

# Sistemas Operativos

---

UNIDAD 3

PLANIFICACIÓN DE PROCESOS

# Multiprogramación y Planificación

Sistema multiprogramado: varios procesos

La clave de la multiprogramación es la planificación de CPU para mejorar:

- Tiempo de respuesta.
- Productividad.
- Eficiencia del procesador.

Elementos del núcleo del S.O. relacionados

- CPU Scheduler: Planificador
- Dispatcher: Despachador
- Interrupt Handler: Manejador de Interrupciones

**Tabla 9.1.** Tipos de planificación.

Frecuencia de ejecución ↑ ↓ +	Planificación a largo plazo	La decisión de añadir un proceso al conjunto de procesos a ser ejecutados
	Planificación a medio plazo	La decisión de añadir un proceso al número de procesos que están parcialmente o totalmente en la memoria principal
	Planificación a corto plazo	La decisión por la que un proceso disponible será ejecutado por el procesador
	Planificación de la E/S	La decisión por la que un proceso que está pendiente de una petición de E/S será atendido por un dispositivo de E/S disponible

# Estados y planificadores

Grafo transiciones entre estados y niveles de planificación:

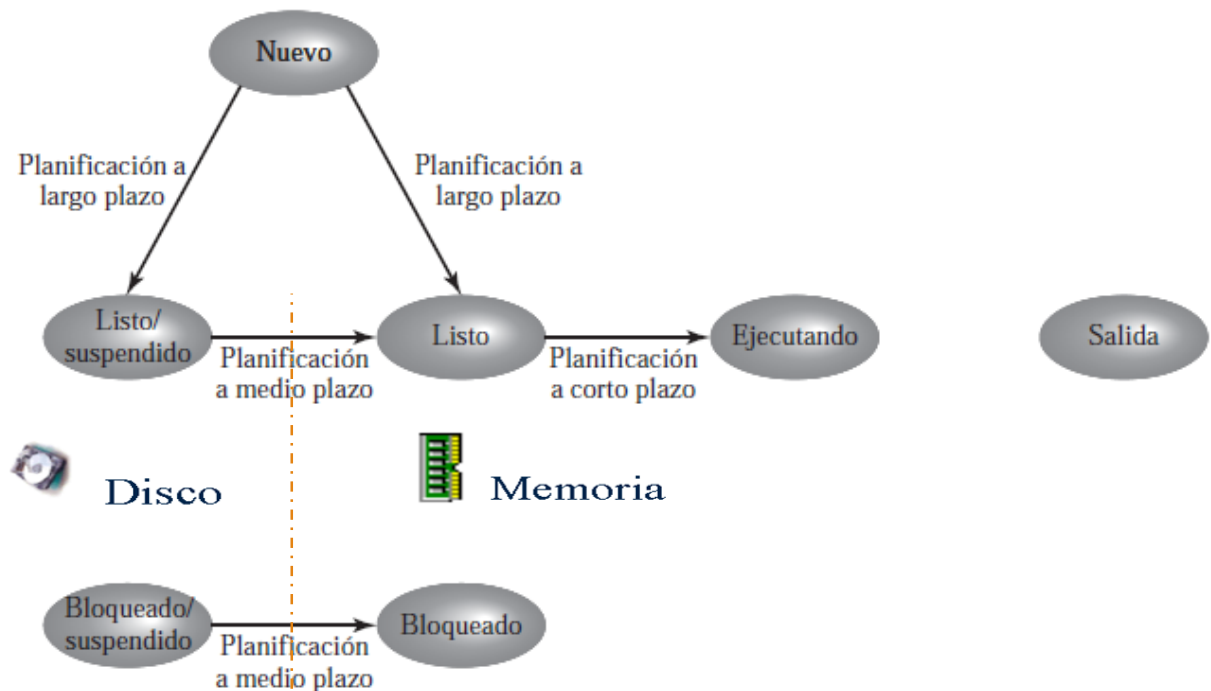


Figura 9.1. Planificación y transiciones de estado de los procesos.

El planificador a medio plazo es parte de la función de intercambio (swapping function)

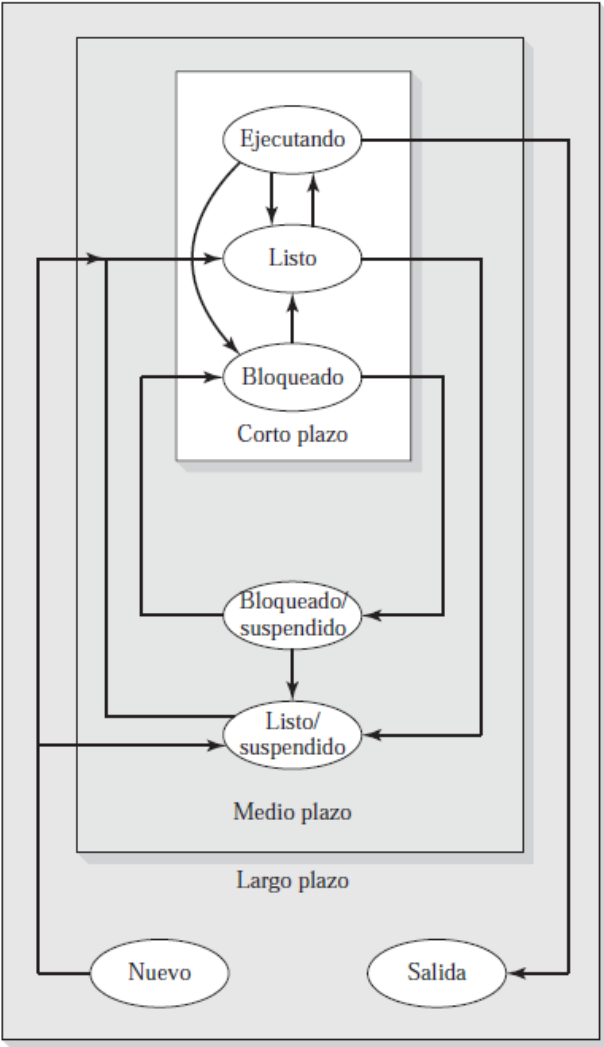
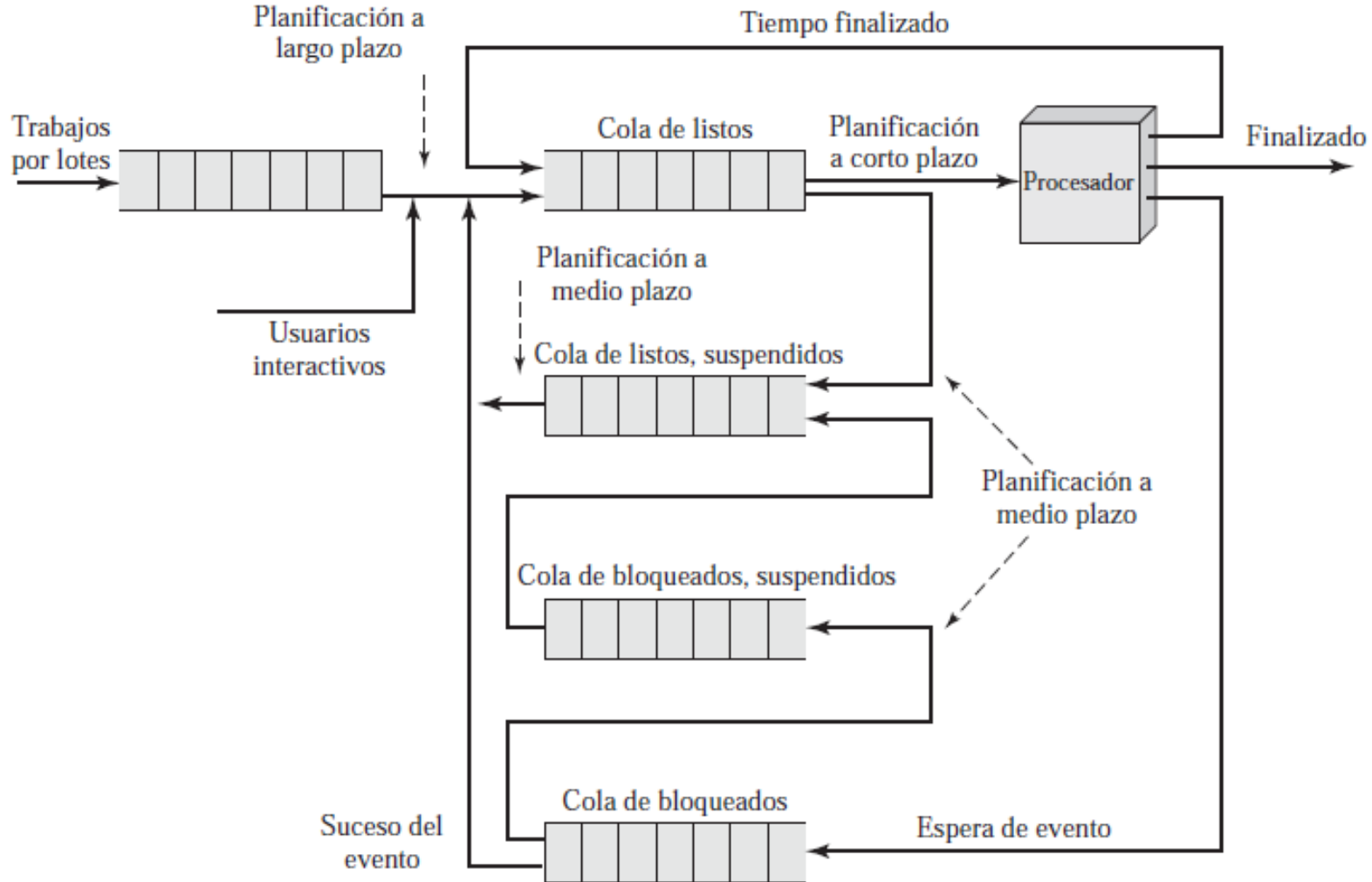


