



UNIVERSIDAD TECNICA  
FEDERICO SANTA MARIA



Departamento de Informática  
Universidad Técnica Federico Santa María

## **Informe: Exclusión Mutua Distribuida v/s Exclusión Mutua Centralizada**

Nombres: Javier Martínez Sabathier 201673530-5  
Bryan Masias Tripailaf 201773108-7



## Introducción

En el presente informe se presentarán, y analizarán, dos algoritmos de exclusión mutua. Estos se usarán para presentar, aceptar y, eventualmente, rechazar propuestas sobre la distribución de chunks de libros en un sistema distribuido. Este sistema trata de emular una biblioteca virtual, en la que un cliente puede descargar y subir libros en formato pdf, los cuales se dividirán en varios chunks de 250KB de tamaño. El sistema consiste de varias partes, las cuales se detallarán a continuación:

- **Cliente:** es quien interactúa con el sistema. Hay de dos tipos; el uploader, quien se encarga de subir los libros al sistema, y el downloader, quien descargará los libros disponibles.
- **Data Node:** es la parte del sistema que se encarga de guardar los chunks de los libros.
- **Name Node:** se encarga de mantener un LOG que indica donde está cada parte de un libro.

Los algoritmos usados serán los de Exclusión Mutua Distribuida y el de Exclusión Mutua Centralizada. En el primero, la propuesta de distribución debe ser aceptada por todos los Data Nodes, mientras que, en el segundo, la propuesta debe aceptarla el Name Node. Además, si hay más de un nodo tratando de escribir en el LOG al mismo tiempo, el primer algoritmo deberá usar el algoritmo de Ricart y Agrawala para resolver el conflicto, mientras que en el segundo será el Name Node quien resolverá el conflicto.



## Implementación

Tanto el Name Node como los Data Node son servidores gRPC que tienen implementada la lógica para el envío de mensajes de acuerdo a lo que los clientes vayan solicitando.

El Cliente Downloader solicita al Name Node la lista de libros, la cual es enviada como un *string* a través de gRPC. De acuerdo a esta, el cliente elige algún libro, y envía su elección al Name Node. Este, como respuesta, envía un *string* con toda la información de los chunks del libro, Cada chunk tiene el formato "NombreChunk#DireccionIP", y son separados por un salto de línea. Con este *string*, el Cliente se conecta a cada Data Node y envía el nombre del chunk, a lo que el Data Node en cuestión responde enviando el chunk en formato []byte. Este, luego de recibir todos los chunks, reconstruye el archivo original y borra los chunks.

Cliente Uploader solicita seleccionar que libro desea enviar, este libro se divide en chunks de 250 [kB] y es enviado a un DataNode mediante Protocol Buffers. Además, este último permite seleccionar una propuesta ya sea Exclusión Mutua Distribuida o Exclusión Mutua Centralizada.

Ambos algoritmos no fueron capaces de implementarse, pero esto era lo que se buscaba realizar:

Exclusión Mutua Distribuida, el DataNode tras recibir la propuesta, le envía a todos los demás DataNode la propuesta mediante Protocol Buffers, tras esto se queda en esperando que todos los pares respondan, si encuentra algún conflicto lo debe comunicar mediante Protocol Buffers, si no posee conflictos se le envía la propuesta al NameNode, esta se registra en el LOG de NameNode. Luego de esto se distribuyen los chunks respectivamente.

La exclusión Mutua Centralizada, tras recibir la propuesta, esta es enviada al NameNode mediante Protocol Buffers, luego el DataNode se queda en espera de la aprobación del Name Node, este último se encarga de buscar si no posee conflictos, en caso de no poseer conflictos se registra la distribución de chunks en el NameNode, luego en respuesta al DataNode le envía la distribución a este, luego DataNode distribuye los chunks dada la distribución dada anteriormente.



## Comparación de algoritmos y conclusión

Debido a la poca cantidad de libros, ambos sistemas se comportan de manera similar, esto no sería así si tuviéramos numerosas peticiones realizadas por distintos nodos, donde debido a la alta cantidad de solicitudes se produciría un cuello de botella.

Una ventaja que además vemos en el algoritmo distribuido es que el retraso de sincronización es solo el tiempo de transmisión de nuestro mensaje, mientras que el otro algoritmo debe esperar el mensaje tanto de ida como otro de vuelta.

Por lo tanto, el algoritmo más rápido y que usaría menos mensajes si usamos una gran cantidad de libros sería el algoritmo distribuido por lo anterior mencionado.