**TP1-T4 (Teoría Control de Concurrencia de Transacciones Parte I)**

**1. Cual la diferencia entre un SGBD Mono Usuario y un SGBD Multiusuario**

Los sistemas monousuarios son aquellos que solo atienden a un usuario a la vez, y su principal uso se da en los ordenadores personales, mientras el usuario lo esté utilizando, no permitirá ningún otro acceso.  SGBD Multiusuario. Varios usuarios pueden tener acceso a la base de datos de manera concurrente para leer la información o modificarla.

**2. Explique el concepto de Multiprogramación en los Sistema Operativos**

También llamado sistemas operativos multitarea o multiproceso, se distingue por la habilidad de soportar dos o más procesos activos simultáneamente. El término multiprogramación denota un sistema operativo que, además de soportar procesos concurrentes múltiples, permite que residan simultáneamente en la memoria primaria las instrucciones y los datos procedentes de dos o más procesos distintos.

El concepto de multiprogramación no se debe confundir con operación de multiproceso. Si bien el primer término implica el último, no sucede en viceversa. La operación multiproceso es la gestión de la totalidad de los recursos del sistema de la computadora mediante la implementación de cauce segmentado.

**3. Explique los conceptos de Concurrencia Intercalada y Paralela**

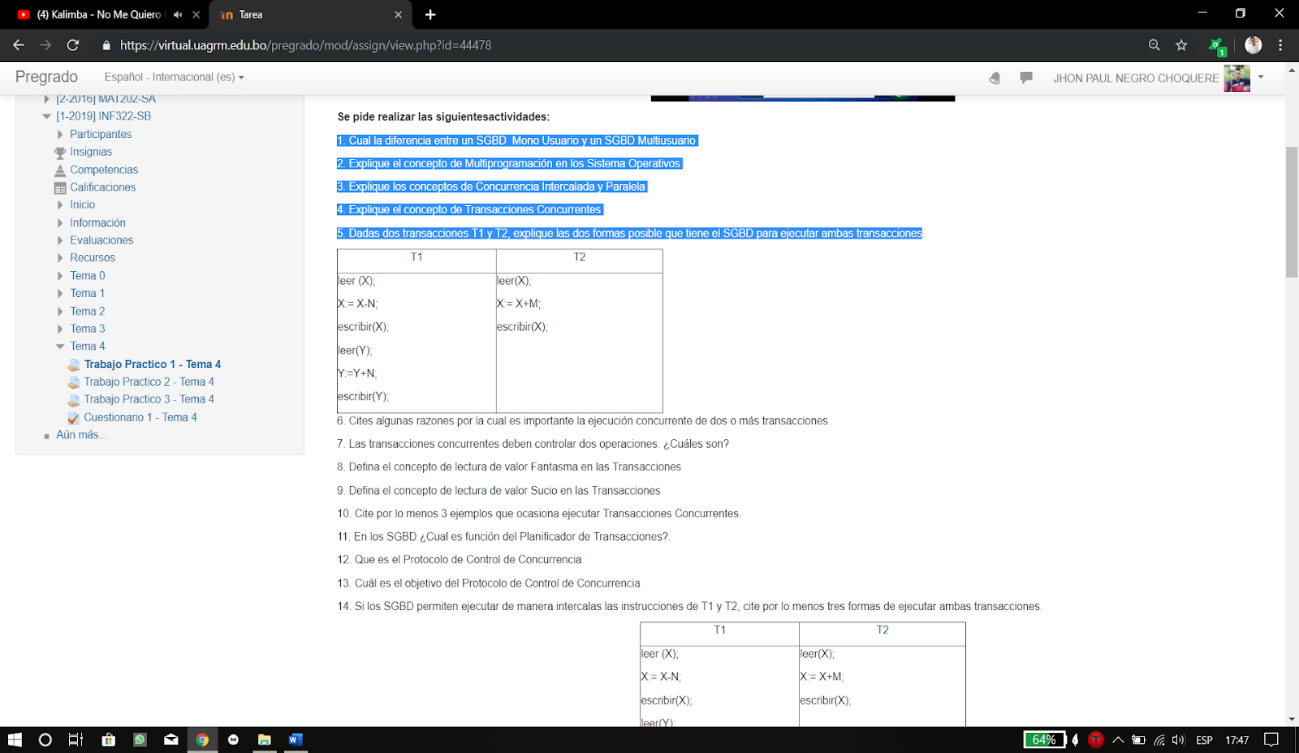
**Intercalada:** Las ejecuciones intercaladas cuyos efectos son los mismos que las ejecuciones seriales son denominadas *serializables* y son correctos ya que soportan la ilusión de la atomicidad de las transacciones.

