitar atacs per diccionari o taules precomputades (rainbow tables), els sistemes afegeixen un element aleatori anomenat **salt**, que es concatena amb la contrasenya abans de fer el hash. Això assegura que dues contrasenyes idèntiques no donin com a resultat el mateix hash.
Els hash moderns utilitzen algoritmes dissenyats perquè siguin deliberadament lents o costosos computacionalment, com ara:

Algorisme	**Característica**
SHA-512 + salt	estàndard en Debian/Ubuntu
bcrypt	ús ampli en serveis web
scrypt	alt cost computacional i memòria
Argon2	recomanat en sistemes moderns

Com més lent és el procés de hashing, més difícil és atacar el sistema per força bruta.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

3.3. Complexitat vs robustesa real

Una contrasenya segura no depèn únicament de la complexitat formal (majúscules, números, símbols), sinó principalment de la seva **llargada** i de la resistència a ser endevinada mitjançant eines automatitzades.
Exemples comparatius:

Contrasenya	Tipus	Robustesa real
`F3r!c0`	breu i "complexa"	feble
`barret-verd-3-gats`	llarga i memoritzable	molt forta
`P@ssw0rd!`	típica substitució	altament vulnerable
`muntanyesdiesalvinyaplenes`	frase llarga	difícil d'atacar

Allò que realment frena un atacant és el **temps de càlcul necessari** per provar combinacions, no els símbols en si mateixos.

3.4. Rotació i vida de contrasenya

En entorns corporatius sovint s'imposa la rotació obligatòria de contrasenyes cada cert temps. En servidors Linux, això es pot definir a `/etc/login.defs` o mitjançant `chage`.
És important destacar que la rotació no és una solució universal: massa rotació pot provocar que els usuaris acabin creant contrasenyes febles o previsibles.
Un enfocament modern prioritza: contrasenyes **robustes i llarges**, control d'accessos per factors, i revocació d'accés immediata en comptes compromesos.

3.5. Autenticació amb claus SSH

En entorns web, especialment quan hi ha desplegaments remots, l'ús de claus SSH és preferible a l'ús de contrasenyes.
Funcionament bàsic:
El client disposa d'un **parell de claus**: una clau pública i una clau privada.
La clau **pública** s'instal·la al servidor (`~/.ssh/authorized_keys`).
La clau **privada** queda només al client i mai viatja per la xarxa.
L'autenticació es realitza mitjançant criptografia asimètrica.
Això aporta diverses millores:

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

Avantatge	Explicació
No hi ha contrasenya en trànsit	Redueix risc d'intercepció
Resistència a força bruta	Les claus tenen entropia molt superior
Pràctic per DevOps	Permet automatitzar desplegaments
Auditoria més clara	Una clau per usuari/servei

En la pràctica DevOps moderna, l'accés per SSH amb clau és la base tant del desplegament continu com del manteniment remot securitzat.

3.6. Autenticació multifactor

En entorns especialment crítics (per exemple, servidors amb bases de dades sensibles) és habitual complementar SSH amb 2FA, mitjançant aplicacions com Google Authenticator o tokens físics (YubiKey). Això redueix el risc d'accés encara que la clau privada hagi estat robada.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

4 — Accés a recursos: permisos locals i ACL

L'accés als recursos és un dels pilars fonamentals de la seguretat del sistema operatiu. Tant en entorns d'usuari com, sobretot, en servidors web, és imprescindible controlar *qui* pot llegir, modificar o executar un recurs determinat.
En Linux, aquest control s'expressa principalment en forma de **permisos** associats a usuaris i grups. Quan aquest model bàsic no és suficient, es fan servir les **ACL** (Access Control Lists), que permeten un control més granular.

4.1. Concepte de recurs i propietat

Per "recurs" entenem qualsevol element gestionat pel sistema:
fitxers,
directoris,
dispositius (ex.: `/dev/sda`),
sockets o canals de comunicació,
processos en execució.
Cada recurs té sempre:

Element	**Descripció**
Usuari propietari (owner)	UID associat
Grup propietari	GID associat
Permisos	lectura, escriptura, execució

Aquest model fa possible restringir l'accés d'un recurs concret únicament a qui realment el necessita.

