

# Administración del Procesador

## Concepto

Involucra las distintas maneras de asignar el control de la CPU a los distintos procesos (planificación - Scheduling).

Para poder evaluar su calidad de una planificación respecto de otra es necesario medirla de alguna forma (métricas).

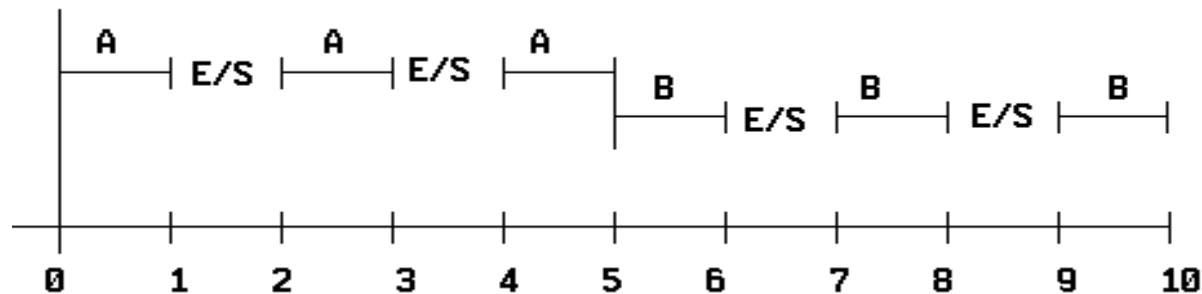
Por ejemplo: THROUGHPUT es la cantidad de procesos en el sistema por unidad de tiempo

## Otra medida: TURNAROUND

Es el tiempo promedio que puede tomarse respecto del tiempo de finalización de los procesos.

También puede tomarse sobre el tiempo de inicio de los procesos.

Ejemplo en un sistema de monoprogramación:

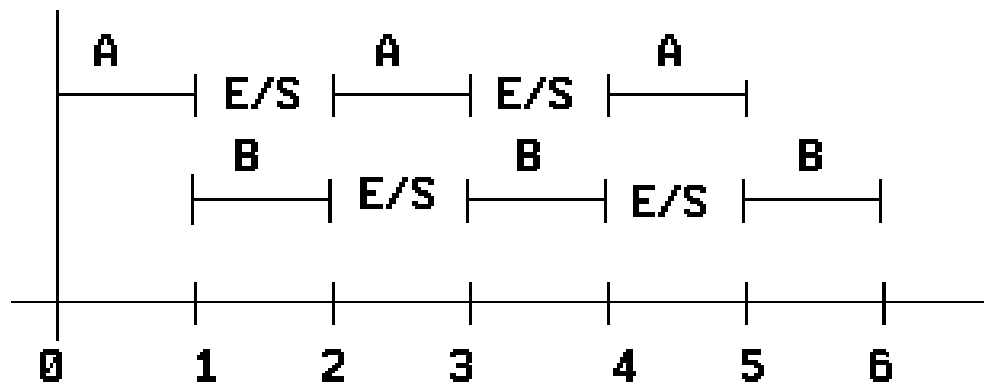


A termina en 5 y B en 10

$$\text{Turnaround} = 5 + 10 / 2 = 7,5$$

## Otro ejemplo de Turnaround

En un sistema de Multiprogramación

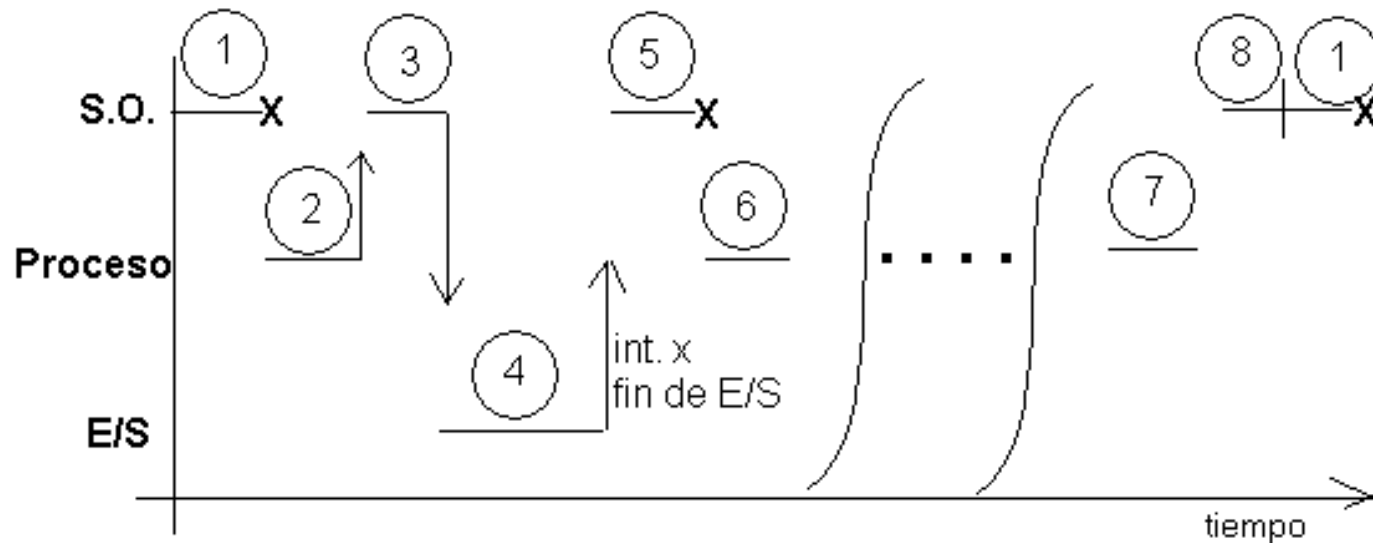


A termina en 5 y B termina en 6

$$\text{Turnaround} = 5 + 6 / 2 = 5,5$$

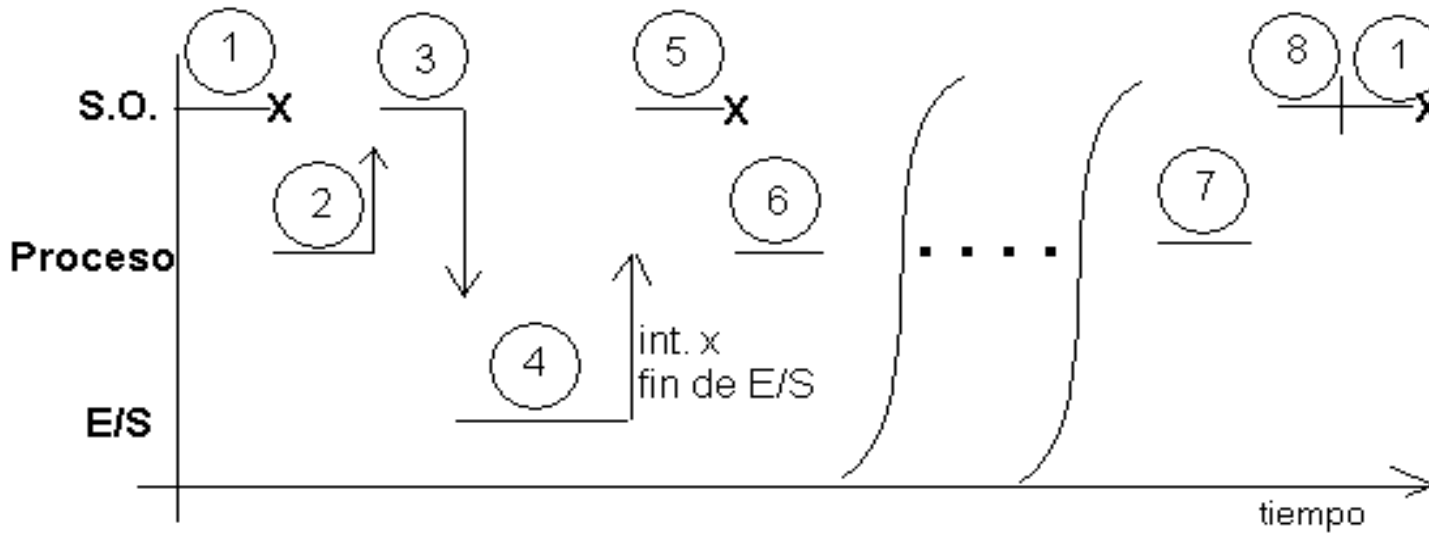
También se denomina al turnaround el **índice de satisfacción del usuario**

Pero en este esquema no tenemos en cuenta los tiempos que insume el SO para administrar al proceso. Veamos un ejemplo en monoprogramación:



1) Tiempo de preparación del proceso: asignación de memoria, carga del código objeto, asignación de periféricos, asignación de archivos

2) el proceso toma el control y comienza a ejecutar

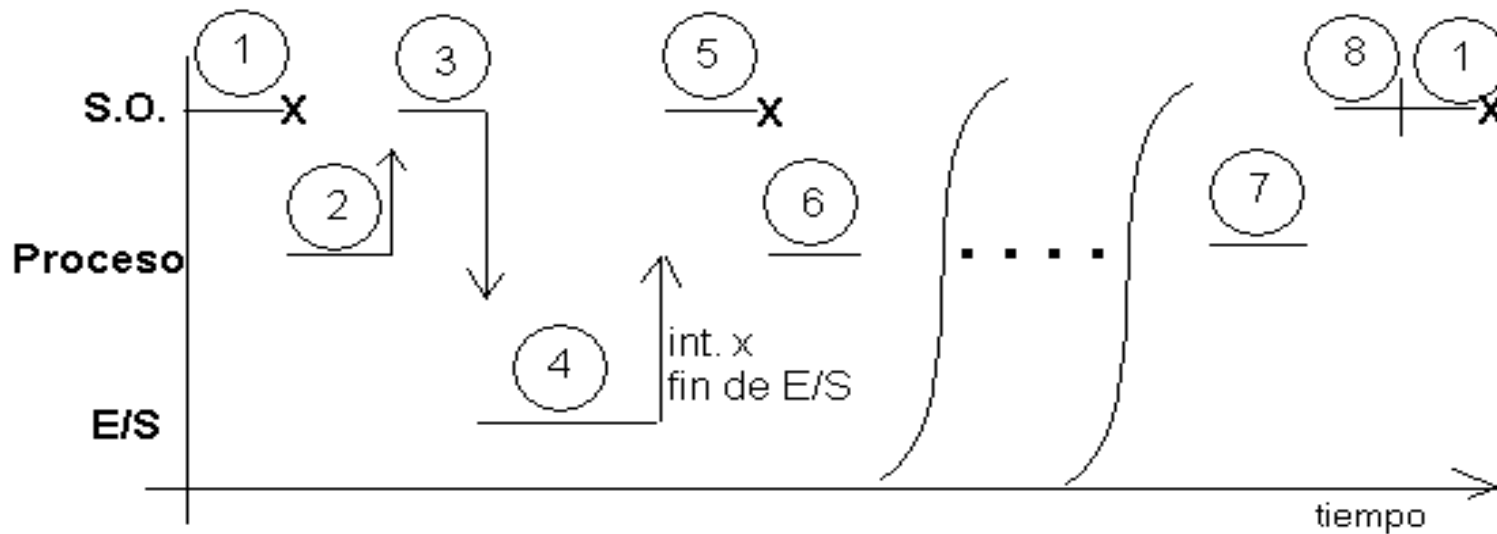


3) el proceso pidió E/S y el sistema la prepara y la lanza

4) el periférico realiza la E/S. CPU ociosa.

5) se produjo una interrupción por fin de E/S y el SO la procesa y verifica fin correcto y pone los datos a disposición del proceso

6) el proceso retoma su ejecución



7) el proceso ejecuta y finaliza su ejecución

8) el SO libera los recursos asignados al proceso:  
libera memoria, descarga el código objeto, desasigna  
periféricos, desasigna de archivos