Figura 9.2. Niveles de planificación.

# Modelo de colas de planificación



**Figura 9.3.** Diagrama de encolamiento para la planificación.

# Planificador a Largo Plazo

## Objetivos de la planificación a largo plazo:

- Controlar el grado de multiprogramación mediante la admisión de nuevos programas en:
  - Listo (cola de corto plazo) Memoria
  - Listo Suspendido (cola de medio plazo) Zona de Intercambio

## Sistemas de proceso por Lotes (o parte Lotes Sistema Generalista)

- Los procesos se incorporan a la cola de procesos por lotes en disco.
- De la cola de proceso por lotes a Listo. Decisiones:
  - Si el SO puede acoger a más procesos en Listo
    - Control del Grado de Multiprogramación
  - Decidir cual será el elegido de la cola de lotes.
    - Por un algoritmo de planificación
    - Por una herramienta de rendimiento
    - Por planificación de E/S

- Si se admiten muchos trabajos se reduce el tiempo de servicio y los procesos compiten por el tiempo
- Si el procesador está mucho tiempo ocioso se llama al planificador a largo plazo.

## Sistemas Interactivos

- Cuando un usuario intenta conectarse se crea una solicitud al planificador a largo plazo.
- Se aceptan todas hasta un límite. Se informa de sistema saturado.

# Planificadores medio y corto plazo

## La Planificación a medio plazo

- Forma parte de la función de intercambio.

## Planificación a Corto Plazo.

- Objetivos:
  - Justicia: todos los procesos sean tratados por igual
  - Optimizar el uso de recursos: Uso del procesador y operaciones E/S
  - Evitar la sobrecarga del sistema
  - Evitar inanición de procesos
  - Evitar la degradación paulatina
- Se ejecuta siempre que un proceso abandona la CPU, por:
  - Interrupción de reloj.
  - Interrupción de E/S
  - Llamadas al Sistema Operativo
  - Señales

# Criterios de planificación a corto plazo

## Orientados al Usuario

### Prestaciones

**Tiempo de estancia**

Desde que se lanza hasta que finaliza.  
Trabajos por lotes.

**Tiempo de respuesta**

Desde que se lanza la petición hasta que se obtienen resultados.  
Sistemas Interactivos.

**Fecha Tope**

Si se especifica hay que maximizar el porcentaje de deadlines conseguidos

### Otros

**Previsibilidad**

Un proceso debería tardar aprox. siempre lo mismo y con el mismo coste, independientemente de la carga del sistema.

## Orientados al Sistema

### Prestaciones

**Rendimiento**

Numero de procesos completados por unidad de tiempo. Mide el trabajo que esta siendo realizado.

**Utilización del procesador**

Porcentaje de tiempo que el procesador está ocupado

### Otros

**Equidad**

Los procesos deben de ser tratados de igual forma, ninguno de be sufrir de inanición

**Imposición de Prioridades**

Si se determinan prioridades el planificador debe favorecer a los procesos en función de su prioridad

**Equilibrado de recursos**

Se debe mantener ocupados los recursos del sistema. Favorecer procesos con poca carga de recursos cuando estos están muy ocupados.

# Criterios de planificación a corto plazo

## Criterios de planificación (corto plazo):

- Utilización de CPU
  - % de tiempo que la CPU está ocupada
  - Maximizar
- Productividad (throughput)
  - Nº procesos terminados por unidad de tiempo
  - Maximizar
- Tiempo de retorno
  - Intervalo entre el lanzamiento de un proceso y su finalización
  - Suma del tiempo de ejecución real + tiempo consumido en espera por recursos (inc. Procesador)
  - Minimizar
- Tiempo de respuesta
  - Intervalo de tiempo transcurrido desde que se emite una solicitud hasta que se comienza a recibir respuesta
  - Minimizar
- Tiempo de espera
  - Total del tiempo que el proceso está esperando por algún recurso incluido el procesador
  - Minimizar
- Para comparar los distintos algoritmos:
  - media
  - varianza

Ver:

[Planificación de procesos: Evaluación del rendimiento](#)

Algunos objetivos son incompatibles entre si, por lo que nos centraremos en optimizar el que más nos interese dependiendo del tipo de sistema:

- Sistemas interactivos: tiempo de respuesta
- Sistemas por lotes (batch): tiempo de retorno

Los Algoritmos se pueden clasificar por grupos en función de la forma de abandonar la CPU:

- Sin reciclaje de procesos (no preemptiva).  
Abandono no forzoso.
- Con reciclaje de procesos (preemptiva).  
Abandono forzoso



# Algoritmos de Planificación de Procesos

La decisión de planificación puede ocurrir:

- a) Cuando un proceso pasa de ejecución a espera
- b) Cuando un proceso termina
- c) Cuando un proceso pasa de ejecución a listo
- d) Cuando un proceso pasa de espera a listo
- e) Cuando llega una interrupción
- f) Periódicamente

Los Algoritmos se pueden clasificar por grupos en función de la forma de abandonar la CPU:

- **Sin expulsión (*nonpreemptive*)**. En este caso, una vez que el proceso está en el estado Ejecutando, continúa ejecutando hasta que (a) termina o (b) se bloquea para esperar E/S o para solicitar algún servicio al sistema operativo.
- **Con expulsión (*preemptive*)**. Un proceso ejecutando en un determinado momento puede ser interrumpido y pasado al estado de listo por el sistema operativo. La decisión de expulsar puede ser tomada cuando llega un nuevo proceso, cuando llega una interrupción que pasa un proceso de bloqueado a estado de listo, o periódicamente, basándose en las interrupciones del reloj.