4.2. Tipus de permisos: R, W, X

Els permisos bàsics són tres:

Permís	**Significat en fitxers**	**Significat en directoris**
`r` (read)	llegir contingut	llistar contingut (`ls`)
`w` (write)	modificar o eliminar	crear/eliminar entrades
`x` (execute)	executar com a programa/script	accedir-hi i recórrer-lo

A efectes pràctics, un directori sense permís `x` no és "transevitable", encara que es pugui llistar.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

4.3. Scope dels permisos: usuari, grup i altres

Els permisos s'apliquen en tres nivells:

Àmbit	**Descripció**
owner (u)	propietari directe
group (g)	membres del grup assignat
other (o)	tots els altres usuaris

Exemple:
`-rw-r--r--` indica:
propietari → lectura + escriptura
grup → lectura
altres → lectura
Aquest esquema és simple, però pot resultar insuficient en entorns compartits i complexos (com un servidor web amb múltiples equips treballant-hi).

4.4. Permisos i desplegament web

En la majoria de servidors web Debian/Ubuntu:
El procés web (nginx/Apache) **no és un usuari humà**, sinó `www-data`.
Aquest usuari necessita **llegir** (i a vegades escriure) contingut a `/var/www/...`.
Els desenvolupadors **no haurien** d'executar el servei com ells mateixos.
L'administrador configura l'accés mitjançant grups.
Model típic:
owner: www-data
group: developers

Això permet que:
el servei web llegeixi i serveixi fitxers,
els desenvolupadors modifiquin contingut,
cap altre usuari local hi accedeixi.

4.5. ACL (Access Control Lists)

Quan el model bàsic es queda curt, entren en joc les ACL, que permeten especificar permisos no només per a "owner", "group" i "other", sinó per a **usuaris i grups concrets addicionals**.

Per exemple, si un projecte requereix que només un desenvolupador concret tingui permís d'escriptura, però que la resta del grup només tingui lectura, amb els permisos Unix tradicionals no és possible expressar-ho. Amb ACL sí.
Sintaxi bàsica:
setfacl -m u:nomusuari:rwx fitxer
setfacl -m g:nomgrup:rx directori
getfacl fitxer

Les ACL són essencials en servidors multiusuari i entorns DevOps on diverses eines automàtiques (CI/CD, git hooks, processos batch) necessiten permisos diferenciats.

4.6. Sticky bit, setuid i setgid

Hi ha permisos especials que afecten el comportament dels fitxers i directoris:

Permís especial	**Efecte**
setuid (s)	executa amb UID del propietari
setgid (s)	executa amb GID del grup
sticky bit (t)	només el propietari pot eliminar en directoris compartits

Exemple típic:
El directori `/tmp` és compartit per tots els usuaris, però té *sticky bit*, per evitar que un usuari elimini fitxers d'un altre.
En entorns web, el *setgid* en directoris és útil per forçar que tots els fitxers nous heretin el **grup** propietari correcte (per exemple `developers`), evitant problemes col·laboratius.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

4.7. Errors habituals en desplegaments web

Error	**Conseqüència**
Donar permisos 777 "perquè funcioni"	Accés indiscriminat i risc crític
Fent deploy amb root	Qualsevol vulnerabilitat = compromís total
Barrejar usuaris humans amb usuaris de servei	Impedeix auditories
No usar grups	Administració impossible a llarg termini
No entendre ACL	Permisos mal expressats → errors aleatoris

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

5 — Serveis i processos

Els serveis i processos constitueixen el nucli d'execució d'un sistema operatiu. En un entorn de desplegament web, pràcticament tot el que fa el servidor —atendre peticions HTTP, gestionar sessions, servir contingut o interactuar amb bases de dades— depèn de processos en execució contínua. Entendre com funcionen i com s'administra la seva vida (inici, aturada, reinici, monitoratge) és essencial per garantir estabilitat i disponibilitat.

5.1. Procés vs. servei

Un **procés** és una instància en execució d'un programa. Té un PID (Process ID), propietari i recursos associats (CPU, memòria, I/O, etc.).
Un **servei** és un procés (o conjunt de processos) gestionat pel sistema, generalment en segon pla (background), dissenyat per executar-se contínuament.

