

## Actividad Principal

### Simulación comparativa de técnicas de entrada/salida

Módulo: Programación – 1.º DAM

 Objetivo:

Simular mediante programación las técnicas clásicas de E/S y comparar su impacto en el uso de la CPU.

## ¿Qué debe hacer tu programa?

Implementa un programa que simule una operación de entrada/salida en tres escenarios diferentes:

### E/S programada (polling)

- La CPU **espera activamente** mientras el dispositivo está ocupado.
- ➤ Alto uso de CPU, sin multitarea.

### E/S con interrupciones

- La CPU puede hacer otras tareas.
- El dispositivo interrumpe al terminar.
- ➤ Uso eficiente del tiempo de CPU.

### DMA (acceso directo a memoria)

## ¿Qué debes medir y comparar?

Para cada técnica, registra:

- Tiempo total de la operación
- Tiempo efectivo de bloqueo o uso de CPU
- Proporción de espera frente a procesamiento real
- Representación gráfica del comportamiento

 Puedes usar barras, líneas temporales o cualquier otra visualización.

## Requisitos mínimos

- ✓ Simular las tres técnicas en un mismo programa
- ✓ Usar `sleep()` o temporizadores para simular espera
- ✓ Mostrar resultados y comparación clara
- ✓ Código estructurado y con funciones
- ✓ Comentarios explicativos sobre cada parte

## Extensiones opcionales

Si completas la actividad básica, puedes:

- Simular múltiples procesos de E/S en paralelo
- Implementar planificación de disco (ej: FCFS vs SCAN)
- Usar peticiones aleatorias de E/S
- Comparar con herramientas reales como `htop` , `iostat` , `iostat`

## Entrega esperada

- Código funcional en Java, Python o pseudocódigo
- Comparativa clara entre las técnicas
- Visualización del comportamiento
- Explicación escrita u oral de las conclusiones

 ¡Haz que tu programa piense como lo haría un sistema operativo real!