

# Procesos

Concepto. Estructura. Bloque de control de procesos (PCB). Entorno y contexto. Estados y transiciones. Planificadores. Algoritmos de planificación. Procesos independientes y cooperativos. Procesos Concurrentes. Paralelismo. Hilos: concepto, objetivos e implementación.

# Proceso

Con el surgimiento de la ***multiprogramación***, con el fin de aprovechar mejor los recursos de la computadora, surge el concepto de **proceso**.

Es decir que podemos tener cargados en memoria varios procesos y estos se ejecutarán concurrentemente multiplexando la CPU.

Esto requiere mayor control y aislamiento.

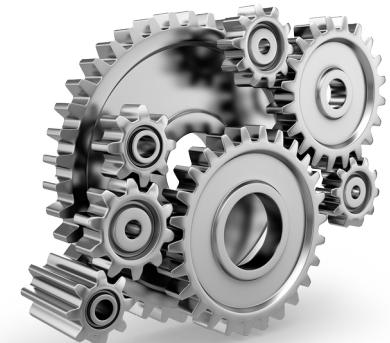


# Procesos

*“Un proceso es un programa en ejecución.”*

*“Es la unidad de trabajo de los Sistemas Operativos modernos.”*

Existen procesos de usuarios y procesos del Sistema Operativo.



# Procesos y Programas

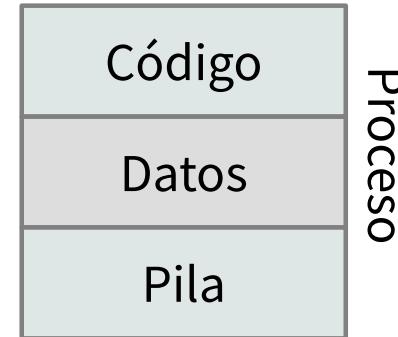
“Un **programa** es una entidad **pasiva** y un **proceso** es una entidad **activa**”

Programa	Proceso
Estático	Dinámico
No tiene un contador de programa (PC)	Tiene un contador de programa (PC)
Existe desde que se crea y edita el archivo hasta que se elimina	Existe desde que se invoca hasta que finaliza su ejecución

# Estructura de un Proceso

Un proceso está compuesto básicamente por 3 regiones:

- **Código:** también llamado texto. Es el código de las funciones del programa. Es de sólo lectura.
- **Datos:** variables globales sobre las cuales trabaja.
- **Pila:** datos temporales (parámetros de funciones, variables locales, direcciones de retorno)



# Bloque de Control de Proceso

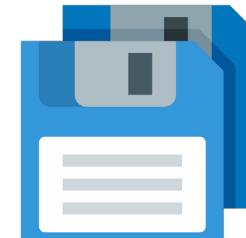
Cada proceso es representado por una estructura llamada ***Bloque de Control de Proceso (PCB)*** que contiene información específica asociada al proceso.

Cumple las siguientes funciones:

- ***Localización***: permite al Sistema Operativo localizar al proceso o acceder a su información.



- ***Persistencia***: permite almacenar la información en caso de que la ejecución del proceso sea suspendida y luego deba ser retomada.



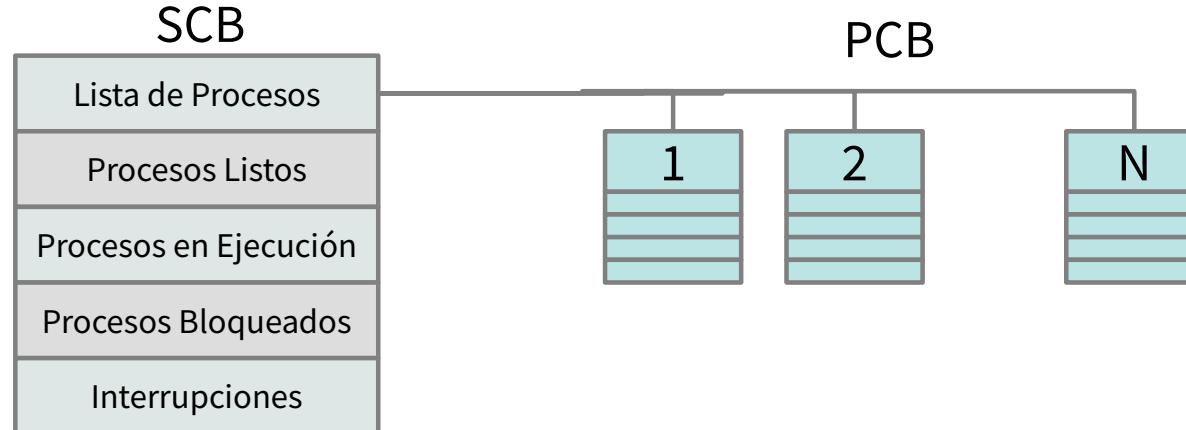
# Bloque de Control de Proceso > Estructura

- Número de Proceso: identificador.
- Estado: nuevo, listo, en ejecución, detenido, etc.
- Contador de Programa: dirección de la instrucción a ejecutar.
- Registros de la CPU: deben guardarse cuando la ejecución se interrumpe, para poder continuar luego.
- Información de Gestión de Memoria: registros base y límite, tabla de páginas o segmentos.
- Información E/S y Archivos abiertos: lista de dispositivos y archivos asignados al proceso.
- Estadísticas: información de utilización de CPU, memoria, etc.

Número de Proceso
Estado
Contador de Programa
Registros de la CPU
Información de Memoria
Archivos Abiertos
Información E/S
Estadísticas

# Bloque de Control del Sistema (SCB)

- Lista de los PCB.
- Punteros a colas de procesos en diferentes situaciones.
- Identificadores de las rutinas para tratar interrupciones



# Contexto y Espacio de Direcciones

**Contexto:** está compuesto por la información que necesita el Sistema Operativo para administrar el proceso y por la información que necesita el procesador para ejecutar el mismo.

¿Cuál es esa información?:

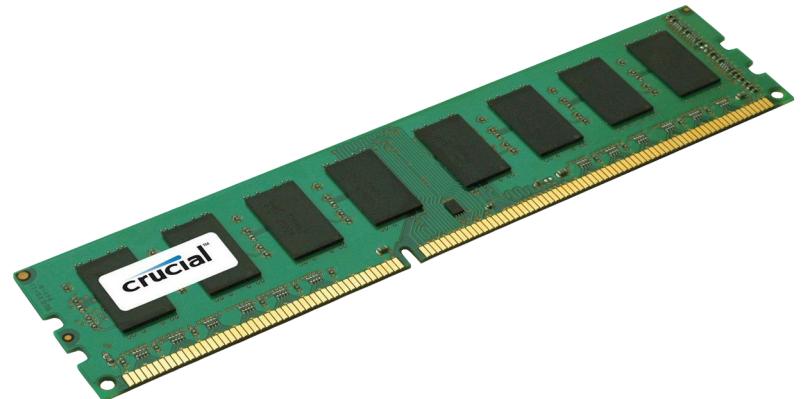
- Registros del procesador.
- Contador de Programa (PC).
- Información de prioridades.
- Información de operaciones E/S pendientes...



# Contexto y Espacio de Direcciones

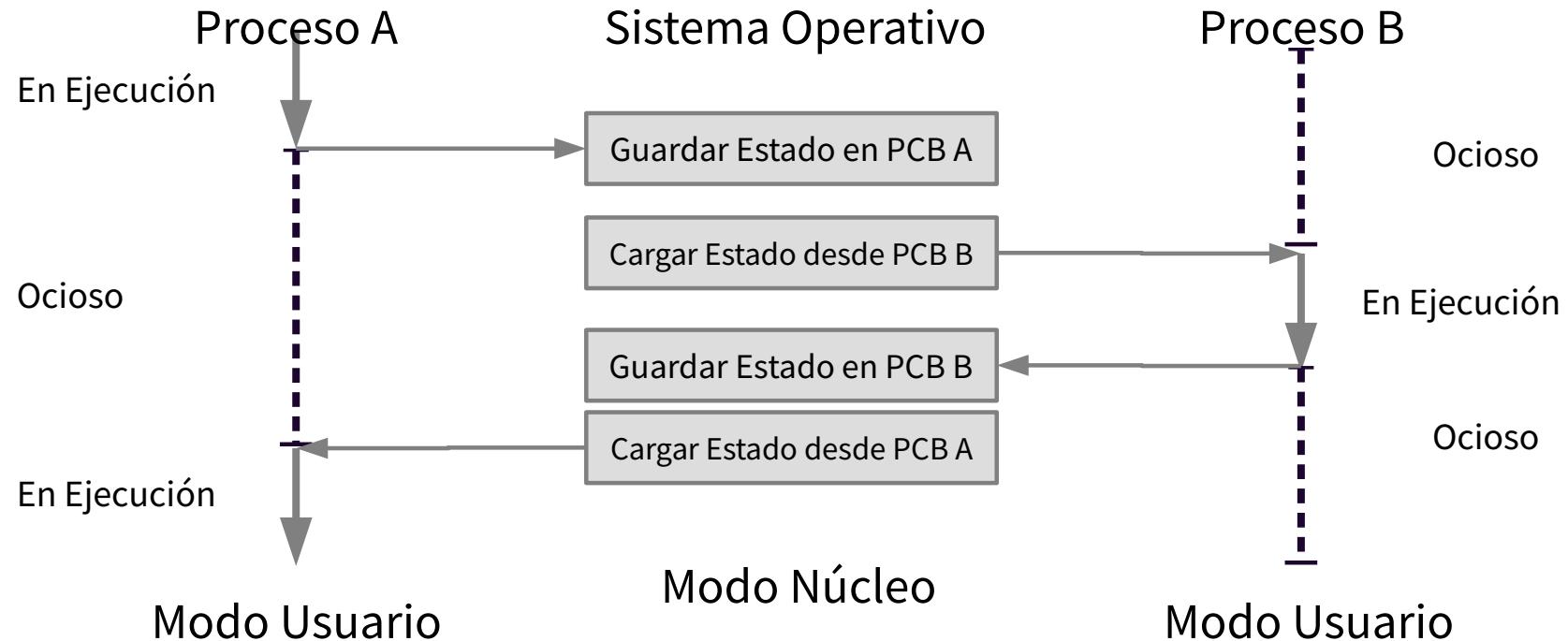
**Espacio de Direcciones:** está compuesto por el conjunto de direcciones de memoria que ocupa el proceso.

- No incluye al PCB, ni demás tablas asociadas al proceso.
- El acceso al mismo lo realiza el proceso en modo usuario.

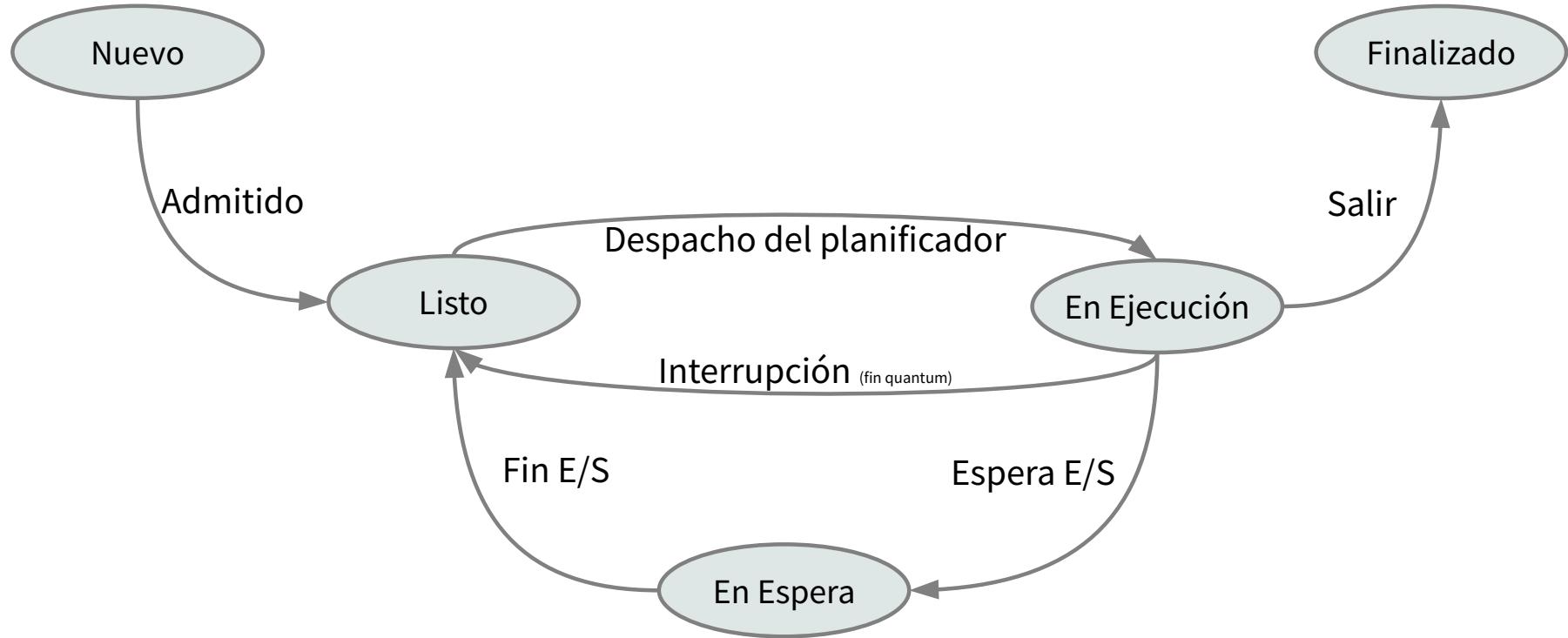


**Entorno:** está compuesto por el contexto junto con el espacio de direcciones.

# Cambio de Contexto



# Estados de los Procesos





Facultad de Ciencias de la Administración

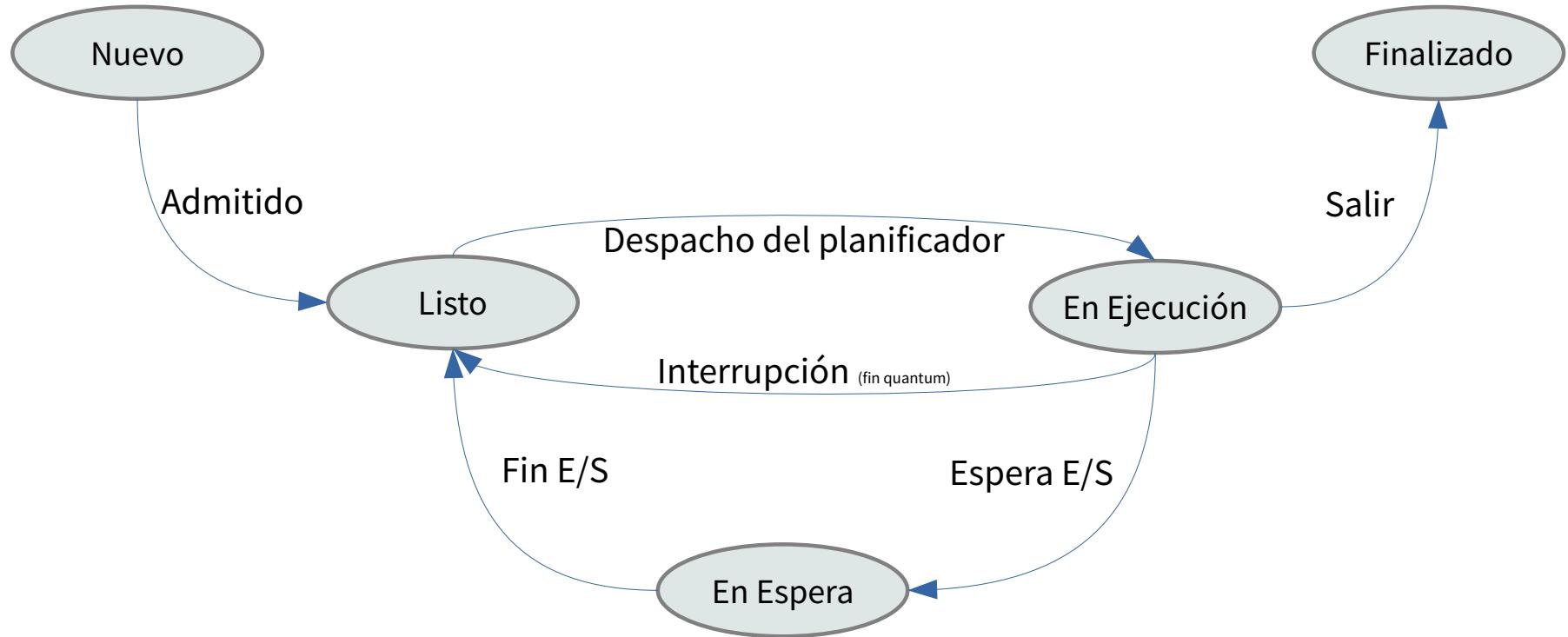
# Planificación de CPU

Colas. Planificadores.

Políticas de Planificación con Desalojo y sin Desalojo.

Algoritmos de Planificación. Planificación con Prioridades.

# Estados de los Procesos



# Introducción



La **planificación del procesador** es una función **fundamental** del Sistema Operativo.

Siendo el procesador uno de los principales recursos, la planificación de su uso es muy importante. Se debe buscar **maximizar la utilización** del procesador.

En un ambiente de **multiprogramación**, el Sistema Operativo debe decidir a qué proceso le dá el procesador para que lo ejecute entre los procesos de la **cola de procesos listos**.

# Colas



El uso compartido de recursos exige una organización en **colas**, de las solicitudes pendientes y de las que llegan.

Existen colas para el uso de la CPU, como también para el uso de los dispositivos de E/S (cola del dispositivo o device queues).

La cola de **procesos listos** (ready queue) es aquella formada por los procesos preparados para la ejecución.

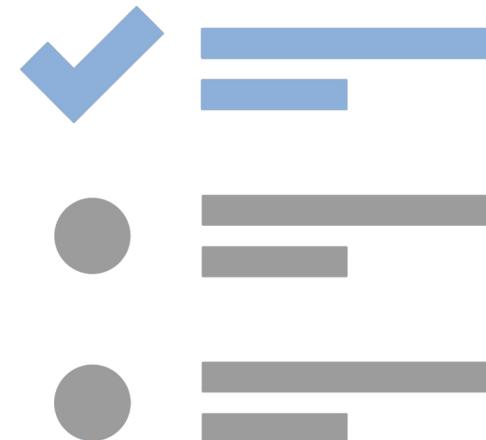
# Planificadores o Schudelers

Para la planificación existen **módulos del Sistema Operativo** que eligen un proceso desde las distintas colas.

Estos módulos tienen una **alta prioridad** de ejecución ya que también necesitan del procesador para ejecutarse.

Hay distintos planificadores que participan en distintos momentos cuyo nombre proviene de la **frecuencia de ejecución** y son:

- Largo plazo
- Medio plazo
- Corto plazo

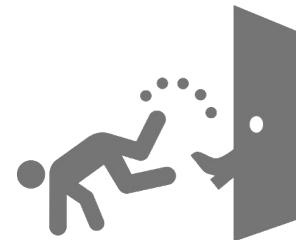


# Políticas de Planificación

**Políticas sin desalojo:** un proceso no deja el procesador desde que se dá la orden de creación hasta que se termine o pasa a estado de espera.



**Políticas con desalojo:** el proceso que se está ejecutando puede ser suspendido temporalmente para darle paso a otro proceso. Hay un temporizador.



