

## ¿Que es un proceso?

En sistemas operativos, un proceso es un programa en ejecución.

- Incluye:
- Código ejecutable
- Espacio de memoria asignado
- Registros del CPU
- Recursos como archivos, dispositivos y permisos
- Cada proceso tiene un PCB (Process Control Block) que almacena su estado, prioridad, contador de programa, etc.
- Es la unidad básica de ejecución que el sistema operativo gestiona para realizar tareas.



# ¿Cuáles son los estados de un proceso?

Algunos sistemas incluyen estados intermedios como *transición* o *zombie*, según el modelo de planificación.

Estado	Descripción
Nuevo	El proceso ha sido creado pero aún no se ejecuta
Listo	Está preparado para ejecutarse, esperando asignación de CPU
En ejecución	Está usando la CPU activamente
Bloqueado	Espera un evento externo (como E/S o liberación de recursos)
Terminado	Ha finalizado su ejecución
Suspendido	Detenido temporalmente, puede estar en espera o bloqueado



## ¿Cómo funcionan los procesos?

- El sistema operativo crea, gestiona y finaliza procesos.
- Usa un planificador para decidir qué proceso se ejecuta.
- Cada proceso tiene su propio espacio de memoria y recursos.
- Se comunican entre sí mediante IPC (Inter-Process Communication).

### Ejemplo de funcionamiento

#### Ejemplo: Proceso de impresión de un documento

- Usuario envía el archivo a imprimir.
- El sistema operativo crea un proceso de impresión.
- El proceso entra en estado Preparado.
- Cuando la impresora está libre, pasa a En ejecución.
- Si falta papel, entra en Bloqueado.
- Al completarse la impresión, pasa a Terminado.

- Los procesos son esenciales para la ejecución de tareas en un sistema operativo.
- Su gestión eficiente permite multitarea, estabilidad y rendimiento.
- Comprender sus estados y funcionamiento ayuda a optimizar recursos y detectar errores.

# Ejemplo detallado

## Proceso de abrir un navegador web

1. El usuario hace clic en el ícono del navegador.
2. El sistema operativo crea un proceso con `exec()` y lo coloca en estado *Nuevo*.
3. Se carga en memoria y pasa a *Listo*.
4. El planificador le asigna CPU → pasa a *En ejecución*.
5. Si el navegador necesita cargar una página, entra en *Bloqueado* esperando respuesta del servidor.
6. Al recibir los datos, vuelve a *Listo* y luego a *En ejecución*.
7. Cuando el usuario cierra el navegador, el proceso pasa a *Terminado* y libera recursos.