



# Construcción de Sistemas Distribuidos

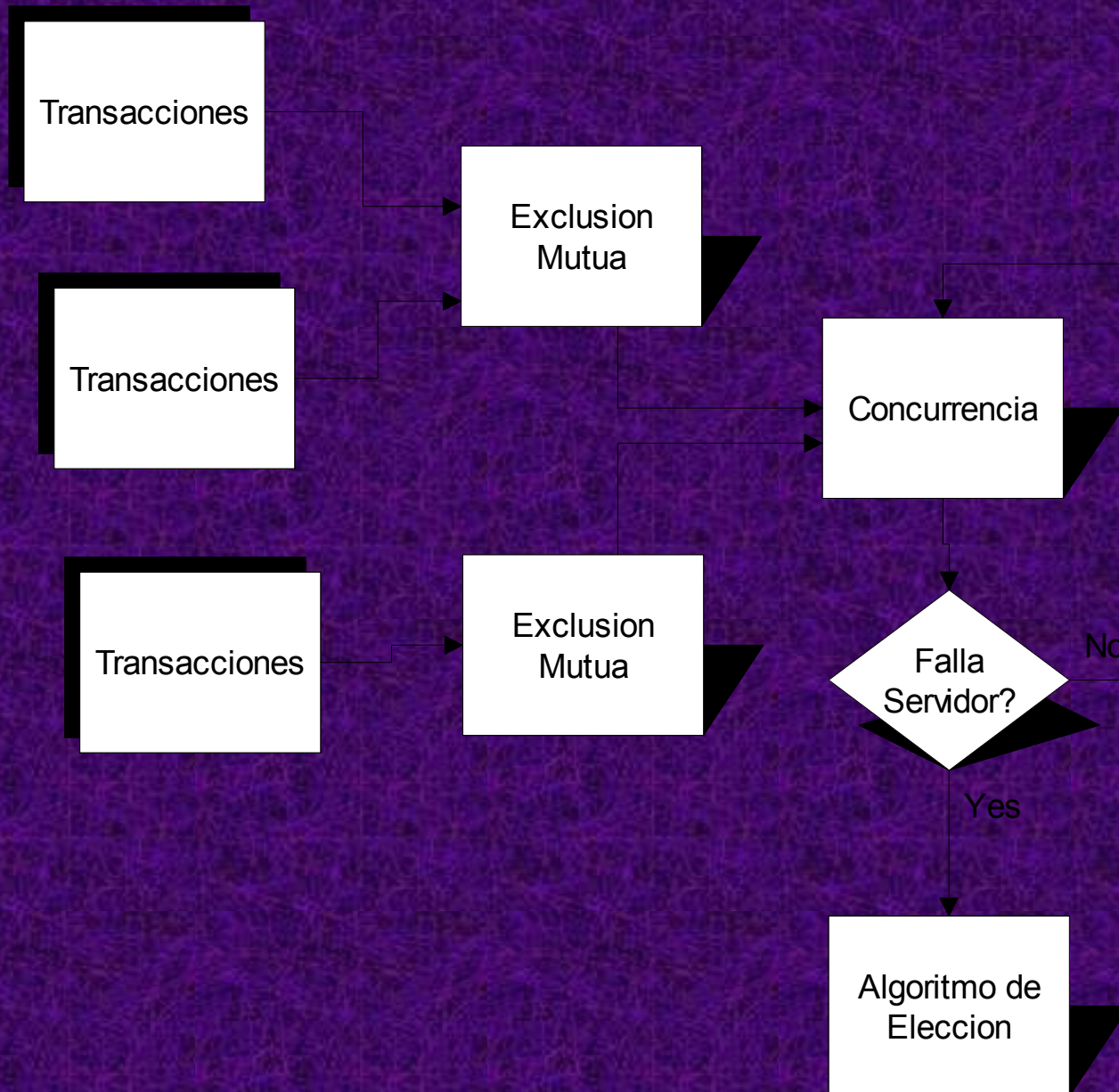
## “Transacciones Distribuidas”

Rogelio Ferreira Escutia



# **Transacciones Distribuidas**

# Transacciones Distribuidas



# ***Transacciones Distribuidas***

---

## **Características:**

### **1) Atómicas:**

**Se hace completa o no se hace**

### **2) Consistentes:**

**Sus datos deben mantener la congruencia entre sus datos**

### **3) Aisladas:**

**Las transacciones concurrentes no interfieren entre sí.**

### **4) Durables:**

**Una vez comprometida la transacción, los cambios son permanentes.**

# *Transacciones Distribuidas*

---

## Primitivas de Transacción:

1) **BEGIN\_TRANSACTION** (inicio de transacción)

1) **END\_TRANSACTION** (fin de transacción)

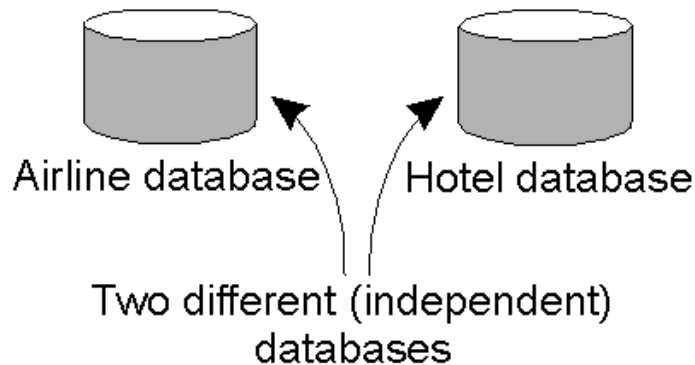
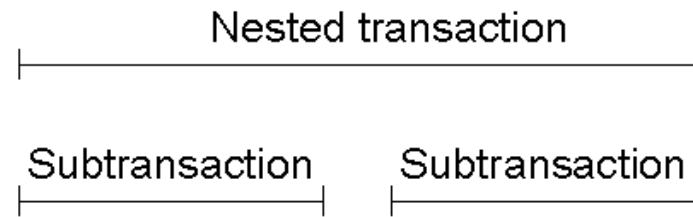
1) **ABORT\_TRANSACTION** (deshacer operación)

1) **READ** (leer datos de un archivo u objeto)

