# Actividad 3: Gestión de procesos (Material complementario)

La **Placa Madre** es el componente principal al que se conectan los demás. Imagina la placa

**Gestión de Procesos: El Corazón de la Multitarea en tu Sistema Operativo**

Imagina que tu computadora es un restaurante con un solo chef (la **CPU**) y muchos clientes (tus **programas**). La **gestión de procesos** es como la labor del jefe de cocina, que se asegura de que todos los pedidos se preparen de manera eficiente, que los clientes no esperen demasiado y que el restaurante (tu sistema) funcione sin problemas, incluso si hay muchos pedidos al mismo tiempo.

Esta es una de las funciones más críticas de un sistema operativo, ya que permite que puedas navegar por internet, escuchar música y escribir un documento, ¡todo al mismo tiempo!

## 1. Procesos e Hilos: Las Unidades de Trabajo

Para gestionar la ejecución de los programas, el sistema operativo define dos conceptos fundamentales: **procesos** e **hilos**.

### 1.1. Proceso: La Instancia Independiente de un Programa

- **Definición:** Un **proceso** es una instancia en ejecución de un programa. Cuando abres un navegador web, se crea un proceso. Cuando abres un editor de texto, se crea otro proceso.
- **Recursos Aislados:** Cada proceso es una entidad **autónoma y aislada**. Esto significa que:
    - Tiene su **propio espacio de memoria virtual** (un conjunto de direcciones de memoria que solo ese proceso puede usar). Esto es crucial para la seguridad, ya que un proceso defectuoso no puede corromper la memoria de otro.
    - Posee sus propios **recursos del sistema**, como archivos abiertos, descriptores de sockets de red, y dispositivos asignados.
    - Los procesos generalmente **no comparten memoria directamente**. Si necesitan comunicarse, utilizan mecanismos especiales de **Comunicación entre Procesos (IPC - Inter-Process Communication)**, como tuberías, colas de mensajes o memoria compartida.
- **Ejemplo:** Tu navegador web es un proceso. Tu reproductor de música es otro proceso. Ambos se ejecutan de forma independiente.

### 1.2. Hilo (Thread): La Unidad de Ejecución dentro de un Proceso

- **Definición:** Un **hilo** es la **unidad más pequeña de ejecución** que el sistema operativo puede planificar. Piensa en ellos como "mini-procesos" dentro de un proceso más grande.
- **Recursos Compartidos:** A diferencia de los procesos, todos los hilos de un mismo proceso **comparten el mismo espacio de memoria y la mayoría de los recursos del proceso** (archivos abiertos, etc.).
- **Contexto Propio:** Sin embargo, cada hilo tiene su propio **contexto de ejecución**:
    - **Pila (Stack):** Un área de memoria para almacenar variables locales y el historial de llamadas a funciones de ese hilo.
    - **Contador de Programa (PC - Program Counter):** La dirección de la próxima instrucción que ese hilo debe ejecutar.
    - **Registros de CPU:** Los valores de los registros del procesador para ese hilo.
- **Ventajas de los Hilos:**
    - **Concurrencia:** Permiten que un solo programa realice **múltiples tareas simultáneamente** dentro del mismo proceso. Por ejemplo, en un navegador, un hilo puede cargar imágenes, otro mostrar texto y un tercero manejar las interacciones del usuario.
    - **Comunicación Eficiente:** Dado que comparten memoria, la comunicación entre hilos del mismo proceso es mucho más rápida y sencilla que la comunicación entre procesos.
    - **Menor Sobrecarga:** Crear y cambiar entre hilos es más rápido y consume menos recursos que crear y cambiar entre procesos.
- **Ejemplo:** En un editor de texto, un hilo puede estar guardando el documento en segundo plano, mientras otro hilo responde a las pulsaciones del teclado del usuario.

## 2. El Registro de la Actividad: Tablas del Sistema Operativo

Para gestionar procesos e hilos, el sistema operativo mantiene estructuras de datos internas que contienen toda la información relevante.

- **Tabla de Procesos (Process Table / PCB - Process Control Block):** Una tabla principal donde cada entrada (un **PCB**) almacena la información vital de un proceso.


- **Identificador del Proceso (PID):** Un número único que el SO asigna a cada proceso (¡como el DNI de un proceso!).
- **Estado del Proceso:** Indica si el proceso está `En ejecución`, `Listo` o `Bloqueado`.
- **Contador de Programa (PC):** La dirección de la próxima instrucción a ejecutar si este proceso vuelve a la CPU.
- **Registros de CPU:** El contenido de los registros del procesador cuando el proceso fue pausado, para poder restaurarlos y que continúe donde lo dejó.
- **Información de Gestión de Memoria:** Detalles sobre el espacio de memoria virtual y física asignada al proceso.
- **Recursos Asignados:** Lista de archivos abiertos, dispositivos de E/S en uso, permisos.
- **Información de Contabilidad:** Uso de CPU, tiempo de inicio, etc.

