UNIDAD 8:

TÉCNICAS DE IMPLEMENTACIÓN DE LOS SGBD

Tipos de Almacenamiento:

- **Volátil:**
  - Rápido Acceso
  - Memoria Principal
  - Se pierde con las caídas del sistema
- **No volátil:**
  - Mas lento
  - Discos, cintas
  - No se pierde con las caídas del sistema
- **Estable:**
  - Repetición de la información en varios medios de almacenamiento no volátiles
  - No se pierde "nunca"

Transacciones

Transacción:

- operación 1
- operación 2

…

- operación n

Unidad lógica de trabajo

El SGBD debe asegurar la atomicidad de las transacciones,

se ejecuta "a todo o nada"

Transacciones

1. Toma un estado consistente de la BD
2. Durante su ejecución no necesariamente mantiene un estado consistente
3. Deja a la base de datos en un estado consistente

¿Qué pasaría si se produce una caída del sistema en medio de esta transacción?

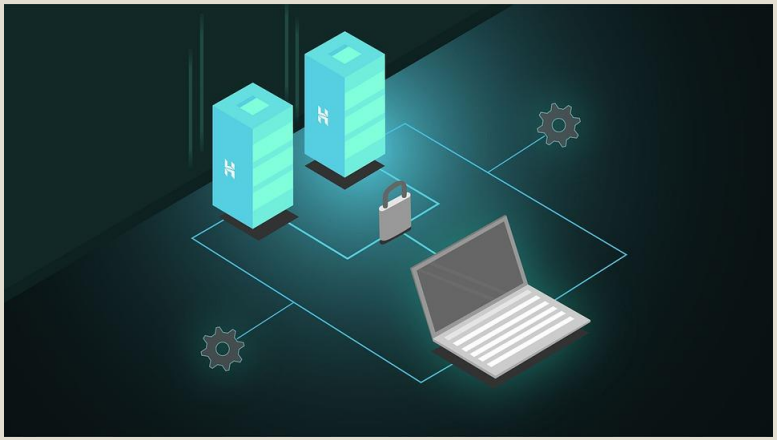
-- transfiere $100 de una cuenta a otra

- *Read(A,a1)*
- *a1:=a1-100*
- *Write(A, a1)*
- *Read(B,b1)*
- *b1:=b1+100*
- *Write(B, b1)*

Nota: *El sistema supone que toda transacción mantiene la correctitud de la BD*

Técnicas de implementación de los SGBDs

1. **Mecanismos de RECUPERACIÓN de transacciones (Bitácora)**
2. Gestión de accesos CONCURRENTES a la base de datos

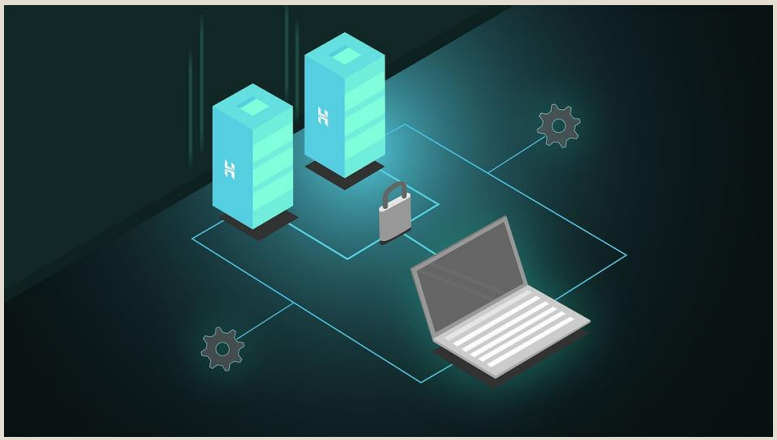


Recuperación de Transacciones

Tipos de Fallas

1. **Locales:** Afectan a la transacción donde se presentó el fallo.

2. **Globales:** Afectan a todas las transacciones que se están ejecutando en el momento del fallo.

