

Bases de Datos

Serializabilidad en Conflictos y en Vistas



Dr. Diego R. Garcia

**DEPARTAMENTO DE CIENCIAS E
INGENIERÍA DE LA COMPUTACIÓN
UNIVERSIDAD NACIONAL DEL SUR**

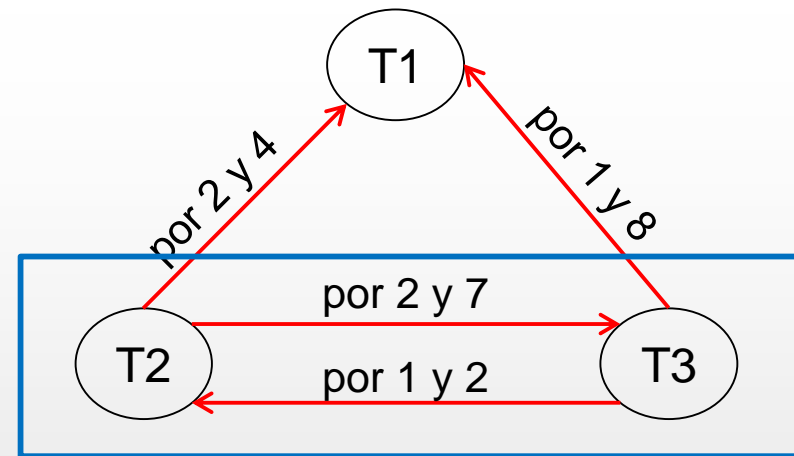


Prueba serializabilidad en Conflictos

Grafo de precedencia:

- Un nodo por cada transacción
- Un arco de T_i a T_j si: T_i ejecuta una instrucción $I(X)$ sobre antes que T_j ejecute una instrucción $J(X)$ (sobre el mismo dato X) y alguna de las instrucciones I o J es $\text{write}(X)$
- Si el grafo **no** tiene ciclos entonces la planificación es serializable. Si el grafo presenta un ciclo la planificación no es serializable en cuanto a conflictos.

	T1	T2	T3
1			R(A)
2		W(A)	
3			W(B)
4	R(A)		
5		R(B)	
6	W(B)		
7			W(A)
8	W(A)		



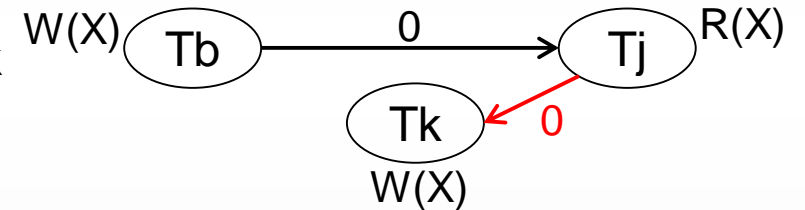
Presenta un ciclo entre
T2 y T3 => no es
serializable en conflictos

Prueba serializabilidad en Vistas

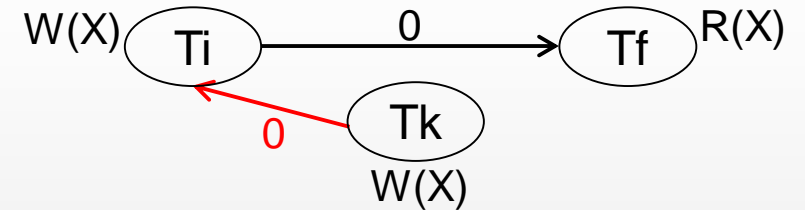
Grafo de precedencia etiquetado

- Se agregan 2 transacciones ficticias: Tb (inicial) escribe todos los datos al comienzo de la planificación y Tf (final) lee todos los datos al final.
- Un nodo por cada transacción incluidas (Tb y Tf)
- Un arco $T_i \xrightarrow{0} T_j$ si T_j lee un dato escrito por T_i
- Por cada dato X tal que T_j lee el valor de X escrito por T_i ($T_i \xrightarrow{0} T_j$) y otra transacción T_k ($T_k \neq T_b$) ejecuta $W(X)$ (en cualquier lugar) se analizan los siguientes casos:

a) Si $T_i = T_b$ y $T_j \neq T_f \Rightarrow$ se inserta $T_j \xrightarrow{0} T_k$