# Objetivos

- Justicia
- Máxima capacidad de ejecución
- Máximo número de usuarios interactivos
- Predecibilidad
- Minimización de la sobrecarga
- Equilibrio en el uso de los recursos
- Seguridad de las prioridades



# Medidas de Tratamiento

**Tiempo de servicio o retorno (T):  $T = t_f - t_i$**

$t_i$  = instante en que se dá la orden de creación del proceso.

$t_f$  = instante en que el proceso termina su ejecución.

**Tiempo de espera (E):  $E = T - t$**

$t$  = tiempo de ejecución (tiempo que necesita para realizar su trabajo)

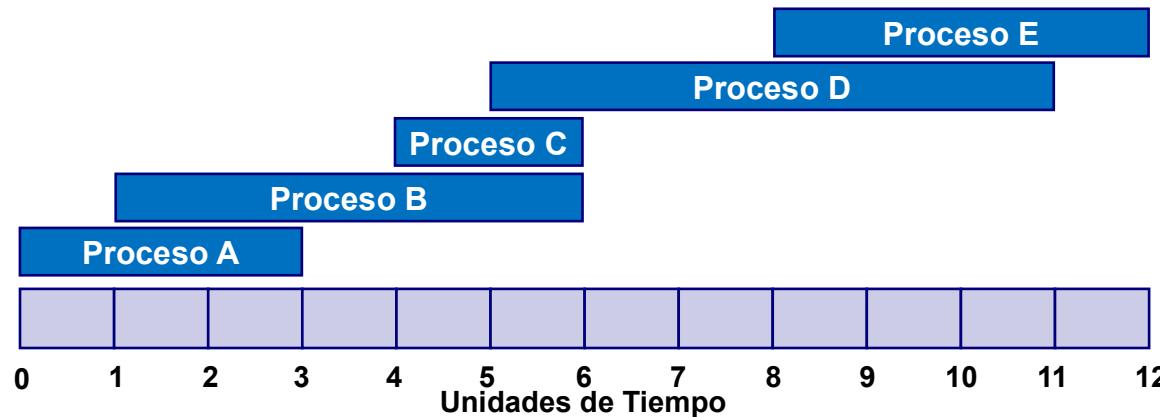
**Indice de servicio (I) :  $I = t / T \quad 0 < I < 1$**

Para más de un proceso se usan valores medios que reflejarán el verdadero comportamiento del sistema y son:

- Tiempo medio de servicio o retorno
- Tiempo medio de espera
- Eficiencia (índice medio de servicio)

# Algoritmos de Planificación

Nombre Proceso	Instante Llegada	Tiempo Ejecución	Prioridad
A	0	3	0
B	1	5	1
C	4	2	0
D	5	6	2
E	8	4	1



# FCFS – First Come, First Served

También llamado ***Primero en Llegar, Primero en Ser Atendido.***

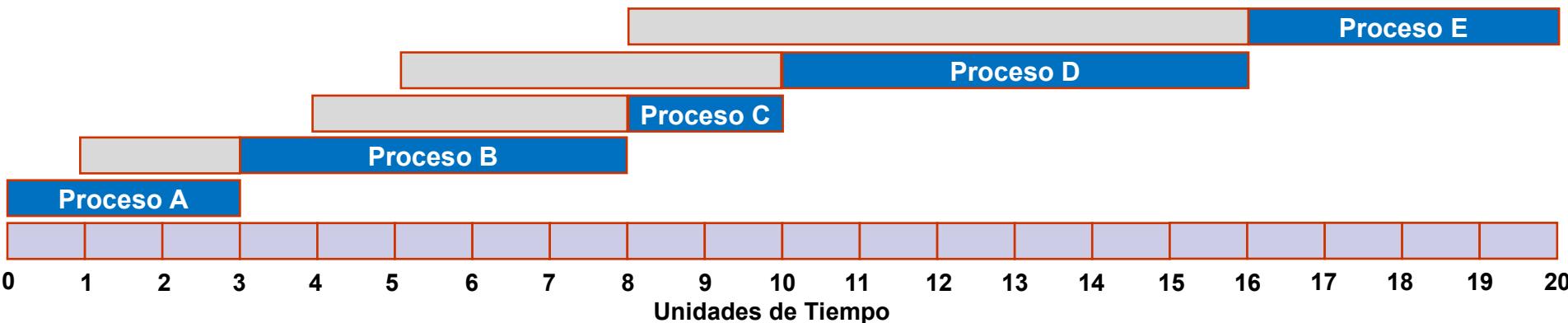
El procesador atiende a un proceso hasta que termina su ejecución y luego atiende al que sigue en la cola.

**Pregunta del Planificador:** ¿Cuál es el proceso más antiguo en la cola de procesos listos?

Características

- Es justa, atiende a todos los procesos.
- Es una política predecible.
- Los procesos largos hacen esperar a los cortos.

Nombre Proceso	Instante Llegada	Tiempo Ejecución	Instante Finalización	$T (t_f - t_i)$	$E (T - t)$	$I (t/T)$
A	0	3	3	3	0	1,00
B	1	5	8	7	2	0,71
C	4	2	10	6	4	0,33
D	5	6	16	11	5	0,54
E	8	4	20	12	8	0,33
<b>Medias</b>				<b>7,8</b>	<b>3,8</b>	<b>0,58</b>



## SJF – Shortest Job First

También llamado **Próximo o Primer proceso, el trabajo más corto.**

Toma de la cola de procesos listos el que necesite menos tiempo de ejecución para realizar su trabajo.

El tiempo de ejecución se puede saber por el propio programa, por usuario o ejecución anterior.

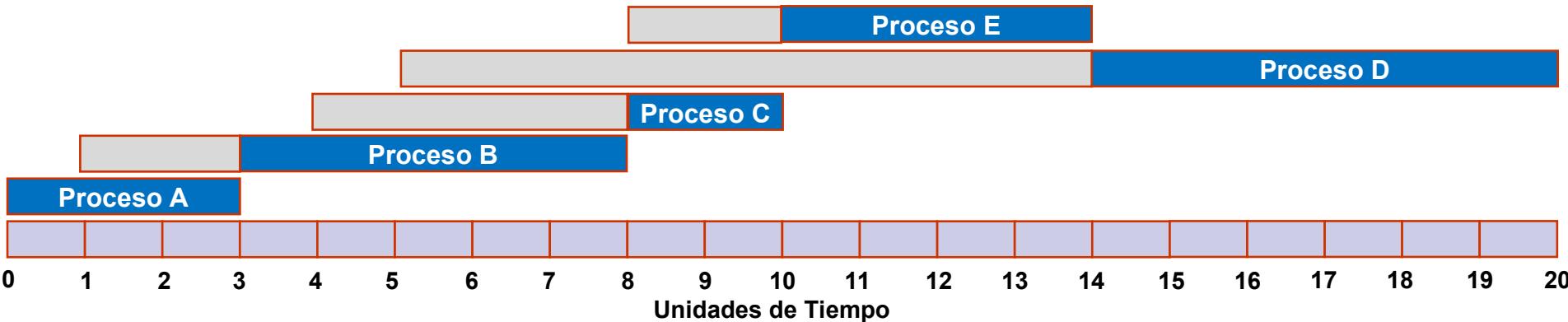
Para igual duración de CPU se emplea planificación FCFS.

**Pregunta del Planificador:** ¿Cuál es el proceso de la cola de procesos listos con menor tiempo de ejecución?

Características

- Es injusta con procesos largos.
- Es poco predecible.
- Difícil puesta en práctica por la información que necesita.

Nombre Proceso	Instante Llegada	Tiempo Ejecución	Instante Finalización	T ( $t_f - t_i$ )	E (T - t)	I ( $t/T$ )
A	0	3	3	3	0	1,00
B	1	5	8	7	2	0,71
C	4	2	10	6	4	0,33
D	5	6	20	15	9	0,40
E	8	4	14	6	2	0,67
<b>Medias</b>				<b>7,4</b>	<b>3,4</b>	<b>0,62</b>



# RR - Round Robin

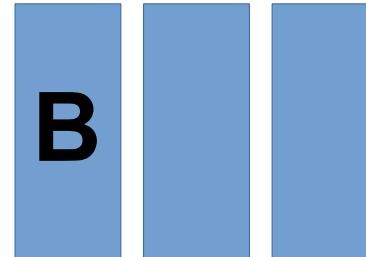
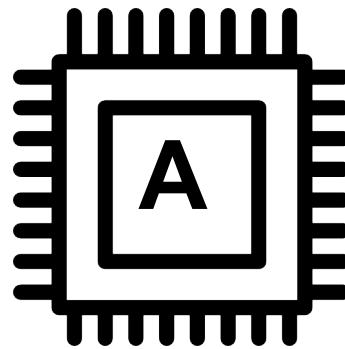
También llamado ***Asignación acíclica o Planificación en rueda.***

Consiste en conceder a cada proceso en ejecución un determinado período de tiempo  $q$  (quantum), luego del cual si el proceso no terminó se lo devuelve al final de la cola de procesos listos.

## Características

- Con cambios de contextos eficientes presenta baja sobrecarga.
- Más utilizada en SO de tiempo compartido.
- El tamaño del quantum depende de:
  - Tipo de sistema.
  - Las cargas que soporte el mismo.
  - El número de procesos en el sistema.

## RR - Round Robin > Funcionamiento básico

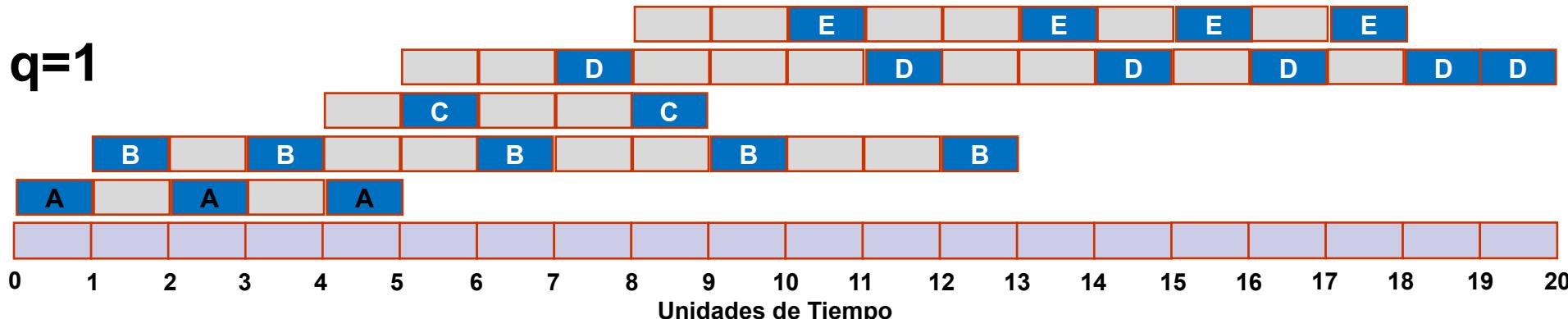


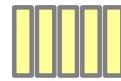
C

- 1º) se agregan a la cola los procesos nuevos.
- 2º) el proceso que se estaba ejecutando pasa al final de la cola.
- 3º) el proceso que quedó primero en la cola pasa a ejecutarse.

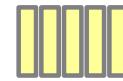
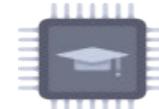
Nombre Proceso	Instante Llegada	Tiempo Ejecución	Instante Finalización	T ( $t_f - t_i$ )	E (T - t)	I ( $t/T$ )
A	0	3	5	5	2	0,60
B	1	5	13	12	7	0,42
C	4	2	9	5	3	0,40
D	5	6	20	15	9	0,40
E	8	4	18	10	6	0,40
<b>Medias</b>				<b>9,4</b>	<b>5,4</b>	<b>0,44</b>

**q=1**





0)	A	
1)	B	A
2)	A	B
3)	B	A
<b>4)</b>	<b>A</b>	CB
5)	C	BD
6)	B	DC
7)	D	CB
<b>8)</b>	<b>C</b>	BED
9)	B	ED



10)	E	DB
11)	D	BE
<b>12)</b>	<b><u>B</u></b>	ED
13)	E	D
14)	D	E
15)	E	D
16)	D	E
<b>17)</b>	<b><u>E</u></b>	D
18)	D	
<b>19)</b>	<b><u>D</u></b>	

q=1

A A A  
B B B B B  
C C  
D D D D D D  
E E E E

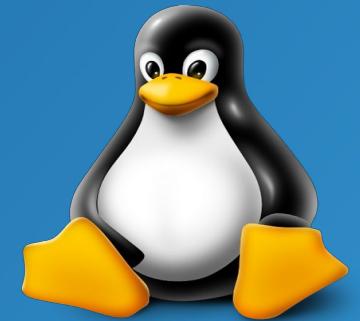


Facultad de Ciencias de la Administración

# Administración de Procesos

Administración de Procesos en Sistemas Operativos Linux.

Administración de Procesos en Sistemas Operativos Windows.



# Administración de Procesos en Linux

Consultar información de Procesos. Jerarquía de Procesos. Finalizar ejecución de Procesos. Establecer Prioridades. Procesos en Background.

# Comando ps

Muestra un listado con los procesos del sistema

```
$ ps
```

```
sisoper@gabsisoper1:~$ ps
  PID  TTY      TIME CMD
 3952 pts/1    00:00:00 bash
 4055 pts/1    00:00:00 ps
sisoper@gabsisoper1:~$ █
```

```
$ ps -aux
```

-a muestra los procesos de todos los usuarios

-x muestra los procesos de todas las terminales

-u muestra la información en forma más amigable

# Comando ps

```
sisoper@gabsisoper1:~$ ps -aux
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.4  0.4  33832  4352 ?        Ss   18:24  0:00 /sbin/init
root         2  0.0  0.0     0     0 ?        S    18:24  0:00 [kthreadd]
root         3  0.0  0.0     0     0 ?        S    18:24  0:00 [ksoftirqd/0]
root         4  0.0  0.0     0     0 ?        S    18:24  0:00 [kworker/0:0]
root         5  0.0  0.0     0     0 ?        S<   18:24  0:00 [kworker/0:0H]
root         6  0.0  0.0     0     0 ?        S    18:24  0:00 [kworker/u2:0]
root         7  0.2  0.0     0     0 ?        S    18:24  0:00 [rcu_sched]
root         8  0.0  0.0     0     0 ?        S    18:24  0:00 [rcuos/0]
root         9  0.0  0.0     0     0 ?        S    18:24  0:00 [rcu_bh]
root        10  0.0  0.0     0     0 ?        S    18:24  0:00 [rcuob/0]
root        11  0.0  0.0     0     0 ?        S    18:24  0:00 [migration/0]
root        12  0.0  0.0     0     0 ?        S    18:24  0:00 [watchdog/0]
root        13  0.0  0.0     0     0 ?        S<   18:24  0:00 [khelper]
root        14  0.0  0.0     0     0 ?        S    18:24  0:00 [kdevtmpfs]
root        15  0.0  0.0     0     0 ?        S<   18:24  0:00 [netns]
root        16  0.0  0.0     0     0 ?        S    18:24  0:00 [khungtaskd]
```

## Comando ps

<b>USER</b>	Nombre de usuario que ejecuta el programa.
<b>PID</b>	Número encargado de identificar el proceso.
<b>%CPU</b>	Porcentaje total de tiempo de CPU que utiliza el proceso.
<b>%MEM</b>	Porcentaje de memoria utilizada por el proceso.
<b>VSZ</b>	Tamaño total del proceso en la memoria virtual, en KB.
<b>RSS</b>	Cantidad de memoria utilizada por el proceso.
<b>TTY</b>	Terminal en la que se ejecuta el proceso.
<b>STAT</b>	Estado del proceso.
<b>START</b>	Indica la hora de ejecución.
<b>TIME</b>	Cantidad total de tiempo de CPU consumido.
<b>COMMAND</b>	Comando utilizado para iniciar el proceso.

## Comando ps

Averiguar información específica de un proceso.

```
$ ps -ax | grep comando
```

```
sisoper@gabsisoper1:~$ ps -ax | grep firefox
 4057 ?          Sl      0:03 /usr/lib/firefox/firefox
 4123 pts/1      S+      0:00 grep --color=auto firefox
sisoper@gabsisoper1:~$ █
```

Averiguar el PID de un proceso

```
$ pgrep comando
```

```
sisoper@gabsisoper1:~$ pgrep firefox
4057
sisoper@gabsisoper1:~$ █
```

# Comando top

top - 18:28:26 up 3 min, 2 users, load average: 0,10, 0,28, 0,14												
Tareas: 172 total, 3 ejecutar, 169 hibernar, 0 detener, 0 zombie												
%Cpu(s): 1,0 usuario, 0,7 sist, 0,0 adecuado, 98,3 inact, 0,0 en espera, 0,												
KiB Mem: 1017576 total, 877448 used, 140128 free, 30984 buffers												
KiB Swap: 1046524 total, 0 used, 1046524 free. 306804 cached Mem												
PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN	
2072	sisoper	20	0	1247700	178332	64176	S	1,0	17,5	0:13.70	compiz	
<b>1109</b>	<b>root</b>	<b>20</b>	<b>0</b>	<b>291588</b>	<b>62900</b>	<b>21896</b>	<b>R</b>	<b>0,7</b>	<b>6,2</b>	<b>0:04.34</b>	<b>Xorg</b>	
1022	root	20	0	232756	1848	1520	S	0,3	0,2	0:00.08	VBoxService	
2099	sisoper	20	0	895096	41096	32352	S	0,3	4,0	0:00.44	nautilus	
2320	sisoper	20	0	578492	29920	23336	S	0,3	2,9	0:00.77	gnome-terminal	
1	root	20	0	33832	4352	2680	S	0,0	0,4	0:00.67	init	
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd	
3	root	20	0	0	0	0	S	0,0	0,0	0:00.01	ksoftirqd/0	
4	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0	
5	root	0	-20	0	0	0	S	0,0	0,0	0:00.00	kworker/0:0H	
6	root	20	0	0	0	0	S	0,0	0,0	0:00.07	kworker/u2:0	
7	root	20	0	0	0	0	S	0,0	0,0	0:00.38	rcu_sched	
<b>8</b>	<b>root</b>	<b>20</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>R</b>	<b>0,0</b>	<b>0,0</b>	<b>0:00.08</b>	<b>rcuos/0</b>	

# Comando htop

Es necesario instalarlo (\$ sudo apt install htop)

CPU [ 2.1% ] Tasks: 129, 181 thr; 2 running											
Mem [ 548/993MB ] Load average: 0.44 0.28 0.16											
Swp [ 6/1021MB ] Uptime: 00:06:17											
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
3063	sisoper	20	0	30044	3684	3020	R	1.4	0.4	0:00.10	htop
2072	sisoper	20	0	1218M	127M	16092	S	1.4	12.8	0:19.53	compiz
1109	root	20	0	290M	56536	14364	S	0.7	5.6	0:06.33	/usr/bin/X -core
2650	root	20	0	33796	4128	3824	S	0.7	0.4	0:00.20	/usr/lib/apt/meth
2470	sisoper	20	0	558M	11148	5472	S	0.7	1.1	0:00.22	update-notifier
2465	sisoper	20	0	558M	11148	5472	S	0.7	1.1	0:00.28	update-notifier
2320	sisoper	20	0	564M	14968	8380	S	0.7	1.5	0:01.32	gnome-terminal
2272	sisoper	20	0	448M	8092	4420	S	0.0	0.8	0:00.03	zeitgeist-databub
2099	sisoper	20	0	874M	20092	11068	S	0.0	2.0	0:00.53	nautilus -n
1022	root	20	0	227M	1356	1028	S	0.0	0.1	0:00.14	/usr/sbin/VBoxSer
2656	root	20	0	37292	7628	3760	S	0.0	0.7	0:00.57	/usr/lib/apt/meth
2645	root	20	0	36656	5128	4380	S	0.0	0.5	0:00.04	apt-get -qq -y up
1814	sisoper	20	0	353M	4496	3212	S	0.0	0.4	0:00.30	/usr/bin/ibus-dae
1	root	20	0	33832	3212	2404	S	0.0	0.3	0:00.67	/sbin/init
255	root	20	0	19480	1428	1428	S	0.0	0.1	0:00.05	upstart-udev-brid
261	root	20	0	51920	2008	1944	S	0.0	0.2	0:00.14	/lib/systemd/syst
357	root	20	0	15280	4	0	S	0.0	0.0	0:00.02	upstart-file-brid

# Comando tree – Árbol de Procesos

Muestra los procesos en forma de árbol, según la relación padre-hijo.

```
$ pstree
```

```
sisoper@sisoper-VirtualBox:~$ pstree -AGU
systemd--ModemManager---2*[{ModemManager}]
          NetworkManager---dhclient
                      ---2*[{NetworkManager}]
          accounts-daemon---2*[{accounts-daemon}]
          acpid
          anacron
          avahi-daemon---avahi-daemon
          boltdd---2*[{boltdd}]
          colord---2*[{colord}]
          cron
          cups-browsed---2*[{cups-browsed}]
          cupsd---3*[dbus]
          dbus-daemon
          fwupd---4*[{fwupd}]
          gdm3---gdm-session-wor---gdm-x-session---Xorg---{Xorg}
                                         ---gnome-session-b
```

# Comando kill – Matando Procesos

Le pedimos al proceso que finalice su ejecución (señal TERM).

```
$ kill PID
```

Si el proceso no obedece, le enviamos la señal KILL al Kernel para que lo finalice.

```
$ kill -SIGKILL PID
```

o

```
$ kill -9 PID
```

Para finalizar un proceso por su nombre:

```
$ killall firefox
```

Finaliza todos los procesos de firefox.