Element	**Procés**	**Servei**
Execució	manual o automàtica	automàtica segons el sistema
Duració	temporal	persistent
Control	directe	gestionat (systemd)
Exemple	`python script.py`	`nginx`, `mysql`, `ssh`

En un entorn web, un procés efímer pot ser una migració de base de dades; en canvi, nginx és un servei, perquè ha de restar actiu permanentment.

5.2. systemd i gestió moderna de serveis

Debian i Ubuntu utilitzen **systemd** com a init system i gestor de serveis. systemd controla:
quins serveis s'inicien a l'arrencada,
l'ordre d'inicialització,
el reinici automàtic en cas de fallada,
i la relació entre serveis (dependencies).
Exemples de comandes fonamentals:

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

Acció	**Comanda**
Comprovar estat	`systemctl status nginx`
Aturar	`systemctl stop nginx`
Iniciar	`systemctl start nginx`
Reiniciar	`systemctl restart nginx`
Habilitar a l'arrencada	`systemctl enable nginx`
Deshabilitar	`systemctl disable nginx`

Aquestes comandes són bàsiques en el desplegament d'aplicacions, perquè qualsevol canvi en la configuració web requereix sovint un reinici controlat.

5.3. Dependències i arrencada

Un servei pot dependre d'un altre. Per exemple:
nginx pot necessitar que la xarxa estigui disponible abans d'iniciar-se.
un servidor d'aplicacions pot requerir que la base de dades estigui operativa.
un servei de logging pot començar abans que la resta, per captar incidents ja a l'inici.
systemd utilitza unit files (fitxers `.service`) ubicats habitualment a `/lib/systemd/system` o `/etc/systemd/system`.

5.4. Execució i privilegis dels serveis

Un servei mai no s'hauria d'executar com a root tret d'excepcions molt concretes. systemd permet definir quin usuari i grup executa cada servei:
[Service]
User=www-data
Group=www-data

Aquest punt connecta directament amb els capítols previs: els permisos correctes són irrelevants si el servei que els utilitza s'executa amb privilegis incorrectes.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

5.5. Control i diagnosi de processos

Per veure processos en temps real:

Eina	Ús
`ps aux`	Instantània dels processos
`top`	Monitoratge en temps real
`htop`	Versió millorada (si instal·lada)
`pgrep`	Trobar PID d'un procés
`kill` / `killall`	Aturar processos

En un servidor web, aquests comandaments permeten detectar consums anormals (fugues de memòria, loops, etc.).

5.6. Supervisió automàtica

systemd pot reiniciar serveis automàticament si detecta que han fallat:
[Service]
Restart=always
RestartSec=3

Això és molt important en entorns DevOps, on les aplicacions poden tenir dependències dinàmiques i és essencial minimitzar temps d'inactivitat.
Quan cal un nivell superior de supervisió (per exemple, múltiples instàncies distribuïdes), s'utilitzen eines com Docker/Kubernetes, però sempre sobre la base d'un sistema amb serveis ben configurats.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

6 — Comandes de sistemes operatius lliures i propietaris

En administració de sistemes, les comandes no són un fi en si mateixes, sinó eines per aconseguir una tasca concreta. Per aquest motiu, la manera més útil d'aprendre-les és agrupar-les en funció d'allò que permeten fer.
Aquest capítol presenta les comandes essencials per treballar amb usuaris, permisos, serveis i logs, i ofereix una comparativa amb l'equivalent en Windows per entendre els conceptes des d'una perspectiva transversal.

6.1. Gestió d'usuaris i grups

Objectiu	Linux	Windows
Crear usuari	`useradd` / `adduser`	`net user nom /add`
Eliminar usuari	`userdel`	`net user nom /delete`
Canviar contrasenya	`passwd`	`net user nom *`
Crear grup	`groupadd`	`net localgroup nom /add`
Afegir usuari a grup	`usermod -aG`	`net localgroup nom usuari /add`

6.2. Permisos i ACL

Objectiu	Linux	Windows
Canviar propietari	`chown`	`icacls /setowner`
Canviar permisos bàsics	`chmod`	`icacls`
Definir ACL granulars	`setfacl` / `getfacl`	`icacls /grant`
Veure permisos	`ls -l`	`icacls fitxer`

