



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO

## TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO TECNOLÓGICO DE TLAXIACO

---

### PRACTICA 5

---

#### CARRERA:

INGENIERÍA EN SISTEMAS COMPUTACIONALES

#### ASIGNATURA:

ARQUITECTURA DE COMPUTADORAS

**SEMESTRE: 5BS**

#### INTEGRANTES:

CRUZ CRUZ DIEGO

Jiménez Sánchez Irvin

Rosas García Marco Uriel

#### DOCENTE:

OSORIO SALINAS EDWARD

Tlaxiaco Oax. 28 de octubre de 2024



***“Educación, ciencia y tecnología, progreso día con día” ®***

## OBJETIVO

El objetivo de esta práctica es comprender y simular la administración de procesos en una CPU, utilizando diferentes estrategias de programación para gestionar las tareas y su ejecución en el sistema. A través de un simulador (como Arena o SimPy), se busca que el alumno identifique, analice y aplique diversas estrategias de administración de procesos, como FCFS, SJN, Round Robin, programación de prioridades, y otras técnicas de programación de colas y multitarea. Al final, el alumno será capaz de evaluar el funcionamiento de cada estrategia y comprender cómo influye en la eficiencia y el rendimiento del sistema operativo.

## PASOS QUE SE LLEVAN A CABO EN LA ADMINISTRACIÓN DE PROCESOS/TAREAS EN LA CPU.

1. **Creación del Proceso:** El sistema operativo crea un nuevo proceso en respuesta a una solicitud de ejecución. Este proceso recibe un ID único y se le asigna un espacio en la memoria.
2. **Planificación de Procesos (Scheduling):** Se determina el orden en el que los procesos usarán la CPU. Existen varias estrategias para decidir la planificación, como FCFS, SJN, Round Robin, entre otras.
3. **Asignación de Recursos:** El sistema operativo asigna los recursos necesarios (CPU, memoria, entrada/salida) a cada proceso de acuerdo con su prioridad y los requisitos del sistema.
4. **Colocación en Estado de Ejecución:** El proceso que obtiene acceso a la CPU se coloca en estado de ejecución. Aquí, el proceso lleva a cabo sus instrucciones hasta completar su ejecución o hasta ser interrumpido.
5. **Administración de Interrupciones:** Si un proceso requiere recursos adicionales o si el sistema operativo detecta una interrupción (como la llegada de un proceso de mayor prioridad), el proceso en ejecución puede ser suspendido temporalmente.
6. **Cambio de Contexto:** Cuando el sistema operativo interrumpe o cambia entre procesos, se realiza un cambio de contexto, que implica guardar el estado actual del proceso y cargar el estado del siguiente proceso en la CPU.
7. **Terminación del Proceso:** Cuando un proceso finaliza su ejecución, el sistema operativo libera los recursos asignados y elimina el proceso de la lista de procesos activos.
8. **Manejo de Cola de Procesos:** Dependiendo de la estrategia de programación, los procesos pueden ser colocados en diferentes colas (por ejemplo, colas de prioridad) para ser seleccionados posteriormente para la ejecución.

Estos pasos se llevan a cabo continuamente, permitiendo que el sistema operativo administre eficientemente los recursos y mantenga la ejecución de múltiples procesos.

## ESTRATEGIAS DE SCHEDULING.

- **First Come First Served (FCFS):** Los procesos son ejecutados en el orden en el que llegan a la CPU.
- **Shortest Job Next (SJN):** Los procesos son ejecutados en orden ascendente de acuerdo a su tiempo de ejecución.
- **Round Robin (RR):** Los procesos son ejecutados en un orden circular, asignando un quantum de tiempo a cada proceso.
- **Priority Scheduling:** Los procesos son ejecutados de acuerdo a su prioridad.
- **Multilevel Queue Scheduling:** Los procesos son asignados a diferentes colas de acuerdo a su prioridad.
- **Multilevel Feedback Queue Scheduling:** Los procesos son asignados a diferentes colas de acuerdo a su prioridad y se les asigna un quantum de tiempo.
- **Real Time Scheduling:** Los procesos son ejecutados de acuerdo a su tiempo de ejecución.
- **Multicore Scheduling:** Los procesos son asignados a diferentes núcleos de la CPU.
- **Multitasking:** Los procesos son ejecutados de manera concurrente y se les asigna un quantum de tiempo.