- **Tabla de Hilos (Thread Control Block - TCB):** Dentro de cada PCB de un proceso que tiene hilos, hay información específica para cada hilo:
  - **Identificador del Hilo (TID):** Un número único para cada hilo dentro de un proceso.
  - **Estado del Hilo:** Similar a los estados del proceso.
  - **Pila del Hilo:** Puntero a la ubicación de la pila de ejecución de ese hilo.
  - **Registros y PC del Hilo:** El contexto específico del procesador para ese hilo.

## 3. Planificación de Procesos: El Arbitraje de la CPU

La **planificación de procesos (CPU Scheduling)** es el arte y la ciencia de decidir qué proceso (o hilo) obtendrá la CPU y por cuánto tiempo. El objetivo es optimizar el uso de la CPU, proporcionar una respuesta rápida a los usuarios y garantizar la equidad.

### 3.1. Los Tres Estados Fundamentales de un Proceso

Los procesos (y por extensión, los hilos) cambian constantemente de estado a lo largo de su vida útil:

1. **Listo (Ready):** El proceso ha sido cargado en memoria principal y está esperando su turno para ser ejecutado por la CPU. Está preparado para correr, solo necesita el procesador.

2. **En Ejecución (Running):** El proceso está utilizando activamente la CPU para ejecutar sus instrucciones.
3. **Bloqueado (Blocked/Waiting):** El proceso está esperando que ocurra algún evento externo para poder continuar. No puede hacer ningún progreso hasta que ese evento suceda. Ejemplos: esperando datos del teclado, esperando que termine una operación de lectura/escritura de disco, esperando un recurso que está siendo utilizado por otro proceso.

### 3.2. El Planificador de Procesos: El Director de Tráfico

El **planificador de procesos (Scheduler)** es la parte del sistema operativo (del kernel) que toma la decisión de qué proceso debe pasar al estado "En ejecución". Utiliza colas para organizar los procesos en los diferentes estados.