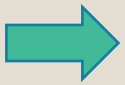


Recuperación de Transacciones

Fallas Globales

- **Fallas del Sistema:**
  - Provocan **pérdida** del almacenamiento **volátil**

Datos Volátiles
MEMORIA
PRINCIPAL



BASE DE DATOS

  - Almacenamiento **no volátil intacto**

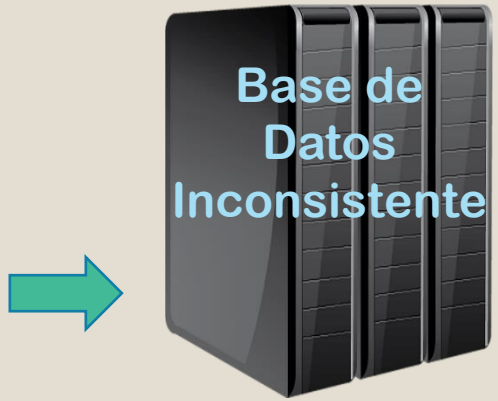
Datos no Volátiles
MEMORIA
SECUNDARIA(disco)

Recuperación de Transacciones

Múltiples usuarios trabajando sobre una base de datos

Recuperación de Transacciones

Fallas Globales



- **Fallas en el Disco**
  - Almacenamiento **no volátil dañado**
  - Uso de **backups**
  - Fundamental definir **políticas de backup** adecuadas:
    - ¿Con que frecuencia hacer los backup?
    - ¿ En qué horario?
    - ¿ Backup completos o parciales, o cuando completos o parciales?
    - ¿ Backup en caliente o en frio?

Fallas del Sistema:

Pérdida de almacenamiento volátil

Nos enfocaremos en este tipo de recuperación

Recuperación ante pérdida de memoria volátil

Recordemos como se gestiona la memoria:

MEMORIA PRINCIPAL – RAM
Datos volátiles

	Bloque1
	Bloque2
	Bloque3
	Bloque4
	Bloque5

MEMORIA DE APLICACIONES

BUFFERS BLOQUES LÓGICOS

¿Cuándo se ejecutan?

INPUT

OUTPUT

- Cuando la gestión de memoria lo necesita
- Checkpoint

MEMORIA SECUNDARIA – DISCO
Datos no volátiles

Bloque físico 1		
Bloque físico 2		
Bloque físico 3		
Bloque físico 4		
Bloque físico 5		

BLOQUES FISICOS EN EL DISCO

¿Cuándo se ejecutan?

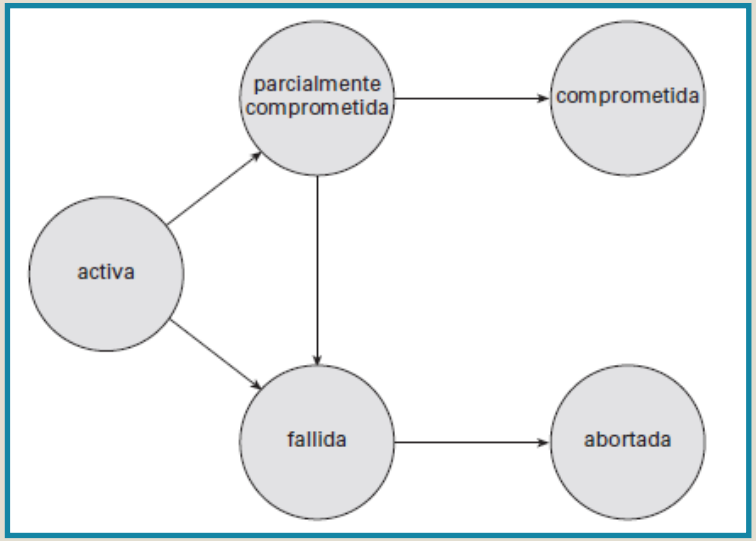
READ

WRITE

Cuando las transacciones lo solicitan

Estados posibles de una transacción

- **Activa**
- **Parcialmente cometida**
- **Fallida:** Después de descubrir que la ejecución normal no puede completarse. (Cuidado con **escrituras externas observables,** en un terminal o en una impresora.)
- **Abortada:** Después de que la transacción retrocedió (rollback)
- **Cometida:** Después que la transacción terminó con éxito, y se registró su commit



Recuperación ante pérdida de memoria volátil

¿Qué debe hacer el SGBD para no caer en un estado inconsistente ante un fallo?