# Algoritmos de Planificación de Procesos

**Tabla 9.3.** Características de algunas políticas de planificación.

	Función de Selección	Modo de Decisión	Rendimiento	Tiempo de Respuesta	Rendimiento	Efecto sobre los Procesos	Inanición
FCFS	$\max[w]$	No expulsiva	No especificado	Puede ser alto especialmente si hay mucha diferencia entre los tiempos de ejecución de los procesos	Mínima	Penaliza procesos cortos; penaliza procesos con mucha E/S	No
Turno Rotatorio (round robin)	constante	Expulsiva (por rodajas de tiempo)	Puede ser mucho si la rodaja es demasiado pequeña	Proporciona buen tiempo de respuesta para procesos cortos	Mínima	Tratamiento justo	No
SPN	$\min[s]$	No expulsiva	Alto	Proporciona buen tiempo de respuesta para procesos cortos	Puede ser alta	Penaliza procesos largos	Posible
SRT	$\min[s-e]$	Expulsiva (a la llegada)	Alto	Proporciona buen tiempo de respuesta	Puede ser alta	Penaliza procesos largos	Posible
HRRN	$\max(w+s/s)$	No expulsiva	Alto	Proporciona buen tiempo de respuesta	Puede ser alta	Buen equilibrio	No
Feedback	(ver texto)	Expulsiva (por rodajas de tiempo)	No especificado	No especificado	Puede ser alta	Puede favorecer procesos con mucha E/S	Posible

$w$  = tiempo usado en el sistema hasta este momento, esperando o ejecutando

$e$  = tiempo usado en ejecución hasta este momento

$s$  = tiempo total de servicio requerido por el proceso, incluyendo  $e$ ; generalmente, esta cantidad debe ser estimada o proporcionada por el usuario

# Planificación de procesos - Algoritmos

## 1. FCFS (First Come, First Served):

- Política **no preemptiva**
- La cola de preparados es FIFO
- Abandona la CPU cuando:
  - Realiza una llamada al sistema (operación de E/S, ..)
  - Termina
- Cuando el proceso actual cesa su ejecución, se selecciona el proceso más antiguo de la cola.

### Ventajas:

- No hay posposiciones indefinidas (inanición)

### Inconvenientes:

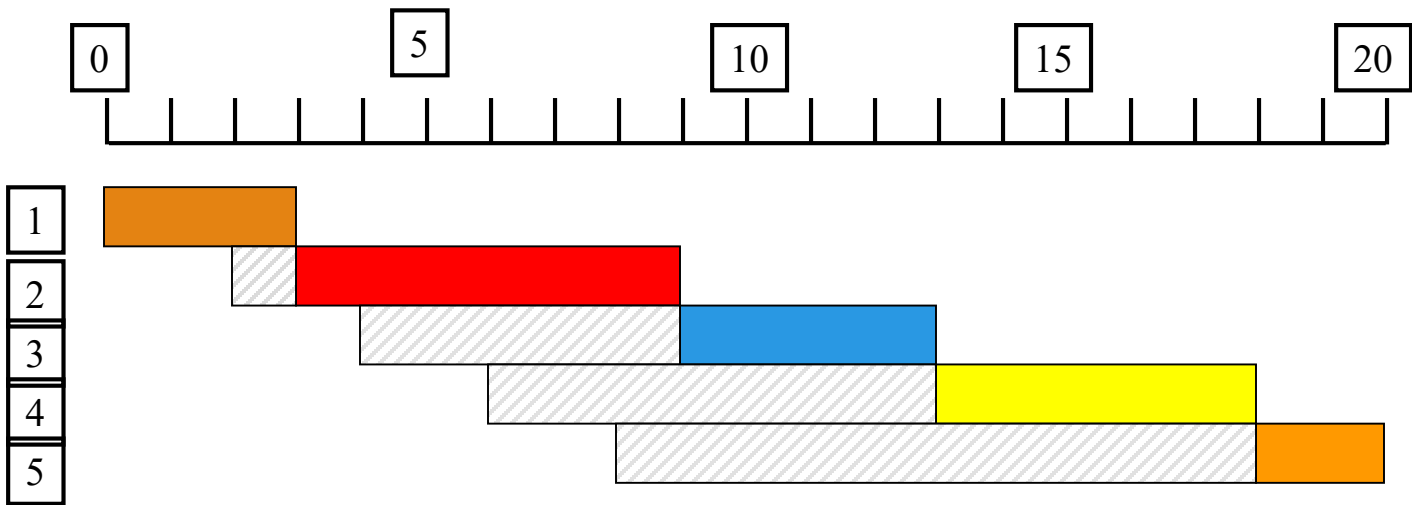
- Favorece a los procesos largos: puede que un proceso corto tenga que esperar mucho tiempo
- Favorece a los procesos con carga de CPU: los procesos con carga de E/S tienen que esperar a que se completen los procesos con carga de CPU.
- No son válidos para sistemas de tiempo compartido (multiusuario)

# Planificación de procesos - Algoritmos

## 1. FCFS:

Proceso	Instante de llegada	Tiempo de servicio
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

 Preparado



# Planificación de procesos - Algoritmos

Procesos	A	B	C	D	E	F	G	H	I	J	Media
Tiempo de Llegada	0	0	0								
Orden de Llegada	1	2	3								
Tiempo de Ejecucion	24	3	2								
Tiempo Max. Permitido											
Tiempo Critico											
Prioridad											
Tiempo de Retorno	24	27	29								26
Tiempo de Espera	0	24	27								17
Tiempo de Respuesta	0	24	27								17
Eficiencia	100	11	6								39