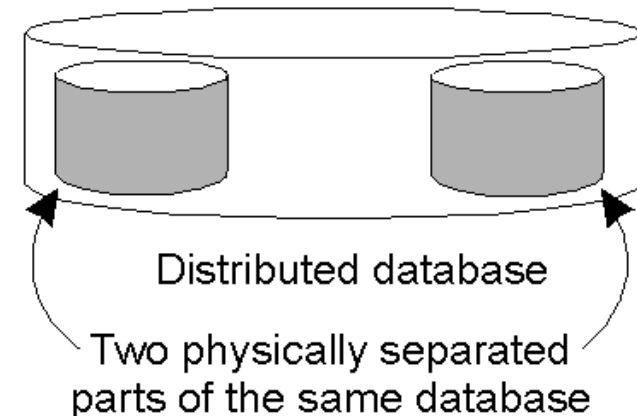
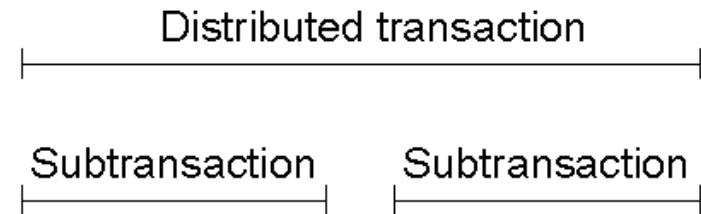
1) **WRITE** (escribir datos a un archivo u objeto)

# Transacciones Distribuidas

## Transacciones Locales y Distribuidas



(a)



(b)

# *Transacciones Distribuidas*

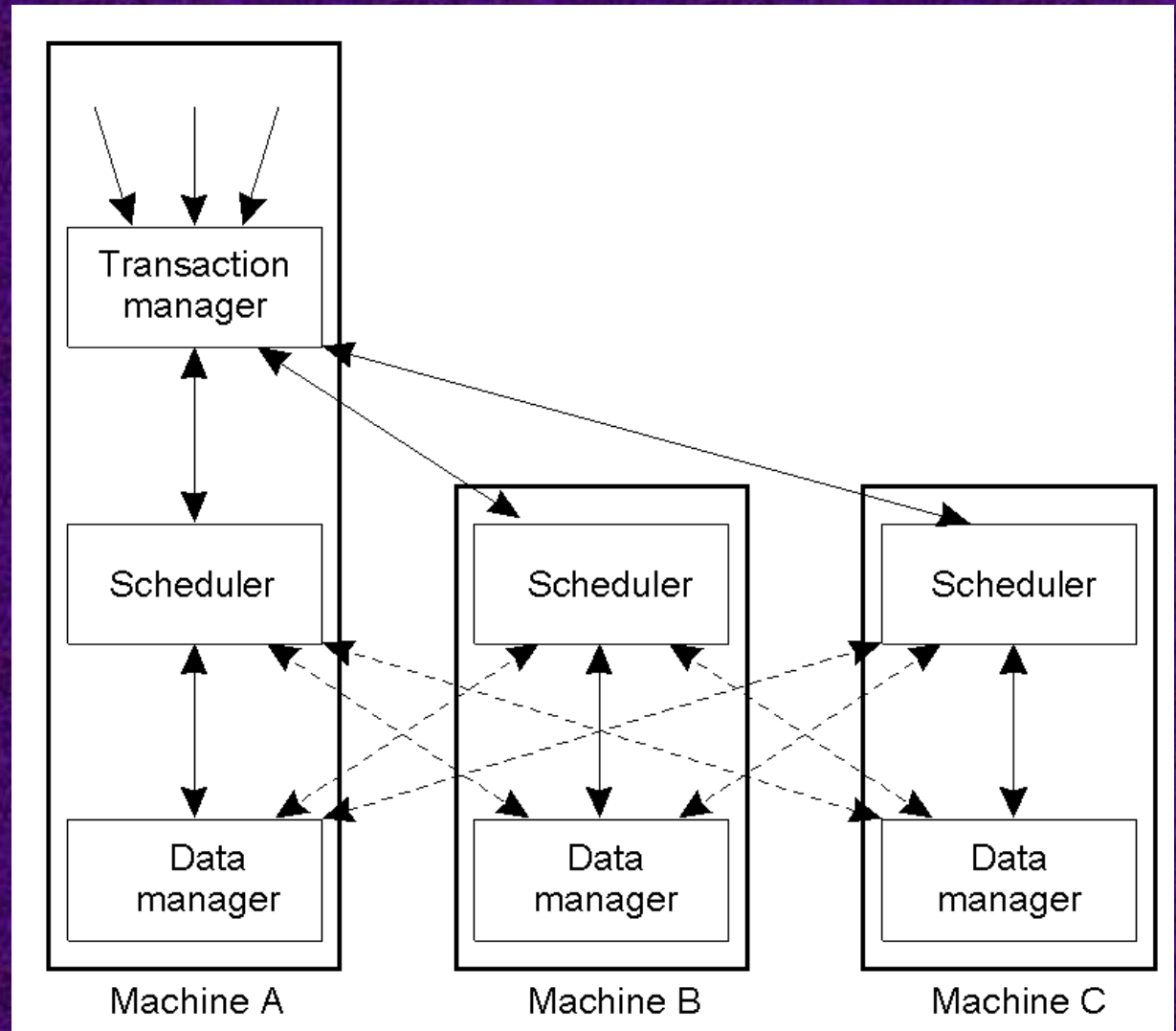
---

## **Bitácora:**

- **Es un archivo que permite deshacer las operaciones realizadas sobre una o varias bases de datos en caso de que falle la transacción.**
- **Esto se hace con el fin de mantener la integridad de la información y que la transacción sea atómica.**

# Transacciones Distribuidas

## Manejador de transacciones





# *Transacciones Distribuidas*

---

## **Bitácora de escritura Anticipada:**

- **También conocida como Lista de Intenciones.**
- **Antes de hacer modificaciones sobre los datos se escribe en la bitácora un registro indicando los valores anteriores y nuevos.**
- **Una vez realizado esto, se modifican los datos, dejando un registro de las operaciones realizadas en la bitácora.**

