

# Laboratorio 2

## Objetivos

1. Comprender y aplicar tecnologías clave en el procesamiento distribuido, específicamente Apache Kafka para la gestión de mensajes y Apache Spark para el procesamiento de datos en clúster.
2. Desarrollar un estudio de caso basado en un sistema distribuido real, identificando y justificando modelos de consistencia y políticas de replicación adecuados para el caso de uso seleccionado.
3. Implementar una maqueta de los esquemas de consistencia propuestos.

## Contexto

En la actualidad, el software está cada vez más integrado con los procesos de negocio en empresas de alto impacto en diversas industrias. Esto conlleva un aumento en la complejidad de las arquitecturas, haciendo que los aspectos fundamentales del diseño distribuido se alineen estrechamente con las necesidades y restricciones del negocio.

Es por esto que para este laboratorio se les pide elegir un caso de estudio para analizar, proponer una solución arquitectónica para un aspecto funcional clave (no para todo el sistema) e implementar un modelo de consistencia acorde a las restricciones funcionales usando Spark y Kafka.

Al momento de elegir un caso de estudio, deben restringirse a aplicaciones o negocios donde claramente se requiera un sistema distribuido para dar solución al problema de negocio (por ejemplo: Netflix, Twitch e Instagram son elegibles; LOA no). Es recomendable que consulten con el profesor, mediante un correo electrónico corto o un mensaje instantáneo, el caso de estudio y el contexto funcional que se quiere abordar. Es fundamental que el contexto funcional seleccionado tenga una relevancia significativa para el negocio. Por ejemplo, la gestión de transacciones en un sistema financiero como Visa presenta desafíos más relevantes para este laboratorio que, por ejemplo, la simple gestión de cuentas de usuario.



Luego, el análisis y propuesta de arquitectura debe estar sujeto a este contexto de negocio y funcional. Aparte de los elementos claves que debe tener una arquitectura y que fueron revisados en clase, es crucial que los modelos de consistencia y las políticas de manejo de réplicas estén claramente definidos y rigurosamente justificados en relación con los requisitos funcionales del caso de estudio.

Esta primera parte se verá reflejada en un informe, cuya nota será considerada para la calificación final de cátedra.

La segunda parte de este laboratorio consiste en implementar una maqueta del modelo de consistencia propuesto usando Apache Spark y Apache Kafka. La maqueta debe demostrar claramente el modelo de consistencia implementado a través de los resultados obtenidos, permitiendo la observación y análisis de cómo se mantienen o comprometen las garantías de consistencia bajo diferentes condiciones.

## Indicaciones

Para el desarrollo del laboratorio, tengan en cuenta las siguientes consideraciones:

1. El laboratorio debe realizarse en grupos de una o dos personas - no más.
2. La calificación final de este laboratorio depende de un repositorio y de un informe, que deben ser entregados al profesor y ayudante dentro de los plazos establecidos.
3. Los repositorios deben ser públicos una vez terminado el plazo de entrega, o deben invitar al profesor y ayudante al repositorio privado. De lo contrario, este componente de la evaluación será considerado con la nota mínima.
4. Se realizarán descuentos por malas prácticas de programación y por mala redacción u ortografía.
5. El repositorio debe incluir un archivo README que contenga las instrucciones de instalación, en conjunto con las versiones de los componentes de software utilizados.
6. Los diseños de arquitectura deben considerar tanto el diseño de software como el diseño de sistemas. Estos deben estar correctamente justificados (por qué creen que es el mejor enfoque al problema y las fuentes que lo respaldan).
7. El plazo de entrega termina a las 23:59 hrs de la fecha de entrega especificada.



8. El procesamiento y la comunicación entre nodos procesadores debe realizarse usando Spark y Kafka.

El informe a presentar debe considerar al menos los siguientes puntos:

1. Introducción: una breve descripción del documento e introducción al contexto del problema.
2. Caso de estudio: Una descripción del problema en estudio y el contexto funcional.
3. Arquitectura propuesta: en esta sección, se describe la arquitectura propuesta con todos sus aspectos clave. Se espera el uso de diagramas relevantes.
4. Metodología: esta sección del informe debe detallar la metodología utilizada para obtener los resultados presentados. Recuerden que sus resultados deben ser reproducibles, mucho de lo cual depende del nivel de detalle presentado en esta sección. Se deben considerar configuraciones de hardware y archivos de entrada, cantidad de ejecuciones para la obtención de métricas, etc.
5. Resultados y discusión: se presentan y discuten brevemente los resultados obtenidos. Es apropiado describir las posibles motivaciones de los resultados, como costos en los que incurre una solución y no otra, etc.
6. Conclusiones: se presentan conclusiones en base a los resultados obtenidos. Aquí es apropiado considerar otros factores de análisis relevantes para una implementación real de una solución de este tipo: escalabilidad (de todo tipo), costos organizacionales y ventajas y desventajas. También es apropiado incluir una subsección sobre Limitaciones y trabajo futuro.

***La calificación se obtendrá mediante un promedio simple entre la nota del código y el informe. Ambas calificaciones deben ser mayor o igual a 4.0, de lo contrario, la calificación final será el mínimo entre ambas.***

**Fecha de entrega: domingo 25 de agosto.**

