

SISTEMAS OPERATIVOS

2014

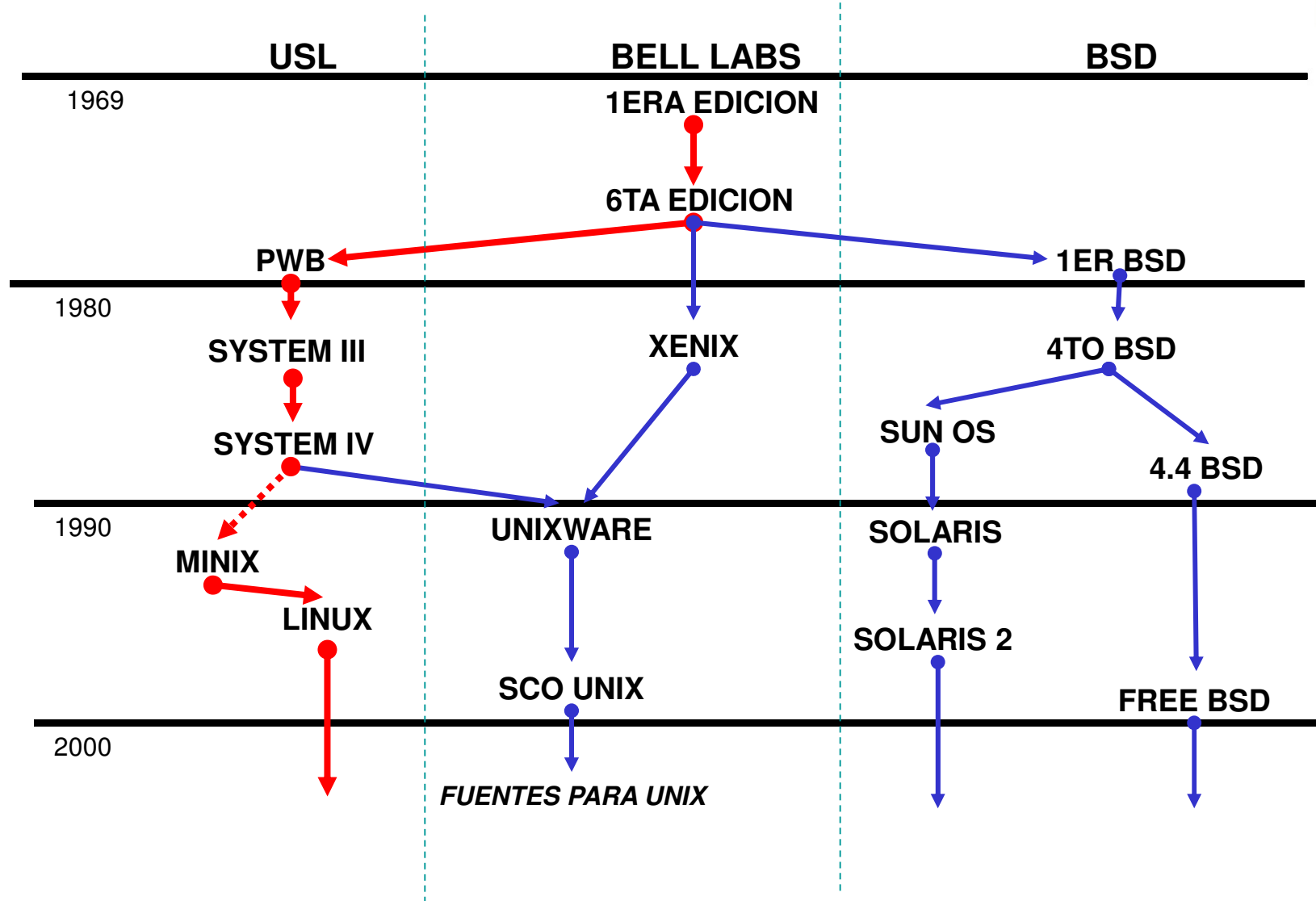
Parte A



TRABAJOS PRACTICOS 1 - 2

**" Creo que hay un mercado mundial para
alrededor de cinco computadoras. "
Tomas J. Watson, Fundador de IBM**

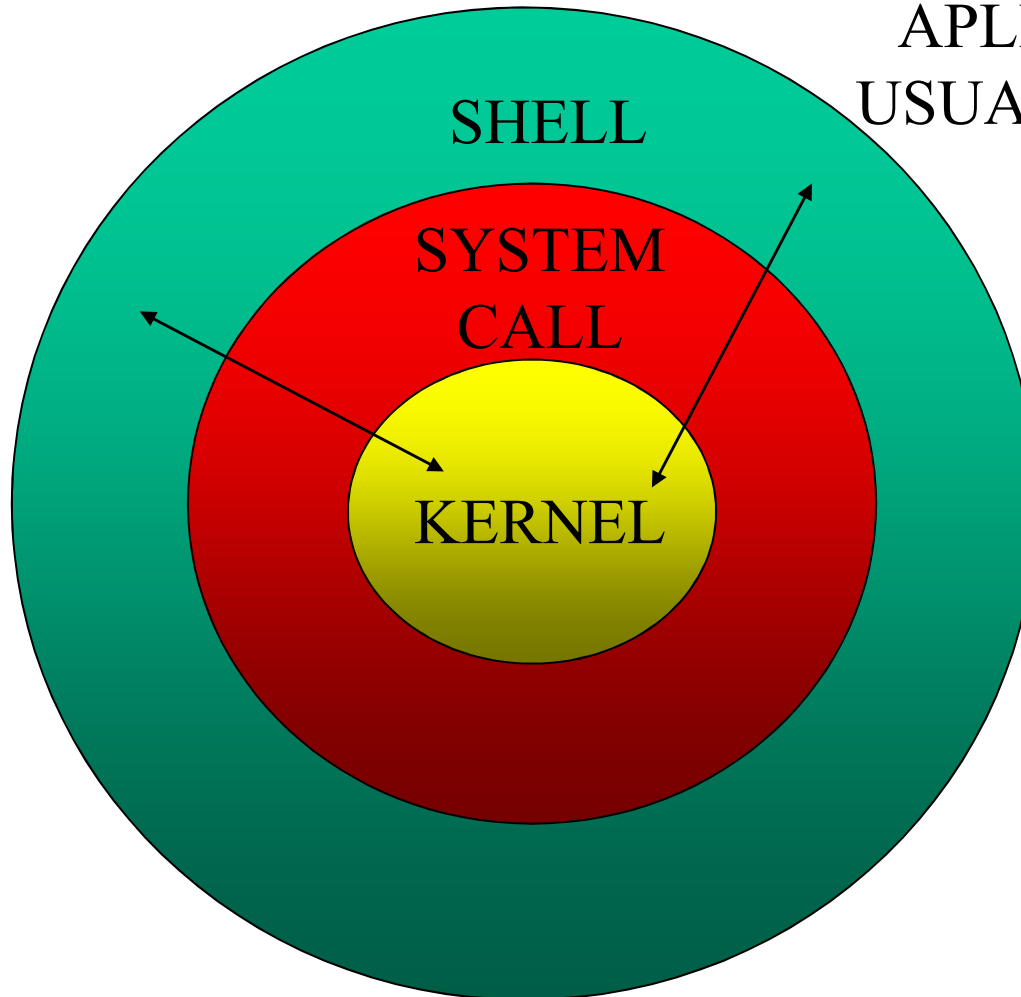
SISTEMA OPERATIVO LINUX - HISTORIA



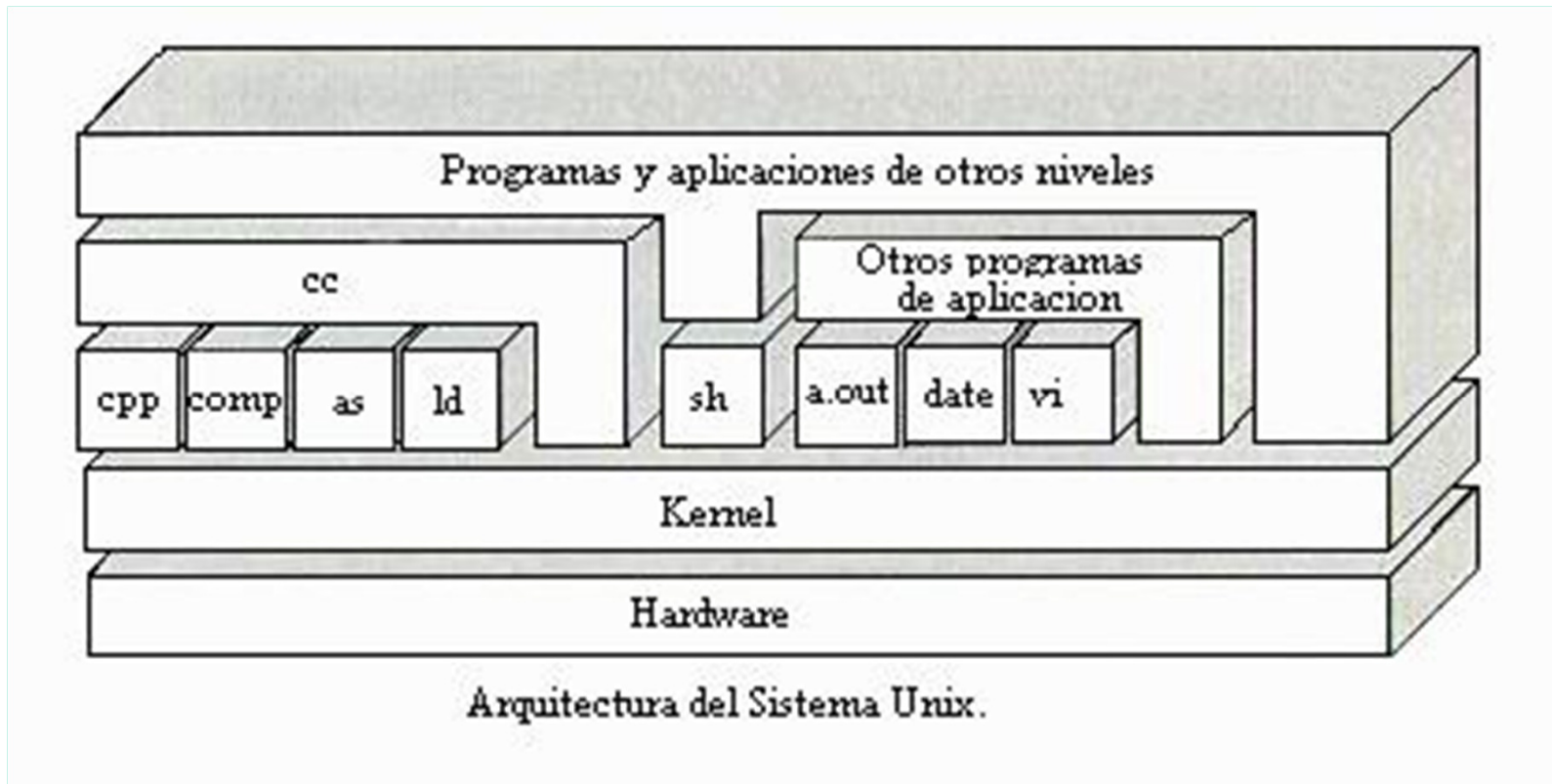
SISTEMA OPERATIVO LINUX