# *Transacciones Distribuidas*

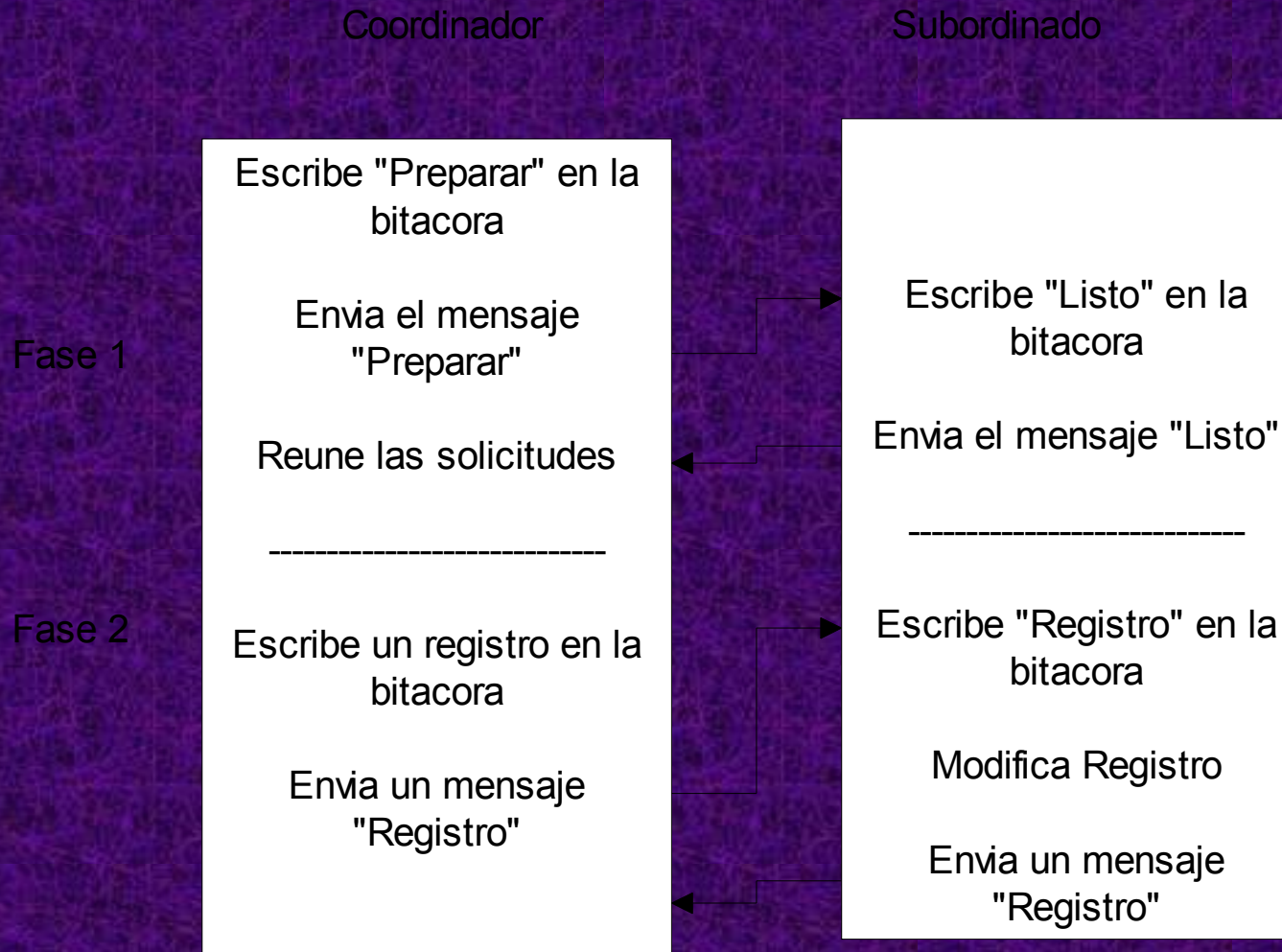
---

## Protocolo de compromiso de 2 fases:

- **Un compromiso es la acción de llevar a cabo una transacción de manera atómica.**
- **En un S.D. Un compromiso puede requerir correr varios procesos en diferentes computadoras para lograr la transacción.**
- **El protocolo para las transacciones más utilizado es el de 2 fases propuesto por Gray en 1978.**

# Transacciones Distribuidas

## Protocolo de compromiso de 2 fases:



# Exclusión Mutua

# *Transacciones Distribuidas*

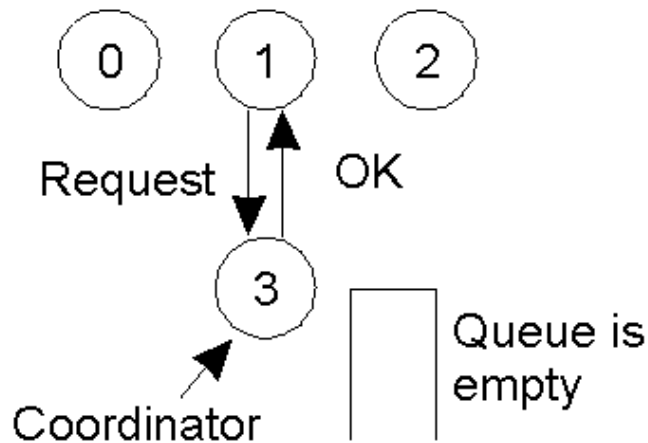
---

## **Exclusión mutua: Centralizada**

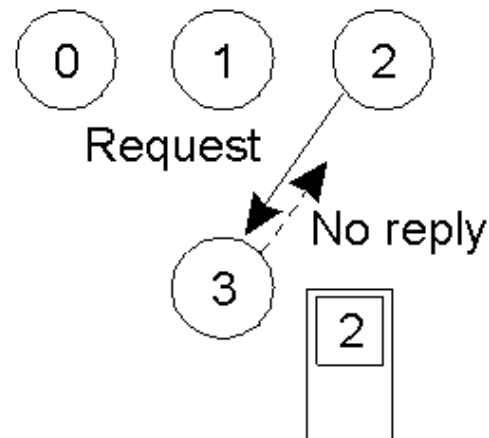
- **Se escoge un proceso como coordinador.**
- **Cuando un proceso requiere entrar a la sección crítica le envía un solicitud al coordinador.**
- **Si no hay petición el coordinador le concede entrar.**
- **Si hay otro proceso que pide entrar, el coordinador selecciona quien entra (puede ser por prioridad, por tiempo, por llegada, etc.)**

# Transacciones Distribuidas

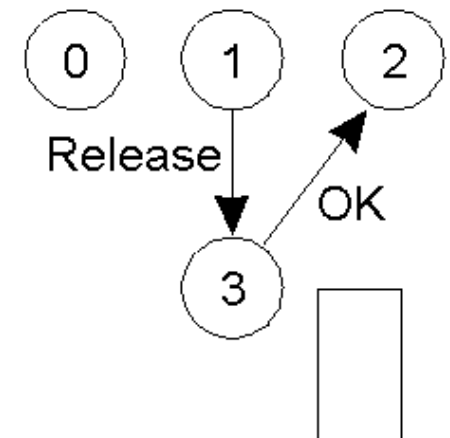
## Exclusión mutua: Centralizada



(a)



