

# Tema 1: Introducció

---

## Tasques de l'administrador de sistemes

---

- Comprovar la seguretat del sistema
- Realitzar còpies de seguretat
- Solucionar problemes
- Ajudar als usuaris en el seu treball diari
  - Respecte el que fa l'administrador de sistemes
- Mantenir documentació local
- Instal·lar nou/mantenir software
- Altes/baixes d'usuaris
- Afegir/canviar hardware
- Monitoritzar el sistema/comprovar que funciona

No seria just fer una llista ordenada segons la prioritat d'aquestes tasques, **totes són importants**. Potser en moments concrets algunes són més prioritàries que altres. *Exemple*: Si han fet fora a un empleat, serà prioritària la gestió d'usuaris, per tal de retirar-li a l'exempleat els seus permisos i privilegis i evitar que sabotegi la nostra empresa.

La gestió de xarxes es una feina que no correspon al 100% als administradors de sistemes. Aquests normalment s'ocupen de la configuració de xarxa dins dels hosts. Els **administradors de xarxes** s'ocupen dels nivells superiors.

## Nivells d'habilitat

---

### Administrador novell

- Coneixements i habilitats
  - Bones habilitats de comunicació
  - Familiaritzat amb el sistema operatiu i les seves comandes a nivell usuari
    - Capacitat per seguir instruccions sense dificultat (no pren decisions)
- Responsabilitats
  - Tasques rutinàries sota supervisió tècnica
  - Atenció directa als usuaris
  - Tasques bàsiques d'administració de xarxes i els seus serveis

## Administrador junior

- Coneixements i habilitats
  - Capaç d'ensenyar als usuaris l'ús de les aplicacions i comandes
  - Nivell elevat amb la majoria de comandes
  - Coneixements sobre fonaments de sistemes operatius
    - Planificació de processos, sistemes de fitxers, proteccions
  - Conceptes de ciberseguretat
- Responsabilitats
  - Administració d'una instal·lació petita
  - Participació en l'administració d'una instal·lació major, sota supervisió
  - Administració de xarxes i dels seus components

## Administrador senior

- Coneixements i habilitats
  - Capaç d'escriure ordres de compra i documentació de processos complexos
  - Comoditat amb la majoria d'aspectes del sistema
    - Configuració de servidors
    - Instal·lació i configuració del sistema operatiu
  - Fonaments de seguretat avançada
    - Eines de seguretat, sistemes pro-actius...
  - Coneixements avançats de sistemes operatius
    - Paginació, swap, comunicació entre processos, dispositius...
  - Familiaritat avançada en conceptes de xarxes
    - Routers, proxies, firewalls, serveis en xarxa
    - La seva configuració i administració a tots nivells
- Responsabilitats
  - Administració d'instal·lacions mitjanes
  - Responsabilitat parcial d'instal·lacions majors
  - Participació en la planificació futura de la instal·lació
  - Responsable d'altres administradors (juniors i novells)

## Codi d'ètica de l'administrador

---

No hi ha limitacions tecnològiques per a que un administrador accedeixi als correus electrònics o monitoritzi els usuaris. El codi d'ètica de l'administrador ens impedeix accedir a aquesta informació.

No podem fer la monitorització pel nostre compte, ha de ser "política de l'empresa" i els usuaris han d'estar informats. Si els contractes dels empleats diuen que la informació dels usuaris és de l'empresa, aquesta pot actuar legalment contra un empleat que usa els recursos de l'empresa de forma personal.

Els principis del codi d'ètica de l'administrador són:

- Professionalitat: Mantenir els aspectes personals al marge de la feina
- Integritat personal: Honestedat, capacitat per admetre les limitacions i els errors propis
- Privacitat: Mantenir la confidencialitat de les dades dels altres i accedir a informació privilegiada només quan és estrictament necessari
- Educació, lleis i polítiques d'ús: Mantenir-se informat sobre les normes que afecten a la seva feina, compartir el coneixement amb els altres
- Comunicació: Informar als usuaris d'allò que els afecta

## Código penal

---

És l'únic d'aquests que pot comportar penes de presó, tots els altres només comporten sancions econòmiques.

## Ley Orgánica de Protección de Datos (LOPD)

---

Regula el tractament dels fitxers de dades. La companyia/institució ha de notificar a la **Agencia de Protección de Datos** l'existència dels fitxers. L'agència ens dona un interval de temps per complir les mesures de seguretat indicades per la llei (backups, tipus de xifrat...). Els contractes hauran d'estar escrits en consonància amb la llei (recollida de dades en formularis).

