Manual de Usuario y Documentación del Proyecto de Gestión de Tareas

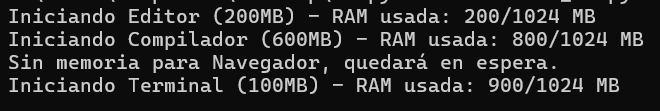
# 1. Introducción

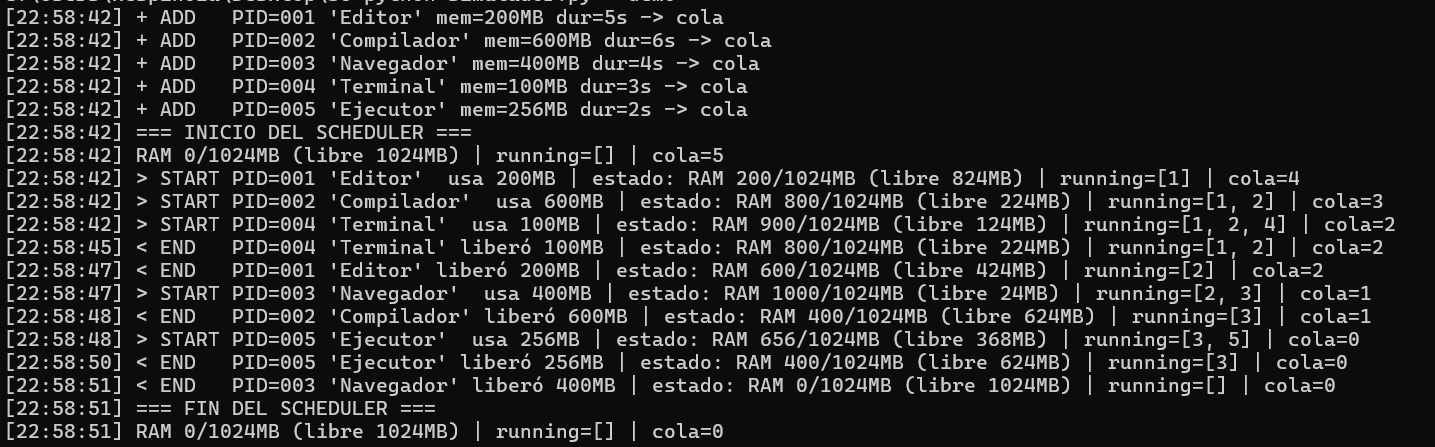
Este documento describe el funcionamiento del sistema de gestión de tareas en Python. El proyecto tiene como objetivo ejecutar múltiples procesos en paralelo o en cola, a partir de archivos JSON que contienen las tareas definidas por el usuario.  
  
A lo largo de las versiones, se han incorporado mejoras en la gestión de procesos, manejo de errores, compatibilidad con diferentes modos de ejecución y una opción de demostración (--demo).