(b)



(c)

# *Transacciones Distribuidas*

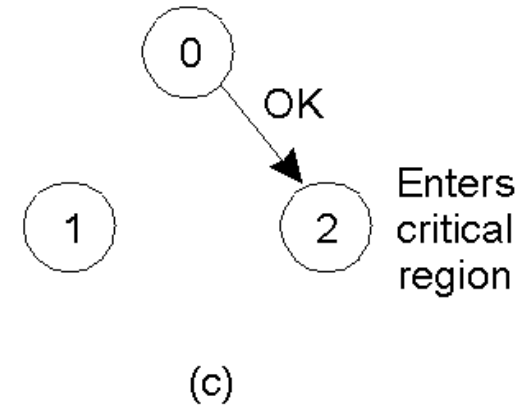
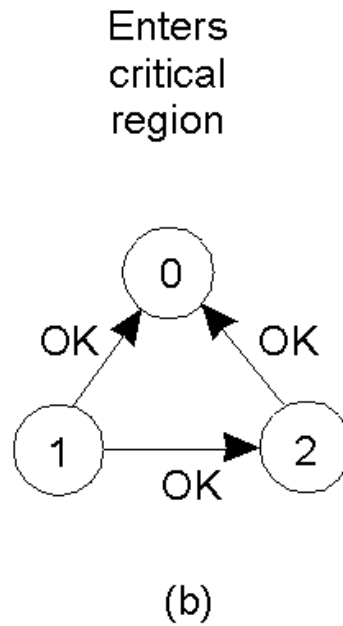
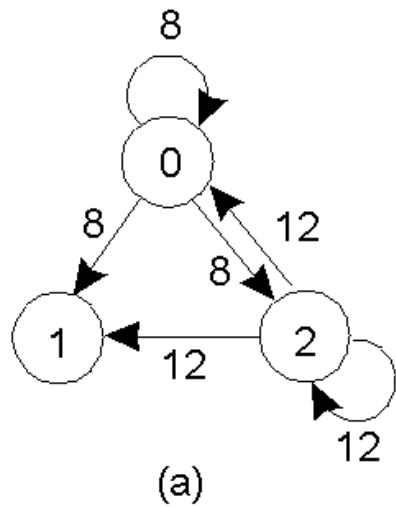
---

## **Exclusión mutua: Distribuida**

- **Un proceso 1 que quiere entrar a la sección crítica envía un mensaje a todos los nodos indicándoles el id de su proceso, fecha/hora y operación.**
- **El proceso 1 debe recibir un OK de los demás procesos para poder entrar.**
- **Si otro proceso 2 desea entrar entonces el proceso 1 recibirá el id del proceso 2 y por medio de la marca de tiempo o prioridad, el proceso 1 entrará o no la sección crítica.**

# Transacciones Distribuidas

## Exclusión mutua: Distribuida





# *Transacciones Distribuidas*

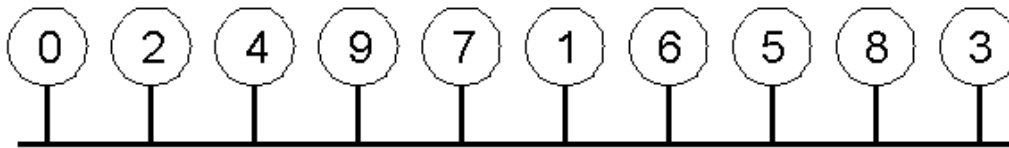
---

## **Exclusión mutua: en Anillo**

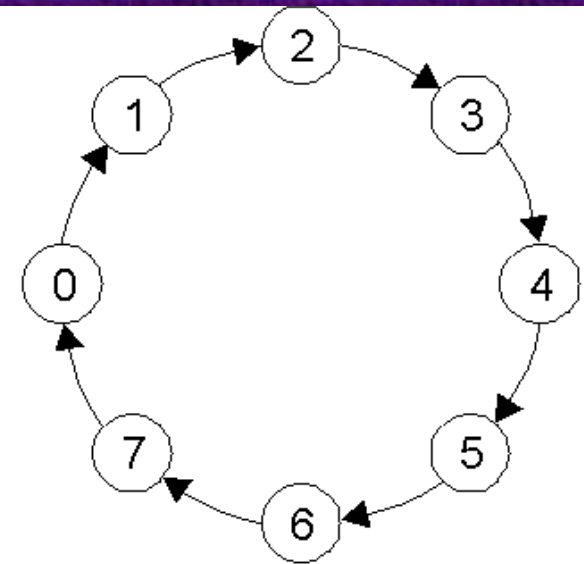
- **Cada proceso tiene una posición dentro del anillo lógico.**
- **Cada proceso sabe quien esta antes y después de él.**
- **Hay sólo una ficha (o token) que se encuentra en el anillo.**
- **El proceso que tiene la ficha es la que entra en la sección crítica.**
- **Si el proceso no desea entrar, pasa la ficha a la siguiente posición y así sucesivamente hasta que se ocupe.**

# Transacciones Distribuidas

## Exclusión mutua: En anillo



(a)



(b)

# *Transacciones Distribuidas*

<b>Algoritmo</b>	<b>Mensajes necesarios</b>	<b>Retraso</b>	<b>Problemas</b>
<b>Centralizado</b>	<b>3</b>	<b>2</b>	<b>Fallo en el coordinador</b>
<b>Distribuido</b>	<b><math>2(n-1)</math></b>	<b><math>2(n-1)</math></b>	<b>Fallo de cualquier proceso</b>
<b>Anillo</b>	<b>Cada 1 a infinito</b>	<b>0 a n-1</b>	<b>Cada</b>

# Concurrencia

# *Transacciones Distribuidas*

## Procesos Concurrentes

Process P1	Process P2	Process P3
x = 1; print ( y, z);	y = 1; print (x, z);	z = 1; print (x, y);