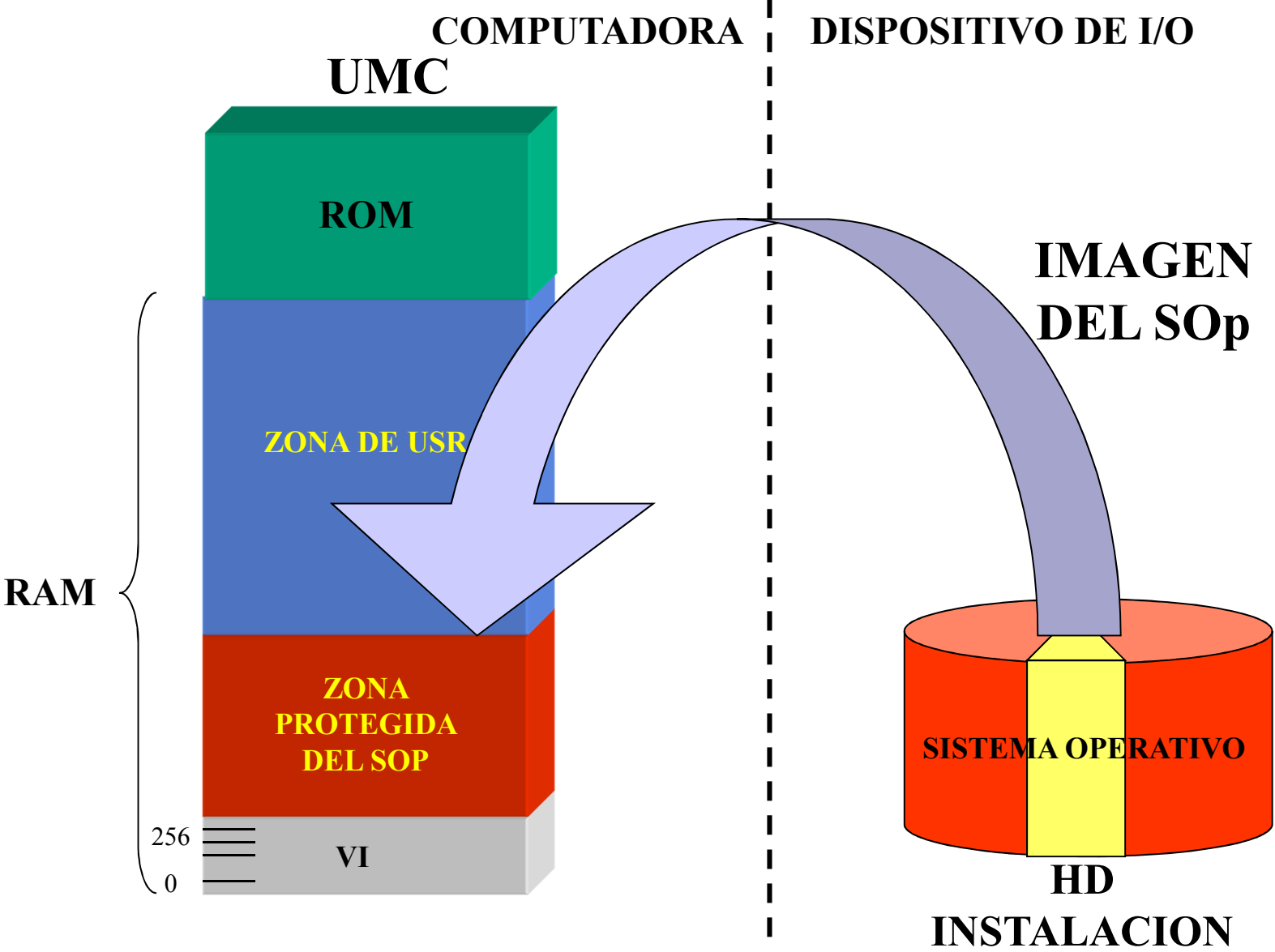
APLIC.
USUARIO



SISTEMA OPERATIVO LINUX



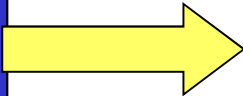
BOOT & CONFIGURACION DE UN SOP





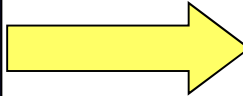
BOOT & CONFIGURACION DE LINUX

**ETAPAS DE
BOOT**



- A. Cargador de la BIOS
- B. Cargador LILO
- C. Núcleo del sistema operativo
- D. Primer programa de usuario /sbin/init

