



**SISTEMAS DISTRIBUTIVOS**

# **EXCLUSIÓN MUTUA EN SISTEMAS DISTRIBUIDOS**

## EXCLUSIÓN MUTUA EN SISTEMAS DISTRIBUIDOS

En los sistemas distribuidos, múltiples procesos ejecutándose en diferentes nodos pueden requerir acceso a recursos compartidos, tales como archivos, bases de datos, variables de estado o incluso dispositivos físicos. Para garantizar la coherencia y consistencia de dichos recursos, se debe evitar que dos o más procesos accedan simultáneamente a la misma sección crítica del sistema. A este control se le denomina exclusión mutua (Pérez Rojas, 2006).

La exclusión mutua es un mecanismo de coordinación que asegura que, en un instante dado, sólo un proceso tenga acceso al recurso crítico, previniendo condiciones de carrera, corrupción de datos y estados inconsistentes.

### Desafíos de la exclusión mutua en entornos distribuidos

A diferencia de los sistemas centralizados, donde los procesos comparten memoria y reloj, en los sistemas distribuidos los nodos:

- Están geográficamente separados.
- Se comunican exclusivamente a través de mensajes.
- Carecen de una memoria compartida.
- No poseen un reloj global sincronizado.

Estas limitaciones hacen que la exclusión mutua en entornos distribuidos sea mucho más compleja, requiriendo algoritmos específicos que garanticen el acceso exclusivo a recursos sin depender de una visión centralizada del sistema (Pérez Rojas, 2006).

### Requisitos de un algoritmo de exclusión mutua distribuida

Un algoritmo distribuido de exclusión mutua debe cumplir con los siguientes requisitos:

- **Seguridad (mutual exclusion):** en ningún momento más de un proceso puede acceder al recurso crítico.
- **Ausencia de inanición (no starvation):** todo proceso que solicite acceso eventualmente lo obtiene.
- **Ausencia de interbloqueo (no deadlock):** no debe producirse una espera circular entre procesos.
- **Equidad:** el orden de acceso debe respetar, preferentemente, la prioridad temporal o causal.
- **Eficiencia en mensajes:** se busca minimizar la cantidad de mensajes intercambiados por acceso.

### Principales algoritmos de exclusión mutua distribuida

#### 1 Algoritmo de Ricart y Agrawala (1981)

Uno de los algoritmos más conocidos para la exclusión mutua sin token. Utiliza mensajes de solicitud y respuesta para coordinar el acceso entre procesos.

### Funcionamiento:

- Un proceso que desea entrar a la sección crítica, envía un mensaje de solicitud a todos los demás.
- Cada proceso responde con un “OK” si no está usando ni desea usar el recurso.
- El solicitante entra a la sección crítica sólo tras recibir la aprobación de todos.
- Al salir, responde a las solicitudes pendientes.

### Ventajas:

- No requiere reloj global.
- Solo usa mensajes de solicitud y respuesta.

### Desventajas:

- Alta cantidad de mensajes ( $2^*(n-1)$  por acceso).

## 2 Algoritmo de token ring

En este enfoque, los procesos están organizados en un anillo lógico. Un token (mensaje especial) circula entre los procesos. Solo quien posee el token puede entrar a la sección crítica (Pérez Rojas, 2006).

### Funcionamiento:

- El token circula continuamente por el anillo.
- Si un proceso desea entrar a la sección crítica, espera a recibir el token.
- Una vez que sale, pasa el token al siguiente nodo.

### Ventajas:

- Uso eficiente de mensajes en condiciones de baja competencia.
- Evita la inanición.

### Desventajas:

- La pérdida del token requiere un protocolo de regeneración.
- Latencia potencial en sistemas con muchos nodos.

## 3 Algoritmo de Raymond

Este algoritmo combina estructuras jerárquicas con token. Los procesos están organizados en forma de árbol lógico (Muñoz Escoí, 2013). El token se transmite siguiendo caminos mínimos.

### Ventajas:

- Reducción significativa de mensajes comparado con Ricart-Agrawala.
- Menor sobrecarga en grandes sistemas.

### Ejemplo ilustrativo: Ricart-Agrawala

Supóngase un sistema con 3 nodos (A, B y C).

1. El nodo A desea entrar a su sección crítica. Envía solicitud a B y C.
2. B está desocupado y responde con OK. C también responde con OK.
3. A entra a la sección crítica.
4. Durante ese tiempo, B solicita acceso, pero debe esperar la salida de A.
5. Al salir, A responde a las solicitudes pendientes, permitiendo el acceso a B.

Este ejemplo ilustra cómo se evita el acceso concurrente sin necesidad de un controlador centralizado.

**Tabla 1.** Comparación entre enfoques de exclusión mutua

Algoritmo	Mensajes por acceso	Control centralizado	Riesgo de bloqueo	Ventajas
Ricart-Agrawala.	$2(n - 1)$ .	No.	Bajo.	Equidad, sin token.
Token Ring.	1 a 2.	No.	Pérdida de token.	Simplicidad, bajo tráfico.
Raymond.	$\log(n)$ .	Parcial (estructura árbol).	Medio.	Escalabilidad, eficiencia en mensajes.

### Consideraciones prácticas

Al implementar mecanismos de exclusión mutua en entornos reales, se deben tener en cuenta:

- **La confiabilidad de la red:** pérdida de mensajes puede generar bloqueos.
- **La tolerancia a fallos:** los algoritmos deben ser resilientes ante la caída de nodos.
- **La cantidad de procesos:** algunos algoritmos no escalan bien con miles de nodos.
- **El tipo de recurso:** recursos replicados o inmutables pueden requerir sincronización diferente.