# *Transacciones Distribuidas*

---

## **Control de Concurrencia:**

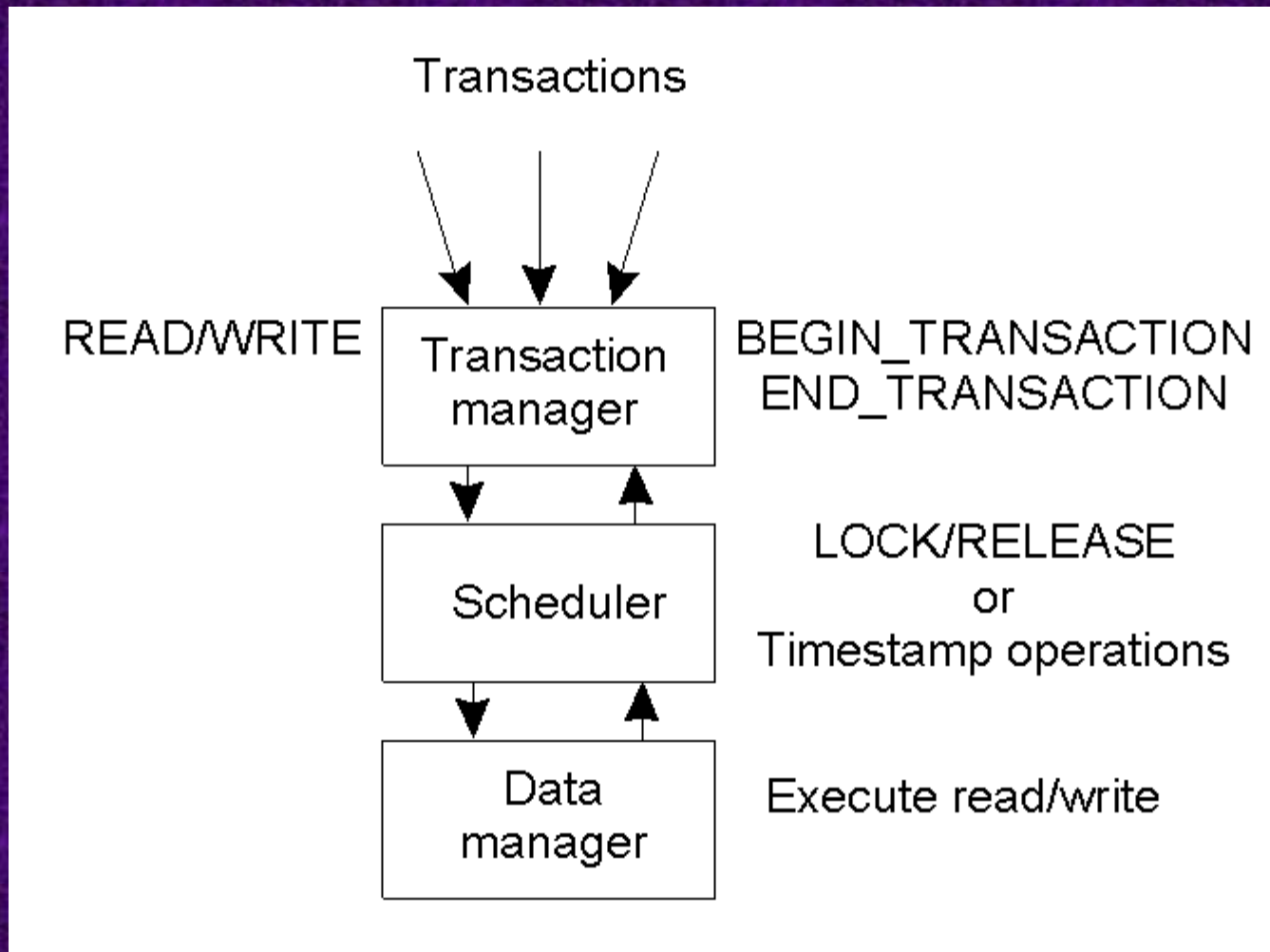
- **Cuando se ejecutan varias transacciones de manera simultánea en distintos procesos se requiere de un algoritmo de control de concurrencia para mantener la sincronía de los procesos.**

## **Los métodos utilizados:**

- **Cerradura**
- **Optimista**
- **Marcas de tiempo**

# Transacciones Distribuidas

## Concurrencia



# *Transacciones Distribuidas*

---

## **Cerradura:**

- **Es el mas antiguo y mas usado.**
- **Cuando un proceso necesita leer o escribir en un archivo como parte de una transacción, primero lo bloquea.**
- **El bloque puede ser a nivel de toda la base, una tabla o un registro.**
- **Para que otros procesos puedan trabajar debe bloquearse solo el registro a modificar.**



