

# Tarea 8: Exclusión mutua y transacciones atómicas

---

## SISTEMAS CONCURRENTES Y DISTRIBUIDOS

---

### Objetivo:

Conocer el manejo de la exclusión mutua en sistemas distribuidos, el manejo de transacciones atómicas y el tratamiento de bloqueos

### Introducción:

En el mundo actual, la tecnología de la información se ha convertido en un pilar fundamental para una amplia variedad de aplicaciones y servicios que utilizamos diariamente. Desde sistemas bancarios hasta redes sociales y aplicaciones en la nube, los sistemas distribuidos están en el corazón de estas tecnologías. Para garantizar que estos sistemas funcionen de manera efectiva, confiable y segura, es esencial comprender y aplicar conceptos clave, como los algoritmos de elección, las transacciones atómicas y el control de concurrencia.

### Actividades a realizar

#### Exclusión mutua:

La exclusión mutua es un concepto fundamental en sistemas informáticos y distribuidos que garantiza que un recurso o sección crítica sea accesible por un único proceso a la vez para evitar conflictos y mantener la integridad de los datos. Para lograr esta exclusión mutua en sistemas distribuidos, se han desarrollado varios tipos de algoritmos.

- Algoritmo Centralizado:** Este enfoque implica la designación de un control central que coordina el acceso a un recurso compartido. Los procesos solicitan el acceso al recurso al control central, que otorga permisos de acceso en función de un protocolo establecido. Un ejemplo común de este enfoque es el uso de un servidor centralizado que administra el acceso a la base de datos.
- Algoritmo Distribuido:** En contraste con el enfoque centralizado, los algoritmos distribuidos permiten que múltiples procesos coordinen sus acciones para lograr la exclusión mutua sin depender de un control central. Estos algoritmos suelen utilizar mensajes y protocolos para gestionar el acceso al recurso compartido de manera distribuida. Ejemplos incluyen el algoritmo de Ricart-Agrawala y el algoritmo de Maekawa.
- Algoritmo de Anillo de Fichas:** En este enfoque, los procesos forman una estructura de anillo, y una "ficha" circula en el anillo. Un proceso que tiene la ficha puede acceder al recurso compartido, y cuando termina, pasa la ficha al siguiente proceso en el anillo. Este método garantiza que solo un

proceso tenga acceso al recurso en un momento dado, y la ficha circula para conceder el acceso a otros procesos en orden. Un ejemplo de este algoritmo es el algoritmo de Peterson.

**Comparación de los Tres Algoritmos:** Cada uno de estos algoritmos tiene sus propias ventajas y desventajas. El algoritmo centralizado es simple de implementar pero puede ser un punto único de fallo. Los algoritmos distribuidos eliminan este punto único de fallo pero pueden ser más complejos de diseñar y pueden requerir más comunicación entre procesos. Los algoritmos de anillo de fichas son eficientes en términos de comunicación, pero pueden ser menos flexibles en situaciones de acceso no predecible. La elección del algoritmo depende de las necesidades específicas del sistema y las consideraciones de diseño.

- **Algoritmos de elección:** son utilizados en sistemas distribuidos para seleccionar un proceso entre varios disponibles, generalmente para asignar una tarea o una responsabilidad particular. Dos de los algoritmos de elección más conocidos son el "Algoritmo del Grandulón" y el "Algoritmo de Anillo". A continuación, se describen brevemente estos dos algoritmos:

1. **Algoritmo del grandulón:** Este algoritmo se utiliza para seleccionar un líder o coordinador dentro de un grupo de procesos distribuidos. Su nombre proviene de la idea de que el proceso "más grandulón" (es decir, el más poderoso o con el mayor número de recursos) tiene la capacidad de asumir el liderazgo. El proceso que inicia la elección envía un mensaje a todos los demás procesos con un ID más alto, y si ninguno de los procesos responde, se convierte en el líder. Si un proceso con un ID más alto responde, se inicia un nuevo proceso de elección. Este algoritmo garantiza la elección de un líder, pero puede requerir una cantidad significativa de mensajes en grandes sistemas.
2. **Algoritmo de anillo:** En este algoritmo, los procesos se organizan en un anillo lógico. La elección se inicia cuando un proceso decide que necesita un líder y envía un mensaje a su sucesor en el anillo. Cuando un proceso recibe este mensaje, puede elegir convertirse en líder o pasar el mensaje a su sucesor. El proceso que decide convertirse en líder envía un mensaje de confirmación al proceso inicial que inició la elección. El proceso que inició la elección espera este mensaje de confirmación. Si no recibe el mensaje de confirmación en un tiempo determinado, asume que ha habido un problema y puede iniciar una nueva elección.

