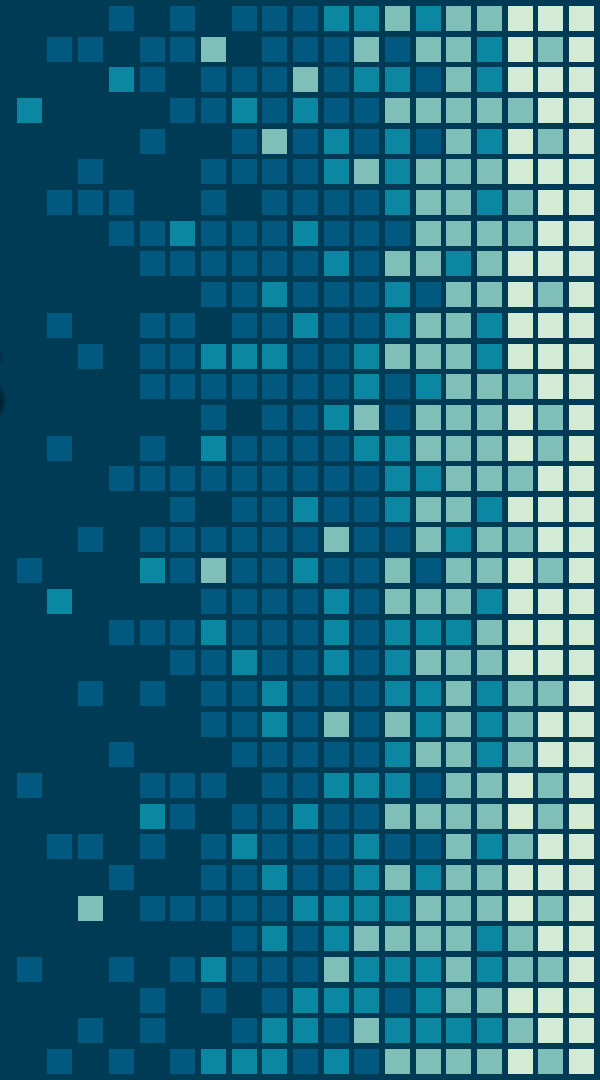

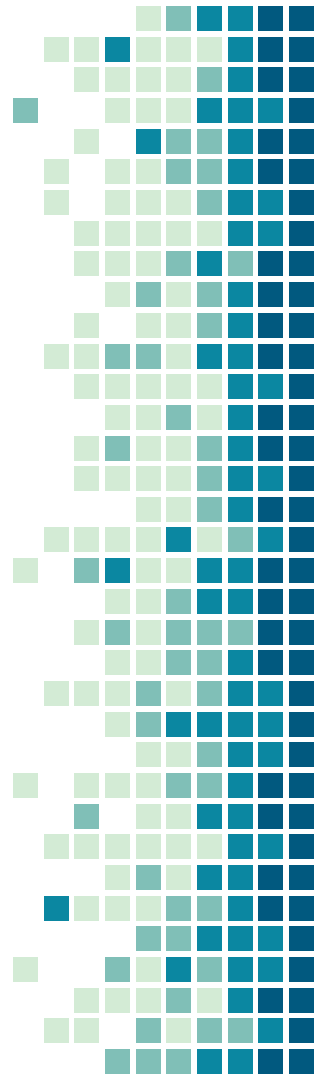


Control de procesos y administración de procesador

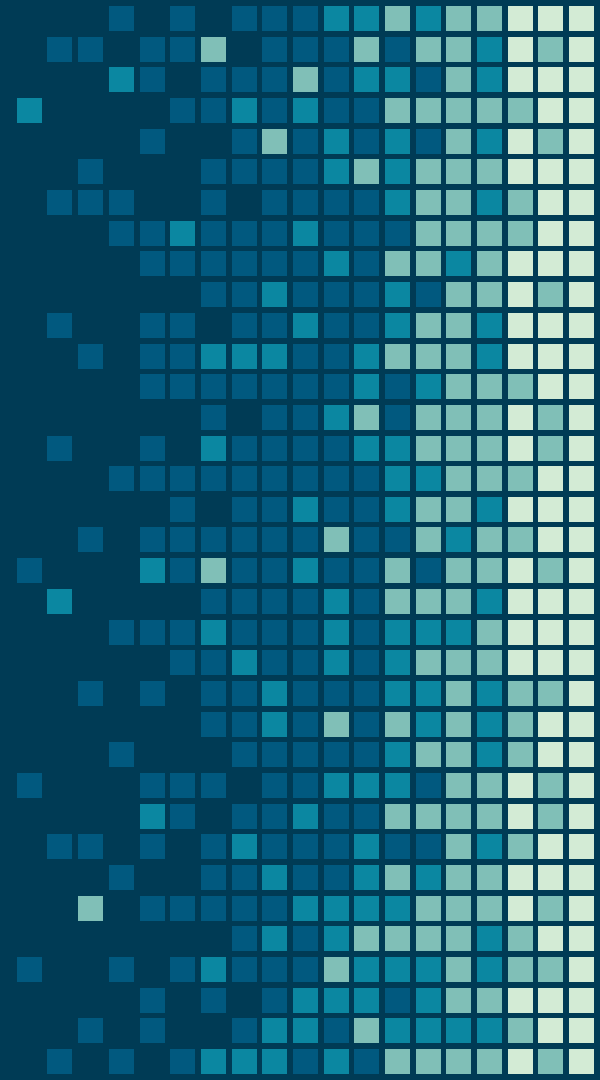


Agenda

- 
- A vertical line with a blue circle at the top and a blue arrowhead at the bottom, with small blue arrows pointing right at each agenda item.
- **Introducción**
 - **Planificación de procesos**
 - **Objetivos de la planificación**
 - **Planificación en sistemas de procesamiento por lotes**
 - **Planificación en sistemas interactivos.**
 - **Planificación en sistemas de tiempo real**