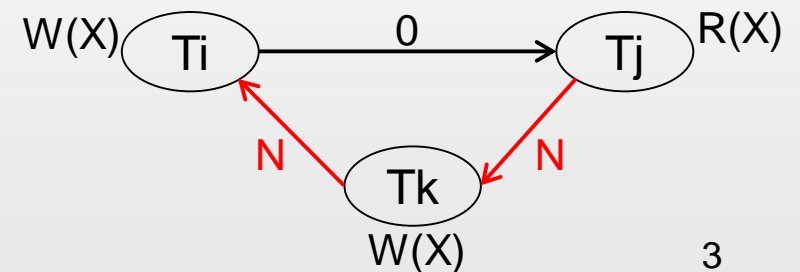
b) Si $T_i \neq T_b$ y $T_j = T_f \Rightarrow$ se inserta $T_k \xrightarrow{0} T_i$



c) Si $T_i \neq T_b$ y $T_j \neq T_f \Rightarrow$ se insertan:

$T_k \xrightarrow{N} T_i$ y $T_j \xrightarrow{N} T_k$ con $N > 0$

Un numero N nuevo cada vez que se agrega para identificar una opción, en caso de ciclo.



Tb (escribe A y B)

	T1	T2	T3
1			R(A)
2		W(A)	
3			W(B)
4	R(A)		
5		R(B)	
6	W(B)		
7			W(A)
8	W(A)		

Tf (lee A y B)