**ARCHIVOS
INTERVINIENTES**



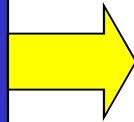
- A. LILO o GRU
- B. Archivo setup.S
- C. Archivo head.S
- D. Función start_kernel
- F. Proceso inicial INIT (init.c)



CONFIGURACION DE LINUX



Configuración del
`/etc/lilo.conf`



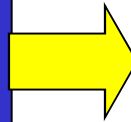
```
10: boot = /dev/had
20: prompt
30: timeout = 50
40: default = Linux
50: map = /boot/map
60: install = /boot/boot.b
70: image = /boot/bzImage
80: label = Linux
90: root = /dev/hda2
100: other = /dev/hda1
110: label = dos
120: table = /dev/hda
```



CONFIGURACION DE LINUX

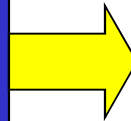


Cambio de Kernel
en /etc/lilo.conf



```
10: image = /boot/bzImage  
20: label = Linux  
30: /dev/hd2
```

Agregar otros
SOs en
/etc/lilo.conf



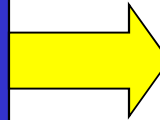
```
10: other = /dev/hda1  
20: label = dos  
30: table = /dev/hda
```



CONFIGURACION DE LINUX



Configuración del
/etc/inittab
(runlevels)



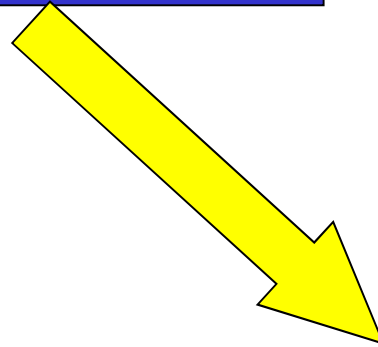
```
# Los runlevels son:  
# 0- Parado (No por default)  
# 1- Modo Monousuario  
# 2- Multiusuario, sin NFS (idem 3 si no hay red)  
# 3- Multiusuario completo  
# 4- Sin uso  
# 5- X11  
# 6- Reset (No por default)  
id:3:initdefault  
# Inicialización del sistema  
si::sysinit:/etc/rc.d/rc.sysinit  
10:0:wait:/etc/rc.d/rc 0  
11:1:wait:/etc/rc.d/rc 1  
12:2:wait:/etc/rc.d/rc 2  
13:3:wait:/etc/rc.d/rc 3  
14:4:wait:/etc/rc.d/rc 4  
15:5:wait:/etc/rc.d/rc 5  
16:6:wait:/etc/rc.d/rc 6
```



CONFIGURACION DE LINUX



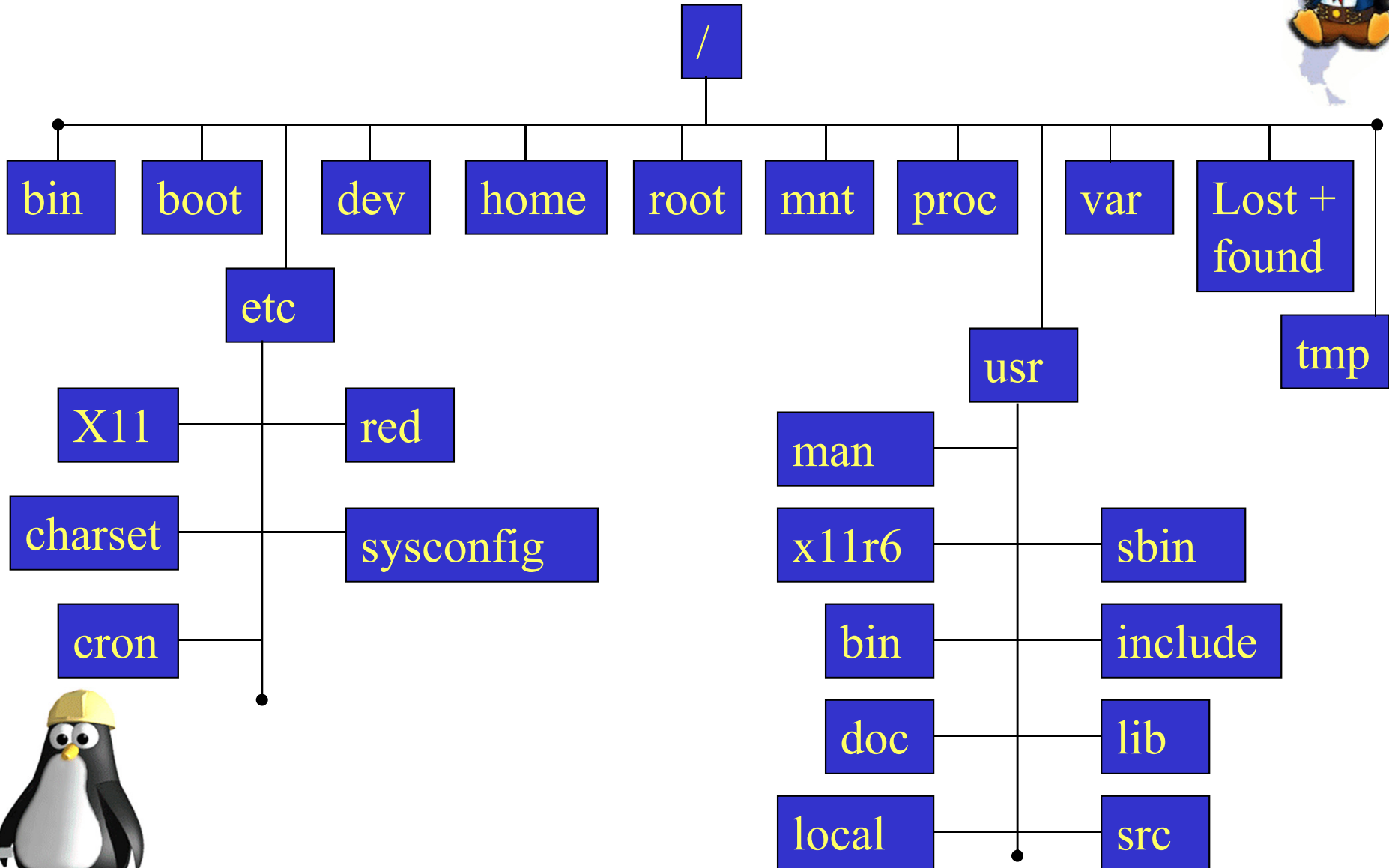
Configuración del
/etc/inittab
(exec in runlevels)



```
#Ejecuciones en cada nivel de ejecución
ud::once:/sbin/uptdate
#Atrapara el Ctrl-Alt-Delete
ca: :ctrlaltdel:/sbin/shutdown -t3 -r now
#Falla de alimentación
Pf::powerfail:/sbin/shutdwn -f -h +2
"Power Failure; System Shutting Down"
#Cancelación del Shutdown
pr:12345:powerkwait:/sbin/shutdown -c
"Power Restored; Shutdown Cancelled"
#Ejecución de gettys en niveles de #ejecución
standard
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
...
6:2345:respawn:/sbin/mingetty tty6
```



