

# Implementación de Algoritmo de Consenso para Elección de Líder

Proyecto N1

ST0263-Tópicos Especiales en Telemática

23 de septiembre de 2024

## 1. Introducción

En el contexto de los sistemas distribuidos, la coordinación y la consistencia entre los distintos procesos son aspectos fundamentales para el correcto funcionamiento de una aplicación. En este sentido, cuando múltiples procesos colaboran para almacenar, procesar y replicar datos, es necesario que uno de ellos actúe como líder para coordinar las operaciones entre ellos. El líder es responsable de recibir y aplicar las solicitudes de los clientes, garantizar la consistencia de los datos replicados y coordinar a los procesos que actúan como seguidores o followers.

Sin embargo, los sistemas distribuidos están expuestos a fallos de red, caídas de procesos y otros tipos de errores, lo que puede causar la indisponibilidad del líder. En estos casos, la elección de un nuevo líder de manera rápida y segura es fundamental para asegurar la tolerancia a fallos y la disponibilidad continua del sistema. Esta elección debe ser realizada de forma consensuada por todos los procesos restantes, garantizando que solo un proceso asuma el rol de líder en cualquier momento.

Algoritmos de consenso como Raft y Paxos han sido diseñados específicamente para resolver este problema. Estos algoritmos aseguran que, a pesar de fallos en uno o varios procesos, el sistema pueda continuar operando correctamente sin perder datos ni comprometer su consistencia. En sistemas modernos, estos algoritmos son esenciales para garantizar la robustez y disponibilidad de aplicaciones distribuidas, como bases de datos, sistemas de archivos distribuidos y entre otro tipo de servicios.

Este proyecto tiene como objetivo que los estudiantes implementen uno de estos algoritmos o propongan una solución propia para manejar la elección de líder en un entorno distribuido de base de datos, proporcionando una experiencia práctica en el diseño de sistemas tolerantes a fallos.

## 2. Descripción

En este proyecto, se tiene un sistema compuesto por cinco procesos que interactúan en el contexto de una base de datos relacional distribuida. La estructura del sistema es la siguiente:

- **Proceso 1:** Es una aplicación cliente que se comunica con la base de datos para realizar consultas y modificaciones. Las operaciones de escritura se realizan en el líder y las operaciones de lectura se hacen contra los followers.
- **Proceso 2:** Es un proxy a nivel de aplicación que intercepta las solicitudes de la aplicación cliente y las dirige a los procesos encargados del almacenamiento y la replicación de datos.
- **Proceso 3:** Es el **líder** del sistema de base de datos. Este proceso coordina las operaciones de escritura, asegurando la consistencia de los datos.
- **Procesos 4 y 5:** Son **followers**. Estos procesos mantienen réplicas del estado de la base de datos y responden a las solicitudes del líder para replicar las actualizaciones.

## 3. Requisito

Se debe garantizar la **tolerancia a fallos** en el sistema. En caso de que el **proceso líder (Proceso 3)** falle, uno de los followers debe asumir el rol de líder para asegurar la continuidad del servicio. El sistema debe ser capaz de detectar la falla del líder y realizar la **elección de un nuevo líder** de manera automática y coordinada entre los followers restantes.

## 4. Separación de los Planos de Datos y de Control

Una parte clave de este proyecto es la separación entre el **plano de datos** y el **plano de control**, lo que garantizará que las operaciones de manejo de datos y la coordinación entre procesos se mantengan organizadas y desacopladas.

- **Plano de datos:** Corresponde a la comunicación que ocurre entre la **aplicación cliente (Proceso 1)** y los procesos de la base de datos, así como la replicación de datos entre el **líder (Proceso 3)** y los **followers (Procesos 4 y 5)**. Este plano debe manejar las peticiones de lectura y escritura que se ejecutan en la base de datos, garantizando que se procesen de manera eficiente y consistente.
- **Plano de control:** Es responsable de gestionar la comunicación de coordinación entre los **followers** y el **líder**, incluyendo la elección de un nuevo líder en caso de fallas. Además, debe notificar al **proxy de aplicación (Proceso 2)** sobre la identidad del nuevo líder una vez que haya sido elegido, para que el proxy pueda redirigir las solicitudes de la aplicación cliente al nuevo líder. Este plano

debe funcionar de manera independiente al plano de datos, asegurando que la coordinación entre los procesos no interrumpa el flujo de datos.