1) el SO comienza a preparar la carga del próximo  
proceso

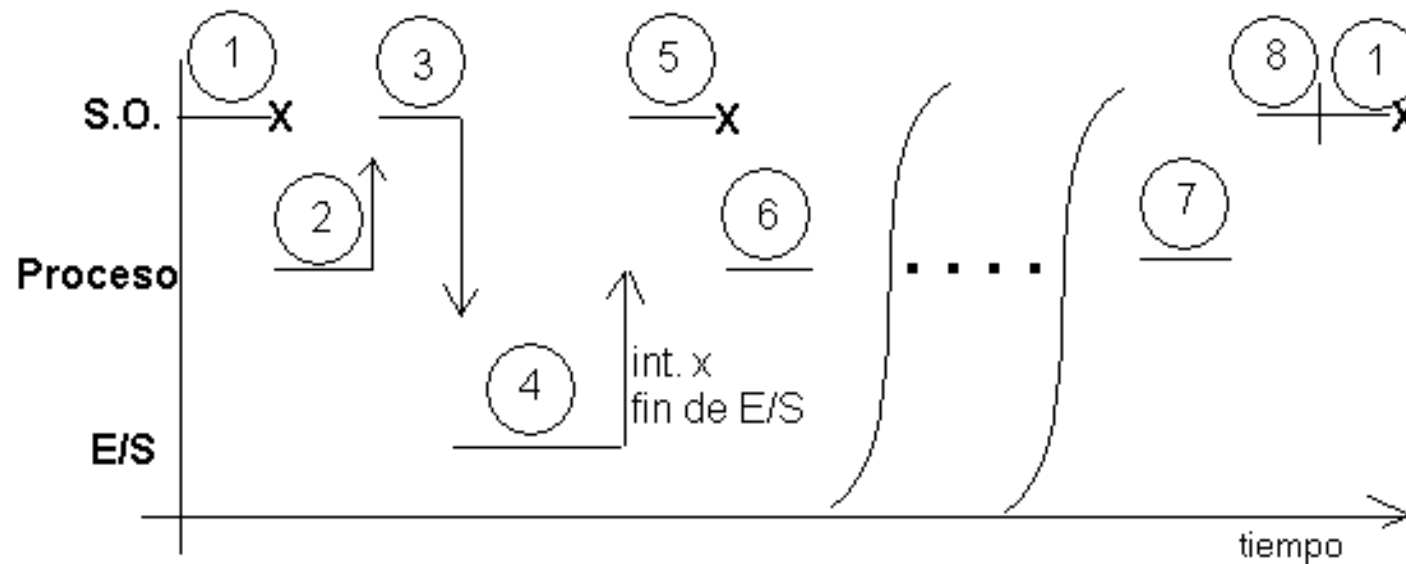
Y en Multiprogramación?

Cuando hay más de un proceso para administrar el SO debe tomar la decisión de a que proceso otorgar el control de la CPU.

Esta acción implica un tiempo considerable no nulo ya que implica la ejecución de un algoritmo de decisión.

La rutina que hace esta función se denomina:  
**Planificador de Procesos**





En todos los momentos marcados con una "X" el Planificador de Procesos debe decidir si el que recibe el control de la CPU es el proceso en cuestión u otro proceso.

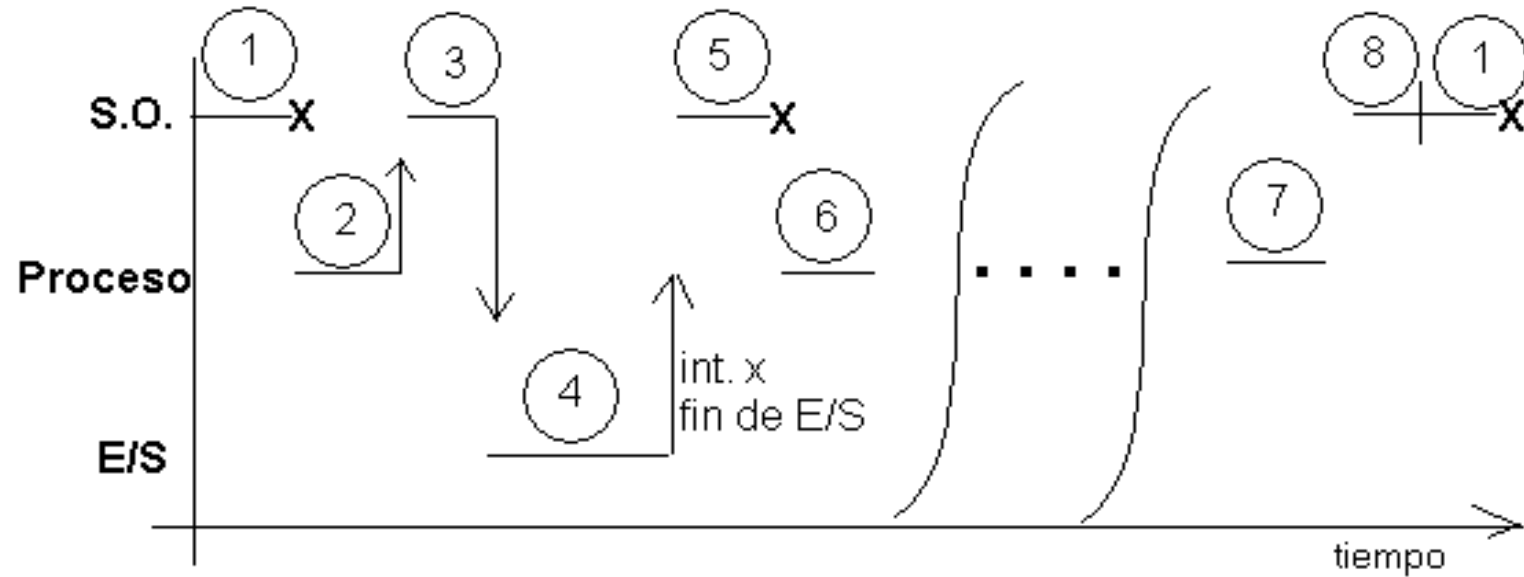
## Y el turnaround?

Si calculo el turnaround de un proceso solo en un ambiente de monoprogramación y luego lo calculo colocando ese proceso (también solo) en un ambiente de multiprogramación concluimos que:

Tiempo en Monoprogramación < Multiprogramación

Por que se agrega el tiempo del Planificador de Procesos!!