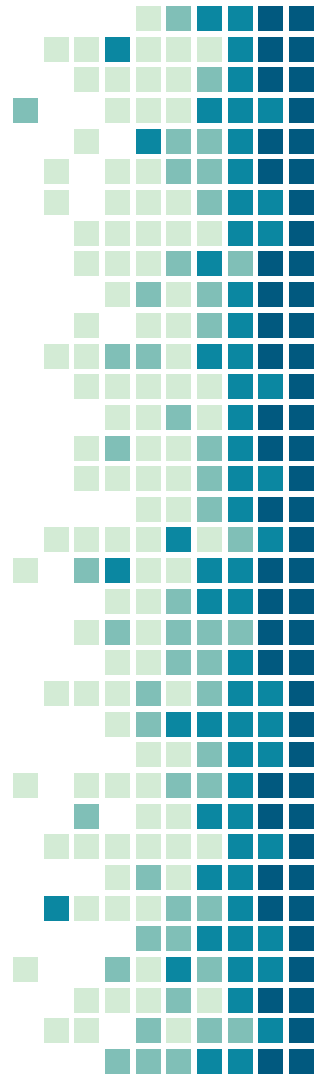


Introducción

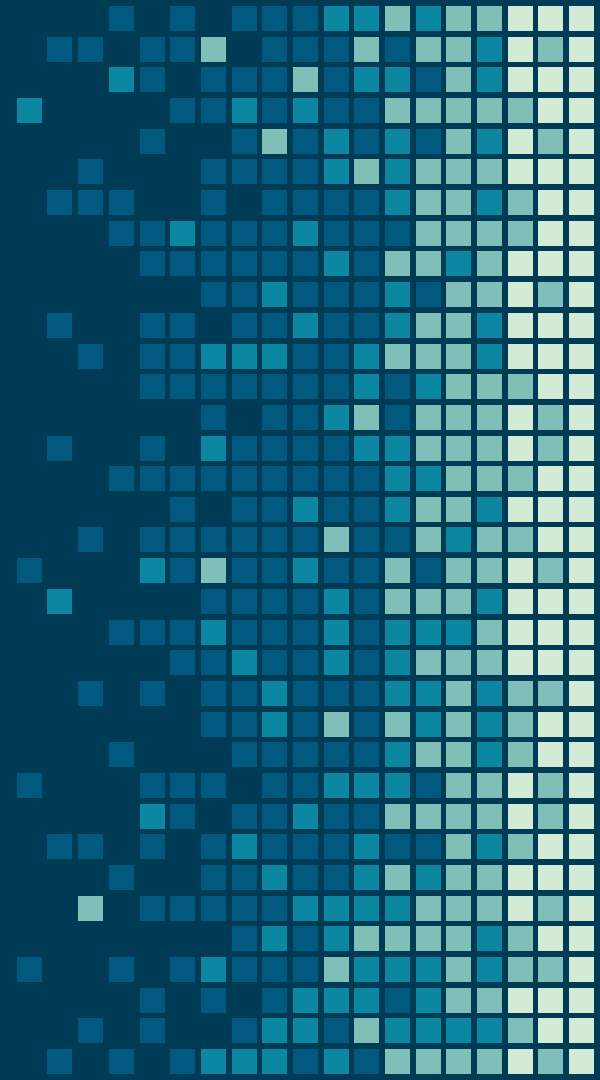




-
- ¿Cuándo y qué momento es necesario la planificación de procesos?
 - ¿Cuál fue el primer algoritmo planificador que existió?
 - ¿Realmente es necesario la planificación?

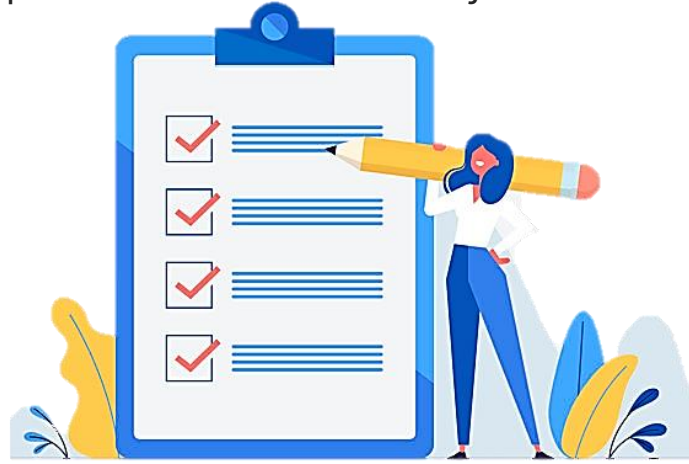


Planificación de procesos

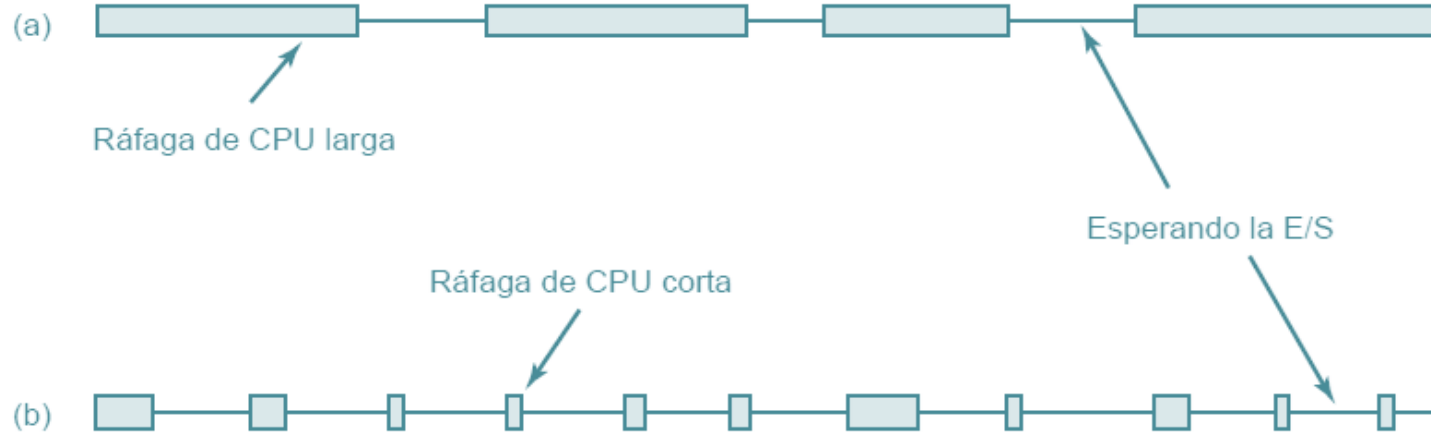


Planificación de procesos





- 💡 Actualmente las computadoras tienen un alto poder computacional.
- 💡 La conmutación de procesos es cara, por tanto los algoritmos deben ser eficientes.
- 💡 La idea es encontrar un punto de equilibrio donde se mejore el rendimiento del computador.