Las prioridades en Linux se manejan con un valor entre -20 (prioridad más alta) a 19 (prioridad más baja). 0 es la prioridad por defecto.

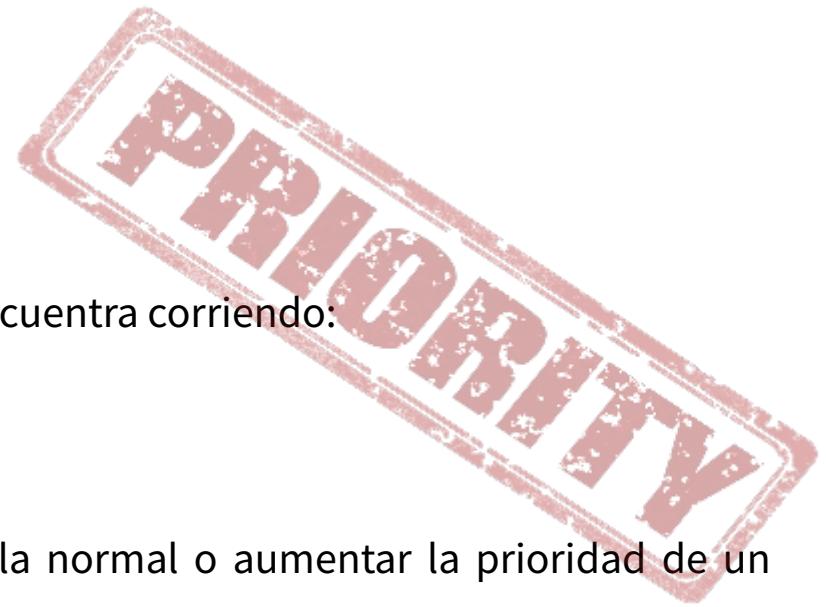
Para correr un proceso con una prioridad 15:

```
$ nice -n 15 comando
```

Si queremos modificar la prioridad de un proceso que ya se encuentra corriendo:

```
$ renice prioridad PID
```

Solo root puede correr un proceso con mayor prioridad de la normal o aumentar la prioridad de un proceso.



# Procesos en Background Plano - Jobs

Para iniciar un proceso en background se utiliza el símbolo &

```
$ comando &
```

```
sisoper@gabsisoper1:~$ gedit &
[1] 4184
sisoper@gabsisoper1:~$
```

Para ver la lista de procesos corriendo en background

```
$ jobs
```

```
sisoper@gabsisoper1:~$ jobs
[1]+  Ejecutando                  gedit &
sisoper@gabsisoper1:~$
```

Para traer un proceso al frente

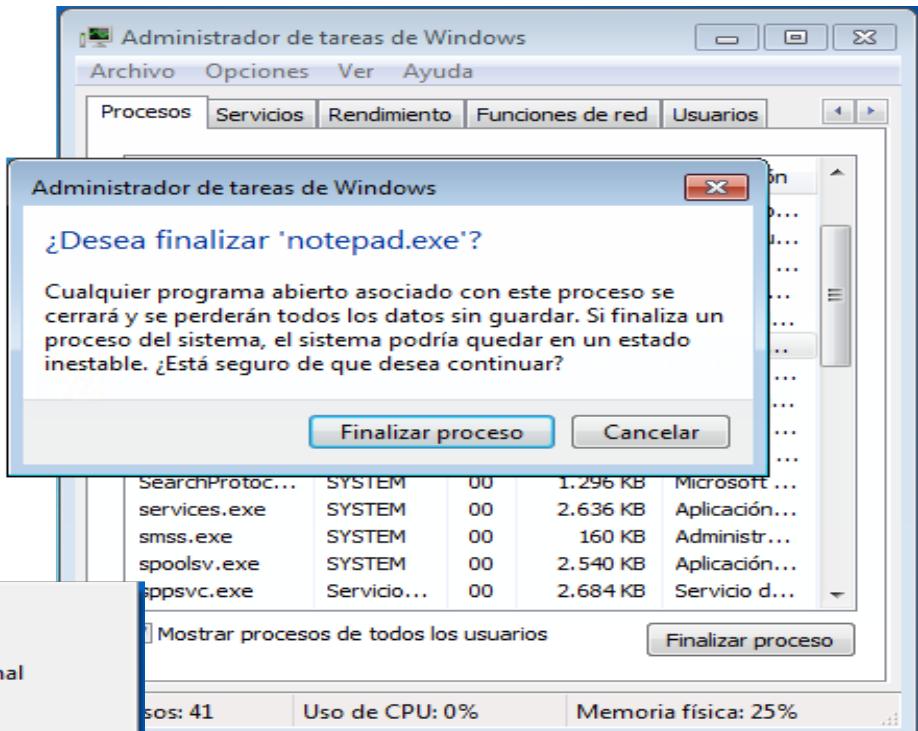
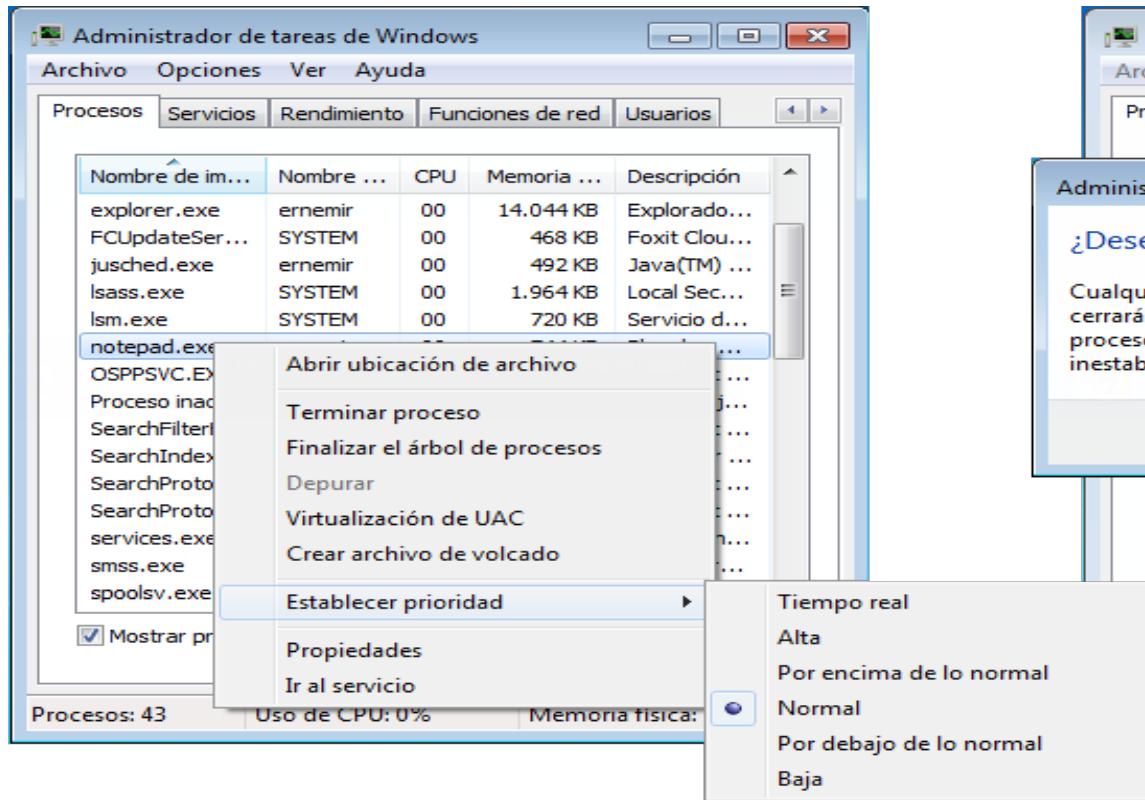
```
$ fg %nro_job
```



# Administración de Procesos en Windows

Consultar Información de Procesos por GUI y CLI – Finalizar ejecución de Procesos

# Administrador de Tareas



# Comando TASKLIST

Muestra una lista de procesos en ejecución

Nombre de imagen	PID	Nombre de sesión	Núm. de ses	Uso de memor
System Idle Process	0	Services	0	12 KB
System	4	Services	0	840 KB
smss.exe	268	Services	0	684 KB
csrss.exe	348	Services	0	2.656 KB
wininit.exe	396	Services	0	2.960 KB
csrss.exe	404		1	6.984 KB
winlogon.exe	444		1	4.204 KB
services.exe	488	Services	0	5.940 KB
lsass.exe	496	Services	0	7.564 KB
lsm.exe	504	Services	0	2.944 KB
suchohost.exe	612	Services	0	6.156 KB
spooler	656	Spooler	0	2.856 KB

Para obtener información más detallada podemos usar:

> tasklist /v

# Comando TASKKILL

Para matar un proceso podemos usar:

```
> taskkill /PID PID  
> taskkill /IM nombre_proceso
```

```
C:\Users\sisoper>taskkill /PID 1612  
Correcto: se envío la señal de término al proceso con PID 1612.  
  
C:\Users\sisoper>taskkill /IM notepad.exe  
ERROR: no se pudo terminar el proceso "notepad.exe" con PID 3372.  
Motivo: Acceso denegado.  
CORRECTO: señal de terminación enviada al proceso "notepad.exe" con PID 552.
```

```
C:\Users\sisoper>_
```

\* PID 3372 ejecutado como Administrador

Para matar un proceso de forma forzada:

```
> taskkill /F /PID PID  
> taskkill /F /IM nombre_proceso
```

# Administración de Procesos con PowerShell > Get-Process

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\usuario> Get-Process

Handles  NPM(K)      PM(K)      WS(K)      CPU(s)      Id  SI ProcessName
-----  -----      -----      -----      -----      --  --  -----
        336      19       6972     26284      0,05     5388  1 ApplicationFrameHost
        178      11       6384     11644      0,09     1596  0 audiodg
        388      18       3764     16212      0,02     5516  1 browser_broker
        110      6        1016     1712          1828  0 CompatTelRunner
        858      29       40168    49528          4648  0 CompatTelRunner
        267      14       3364     14156          5044  0 CompatTelRunner
        154      10       6500      1576          1928  0 conhost
        265      14       4088     17716      0,09     6564  1 conhost
        429      16       1572      5132          420   0 csrss
        353      17       1660      4792          496   1 csrss
        449      17       4380     20740      0,14     3152  1 ctfmon
        376      18       3752     12616          1936  0 dasHost
        224      22       4732     12496      0,08     5640  1 dllhost
```

## Administración de Procesos con PowerShell > Get-Process

```
PS C:\Users\usuario> Get-Process note*
Handles  NPM(K)      PM(K)      WS(K)      CPU(s)      Id  SI ProcessName
-----  -----      -----      -----      -----  --  --  -----
        228          13        2500      15084      0,02    3224    1 notepad
```

```
PS C:\Users\usuario> (Get-Process note*).CPU
0,015625
```

```
PS C:\Users\usuario> (Get-Process note*).WS
15446016
```

```
PS C:\Users\usuario> ■
```

## Administración de Procesos con PowerShell > Stop-Process

```
PS C:\Users\usuario> Stop-Process -id 3224 -confirm  
  
Confirmar  
¿Está seguro de que desea realizar esta acción?  
Se está realizando la operación "Stop-Process" en el destino "notepad (3224)".  
[S] Sí [O] Sí a todo [N] No [T] No a todo [U] Suspender [?] Ayuda (el valor predeterminado es "S"): S  
PS C:\Users\usuario> Get-Process note*  
PS C:\Users\usuario> ■
```