### 3.3. Transiciones de Estado Detalladas:

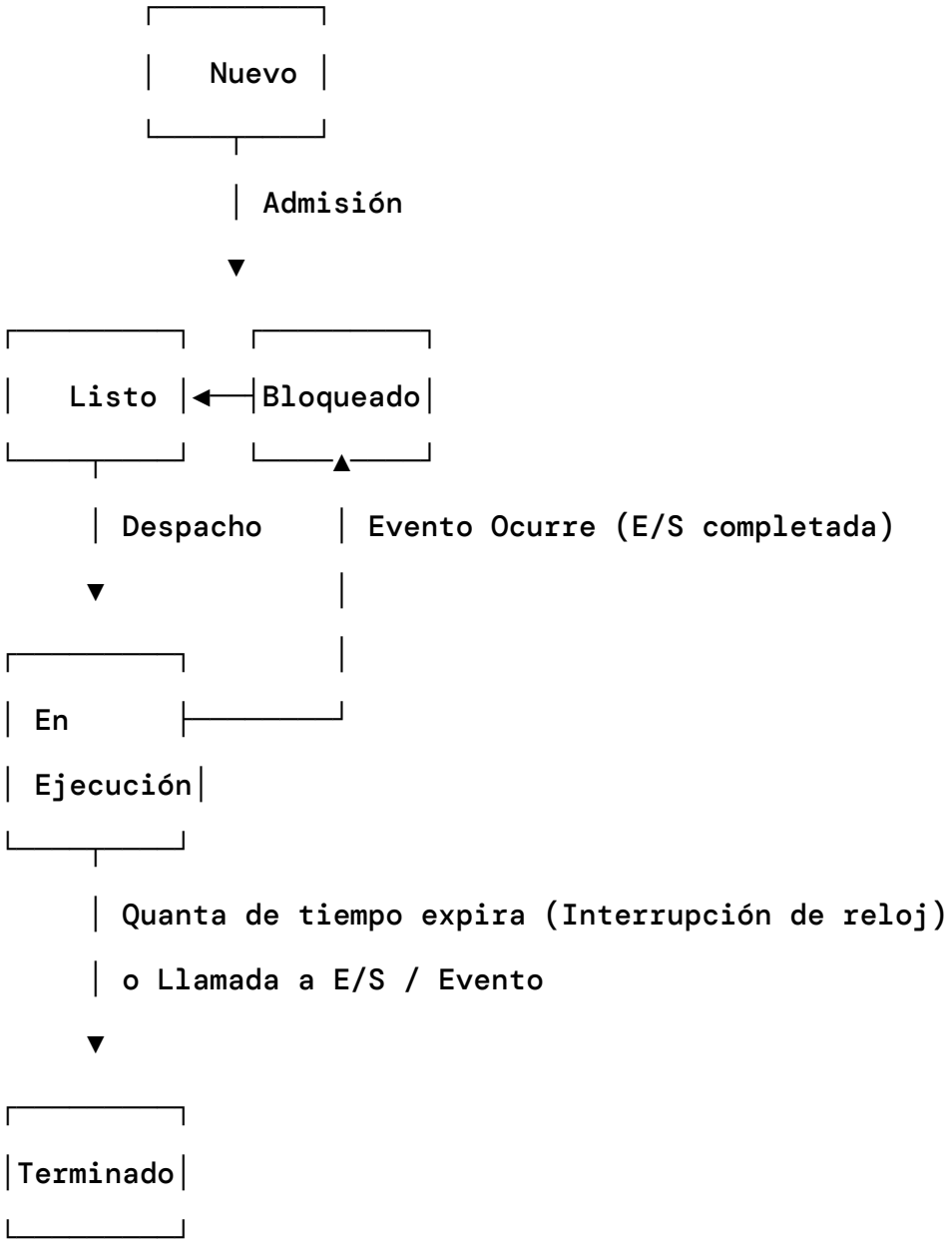
- **De "Listo" a "En Ejecución": El Despacho**
    - El planificador selecciona un proceso de la **cola de listos** (generalmente usando algún algoritmo de planificación) y le asigna la CPU. A esto se le llama **despacho**.
- **De "En Ejecución" a "Listo": La Interrupción por Tiempo (Preemption)**
    - Para evitar que un proceso monopolice la CPU y garantizar la **multitarea real** (o pseudo-paralelismo en un solo núcleo), el sistema operativo usa un **temporizador de reloj**.
    - El SO configura este temporizador para generar una **interrupción de reloj** después de un cierto **cuanto de tiempo (time slice)**.
    - Cuando el temporizador expira, se genera una interrupción de reloj, y el control pasa al kernel.
    - **Cambio de Contexto (Context Switch):** Este es un paso crítico y costoso:
        1. El kernel **guarda todo el estado actual** del proceso que se estaba ejecutando (todos sus registros de CPU, el valor de su contador de programa, etc.) en su **PCB** en la tabla de procesos.
        2. El proceso actual se mueve de la CPU y se coloca de nuevo en la **cola de listos**.
        3. El planificador elige el **siguiente proceso** de la cola de listos.

4. El kernel **carga el estado guardado** de ese nuevo proceso desde su PCB en los registros de la CPU.
5. El nuevo proceso **recomienza su ejecución** exactamente desde donde lo dejó.

- Este ciclo se repite rápidamente, dando la ilusión de que varios programas se ejecutan simultáneamente.

- **De "En Ejecución" a "Bloqueado": Esperando un Evento**
  - Un proceso pasa a este estado cuando necesita algo que no está inmediatamente disponible o que requiere una operación de E/S.
  - **Llamada al Sistema:** El proceso ejecuta una **llamada al sistema** (ej. `read()` para leer del disco, `sleep()` para esperar un tiempo).
  - El kernel recibe la llamada y, como la operación de E/S es lenta, coloca el proceso en una **cola de bloqueados** asociada a ese evento o recurso.
  - El planificador selecciona inmediatamente otro proceso de la cola de listos para usar la CPU. El proceso bloqueado libera la CPU.
- **De "Bloqueado" a "Listo": Evento Completado**
  - Cuando el evento que el proceso bloqueado estaba esperando finalmente ocurre (ej. los datos del disco han sido leídos, el usuario ha pulsado una tecla):
  - El dispositivo de hardware (ej. el controlador de disco) genera una **interrupción de hardware** para notificar al sistema operativo.
  - El kernel (a través del manejador de interrupciones) identifica qué proceso estaba esperando ese evento.
  - El proceso se mueve de la cola de bloqueados y se coloca de nuevo en la **cola de listos**, esperando su turno para volver a usar la CPU.

**Diagrama de Estados de un Proceso (Simplificado):**




La gestión de procesos es el arte que permite que tu computadora sea una máquina multitarea, responsive y estable. Desde el aislamiento de recursos que garantiza la seguridad hasta la planificación inteligente que simula la ejecución simultánea, el sistema operativo trabaja incansablemente para orquestar la vida de cada programa en tu sistema.