**Paralela:** Es un programa concurrente en el que hay más de un contexto de ejecución o hebra activo simultáneamente; desde un punto de vista semántica no hay diferencia entre un programa paralelo y concurrente.

**4. Explique el concepto de Transacciones Concurrentes**

Concurrencia se refiere al hecho de que los Sistemas Administradores de Base de Datos permiten que muchas transacciones accedan a una misma Base de Datos a la vez. Cuando existen varios usuarios intentando modificar los datos al mismo tiempo, se necesita establecer algún tipo de control para que dichas modificaciones de un usuario no interfieran en las de los otros, a este sistema se le denomina control de concurrencia.

**5. Dadas dos transacciones T1 y T2, explique las dos formas posibles que tiene el SGBD para ejecutar ambas transacciones**



**Ejecución en secuencia:** se realiza primero la transacción T1 luego la transacción T2

**Ejecución intercalada:** se ejecuta parte de las instrucciones de T1 luego se ejecuta parte de las instrucciones de T2 y así hasta acabar ambas transacciones.

**6. Cites algunas razones por la cual es importante la ejecución concurrente de dos o más transacciones.**

\* Aumentar la productividad: número de transacciones ejecutadas por minuto.

\* Aumentar la utilización de la CPU (menos tiempo ocioso) y control de disco.

\* Reduce el tiempo medio de respuesta de transacciones (las pequeñas no esperan a las grandes)

**7. Las transacciones concurrentes deben controlar dos operaciones. ¿Cuáles son?**

La lectura y escritura que se realizan en la base de datos para proteger la consistencia de la misma.

**8. Defina el concepto de lectura de valor Fantasma en las Transacciones**

Una *lectura fantasma* ocurre cuando, durante una transacción, se ejecutan dos consultas idénticas, y los resultados de la segunda no son iguales a de los de la primera.

Esto puede ocurrir cuando no se realizan *bloqueos de rango* al realizar una operación [*SELECT*](https://es.wikipedia.org/w/index.php?title=SELECT&action=edit&redlink=1)*... WHERE*.

La anomalía de las *lecturas fantasma* es una caso particular de las *lecturas no repetibles* cuando la transacción 1 repite una consulta acotada en rango [*SELECT*](https://es.wikipedia.org/w/index.php?title=SELECT&action=edit&redlink=1)*... WHERE* y, entre ambas operaciones la transacción 2 crea (i.e. [INSERT](https://es.wikipedia.org/w/index.php?title=INSERT&action=edit&redlink=1)) nuevas filas (en la misma tabla) que entran dentro de esa cláusula *WHERE*.

**9. Defina el concepto de lectura de valor Sucio en las Transacciones**

Una lectura sucia ocurre cuando se le permite a una transacción la lectura de una fila que ha sido modificada por otra transacción concurrente pero todavía no ha sido cometida.

Las lecturas sucias funcionan de modo similar a las lecturas no repetibles; sin embargo, la segunda transacción no necesita ser cometida para que la primera dé un resultado diferente. Lo único que se puede prevenir en el nivel de aislamiento LECTURAS NO COMETIDAS es que las actualizaciones aparezcan en desorden en el resultado; esto es, que las primeras actualizaciones siempre aparecerán antes que las actualizaciones posteriores.