FILE SYSTEM GENERAL DE LINUX



FILE SYSTEM GENERAL DE LINUX



bin

→ Arch. Binarios(ejecutables), shells, bash, configuraciones del sistema, utilerias básicas de manejo de archivos

sbin

→ Arch. Binarios(ejecutables) del sistema como init, LILO, mkfs. La mayoría solo ejecutables por el root

boot

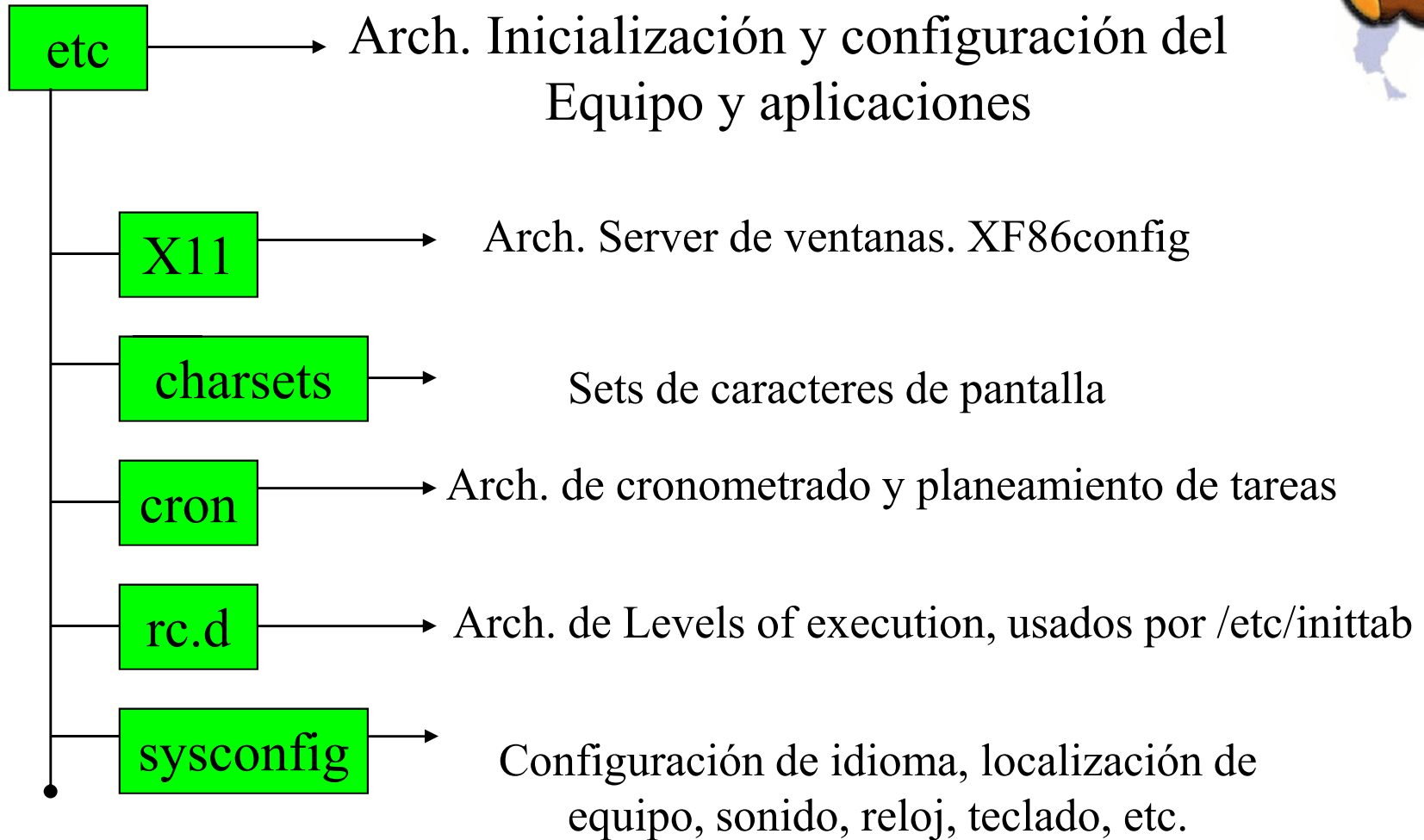
→ Arch. del Kernel y de boot. zImage, vmlinuz-version, bzImage.

dev

→ Archivos de Dispositivos, /dev/lpt01: /dev/pts01: /dev/stdout : etc.

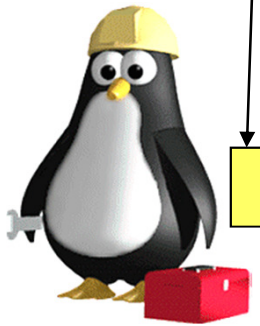
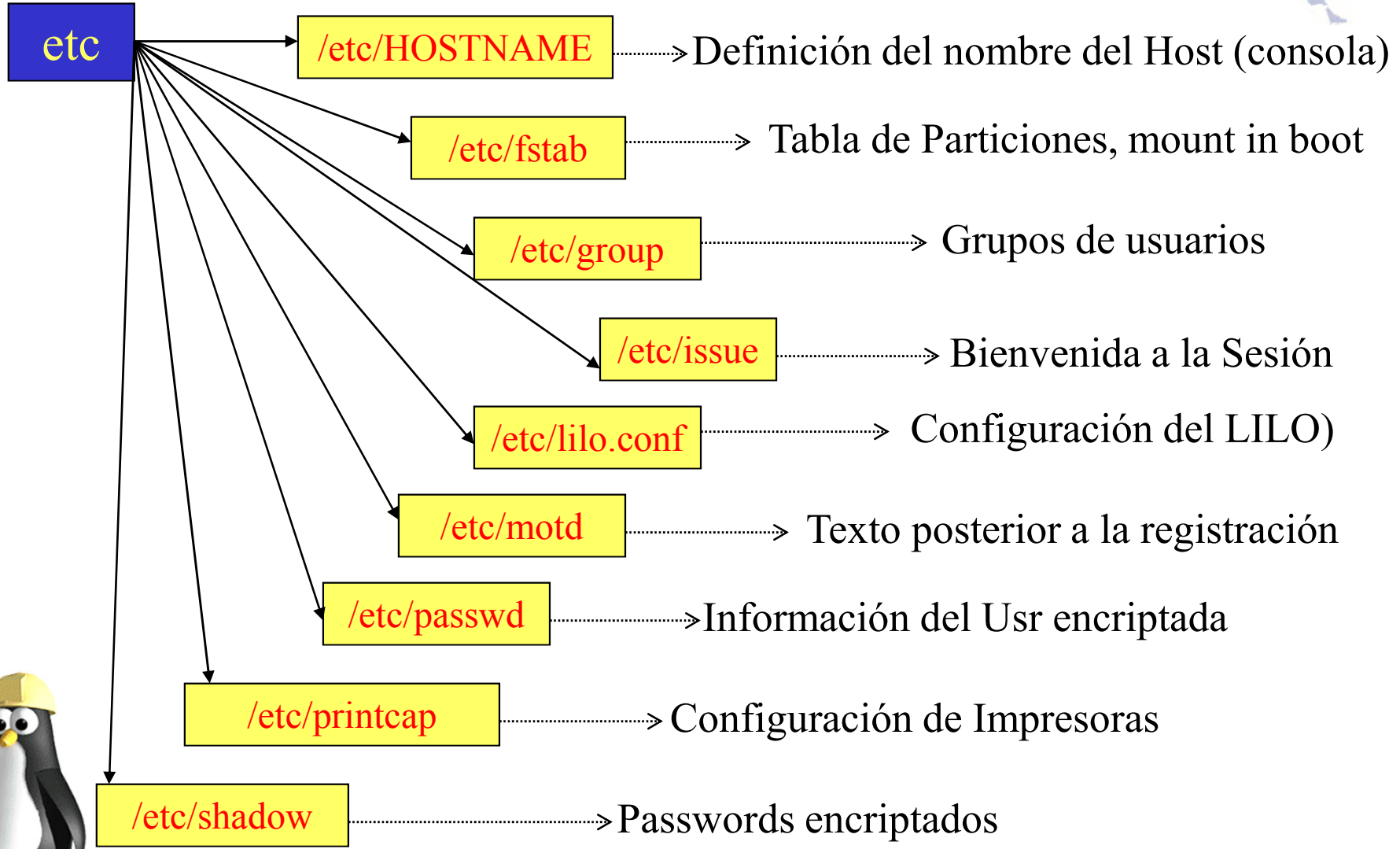