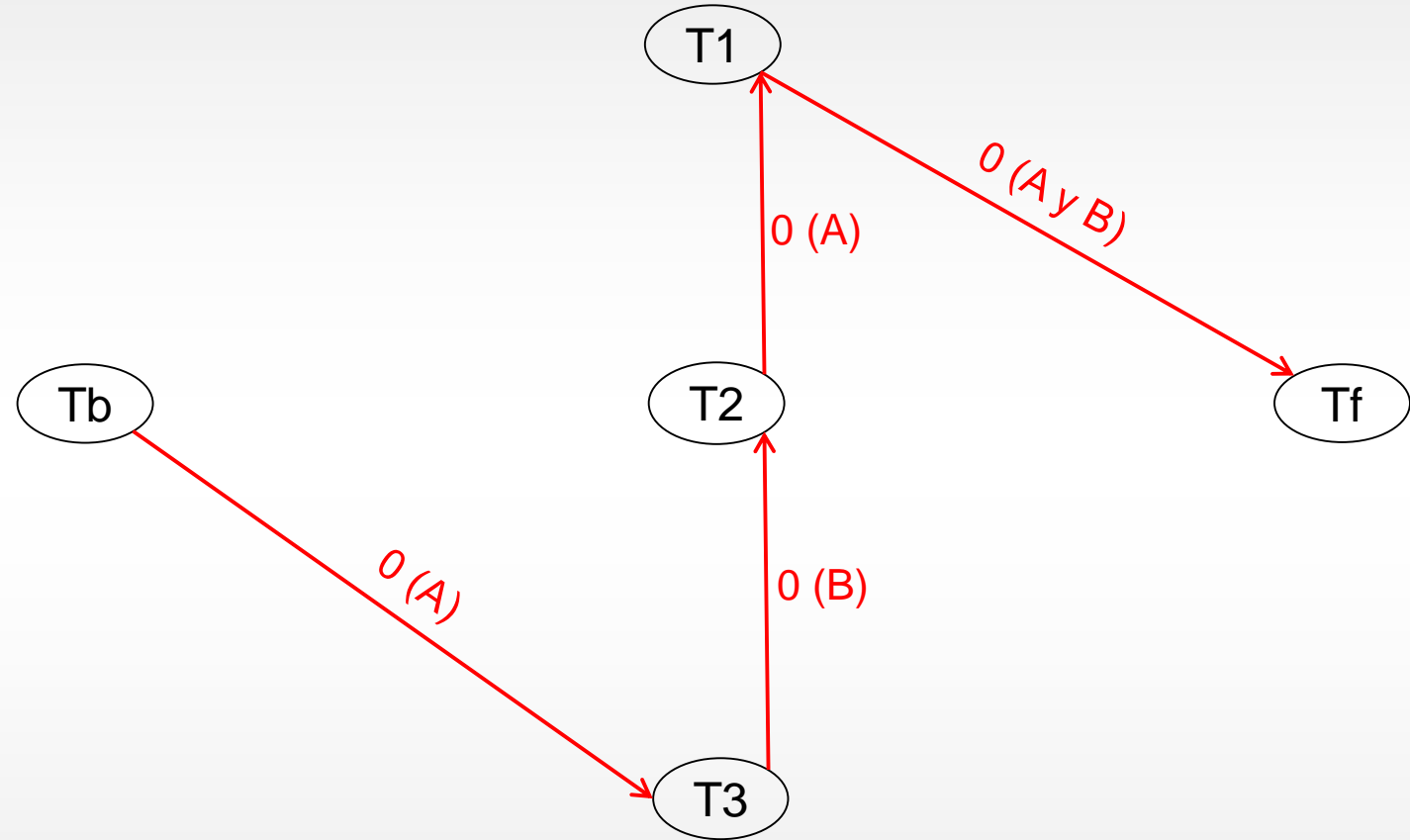
Prueba serializabilidad en Vistas

Grafo de precedencia etiquetado

Tb (escribe A y B)

	T1	T2	T3
1			R(A)
2		W(A)	
3			W(B)
4	R(A)		
