



- Sistemas Distribuidos -

# **Comparativa en implementación de Algoritmos de Exclusión mutua para sistema distribuido de biblioteca**

José Follert & Harold Melo

### Que se hizo y cómo se hizo:

Se realizó la implementación de un sistema distribuido que conecte clientes, nodos de almacenamiento de datos y un nodo directorio, en donde los primeros sean capaces de descargar o cargar archivos pdf a los segundos, siendo guiados por el tercero durante el proceso. Para generar esto se hizo uso de Go y GRPC, generando conexiones sincrónicas entre todos los entes, permitiendo así su comunicación mediante Protocol Buffer, y el intercambio de los paquetes necesarios.

Notar que el objeto de estudio se da cuando dos entidades requieren del directorio a la vez, y es posible resolver esto mediante un algoritmo en donde el directorio (nameNode) es quien confirma la posibilidad y acepta o rechaza la opción, generando una de reemplazo en caso de no acceder; o un algoritmo en donde la entidad que pide espacio en directorio confirma con todos sus pares que es posible su uso; para mediante el común acuerdo conseguir la condición de aprobada para la escritura en el directorio.

Se adjunta el paso a paso para ejecutar el sistema:

1. Ejecutar los DataNodes en sus respectivas VM.  
> *make Datanode*
2. Ejecutar el NameNode en sus respectiva VM  
> *make Namenode*
3. Ejecutar el cliente en cualquiera de las VM  
> *make Cliente*
4. Al ejecutar el cliente este presenta un menú con las opciones de subir o descargar archivos.

### Resultados

Los resultados que obtuvimos no fueron del todo conclusivos, en particular por el tiempo de desarrollo de este proyecto. Cabe notar que, consiguiendo fuentes de terceros, se nota una leve diferencia en el tiempo de ejecución entre ambos métodos, pero de muy bajo peso.

### Análisis

Es posible notar que, aun subiendo una lista de algunas decenas de libros, el sistema seguirá trabajando con cantidades bajas de clientes y dataNodes, en este caso, la diferencia entre los métodos demuestra ser sutil. Esto se entiende como una baja diferencia entre los mensajes enviados para este fin y poca diferencia en el tiempo hasta haber terminado el procedimiento.

En el caso de aumentar drásticamente los dataNode, es posible encontrar una mayor diferencia entre uno y el otro; principalmente en los mensajes enviados para un sistema de toma de decisiones distribuido, en donde la opción de aceptar propuestas y generar nuevas costará cada vez más recursos, tomando la forma de  $(\text{número de dataNodes})^{(Cte)}$ . En donde con un aumento del número de dataNode podría crecer de manera importante los mensajes que se necesitan

enviar; asumiendo un valor de la constante entre 1 y 1.1 el aumento en los mensajes simularía ser lineal, pero con un valor ligeramente mayor, el resultado crecería con una cota demasiado alta.

## Discusión

Los datos obtenidos no alcanzan a ser concluyentes, debido a su poca diferencia entre los casos, las cuales se pueden dar por motivos tan dispares como lo son la eficiencia del código empleado para la realización del algoritmo. El análisis mayor es cuando vemos el comportamiento de estos algoritmos en casos distintos, variando alguno de los parámetros bases que se nos dio en este caso de estudio.

## Conclusión

Como conclusión ante este caso de estudio, es que si bien en el desarrollo es donde mayor diferencia se puede encontrar entre un método y el otro, para este tipo de instancias es indiferente que formula para obtener la gestión se utilizará. Aunque al momento del análisis es visible que en casos de mayor envergadura si haya una diferenciación evidente entre los métodos. Con ligeras ventajas en los tiempos y mensajes enviados, el algoritmo centralizado demuestra ser más eficiente que el distribuido; cosa a tener en cuenta para posibles puestas en marchas con más actores en simultáneo.