**10. Cite por lo menos 3 ejemplos que ocasiona ejecutar Transacciones Concurrentes.**

**\* Problema de actualización perdida.** Una transacción cambio un dato, pero no la guardo, viene otra transacción, hace operaciones con ese dato, y lo guarda.

**\* Problemas de actualización temporal o lectura sucia.** Una transacción T1 cambia un dato y lo guarda, la T2 lee ese dato lo actualiza y lo guarda, pero T1 deshace su transferencia y restaura el dato, pero T2 ya tiene el mismo dato con otro valor.

**\* Problemas de resumen incorrecto.** Cuando una T1 usa una función y le suma un dato, pero T2 cambia el dato sumado anteriormente, la suma será incorrecta debido a que se usó un dato antes de que lo actualicen.

**\* Problema de lectura no repetible.** Se leen dos vece el mismo dato y dan resultados diferentes debido a que una transacción realizo cambios en dicho dato.

**11. En los SGBD ¿Cual es función del Planificador de Transacciones?**

Controla el acceso concurrente de las transacciones en los SGBD, el planificador crea agendas, secuencias ordenadas de Lectura/Escritura tomadas por una o más transacciones.

Si el dato esta en el buffer el planificador interactúa con ellos o en su defecto solicita la lectura de los datos en el disco.

**12. Que es el Protocolo de Control de Concurrencia**

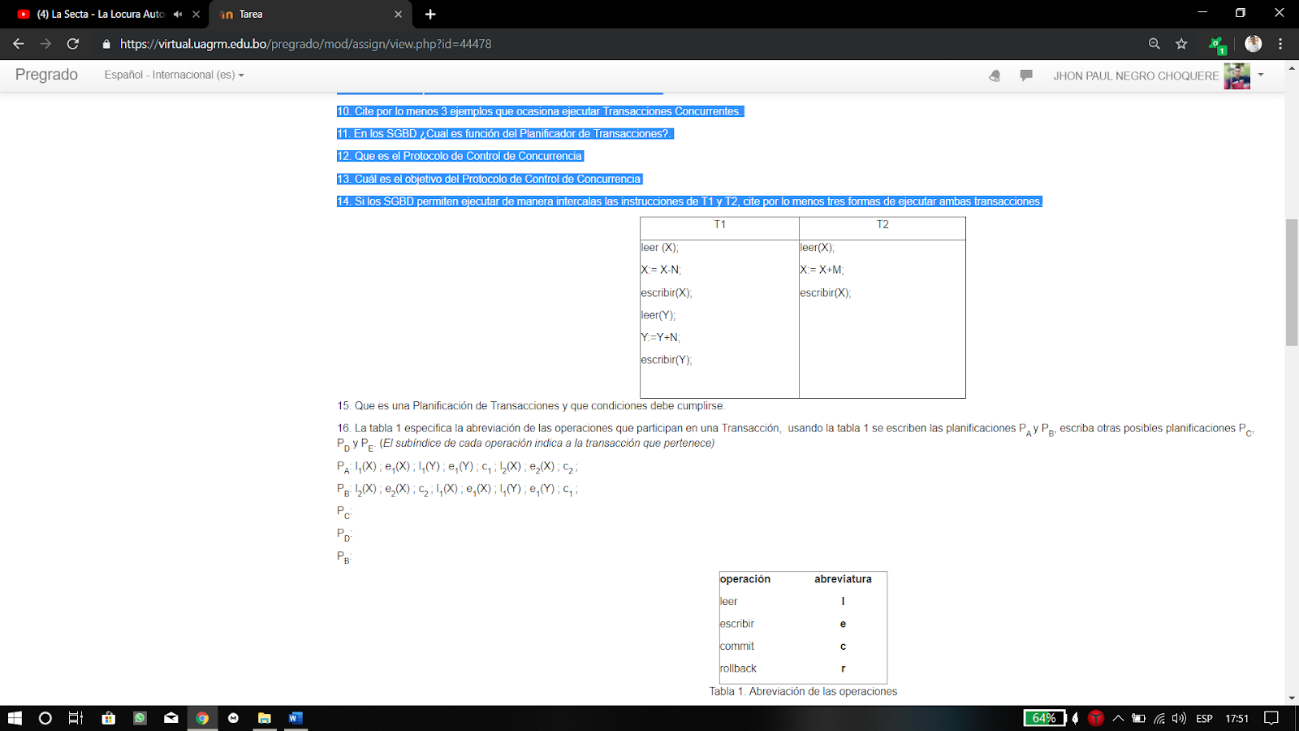
Es un mecanismo que evita que se destruya la consistencia de la base de datos.