FILE SYSTEM GENERAL DE LINUX





FILE SYSTEM GENERAL DE LINUX



FILE SYSTEM GENERAL DE LINUX



home

→ Contiene todos los Directorios de Usr y Server de ftp anónimo

root

→ Aplicaciones del Superusuario, propia configuración, documentación. Solo root

proc

→ Contiene toda la informacion de configuración del equipo. En ASCII Esta kcore que representa toda la memoria física del sistema

var

→ Definiciones de configuración, ej. Colas de impresión

mnt

→ Montaje del floppy y cdrom

lost + found

→ Papelera del sistema



FILE SYSTEM GENERAL DE LINUX



usr

/usr/X11R6

→ Arch. del sistema de ventanas X

/usr/bin

→ Mayoría de comandos y utilerías de Linux

/usr/doc

→ Docum. de Aplic. y comandos

/usr/games

Juegos

/usr/include

Headers de C y C++

/usr/lib

Librerías de C y C++

/usr/man

Manuales de Linux

/usr/sbin

Ejecutables de Admin. del sistema

/usr/src

→ Código fuente del Kernel. Debe descomprimirse con cada upgrade





COMANDOS DE LINUX

Estructura de los Comandos:

```
$ COMANDO PARAMETRO 1 PARAMETRO 2...
```

Concatenación de Comandos:

```
$ COMANDO; COMANDO; COMANDO...
```

Caracteres comodín:

```
$ COMANDO <*, ?, ...>
```





COMANDOS DE LINUX

Redireccionamiento de Entrada/Salida:

(Entre Archivos comunes)

```
$ ARCHIVO1 > ARCHIVO 2
```

```
$ ARCHIVO1 >> ARCHIVO 2
```

(Sólo Ordenes/Comandos):

```
$ COMANDO | COMANDO
```

Ejercicio – Que hace esta línea escrita?-

```
/home/pp/papels$ ls | sort -r | head -1
```





COMANDOS DE LINUX

Permisos de un Archivo:

```
$ chmod {a,u,g,o} {+,-} {r,w,x} <nombre archivo>
```

Ejercicio – Que hace esta línea escrita?-

```
/home/pp/papels$ chmod og -x pp.bin
```

Hard Links a un archivo:

```
$ ln <archivo> <archivo>
```

Ejercicio – Que hace esta línea escrita?-

```
$ ln pp.txt perez.txt
```





COMANDOS DE LINUX

PROCESOS EN BACKGROUND

```
$ COMANDO &
```



```
RESP: [1] 638 <<n° trabajo en back ; PID>>
```

Ejercicio – Que hace esta línea escrita?-

```
$ yes > /dev/null &
```

MUERTE DE UN PROCESO

```
$ KILL <SIGNAL SOP> <PID PROC>
```





COMANDOS DE LINUX

Comando/Sintaxis	Descripción	Ejemplos
<code>cat fich1 [...fichN]</code>	Concatena y muestra un archivos	<code>cat /etc/passwd</code>
	archivos	<code>cat dict1 dict2</code>
<code>cd [dir]</code>	Cambia de directorio	<code>cd /tmp</code>
<code>chmod permisos fich</code>	Cambia los permisos de un archivo	<code>chmod +x miscript</code>
<code>chown usuario:grupo fich</code>	Cambia el dueño un archivo	<code>chown nobody miscript</code>
<code>cp fich1...fichN dir</code>	Copia archivos	<code>cp foo foo.backup</code>
<code>diff [-e]arch1 arch2</code>	Encuentra diferencia entre archivos	<code>diff foo.c newfoo.c</code>
<code>du [-sabr] fich</code>	Reporta el tamaño del directorio	<code>du -s /home/</code>
<code>file arch</code>	Muestra el tipo de un archivo	<code>file arc desconocido</code>





COMANDOS DE LINUX

<code>find dir test acción</code>	Encuentra archivos.	<code>find . -name ``</code>
<code>grep [-c l nv] expr archivos</code>	Busca patrones en archivos	<code>grep mike /etc/passwd</code>
<code>head -count fich</code>	Muestra el inicio de un archivo	<code>head prog1.c</code>
<code>mkdir dir</code>	Crea un directorio.	<code>mkdir temp</code>
<code>mv fich1 ...fichN dir</code>	Mueve un archivo(s) a un directorio	<code>mv a.out prog1</code>
<code>mv fich1 fich2</code>	Renombra un archivo.	<code>mv</code>
<code>less / more fich(s)</code>	Visualiza página a página un archivo.	<code>more muy_largo.c</code>
	less acepta comandos vi.	<code>less muy_largo.c</code>
<code>ln [-s] fich acceso</code>	Crea un acceso directo a un archivo	<code>ln -s /users/mike/.profile .</code>





COMANDOS DE LINUX

ls	Lista el contenido del directorio	ls -l /usr/bin
pwd	Muestra la ruta del directorio actual	pwd
rm <i>fich</i>	Borra un fichero.	rm foo.c
rm -r <i>dir</i>	Borra un todo un directorio	rm -rf prog_dir
rmdir <i>dir</i>	Borra un directorio vacío	rmdir prog_dir
tail -count <i>fich</i>	Muestra el final de un archivo	tail prog1.c
vi <i>fich</i>	Edita un archivo.	vi .profile