6.3. Serveis i processos

Objectiu	Linux (systemd)	Windows
Veure estat servei	`systemctl status`	`sc query`
Iniciar	`systemctl`	`net start`

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

Objectiu	Linux (systemd)	Windows
	`start`	
Aturar	`systemctl stop`	`net stop`
Reiniciar	`systemctl restart`	`sc stop + start`
Llistar processos	`ps aux`	`tasklist`
Aturar procés	`kill`	`taskkill`

6.4. Monitoratge i informació del sistema

Objectiu	Linux	Windows
Ús de CPU/memòria	`top` / `htop`	`taskmgr`
Espai en disc	`df -h`	`wmic logicaldisk` / Explorador
Ús de directori	`du -sh`	PowerShell `Get-ChildItem`
Versionat kernel/OS	`uname -a` / `lsb_release -a`	`winver` / `systeminfo`

6.5. Gestió de xarxa

Encara que es tractarà amb més profunditat en altres mòduls, cal entendre la base:

Objectiu	Linux	Windows
IP / interfícies	`ip a`	`ipconfig`
Ports oberts	`ss -tulpn`	`netstat -ano`
DNS i connectivitat	`dig`, `nslookup`, `ping`	`nslookup`, `ping`

En desplegaments web, aquestes comandes són imprescindibles per:
comprovar si el servidor web està escoltant
validar resolució DNS en producció
depurar problemes de tallafocs o bloqueig de ports

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

6.6. Eines de suport a DevOps

Tot i que aquest manual se centra en el sistema base, és útil destacar un conjunt inicial d'eines que connecten l'administració del SO amb workflows de desplegament moderns:

Funció	Eina	Notes
Automatització remota	`ssh` / `scp`	base per CI/CD manual
Gestió de paquets	`apt`	instal·lació de serveis
Supervisió de logs	`journalctl`	essencial en diagnosi
Execució controlada	`sudo`	substitut segur de root

Aquestes eines són la preparació natural abans d'introduir Docker, pipelines CI/CD, o eines com Ansible.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

7 — Eines de monitoratge del sistema. Registres i logs

El control i la vigilància del sistema operatiu són essencials per mantenir la disponibilitat dels serveis i detectar anomalies abans que afectin els usuaris. En un servidor web, el monitoratge adequat permet anticipar problemes de rendiment, saturació o seguretat.

7.1. El valor del monitoratge

Monitoritzar no és només observar: és **recollir, analitzar i reaccionar** davant la informació que el sistema genera constantment. Els objectius principals són:
detectar errors i inestabilitats,
optimitzar recursos,
controlar l'activitat d'usuaris i serveis,
i registrar evidències per a auditories o investigacions posteriors.
El monitoratge ha de combinar **mesures reactives** (logs, alertes) amb **mesures proactives** (gràfiques d'ús, tendències).

7.2. Registres del sistema (logs)

Un log és un fitxer que registra esdeveniments o missatges generats pel sistema o per aplicacions específiques. A Debian/Ubuntu, la majoria dels logs es troben a `/var/log/`.
Exemples habituals:

Log	**Contingut**	**Exemple d'ús**
`/var/log/syslog`	missatges generals del sistema	diagnosi de fallades o arrencada
`/var/log/auth.log`	accessos i autenticacions	detecció d'intents d'intrusió
`/var/log/nginx/error.log`	errors de servidor web	depuració d'aplicacions web
`/var/log/mysql/error.log`	incidències de base de dades	resolució de problemes d'arrencada
`/var/log/journal/`	base binària de systemd	anàlisi amb `journalctl`

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

7.3. journalctl: la nova gestió de logs a systemd

Amb l'arribada de systemd, el sistema de logs es va centralitzar. Ara, la major part de la informació de serveis es pot consultar directament amb `journalctl`.

Comandes bàsiques:

Objectiu	Comanda
Veure tot el registre	`journalctl`
Filtrar per servei	`journalctl -u nginx`
Veure els últims missatges	`journalctl -xe`
Consultar des d'una data	`journalctl --since "2025-10-20"`
Mostrar logs en temps real	`journalctl -f`

Aquesta eina és fonamental per entendre què passa amb un servei després d'un reinici o desplegament. És també la primera eina de diagnosi en entorns DevOps.