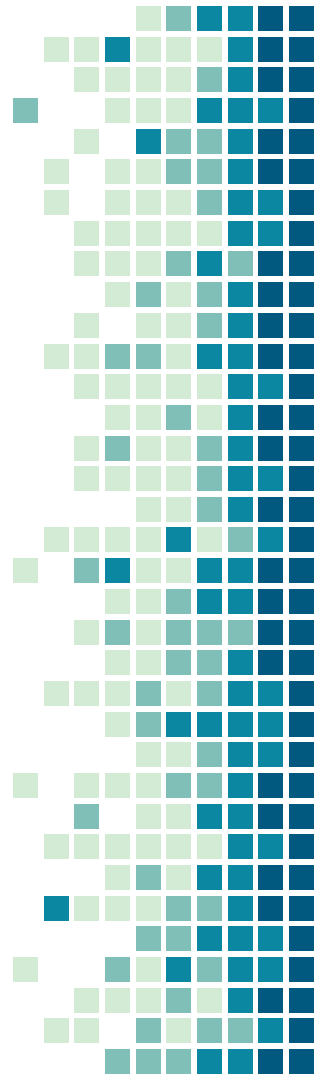


Compartimiento general de los procesos.



¿En qué momento es necesario planificar procesos?

-  Cuando se crea un nuevo proceso.
-  Cuando un proceso termina.
-  Cuando un proceso se bloquea por operación de E/S.
-  Cuando ocurre una interrupción.



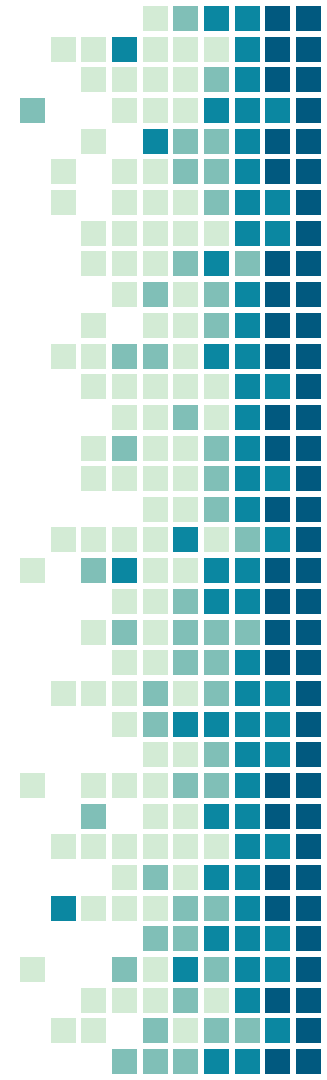
Categorías de los algoritmos de planificación.

① Procesamiento por lotes.

② Interactivo.

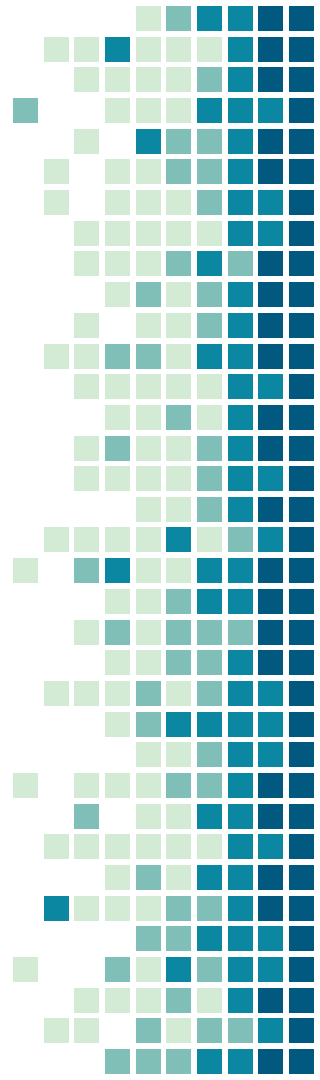
③ De tiempo real.

Las categorías clasifican los procesos dependiendo de la utilidad que estos tengan, ya que en la mayoría de los casos son diferentes.

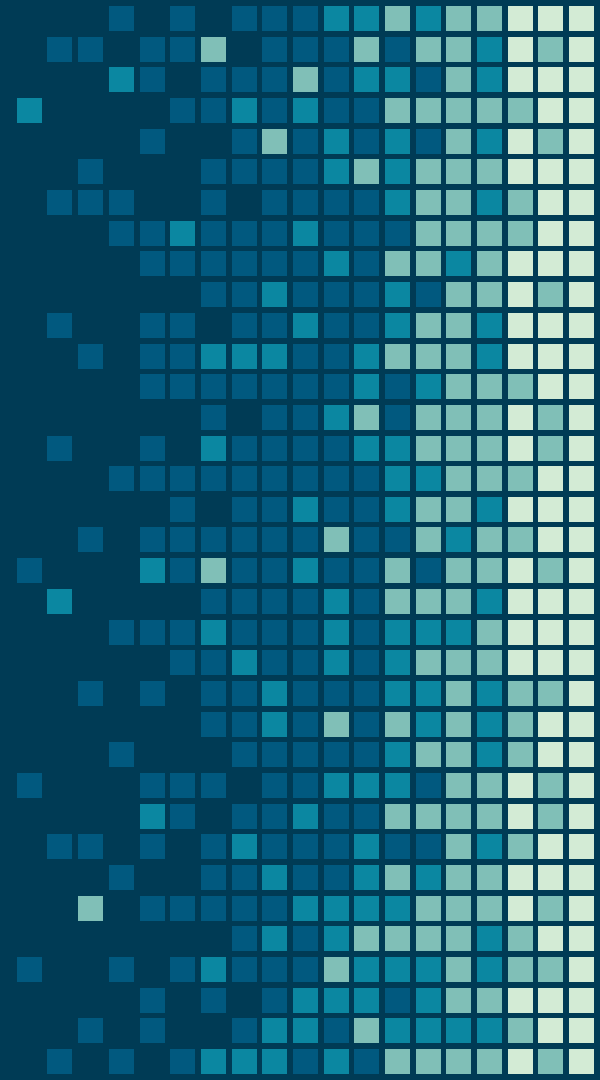


Dispatcher.

- Módulo encargado de colocar procesos en CPU.
- Cambio de contexto.
- Latencia.
- Jump a la localización correcta.