## 5. Simulación de Fallos

Además de la implementación del algoritmo de consenso, los estudiantes deberán **simular fallos** en el sistema. Para ello, se deben cumplir los siguientes requisitos:

- **Fallo del líder:** Se debe simular la caída (inactividad) del proceso líder (Proceso 3). El sistema debe detectar automáticamente esta falla e iniciar el proceso de **elección de un nuevo líder**.
- **Fallo de followers:** Se debe permitir la simulación de fallas en uno o más followers (Procesos 4, 5 o 6). El sistema debe seguir funcionando, siempre que exista una mayoría de procesos operativos. Los estudiantes deberán demostrar que el sistema sigue siendo consistente y disponible en estas condiciones.
- **Reincorporación de procesos:** Los procesos que fallaron deberán poder reincorporarse al sistema, ya sea como seguidores o líderes, sin comprometer la consistencia de los datos.
- **Pruebas de consistencia:** Durante la simulación de fallos, el sistema debe garantizar que no se pierdan solicitudes del cliente ni se comprometa la consistencia de los datos en los followers.

## 6. Objetivo

El objetivo de este proyecto es que los estudiantes implementen un **algoritmo de consenso** que resuelva el problema de la elección de un nuevo líder en el caso de una falla del proceso líder actual. Se pueden usar las siguientes estrategias:

- Implementar el algoritmo de consenso **Raft**.
- Implementar el algoritmo de consenso **Paxos**.
- Proponer e implementar una solución propia que resuelva el problema de elección de líder en un entorno distribuido.

## 7. Requerimientos

- El sistema debe ser capaz de detectar la falla del líder de manera automática.
- Se debe asegurar que solo un proceso sea elegido como líder en cualquier momento, garantizando la consistencia en la replicación de la base de datos.

- El nuevo líder debe asumir su rol sin perder las solicitudes del cliente y sin comprometer la consistencia de los datos.
- **Simulación de fallos:** Los estudiantes deberán demostrar que su sistema responde correctamente ante la simulación de caídas de procesos.

## 8. Entregables

El proyecto tendrá una duración total de **tres semanas** y se deberá cumplir con los siguientes entregables:

### 1. Primer punto de chequeo (al final de la semana 1):

- **Marco teórico:** Presentación del marco teórico sobre algoritmos de consenso y la elección de líder en sistemas distribuidos, incluyendo Raft y Paxos.
- **Especificaciones de comunicación:** Definición final de la arquitectura del sistema, los protocolos y APIs para la comunicación entre los diferentes procesos, incluyendo la separación entre el plano de datos y el plano de control.
- **Diseño del sistema:** El equipo deberá entregar un diseño detallado de la arquitectura del sistema, con los ajustes necesarios.

### 2. Segundo punto de chequeo (entrega final al final de la semana 3):

- **Implementación completa:** El sistema debe estar completamente implementado con el algoritmo de consenso elegido, y debe ser capaz de simular fallos, detectar la caída del líder y realizar una nueva elección.
- **Informe final:** Incluir un análisis detallado de las pruebas realizadas, simulación de fallos y resultados.
- **Código fuente:** Implementación del algoritmo de consenso y su integración con los procesos del sistema.
- **Documentación técnica:** Explicación detallada del algoritmo elegido (Raft, Paxos o solución propia), incluyendo los desafíos enfrentados y cómo se garantiza la consistencia y la disponibilidad en el sistema.
- **Informe de pruebas:** Evidencia de las pruebas realizadas que demuestren la funcionalidad del sistema ante la falla del líder y la correcta elección de un nuevo líder. El informe debe incluir la simulación de fallos y cómo el sistema se comporta en estas situaciones.

## Evaluación

La evaluación se basará en los siguientes criterios:

- Funcionamiento completo de la solución.

- Correcta implementación del algoritmo de consenso.
- Eficiencia en la detección de fallos y elección del nuevo líder.
- Claridad y justificación de la solución implementada.
- Robustez del sistema ante fallos, demostrada a través de las simulaciones.