7.4. Eines de monitoratge en temps real

A més dels logs, cal supervisar el comportament dinàmic del sistema. Algunes eines essencials:

Eina	Funció
`top` / `htop`	ús de CPU, memòria i processos actius
`vmstat`, `iostat`	rendiment de CPU i disc
`free -h`	ús de memòria RAM
`df -h`, `du -sh`	espai de disc disponible
`uptime`, `who`, `last`	activitat i sessions d'usuari

Aquestes eines permeten avaluar l'estat global del servidor sense instal·lar software addicional.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

7.5. Monitoratge orientat a serveis web

En un servidor web, els indicadors més crítics són:
temps de resposta del servei (latència HTTP),
ús de CPU i memòria de processos `nginx`, `php-fpm`, `mysql`,
estat del disc i particions (logs o fitxers temporals plens poden bloquejar el sistema),
nombre de connexions actives i ports oberts (`ss -tulpn`).
En entorns de desenvolupament moderns, aquesta informació s'integra en eines com **Grafana**, **Prometheus**, o **Zabbix**, però la base de totes elles continua essent la mateixa: dades extretes del sistema operatiu.

7.6. Gestió i rotació de logs

Els logs poden créixer indefinidament. Per evitar saturar el disc, Debian utilitza el servei `logrotate`, que:
comprimeix logs antics,
elimina registres massa antics,
i manté una mida controlada.
La configuració es troba a `/etc/logrotate.conf` i `/etc/logrotate.d/`.
Aquest punt és especialment important en servidors web, on `access.log` i `error.log` poden créixer ràpidament amb gran volum de trànsit.

7.7. Registres en entorns DevOps

En desplegaments amb contenidors (Docker, Kubernetes), els logs poden viure fora del sistema host. Tot i així, el principi és el mateix: cada servei ha de deixar rastre dels seus esdeveniments per poder ser auditat. El control centralitzat de logs —anomenat *log aggregation*— és una pràctica DevOps bàsica, utilitzant eines com **ELK Stack (Elasticsearch + Logstash + Kibana)** o **Loki**.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

8 — Casos pràctics aplicats a entorns web

Aquest capítol mostra com els conceptes previs s'apliquen en un servidor Linux Debian/Ubuntu utilitzat per desplegar aplicacions web. No es tracta de "com fer una instal·lació web completa", sinó de veure **com intervenen usuaris, permisos, serveis i logs en un escenari real**.

8.1. Preparació d'un entorn per a un servidor web

Com a regla general, abans de desplegar cap aplicació web, el sistema ha d'estar preparat tant a nivell d'usuaris com de permisos.
Model de configuració recomanat:

Component	Configuració correcta	Objectiu
Usuari del servei	`www-data`	Execució segura del webserver
Grup de desenvolupadors	`developers`	Accés d'escriptura controlat
Directori web	`/var/www/projecte/`	Ubicació del codi
Propietari	`www-data`	Coherència amb l'execució
Grup	`developers`	Modificació pel grup
Permisos	`r-x` per www-data, `rwx` per devs	Aïllament i control

Això garanteix que el servidor web pot llegir i servir el contingut, però només els desenvolupadors autoritzats poden modificar-lo.

8.2. Creació del directori del projecte

sudo mkdir -p /var/www/projecte
sudo chown www-data:developers /var/www/projecte
sudo chmod 750 /var/www/projecte

Justificació:
www-data (servei web) → lectura/execució
developers (equip tècnic) → lectura/escriptura/execució
"altres" → sense cap accés
Aquestes ordres reflecteixen el principi de privilegi mínim.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

8.3. Configuració amb ACL per a escenaris avançats

Si només un subconjunt dels `developers` ha de tenir permisos especials:
sudo setfacl -m u:joan:rwx /var/www/projecte
sudo setfacl -m g:developers:rx /var/www/projecte

Aquesta configuració no seria possible només amb `chmod`, que només permet un únic grup.

8.4. Gestió del servei web (nginx a mode d'exemple)

Comprovació d'estat i diagnosi bàsica:
systemctl status nginx
journalctl -u nginx -f

Si després d'un canvi de configuració nginx no arrenca, `journalctl` proporciona l'error exactament, sovint amb l'arxiu i línia específica.