# *Transacciones Distribuidas*

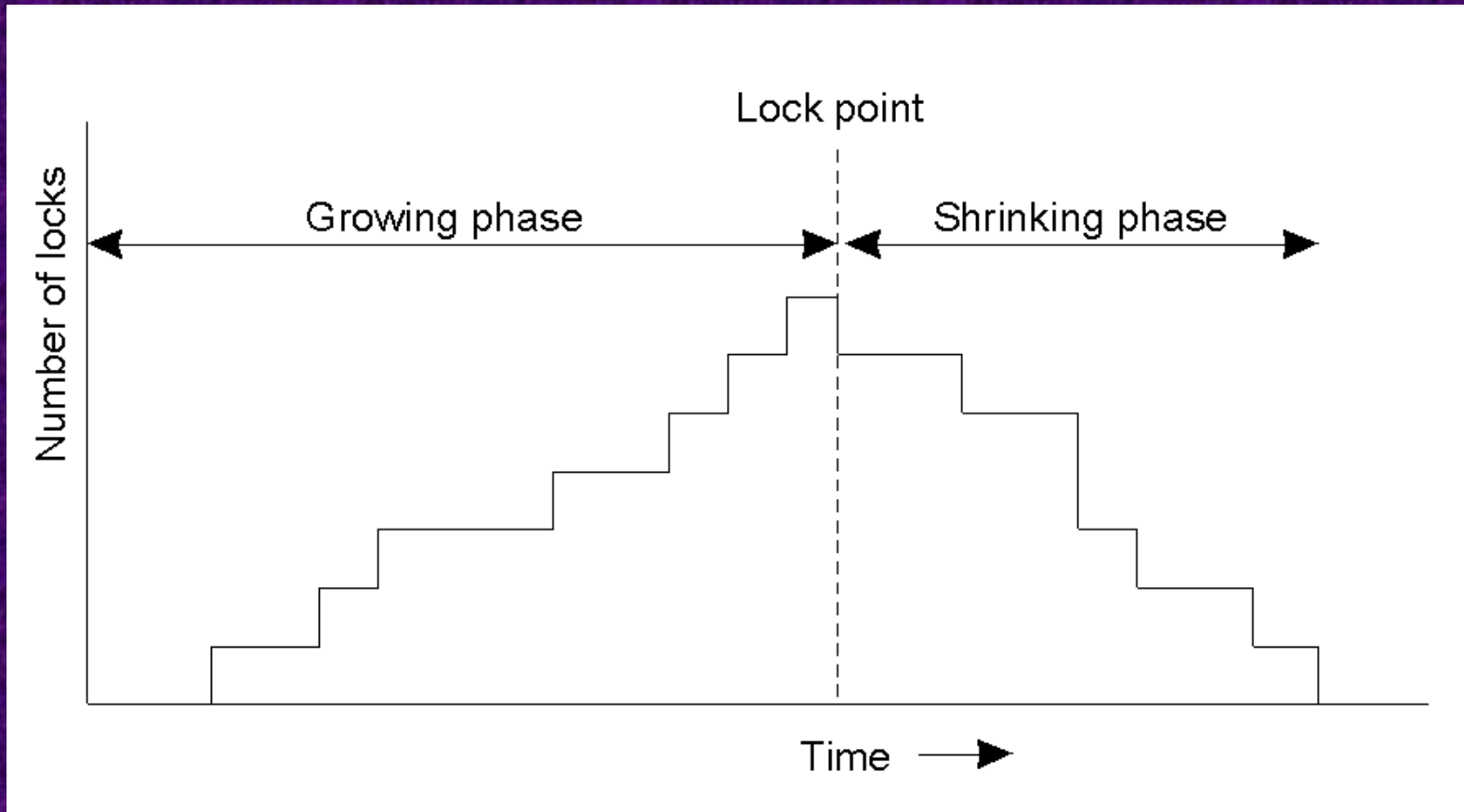
---

## **Cerradura de 2 fases:**

- **Para lograr el bloqueo, primero se van aplicando las cerraduras adecuadas hasta lograr bloquear el / ó los registros necesarios (Primera fase: de crecimiento).**
- **Después se modifica la información.**
- **Por último se liberan las cerraduras (Segunda fase: de contracción).**
- **Mediante esta técnica, todos los procesos son serializables.**

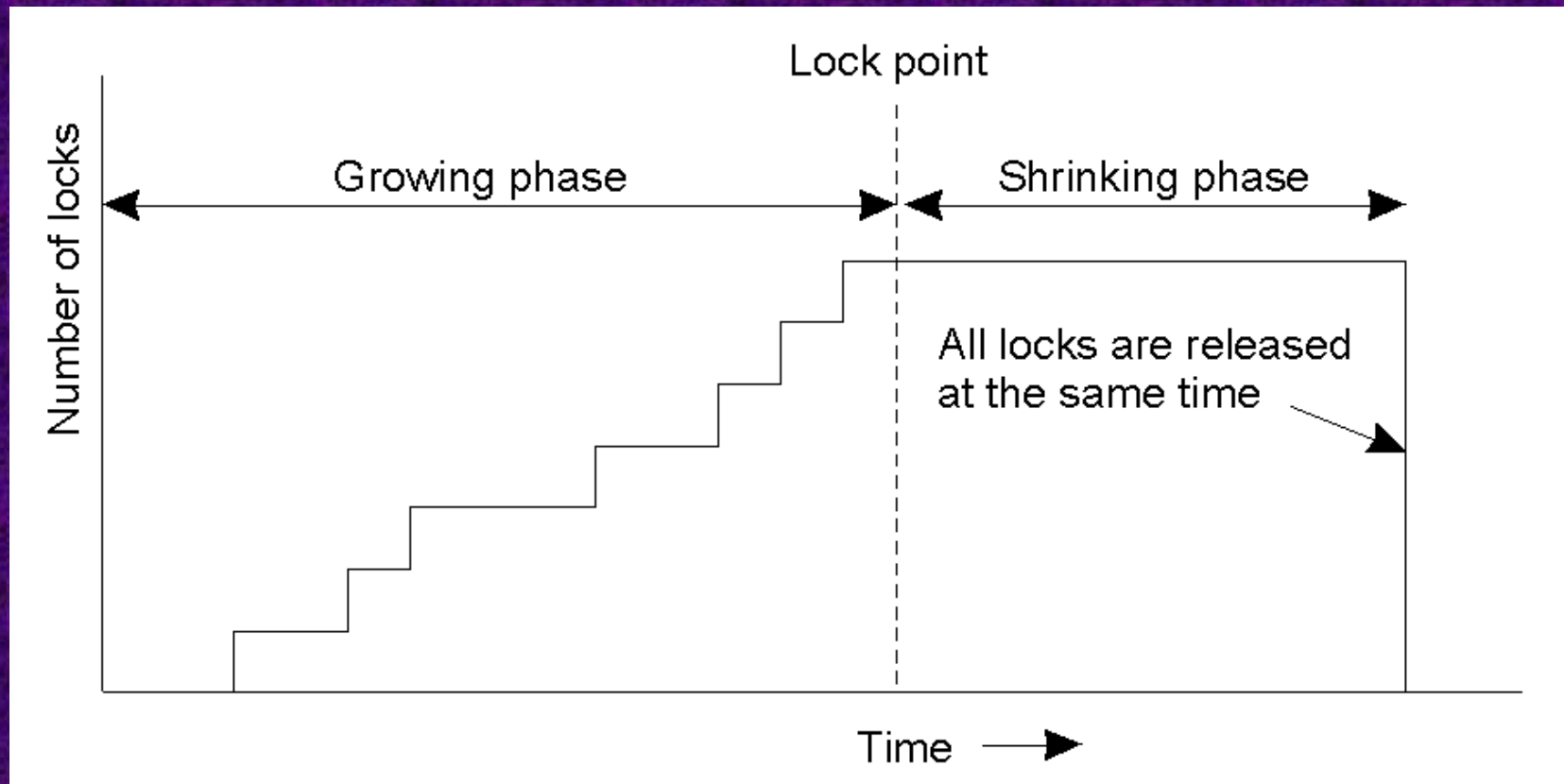
# Transacciones Distribuidas

## Bloqueo de 2 fases



# Transacciones Distribuidas

## Bloqueo de 2 fases



# *Transacciones Distribuidas*

## **Control Optimista:**

- Propuesto por Kung y Robinson en 1981 se basa en realizar todas las operaciones que se quieran sin importar lo que hagan los demás.
- Si existe un problema de concurrencia con otro proceso que modificó la misma información entonces se procede a deshacer la operación.
- Para ello utiliza un registro donde se encuentran las operaciones que se han realizado.
- En el momento de un compromiso se verifican las demás transacciones para ver si alguno de los archivos ha sido modificado desde el inicio de la transacción, si esto ocurre se aborta la transacción.