5		R(B)	
6	W(B)		
7			W(A)
8	W(A)		

Tf (lee A y B)



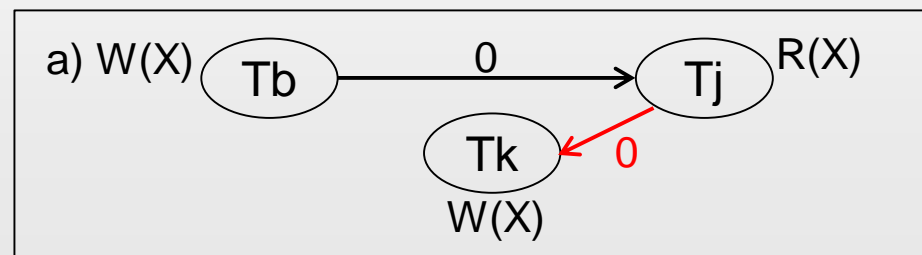
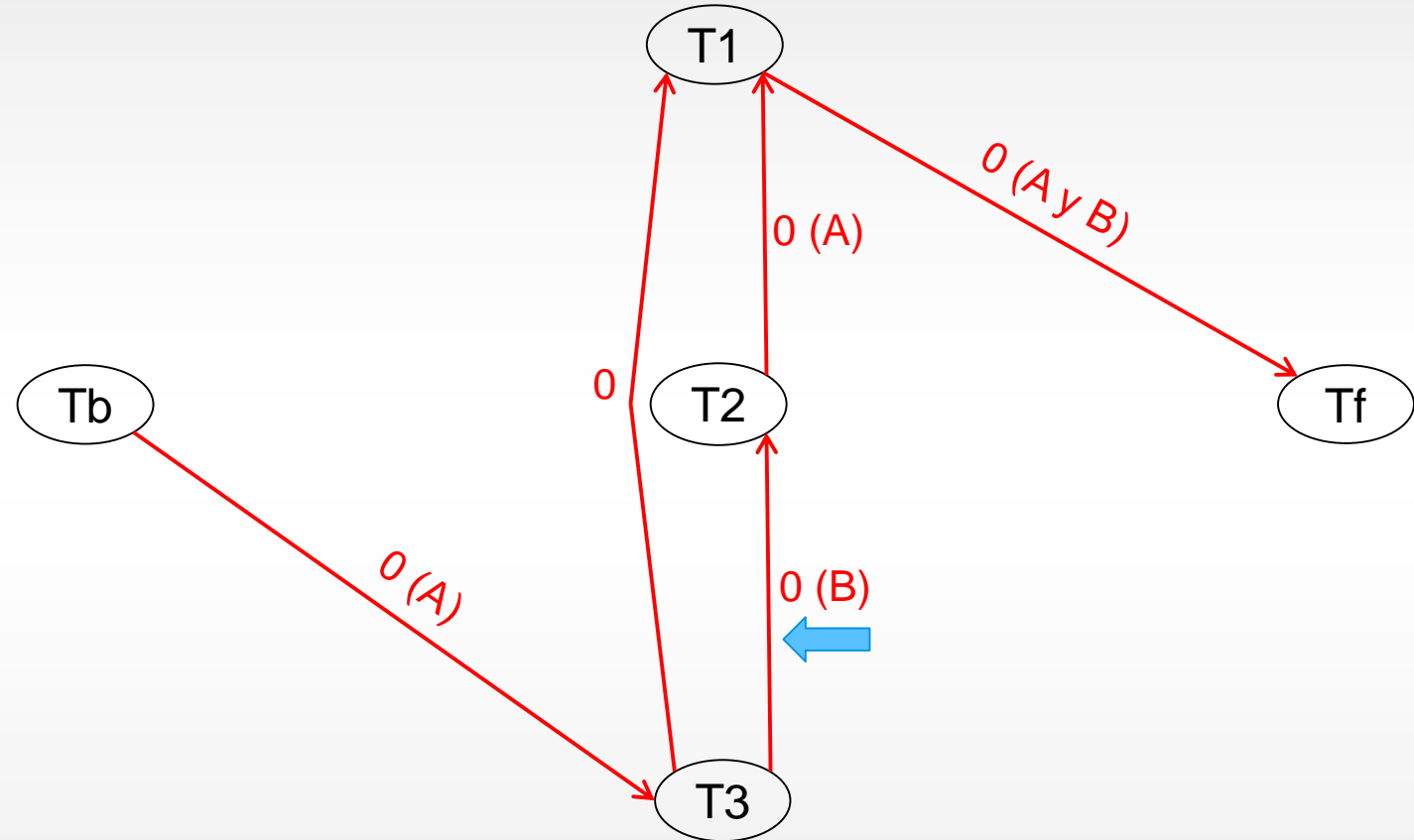
Prueba serializabilidad en Vistas