## Rutinas del SO para administrar procesos



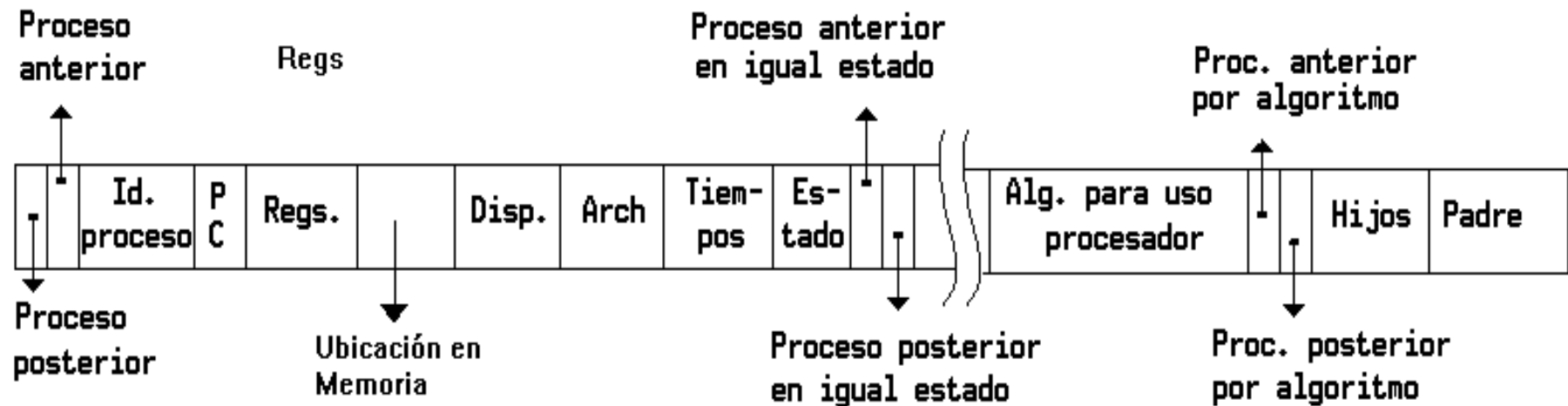
1) y 8) Planificador de Trabajos

3) Rutina de lanzamiento de E/S

5) Rutina de atención de interrupciones por fin de E/S

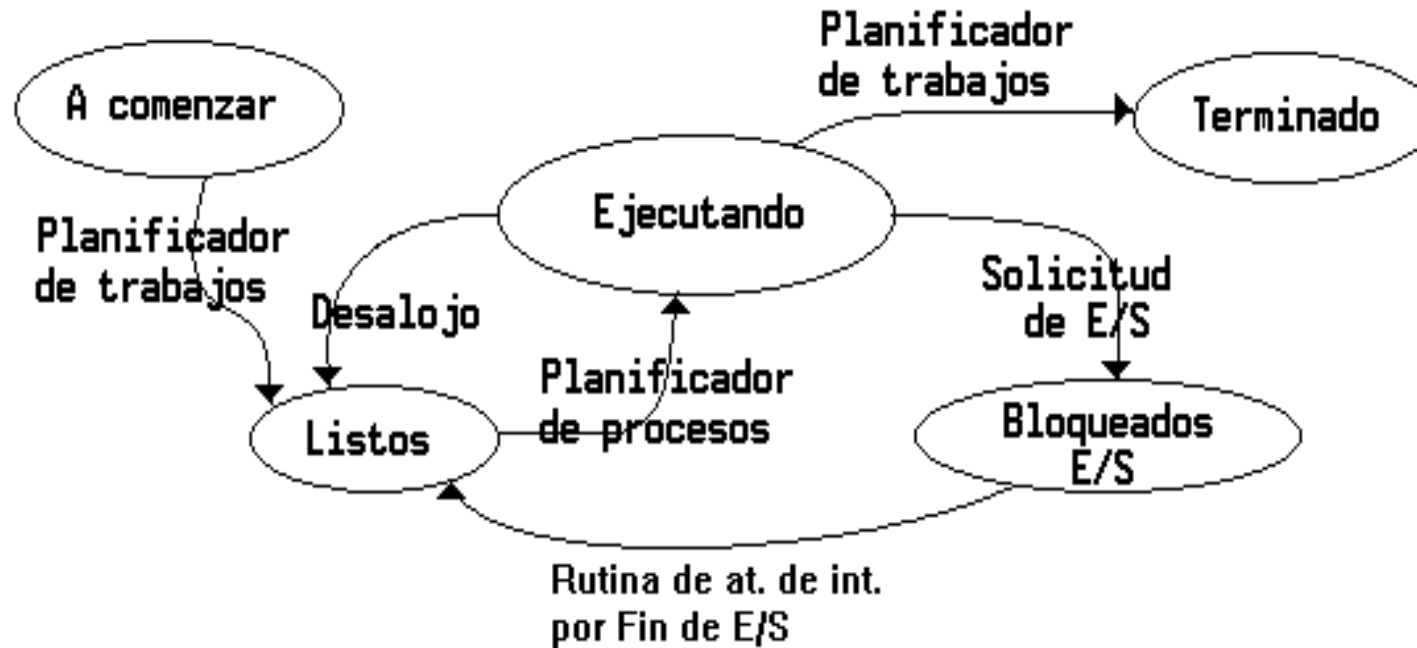
Para administrar los procesos necesitamos identificar a cada uno de ellos:

## Bloque de Control de Proceso (BCP)



Los BCP se agrupan en la Tabla de Bloques de Control de Procesos (TBCP). Esta tabla tiene encadenamientos entre los procesos según su estado.

## Diagrama de transición de estados



Burbujas: Estados

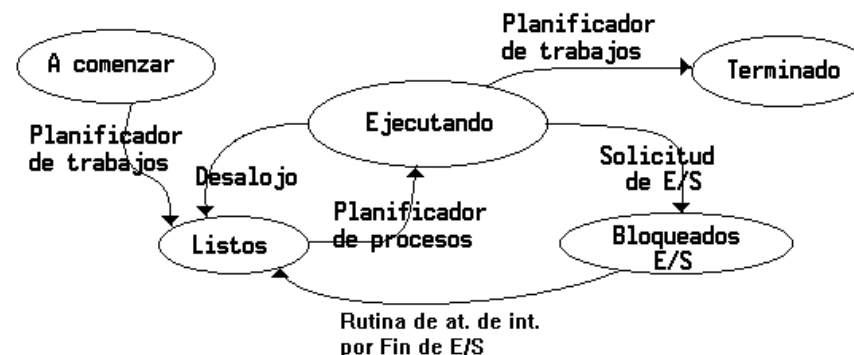
Flechas: Transiciones entre estados

## Eventos y rutinas que intervienen (1)

### A Comenzar ---> Listos

Evento: fin de E/S del periférico por el cual ingresan los procesos al sistema. (en un sistema interactivo sería el enter despachando un comando u orden al sistema)

Interviene: Planificador de Trabajos

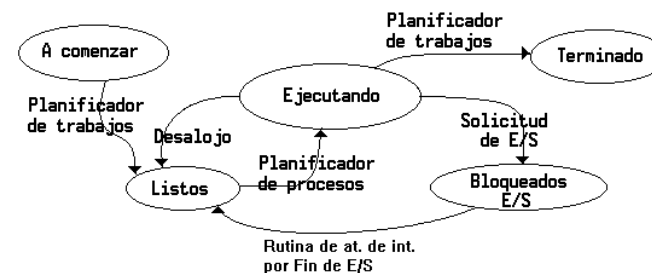


## Eventos y rutinas que intervienen (2)

### Listos ---> Ejecutando

Evento: La CPU se acaba de liberar porque la abandonó algún proceso

Rutina: el Planificador de Procesos selecciona al siguiente para ejecutar. El ultimo paso del P.P. cuando el S.O. entrega el control al proceso lo realiza el **Dispatcher**.