# *Transacciones Distribuidas*

---

## **Marcas de Tiempo:**

- **Propuesto por Reed en 1983 y se basa en el algoritmo de Lamport.**
- **Hace una marca de tiempo al momento de iniciar una transacción.**
- **Se debe garantizar que las marcas son únicas.**
- **Las marcas permiten realizar la atención de transacciones en forma ordenada.**

# **Algoritmos de Exclusión en caso de falla**

# *Transacciones Distribuidas*

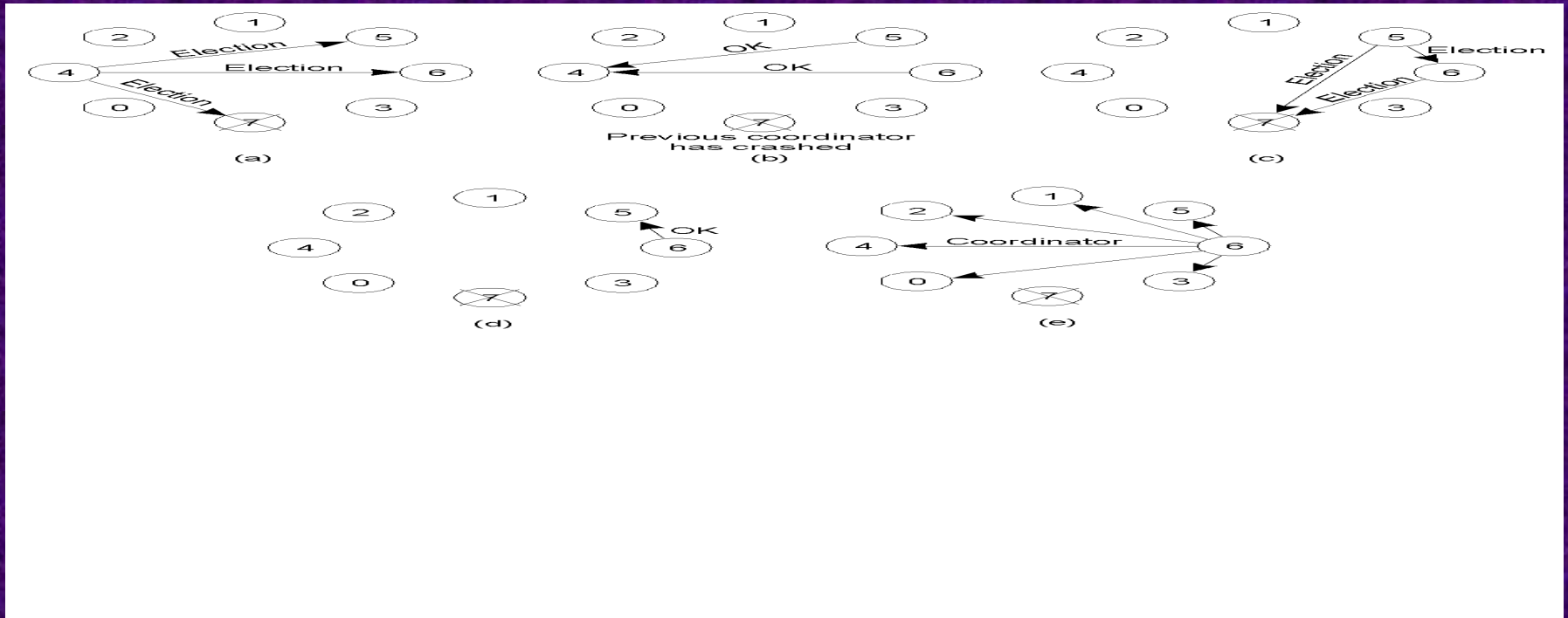
---

## Algoritmos de Elección: Grandulón

- P envía un mensaje de elección a los demás procesos.
- Si nadie responde, P gana la elección y se convierte en el coordinador.
- 3) Si uno de los procesos con un número mayor responde, toma el control y el trabajo de P termina

