



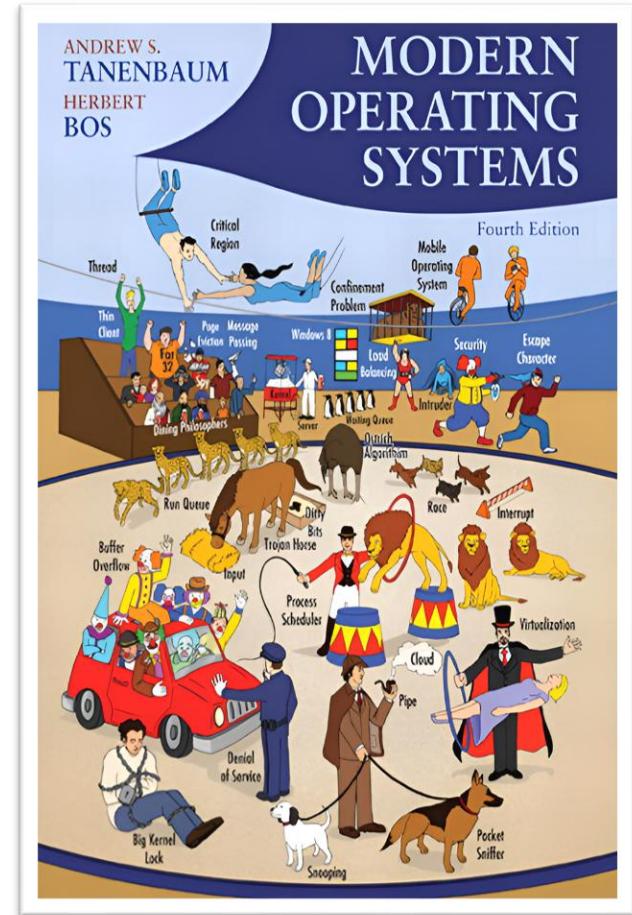
# Sistemas Operativos

Diego Iván Cruz Ordiéres  
2025

# Preliminares

# Libro guía:

Tanenbaum, A. S., & Bos, H. J. (2015). Modern Operating Systems, 4th Edition. Pearson Higher Education.



# Proyecto Final

## Objetivos

- Construir el esqueleto del simulador y un planificador simple (Round Robin).
- Tener una CLI mínima para crear/listar/terminar procesos.
- Implementar el primer algoritmo de reemplazo de páginas (FIFO o LRU) y mostrar estadísticas.
- Visualizar marcos, fallos de página y tasa de aciertos.
- Integrar mecanismos de sincronización: semáforos o mutex.
- Resolver un problema canónico (productor-consumidor o lectores-escritores).

# Proyecto Final

## Actividades en clase

Llevar a cabo el desarrollo de los siguientes puntos:

- Live coding:
  - Modelo de PCB (id, estado, ráfagas).
  - Round Robin con quantum fijo y cola lista.
  - CLI con comandos: new, ps, tick, kill, run n.
- Pruebas: ver tiempos de espera/retorno y fairness.
- Live coding: gestor de marcos + tabla de páginas por proceso.
- Implementación de FIFO y trazas de acceso; registrar: fallos totales, PFF puntual (si se mide), tiempo por acceso.
- Visualización por consola o pequeño tablero ASCII/CLI.

# Proyecto Final

## Actividades en clase

Llevar a cabo el desarrollo de los siguientes puntos:

- Mini-framework de semáforos/mutex (o usar std::mutex/condition\_variable para la simulación).
- Implementar productor-consumidor sobre un buffer simulado por el SO y exponerlo vía CLI (comandos produce, consume, stat).

# Proyecto Final

## Actividades en clase

¿Qué se espera al final de la clase?

- Planificador RR funcional + CLI básica.
- Métricas iniciales (tiempo de espera/promedio en un set pequeño).
- Algoritmo de paginación 1 integrado (FIFO o LRU) con métricas y visualización básica.
- Módulo de sincronización operativo + demo reproducible (script de comandos).

# Proyecto Final

## Fundamentos, alcance y arquitectura

### Actividad

Llevar a cabo el desarrollo de los siguientes puntos:

- Agregar SJF no expropiativo como segundo algoritmo de CPU (requisito: “al menos dos”).
- Documentar en /docs diseño del planificador y supuestos.
- Implementar algoritmo de paginación 2 (LRU o PFF/Working Set) para cumplir “al menos dos”.
- Gráficas simples (CSV → notebook o script) de fallos vs. tamaño de marcos.
- Documentar el diseño de sincronización y sus invariantes.
- Ensayar la sustentación parcial (5 minutos) con video corto o guion.