**13. Cuál es el objetivo del Protocolo de Control de Concurrencia**

**\*** Planificar las transacciones de forma que no ocurran interferencias entre ellas, y así evitar la aparición de los problemas mencionados anteriormente.

\* Solución obvia: no permitir intercalación de operaciones de varias transacciones.

**14. Si los SGBD permiten ejecutar de manera intercaladas las instrucciones de T1 y T2, cite por lo menos tres formas de ejecutar ambas transacciones.**



**PA:** l1(X); e1(X); l2(X); e2(X); c2; l1(Y); e1(Y); c1;

**PB:** l1(X); e1(X); l1(Y); l2(X); e2(X); e1(Y); c1; c2;

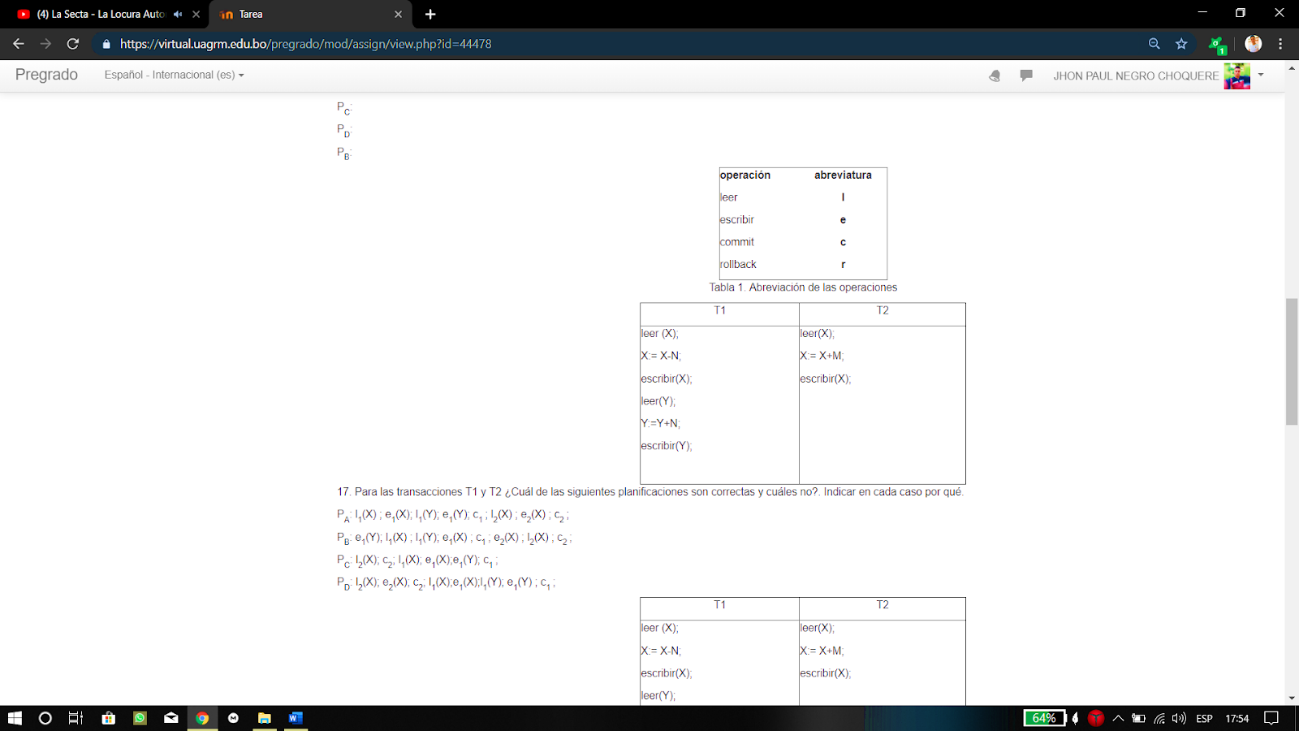
**PC:** l2(X); e2(X); l1(X); e1(X); l1(Y); c2; e1(Y); c1;