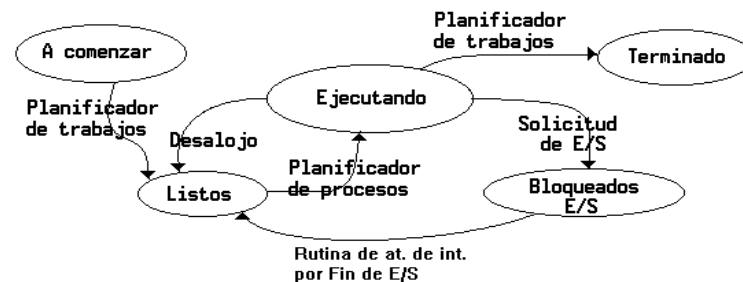


## Eventos y rutinas que intervienen (3)

### Ejecutando --> Listos

Evento: El proceso que está ejecutando pierde el control de la CPU (es desalojado). Puede ser porque excedió su tiempo, por fin de E/S (de otro proceso), por ingreso al sistema de un proceso con mayor prioridad, etc.

Rutina: la rutina de desalojo que corresponda, por ejemplo desalojo por reloj.



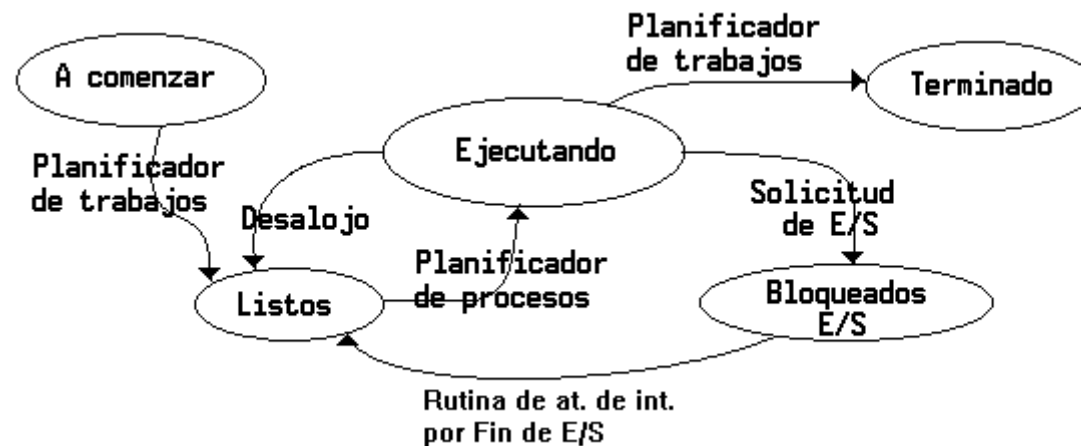


## Eventos y rutinas que intervienen (4)

Ejecutando --> Bloqueado por espera de fin de E/S

Evento: El proceso solicitó realizar una E/S

Rutina: rutina de lanzamiento de E/S

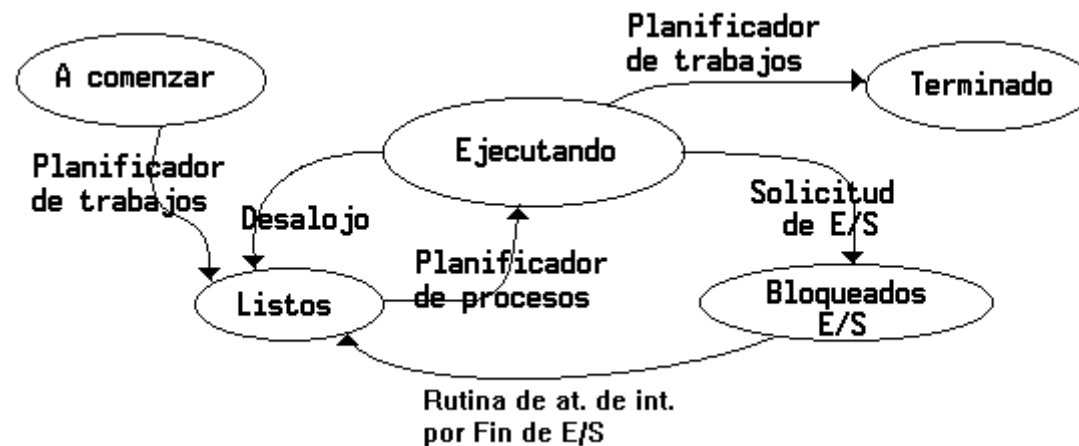


## Eventos y rutinas que intervienen (5)

### Bloqueado por espera de fin de E/S --> Listos

Evento: se produjo una interrupción por fin de E/S

Rutina: rutina de atención de interrupciones por fin de E/S

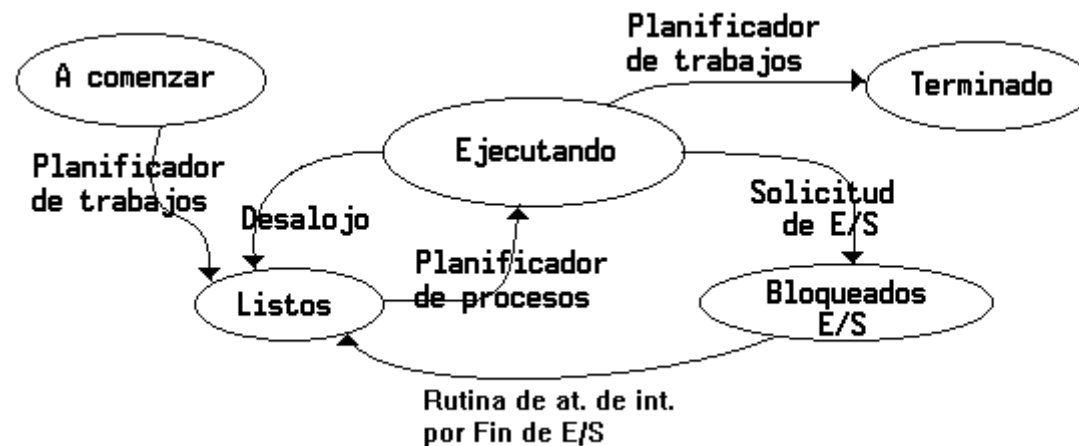


## Eventos y rutinas que intervienen (6)

### Ejecutando --> Terminado

Evento: el proceso finaliza su ejecución por fin normal o anormal

Rutina: Planificador de Trabajos



## Eventos y rutinas que intervienen (7)

### Todo cambio de estado

En todo cambio de estado interviene el Controlador de Tráfico, que es la rutina que se encarga de acceder la TBCP y modificar los estados de los procesos

