

**Universidad Autónoma de Tamaulipas**

**Facultad de Ingeniería Tampico**

**Sistemas Operativos**

**Muñoz Quintero Dante Adolfo**

**Ing. Sistemas Computacionales**

**Barahona Romero Billy Antonio**

**Espinosa Vázquez Cristopher**

**González Gonzales Oswaldo**

**Polanco Tijerina José Luis**

**Ponce Medina Jesús**

**6° “J”      2025-3**

**Fecha: 02/12/2025**

# Manual de Usuario - Simulador de Gestor de Procesos

## 1.- Introducción

El Simulador de Planificación de Procesos permite visualizar cómo un sistema operativo gestiona la planificación, administración de recursos y sincronización entre procesos utilizando memoria compartida y exclusión mutua (mutex). Este entorno gráfico permite crear procesos, observar sus estados, monitorear recursos de CPU y memoria, y ejecutar una demostración del problema Productor-Consumidor.

### Requisitos del Sistema

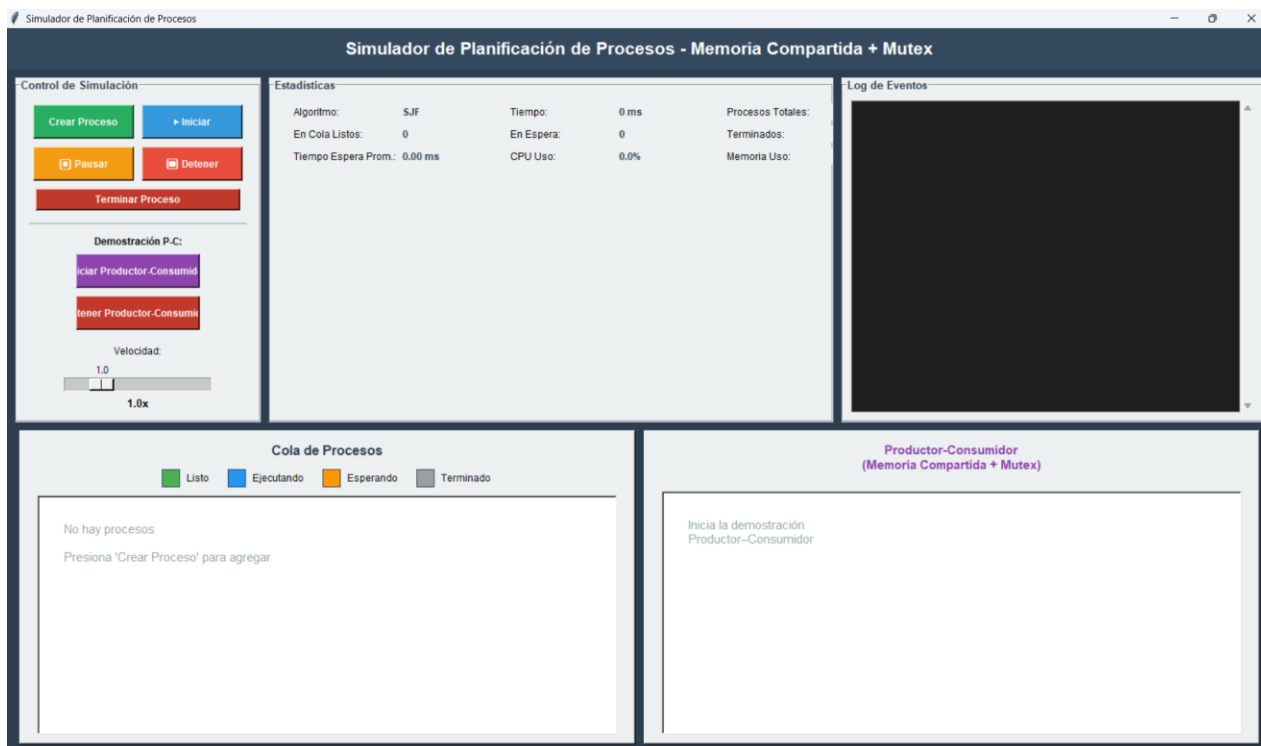
- Hardware:
  - 4 GB de RAM
  - Procesador Dual Core
- Software:
  - Python 3.x
  - Librería Tkinter (incluida en Python)
  - Sistema operativo Windows.

### Instalación y Ejecución

1. Descargar o clonar el proyecto.
2. Abrir una terminal en la carpeta del proyecto.
3. Ejecutar el comando:
4. `python main.py`
5. La interfaz principal aparecerá automáticamente.

## 2.- Interfaz Principal

La interfaz está dividida en varios paneles:

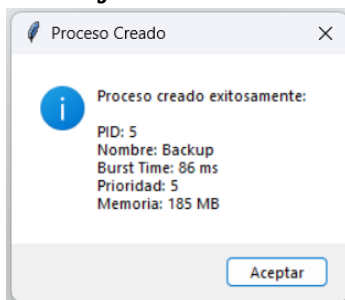


- **Control de Simulación:** Permiten crear, iniciar, pausar, detener y terminar procesos.

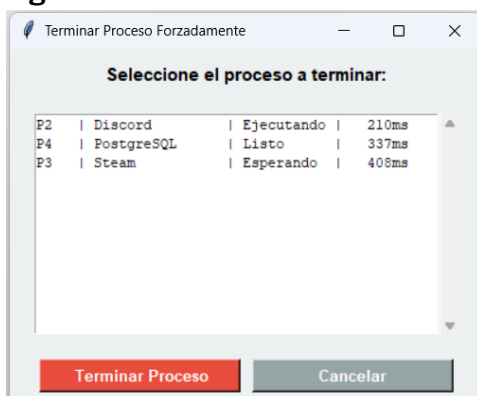


A excepción de “Iniciar”, “Pausar” y “Detener” Los demás botones de “Control de Simulación” si tienen otras interacciones, como las siguientes.