```
> Get-Process | where-object {$_ .Responding -eq $false} | Stop-Process
```



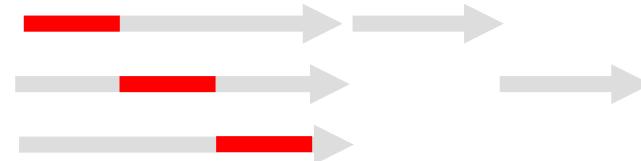
Facultad de Ciencias de la Administración

# Procesos

Procesos Concurrentes. Paralelismo. Procesos independientes y cooperativos.  
Comunicación entre procesos. Hilos: concepto, objetivos e implementación.

# Concurrencia y Paralelismo

**Concurrencia:** es la “**convivencia**” de varios procesos en un mismo instante de tiempo.



**Paralelismo:** es la capacidad de un sistema de **ejecutar** más de un proceso al mismo tiempo.



Paralelismo => Concurrencia: si varios procesos se ejecutan a la vez, implica que conviven.

Concurrencia  $\neq$  Paralelismo: si varios procesos conviven, no implica que se ejecutan a la vez.

# Procesos Independientes y Cooperativos

Los procesos que se ejecutan **concurrentemente** pueden ser:

- **Procesos Independientes:** no afectan o son afectados por la ejecución de otros procesos del sistema.
- **Procesos Cooperativos:** afectan o son afectados por la ejecución de otros procesos del sistema. Se comunican entre ellos a los efectos de intercambiar información. Comparten un espacio de memoria, utilizan un archivo o intercambian mensajes.



# Razones para Cooperar

- Compartir información
- Acelerar cálculos: (dividir una tarea en subtareas y ejecutarlas en paralelo)
- Modularidad: diferentes procesos se encargan de diferentes funcionalidades.
- Sincronización
- Conveniencia

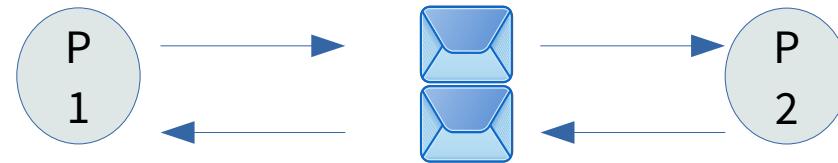
# Comunicación entre Procesos

- Memoria Compartida

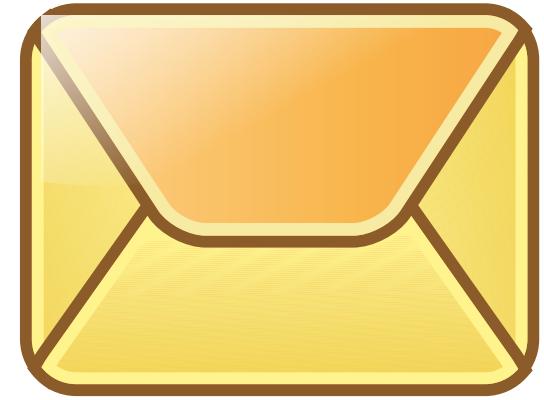
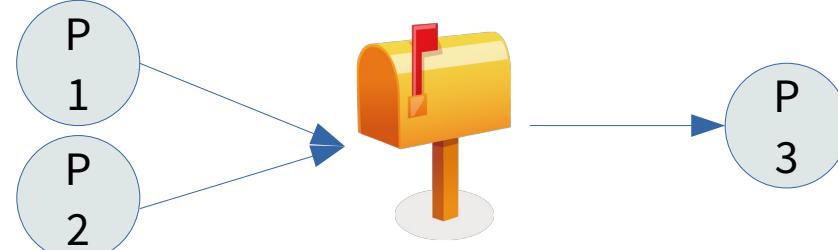


# Comunicación entre Procesos

- Intercambio de Mensajes
  - Directo



- Indirecto

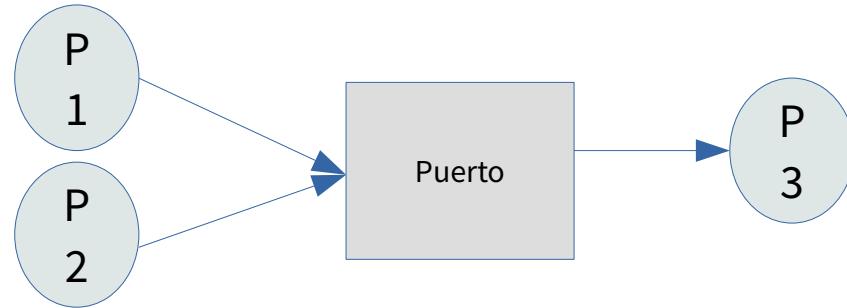


`send(destino, mensaje)`

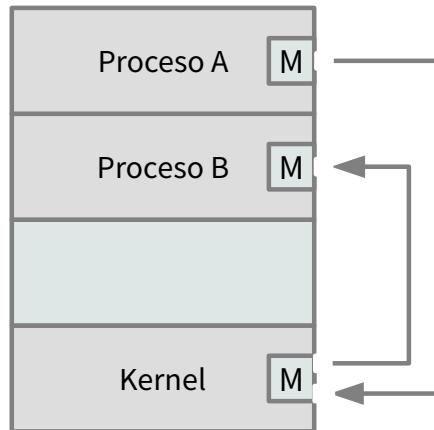
`receive(origen, mensaje)`

# Comunicación entre Procesos

- Puertos



# Comunicación entre Procesos



Intercambio de  
Mensajes



Memoria  
Compartida

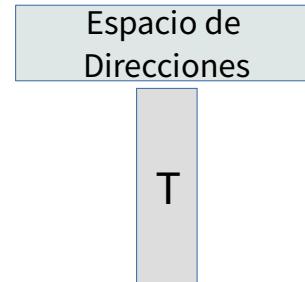
# Hilos (Threads)



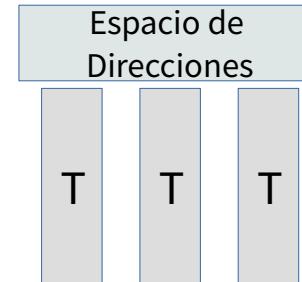
Un **hilo** es un flujo de ejecución que comparte la imagen de memoria y otras informaciones con otros hilos.

Un proceso puede contener un único flujo de ejecución (clásico) o más de un flujo de ejecución.

Función cuya ejecución se puede lanzar en paralelo con otras. Permite a los usuarios realizar múltiples tareas en forma concurrente.



Proceso Mono Hilo



Proceso Multi Hilo

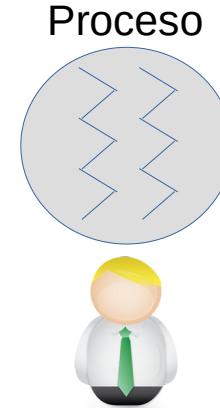
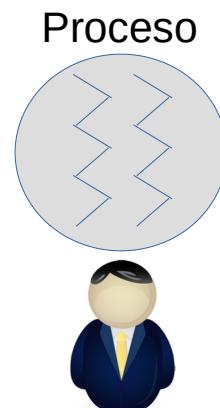
# Hilos (Threads)



Los hilos de un mismo proceso comparten el espacio de direcciones, las variables globales y conjuntos de recursos del sistema.

Pero tienen su propio contador de programa, sus propios estados de registros y su propia pila.

Entre ellos **no existe protección**.

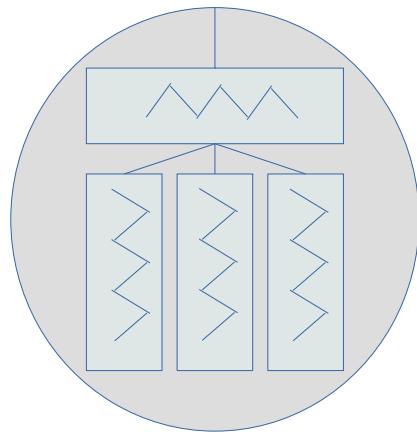


## Hilos (Threads) > Ventajas sobre Procesos

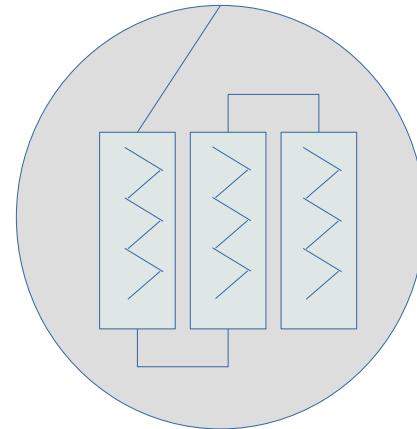
- Menor sobrecarga en la creación.
- Conmutación más eficiente.
- Compartición de recursos más natural y eficiente.
- Paralelismo



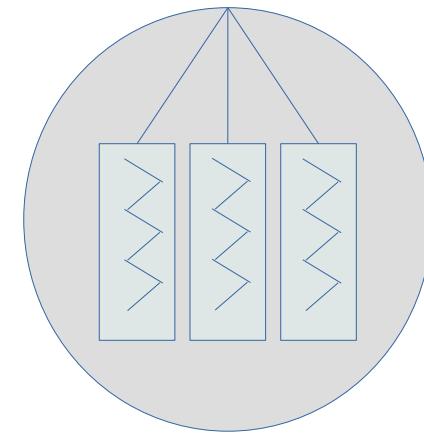
# Hilos (Threads) > Modelos de Organización



Despachador -  
Trabajadores



Pipeline



Equipo

# Hilos (Threads) > Implementación

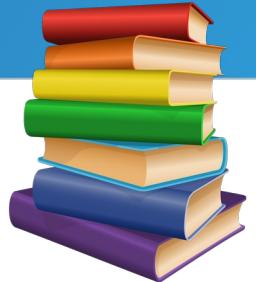
## En espacio de Usuario

Espacio Usuario	Procesos y sus Hilos
	Runtime (información de estado de los Hilos)
Espacio Núcleo	Núcleo (información de estado de los Procesos)

## En espacio de Núcleo

Espacio Usuario	Procesos y sus Hilos
	Núcleo (información de estado de los Procesos e Hilos)
Espacio Núcleo	

- Se puede implementar hilos si no están soportados por el SO.
- Flexibilidad en el uso de planificadores personalizados.
- Permutación más eficiente.
- Se puede emplear planificador por turno circular (interrupción)
- Un hilo puede realizar llamadas al sistema bloqueante.



# Fuentes y Bibliografía

- **SILBERSCHATZ** - *Fundamentos de Sistemas Operativos*
  - Capítulo 3 - Procesos
  - Capítulo 4 – Hebras
- **SILVA** – *Sistemas Operativos*
  - Capítulo 2 - Procesos



Facultad de Ciencias de la Administración



Facultad de Ciencias de la Administración

# Introducción a los Sistemas Operativos

Definición. Evolución. Clasificación según diversos conceptos. Sistemas de red y Sistemas Distribuidos. Sistemas Operativos para dispositivos móviles. Funciones de los Sistemas Operativos. Llamadas al sistema, excepciones y protecciones. Núcleo del sistema operativo.

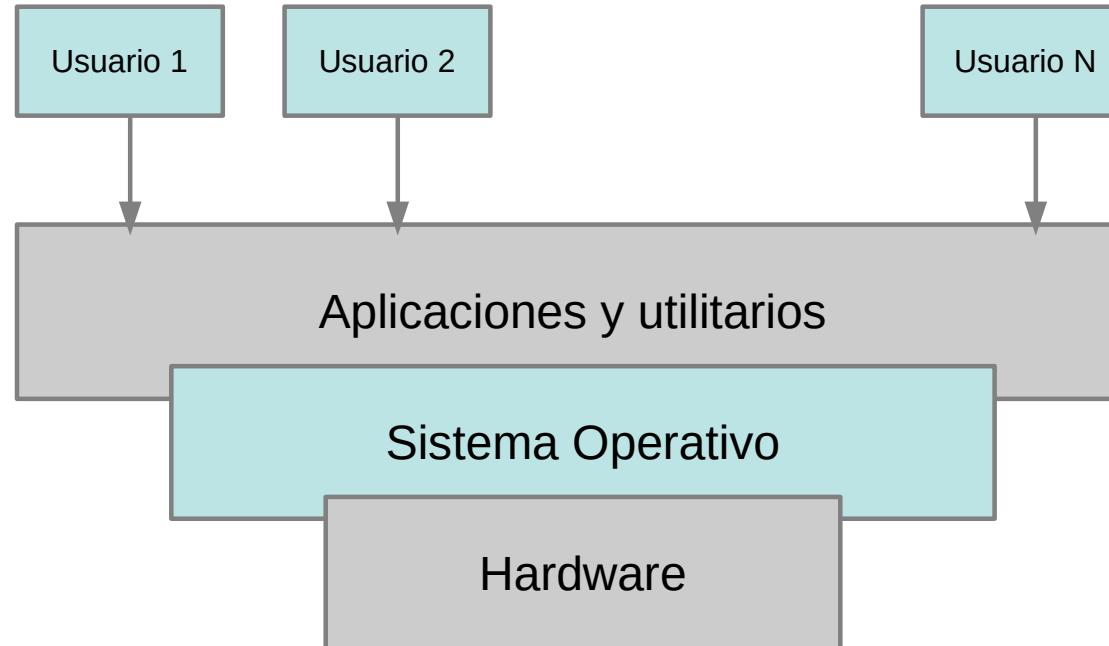
# Definición

*“Es un programa que administra el hardware de la computadora”.*

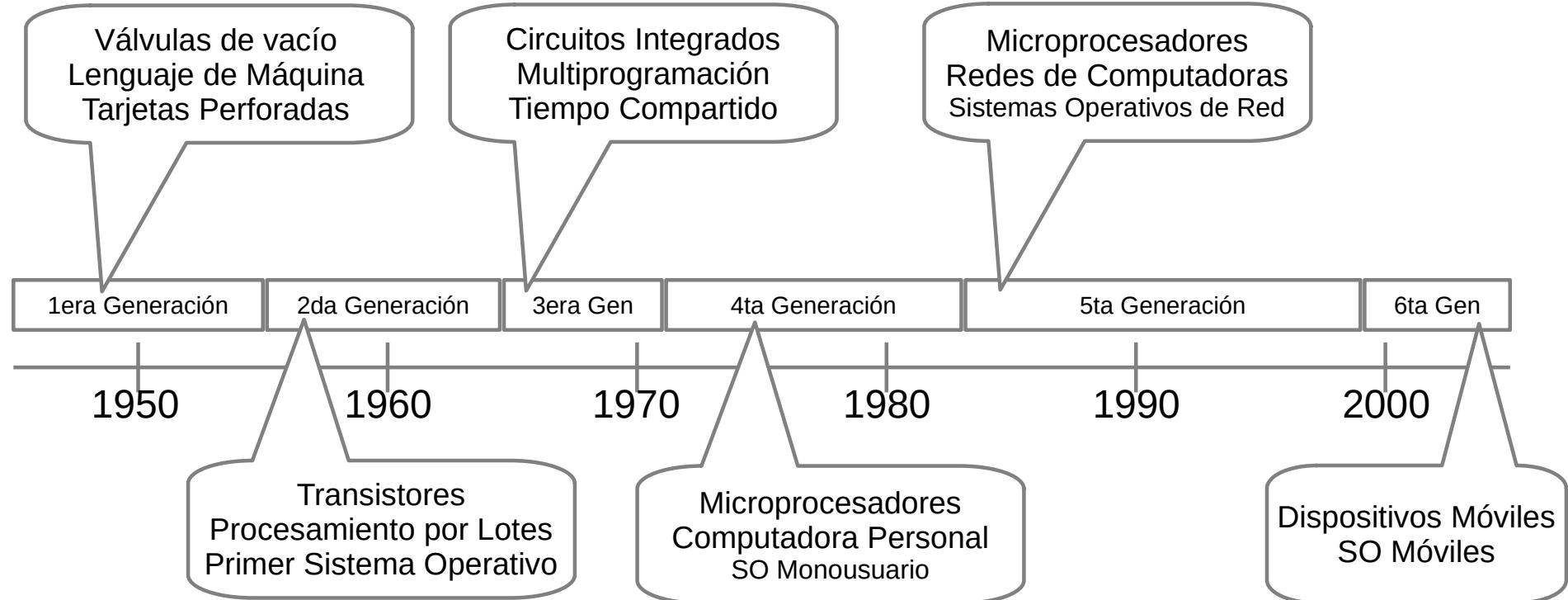
*“Proporciona las bases para los programas de aplicación y actúa como intermediario entre el usuario y el hardware”*



# Sistema Informático

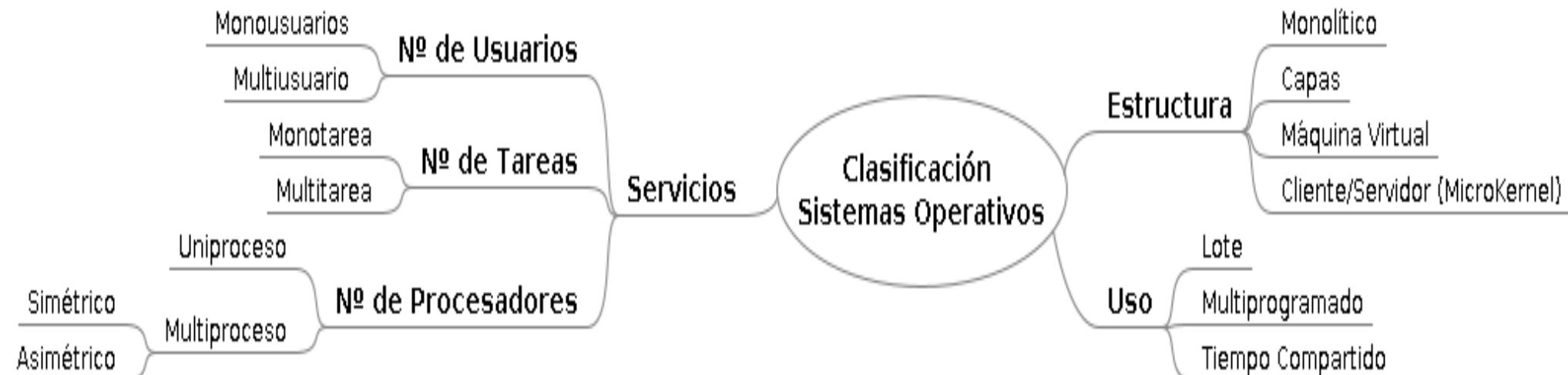


# Evolución

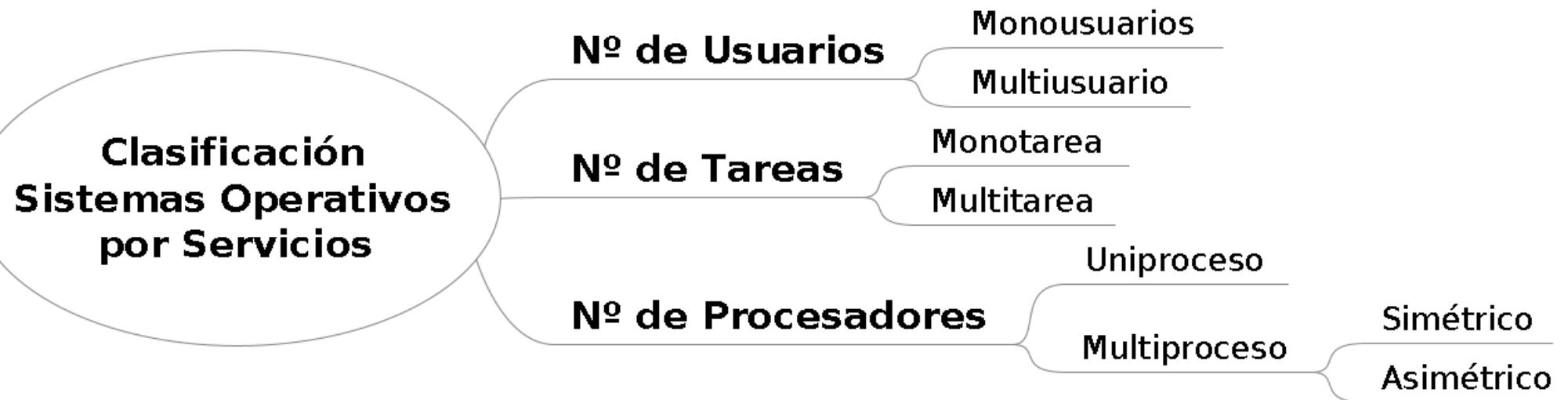


# Clasificación

Existen varias formas de clasificar a los Sistemas Operativos:



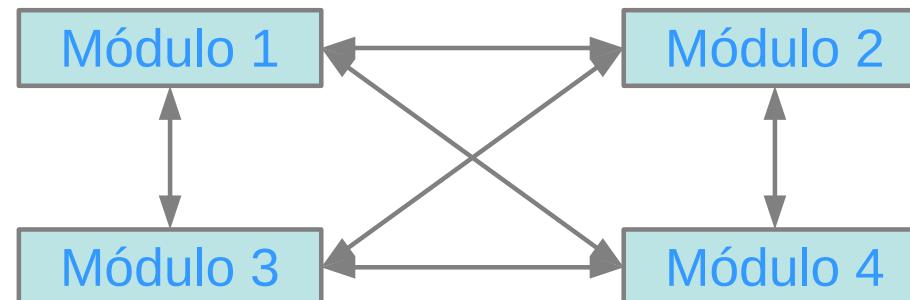
# Clasificación por Servicios



# Clasificación por Estructura

## Monolíticos

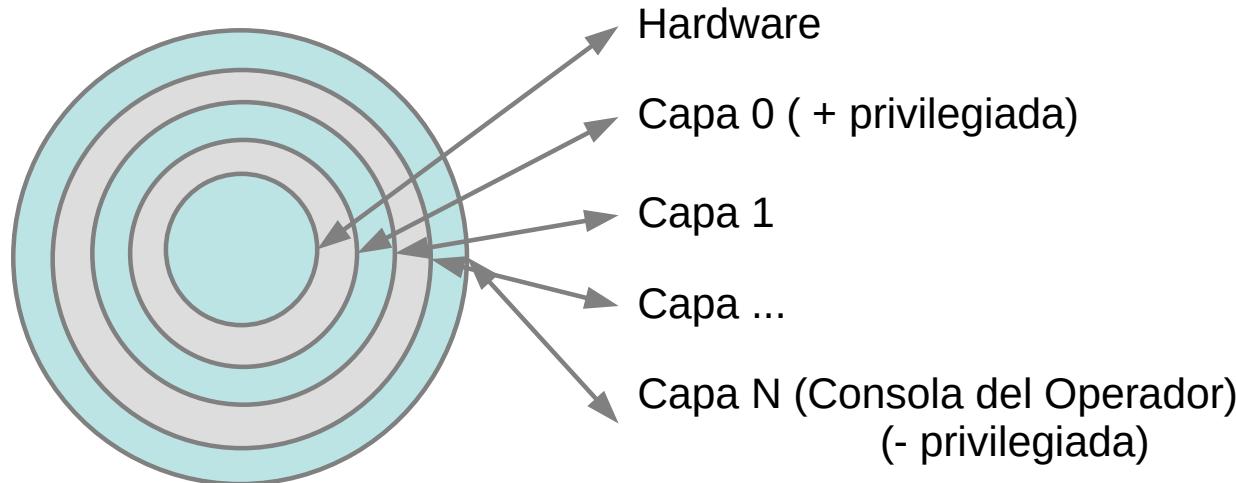
- El Sistema Operativo está constituido por un único programa.
- Ese programa está formado por módulos que se llaman entre si.
- Carece de protecciones y resulta menos flexible.



# Clasificación por Estructura

## Jerárquica o Por Capas

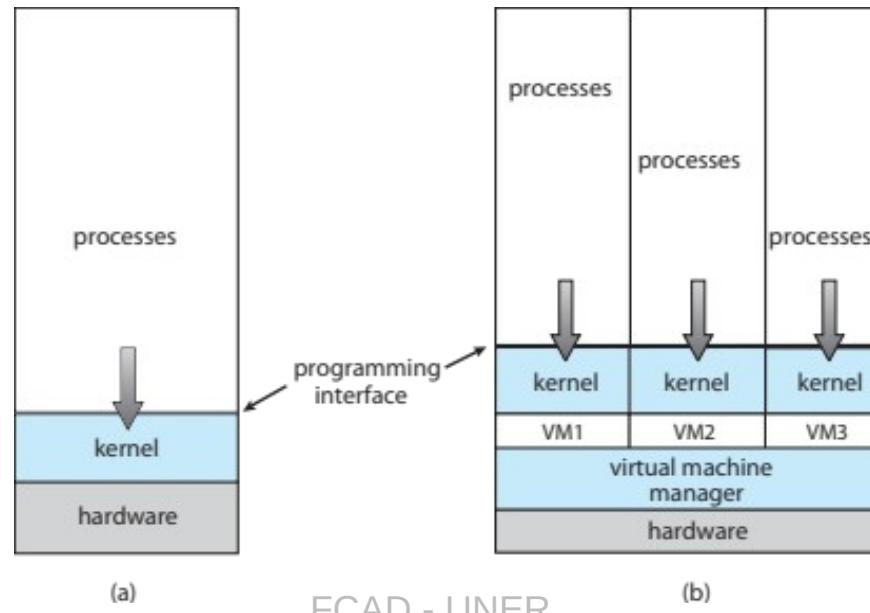
- Se definen capas con funciones bien definidas, y también interfaces entre ellas.



# Clasificación por Estructura

## Máquina Virtual

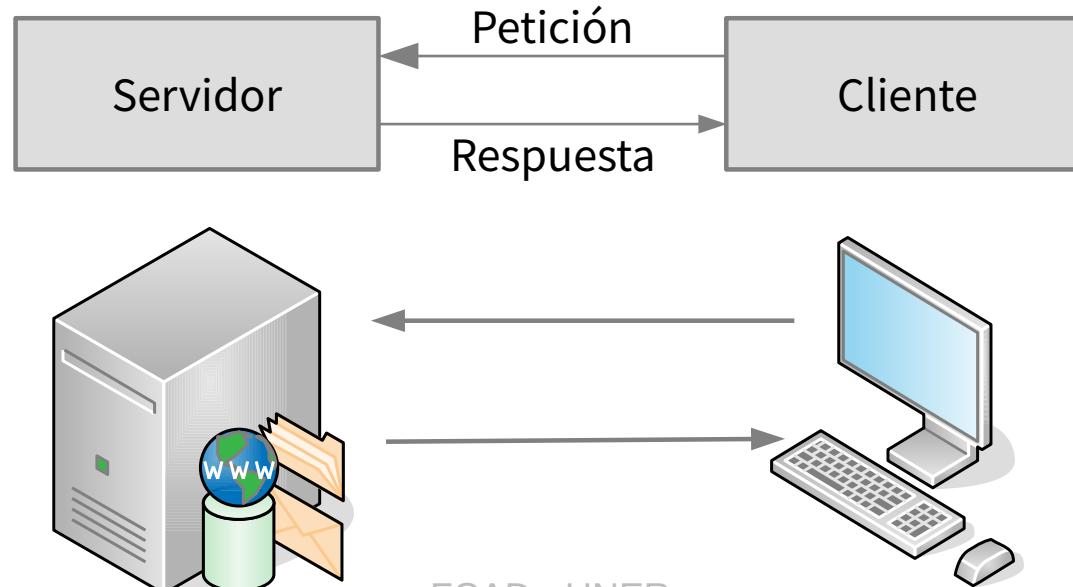
- Ejecutan diferentes Sistemas Operativos sobre una misma computadora, dando la sensación de ser varias.



# Clasificación por Estructura

## Cliente / Servidor

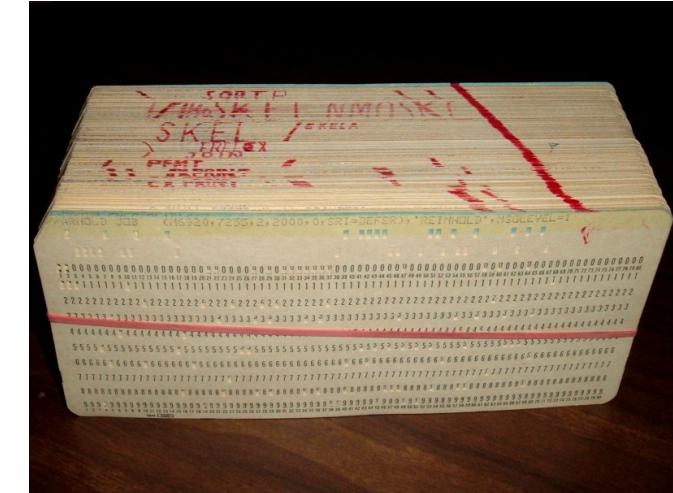
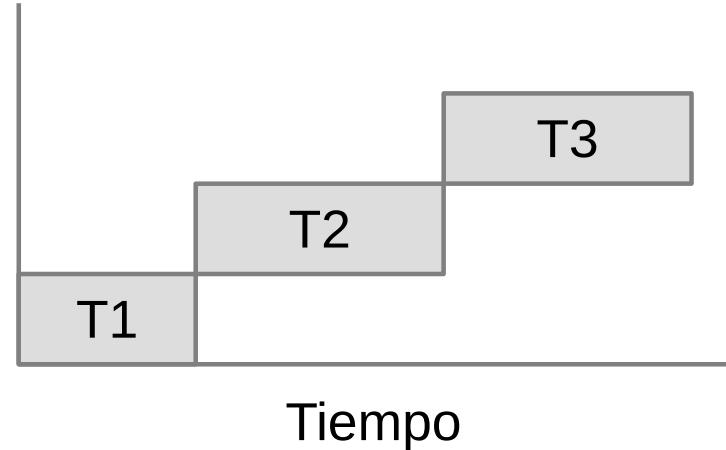
- Los servidores van a responder a las solicitudes de los clientes.



# Clasificación por uso

## Sistemas Operativos por Lote (batch)

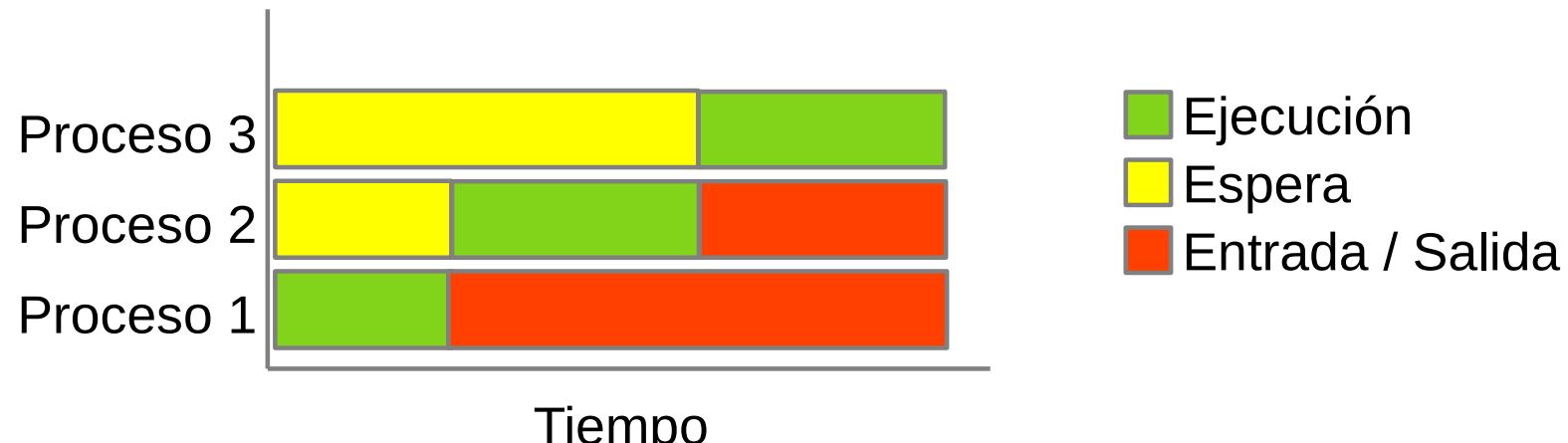
- Los trabajos se ejecutan uno a continuación del otro.
- En las operaciones de E/S la CPU no hace nada.



# Clasificación por Uso

## Sistemas Operativos Multiprogramados

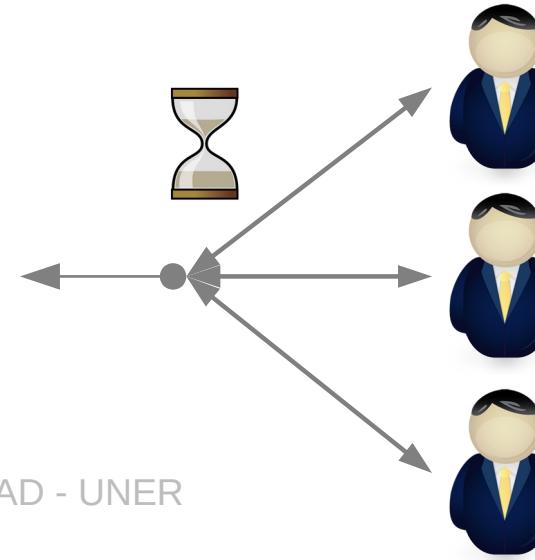
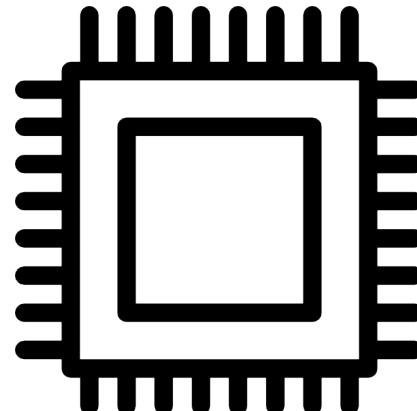
- Se mantiene en memoria varios procesos para ejecutar.
- Si un proceso realiza una E/S, se selecciona otro proceso de los disponibles para ejecutar de acuerdo a su prioridad.



# Clasificación por Uso

## Sistemas Operativos de Tiempo Compartido

- Se divide el uso del procesador por períodos cortos de tiempo (multiprogramación + calendarización).