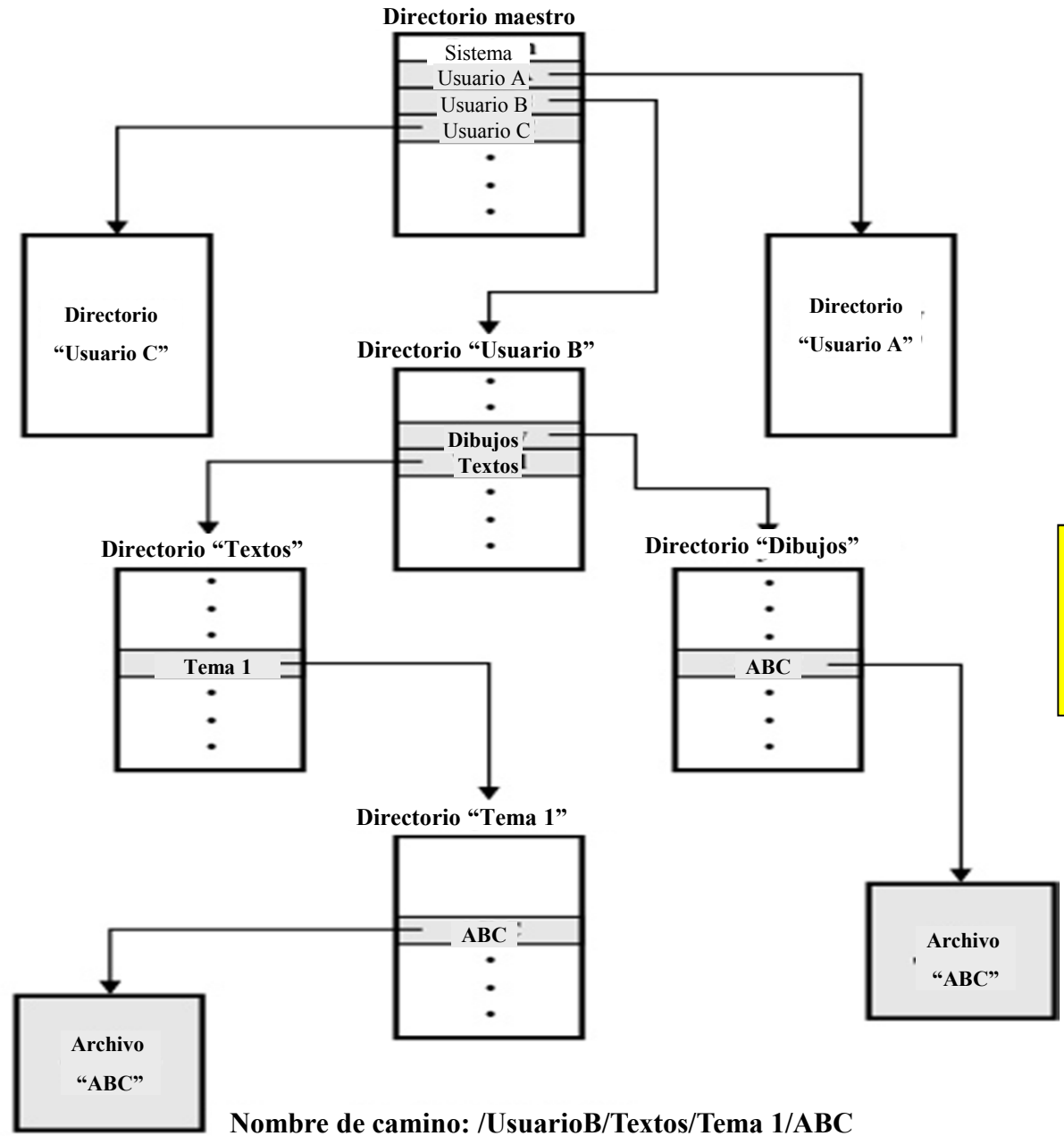


COMANDOS DE ARCH. Y DIRECTORIOS DE LINUX



Comando/Sintaxis	Descripción	Ejemplos
<code>at [-lr] hora [fecha]</code>	Ejecuta un comando mas tarde	<code>at 6pm Friday</code>
<code>cal [[mes] año]</code>	Muestra un calendario del mes/año	<code>cal 1 2025</code>
<code>date [mmdhmm] [+form]</code>	Muestra la hora y la fecha	<code>date</code>
<code>echo string</code>	Escribe mensaje en la salida estándar	<code>echo ``Hola mundo``</code>
<code>finger usuario</code>	Muestra información general sobre un usuario en la red	<code>finger nn@maquina.aca.com.co</code>
<code>id</code>	Número id de un usuario	<code>d usuario</code>
<code>kill [-señal] PID</code>	Matar un proceso	<code>kill 1234</code>
<code>man comando</code>	Ayuda del comando especificado	<code>man gcc</code>
		<code>man -k printer</code>
<code>passwd</code>	Cambia la contraseña.	<code>passwd</code>
<code>ps [axiu]</code>	Muestra información sobre los procesos que se están ejecutando en el sistema	<code>ps -ux</code>
		<code>ps -ef</code>
<code>who / rwho</code>	Muestra información de los usuarios conectados al sistema.	<code>who</code>



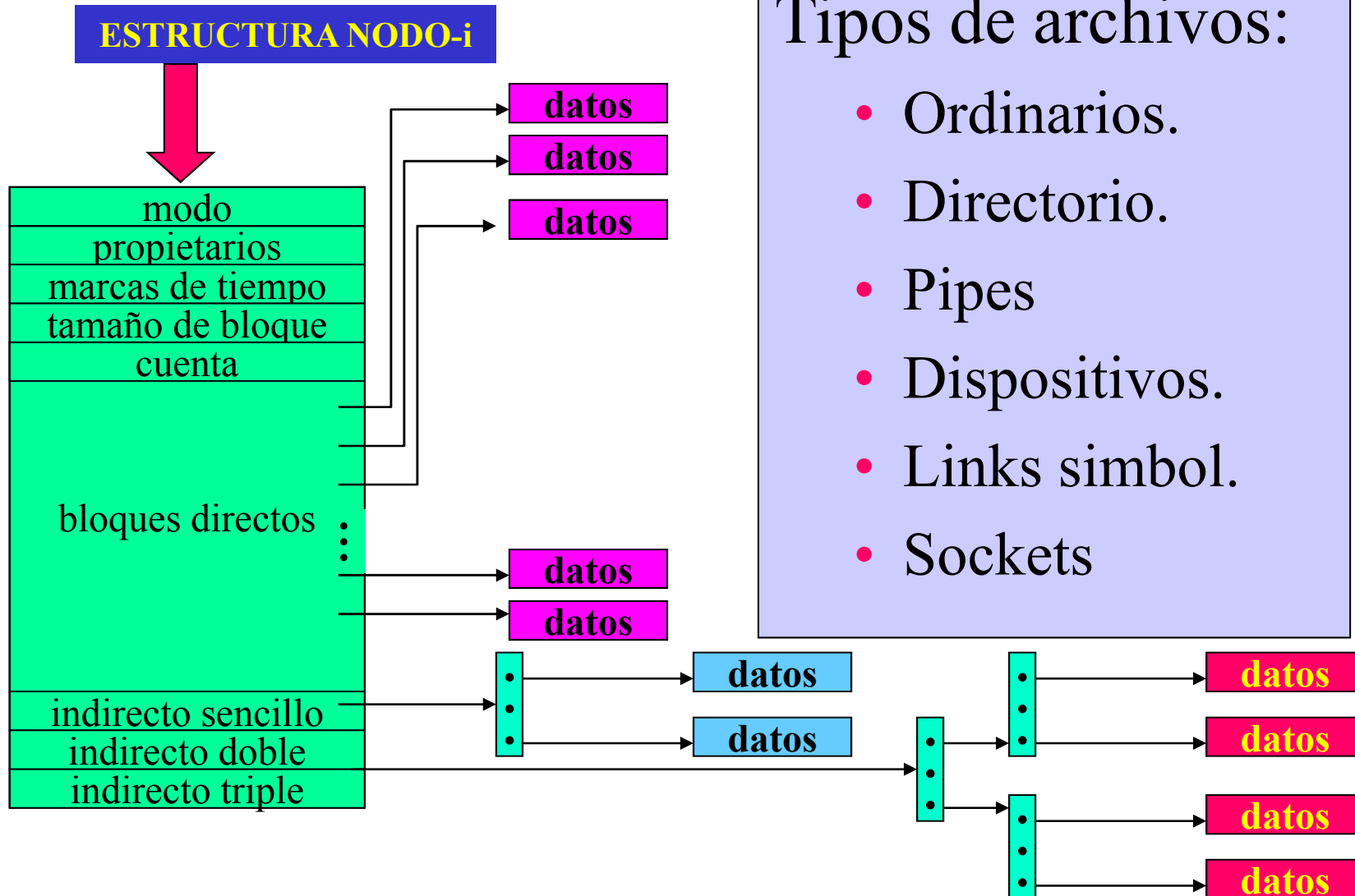


**DIRECTORIOS:
ESTRUC. LOGICA**

Figura 12.5. Ejemplo de directorio estructurado en Árbol.



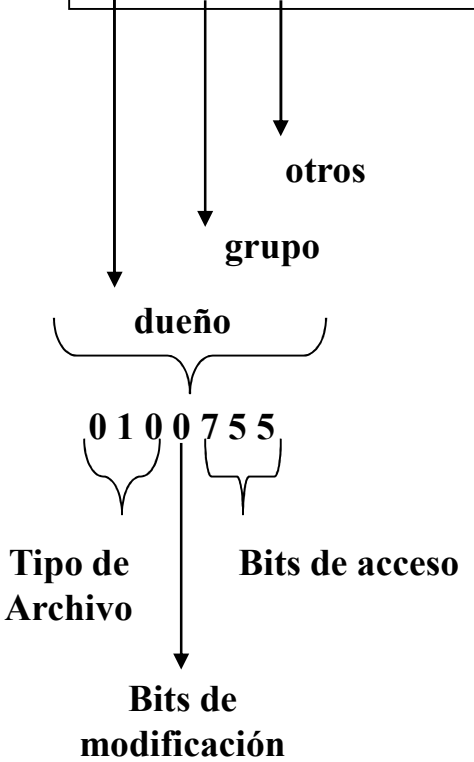
GESTION DE ARCHIVOS EN LINUX





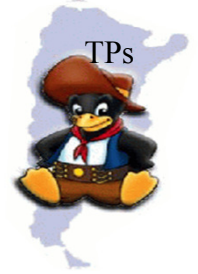
MODO DE UN ARCHIVO

```
$ ls /usr/bin/ls  
-rwxr-wr-x 1 root root 122+2 Mar 26 12:45 /usr/bin/ls
```