Objetivos de los algoritmos de planificación



Objetivos de los algoritmos de planificación

Equidad.

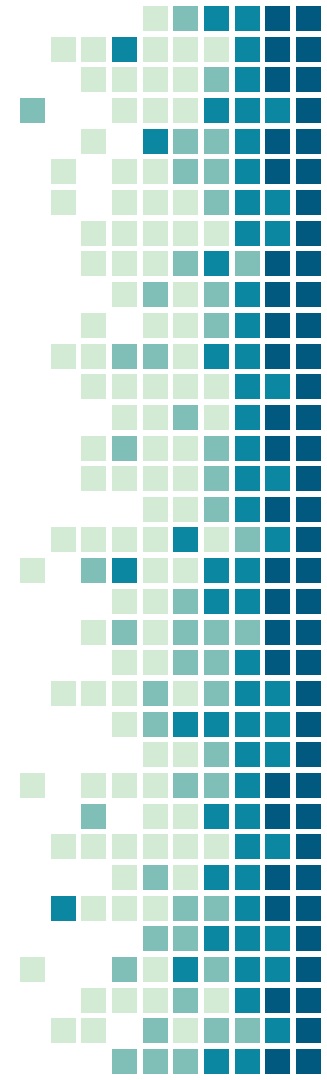
- Otorgar a cada proceso una parte justa de la CPU.

Aplicación de políticas.

- Verificar el cumplimiento de estas.

Balance.

- Mantener ocupadas todas las partes del sistema.

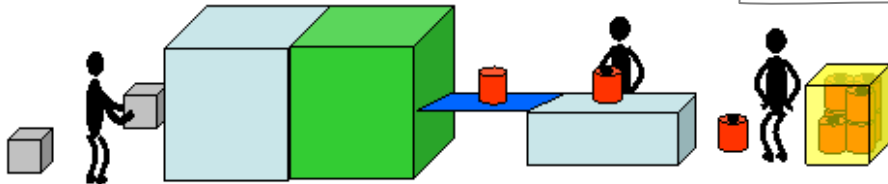


Sistemas de Procesamiento

Rendimiento: Maximizar el número de trabajos por unidad de tiempo.

Tiempo de retorno: Minimizar el tiempo entre la entrega y la terminación.

Utilización de la CPU: Mantener ocupada la CPU todo el tiempo.



Sistemas de procesamiento interactivo



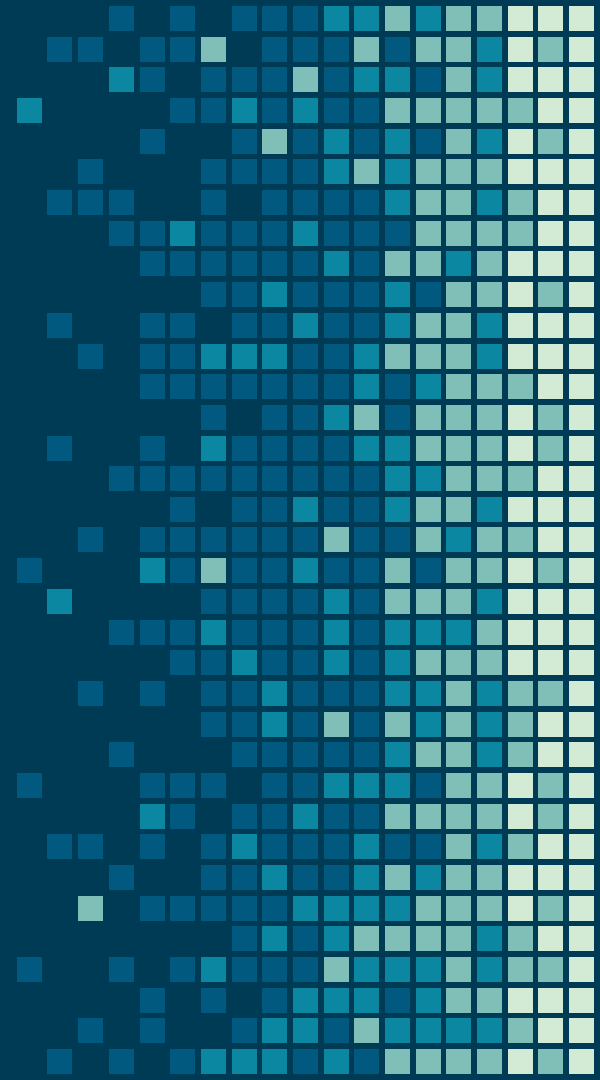
- ✓ Tiempo de respuesta: Responder a las peticiones con rapidez.
- ✓ Proporcionalidad: Cumplir las expectativas de los usuarios.

Sistemas de tiempo real

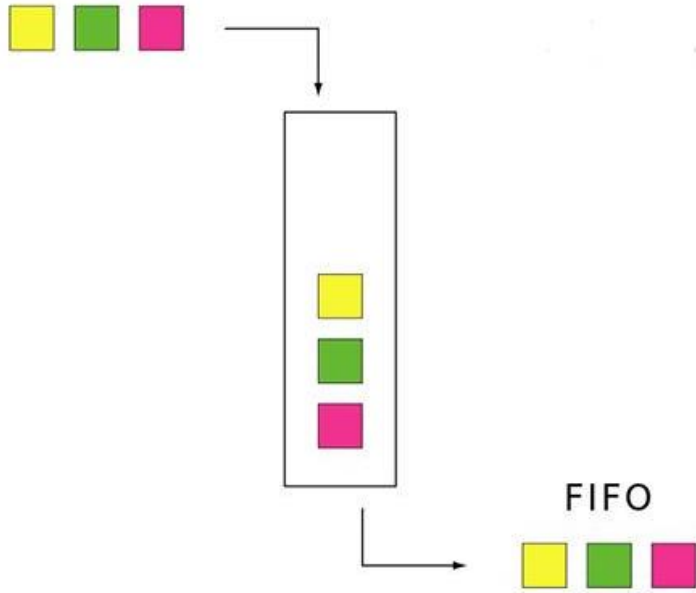
- Cumplir con los pasos: Evitar perder datos.
- Predictibilidad: Evitar la degradación de la calidad en los sistemas multimedia.



Planificación en sistemas de procesamiento por lotes



Primero en entrar, primero en ser atendido



- ➔ Es el más sencillo de todos.
- ➔ Es no apropiativo.
- ➔ El procesador se asigna de acuerdo con el orden de llegada.
- ➔ El proceso utiliza el procesador hasta que se complete el mismo.
- ➔ Una única cola de listos.