Desv. Est.  
2,05  
12,08  
12,08

Estados

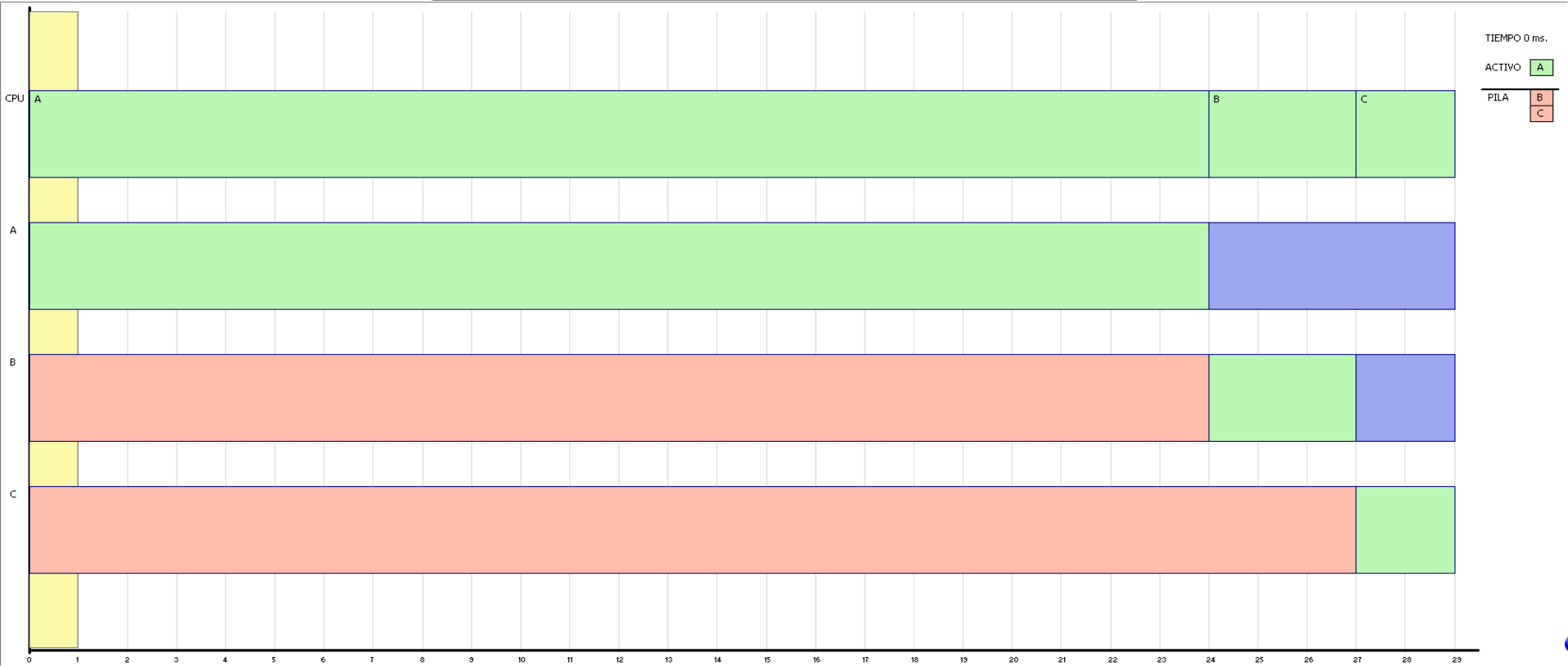
Inactivo

Preparado

Activo

Suspendido

Finalizado



# Planificación de procesos - Algoritmos

Procesos	A	B	C	D	E	F	G	H	I	J	Media
Tiempo de Llegada	0	0	0								
Orden de Llegada	3	1	2								
Tiempo de Ejecucion	24	3	2								
Tiempo Max. Permitido											
Tiempo Critico											
Prioridad											
Tiempo de Retorno	29	3	5								12
Tiempo de Espera	5	0	3								2
Tiempo de Respuesta	5	0	3								2
Eficiencia	82	100	40								74

Desv. Est.  
11,81  
2,05  
2,05

Estados

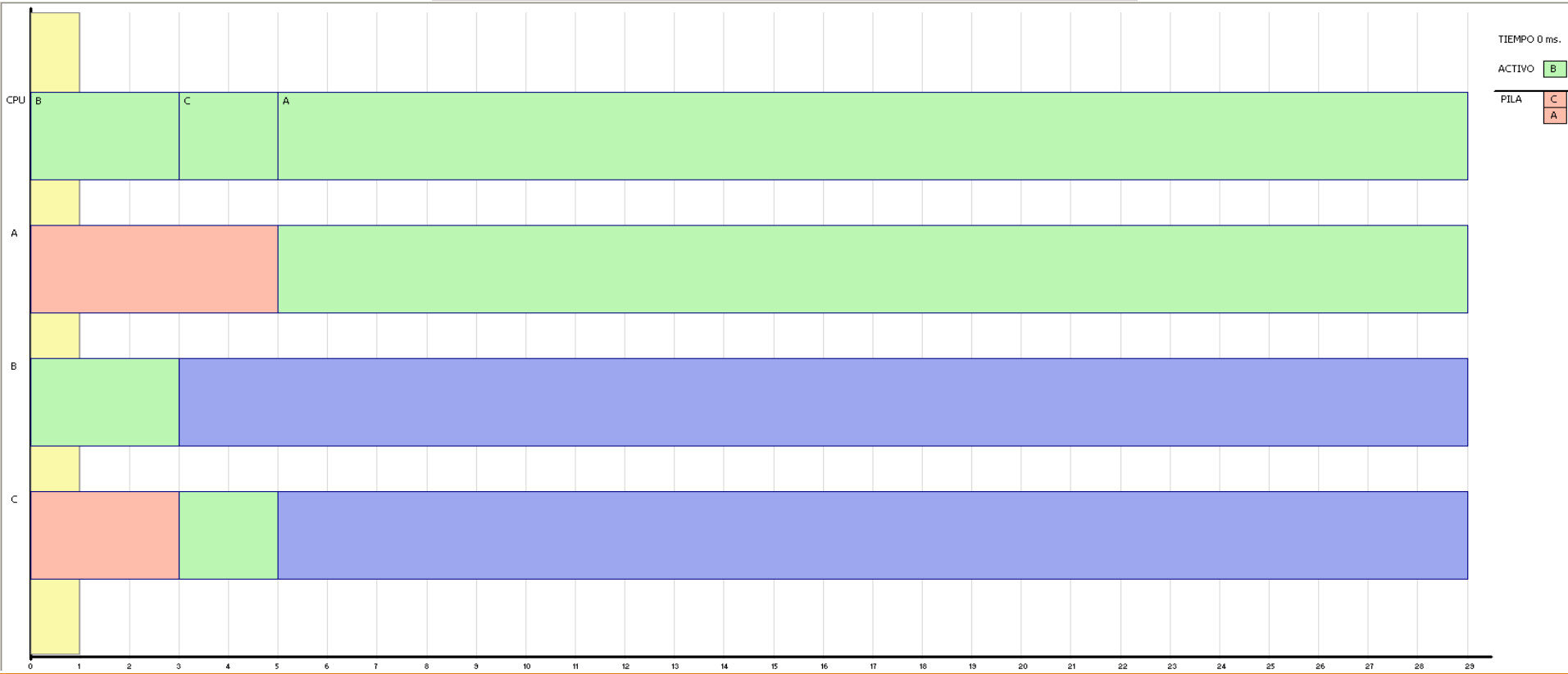
Inactivo

Preparado

Activo

Suspendido

Finalizado



# Planificación de procesos - Algoritmos

## 2. SPN (Shortest Process Next):

- Política **no preemptiva**
- Abandona la CPU cuando:
  - Realiza una llamada al sistema (operación de E/S, ..)
  - Termina
- Se selecciona el proceso con menor tiempo esperado de ejecución.
- Un proceso corto saltará a la cabeza de la cola, sobrepasando a trabajos largos.