- Se otorga a cada usuario un quantum.



# Sistemas Operativos en Tiempo Real (RTOS)

Estos sistemas deben cumplir con un tiempo de respuesta predeterminado.

Son utilizados en ambientes industriales, científicos y médicos.

- **Sistema de tiempo real duro:** el tiempo de respuesta no puede variar. Es un sistema crítico.
- **Sistema de tiempo real blando:** pueden existir pequeñas variaciones en los tiempos de respuesta.



# Sistemas Operativos Embebidos

Están integrados en los circuitos de los dispositivos electrónicos (Memoria ROM). Entre estos dispositivos se encuentra electrodomésticos, radios, televisores, automóviles, equipos médicos, PLC, etc.

Suelen tener algunas características de los Sistemas en Tiempo real pero con limitaciones de tamaño, memoria y consumo de electricidad.

Características:

- Fiabilidad y Seguridad
- Eficiencia
- Interacción con dispositivos físicos y mecánicos
- Robustez

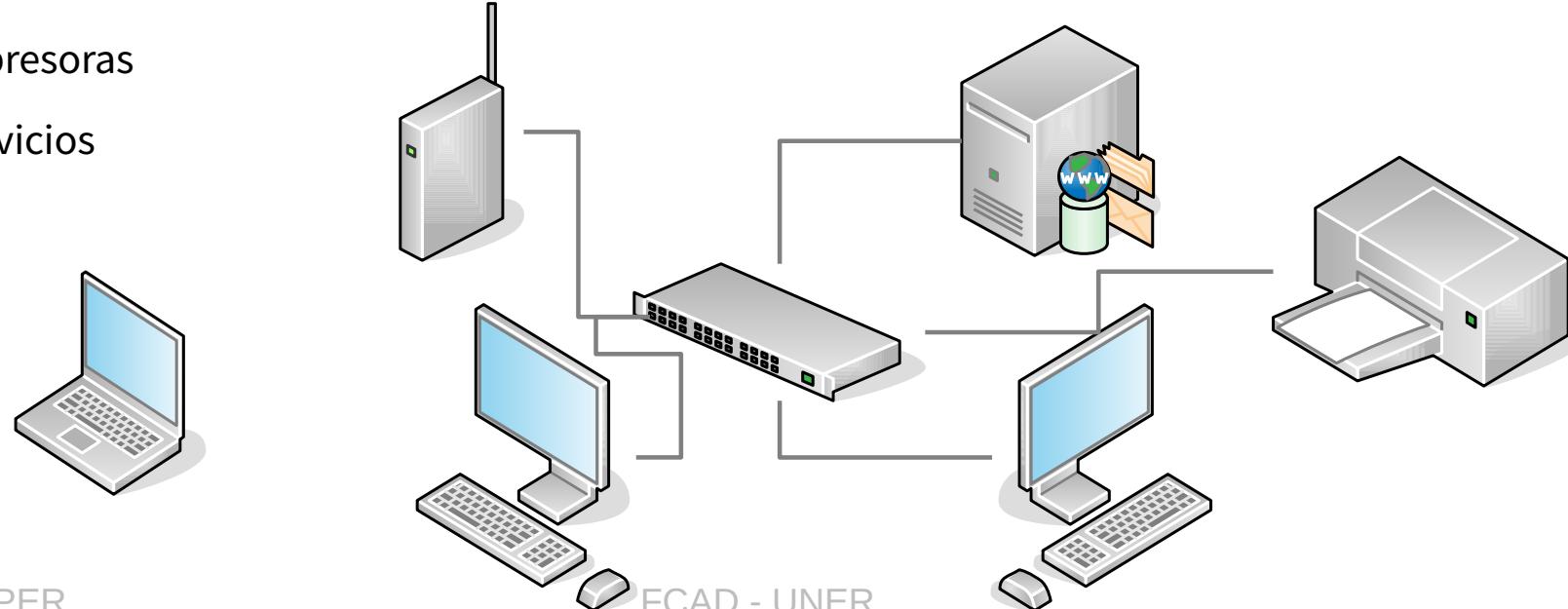


# Sistemas Operativos de Red

Surgieron luego de la descentralización de la información. (Computadora Personal)

Forman redes con otros equipos con Sistemas Operativos de Red con el objetivo de compartir recursos:

- Sistema de Archivos
- Impresoras
- Servicios



# Sistemas Distribuidos

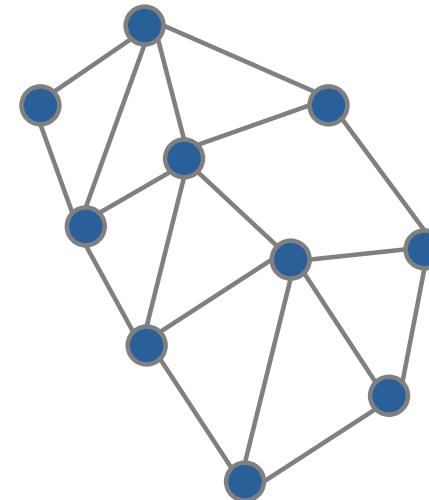
*“Conjunto de computadoras independientes que se presenta a los usuarios como un sistema único y centralizado”*

Los nodos (hardware) son autónomos, es decir no requieren un nodo que coordine la operación del resto.

Los usuarios acceden y usan (software) el sistema como si se tratara de una única computadora.

*Objetivos:*

- *Compartir recursos, Comunicación*
- *Agilizar cálculos*
- *Confiabilidad, Tolerancia a fallos*
- *Transparencia*



## Sistemas Distribuidos > Ventajas

- Mayor economía de adquisición.
- Sin limitaciones geográficas.
- Mayor potencia de cómputo.
- Mayor confiabilidad.
- Crece agregando nodos al sistema.
- Aprovecha recursos ociosos.

**Pero...**

- Depende de la red.
- Hay poco software para Sistemas Distribuidos.
- La gestión de la seguridad puede ser compleja.



# Sistemas Operativos para Dispositivos Móviles

Sistemas operativos para teléfonos inteligentes, tablets, relojes inteligentes, anteojos inteligentes, etc.  
(No considera a las notebooks).

Combinan características de los Sistemas Operativos de Escritorio con características propias:

- Conectividad inalámbrica:
  - Wi-Fi
  - Redes telefónicas
- Interfaz gráfica táctil y limitada.
- Dependencia de una batería.





Facultad de Ciencias de la Administración

# Introducción a los Sistemas Operativos

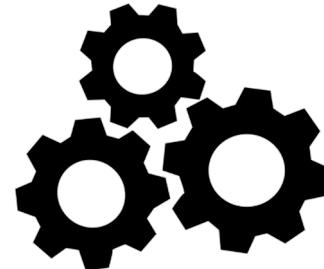
Funciones. Llamadas al Sistema, Excepciones y Protecciones. Núcleo del Sistema Operativo.

# Funciones de los Sistemas Operativos

- Interfaz de Usuario: CLI (de línea de comandos) o GUI (gráfica).



- Ejecución de programas: crear el ambiente adecuado para su ejecución.



# Funciones de los Sistemas Operativos

- Operaciones de E/S: los usuarios no manipulan directamente esos dispositivos.



- Manipular el Sistema de Archivos: acceder a los archivos, recuperar y escribir información en ellos. Gestión de permisos.



# Funciones de los Sistemas Operativos

- Comunicaciones: entre procesos en la misma o distintas computadoras.



- Detección de Errores: tanto de hardware, como de software. Debe proveer una serie de acciones a los fines de recuperar el normal funcionamiento.



# Funciones de los Sistemas Operativos

- Asignación de Recursos: uso eficiente. Algoritmos de planificación.



- Responsabilidad: estadísticas sobre el uso del sistema por diferentes usuarios.



- Protección y Seguridad: controlar el acceso a la información. Además de que los diferentes usuarios/procesos, no interfieran unos con otros.



# Llamadas al Sistema

Peticiones que realizan los procesos en modo usuario al Sistema Operativo para que ejecute un procedimiento o rutina proporcionadas por el núcleo (instrucciones privilegiadas).

Es la forma en que los procesos se comunican con el Sistema Operativo.

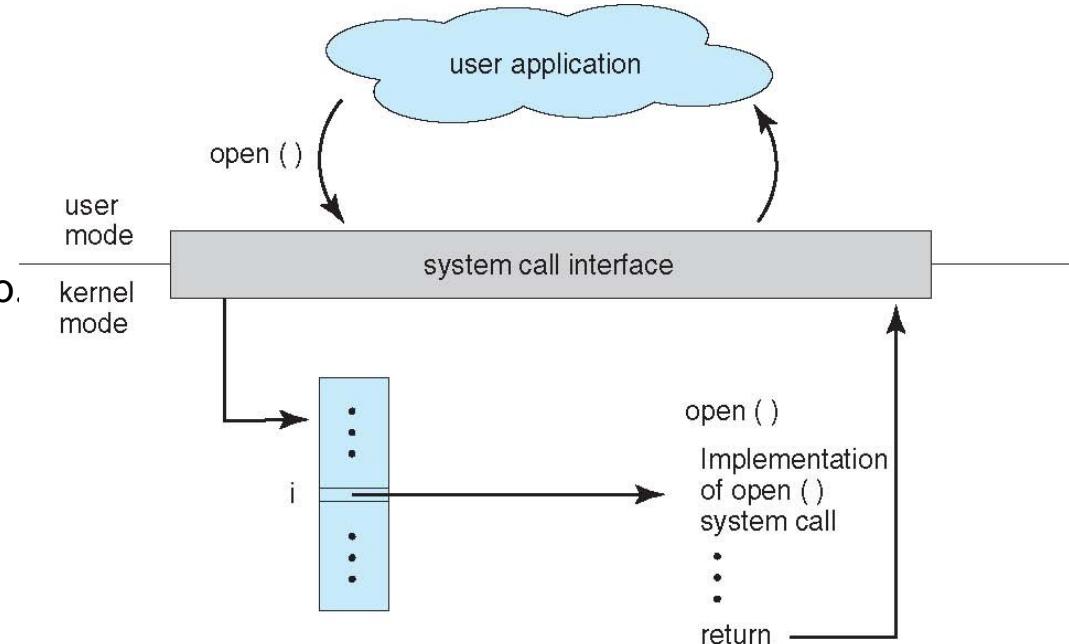
Categorías:

- Control de procesos: finalizar, cargar, ejecutar, establecer y obtener atributos, esperar...
- Manipulación de archivos: crear, eliminar, abrir, escribir, establecer y obtener atributos ...
- Manipulación de dispositivos: solicitar, liberar, leer, escribir...
- Mantenimiento de Información: obtener fecha y hora, datos del sistema...
- Comunicaciones: establecer y cerrar comunicación, enviar y recibir mensajes...



# Llamadas al Sistema > Secuencia de Operación

- 1) Interrupción software.
- 2) Despacho (Modo Núcleo+Salvar contexto).
- 3) Identificación de función de la llamada.
- 4) Comprobación de los parámetros.
- 5) Copia de los parámetros a memoria del Núcleo.
- 6) Ejecución de la función.
- 7) Preparar valores a retornar.
- 8) Restaurar contexto → Modo usuario .



# Llamadas al Sistema > Ejemplos

	Windows	Unix
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
File Manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	open() read() write() close()
Device Manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
Information Maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	pipe() shmget() mmap()
Protection	SetFileSecurity() InitializeSecurityDescriptor() SetSecurityDescriptorGroup()	chmod() umask() chown()

# Gestión de Excepciones

Son condiciones de error, que el procesador reconoce como excepciones.

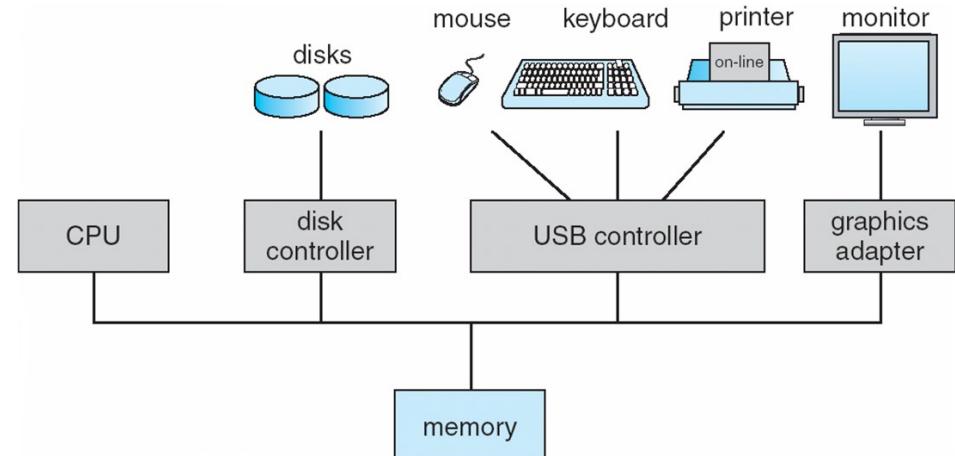
- Generada por una condición que surge de la ejecución de una instrucción, como por ejemplo un desbordamiento aritmético, una división por cero, un intento de ejecutar una instrucción ilegal o una referencia errónea a memoria.
- Es un evento síncrono y es interno al proceso.



# Interrupciones de Dispositivos de E/S

Una interrupción es un suceso externo al procesador que cambia el flujo normal de ejecución del procesador.

- Es un evento asíncrono.
- Los dispositivos de E/S y el procesador operan concurrentemente.



# Protecciones

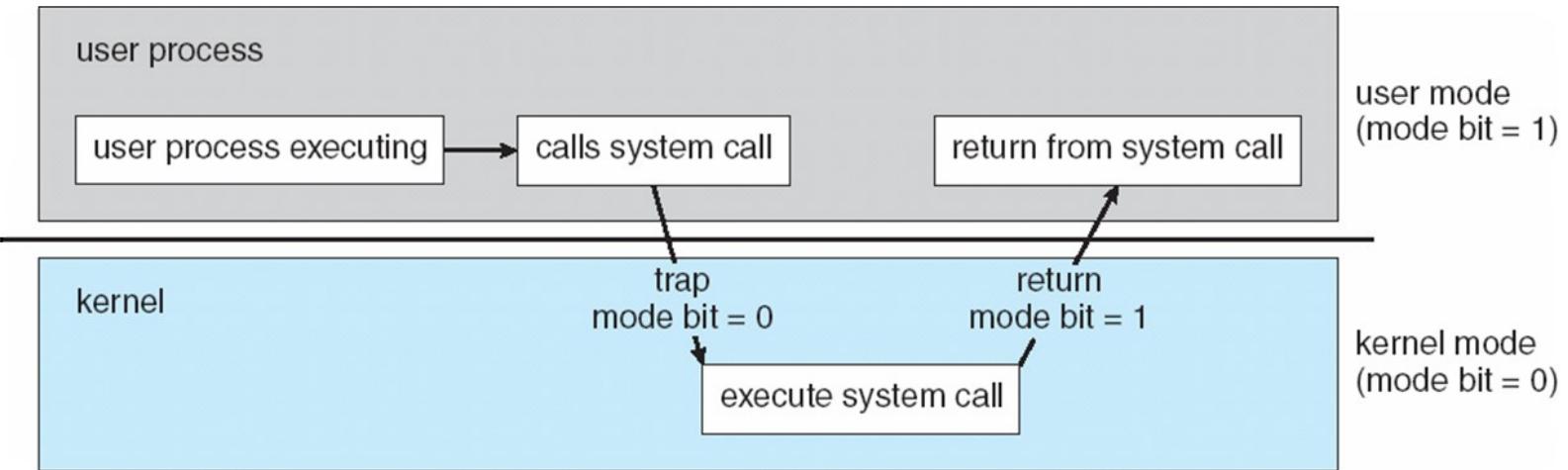
El Sistema Operativo debe proveer mecanismos para evitar conflictos entre procesos y entre procesos y el Sistema Operativo:

- Operación dual
- Protección de Entrada/Salida
- Protección de Memoria
- Protección de Procesador



## Protecciones > Operación Dual

- Modo usuario y modo núcleo.
- Bit provisto por hardware.



## Protecciones > Entrada / Salida

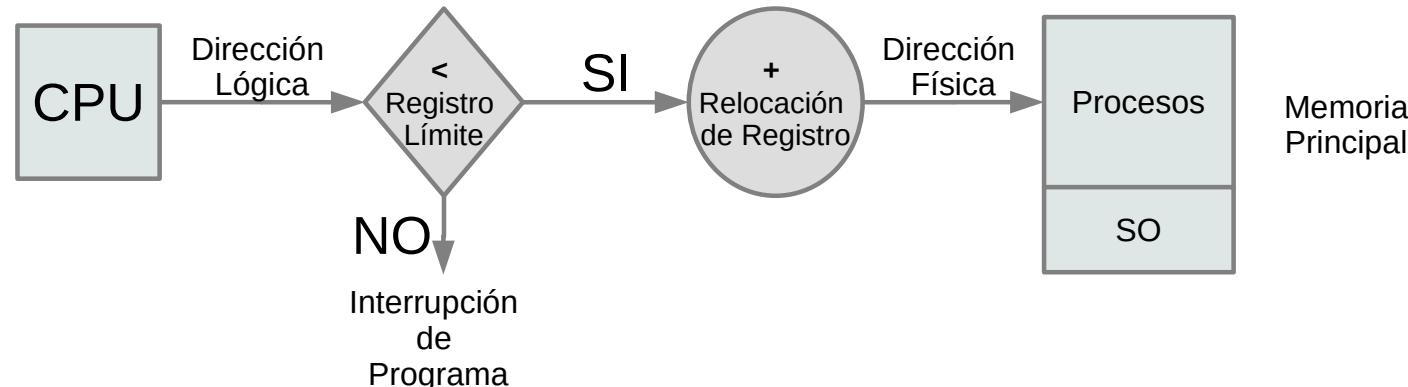
- Los usuarios no tienen acceso directo a los dispositivos.
