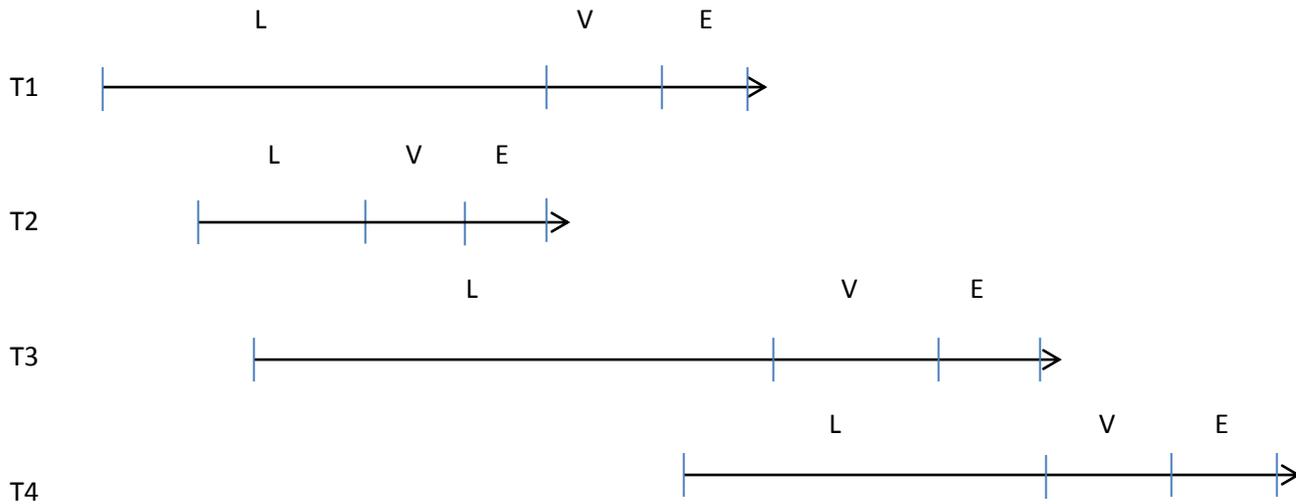


## Protocolo de Validación



T1: Read(A), Read(B), Write(C)

T2: Write(C), Write(D), Write(E)

T3: Read(D), Write(D)

T4: Read(D), Read(E)

**Inciso a:** Si la ejecución del protocolo fuese exitosa, cuál sería la serie equivalente bajo validación?

**Inciso b:** Para cada fase de validación de transacción y en el orden en que se presentan, especifique contra cuál/es transacciones debe validar y justifique por qué valida o por qué debe abortar.

En la *fase de lectura*, la transacción lee los datos y almacena los valores en variables locales.

En la *fase de validación* es donde se define si comprueba la serializabilidad. De ser así se aborta, en otro caso valida.

En la *fase de escritura* se actualiza la BD de acuerdo a los valores locales de la transacción.

Se distinguen tres etiquetas por transacción: **Start(Ti)** donde comienza su ejecución, **Validate(Ti)** donde termina su fase de lectura y comienza la validación, **Finish(Ti)** donde termina su fase de escritura.

La etiqueta de tiempo de cada transacción equivale su etiqueta de Validación: **Ts(Ti) = V(Ti)**

Es decir que el *orden de serializabilidad* es igual al orden en que las transacciones validan.

Para ver si una transacción Tj puede validar, debe verificarse que PARA TODA transacción Ti tal que  $Ts(Ti) < Ts(Tj)$  se cumpla alguna de las siguientes condiciones:

1.  $Finish(Ti) < Start(Tj)$  ó
2.  $Start(Tj) < Finish(Ti) < Validate(Tj)$  y la intersección de los datos que Ti escribe con los que Tj lee sea vacía,  $Lee(Tj) \cap Escribe(Ti) = \emptyset$

**Inciso a:** Según el orden de validación, tenemos que  $Ts(T2) < Ts(T1) < Ts(T3) < Ts(T4)$ . Por lo tanto, si la ejecución del protocolo fuese exitosa, la serie equivalente sería **T2, T1, T3, T4**

**Inciso b:**

Validación de T2: Valida porque no tiene ninguna transacción de estampilla menor. **T2 valida.**

Validación de T1:

Debe validar contra T2. La condición 1 no se verifica, la condición 2 sí ya que  $Start(T1) < Finish(T2) < Validate(T1)$  y  $Lee(T1) \cap Escribe(T2)$  es vacío. Por lo tanto **T1 valida.**

Validación de T3:

Debe validar contra T2 y T1.

*Contra T2:* La condición 1 no se verifica y la condición 2 tampoco ya que  $Start(T3) < Finish(T2) < Validate(T3)$ , pero  $Lee(T3) \cap Escribe(T2) = \{D\}$ . Por lo que **T3 no valida y debe abortar.**

*Contra T1:* No importa, porque ya se verificó que no valida contra T2 y por lo tanto retrocede.

Validación de T4:

Debe validar contra T2 y T1. No debe validar contra T3 ya que ya retrocedió.

*Contra T2:* Se verifica la condición 1 ya que  $Finish(T2) < Start(T4)$ , con lo que valida contra T2.

*Contra T1:* La condición 1 no se verifica, la condición 2 sí ya que  $Start(T4) < Finish(T1) < Validate(T4)$  y  $Lee(T4) \cap Escribe(T1)$  es vacío. Por lo tanto T4 valida contra T1.

Como T4 valida contra todas las transacciones no abortadas de estampilla menor, **T4 valida.**