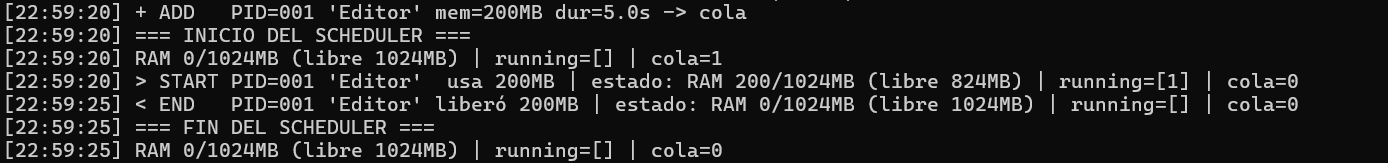
# 2. Librerías utilizadas

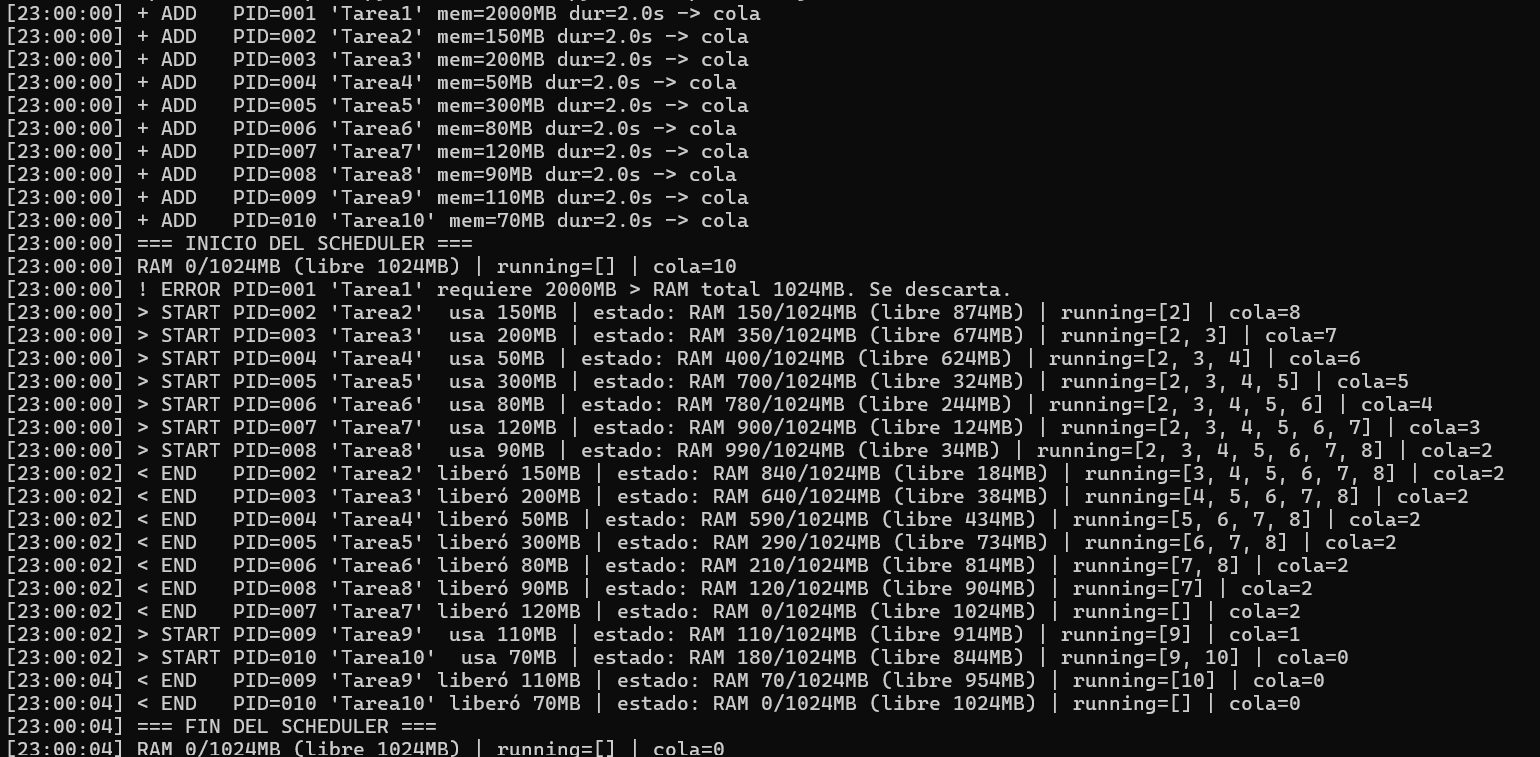
- json: para la lectura de archivos JSON donde se definen las tareas.  
- time: para simular la duración de cada tarea mediante la función sleep().  
- argparse: para procesar los argumentos de línea de comandos (--demo, --file).  
- multiprocessing: para ejecutar múltiples procesos en paralelo o en cola.  
- os: para gestión básica del sistema operativo.  
- python-docx: para la generación de este documento README en formato Word.

# 3. Descripción del código

El código principal está dividido en tres versiones:  
  
➤ Versión 1:  
 - Ejecuta tareas definidas en un archivo JSON.  
 - Simula cada tarea con una pausa (sleep).  
 - Procesos en paralelo.  


➤ Versión 2:  
 - Se agregó compatibilidad con el argumento --demo, que ejecuta un ejemplo predefinido.  
 - Se mejoró el manejo de procesos para que queden en cola sin mostrar advertencias.  


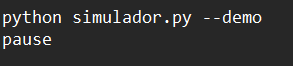
  
➤ Versión 3:  
 - Permite cargar un archivo JSON con múltiples tareas.  
 - Ejemplo incluido con 10 tareas de 2 segundos cada una.  
 - Mantiene compatibilidad con los modos anteriores (--demo y --add).

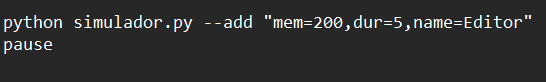


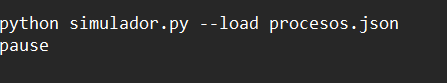
➤ Versión 4:  
 - Validaciones de procesos menores a la capacidad de memoria   
 - Mantiene compatibilidad con los modos anteriores (--demo y –add y --load).

# 4. Pasos para ejecutar el proyecto

1. Instalar Python 3.x en el sistema.  
2. Guardar el código principal en un archivo llamado main.py.  
3. Crear un archivo JSON (por ejemplo, tareas.json) con las tareas a ejecutar.  
4. Abrir la terminal y ubicarse en la carpeta del proyecto.  
5. Ejecutar el script según la modalidad:  
  
 - Ejemplo en modo demo:  
 python simulador.py –demo

  
 - Ejemplo agregar procesos:  
 python main.py --add python simulador.py



- Ejemplo con archivo JSON:  
 python simulador.py --load tareas.json  


# 5. Ejemplo de archivo JSON

Un archivo JSON válido para este proyecto puede ser:  
  
[  
 {"id": 1, "nombre": "Tarea 1", "duracion": 2},  
 {"id": 2, "nombre": "Tarea 2", "duracion": 2},  
 {"id": 3, "nombre": "Tarea 3", "duracion": 2}  
]  
  
Cada objeto representa una tarea con un identificador, un nombre descriptivo y la duración en segundos.

# 7. Conclusiones

El sistema de gestión de tareas en Python permite ejecutar múltiples procesos de forma sencilla, simulando cargas de trabajo definidas en un archivo JSON. Las diferentes versiones del código muestran una evolución progresiva en términos de flexibilidad, compatibilidad y facilidad de uso.