Durante el proceso normal

Almacenar información en el **REGISTRO HISTÓRICO/ BITÁCORA**

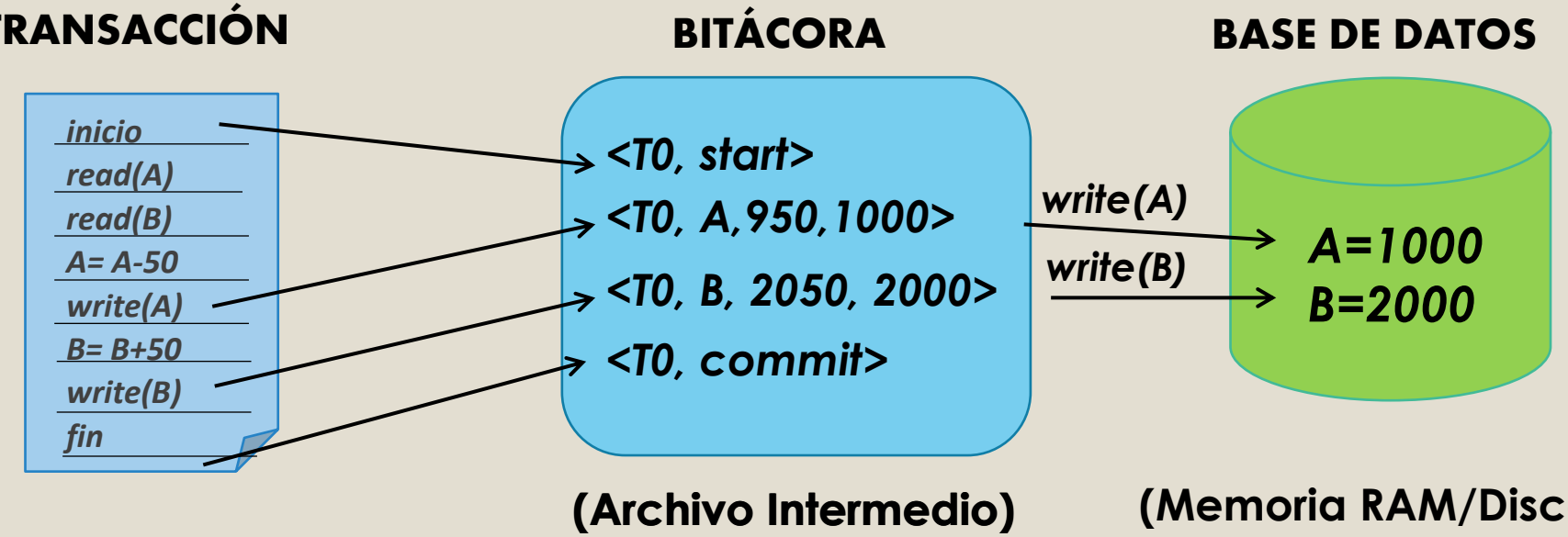
Después de la falla

Reestablecer la base de datos a un **ESTADO CONSISTENTE**

Recuperación ante pérdida de memoria volátil

Durante el Proceso Normal

PROCESO GENERAL: Almacenar información en el **REGISTRO HISTÓRICO o BITÁCORA**



Nota: *Formato registro de escritura utilizado <Ti, dato, valor-nuevo, valor-viejo>*