8.5. Un exemple típic d'error de permisos

Símptoma:
La web retorna "403 Forbidden" o "500 Internal Server Error".
Causes habituals:
Directori propietari incorrecte.
Permís `x` absent en el directori.
El codi llegeix o escriu fitxers temporals sense permisos.
Diagnosi:
ls -ld /var/www/projecte
getfacl /var/www/projecte

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

8.6. Logs indispensables en un entorn web

Fitxer	Funció
`/var/log/nginx/error.log`	Errors d'aplicació o configuració
`/var/log/nginx/access.log`	Peticions rebudes (control, auditoria)
`/var/log/php*-fpm.log`	Errors d'execució PHP (si s'utilitza)
`/var/log/mysql/error.log`	Problemes de base de dades

L'ús de `tail -f` o `journalctl -f` és habitual en depuració en temps real.

8.7. Execució d'scripts i processos lligats al servidor

Moltes aplicacions web necessiten treballs periòdics (cron jobs). Igual com amb els serveis, aquests processos també han d'executar-se amb l'usuari adequat.
sudo -u www-data crontab -e

Això evita que els cron jobs corrin com root i comprometin el sistema en cas d'error o vulnerabilitat.

8.8. Punt d'enllaç amb DevOps

Amb aquesta base, es pot fer un pas més cap al desplegament automàtic:

Tecnologia	Paper
SSH keys	accés segur als servidors
Git hooks	desplegament automatitzat al commit
systemd	arrencada i supervisió
logs + monitoratge	diagnosi i control continu
principis de permisos	seguretat i traçabilitat

DevOps no comença amb contenidors, sinó amb **entendre el sistema**. Sense aquesta base, qualsevol pipeline és inestable o insegur.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

9 — Connexió amb DevOps: base operativa per al desplegament

DevOps no és només una tecnologia, sinó una manera de treballar que combina desenvolupament i operacions. Per això, abans d'utilitzar eines avançades com Docker, Kubernetes o pipelines de CI/CD, és imprescindible entendre la base: **què està passant realment al sistema operatiu**.
Aquest capítol mostra com allò que s'ha vist fins ara és, de fet, DevOps fonamental.

9.1. Què és realment DevOps

En un nivell tècnic i funcional, DevOps significa:

- que els desenvolupadors entenen com s'executa el seu programari,
- que el desplegament és fiable i repetible,
- que l'operativa no depèn d'una persona concreta,
- i que els incidents poden ser diagnosticats amb dades reals (logs + monitoratge).

Abans d'eines, DevOps és **procediment i coneixement**. Les eines només automatitzen aquesta base.

9.2. Pont entre desenvolupament i sistema operatiu

Cada part del sistema operatiu estudiada en els capítols anteriors té una traducció directa en tasques DevOps:

Administració	**Rol DevOps**
usuaris i grups	control de rols en entorns de deploy
permisos i ACL	seguretat d'actius i fitxers d'aplicació
serveis i processos	gestió d'entorns d'execució
logs	diagnosi i observabilitat
monitoratge	prevenció i resposta a incidències

Sense dominar aquests punts, qualsevol pipeline CI/CD o contenidor acaba sent opac o fràgil.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

9.3. Automatització: el primer pas real

El primer pas de DevOps no és Docker, sinó **automatitzar tasques repetitives** que, com a administradors o desenvolupadors, no volem executar manualment sempre.
Exemples bàsics:

Tasca	**Exemple d'automatització**
copiar fitxers al servidor	`rsync` o `scp`
reiniciar servei després de deploy	`systemctl restart` en script
comprovar estat abans/després	combinació `curl` + `journalctl`
ajustar permisos post-deploy	`chown` / `setfacl` dins script

Això és ja *pre-CI/CD*, i forma l'arrel dels pipelines.

