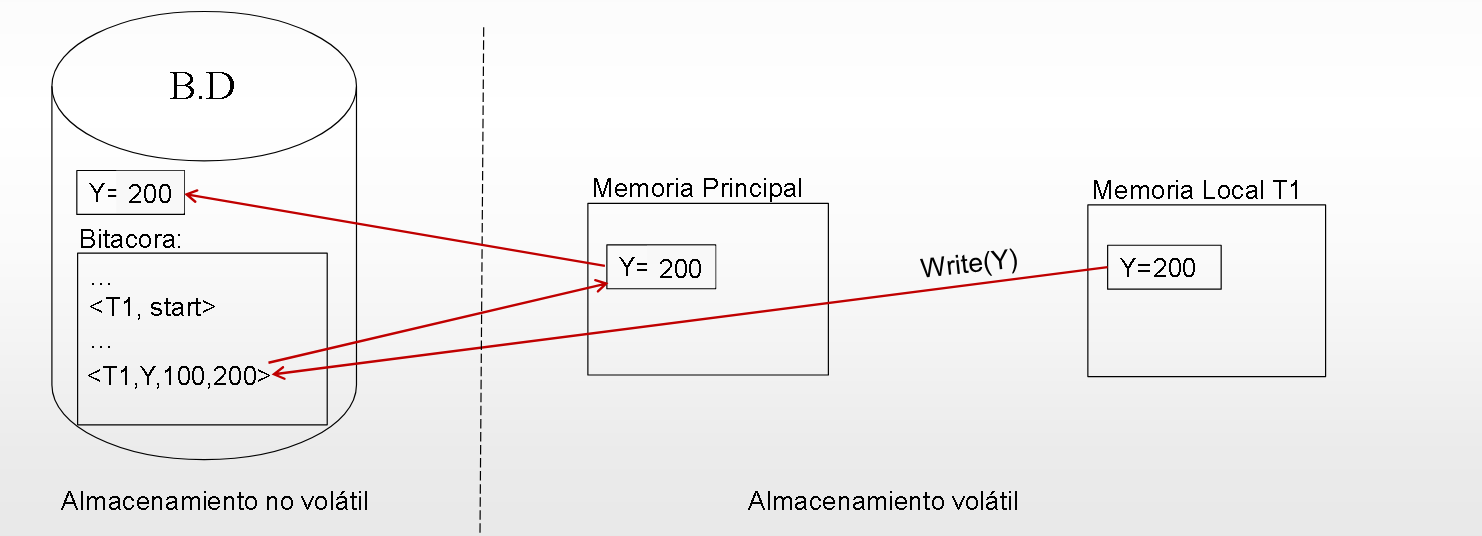
**Modificación inmediata**

Las escrituras sobre la Base de Datos se realizan mientras la transacción está activa.