Recuperación ante pérdida de memoria volátil

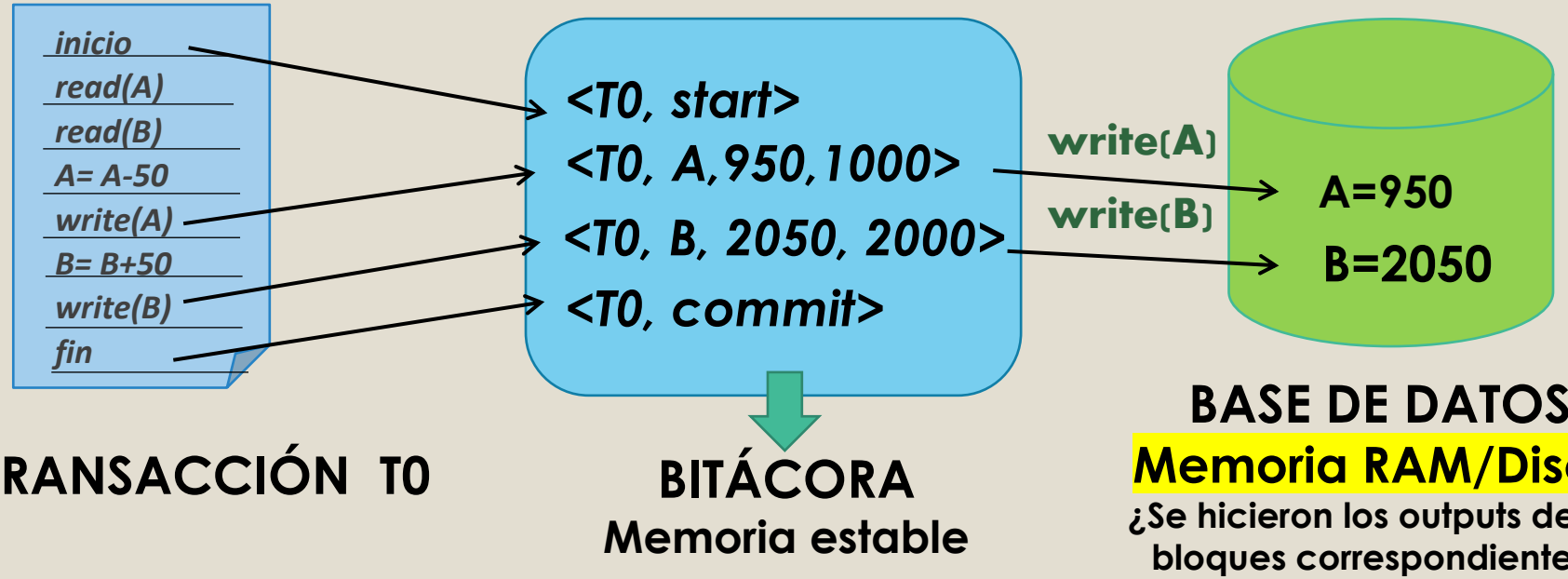
Mecanismos de Recuperación:

Modificación Inmediata **Modificación Diferida**

Modificación Inmediata

Durante el Proceso Normal

La ejecución de los write's se realizan **inmediatamente** después de almacenar cada registro de bitácora correspondiente (write) en memoria estable:

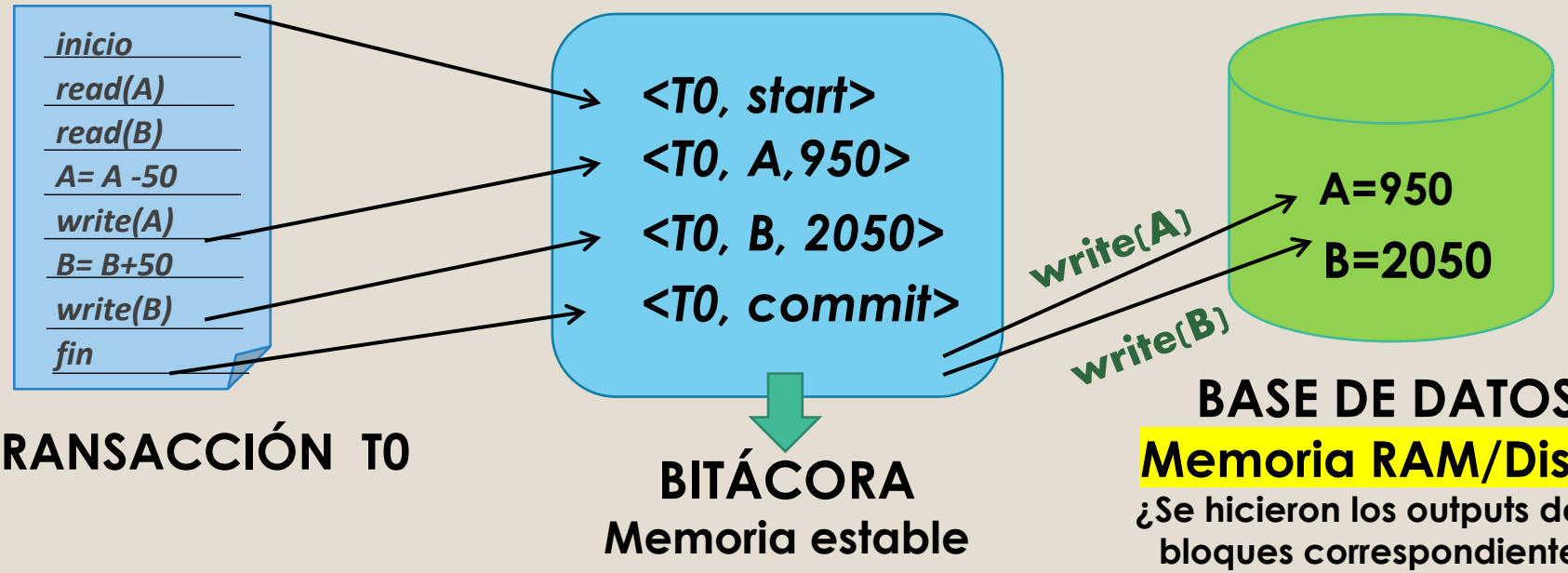


Nota: *Formato registro de escritura utilizado <Ti, dato, valor-nuevo, valor-viejo>*

Modificación Diferida

Durante el Proceso Normal

- La ejecución de los write's se **postergan** hasta después de almacenar en bitácora (memoria estable) el commit de la transacción:



Nota: *Formato registro de escritura utilizado <Ti, dato, valor-nuevo, valor-viejo>*

Recuperación ante pérdida de memoria volátil

¿Qué debe hacer el SGBD para no caer en un estado inconsistente ante un fallo?

Durante el proceso normal

Almacenar información en el **REGISTRO HISTÓRICO/ BITÁCORA**

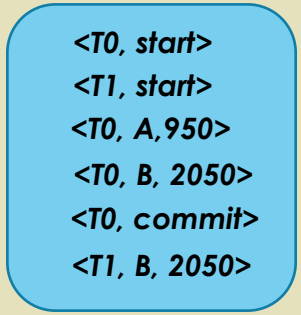
Después de la falla

Reestablecer la base de datos a un **ESTADO CONSISTENTE**

Procedimiento general para restablecer el estado de la BD - Después de la falla

I. **Identificación (en base a la bitácora) de las transacciones** a aplicarles los procedimientos:

- **REDO** (REHACER): **start y commit**
- **UNDO** (DESHACER): **start y NO commit**



Bitácora en Memoria Estable

Especializado para cada mecanismo

II. **Ejecución de los procedimientos** REDO y/o UNDO

- **REDO**: Ejecuta las operaciones **write** con los **valores nuevos**.
- **UNDO**: Ejecuta las operaciones **write** con los **valores viejos**.