Grafo de precedencia etiquetado

Tb (escribe A y B)

	T1	T2	T3
1			R(A)
2		W(A)	
3			W(B)
4	R(A)		
5		R(B)	
6	W(B)		
7			W(A)
8	W(A)		

Tf (lee A y B)



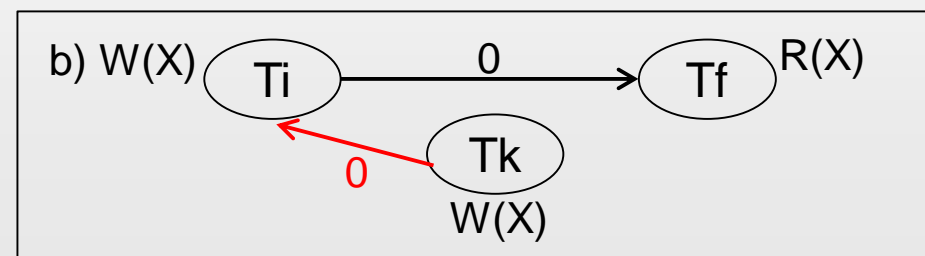
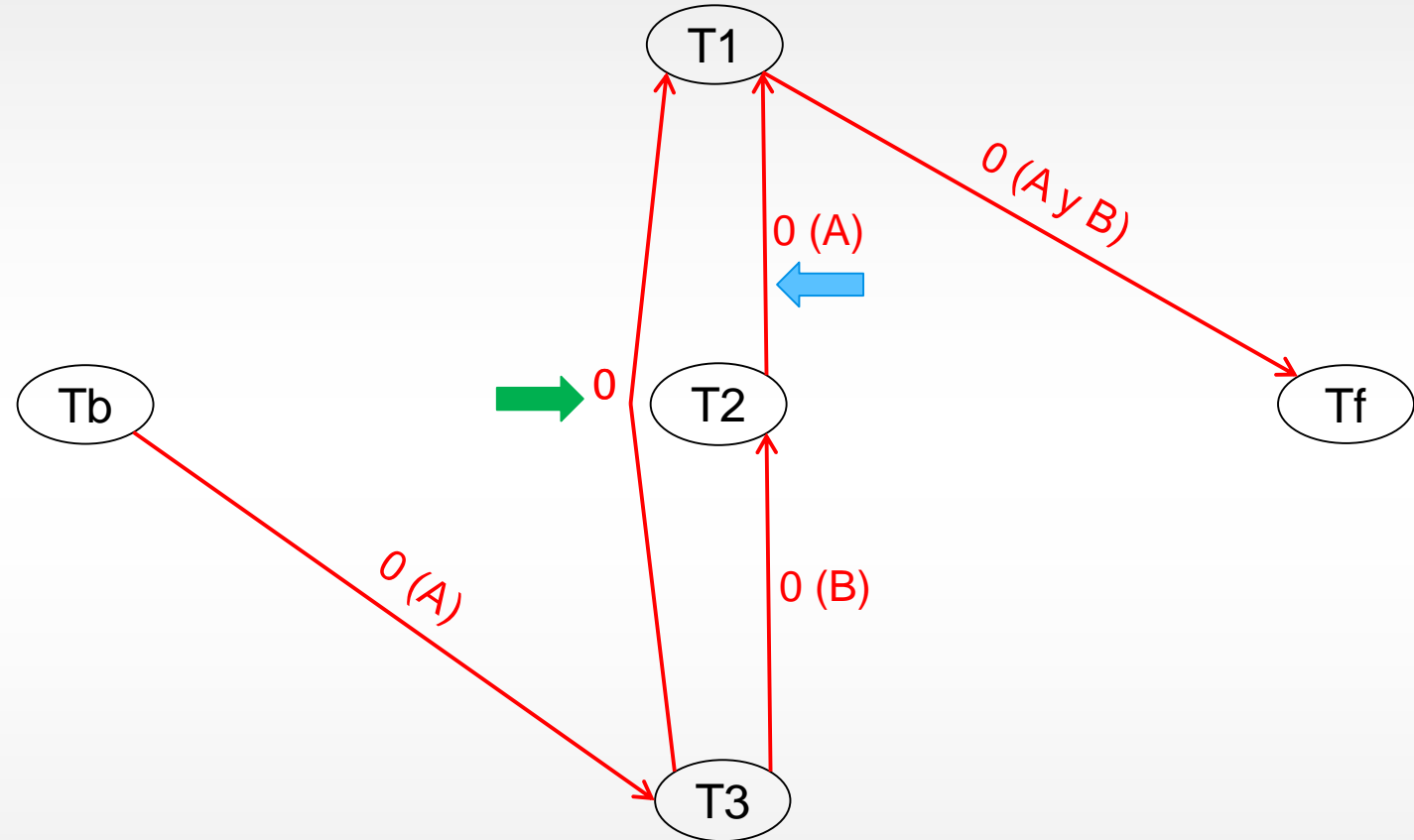
Prueba serializabilidad en Vistas

Grafo de precedencia etiquetado

Tb (escribe A y B)

	T1	T2	T3
1			R(A)
2		W(A)	
3			W(B)
4	R(A)		
5		R(B)	
6	W(B)		
7			W(A)
8	W(A)		

Tf (lee A y B)



Si en este punto el grafo presenta un ciclo con arcos etiquetados con 0 se puede concluir que **no** es serializable en vistas.