9.4. Infraestructura com a codi (introducció conceptual)

Quan aquestes operacions bàsiques s'automatitzen i es documenten en fitxers o scripts, estem fent **Infrastructure as Code** a petita escala: el sistema es pot recrear sense intervenció manual directa.
Per exemple:

deploy.sh
- *crea usuari/grup si no existeix*
- *copia fitxers*
- *aplica permisos*
- *reinicia servei*
- *comprova status*

Aquest esquema, si després es trasllada a eines com Ansible o pipelines GitLab/GitHub Actions, es converteix en DevOps complet.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

9.5. La importància de la traçabilitat

La traçabilitat és essencial en entorns on hi ha diversos actors (equips de desenvolupament, clients, administradors):

Element	**Paper**
Permisos correctes	defineixen qui pot fer què
Logs	documenten què ha passat
Monitoratge	detecta anomalies
Auditories	relacionen acció-usuari-efecte

Un entorn sense traçabilitat és insegur, encara que aparentment "funcioni".

9.6. Punt de maduresa cap a DevOps complet

Amb la base anterior, l'evolució natural d'un equip és:

1. Administració sistemàtica i segura del servidor
2. Automatització de tasques repetitives
3. Versionat de l'estat del sistema (infraestructura com a codi)
4. Contenització d'aplicacions (Docker / Podman)
5. Orquestració i alta disponibilitat (Kubernetes)
6. Monitoratge avançat i alertes
7. CI/CD completament integrat

Aquest manual es troba en el punt 1 → 2, que és necessari abans d'implementar 3 → 7.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

10 — Conclusions generals

L'avaluació i configuració d'un sistema operatiu no és un procés aïllat, sinó el fonament sobre el qual es construeixen els entorns d'execució d'aplicacions. Tot desplegament web —tant simple com complex— depèn de:

- **identitats ben definides** (usuaris i grups),
- **accés correctament controlat** (permisos i ACL),
- **serveis gestionats de forma segura** (systemd),
- **diagnosi i estabilitat** (logs i monitoratge),
- **traçabilitat i responsabilitat** (auditoria implícita),
- **procediments repetibles** (base DevOps).

La pràctica real demostra que els problemes en servidors web no solen venir del codi de l'aplicació, sinó d'una configuració de sistema incorrecta o incompleta. Per això, un bon desenvolupador o administrador ha de entendre els dos costats: el programari que s'executa i el sistema que el suporta.

Aquesta base és el pas imprescindible per evolucionar cap al desplegament automatitzat, contenidors, pipelines i escenaris d'alta disponibilitat.

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

11 — Annexos

A. Ubicacions habituals a Debian/Ubuntu

Element	Ruta
Usuaris i grups	`/etc/passwd`, `/etc/group`
Hash de contrasenyes	`/etc/shadow`
Unit files (systemd)	`/etc/systemd/system`
Logs del sistema	`/var/log/`
Config Nginx	`/etc/nginx/`
Arrel web	`/var/www/`

B. Bones pràctiques resumides

- privilegi mínim
- separació clara usuari ↔ servei
- documentació d'accessos
- supervisió contínua

Generalitat de Catalunya
Departament d'Educació
Institut Esteve Terradas i Illa
Bonavista 70
08940 Cornellà de Llobregat
Tel. 93 377 11 00
a8016781@xtec.cat
http://www.esteveterradas.cat

12 — Bibliografia i fonts

Manuals clàssics i bibliografia recomanada

Referència	Observació
Nemeth, Evi et al. *UNIX and Linux System Administration Handbook*	Considerat el text clàssic internacional
Machtelt Garrels, *Introduction to Linux*	Base tècnica per perfils en formació
Richard Blum, *Linux Command Line and Shell Scripting Bible*	Focus en administració i automatització
Michael Kerrisk, *The Linux Programming Interface*	Referència profunda sobre sistemes GNU/Linux
Valerie Quercia, *Linux System Administration*	Guia pràctica i operativa

Documentació oficial

Debian Admin Handbook — https://www.debian.org/doc/
Ubuntu Server Guide — https://ubuntu.com/server/docs
systemd — https://www.freedesktop.org/wiki/Software/systemd/
Nginx — https://nginx.org/en/docs/
Apache — https://httpd.apache.org/docs/
MariaDB/MySQL — https://mariadb.com/kb/en/
OWASP DevOps controls — https://owasp.org/

Fonts complementàries per DevOps

HashiCorp — https://www.hashicorp.com/resources
CNCF (Cloud Native Computing Foundation)
GitLab Documentation — CI/CD i runners
GitHub Actions docs