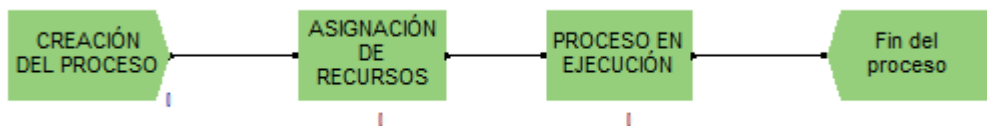
## [ESTRATEGIA 1] SIMULACIÓN DE ADMINISTRACIÓN DE PROCESOS/TAREAS EN LA CPU.

La primera estrategia seleccionada fue **First Come First Served (FCFS)**: Los procesos son ejecutados en el orden en el que llegan a la CPU.

### Descripción de las etapas de la administración de procesos/tareas.

- **Creación del Proceso:** Cuando un proceso es creado y se solicita su ejecución, el sistema operativo lo coloca en una cola de espera en el orden en el que llega. No se toma en cuenta su tiempo de ejecución ni su prioridad; únicamente su orden de llegada.
- **Cola de Listos:** Los procesos se alinean en la "cola de listos" de acuerdo al orden de llegada. La CPU seleccionará siempre el primer proceso en esta cola para ser ejecutado, respetando el orden en el que fueron introducidos.
- **Asignación de la CPU:** El primer proceso en la cola de listos se asigna a la CPU y comienza su ejecución. Este proceso ocupará la CPU hasta completar su ejecución sin interrupciones, excepto en casos de solicitudes de entrada/salida o eventos externos que lo requieran.
- **Ejecución:** Durante la ejecución, el proceso tiene acceso exclusivo a la CPU. Si se encuentra con una operación de entrada/salida, el proceso puede ser temporalmente suspendido y otro proceso de la cola de listos puede comenzar a ejecutarse.

- **Liberación de la CPU:** Al finalizar su ejecución, el proceso libera la CPU, y el sistema operativo asigna la CPU al siguiente proceso en la cola de listos. El proceso finalizado es removido de la lista de procesos activos y sus recursos son liberados.
- **Manejo de Interrupciones:** Aunque FCFS no implica cambios de contexto frecuentes, el sistema operativo debe manejar interrupciones en caso de eventos externos o bloqueos de E/S. En estos casos, el proceso afectado espera a que la operación se complete antes de retomar su lugar.
- **Retorno al Sistema Operativo:** Una vez que todos los procesos han terminado, el sistema operativo retorna al estado inactivo o espera nuevos procesos. FCFS puede ocasionar tiempos de espera elevados para procesos cortos que llegan después de procesos largos, un fenómeno conocido como "convoy effect".



1. Se crea un proceso
2. Se asignan los recursos en la CPU
3. Se mantiene el proceso en ejecución
4. Se finaliza el proceso

Al ser el primero en crear es el primero en realizarse no hay mayor complejidad en el modelo de simulación

## [ESTRATEGIA 2] SIMULACIÓN DE ADMINISTRACIÓN DE PROCESOS/TAREAS EN LA CPU.

La segunda estrategia seleccionada fue **Priority Scheduling**: Los procesos son ejecutados de acuerdo a su prioridad.

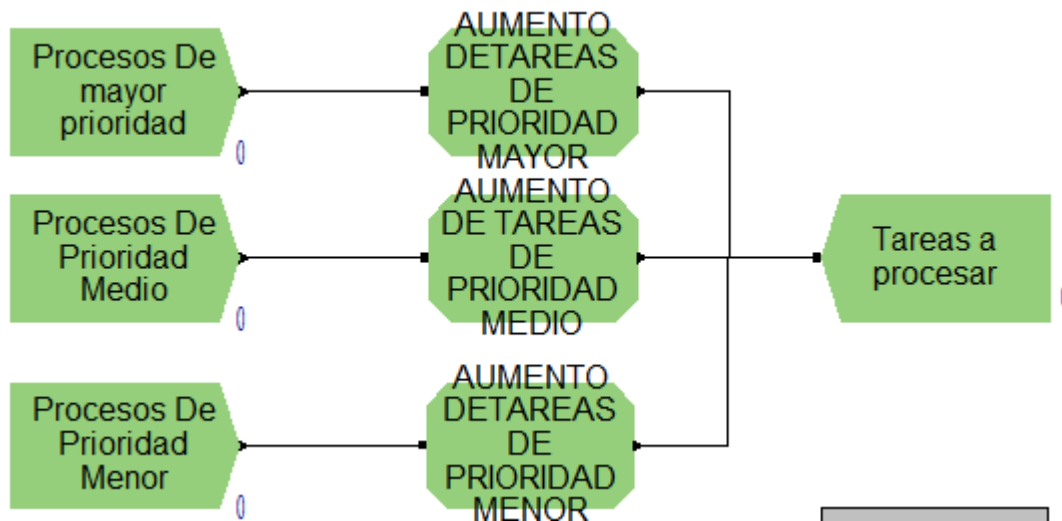
### Descripción de las etapas de la administración de procesos/tareas.