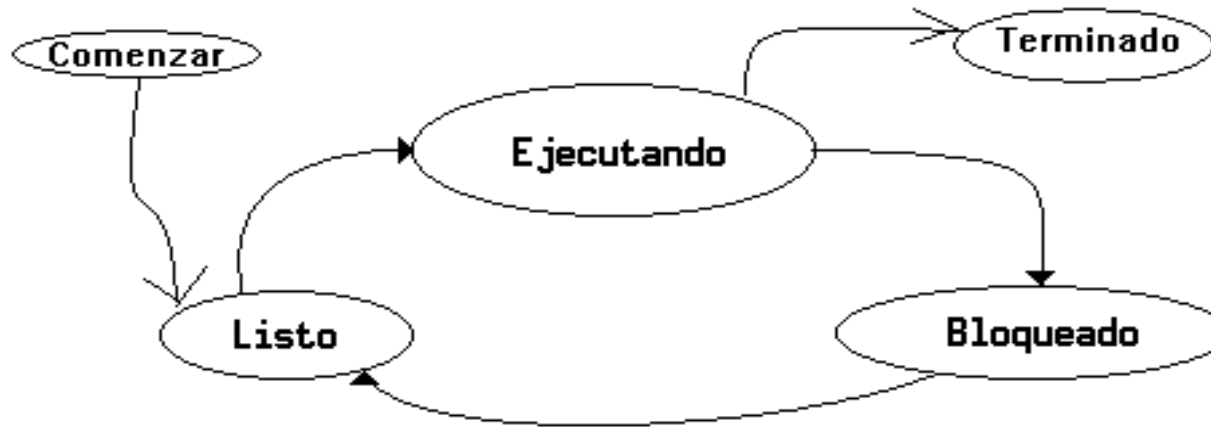
The diagram shows a table with two columns: 'Id. Proceso' and 'Estado'. The rows are: A (Ejec.), B (Listo), C (Bloq.), and D (Listo). A pointer labeled 'Puntero al primero de los Listos' points to row B. A vertical line to the right of the 'Listo' states (rows B and D) has an arrow pointing from row B down to row D, indicating a linked list of ready processes.

Id. Proceso	Estado
A	Ejec.
B	Listo
C	Bloq.
D	Listo

## Administraciones del Procesador

Lo más importante es el algoritmo de decisión implementado en el Planificador de Procesos ya que cambiando el algoritmo el sistema tiene un comportamiento y rendimiento diferente.

FIFO: Primero que llega primero que sale

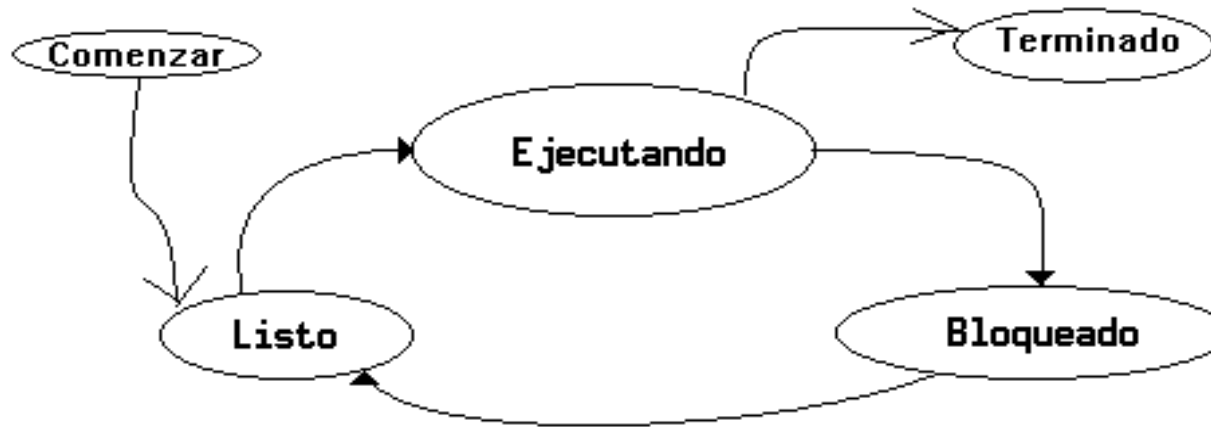


Un proceso ejecuta y solo abandona el control de la CPU cuando lanza E/S o termina.

La cola de listos se ordena FIFO

Ante un fin de E/S (de otro proceso, no el que está ejecutando) se atiende el fin de E/S y luego se continúa ejecutando al proceso interrumpido.

## Más corto primero sin desalojo



Los procesos en la cola de listos se ordenan según la duración de sus ráfagas (tiempo de uso continuo de la CPU).

Provee el mejor turnaround.

Problema: Procesos largos pueden no llegar a terminar nunca porque no logran obtener la CPU (inanición)

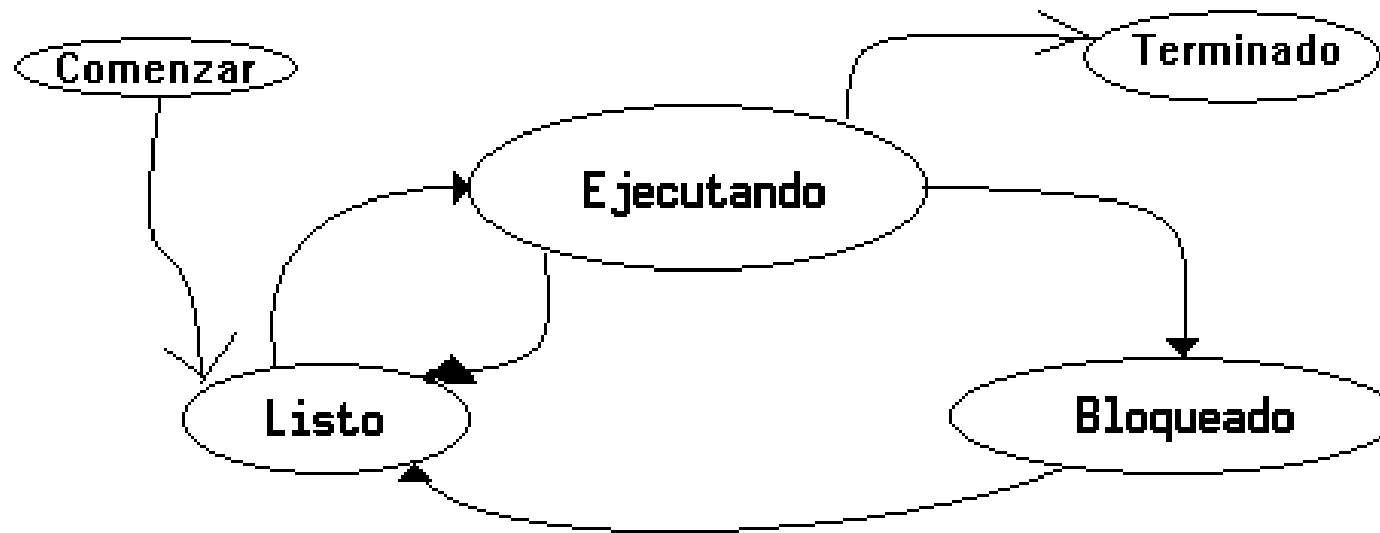
## Cómo calcular la ráfaga de un proceso cuando aún no se ejecutó?