## Ley de Servicios de la Sociedad de la Información y de Comercio Electrónico (LSSICE)

---

Regula les webs que tinguin finalitats econòmiques. Afecta més als desenvolupadors web que als administradors. Entre d'altres, exigeix que s'especifiqui clarament el preu dels productes i Spam. El Spam és l'enviament de correus (sovint publicitaris) sense autorització prèvia.

## Ley para el Impulso de la Sociedad de la Información (LISI)

---

Regula aspectes d'accessibilitat a les pàgines web. Entre d'altres, controla característiques com ara la mida de les fonts o el contrast dels colors.

## Coneixements de UNIX a nivell d'usuari (que no coneixia)

- `rmdir` : Elimina un directori buit. Amb `rmdir -p` eliminem els directoris superiors que també estiguin buits. Per exemple, `rmdir -p a/b/c` equival a `rmdir a/b/c a/b a` . *Ojo*, perquè en aquest cas, la comanda `rmdir a a/b a/b/c` donaria error al intentar eliminar primer `a` , que no està buit.
- `apropos` : Amb `apropos keyword` , busquem a quines pàgines del manual apareix la paraula `keyword` a la secció `NAME` de la pàgina. La secció sol ser una descripció en una sola línia de la comanda, com ara `ls - list directory contents` .
- `find path -name pattern` : Busca recursivament fitxers a partir del `path` que el seu nom coincideixi amb el `pattern` . *Ojo* amb aquest exemple: `find /home/david/ -name "*.err"` . Posem les `"` per evitar que la shell expandeixi l'asterisc, ja que volem que la shell no faci expansió i l'asterisc arribi a l'input del `find` .

## Particionat i preparació del disc

- `fdisk` : Fa particions en un disc
- `mkfs` : Crea un sistema de fitxers dins d'una partició
- `mkswap` : Crea una partició de swap
- `mount` : Munta un sistema de fitxers dins d'un altre.

## Operacions de redirecció de la shell (bash)

Per defecte, els *file descriptors* `0` , `1` , i `2` corresponen a `stdin` , `stdout` i `stderr` . Normalment els tres apunten a un mateix dispositiu terminal com `/dev/pts/n` (terminal gràfic) o `/dev/tty` (terminal com el que entres fent CTRL+ALT+F1).

Els *file descriptors* sempre tenen un fitxer o un dispositiu al darrere, que està obert amb uns permisos d'input i/o output.

### Redireccions `<`, `>`, `>>` i pipes |

- `command n<file` redirecciona el contingut del fitxer `file` cap al `fd` `n` . `<file` és el mateix que `0<file` .
- `command n>file` crea un nou fitxer `file` (l'elimina prèviament si ja existia) i redirecciona l'output del `fd` `n` cap al fitxer `file` . `>file` és el mateix que `1>file` . Si utilitzem l'operador `|>` , la redirecció fallarà si `file` ja existeix.
- `command n>>file` : similar a `>` , però si `file` ja existeix fa un *append* del nou output. `>>file`

equivale a redireccionar `stdout` cap a `file` .

- `command1 | command2` redirecciona l' `stdout` del `command1` cap a l' `stdin` del `command2` .

Ojo, si el *file descriptor* `n` no existeix, estarem obrint un nou *file descriptor*. També cal tenir en compte que la shell fa expansió de `file` . Si volem evitar-ho, caldrà escapar-ho d'alguna manera.

## Here strings i here documents

En resum, els *here strings* ( `<<<` ) ens permeten redireccionar un string cap al `stdin` i els *here docs* ens permeten fer el mateix amb un bloc multilínia ( `<<` ).

## Duplicació de *file descriptors*

A partir d'ara, `word` pot ser un dígit o un nom de fitxer.

- `n<&word` duplica un input *fd*. `n` passarà a apuntar a on apunta `word` . Per exemple, `>file 2>&1` redirecciona `stderr` i `stdout` cap a `file` . Però *muchísimo ojo* perquè bash interpreta les redireccions d'esquerra a dreta i `2>&1 >file` no seria equivalent, ja que acabaria amb `stderr` apuntant al terminal i `stdout` apuntant a `file` . Si no posem `n` equival a `stdin` .
- `n>&word` duplica un output *fd*. Es comporta de manera similar.
- `&>>word` equival a redireccionar `stdin` i `stderr` cap a `word` (amb *append*).

## Moure *file descriptors*

`n<&digit-` i `n>&digit-` fan que el *fd* `n` apunti a `digit` , i després tanquen el *fd* `digit` . `n<&-` i `n>&-` tanquen l'*fd* `n` . Si no especifiquem `n` , equivalen a `stdin` i `stdout` .