- **Creación del Proceso:** Al crearse un nuevo proceso, el sistema operativo asigna una prioridad a este proceso, ya sea de forma automática (basada en el tipo de proceso) o manual (establecida por el usuario o el sistema).
- **Cola de Prioridades:** Los procesos se colocan en una cola de listos organizada en función de sus prioridades. Los procesos con mayor prioridad se colocan al inicio de la cola, mientras que los de menor prioridad quedan en posiciones posteriores.
- **Asignación de la CPU:** La CPU se asigna al proceso con la prioridad más alta disponible en la cola de listos. Si dos o más procesos tienen la misma prioridad,

se recurre a un criterio secundario, como el orden de llegada, para decidir cuál se ejecutará primero.

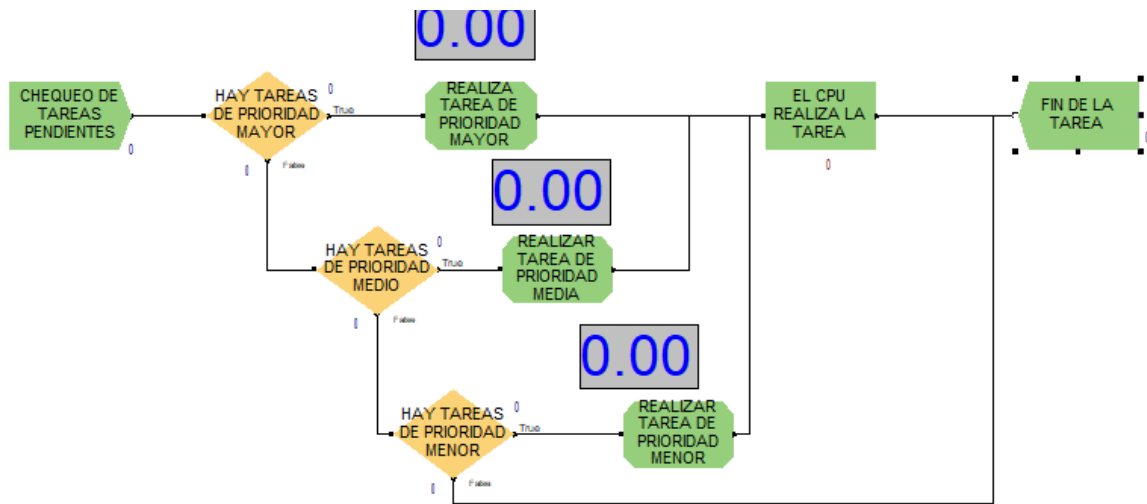
- **Ejecución:** El proceso con la mayor prioridad comienza a ejecutarse. Puede mantener la CPU hasta que complete su tarea o hasta que ocurra una interrupción, como la llegada de un proceso con una prioridad aún más alta (en sistemas de "preemptive priority scheduling").
- **Cambio de Contexto:** Si llega un proceso con una prioridad más alta que el que está actualmente en ejecución, el sistema operativo realiza un cambio de contexto. Esto implica guardar el estado del proceso en ejecución y dar paso al nuevo proceso prioritario.
- **Manejo de Interrupciones y Suspensiones:** Si un proceso entra en espera debido a una solicitud de entrada/salida, este es suspendido temporalmente. Una vez que el proceso vuelve a estar listo, se reevalúa su prioridad y se reinserta en la cola en la posición correspondiente a su nivel de prioridad.
- **Liberación de la CPU:** Cuando el proceso de mayor prioridad finaliza su ejecución, la CPU se asigna al siguiente proceso de mayor prioridad en la cola de listos. Este proceso es removido de la lista de procesos activos y sus recursos se liberan.
- **Retorno al Sistema Operativo:** Cuando no hay procesos en la cola de listos, la CPU retorna a un estado de espera. En sistemas de prioridad, es común que los procesos críticos tengan prioridad para asegurar su ejecución oportuna.