**15. Que es una Planificación de Transacciones y que condiciones debe cumplirse.**

Representan el orden cronológico en el que se ejecutan las instrucciones de diferentes transacciones concurrentes.

Una transacción **P** de **n** transacciones concurrentes T1, T2… Tn es una secuencia de las operaciones realizada por dichas transacciones, sujetas a la restricción de que:

Para cada transacción **T** que participa en P, sus operaciones aparecen en **P** en el mismo orden en el que ocurre en Ti.

**16. La tabla 1 especifica la abreviación de las operaciones que participan en una Transacción, usando la tabla 1 se escriben las planificaciones PAy PB, escriba otras posibles planificaciones PC, PD y PE. (*El subíndice de cada operación indica a la transacción que pertenece)***

**PA: l1(X); e1(X); l1(Y); e1(Y); c1; l2(X); e2(X); c2;**

**PB: l2(X); e2(X); c2; l1(X); e1(X); l1(Y); e1(Y); c1;**

**PC:** l1(X); e1(X); l2(X); e2(X); c2; l1(Y); e1(Y); c1;

**PD:** l1(X); e1(X); l1(Y); l2(X); e2(X); e1(Y); c1; c2;

**PE:** l2(X); e2(X); l1(X); e1(X); l1(Y); c2; e1(Y); c1;

**17. Para las transacciones T1 y T2 ¿Cuál de las siguientes planificaciones son correctas y cuáles no? Indicar en cada caso por qué.**

