

# Proyecto Final - Curso de Sistemas Operativos y Laboratorio

## Título del proyecto

Ingeniería del Caos en Sistemas Operativos: Simulación y Análisis de Resiliencia en Entornos Distribuidos

## Miembros del equipo

Emmanuel Bustamante Valbuena

Sebastian Amaya Perez

## Resumen

El proyecto propone el desarrollo de un sistema compuesto por dos simuladores: uno a nivel de sistema operativo y otro a nivel de microservicios, integrados mediante un motor de ingeniería del caos. El objetivo es generar fallas controladas en entornos simulados para analizar su comportamiento y mejorar la resiliencia de sistemas distribuidos. Se busca entregar una solución robusta, educativa y práctica, que permita validar hipótesis, mejorar la tolerancia a fallos y formar en prácticas modernas de sistemas resilientes.

## Introducción

¿Cuál es la necesidad y/o problema que aborda el desafío seleccionado?

Actualmente, los sistemas informáticos son vulnerables a múltiples tipos de fallos: hardware, red, errores de software, picos de tráfico, etc. Más del 70% de las empresas han reportado fallas críticas, y el costo por hora de inactividad puede superar los \$5.6 millones de dólares.

¿Por qué es importante el desarrollo de este desafío en el contexto tecnológico actual?

La ingeniería del caos se ha posicionado como una práctica esencial en empresas líderes como Netflix y Amazon, ayudando a diseñar sistemas más robustos y resilientes. Introducir fallas controladas permite anticipar problemas y mejorar la calidad, estabilidad y confiabilidad del software en producción.

## Antecedentes o marco teórico

¿Cuáles son los principales aspectos teóricos necesarios para comprender el desafío y llevarlo a cabo?

- Fundamentos de sistemas operativos: planificación de CPU, manejo de memoria, sistemas de archivos, concurrencia.
- Arquitectura de microservicios: servicios distribuidos, APIs, balanceo de carga, bases de datos, tolerancia a fallos.

- Ingeniería del caos: diseño de experimentos, observación de métricas, aprendizaje del comportamiento del sistema ante errores inducidos.

¿Qué relación tiene esta teoría con los temas del curso de Sistemas Operativos?

El simulador de SO trabaja directamente con conceptos vistos en clase como planificación de procesos, manejo de recursos, y sistemas de archivos. A nivel de laboratorio, se experimenta con concurrencia (threading, multiprocessing) y análisis de rendimiento bajo carga y fallos.

## Objetivos (principal y específicos)

### Objetivo principal

Desarrollar un entorno de simulación que permita aplicar prácticas de ingeniería del caos sobre sistemas operativos y microservicios para evaluar su resiliencia y mejorar su diseño.

### Objetivos específicos

- Implementar un simulador básico de sistema operativo con fallos inducidos.
- Desarrollar un simulador de microservicios distribuidos con componentes críticos.
- Integrar ambos simuladores bajo un motor de caos para análisis conjunto.
- Visualizar métricas de comportamiento y estrategias de recuperación.
- Validar resultados mediante casos prácticos y documentación técnica.

¿Qué implementación propone realizar?

Un sistema compuesto por:

- Simulador de SO (CPU, memoria, archivos, procesos).
- Simulador de microservicios (servicios, red, base de datos).
- Motor de inyección de caos.
- Dashboard de visualización.
- Documentación y manual de usuario.

## Metodología

¿Cuáles son las principales herramientas que podrían utilizarse para implementar la solución?

- Python 3.11+
- Módulos de concurrencia: threading, multiprocessing
- Análisis: NumPy, Pandas
- Visualización: Matplotlib, Plotly
- Almacenamiento: SQLite, JSON
- Testing: pytest, coverage.py
- Documentación: Sphinx, LaTeX, Jupyter

¿Qué actividades son necesarias para cumplir los objetivos?

1. Investigación teórica y diseño de arquitectura.
2. Desarrollo del simulador básico de SO.
3. Implementación del motor de inyección de fallos.
4. Desarrollo del simulador de microservicios.
5. Integración de todos los componentes.
6. Diseño de dashboard y visualización de métricas.
7. Elaboración de la documentación técnica.
8. Pruebas y validación final del sistema.
9. Presentación del proyecto.

### Cronograma

Semana	Actividad	Entregable
1	Investigación, diseño y arquitectura del sistema	Documento de arquitectura
2	Desarrollo del simulador de SO y sus algoritmos	Simulador básico + Algoritmos
3	Implementación del motor de caos para el SO	Demo 1: Simulador SO funcionando
4	Desarrollo del simulador de microservicios	Demo 2: Microservicios
5	Integración, visualización, documentación y presentación	Proyecto final + Manual + Reporte

### Referencias

1. Netflix Technology Blog. Chaos Engineering. <https://netflixtechblog.com/chaos-engineering-upgraded-878d341f15fa>
2. Principles of Chaos Engineering. <https://principlesofchaos.org/>
3. Amazon Web Services. AWS Fault Injection Simulator. <https://aws.amazon.com/fis/>
4. Google Cloud. Chaos Engineering at Scale. <https://cloud.google.com/blog/products/devops-sre/chaos-engineering-at-scale>
5. Rosenthal, C., & Jones, N. Chaos Engineering: Building Confidence in System Behavior through Experiments. O'Reilly Media, 2020.
6. Newman, S. Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, 2021.