### Ventajas:

- Siempre obtendremos el mínimo tiempo medio de respuesta

### Inconvenientes:

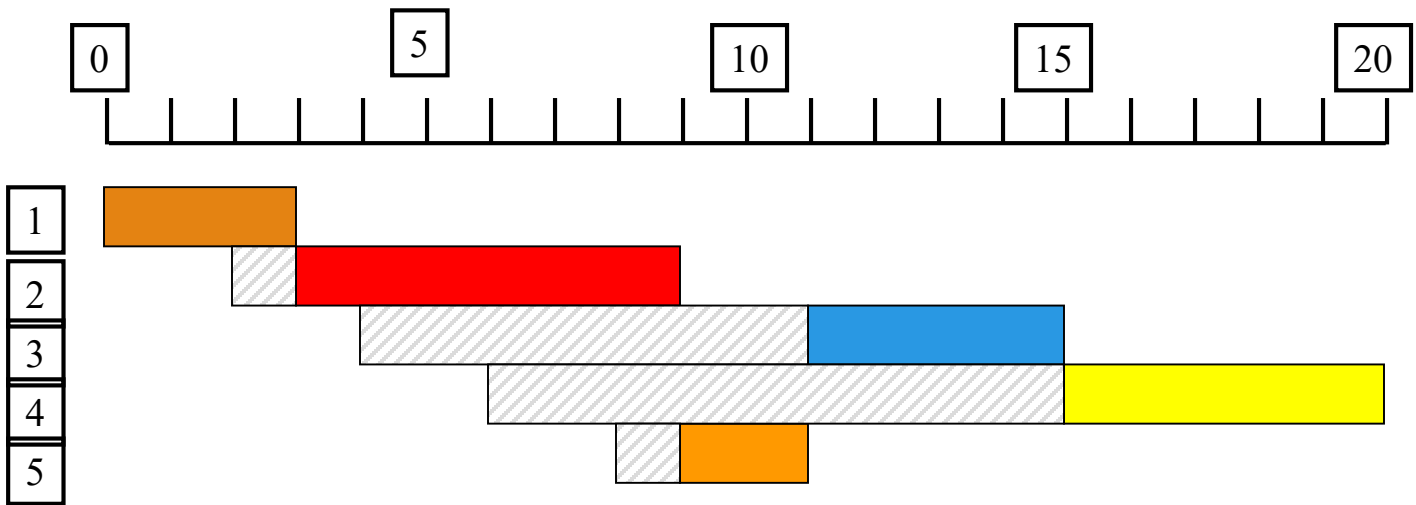
- Los procesos largos se penalizan (inanición)
- Casi nunca es posible conocer de antemano el tiempo de ejecución de cada proceso, por lo que se recurre a estimaciones de ese tiempo.

# Planificación de procesos - Algoritmos

2. SPN:

Proceso	Instante de llegada	Tiempo de servicio
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

 Preparado





# Planificación de procesos - Algoritmos

Procesos	A	B	C	D	E	F	G	H	I	J	Media
Tiempo de Llegada	0	0	0								
Orden de Llegada	1	2	3								
Tiempo de Ejecucion	24	3	2								
Tiempo Max. Permitido											
Tiempo Critico											
Prioridad											
Tiempo de Retorno	29	5	2								12
Tiempo de Espera	5	2	0								2
Tiempo de Respuesta	5	2	0								2
Eficiencia	82	60	100								80

Desv. Est.  
12,08  
2,05  
2,05

Estados

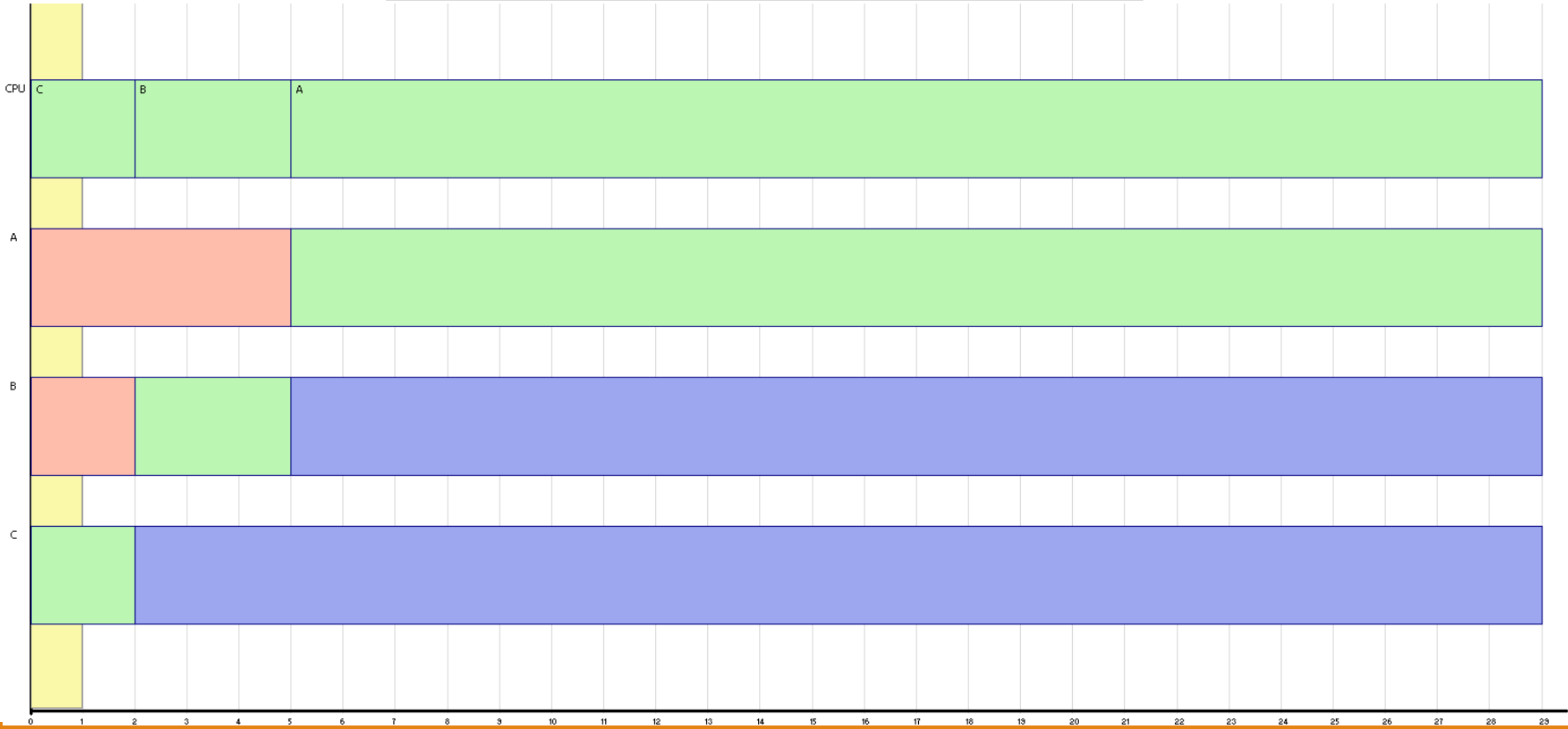
Inactivo

Preparado

Activo

Suspendido

Finalizado



# Planificación de procesos - Algoritmos

## 3. Uso de prioridades:

- Política **no preemptiva**
- Tiene múltiples colas de “preparados” para representar cada nivel de prioridad
- Un proceso no se ejecutará mientras haya procesos de mayor prioridad esperando
- Abandona la CPU cuando:
  - Realiza una llamada al sistema (operación de E/S, ..)
  - Termina
- Los procesos de prioridad más baja pueden sufrir inanición:
  - Permite que un proceso cambie su prioridad en función de su edad o su historial de ejecución.

# Planificación de procesos - Algoritmos

## 4. RR (Round Robin) o turno rotatorio:

- Política **apropiativa** (preeemptiva)
- Asignación de intervalo de tiempo: quantum
- Es el más sencillo y justo
- La cola de preparados se gestiona como si fuera una cola FIFO circular.
- Habiendo  $n$  procesos, un proceso no esperará mas de  $(n-1)q$  unidades de tiempo, siendo  $q$  la duración del quantum.
- Abandona la CPU cuando:
  - Realiza una llamada al sistema (operación de E/S, ..)
  - Termina
  - Agotamiento de quantum
- El quantum puede ser fijo o variable durante la vida de un proceso.
- Hay que buscar el equilibrio en la duración del quantum

### Ventajas:

- No hay posposiciones indefinidas (inanición)
- Especialmente diseñado para sistemas de tiempo compartido (multiusuario)

### Inconvenientes:

- Rendimiento pobre de los procesos que realizan muchas operaciones de E/S, con el consiguiente desaprovechamiento de los dispositivos de E/S.