- Bits de modificación:**
- setuid
 - setgid
 - sticky

- Tipo de Archivo:**
- Socket
 - Vinculo simbólico
 - FIFO
 - Arch. Convencional
 - Directorio
 - Dispositivo de Bloque
 - Dispositivo de Caracter



INTERFAZ DE ADM. DE ARCHIVOS

a. Apertura y Cierre de Archivos (open y creat)

`int open (const char, *pathname, int flags)`

`int open (const char, *pathname, int flags, mode_t mode)`

`int creat(const char, *pathname, mode_t mode)`

equivale a:

`open(pathname,
O_CREAT|O_TRUNC|O_WRONLY, mode)`

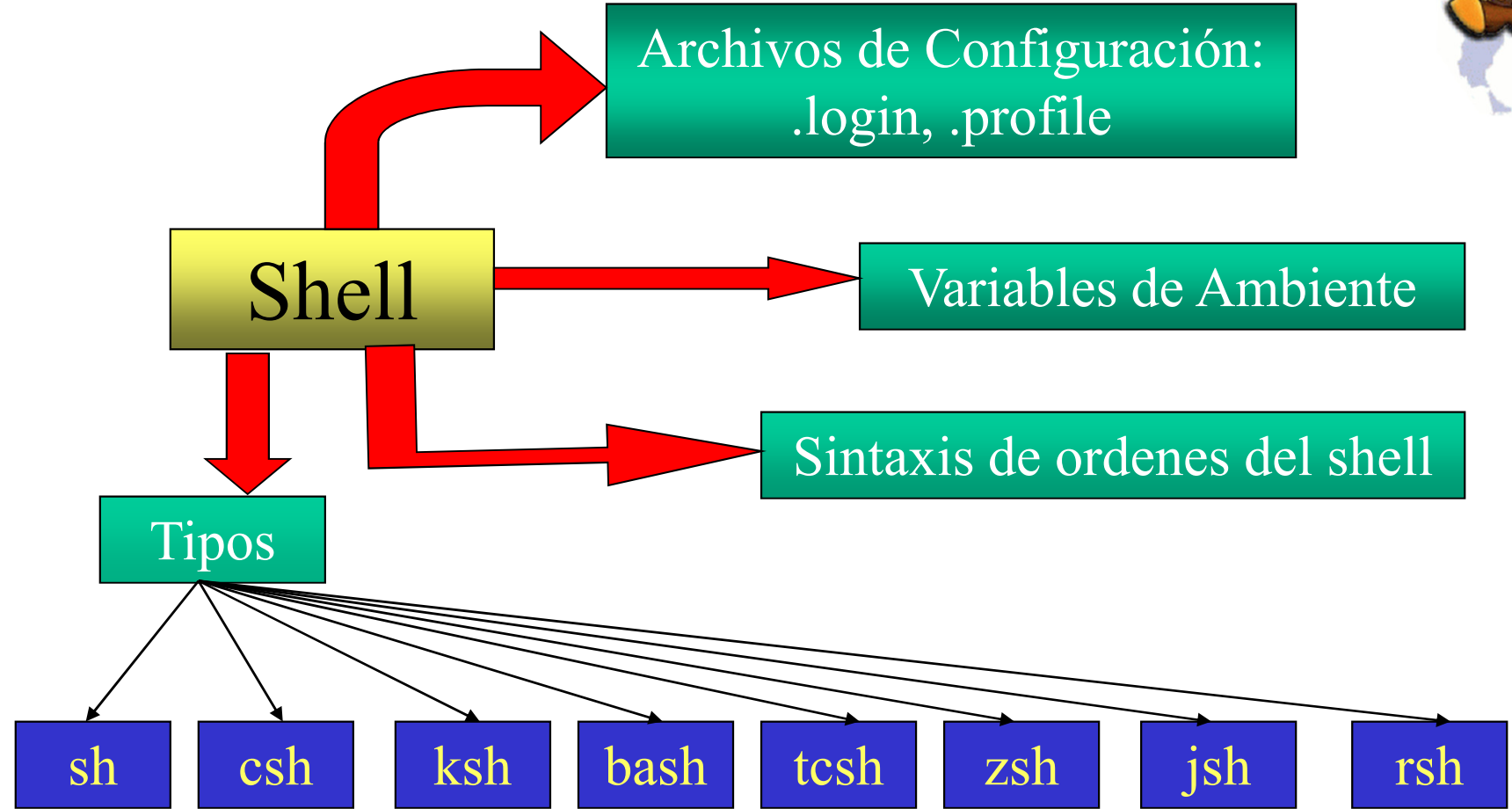
b. Cierre de Archivos.

c. Lectura/Escritura.

d. Truncamiento.