III. **Notificación al usuario de lo realizado**

Nota: *Las operaciones REDO y UNDO son idempotentes.*

Identificación de las operaciones, según el mecanismo de recuperación

Modificación Inmediata

Para cada transacción con:

- *Start*
- *Commit*

◦ **REDO**: Ejecuta las operaciones write con los **valores nuevos**.

Para cada transacción con:

- *Start*
- ~~*Commit*~~

◦ **UNDO**: Ejecuta las operaciones write con los **valores viejos**.

Modificación Diferida

Para cada transacción con:

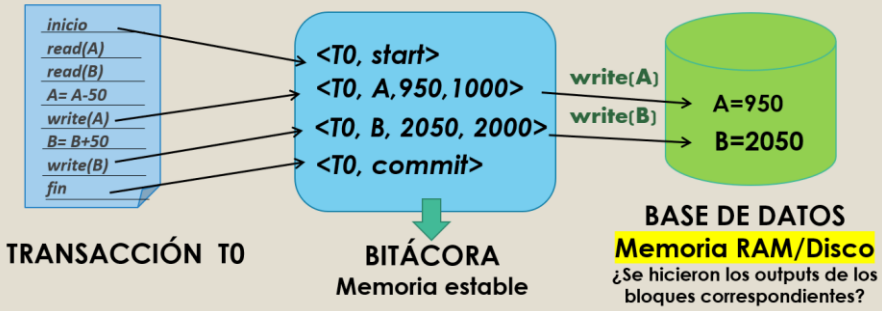
- *Start*
- *Commit*

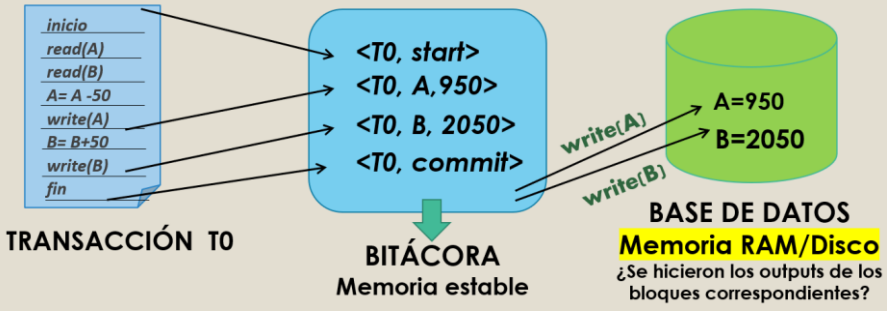
◦ **REDO**: Ejecuta las operaciones write con los **valores nuevos**.

Para cada transacción con:

- *Start*
- ~~*Commit*~~

◦ **No hace nada**





Optimización del Procedimiento de Reinicio

I. **Identificación de las transacciones** a aplicarles los procedimientos:
   - **REDO** (REHACER): **start y commit**
   - **UNDO** (DESHACER): **start y NO commit**