- No pueden manipularlos directamente.
- Las funciones que pueden realizar están limitadas por las que define el driver.



## Protecciones > Memoria

Se debe evitar la asignación de memoria o el acceso a zonas ya asignadas al sistema operativo u otros procesos.

Para ello existen registros de frontera que indican el límite de memoria asignado a proceso. Si la dirección está fuera de rango se genera una interrupción que la atiende el Sistema Operativo.



## Protecciones > Procesador

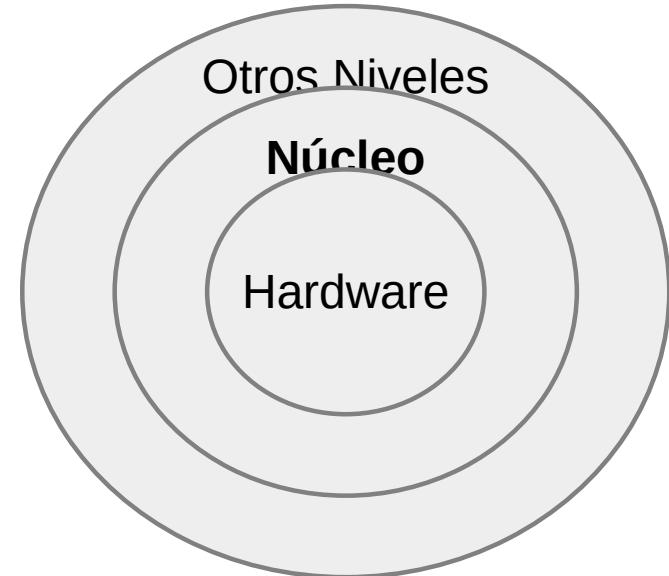
Algunos procesos pueden ejecutar bucles infinitos o adueñarse del procesador, no liberándolo nunca.

- Evitar la apropiación del procesador.
- Utilizar temporizadores.



# Núcleo del Sistema Operativo

- Es la parte más cercana y dependiente del hardware.
- Se encarga de realizar las funciones más básicas del Sistema Operativo.
- Suele permanecer en memoria principal.

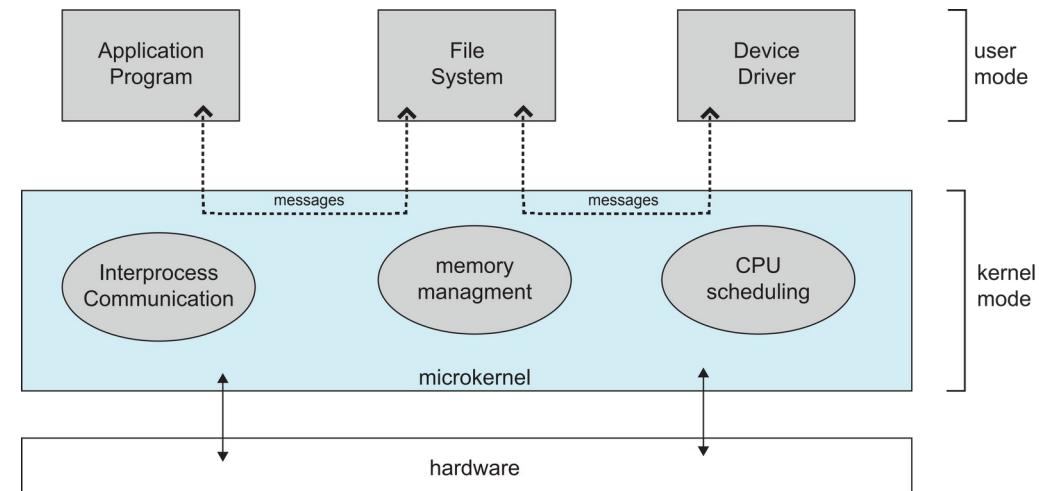


# Núcleo del Sistema Operativo > Categorías

## Monolíticos:

- Dependientes del Hardware: manejo de interrupciones, memoria y dispositivos E/S
- Independientes del Hardware: manejo de llamadas al sistema, archivos y planificación de procesos.

## Micro-Núcleo:



# Núcleo del Sistema Operativo > Categorías > Micro núcleo > Minix

- Capa 1: Administración de procesos, comunicación con el hardware, interrupciones trampas, manejo de mensajes.
- Capa 2: tareas de E/S por cada tipo de dispositivo. Copian los datos en regiones d memoria de los procesos que solicitan la operación.
- Capa 3: procesos servidores. Menos privilegiados que capas anteriores (no pueden acceder a puertos E/S ni a regiones de memoria de otros procesos)
- Capa 4: procesos de usuario



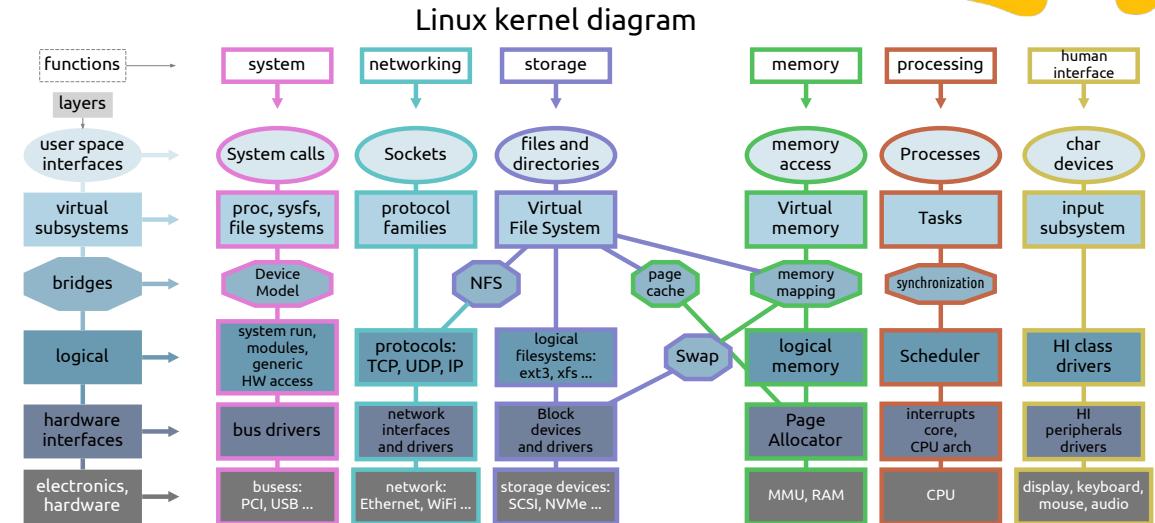
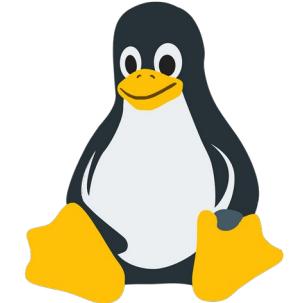
Capa							
4	Init	Proceso de usuario	Proceso de usuario	Proceso de usuario	...		Procesos de usuario
3	Administrador de memoria	Sistema de archivos		Servidor de red	...		Procesos servidores
2	Tarea de disco	Tarea de terminal	Tarea de reloj	Tarea de sistema	Tarea de Ethernet	...	Tareas de E/S
1	Administración de procesos						

# Núcleo del Sistema Operativo > Categorías > Monolítico > Linux

Es un kernel **Monolítico Híbrido**:

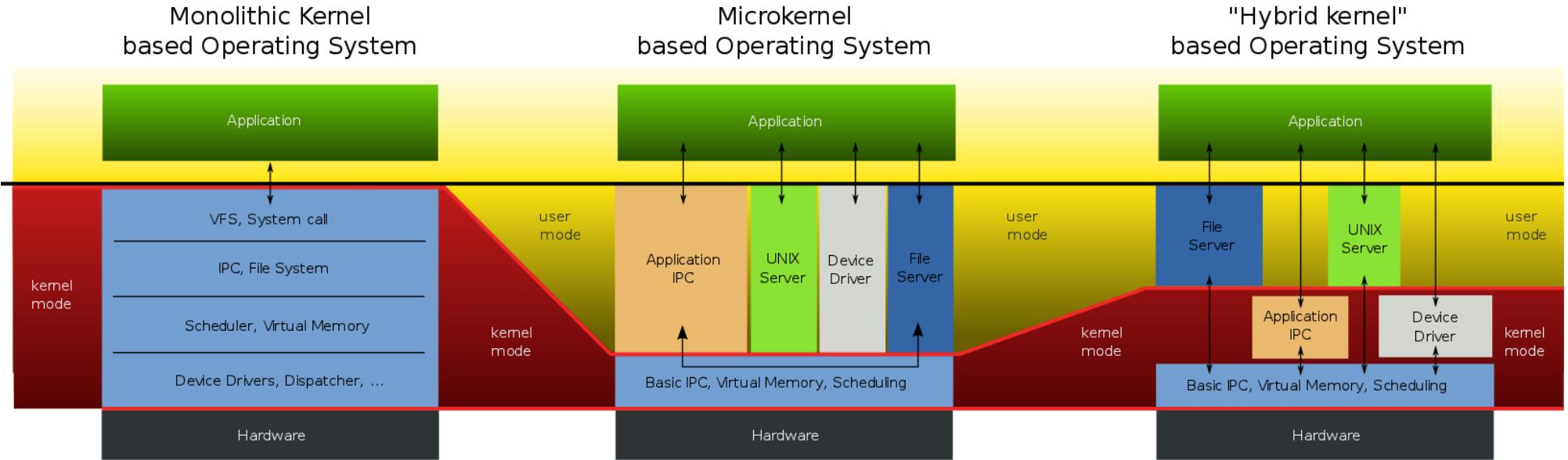
Pueden cargarse y descargarse como módulos sin interrumpir el sistema:

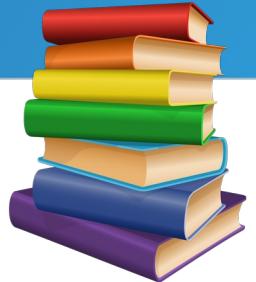
- Controladores de dispositivos
- Extensiones del núcleo



© 2007-2019 Constantine Shulyupin <http://www.MakeLinux.net/kernel/diagram>

# Núcleo del Sistema Operativo > Comparación





## Fuentes y Bibliografía

- **SILBERSCHATZ** - *Fundamentos de Sistemas Operativos*
  - *Capítulo 1 - Introducción*
  - *Capítulo 2 - Estructuras de los Sistemas Operativos*



Facultad de Ciencias de la Administración

# Memoria

Concepto, Tipos y Jerarquías. Memoria RAM. Requerimientos. Direcciones Lógicas y Físicas y su resolución. Asignación de Memoria: Paginación, Segmentación y Segmentación Paginada. Memoria Virtual. Fallos de Página. Algoritmos de Reemplazo de Páginas.

# Introducción

La función de la computadora es la de procesar información mediante la ejecución de programas.

Esos programas deben co-residir en la memoria principal durante su ejecución, junto con los datos que procesan y el Sistema Operativo.

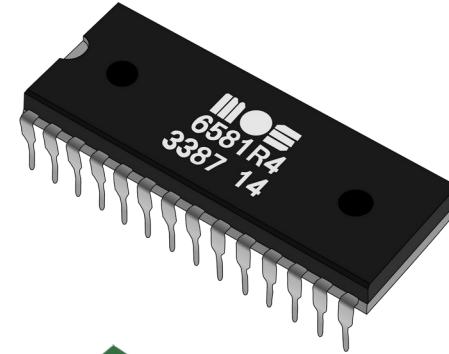
Son funciones de la administración de memoria:

- Asignar memoria a los procesos.
- Proteger la memoria asignada de accesos no autorizados.

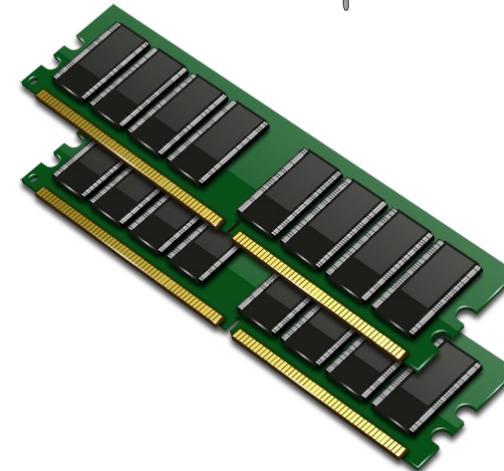


# Tipos de Memoria

- Memoria ROM: de sólo lectura.



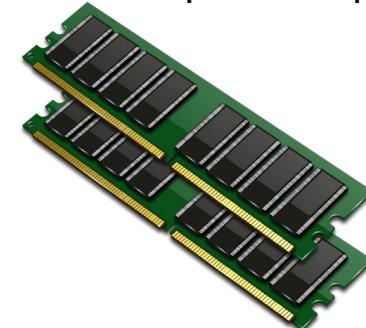
- Memoria RAM: de lectura y escritura.



# Memoria Principal

Es una serie de circuitos donde se almacenan **temporalmente** los datos y programas que la CPU está **procesando o va a procesar**.

- Al ser una memoria de tipo RAM, la CPU, puede **acceder a directamente** a cualquier posición de la misma.
- La CPU solo puede acceder a sus registros y a la memoria principal en forma directa.
- A los fines de acelerar el acceso a la memoria, se dispone de una **memoria cache** entre la memoria principal y la CPU.
- Cada proceso dispone de un espacio de memoria **separado**. Se debe garantizar que cada proceso solo acceda a su espacio.



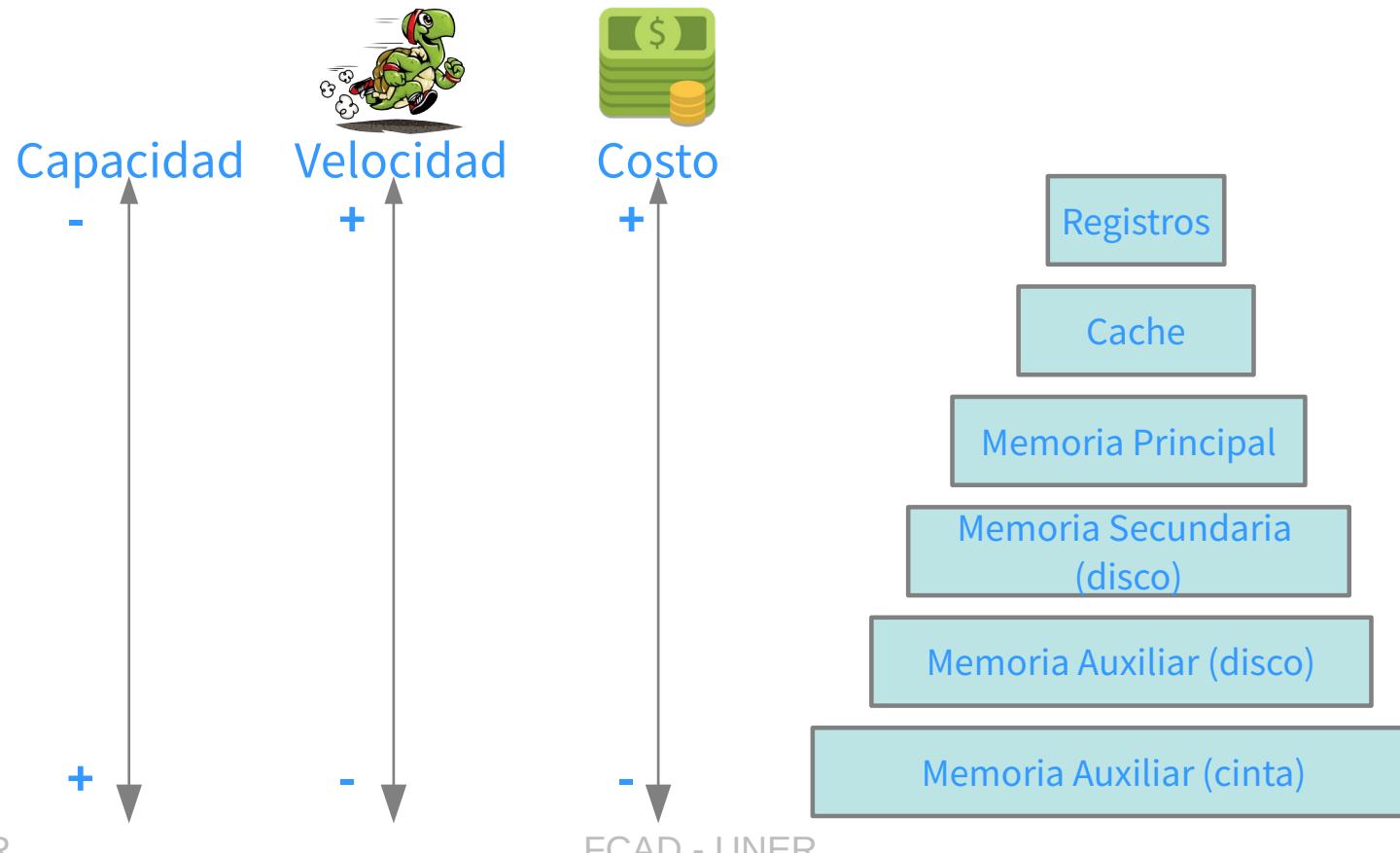
# Jerarquías de Memoria

Se deben tener en cuenta diversos criterios para el diseño de memorias:

- Costo/bit
- Velocidad
- Capacidad
- Consumo
- Fiabilidad

Existen ciertas incompatibilidades entre los criterios. Por eso se utiliza una **organización jerárquica** de la memoria de una computadora.

# Jerarquías de Memoria

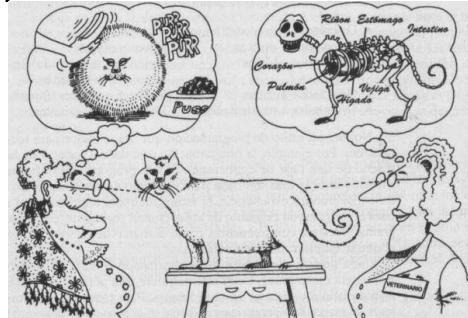


# Requerimientos de la Administración de Memoria

- Protección



- Transparencia



- Segmentos múltiples



- Código Compartido



# Direcciones Físicas y Lógicas

- **Direcciones Físicas:** es la ubicación absoluta en la memoria principal. Es única para todo el sistema.
- **Direcciones Lógicas:** hacen referencia a una locación de memoria independientemente de la asignación de los datos en la memoria.  
Son una abstracción de las direcciones físicas, por lo que debe haber un proceso de **traducción**.



# Espacio de Direcciones

Cada proceso posee su espacio de direcciones, al que accede en **modo usuario**.

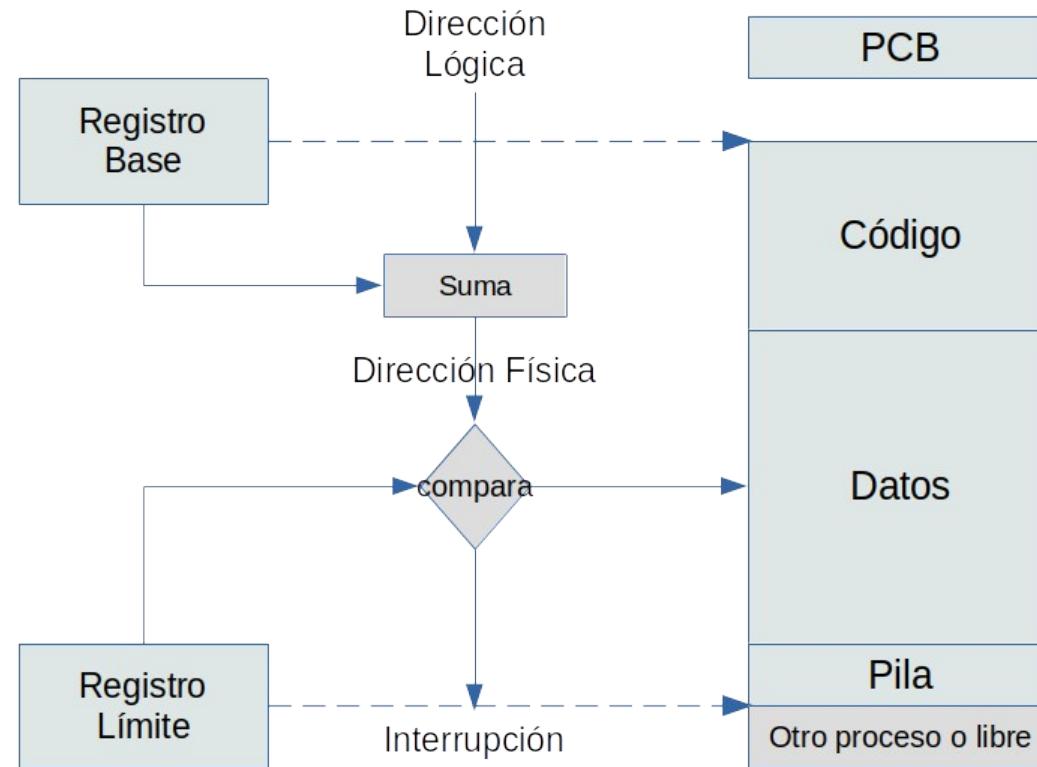
Es decir que puede realizar modificaciones dentro de su espacio de direcciones sin que intervenga el Sistema Operativo.

Este espacio está limitado por:

- **Registro Base**: dirección donde comienza el proceso.
- **Registro Límite**: dirección donde termina el proceso.



# Espacio de Direcciones

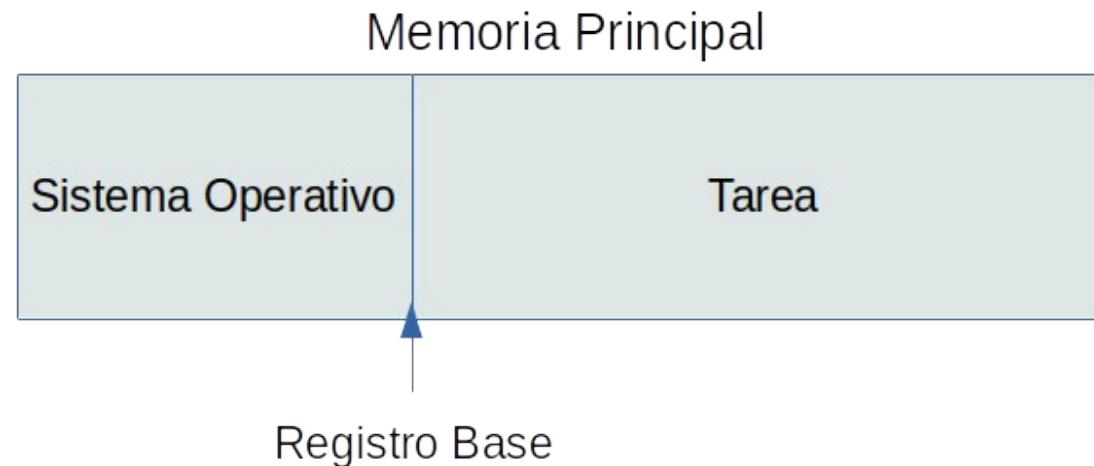


# Resolución de Direcciones

- **En tiempo de Compilación:** se generan direcciones **absolutas**. Exige la carga siempre en el mismo lugar de memoria. Se debe recomilar para realizar un cambio de direcciones. Las direcciones coinciden.
- **En tiempo de Carga:** cuando se conoce la dirección del inicio del proceso. De ahí en más ese proceso se ejecuta desde el mismo lugar en memoria. Requiere un mecanismo de resolución de direcciones. Las Direcciones difieren en un **factor de reubicación** (estático).
- **En tiempo de Ejecución:** permite realizar el **intercambio** de procesos entre memoria y disco, y que el proceso vuelva a ser cargado en cualquier ubicación. Direcciones lógicas y físicas difieren. Las direcciones lógicas son llamadas **direcciones virtuales**. La correspondencia se establece por una **tabla de mapeo** (MMU).

