Reporte técnico: Proyecto final de Sistemas Operativos y Laboratorio

---

1. Información del Proyecto

Título del Proyecto: Ingeniería del Caos en Sistemas Operativos: Simulación y Análisis de Resiliencia en Entornos Distribuidos
Curso/Materia: Sistemas Operativos
Integrantes:

Emmanuel Bustamante Valbuena (emmanuel.bustamante@udea.edu.co)

Sebastian Amaya Perez (sebastian.amaya1@udea.edu.co)

Fecha de Entrega: 29 de mayo de 2025

---

2. Introducción

2.1. Objetivo del Proyecto

El objetivo del proyecto es construir un entorno simulado que permita aplicar principios de ingeniería del caos sobre sistemas operativos y arquitecturas distribuidas. Se busca introducir fallos controlados para analizar la resiliencia del sistema y proponer estrategias de mejora.

2.2. Motivación y Justificación

En la actualidad, los sistemas enfrentan fallos constantes y críticos. Empresas como Netflix o Amazon ya usan ingeniería del caos para anticiparse a estos problemas. Este proyecto adopta esa filosofía, permitiendo a estudiantes y desarrolladores experimentar con fallos en un entorno controlado.

2.3. Alcance del Proyecto

El proyecto abarca:

- Simulación de un sistema operativo (planificación de CPU, memoria, archivos).
- Simulación de una arquitectura de microservicios (APIs, bases de datos, red).
- Motor de caos para inyección de fallas.
- Dashboard para análisis de métricas.

Fuera de alcance:

- Sistemas operativos reales o implementaciones sobre hardware.
- Pruebas de resiliencia en entornos de producción reales.

---

3. Marco Teórico / Conceptos Fundamentales

- **Ingeniería del Caos:** Estrategia que consiste en introducir fallos de forma deliberada para observar y mejorar la tolerancia a fallos.
- **Planificación de CPU:** Selección de procesos para ejecución.
- **Microservicios:** Arquitectura distribuida donde los servicios son independientes y comunicados por red.
- **Concurrencia y paralelismo:** Uso de threading y multiprocessing.
- **Análisis de datos:** Visualización y procesamiento con pandas y matplotlib/plotly.

---

4. Diseño e Implementación

4.1. Diseño de la Solución

La arquitectura consta de:

- **Motor de Caos:** Encargado de inyectar fallas y recopilar datos.
- **Simulador SO:** Estructuras que simulan procesos, planificadores y archivos.
- **Simulador MS:** Servicios simulados con fallos de red, base de datos y métricas.
- **Dashboard:** Visualiza latencia, tiempo de recuperación, uso de CPU.

4.2. Tecnologías y Herramientas

- Lenguaje: Python 3.11+
- Librerías: NumPy, Pandas, Matplotlib, Plotly
- Testing: Pytest, Coverage.py
- Documentación: Sphinx, Jupyter, LaTeX

- Persistencia: SQLite, JSON, CSV
- Contenedores: Docker

4.3. Detalles de Implementación

- Se implementaron planificadores Round Robin, FIFO y prioridad.
- Simulador de MS incluye balanceadores de carga y circuit breakers.
- Motor de caos permite inyectar CPU spikes, caídas de servicio y errores aleatorios.
- Se estructuraron pruebas automáticas y generación de reportes desde la línea de comandos.

---

5. Pruebas y Evaluación

5.1. Metodología de Pruebas

Se usó un enfoque basado en pruebas unitarias y simulaciones completas. Se desarrollaron casos específicos para observar fallos intencionales y reacciones del sistema.

5.2. Casos de Prueba y Resultados

ID	Descripción	Resultado Esperado	Resultado Obtenido	Éxito/Fallo
CP-001	Caída de microservicio	Activación de fallback	Fallback activo en <1s	Éxito
CP-002	Sobreuso de CPU por proceso malicioso	Detención del proceso	Planificador lo excluye	Éxito
CP-003	Error en base de datos simulada	Retry con delay y logging	3 intentos, logging registrado	Éxito

5.3. Evaluación del Rendimiento

- Tiempo promedio de recuperación: < 800 ms
- Latencia promedio simulada: < 1 s

- Uso de CPU con caos activo: 35%-70% (según carga)

5.4. Problemas Encontrados y Soluciones

- Problemas de sincronización de procesos: solucionado usando threading.Lock
- Fallas de visualización en dashboard: se migró de matplotlib a plotly

---

6. Conclusiones

- Se logró implementar un sistema completo de simulación para ingeniería del caos.
- Se validaron estrategias de resiliencia en ambientes simulados.
- Aprendimos sobre planificadores, gestión de fallos, concurrencia y monitoreo.
- El proyecto cumple con los objetivos propuestos y abre camino para aplicaciones reales en docencia e industria.

---

7. Trabajo Futuro

- Integrar inteligencia artificial para toma de decisiones ante fallos.
- Soporte para clústers distribuidos reales (Kubernetes).
- Implementar simulación de red más detallada (latencia, pérdida de paquetes).

---

8. Referencias

- https://netflixtechblog.com/chaos-engineering-upgraded-878d341f15fa
- https://principlesofchaos.org/
- https://aws.amazon.com/fis/
- Rosenthal, C., & Jones, N. *Chaos Engineering*. O'Reilly Media, 2020.
- Newman, S. *Building Microservices*. O'Reilly Media, 2021.