## PRIMERA PARTE DE LA SIMULACIÓN



1. Se crean los procesos de prioridad mayor, medio y menor
2. Se tiene un conteo de cuantas tareas de las distintas prioridades hay y al llegar una nueva se suma 1 a esa prioridad
3. Todas las tareas llegadas serán las tareas a realizar

## SEGUNDA PARTE DE LA SIMULACIÓN

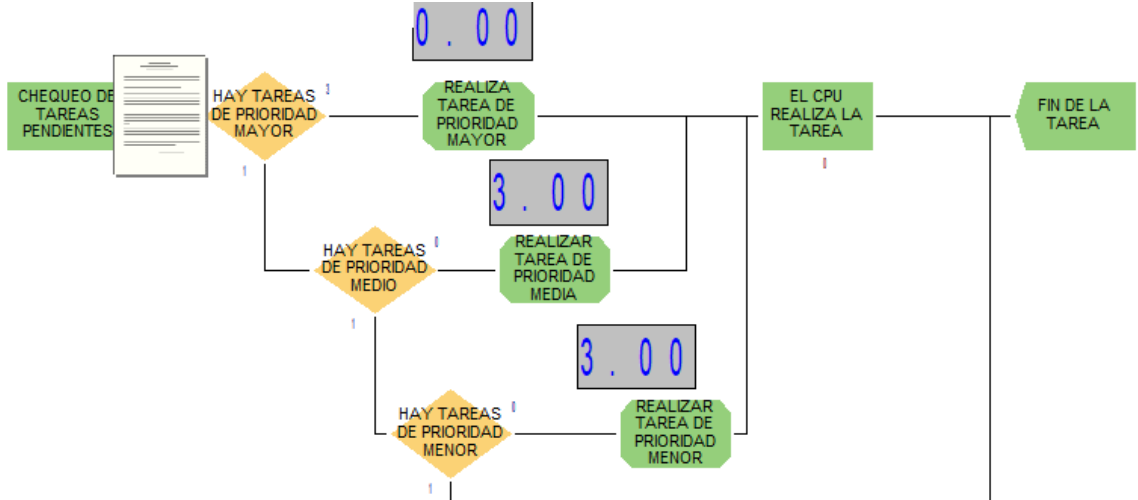


Tenemos un sistema que va checando las tareas/procesos pendientes

En la primer **decide** es si hay tareas con mayor prioridad se realizan primero y si no se van realizando las de menor prioridad y eso aplica para los otros **decide**.

Tenemos un contador de tareas que indica cuantas tareas hay en cada prioridad

Ejemplo:



En este caso se puede observar que no hay tareas de mayor prioridad ya que ya se realizaron entonces se realizara una de prioridad media y en dado caso llegara una de mayor prioridad después de realizar esa tarea se realizaría enseguida.

Las de menor prioridad se realizan solo si no hay ni de prioridad mayor ni de prioridad media.

En dado caso de que no haya tareas se espera hasta que llegue una nueva

## CONCLUSIÓN.

En esta práctica, logramos simular y analizar las diferentes estrategias de administración de procesos en la CPU, comprendiendo cómo cada una afecta el rendimiento y la eficiencia del sistema operativo. Las estrategias como First Come First Served (FCFS), demostraron tener ventajas y limitaciones específicas, dependiendo del tipo de carga de trabajo y los requisitos del sistema. Mediante el uso de un simulador Arena, observamos cómo cada estrategia distribuye el tiempo de CPU entre los procesos, afectando factores como el tiempo de espera, el tiempo de respuesta y la equidad en la asignación de recursos.

Este ejercicio permitió una comprensión más profunda del funcionamiento interno de un sistema operativo en cuanto a la administración de tareas, así como la importancia de seleccionar una estrategia de programación adecuada para optimizar el rendimiento del sistema y atender las necesidades de los procesos.

## Referencias.

Equipo editorial de IONOS. (2023, 19 enero). *Administrador de tareas: gestionar procesos en Windows*. IONOS Digital Guide.

<https://www.ionos.mx/digitalguide/servidores/configuracion/abrir-el-administrador-de-tareas/#:~:text=Iniciar%20el%20administrador%20de%20tareas%20de%20Windows%20en%20el%20men%C3%BA,de%20la%20barra%20de%20tareas.>

AIX 7.1. (s. f.). <https://www.ibm.com/docs/es/aix/7.1?topic=administration-process-management>