



# Sistemas Distribuidos I (75.74)

## Amazon Books Analyzer HA

TP Tolerancia a Fallos

### Docentes

- Pablo D. Roca
- Gabriel Robles
- Franco Barreneche
- Tomás Nocetti
- Nicolás Zulaica



# Requerimientos Funcionales

- Se solicita un sistema distribuido que analice las reseñas a libros en el sitio de Amazon para proponer campañas de marketing.
- Las reseñas poseen título del libro, texto del comentario y rating. Por cada título de libro, se conoce categoría, fecha de publicación y autores.
- Se debe obtener:
  - Título, autores y editoriales de los libros de categoría "Computers" entre 2000 y 2023 que contengan 'distributed' en su título.
  - Autores con títulos publicados en al menos 10 décadas distintas
  - Títulos y autores de libros publicados en los 90' con al menos 500 reseñas.
  - 10 libros con mejor rating promedio entre aquellos publicados en los 90' con al menos 500 reseñas.
  - Títulos en categoría "Fiction" cuyo sentimiento de reseña promedio esté en el percentil 90 más alto.



# Requerimientos No Funcionales

- Para construir una simulación realista se define la serie de datos:
  - <https://www.kaggle.com/datasets/mohamedbakhet/amazon-books-reviews>
- Se permite el uso de librerías de procesamiento de sentimientos tales como TextBlob - NLTK (Python), Sentiment - GoVader (Go)
- El sistema debe estar optimizado para entornos multicomputadoras
- Se debe soportar el incremento de los elementos de cómputo para escalar los volúmenes de información a procesar
- Se requiere del desarrollo de un Middleware para abstraer la comunicación basada en grupos.
- Se debe soportar una única ejecución del procesamiento y proveer *graceful quit* frente a señales SIGTERM.



# Nuevos Requerimientos

- El sistema debe permitir la ejecución de múltiples análisis de clientes en paralelo si ser reiniciado.
- Los clientes deben enviar el conjunto completo de datos a analizar, permitiendo futuras extensiones en las consultas implementadas por el sistema.
- El sistema debe mostrar alta disponibilidad hacia los clientes
- El sistema debe ser tolerante a fallos por caídas de procesos
- En caso de usar un algoritmo de consenso, el mismo tiene que ser implementado por los alumnos
- Se puede utilizar docker-in-docker para levantar procesos caídos
- No está permitido utilizar la API de docker para verificar si un nodo está disponible.



Se espera del alumno:

- Empleo del tiempo de consultas en clase para resolver dudas y clarificar el negocio del sistema a construir previo a su diseño
- Exposición y verificación en clase de la arquitectura propuesta antes de iniciar su implementación
- Empleo del grupo de correos para realizar consultas que no pudieran ser resueltas en clase
- Consideración de prácticas distribuidas según lo estudiado en clase para elaborar una arquitectura flexible, escalable y robusta
- Aprobación del cuerpo docente para el uso de cualquier librería.
- Demo del sistema en funcionamiento previamente ensayada



- Fecha de entrega: 27/06/2024
- Formato de entrega:
  - Entrega digital mediante correo personal.
  - Demostración del sistema en funcionamiento.
  - Documento de arquitectura 4+1 Views o C4Model incluyendo al menos: diagramas de robustez, despliegue, actividades, paquetes, secuencia y DAG.