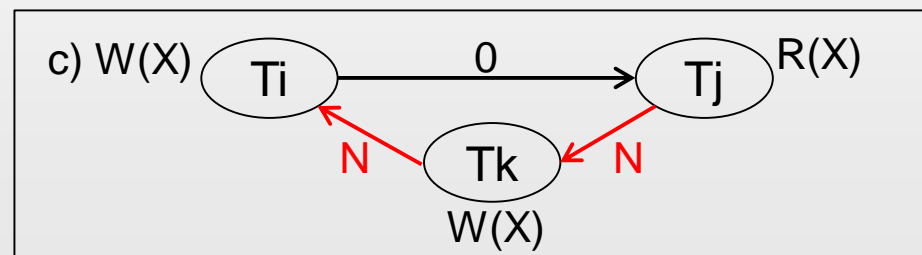
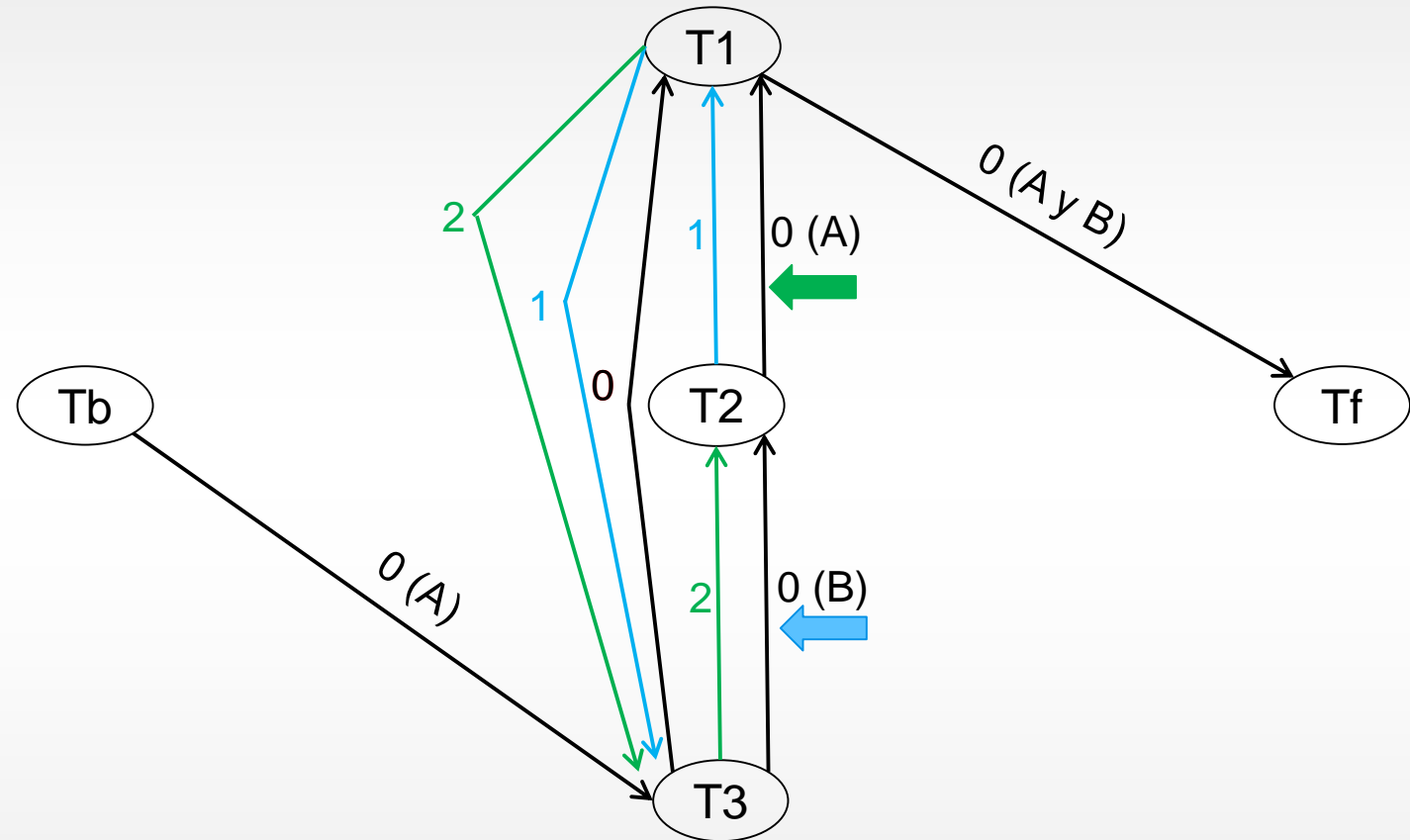
Prueba serializabilidad en Vistas

Grafo de precedencia etiquetado

Tb (escribe A y B)

	T1	T2	T3
1			R(A)
2		W(A)	
3			W(B)
4	R(A)		
5		R(B)	
6	W(B)		
7			W(A)
8	W(A)		