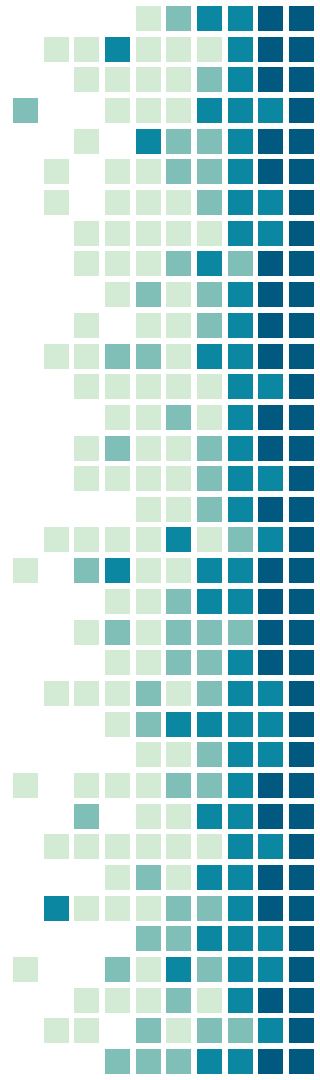
El trabajo más corto primero



Supone que los tiempos de ejecución son conocidos. El algoritmo evalúa el parámetro del tiempo y elige el que sea menor.

Suponga que se tiene cuatro procesos con tiempos a, b, c, d . Entonces el tiempo de espera es $a, a+b \dots$

Es sólo óptimo cuando todos los trabajos están disponibles al mismo tiempo.



Ejemplo

Proceso	Tiempo de llegada	Tiempo de ejecución
A	0	2
B	0	4
C	3	1
D	3	1
F	3	1

El menor tiempo restante a continuación (SRTN, Shorest Remaining Time Next)

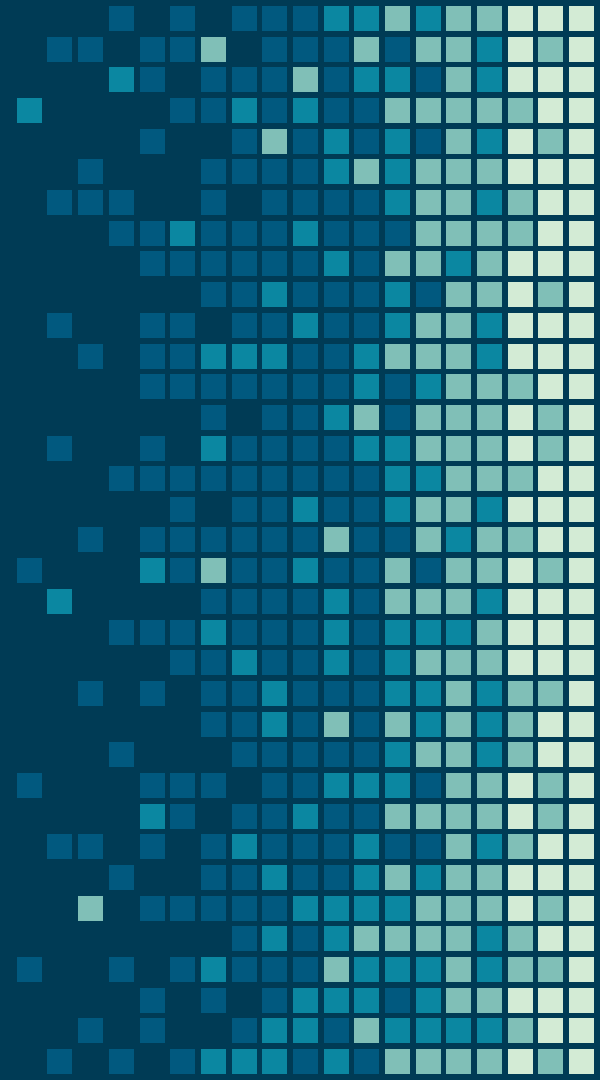


- Una versión apropiativa del algoritmo anterior.
- El algoritmo toma el proceso cuyo tiempo restante sea menor.
- Es apropiativo porque si el proceso nuevo necesita menos que el actual, se hace el cambio.

Ejemplo

Proceso	Tiempo de llegada	Tiempo de ejecución
1	0	3
2	1	5
3	2	7
4	2	10
5	4	2

Planificación en sistemas interactivos



Planificación por turno circular o round robin.

A cada proceso se le asigna un intervalo de tiempo, conocido como cuántum, durante el cual se le permite utilizar el recurso. Si el proceso no termina, el procesador es cedido a otro proceso.



¿Cuál es una desventaja?

¿Qué ocurre con los tiempo del cuántum?

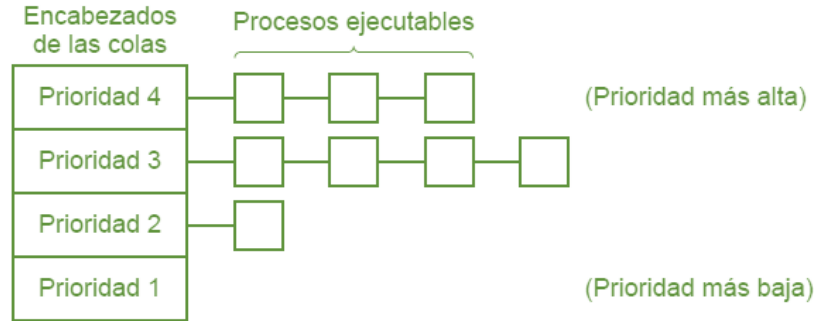
Planificación por prioridad



✎ Para evitar ejecuciones indefinidas se puede disminuir la prioridad cada n ciclos del procesador. O asignar un tiempo limitado.

A cada proceso se le asigna una prioridad y se ejecutará primero el que tenga menor número, es decir, mayor prioridad.