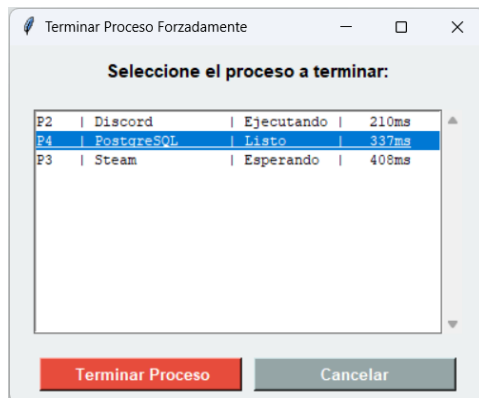
- **Crear**, cuando presionamos el botón “Crear Proceso” nos muestra el siguiente mensaje donde viene información del proceso. Solo damos en “Aceptar”.



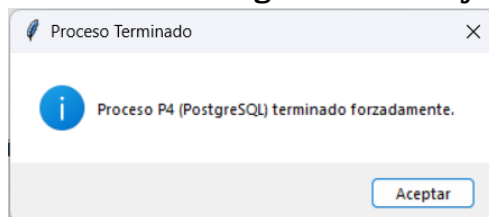
- **Terminar Proceso**, cuando presionamos el botón “Terminar Proceso” nos muestra la siguiente ventana.



En la cual podemos seleccionar un proceso y dar en “Terminar Proceso” para cerrarlo forzosamente.



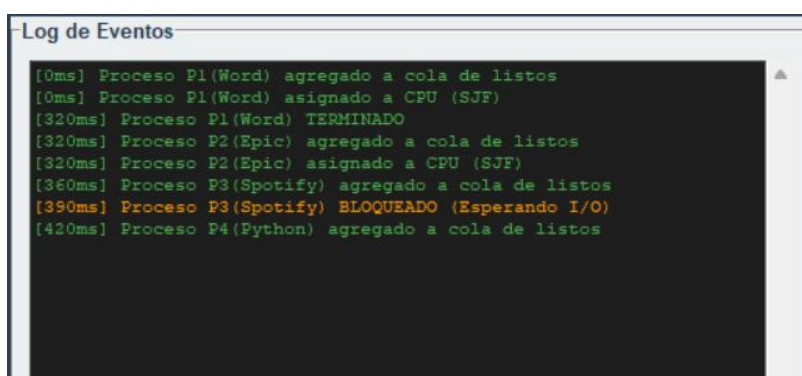
Nos muestra el siguiente mensaje al cual simplemente le daremos en “Aceptar”.



- Los botones de “Iniciar Productor-Consumidor” y “Detener Productor-Consumidor” se explicarán en la sección ‘Panel Productor-Consumidor’.
- La barra de Velocidad nos permite visualizar de manera más rápida o lenta como los procesos cambian de estado. La barra tiene 6 velocidades “0.5, 1.0, 1.5, 2.0, 2.5, 3.0”
- **Estadísticas:** Muestra información como CPU usada, memoria usada, procesos terminados, etc.

Estadísticas			
Algoritmo:	SJF	Tiempo:	420 ms
En Cola Listos:	3	En Espera:	3
Tiempo Espera Prom.:	-167.50 ms	CPU Uso:	0.0%
		Procesos Totales:	
		Terminados:	
		Memoria Uso:	

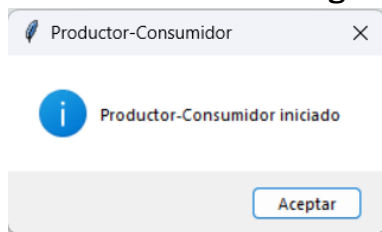
- **Log de Eventos:** Registra cada acción del simulador.



- **Cola de Procesos:** Visualiza procesos Listos(verdes), Ejecutando(azul), Esperando(amarillo) o Terminados(gris).



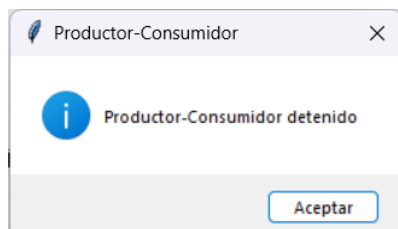
- **Panel Productor-Consumidor:** Muestra el buffer, mutex e interacción entre procesos.
  - Cuando presionas el botón “Iniciar Productor-Consumidor” del panel “Control de Simulación” envía el siguiente mensaje en el cual presionaremos el botón “Aceptar”.



- Mostrará el siguiente panel.



- Para terminarlo debemos ir al panel “Control de Simulación” y presionar el botón “Detener Productor-Consumidor” y darle en “Aceptar”.



## En Resumen

1. Crear Proceso: Genera un proceso con nombre, ráfaga y prioridad aleatorios.
2. Iniciar: Comienza la ejecución del planificador.
3. Pausar / Detener: Controla la ejecución.
4. Terminar Proceso: Permite finalizar manualmente un proceso.
5. Iniciar Productor-Consumidor: Activa la demostración con memoria compartida y mutex.

6. Ajustar Velocidad: Modifica la rapidez de la simulación.

### **3.- Interpretación de Resultados**

Cada proceso avanza de estado según la planificación (SJF o Prioridad). Los procesos pueden bloquearse, esperar recursos o ser terminados. Los colores representan los estados, y el sistema actualiza estadísticas en tiempo real.

### **4.- Conclusiones**

El simulador proporciona una herramienta visual potente para comprender los conceptos fundamentales de SO como: planificación de procesos, uso de CPU/memoria, sincronización con mutex y comunicación mediante memoria compartida.