**PA: l1(X); e1(X); l1(Y); e1(Y); c1; l2(X); e2(X); c2;**

**PA** es correcta porque la transacción se ejecuta de forma secuencial, es decir, una después de la otra.

**PB: e1(Y); l1(X); l1(Y); e1(X); c1; e2(X); l2(X); c2;**

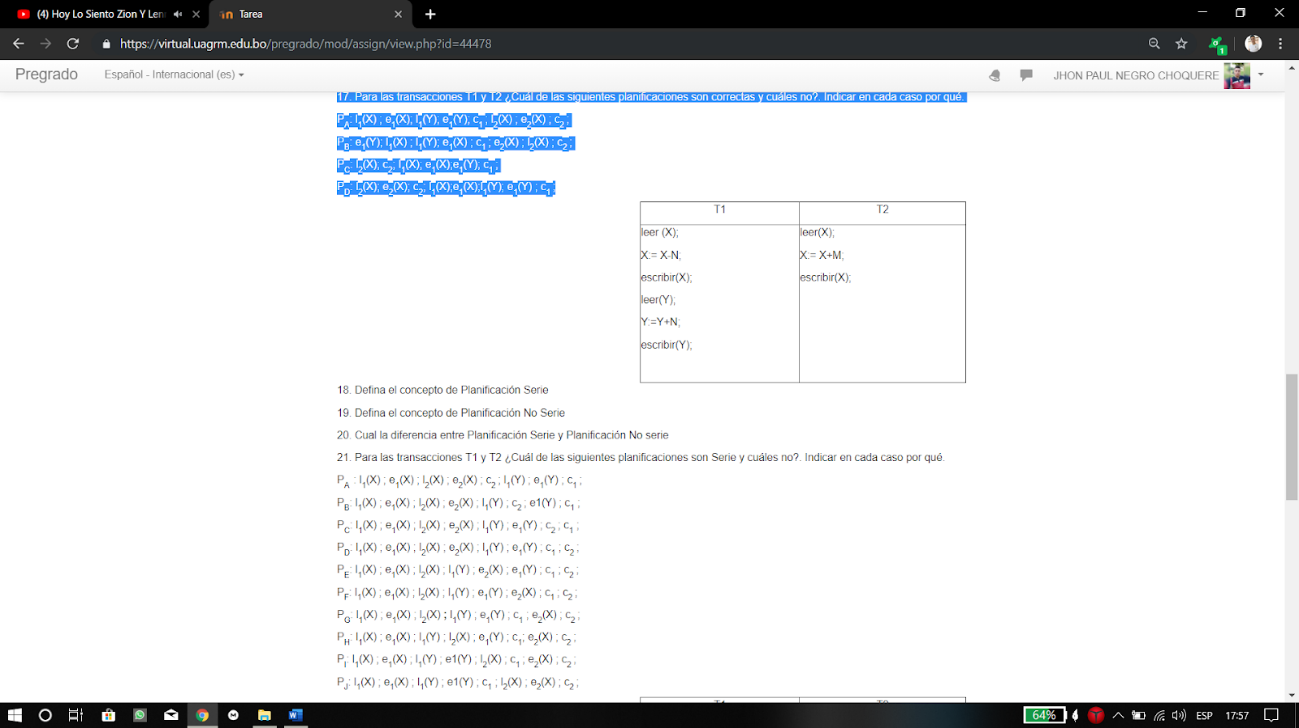
**PB** es incorrecta debido a que la primera instrucción es escribir Y, en la transacción esa instrucción esta en medio de T1

**PC: l2(X); c2; l1(X); e1(X); e1(Y); c1;**

**PC** es incorrecta, faltan instrucciones.

**PD: l2(X); e2(X); c2; l1(X); e1(X); l1(Y); e1(Y); c1;**

**PD** es correcta, al igual que **PA** las transacciones se ejecutan en forma secuencial.



**18. Defina el concepto de Planificación Serie**

Es aquella en que las operaciones de cada transacción se ejecutan consecutivamente sin que se intercalen operaciones de otras transacciones.

**19. Defina el concepto de Planificación No Serie**

Es aquella en la que las operaciones de un conjunto de transacciones concurrentes se ejecutan intercaladas.

**20. Cual la diferencia entre Planificación Serie y Planificación No serie**

La planificación en serie implica que la segunda transacción debe esperar que la primera termine para ser ejecutada, en la planificación no serie se ejecutan ambas transacciones al mismo tiempo.

**21. Para las transacciones T1 y T2 ¿Cuál de las siguientes planificaciones son Serie y cuáles no? Indicar en cada caso por qué.**

**PA: l1(X); e1(X); l2(X); e2(X); c2; l1(Y); e1(Y); c1;**

No es serie, porque la primera instrucción de la transacción 2 se ejecuta antes que termine la transacción 1.

**PB: l1(X); e1(X); l2(X); e2(X); l1(Y); c2; e1(Y); c1;**

No es serie, porque la primera instrucción de la transacción 2 se ejecuta antes que termine la transacción 1.

**PC: l1(X); e1(X); l2(X); e2(X); l1(Y); e1(Y); c2; c1;**

No es serie, porque la primera instrucción de la transacción 2 se ejecuta antes que termine la transacción 1.

**PD: l1(X); e1(X); l2(X); e2(X); l1(Y); e1(Y); c1; c2;**

No es serie, porque la primera instrucción de la transacción 2 se ejecuta antes que termine la transacción 1.

**PE: l1(X); e1(X); l2(X); l1(Y); e2(X); e1(Y); c1; c2;**

No es serie, porque la primera instrucción de la transacción 2 se ejecuta antes que termine la transacción 1.