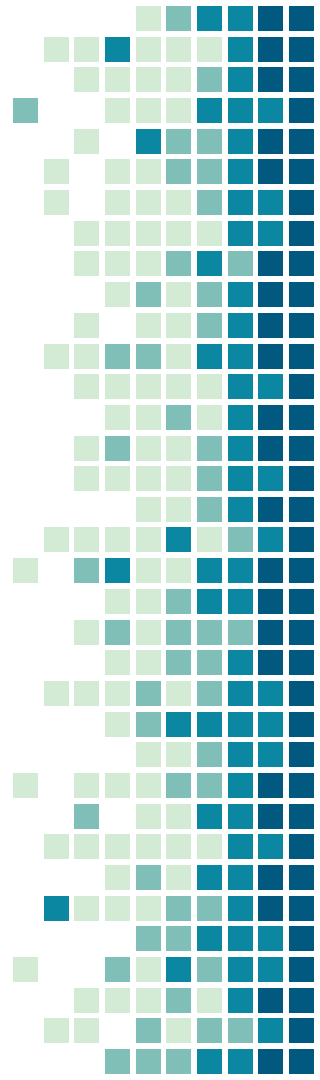
Múltiples colas



- ✚ Establecer clases de prioridad.
- ✚ Los procesos en la primera clase se ejecutan en un cuántum, los de la segunda se ejecutan por dos cuántums, la siguiente por cuatro y así sucesivamente.
- ✚ Considere un proceso que necesita realizar cálculos durante 10 cuántums.

Proceso más corto a continuación

- Parte del hecho que produce el tiempo de respuesta óptimo.
- Los procesos interactivos siguen el patrón de esperar comando, ejecutarla y volver a esperar.
- Si se considera ejecutar cada comando como todo un proceso entonces se podría minimizar el tiempo de respuesta.



Planificación garantizada

GUARANTEE

Consiste en que todos los procesos posean una misma cantidad de potencia del procesador, es decir, la misma cantidad de ciclos.

Debe llevar cuenta de potencia consumida por cada proceso.

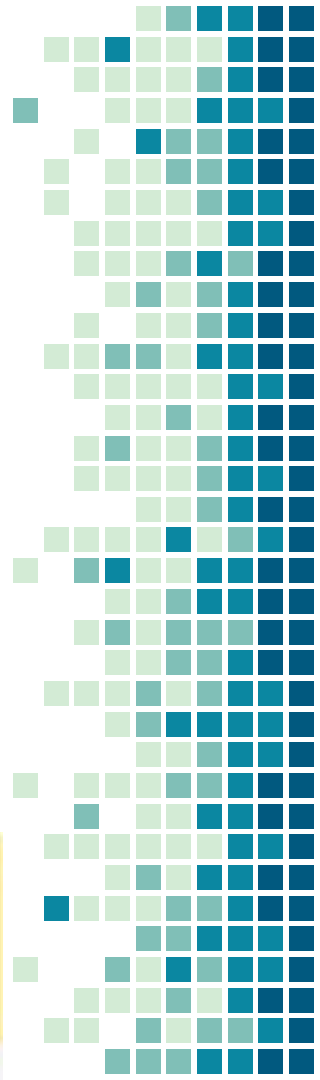
Una proporción 0,5 indica que es la mitad del tiempo transcurrido.

Planificación por sorteo

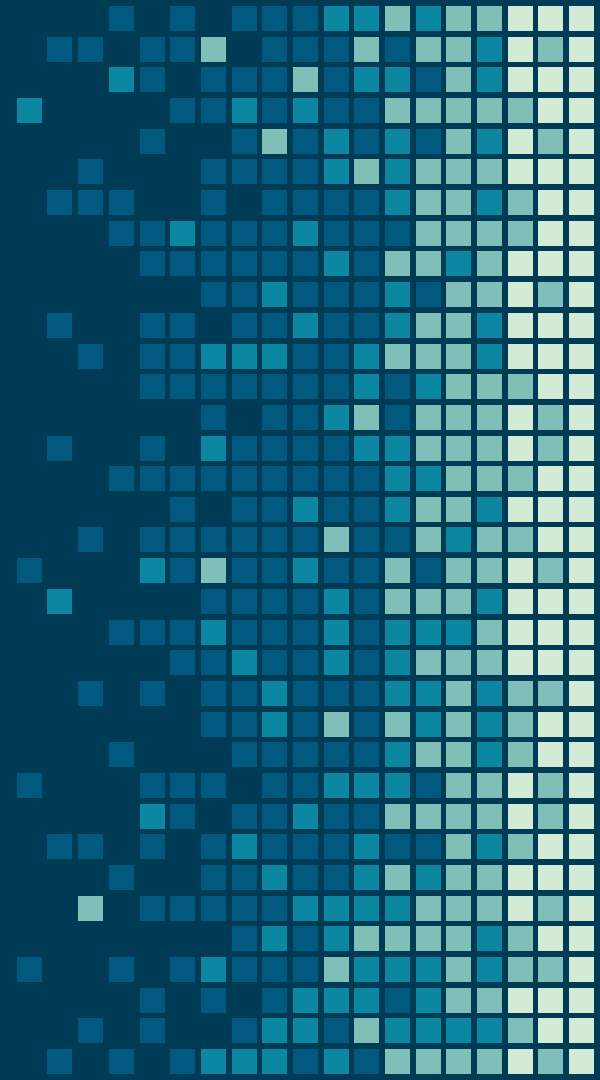
La idea fundamental de este algoritmo es que cuando se tenga que tomar una decisión de planificación, se selecciona un proceso al azar. Utiliza boletos.



Por ejemplo si el algoritmo toma la decisión 50 veces en un segundo, entonces cada proceso utilizará el procesador por 20 ms.



Planificación en sistemas de tiempo real



Planificación en sistemas de tiempo real.

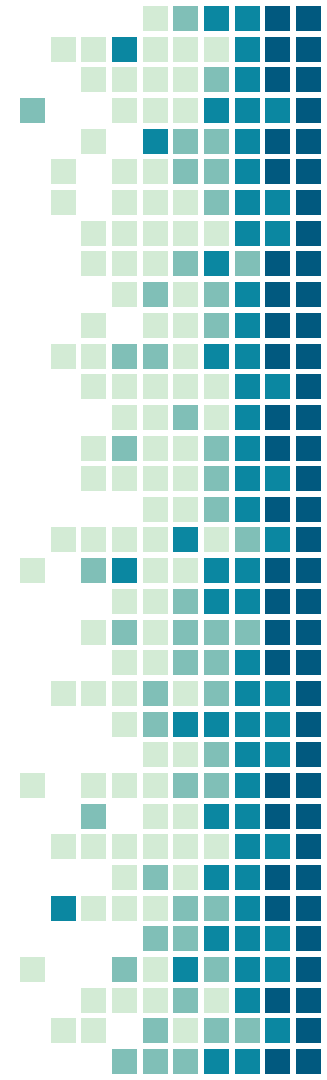


En un sistema de tiempo real, el tiempo juega un papel esencial.