- 1°) Asigno un valor estimado  $\alpha$  a la primer ráfaga  $T_0$
- 2°) Luego de ejecutar esa ráfaga ajusto con un valor  $\delta$  para la próxima ráfaga  $T_1$
- 3°) y así siguiendo se ajusta la ráfaga futura en base a las mediciones de las ráfagas anteriores del proceso

A fines de los ejercicios prácticos consideraremos el tiempo total de CPU para estimar cuál es el proceso más corto.

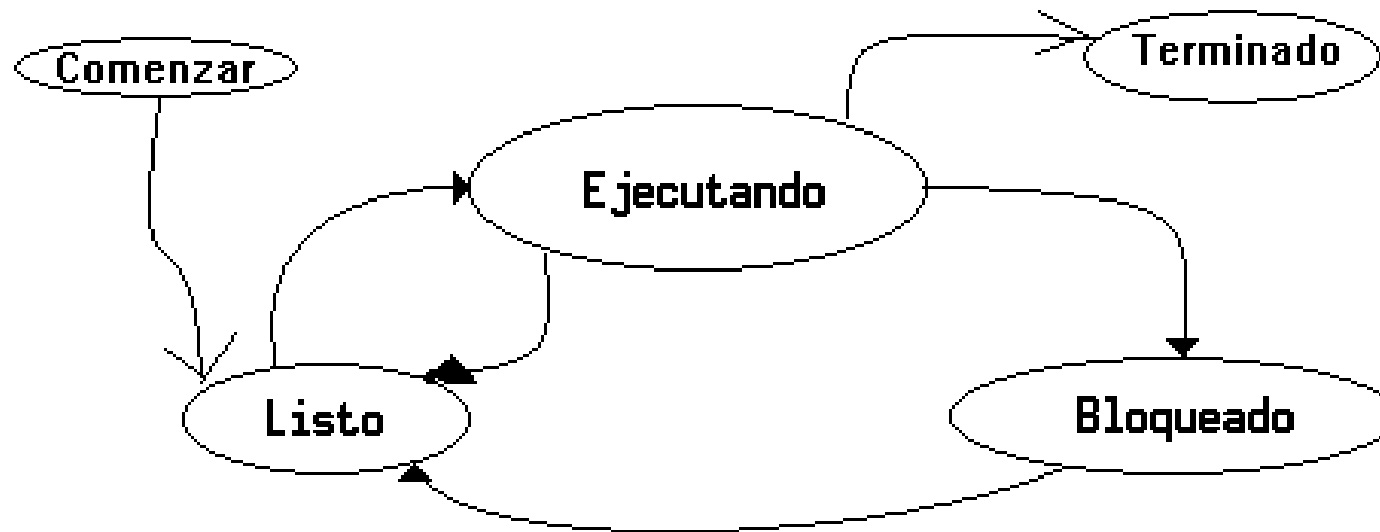


## Más corto primero con desalojo



Ingresa un proceso en la cola de listos con menor tiempo que el que está en control de la CPU y lo desaloja

## Más corto primero con desalojo por tiempo remanente



Ingresa un proceso en la cola de listos con **menor tiempo que el tiempo remanente** del que está en control de la CPU y lo desaloja

## Turnaround.

La administración Mas Corto Primero en todas sus formas es la que provee el mejor turnaround de todas las administraciones, sin embargo tiene un alto overhead.

Overhead: es el tiempo que el sistema operativo consume en ejecutar sus propias rutinas. A mayor overhead menor performance del sistema ya que no hay respuesta a los procesos de los usuarios.

## Prioridades

Cada proceso tiene asignada una prioridad de ejecución y se seleccionan de la cola de listos en base a esa prioridad.

La cola de listos esta ordenada por la prioridad.

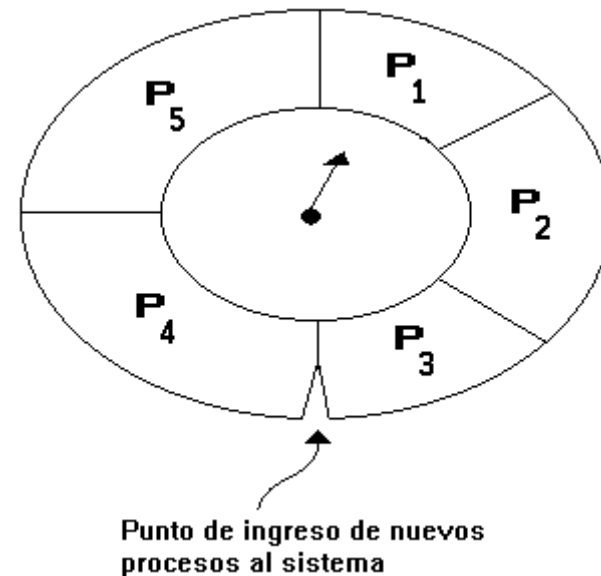
Puede tener desalojo por haber llegado un proceso a Listos de menor prioridad.