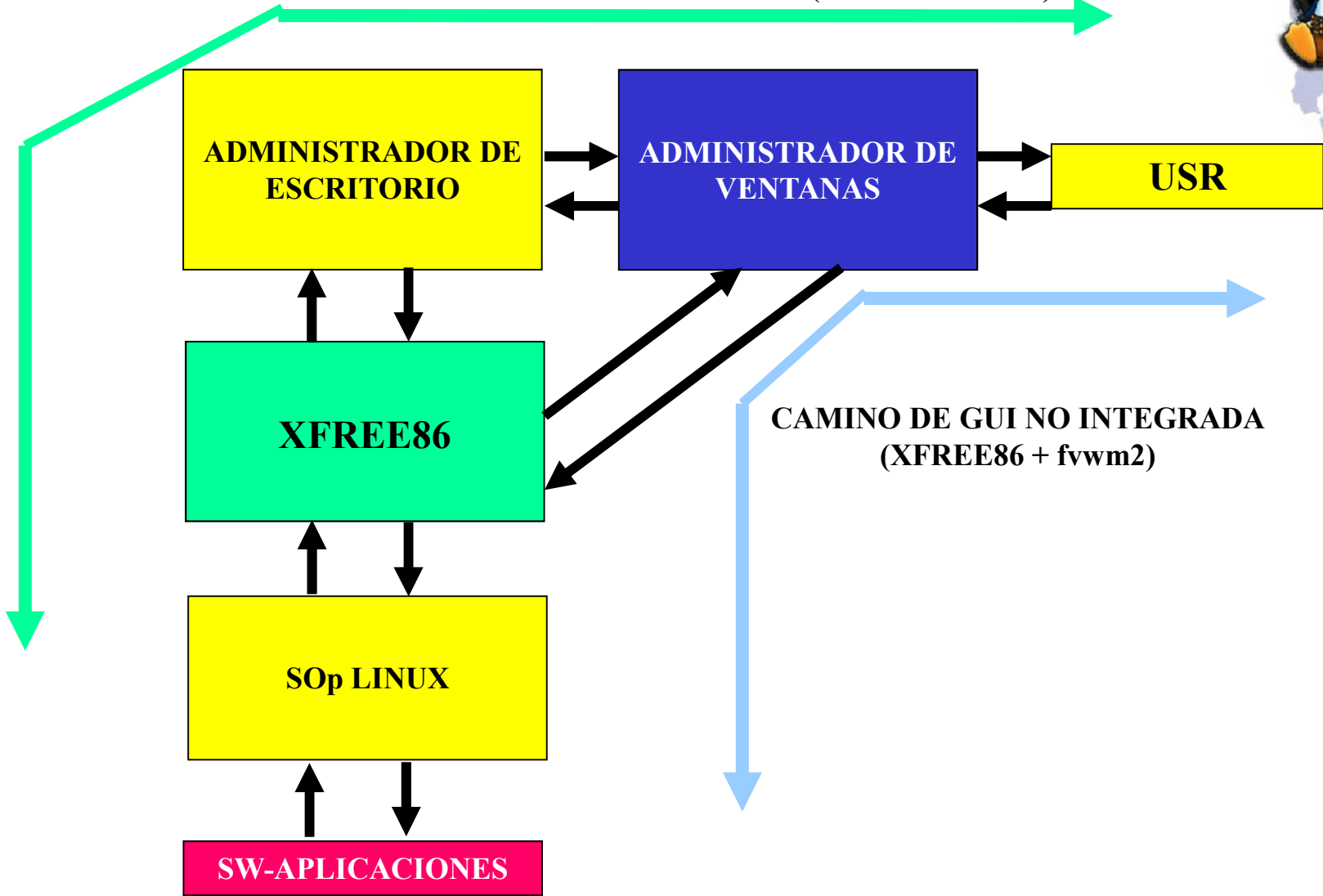
INTERPRETE DE COMANDOS DE LINUX



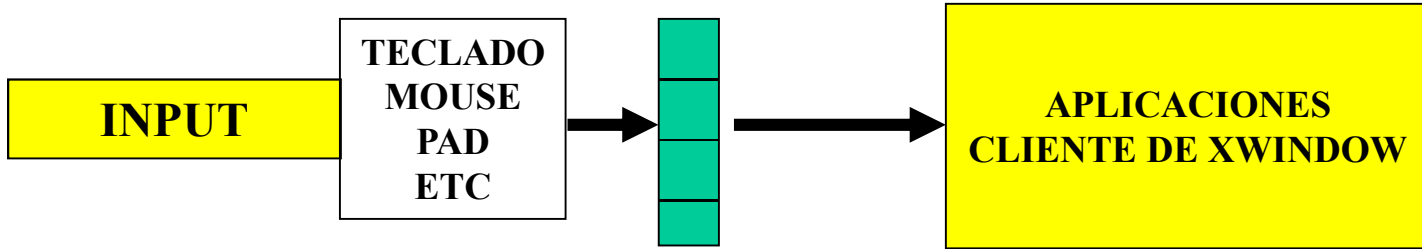
INTERFACES GRAFICAS



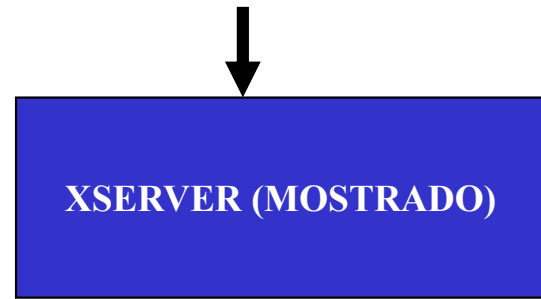
CAMINO DE GUI INTEGRADA (GNOME - KDE)



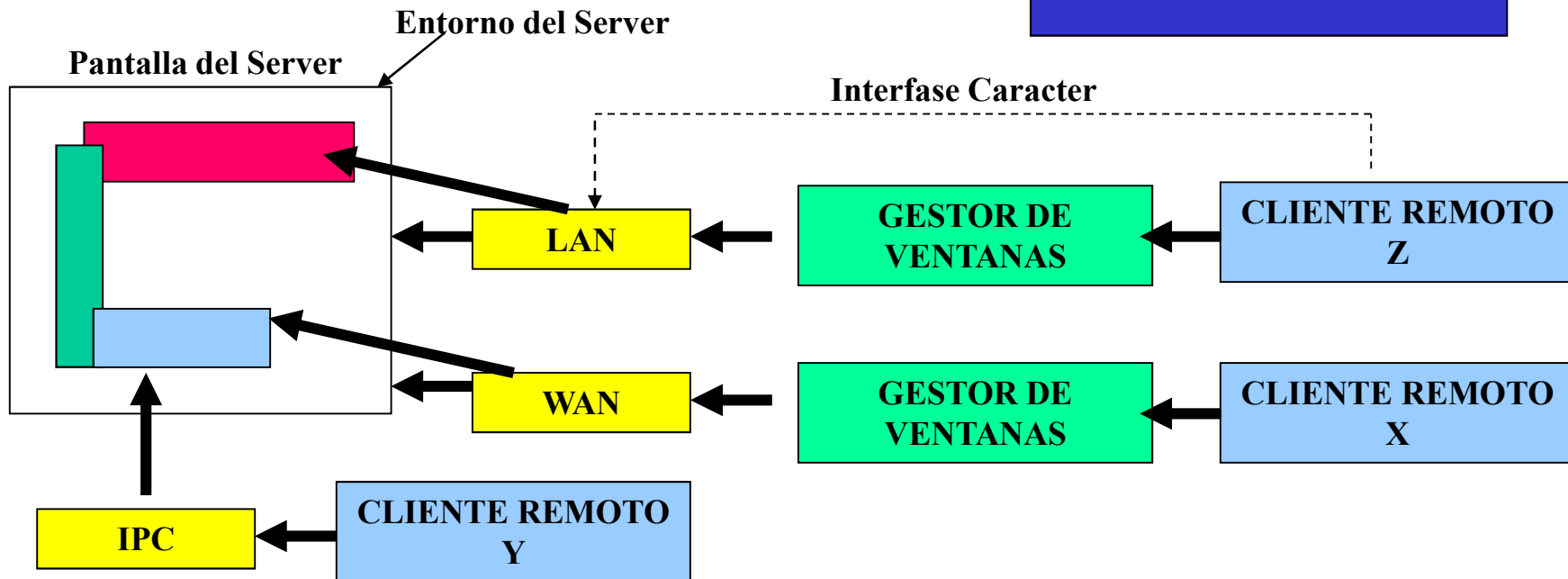
INTERFACES GRAFICAS



MODELO CLIENTE/SERVIDOR



TOPOLOGIA



COMUNICACIÓN ENTRE USUARIOS



Uso del mail:

```
# mail pepe
```

```
Subject: Trabajo Practico
```

```
Queria hacerte acordar que el TP IV y V debemos entregarlo antes del fin del 2do Cuatrimestre!
```

```
[CTRL-D]
```

```
Cc: [ENTER]
```

```
[root@localhost /] #
```

Que ve el usuario pepe:

```
Mail version 8.1.6/7/94. Type ? for help
```

```
“/var/spool/mail/pepe”: 1 message new
```

```
>N 1 root@localhost Sa Sep 27 11:15 13/999 “Trabajo Practico”
```

```
&
```



COMUNICACIÓN ENTRE USUARIOS



Uso del write:

```
# write pepe  
Mensaje (.....)  
[CTRL-D]
```

Inhibición de recepción de mensajes :

```
# mesg [y/n]
```

Uso del talk:

```
# talk pepe/localhost
```

Mensajes para todos los usuarios:

```
# wall Bajar las sesiones El sistema se cerrara en 5 min. El Admin
```

Que ve cada usuario:

```
Broadcast message from root: Bajar las sesiones El sistema se cerrara en 5 min
```





BIBLIOGRAFIA DE REFERENCIA

1. Programación en Linux, con ejemplos. Kurt Wall. QUE, Prentice Hall. Madrid. 2000.
2. Sistemas Operativos. 5ta Ed. William Stalling. Pearson Prentice Hall. Madrid. 2006
3. Sistemas Operativos. 7ma Ed. William Stalling. Pearson Prentice Hall. Madrid. 2012
4. Sistemas Operativos Modernos. Andrew. S. Tanenbaum. Prentice-Hall. Interamericana S.A. Madrid, 2009.
5. Unix, Sistema V Versión 4. Rosen, Rozinsky y Farber. McGraw Hill. NY 2000.
6. Linux, Edición especial. Jack Tackett, David Guntery Lance Brown. Ed. Prentice Hall. 1998.
7. El Libro de Linux. Syed M. Sarwar, Robert Koretsky y Syed. A. Sarwar. Ed. Addison Wesley. 2007. España.



**FIN DE TPs PARTE A
DE SOp**