II. **Ejecución de los procedimientos** REDO y/o UNDO

III. **Notificación al usuario de lo realizado**

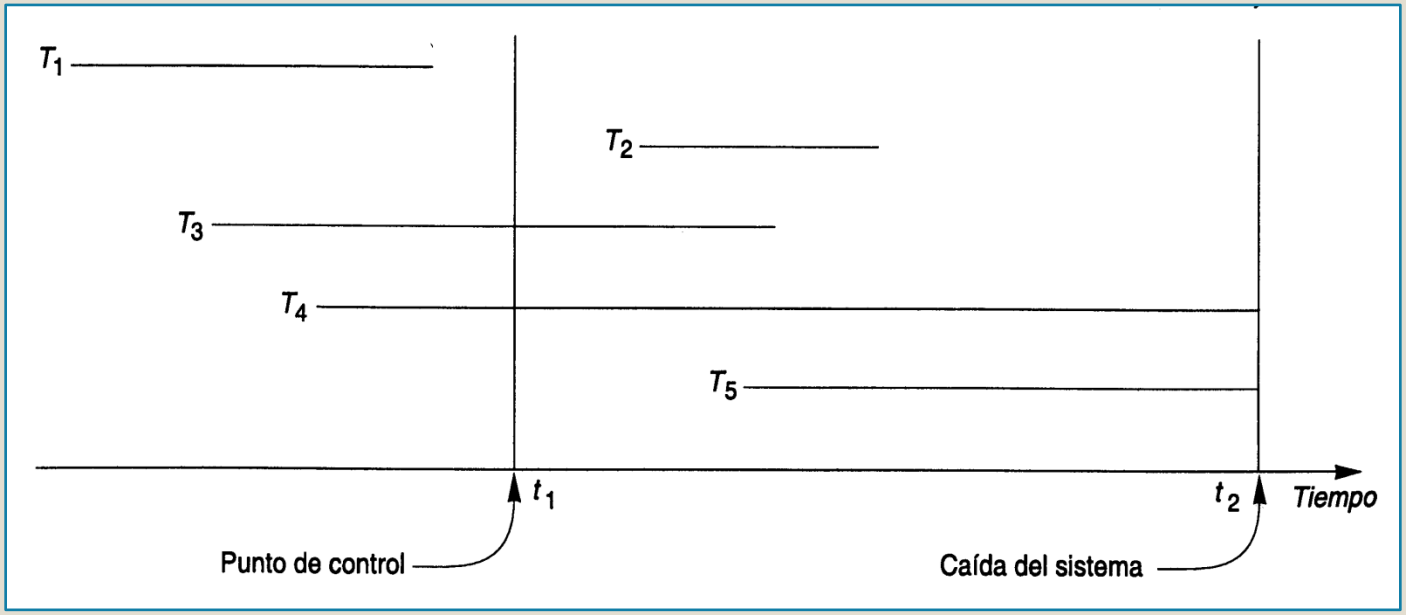
- Consume mucho tiempo
- La mayoría de los cambios ya están plasmados en el disco (no genera resultados erróneos)

PUNTO DE CHEQUEO (CHECKPOINT):

1. Escritura en almacenamiento estable de los registros de bitácora que residen en memoria principal
2. Escritura en disco de todos los bloques de memoria intermedia (buffers) que se hayan modificado
3. Escritura en almacenamiento estable del registro correspondiente al punto de chequeo <checkpoint>

Optimización del Procedimiento de Reinicio

Proceso de Identificación de transacciones a recuperar considerando el punto de chequeo/control



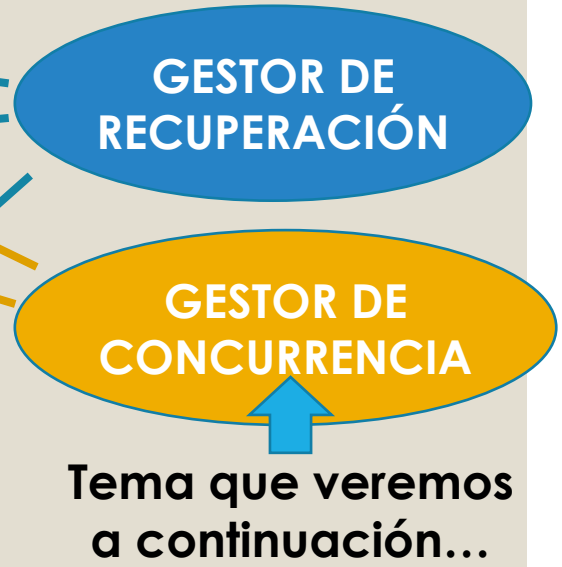
Registros de Bitácora mostrados en la línea del tiempo

Recuperación de Transacciones

Propiedades ACID

Los SGBDs Relacionales aseguran las siguientes propiedades:

- **A**tomicity (Atomicidad)
- **C**onsistency (Consistencia)
- **I**solation (Aislamiento)
- **D**urability (Durabilidad)



Recuperación de Transacciones

Seguimos en la próxima!!!