### Transacciones atómicas:

- **Introducción a las Transacciones Atómicas:** Una transacción atómica es un conjunto de operaciones que se considera una unidad indivisible. En otras palabras, o todas las operaciones dentro de una transacción se ejecutan con éxito o ninguna de ellas se ejecuta. Esto garantiza la integridad y consistencia de los datos en sistemas distribuidos, incluso en presencia de fallas. Las transacciones atómicas se basan en el principio ACID, que significa Atomicidad, Consistencia, Aislamiento y Durabilidad.
- **Modelo de Transacción:** El modelo de transacción describe cómo se deben realizar las transacciones en un sistema distribuido. Incluye operaciones como "commit" para confirmar una transacción exitosa y "rollback" para deshacer una transacción en caso de un error. El modelo también aborda cuestiones como la serialización de transacciones, donde se establece un orden en el que se ejecutan las transacciones para evitar problemas de concurrencia.

- **Implementación:** La implementación de transacciones atómicas involucra la programación de aplicaciones y sistemas para seguir el modelo de transacción deseado. Esto puede requerir el uso de mecanismos de control de concurrencia y bloqueo para garantizar que las transacciones no interfieran entre sí y que se cumplan las propiedades ACID.
- **Control de Concurrencia:** El control de concurrencia se refiere a cómo se gestionan múltiples transacciones concurrentes para evitar conflictos y garantizar la consistencia de los datos. En sistemas distribuidos, esto se vuelve aún más crítico, ya que múltiples nodos pueden realizar transacciones simultáneamente. Se utilizan técnicas como el bloqueo, la concurrencia optimista y la serialización para lograr un control de concurrencia eficiente.
- **Bloqueo en Sistemas Distribuidos:** El bloqueo es una técnica común para controlar la concurrencia en sistemas distribuidos. Los nodos pueden adquirir bloqueos en recursos compartidos para evitar que otros nodos los modifiquen simultáneamente. Sin embargo, el bloqueo excesivo puede llevar a problemas de rendimiento y bloqueo mutuo.
- **Detección Distribuida de Bloqueos y Prevención Distribuida de Bloqueos:** La detección distribuida de bloqueos implica identificar y resolver situaciones de bloqueo en sistemas distribuidos. La prevención distribuida de bloqueos busca evitar que ocurran bloqueos en primer lugar, utilizando algoritmos y técnicas para asignar recursos de manera eficiente.

## Conclusión

Los algoritmos de elección son utilizados en sistemas distribuidos para seleccionar un proceso entre varios disponibles, generalmente para asignar una tarea o una responsabilidad particular. Ambos algoritmos tienen sus propias características y ventajas. El Algoritmo del Grandulón es simple y garantiza la elección de un líder, pero puede generar un alto tráfico de mensajes en sistemas grandes. El Algoritmo de Anillo utiliza una estructura de anillo para coordinar la elección y es eficiente en términos de mensajes, pero puede ser más complejo de implementar y puede no ser tan robusto como el Algoritmo del Grandulón en términos de garantizar la elección en todas las circunstancias. La elección del algoritmo depende de las necesidades y restricciones específicas del sistema distribuido.

En resumen, las transacciones atómicas, el control de concurrencia y la gestión de bloqueos son elementos cruciales en la implementación y mantenimiento de sistemas distribuidos confiables y coherentes. Comprender estos conceptos es esencial para diseñar sistemas que puedan manejar transacciones complejas y garantizar la integridad de los datos en entornos distribuidos.

## Bibliografía

- Tanenbaum Andrew. (1995). Sistemas Operativos Distribuidos. España. Prentice-Hall Hisp.
- McIver Ann. (2011). Sistemas Operativos. México. Cengage Learning.
- Tanenbaum, A., & Van Steen M. (2008). Sistemas Distribuidos, Principios y Paradigmas. (Segunda ed.). Prentice Hall.
- Tanenbaum, A. (2011). Redes de Computadoras. (Quinta ed.). Prentice Hall.
- Elmasri, R., Gil Carrick, A., & Levine, D. (2010). Sistemas Operativos, Un enfoque en espiral. McGraw--Hill.

- Canini, M. (2018). Time Synchronization and Logical Clocks
- Nacimiento, M. (2020). Algoritmos de sincronización de relojes. DEV Community.  
<https://dev.to/martinnacimiento/algoritmos-de-sincronizacion-de-relojes-56e8>
- Sánchez Ruiz, D., Castillo Zacatelco, H., Zepeda Cortés, C., de la Rosa Flores, R., Cervantes Márquez, A. P., Limón Martínez, M., & Carballido Carranza, J. L. (2019). Avances en algoritmos de exclusión mutua en sistemas distribuidos. Programación Matemática y Software, 11(1), 15-25.  
Recuperado de <http://www.progmat.uaem.mx:8080/Vol11num1/vol11num1art3.pdf>