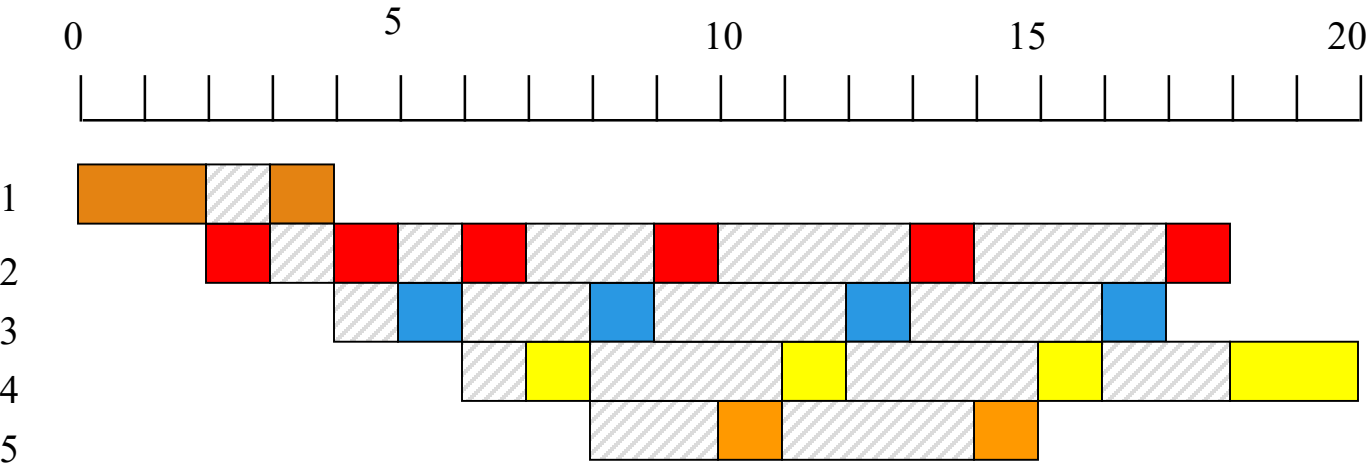
¡ El valor del quantum es muy importante !

# Planificación de procesos - Algoritmos

RR (q=1):

Proceso	Instante de llegada	Tiempo de servicio
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

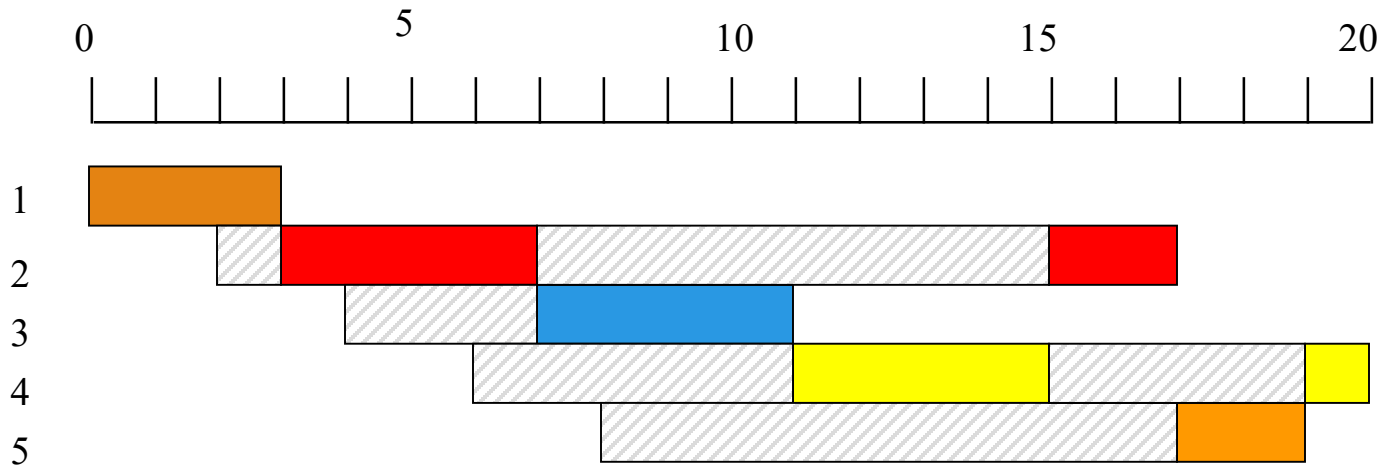
 Preparado



# Planificación de procesos - Algoritmos

RR (q=4):

Proceso	Instante de llegada	Tiempo de servicio
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2



# Planificación de procesos - Algoritmos

## EJERCICIO

\* SIMULACION RR (q=4) \*

Proceso	Instante llegada	Tiempo servicio
=====	=====	=====
P1	0	24
P2	2	3
P3	4	2

\* SIMULACION RR (q=1) \*

Proceso	Instante llegada	Tiempo servicio
=====	=====	=====
P1	0	24
P2	2	3
P3	4	2

# Planificación de procesos - Algoritmos

## 5. SRTN (Shortest Remaining Time Next):

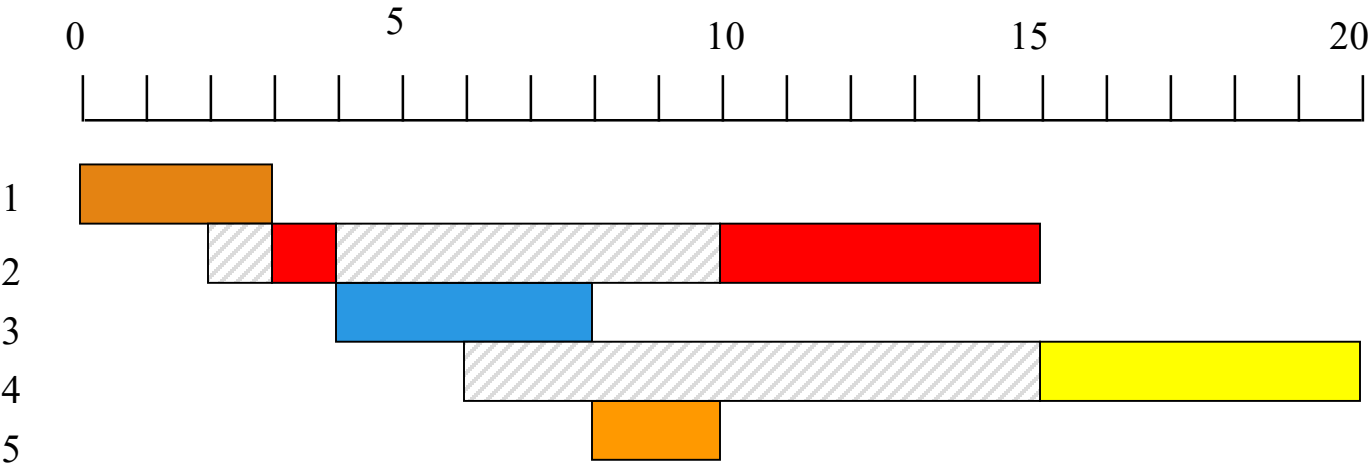
- Es una versión **apropiativa** de la política de SPN (primero el proceso más corto).
- Debe estimar el tiempo de proceso.
- Abandona la CPU cuando:
  - Realiza una llamada al sistema (operación de E/S, ..)
  - Termina
  - Al ser apropiativo, si llega un proceso más corto que el que actualmente está en ejecución se selecciona.