## Round-Robin

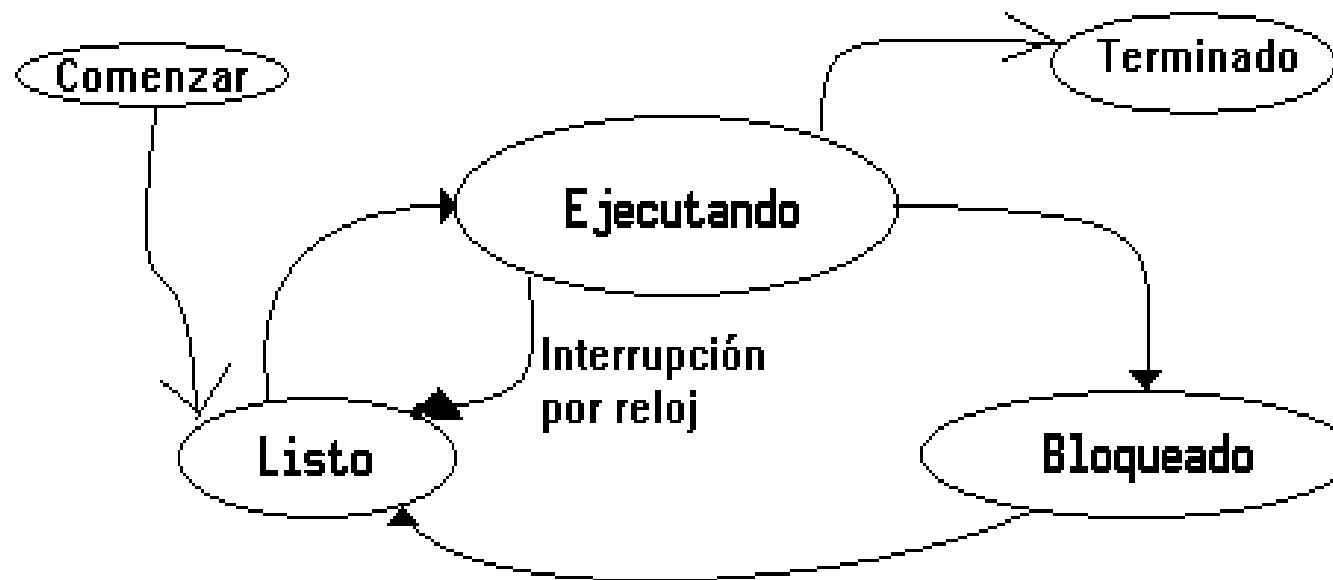
Todos los procesos independientemente de su estado están en una calesita y se van seleccionando avanzando un apuntador que se mueve en el sentido de las agujas del reloj.

A cada proceso seleccionado se le otorga una cantidad de tiempo fijo de uso de la CPU (**quantum**).

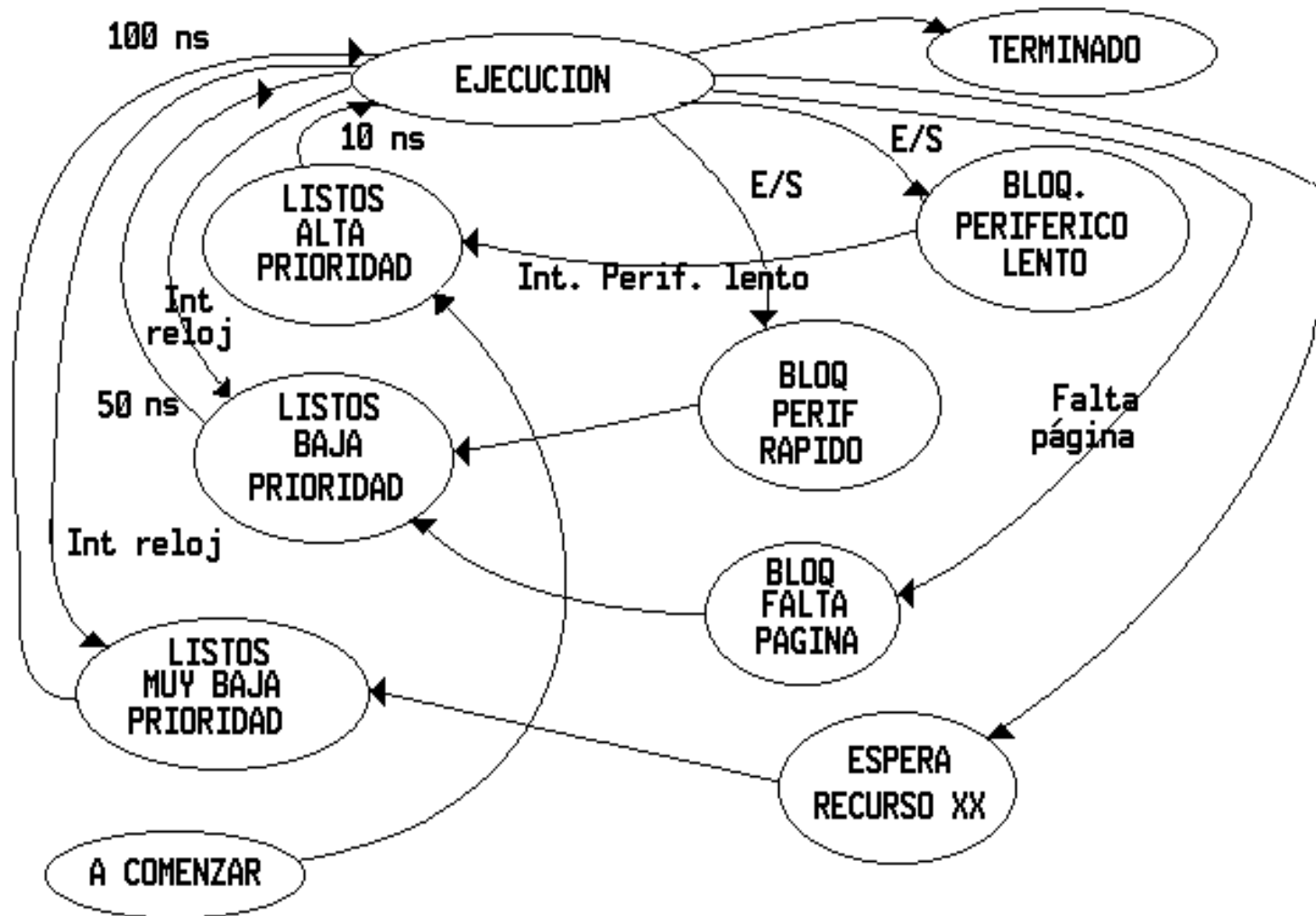
Una vez agotado el quantum se produce una interrupción por reloj y el proceso es desalojado a la cola de Listos.



## Round-Robin (continuación)



## Multicolos



## Multicolos (continuación)

Hay colas de Listos de diferente prioridad y se selecciona primero de la de > prioridad y si está vacía de la siguiente y así siguiendo.

Los estados de bloqueados se diferencian según la velocidad del periférico.

Se balancea el uso de la CPU mediante quantums asignados a cada cola de Listos.

Se balancea el uso de los periféricos según el tiempo que el proceso tarda en estar en cola de Listos nuevamente (periféricos rápidos tarda poco en volver, periféricos lentos tarda mucho en volver)