Por lo general un elemento externo genera un evento y la computadora debe reaccionar en un tiempo en específico.

Generalmente los procesos son cortos y se ejecutan en su totalidad.

Cuando se detecta un evento, es responsabilidad del planificador velar que se cumplan los tiempos.

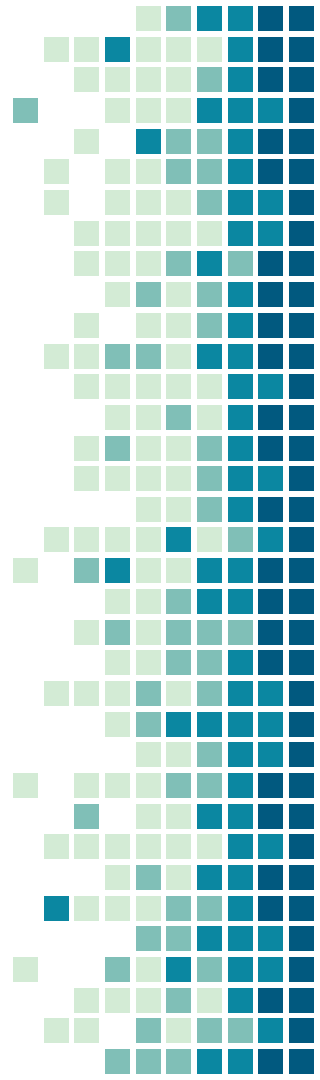


Eventos periódicos y aperiódicos.

- **Evento periódico:**
Ocurre cuando los estímulos son cada cierto tiempo definido.
- **Evento aperiódico:**
Los estímulos son impredecibles ya que no siguen un patrón.

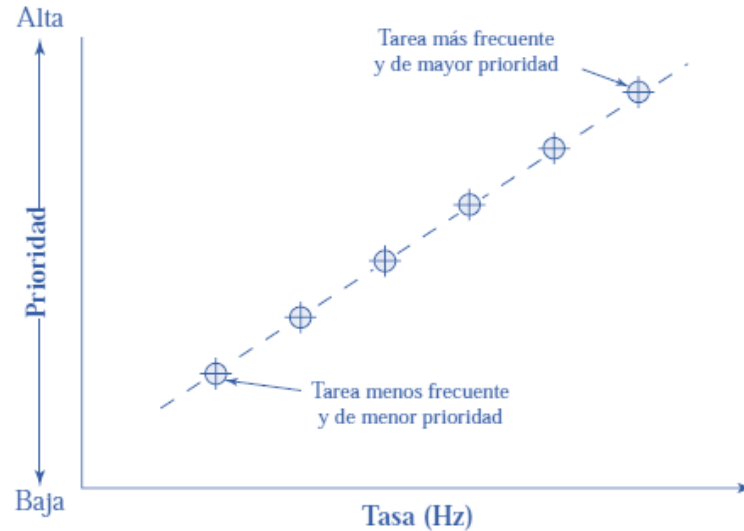
Se puede dar el caso en que el SO no pueda administrar todos los procesos. ¿Cómo darse cuenta?

$$\sum_{i=0}^m \frac{C_i}{P_i} < 1 \quad C = \text{Tiempo en el procesador}, P = \text{periodo}$$



Planificador RMS (Tasa monótona)

- Asigna prioridades a las tareas con base a sus periodos.
- El proceso con periodo más breve obtendrá una prioridad mayor.



Algoritmo RM

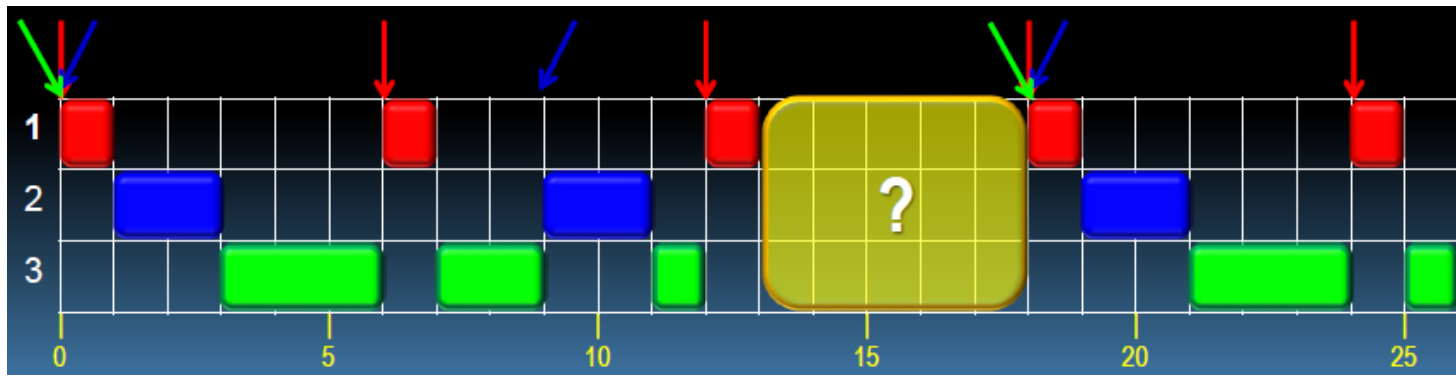
- ✓ El deadline de cada tarea es igual a su periodo.
- ✓ Se conoce el tiempo de ejecución.
- ✓ Cambio de contexto contemplado en el deadline.
- ✓ Sea: $\mu = \sum_{i=1}^n \frac{c_i}{p_i}$
- ✓ Una condición suficiente para la calendarización por RM es:
 $\mu \leq U(n) = n(2^{\frac{1}{n}} - 1)$, analizado por L y L.



¿Habr  un valor de asegurar qu  es calendarizable por RM?