# Planificación de procesos - Algoritmos

SRTN:

Proceso	Instante de llegada	Tiempo de servicio
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

 Preparado





# Planificación de procesos - Algoritmos

## 6. Uso de prioridades:

- Prioridades apropiativo
- Abandona la CPU cuando:
  - Realiza una llamada al sistema (operación de E/S, ..)
  - Termina
  - Al ser apropiativo, si llega un proceso con mayor prioridad que el que actualmente está en ejecución. Se asignaría el procesador directamente al nuevo proceso, en vez de esperar el primero en la cola de preparados, tal y como haría el algoritmo por prioridades no apropiativo.

# Planificación de procesos - Algoritmos

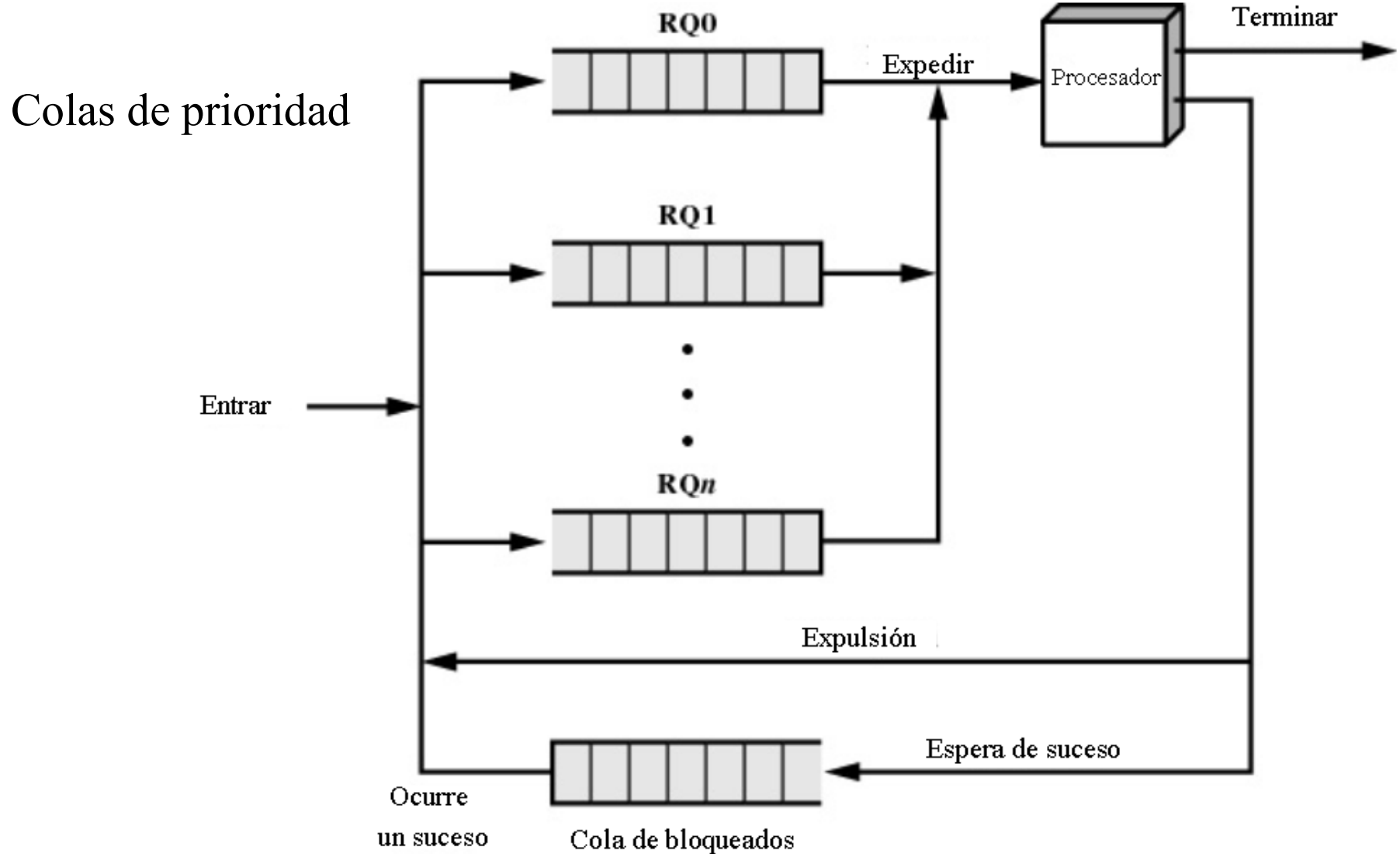
## 7. MLQ (Multiple Level Queues):

- Política **apropiativa** (preemptiva)
- Apropiado cuando los diferentes procesos son fácilmente clasificables en diferentes grupos: Foreground (interactivos) / Background (por lotes)
- Tendremos distintas colas de preparado, con una prioridad distinta y algoritmo de planificación común o diferente.
- No se podrá ejecutar ningún proceso de una cola mientras queden procesos en colas de mayor prioridad.

## MLQ Inconvenientes:

- Puede haber inanición para los procesos de menor prioridad si continuamente están llegando procesos de mayor prioridad.

# Planificación de procesos - Algoritmos



# Planificación de procesos - Algoritmos

## 8. MLQ con realimentación:

- Política **apropiativa** (preemptiva)
- Soluciona el problema de inanición del algoritmo anterior.
- Penaliza a los trabajos que han estado ejecutándose durante más tiempo (cambiando a colas de menor prioridad)
- No se conoce el tiempo de ejecución restante del proceso.