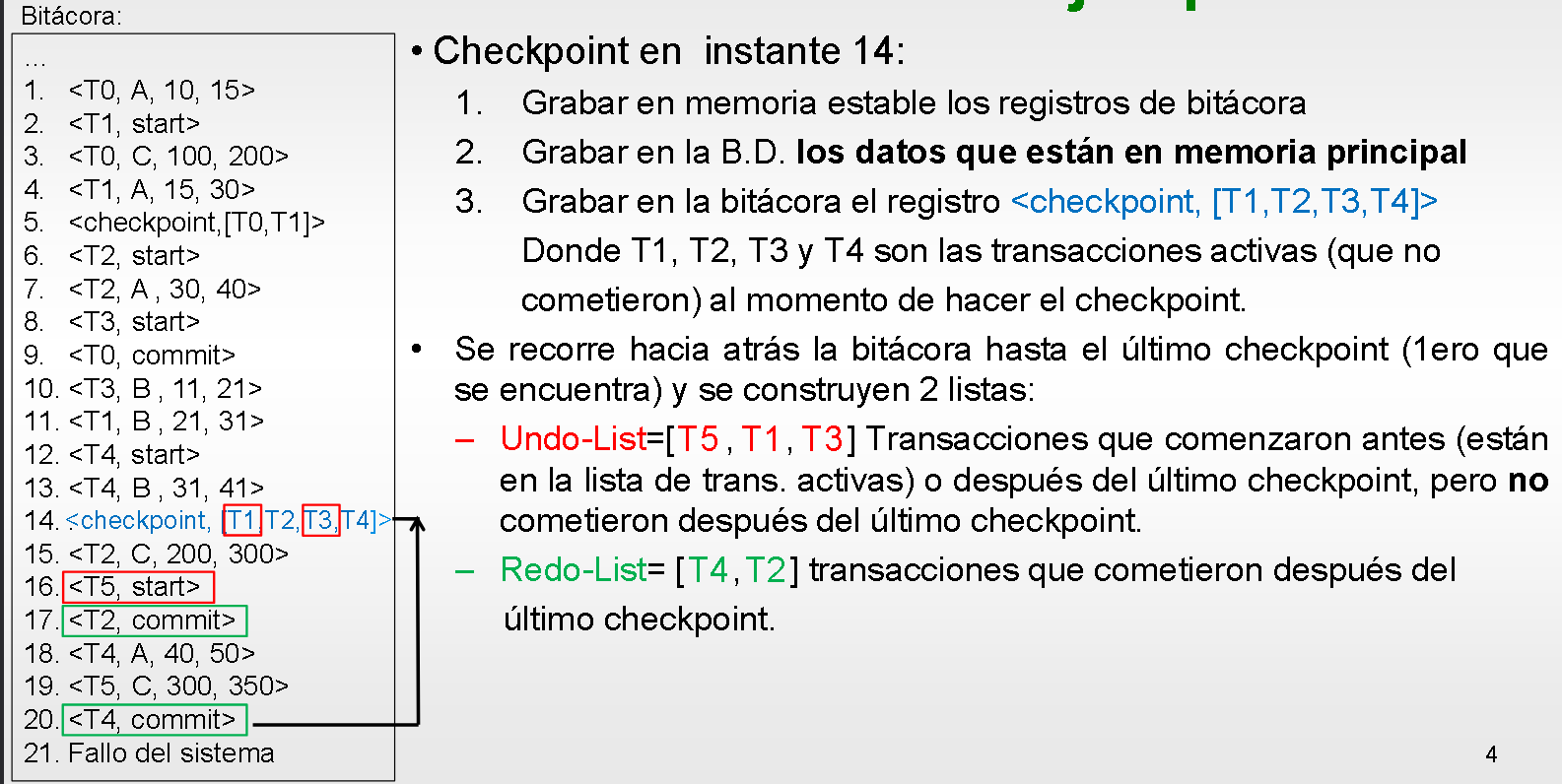
En la bitácora se almacenan los registros <Ti, start> y <Ti, commit> cuando una transacción Ti comienza y termina respectivamente

Si una transacción Ti escribe Y, primero se almacena en la bitácora el registro:

**<Ti, Y, valor\_viejo, valor\_nuevo>** y luego se modifica el valor de Y.



Ejemplo:



En un checkpoint guardamos las transacciones que arrancaron y no cometieron, en este caso como hay otro checkpoint arrancamos viendo que transacciones no habían cometido hasta ese momento (sino habría q ver desde arriba del todo) y vamos viendo que transacciones arrancan y no hacen commit.

En este caso tenemos T0 y T1 en el checkpoint anterior y se suma T2, T3, luego como T0 comete entonces la sacamos, y se suma T4, nos queda entonces: <checkpoint, [T1,T2,T3,T4]>

Cuando ocurre la falla del sistema se busca al ultimo checkpoint y se hacen 2 listas, una para rehacer y otra para deshacer.

En la Undo-list (lista para deshacer) van todas las transacciones que no cometieron después del último checkpoint

En la Redo-list (lista para rehacer) van todas las transacciones que cometieron después del checkpoint. Esto es, desde el fallo del sistema vamos recorriendo para arriba hasta llegar al checkpoint y si la transacción cometio entonces va en la lista.

**Redo-list** = [T4, T2]

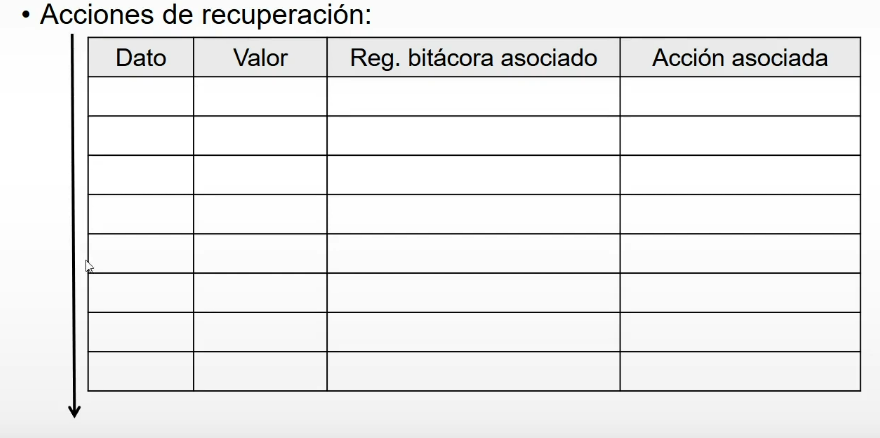
Como T5 empezo luego del checkpoint pero no cometio entonces va en la Undo-list