# Transacciones Distribuidas

## Algoritmos de elección: Grandulón





# *Transacciones Distribuidas*

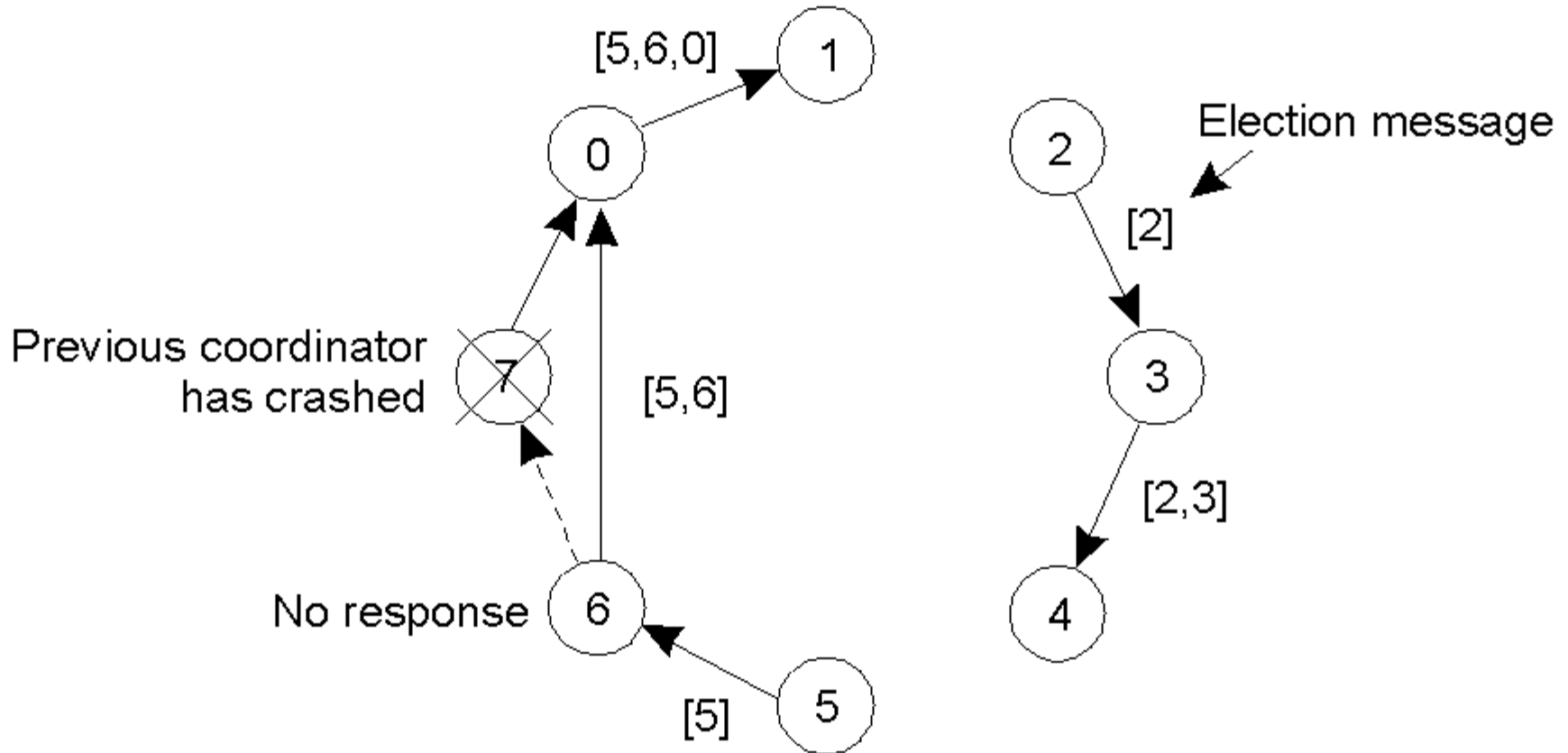
---

## **Algoritmos de Elección: Anillo**

- **Se ordenan en orden jerárquico los procesos.**
- **El sucesor del coordinador detecta el error y envía un mensaje de elección a los demás procesos en anillo, cada proceso pone su número y lo pasa al siguiente activo.**
- **Al final regresa al que lo inició un manda un mensaje donde indica quién ganó.**

# Transacciones Distribuidas

## Algoritmos de elección: Anillo



# *Transacciones Distribuidas*

---

## **Interbloqueo:**

- **Surge cuando un proceso busca el recurso ocupado por otro proceso y a su vez este proceso busca otro recurso, formando una cadena que al final se cierra, por lo cual ningún proceso puede avanzar.**

# *Transacciones Distribuidas*

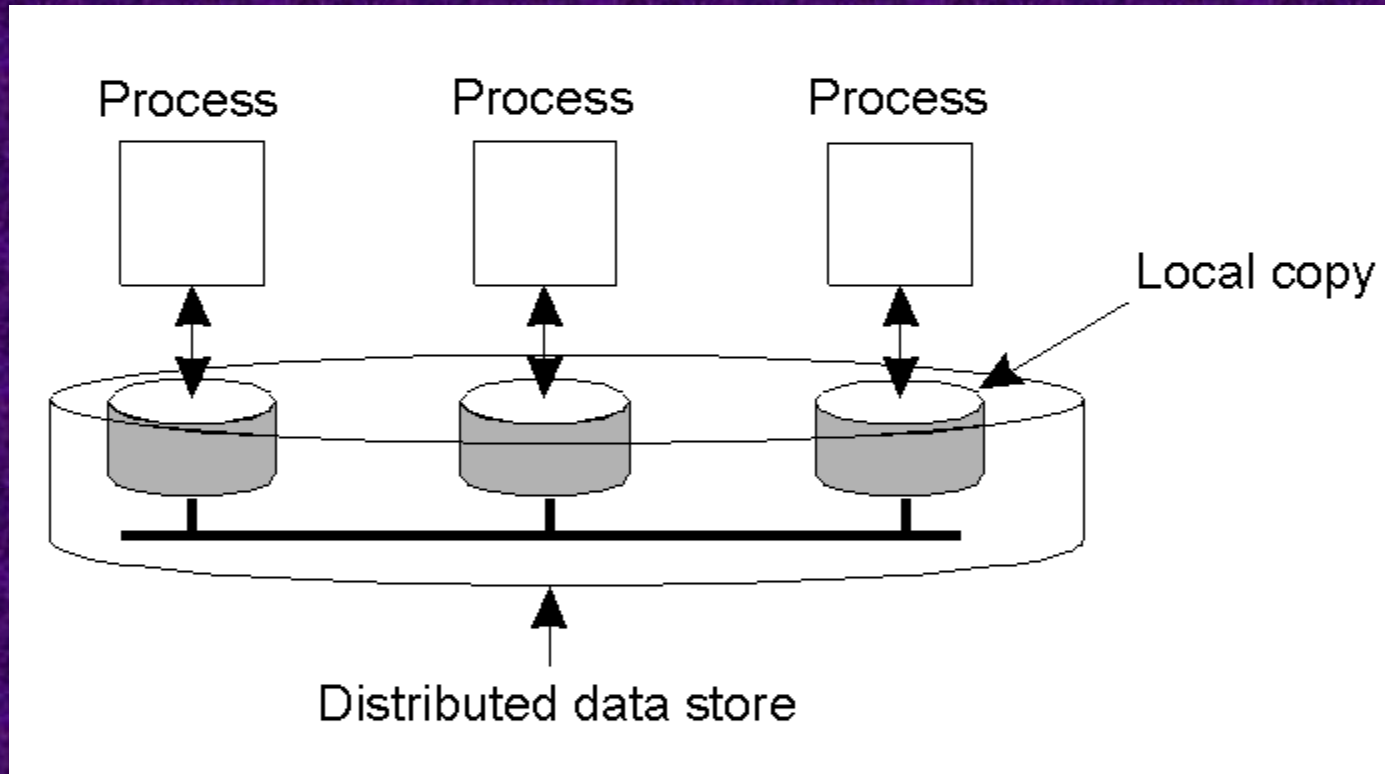
---

## **Estrategias para combatir los interbloqueos:**

- **Detección: Permitirlos y después recuperarlos.**
- **Prevención: Implementarlos desde la estructura.**
- **Evitarlos: Asignar recursos cuidadosamente (estimar los recursos es difícil).**

# Transacciones Distribuidas

## Réplicas



**Fin**

***Transacciones Distribuidas***