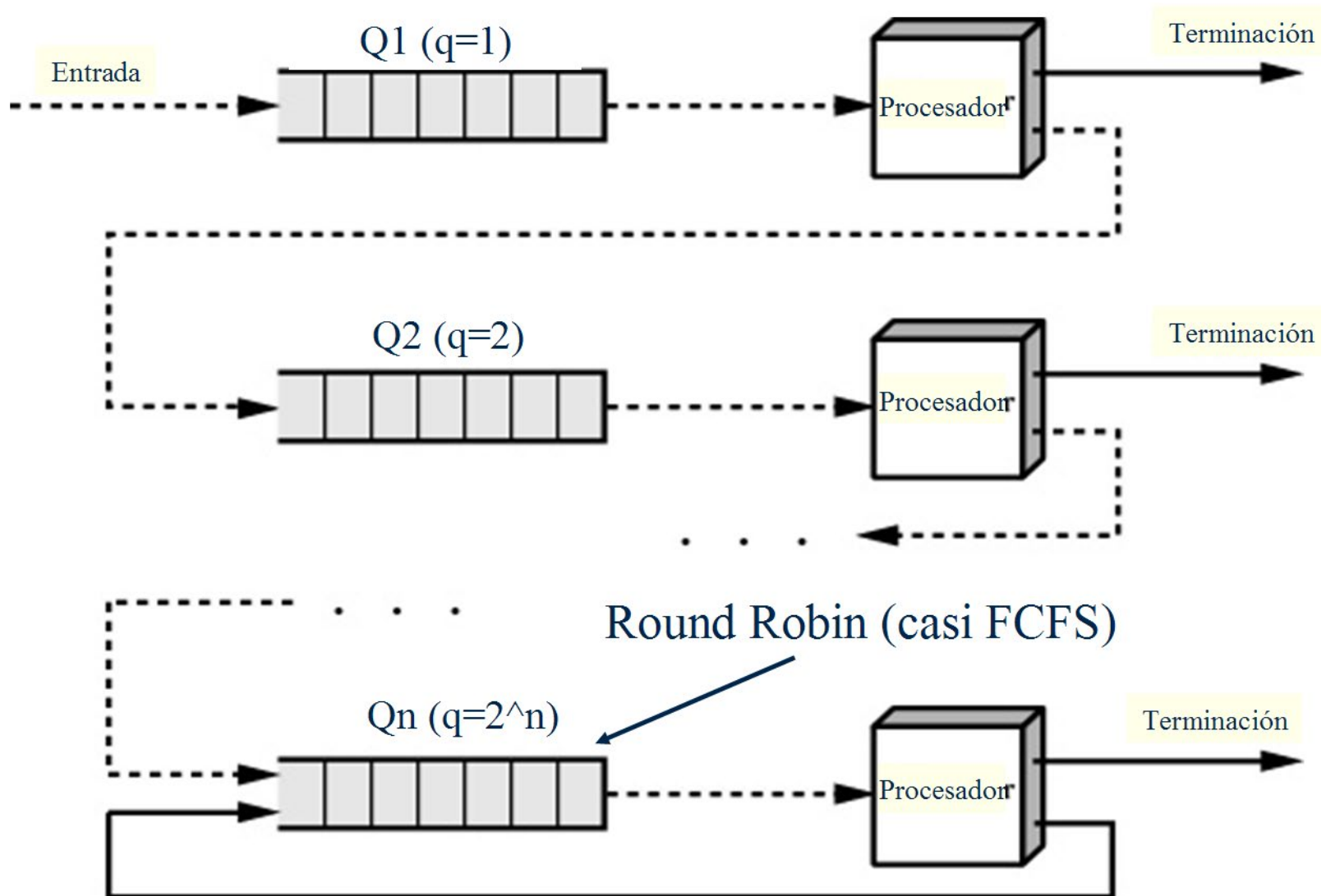
### Inconvenientes:

- Este algoritmo es el más complejo de todos

### Ventajas:

- Es el más general, ya que estableciendo los parámetros correctos podremos simular cualquier otro algoritmo:
  - Número de colas
  - Algoritmo de planificación de cada cola
  - Métodos para mover los procesos de cola (a mayor o a menor prioridad)

# Planificación de procesos - Algoritmos

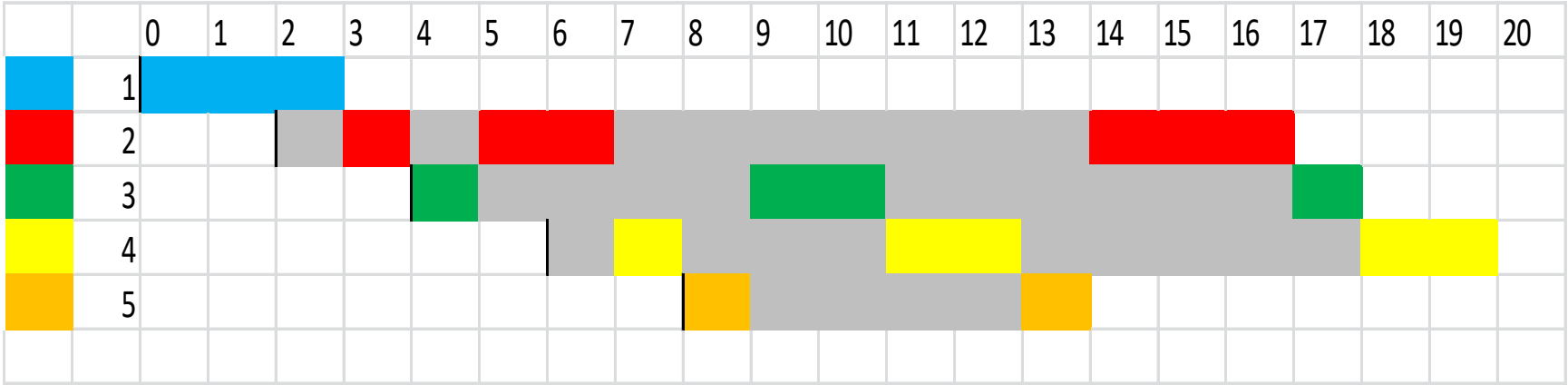


# Planificación de procesos - Algoritmos

MLQ ( $q=2^i$ ):

Proceso	Instante de llegada	Tiempo de servicio
1	0	3
2	2	6
3	4	4
4	6	5
5	8	2

 Preparado



# Planificación de procesos - Algoritmos

Ejercicio:

Proceso	Tiempo de llegada	Duración de ráfaga de CPU
P0	0	8
P1	1	4
P2	3	2
P3	5	3

- Diagrama de gant
- Tiempo medio de retorno
- Tiempo medio de espera
- Eficiencia media
- Con:
  - FIFO
  - SRTF (expulsivo)
  - SJF (no expulsivo)
  - SJF (expulsivo)
  - RR  $q=2$ ,  $q=4$
  - MLQ con realimentación  $q=2^n$

# Planificación de procesos - Algoritmos

## Ejercicio:

- Diagrama de Gant
- Tiempo medio de retorno
- Tiempo medio de espera
- Eficiencia media
- Con:
  - FIFO
  - SRTF (expulsivo)
  - SJF (no expulsivo)
  - SJF (expulsivo)
  - Prioridades (no expulsiva)
  - Prioridades (expulsiva)
  - RR  $q=4$ ,  $q=2$
  - MLQ con realimentación  $q=2^n$

(los trabajos llegan en el orden de la tabla. Prioridad max=1)

Trabajos - LLegada	Unidades tiempo	Prioridad
1 – 2	8	2
2 – 0	5	4
3 – 8	7	2
4 – 3	2	3



# Planificación de procesos - Algoritmos

## Ejercicio:

3. **(2.5 ptos)** Sea un sistema operativo cuyo kernel gestiona los hilos de los procesos. El planificador que utiliza es SRTF para las tareas y Round Robin con quantum de 10 ms para los hilos. El tiempo que necesita el sistema para realizar un cambio de tarea es de 20 ms. Dada la tabla adjunta se pide dibujar el diagrama de asignación de procesos y calcular:

- El uso de CPU
- El uso efectivo de CPU
- La sobrecarga generada por el sistema operativo.

	Llegada		Duración Ráfaga	E/S	Duración Ráfaga
Tarea 1	0	Hilo 1	20		
		Hilo 2	20	30	20
		Hilo 3	30		
Tarea 2	40	Hilo 1	20		
		Hilo 2	10	40	10
Tarea 3	70	Hilo 1	10		
		Hilo 2	30	20	10

- Uso CPU  
 $(T_{total} - T_{ociosa}) / T_{total}$
- Uso efectivo CPU  
 $T_{cpu\ tareas} / T_{total}$
- Sobrecarga SO  
 $T_{cambio} / T_{total}$

# Planificación de procesos - Algoritmos

## Ejercicio:

- Sea un sistema que usa un algoritmo de planificación Round-Robin con quantum=100ms.
- El sistema ejecuta 2 procesos
- El primer proceso P1 no utiliza E/S
- El segundo proceso P2 solicita E/S cada 50ms.
- ¿Cuál es el porcentaje de utilización de la CPU?
- Representa el cronograma de ejecución.

Fin

---

UNIDAD 3

PLANIFICACIÓN DE PROCESOS