**Undo-list** = [T5, \_, \_]

Cuando llegamos al checkpoint puede haber transacciones activas que no cometieron después, entonces se compara la lista del checkpoint con la lista de Redo, toda transaccion que este en el checkpoint y no este en la del Redo se agrega a la lista Undo

**Undo-list** = [T5, T1, T3]

Hacemos lista de recuperación:

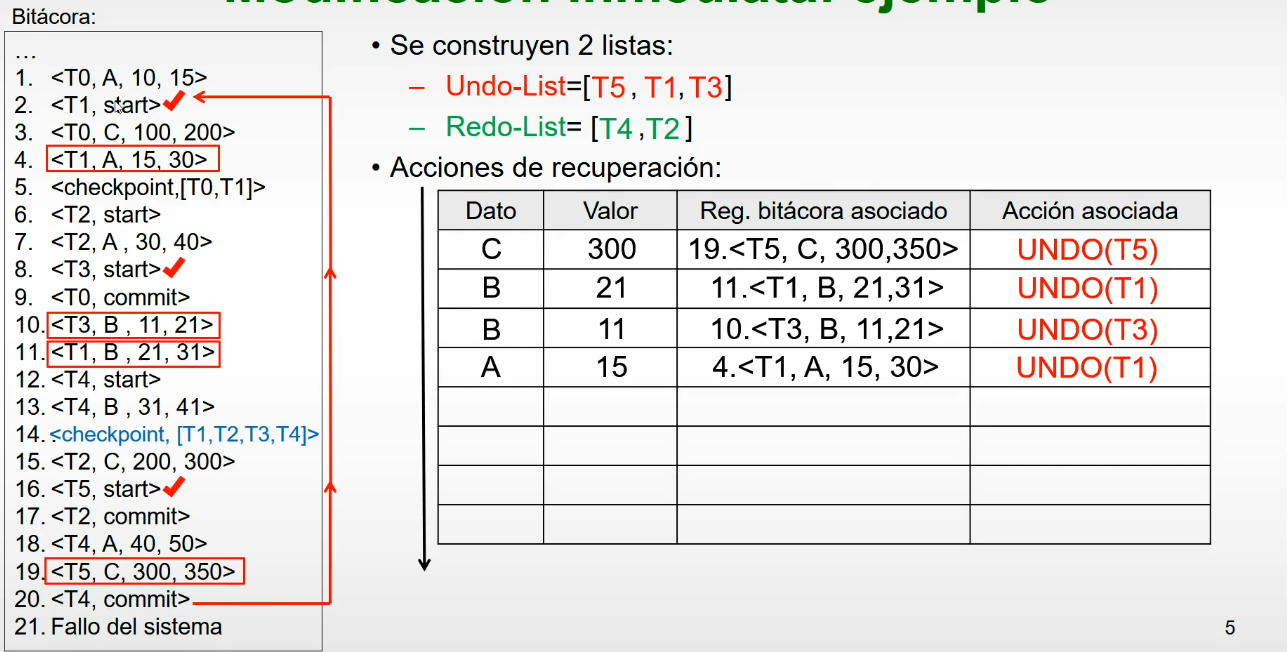


Primero deshacemos todas las transaccion que esten en la Undo-list y luego rehacemos todas las transacciones que esten en la lista Redo-list

Recorremos de abajo para arriba en la bitácora y buscamos los start que esten en la lista de Undo y trabajaremos con la lista desde el fallo hasta el último start de la Undo-list que falte encontrar.