# Asignación de Memoria

En sistemas operativos **monoprogramados** se puede utilizar un esquema simple. Se divide el espacio de direcciones en dos, uno para el sistema operativo y otro para la tarea del usuario.



# Asignación de Memoria

En los sistemas ***multiprogramados***, necesitamos compartir la memoria entre todos los procesos.

- Particiones Fijas de igual tamaño.
- Particiones Fijas de diferente tamaño.
- Particiones Dinámicas.

Se van a emplear diferentes métodos para asignar los procesos a las particiones:

- Primer Ajuste
- Siguiente Ajuste
- Peor Ajuste
- Mejor Ajuste

## Asignación de Memoria > Particiones Fijas

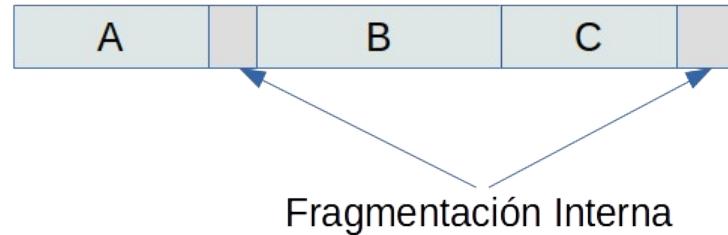
- El operador dividía la memoria principal en particiones de igual o diferente tamaño.
- La partición se asigna por completo al proceso. Puede producirse un desperdicio si el proceso es más pequeño.
- Hay que determinar un tamaño óptimo para las particiones.

## Asignación de Memoria > Particiones Dinámicas

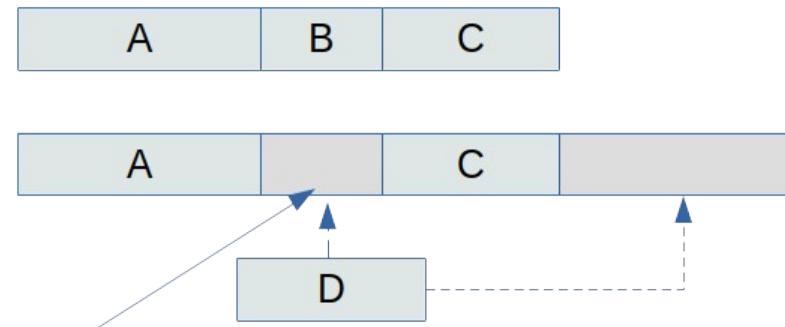
- Inicialmente toda la memoria está disponible y se considera como un gran hueco.
- Al llegar una tarea se busca un hueco disponible y se asigna la porción de memoria que necesita. El sobrante constituye un nuevo hueco.
- Cuando un proceso finaliza libera su partición de memoria, constituyendo un nuevo hueco. Puede fusionarse con huecos adyacentes.
- Pueden producirse huecos muy pequeños, no adyacentes con otros huecos.

# Fragmentación de Memoria

- **Fragmentación Interna:** espacio libre de memoria que no puede ser utilizado por estar asignado a un proceso.



- **Fragmentación Externa:** espacio libre de memoria que no puede ser utilizado por ser muy pequeño



# Paginación

La memoria principal es dividida lógicamente en bloques de tamaño fijo llamados **marcos**.

Los procesos se dividen en partes iguales llamadas **páginas**, del mismo tamaño que los marcos.

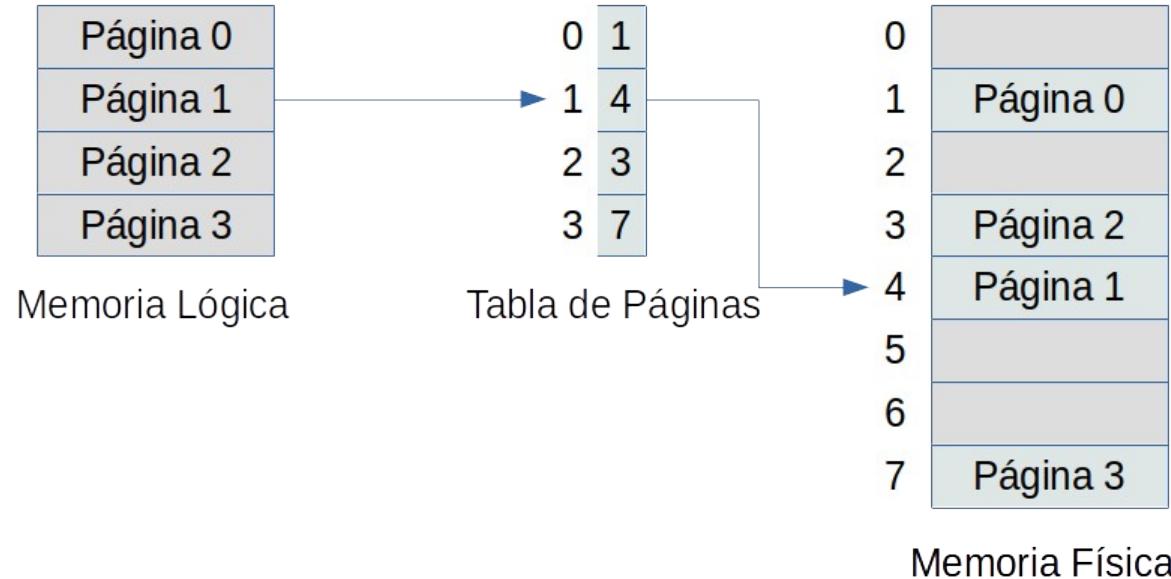
La correspondencia entre páginas y marcos de página está dada por la **Tabla de Páginas** de cada proceso.

Permite que el espacio de direcciones físicas de un proceso no sea contiguo.

Puede haber **fragmentación interna** en el último marco de página de cada proceso.

**Tamaño de página** → sobrecarga vs desperdicio de memoria.

# Paginación



## Paginación > Direcciones

Las direcciones lógicas van a estar formadas por:



Las direcciones físicas van a estar formadas por:



## Paginación > Soporte Hardware

Las tablas de páginas de los procesos residen en memoria y deben ser consultadas cada vez que se quiere acceder al espacio de direcciones del mismo.

### **¿Podemos optimizar la velocidad de acceso?**

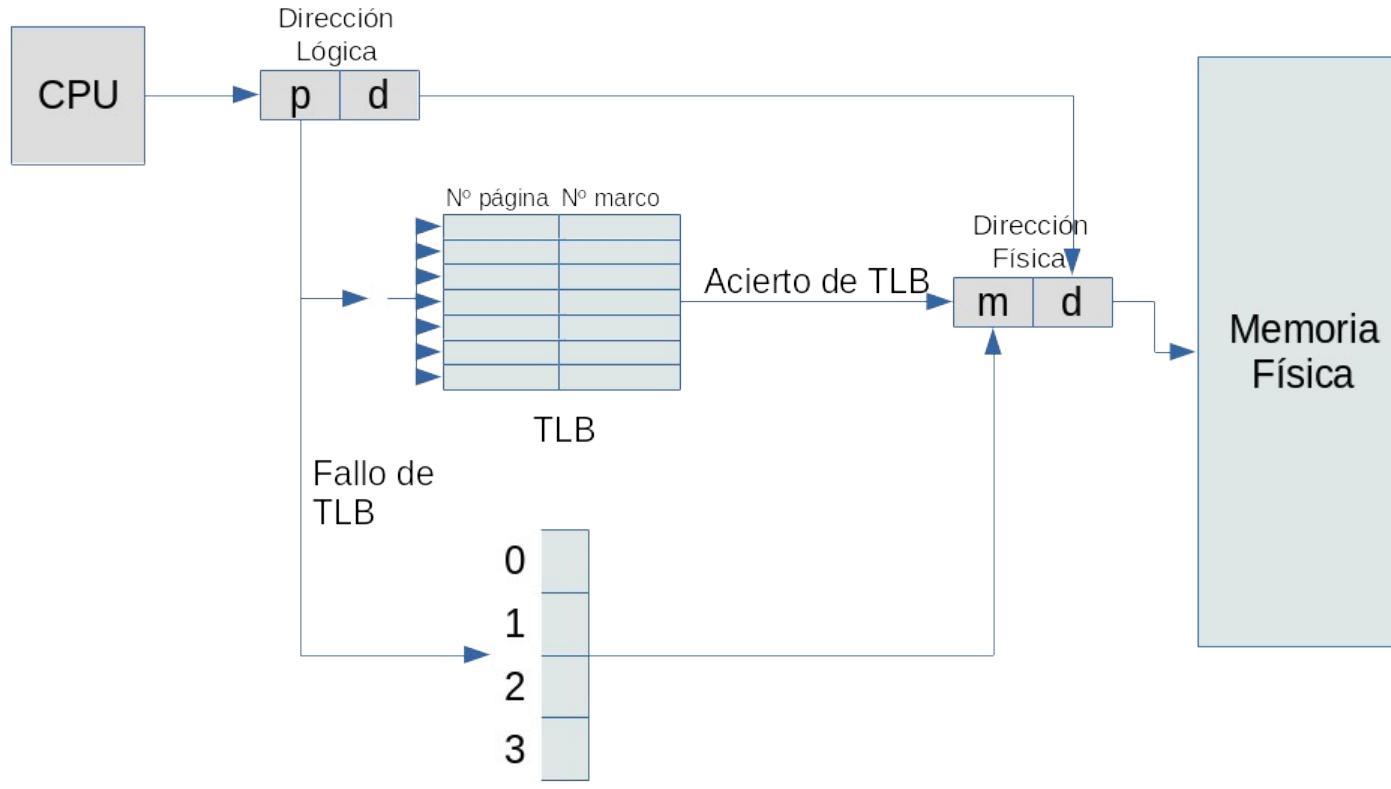
Utilizamos una memoria cache llamada **TLB** (Translation Look-aside Buffer). Es una memoria asociativa, de gran velocidad, pero pequeña (elevado costo).

El TLB va a almacenar algunas de las entradas de la tabla de páginas.

La consulta se realiza en paralelo.



## Paginación > Soporte Hardware





Facultad de Ciencias de la Administración

# Caso de Estudio: GNU/Linux

Software libre. Licencias. Surgimiento y evolución de GNU/Linux. Características. Distribuciones. Trabajo desde la terminal. Comandos. Gestores de paquetes.

# ¿GNU/Linux es Gratis?

No es gratis, es **libre**.

*you should think of “free” as in “free speech,” not as in “free beer”*

El “Software Libre” es un asunto de **libertad**, no de **precio**.

Que un software sea libre implica que se respetan ciertas libertades que favorecen a quienes lo desarrollan y sobre todo a quienes lo usan.

# Libertades del Software Libre

Licencia GNU/GPL (Licencia Pública General)

- 0) La libertad de **usar** el programa, con cualquier propósito.
- 1) La libertad de **estudiar** cómo funciona el programa, y adaptarlo a tus necesidades. (\*)
- 2) La libertad de **distribuir copias**, con lo que puedes ayudar a tu vecino.
- 3) La libertad de **mejorar** el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (\*)



(\*) El acceso al código fuente es una condición previa para esto.

# CopyRight y CopyLeft

Copyleft es la regla que implica que, cuando se redistribuya el programa, **no se puede agregar restricciones** para denegar a otras personas las libertades centrales.

Esto **protege** a las libertades.



# Comunidad del Software Libre

Hay una gran cantidad de personas, en todo el mundo, **colaborando** en proyectos de Software Libre.

Colaborar implica:

- Programar
- Documentar
- Probar y reportar errores
- Hacer sugerencias
- ...y usarlo

Esto hace que haya una **gran diversidad de soluciones a una gran diversidad de problemas**.



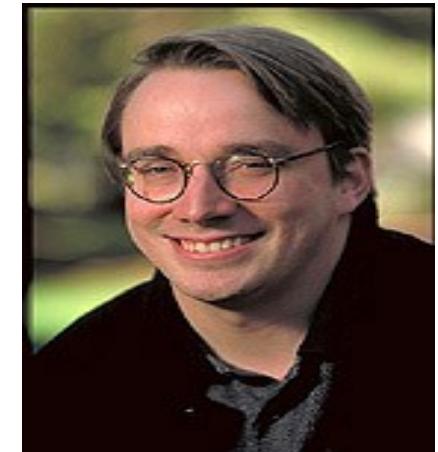
## Historia > Richard Stallman (RMS)

- Creador de Emacs gcc y GDB.
- Manifiesto GNU y concepto de CopyLeft.
- Laboratorio de Inteligencia Artificial del MIT.
- Proyecto GNU (1984).

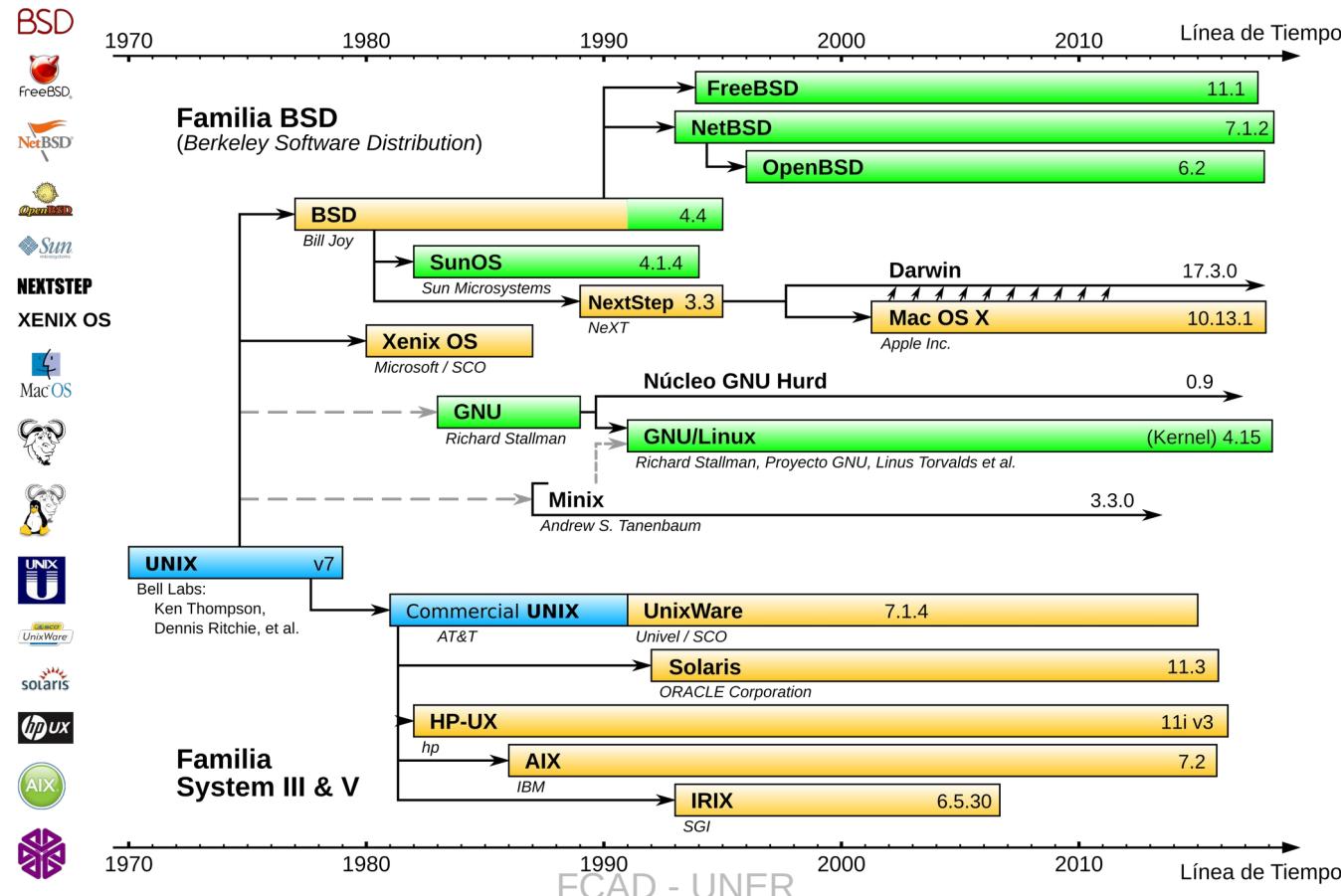


## Historia > Linuz Torvads

- Helsinki (Finlandia)
- Decidió crear una especie de UNIX basándose en el kernel de Minix.
- Primera versión del núcleo (1991)
- Liberó el kernel bajo la licencia GNU/GPL.



# Historia > Surgimiento y Evolución



# Características

- Tiempo Compartido
- Multiusuario
- Multitarea
- Escrito en su mayor parte en lenguaje C.

# Distribuciones

**Distribución = Núcleo + Sistema Base + Aplicaciones**

Hay distribuciones desarrolladas por la comunidad y otras por empresas.



## Ubuntu > Historia

- Distribución de GNU/Linux basada en Debian.
- Es desarrollado y mantenido por Canonical / Ubuntu Foundation.
- Mark Shuttleworth (Sudáfrica).
- Lanzamiento en Octubre de 2004.

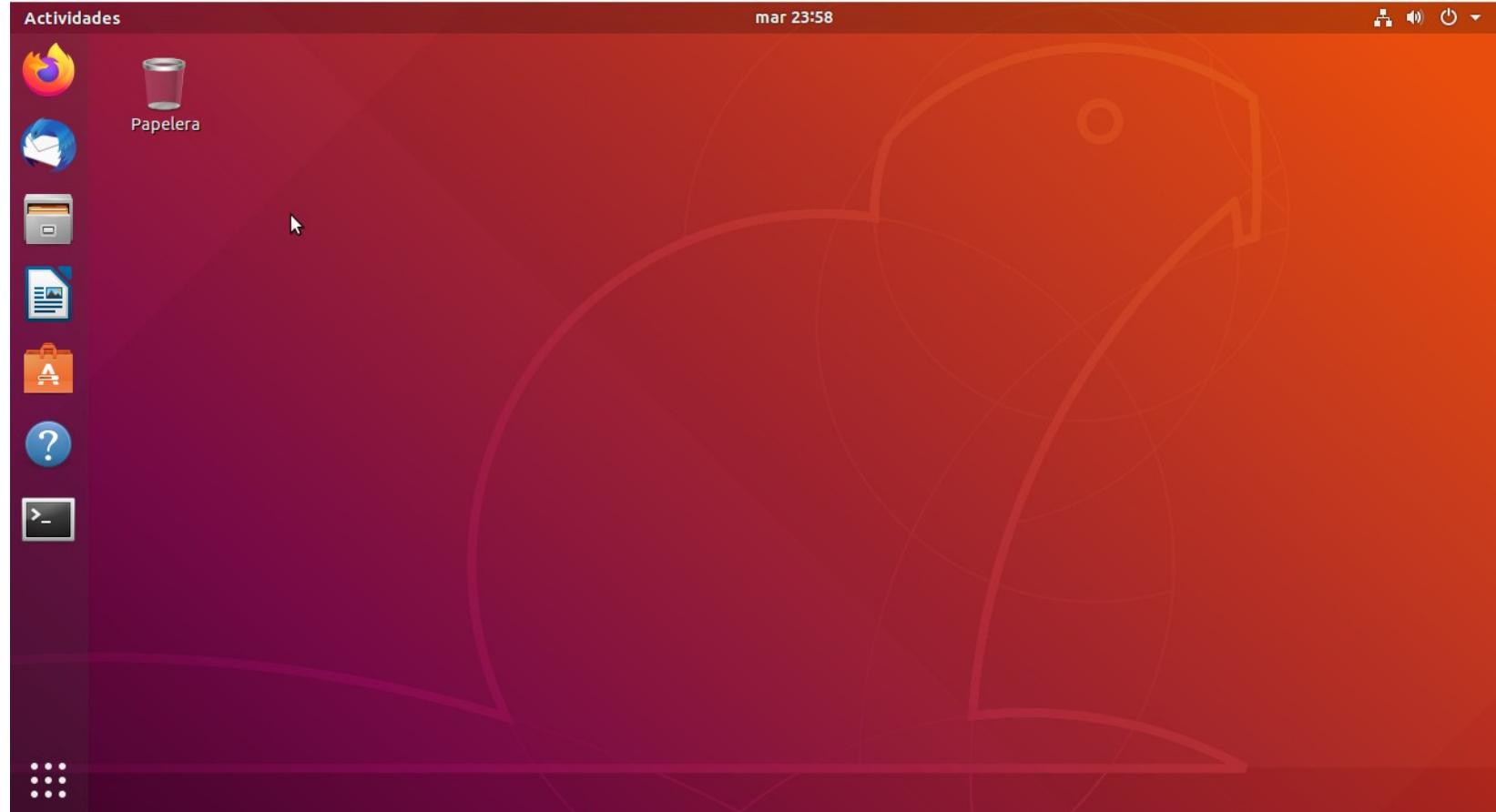


## Ubuntu > Versiones

- Cada 6 meses se libera una nueva versión → Soporte 9 meses
- Cada 2 años se libera una versión LTS → Soporte 5 años.
- Las versiones tienen un nombre en clave (adjetivo + animal) con la misma letra inicial y en orden alfabético.

Versión	Nombre en clave	Lanzamiento	Fin Soporte
18.04 LTS	Bionic Beaver	26/04/2018	04/2023
20.04 LTS	Focal Fossa	23/04/2020	04/2025
21.10	Impish Indri	14/10/2021	07/2022
22.04 LTS	Jammy Jellyfish	21/04/2022	04/2027
22.10	Kinetic Kudu	20/10/2022	07/2023

# Ubuntu > Interfaz Gráfica

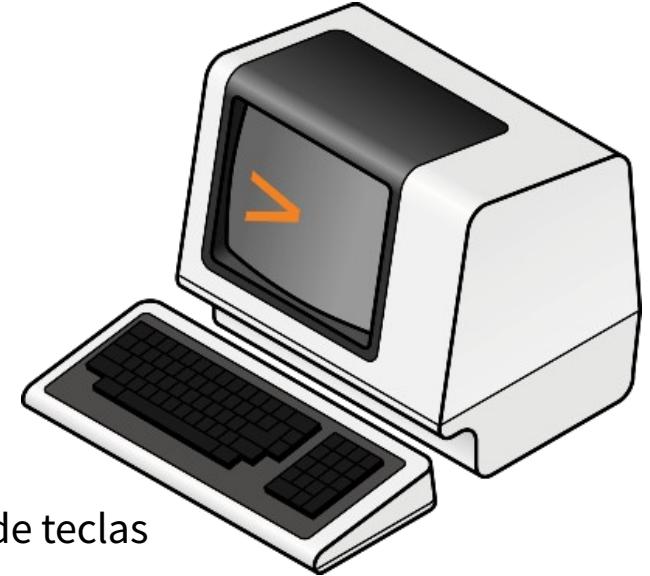


# Terminal vs Consola

- Terminal: entorno de entrada / salida de texto.
- Consola: terminal física.
- Shell: intérprete de comandos.

En Linux podemos abrir varias terminales virtuales con la combinación de teclas

Alt + Ctrl + F1...F6



## Terminal > Características

- Prompt

username@hostname:workingdirectory\$

- Autocompletar (tab)
- Historial de comandos (flecha arriba y abajo)
- Sensible a mayúsculas/minúsculas

## Terminal > Comandos