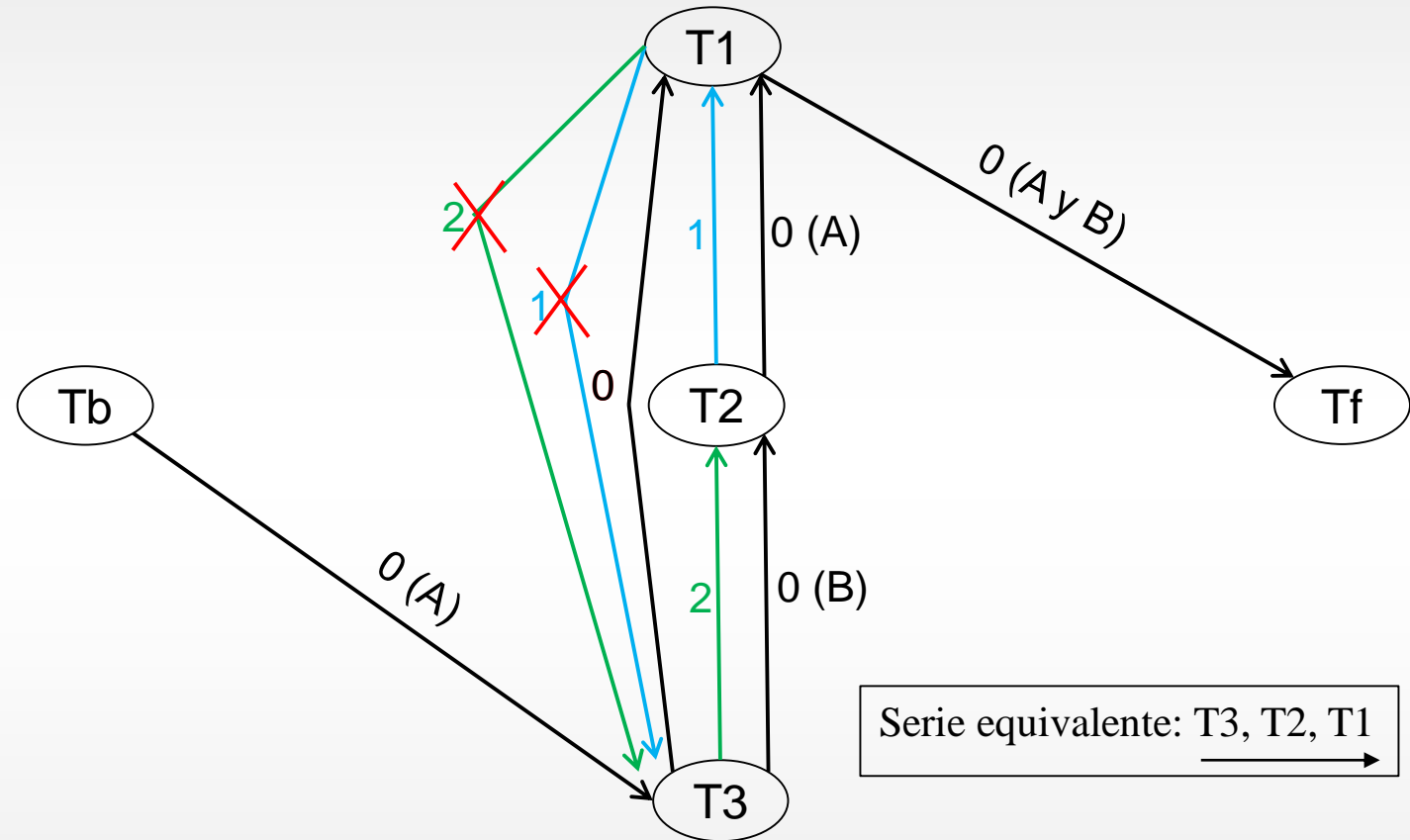
Tf (lee A y B)



Prueba serializabilidad en Vistas

Grafo de precedencia etiquetado

	T1	T2	T3
1			R(A)
2		W(A)	
3			W(B)
4	R(A)		
5		R(B)	
6	W(B)		
7			W(A)
8	W(A)		



Cada arco etiquetado con $N > 0$ representa una opción, solo debe quedar uno de los arcos por cada $N > 0$:

- Si logro obtener un grafo sin ciclos eliminando un arco por cada N entonces es serializable en vistas.
- Sino no se puede obtener un grafo sin ciclos entonces **no** es serializable en vistas.