Buscaremos los registros que hagan una operación sobre las transacciones de la Undo-list

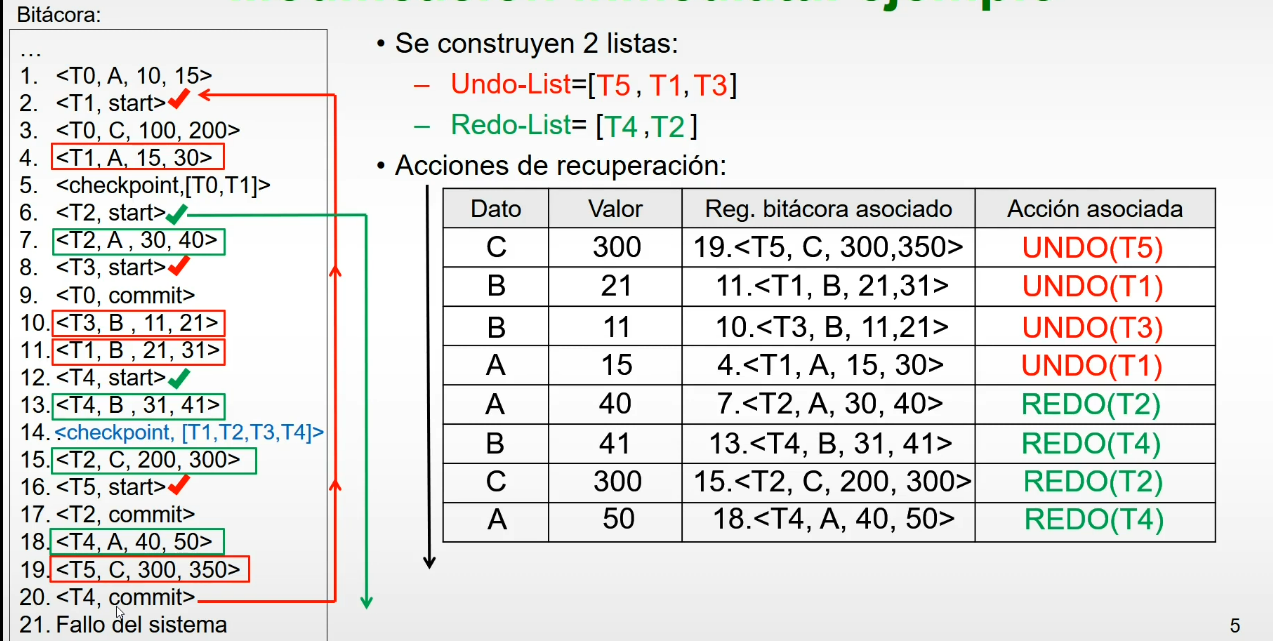
Entonces por ej encontramos T5 y escribimos el dato C, el **valor** **viejo** 300, el registro es escribir todo el registro incluido número de línea que seria “19. <T5, C, 300, 350>” y la acción asociada es Undo(T5) que quiere decir que deshacemos la transacción T5. Repetimos para el resto de la undo-list



Para la Redo-list recorremos nuevamente desde el fallo del sistema y buscamos hasta el ultimo start de la Redo-list, el start de T4 esta en la línea 12 y el de T2 está en la línea 6, por lo tanto, desde la línea 6 hasta la falla del sistema hay que rehacer las transacciones de ambas.

Las acciones de recuperación de la Redo-list se hacen en el mismo orden que se fueron dando antes, por eso se recorre desde arriba para abajo.

Cada vez que veamos un registro de modificación de datos de T2 o T4 ponemos un **valor** **nuevo** en ese dato



**Los valores finales que quedan son la ultima modificación que hicimos sobre los datos:**

**A=50, B=41, C=300**

**Resumen**:

En el siguiente orden hacer:

1. La lista de checkpoint van las transacciones que hasta el checkpoint no comitearon.
2. La **Redo-list**, se recorre de abajo para arriba desde el fallo del sistema hasta el checkpoint y si una transaccion **hizo** un commit se agrega en la lista
3. La **Undo-list** se recorre de abajo para arriba desde el fallo del sistema hasta el checkpoint y si una transaccion **no hizo** un commit se agrega en la lista, cuando llegamos al checkpoint agregamos las transacciones que esten en el checkpoint y no esten en la Redo-list.
4. Hacemos lista de recuperación

En la lista de recuperación se agregan las transacciones que esten en la Undo-list y luego todas las transacciones que esten en la lista Redo-list.

Para la undo-list

* Localizamos los start de las transacciones de la undo-list y recorremos de abajo para arriba buscando los registros que hagan una modificación sobre las transacciones de la Undo-list y agregamos el dato, el valor viejo, etc., hasta llegar al ultimo start de la undo-list.

Para la redo-list

* Recorremos nuevamente desde el fallo del sistema y buscamos hasta el último start de la Redo-list y luego recorremos de arriba para abajo desde el primer start de las transacciones de la redo-list y buscando las modificaciones que se hagan sobre esas transacciones y agregamos el dato, el valor nuevo, etc., hasta llegar a la falla del sistema.
* Los valores finales que quedan son la última modificación que hicimos sobre los datos.