```
$ comando -opciones parámetro_1 ... parametro_n
```

← →  
Argumentos

**Opciones:** modifican el comportamiento del comando. Están establecidas en el código del programa.

Opciones cortas: -a

Opciones largas: --all

**Parámetros:** son datos que se ajustan a las necesidades del usuario. Suelen ser obligatorios y se debe respetar un orden.

Las opciones también pueden tener parámetros

```
$ ls --tabsize=3 o $ ls -T 3
```

## Terminal > Obteniendo Ayuda

**man**: muestra la documentación (páginas del manual) sobre el comando, archivo o función solicitada.

```
$ man ls
```

**--help**: muestra un mensaje por pantalla sobre el modo de empleo del comando.

```
$ ls --help
```

**whatis**: muestra un breve mensaje por pantalla sobre el propósito del comando.

```
$ whatis ls
```

# El usuario root

**root** es el usuario encargado de la administración del sistema (super usuario).

\$ vs #

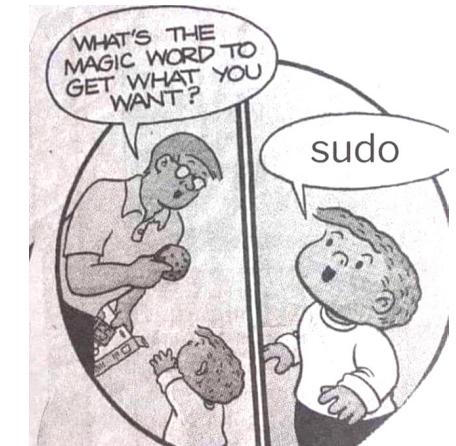
En Ubuntu esta cuenta está deshabilitada por razones de seguridad.

Los usuarios hacen uso del comando sudo para ejecutar tareas que requieren permisos especiales.

\$ sudo apt-get update

¿Quiénes pueden usar sudo?

Usuarios **sudoers**: primer usuario creado durante la instalación y miembros del grupo **admin**.



## Comandos > Manejo de Directorios

- ***ls***: muestra un listado de los archivos y directorios

```
$ ls [-opciones] [directorio]
```

- ***pwd***: muestra el directorio de trabajo

- ***cd***: cambia de directorio

```
$ cd directorio_destino
```

## Comandos > Manejo de Directorios

- ***mkdir***: crea un directorio

```
$ mkdir /home/user/directorio_nuevo
```

```
$ mkdir -p /home/user/directorio1/directorio2
```

- ***rmdir***: elimina un directorio (siempre que se encuentre vacío)

```
$ rmdir /home/user/directorio1/directorio2
```

## Comandos > Manejo de Archivos

- ***ls***: muestra un listado de los archivos y directorios.

```
$ ls [-opciones] [directorio]
```

- ***cat more less***: muestra el contenido del archivo.

```
$ cat archivo
```

```
$ more archivo
```

```
$ less archivo
```

## Comandos > Manejo de Archivos

- **cp**: copia uno o varios archivos a otro archivo o directorio

```
$ cp archivo1 archivo2
```

```
$ cp /home/user/archivo /home/user/Escritorio
```

- **mv**: mueve o cambia el nombre de un archivo o directorio

```
$ mv archivo1 archivo_uno
```

```
$ mv archivo1 /home/user/Escritorio
```

## Comandos > Manejo de Archivos

- ***rm***: elimina o borra un archivo

```
$ rm archivo1
```

```
$ rm -rf /home/user/copias
```

- ***ln***: crea un enlace entre archivos

```
$ ln archivo1 enlace_archivo1
```



# Actividad para Promocionar (Alternativa 1)

Instalar una versión de Ubuntu en una máquina Virtual (VirtualBox).

**IMPORTANTE:** Utilizar su nombre e inicial del apellido para el nombre de usuario!!

Instalar

¿Quién es usted?

Su nombre: Juan Perez ✓

El nombre de su equipo: sisoperVirt ✓  
El nombre que utiliza al comunicarse con otros equipos.

Elija un nombre de usuario: juanp ✓ ←

Elija una contraseña: ●●●●●●●●  Contraseña aceptable

Confirme su contraseña: ●●●●●●●● ✓

Iniciar sesión automáticamente

Luego de que lo instalen vamos a pedirles algunos ejercicios sencillos y que capturen las pantallas para verificar.



## Actividad para Promocionar (Alternativa 2)

Seleccionar un tema de los tratados en la materia o que guarden relación con ella.

**IMPORTANTE:** acordar previamente el tema con el equipo de cátedra.

Investigar sobre el tema seleccionado.

Realizar y publicar una infografía donde se exponga y amplíe dicho tema.



Facultad de Ciencias de la Administración