Ejemplo de RM

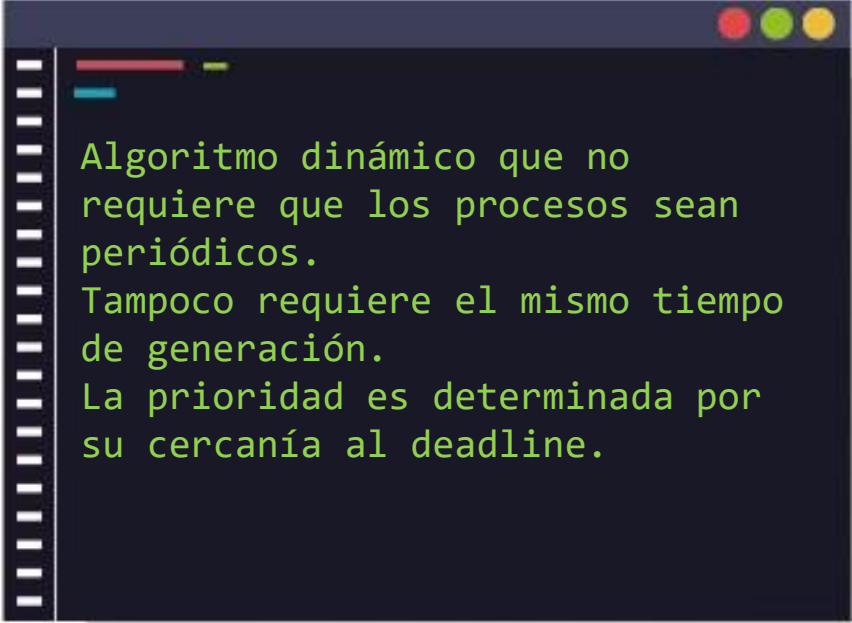
Proceso	C_i	P_i
1	1	6
2	2	9
3	6	18



Planificador EDF (Earliest deadline first)

Sea: $\mu = \sum_{i=1}^n \frac{C_i}{P_i}$

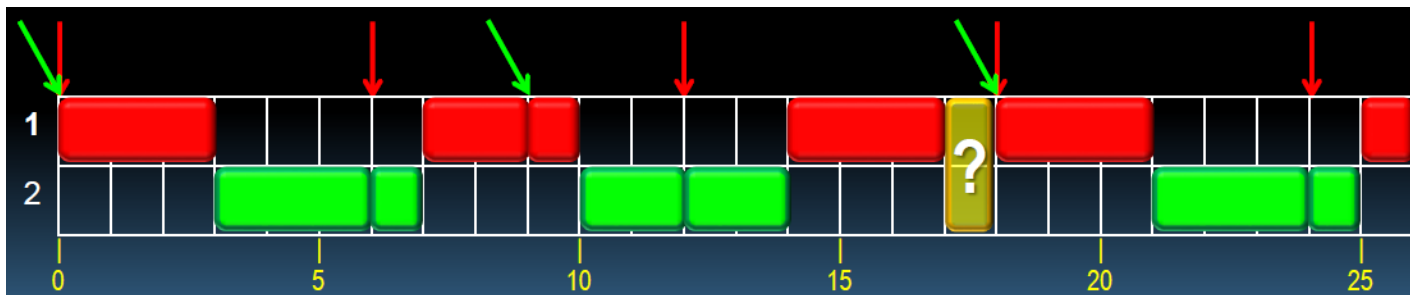
Una condición
suficiente para la
calendarización por EDF
es:
 $\mu \leq 1$, analizado por L y
L.



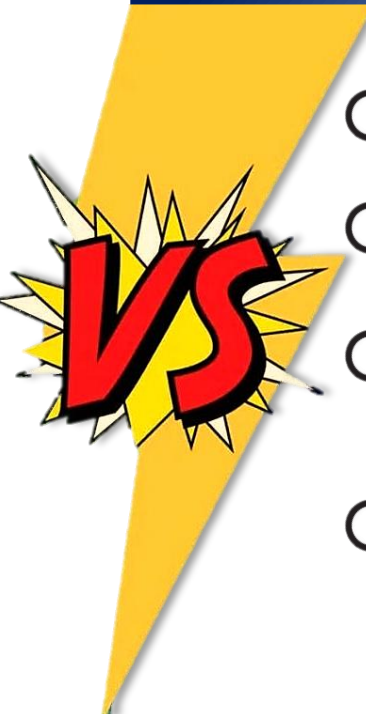
```
Algoritmo dinámico que no  
requiere que los procesos sean  
periódicos.  
Tampoco requiere el mismo tiempo  
de generación.  
La prioridad es determinada por  
su cercanía al deadline.
```

Ejemplo de EDF

Proceso	C_i	P_i
1	3	6
2	4	9



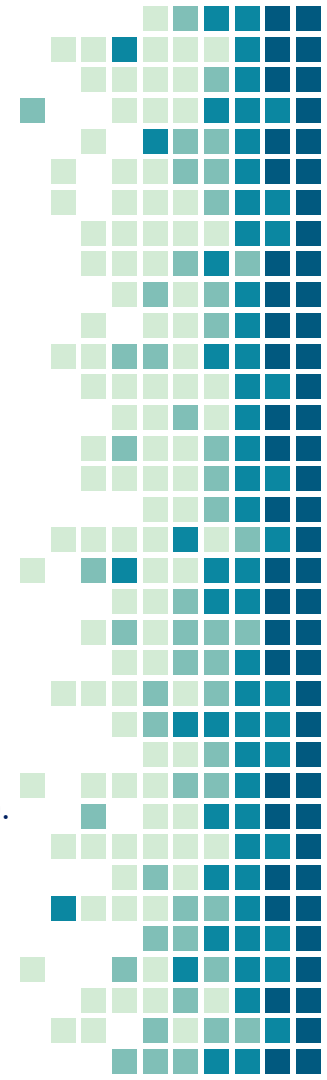
RMS vrs EDF

- 
- ➡ RMS es estático y EDF es dinámico.
 - ➡ RMS es más sencillo de entender e implementar.
 - ➡ EDF al ser dinámico puede lograr obtener 100% de utilización del procesador.
 - ➡ * Ambos se deben analizar, según el caso que corresponda.



Referencias

- ▶ Tanenbaum, A. S. (2015).
Sistemas operativos modernos. Pearson Educación
- ▶ Stallings, W. (2003).
Sistemas operativos. Martin Iturbide.
- ▶ Calvaresi, D., Marinoni, M., Sturm, A., Schumacher, M., & Buttazzo, G. (2017, August).
The challenge of real-time multi-agent systems for enabling IoT and CPS.
In *Proceedings of the international conference on web intelligence* (pp. 356-364).
ACM.



¿Preguntas?

Realizado por: Jason Leitón Jiménez.

Tecnológico de Costa Rica

Ingeniería en Computadores

2024

TEC