**PF: l1(X); e1(X); l2(X); l1(Y); e1(Y); e2(X); c1; c2;**

No es serie, porque la primera instrucción de la transacción 2 se ejecuta antes que termine la transacción 1.

**PG: l1(X); e1(X); l2(X); l1(Y); e1(Y); c1; e2(X); c2;**

No es serie, porque la primera instrucción de la transacción 2 se ejecuta antes que termine la transacción 1.

**PH: l1(X); e1(X); l1(Y); l2(X); e1(Y); c1; e2(X); c2;**

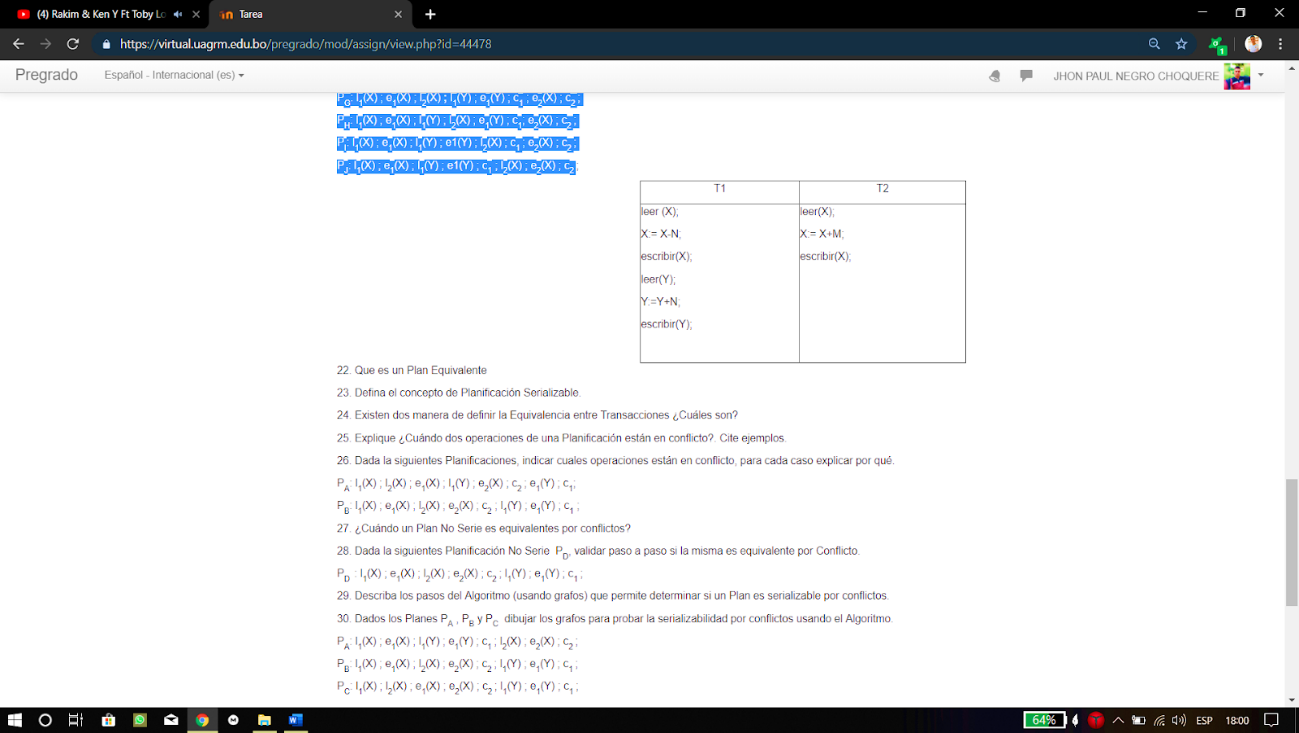
No es serie, porque la primera instrucción de la transacción 2 se ejecuta antes que termine la transacción 1.

**PI: l1(X); e1(X); l1(Y); e1(Y); l2(X); c1; e2(X); c2;**

No es serie, porque la primera instrucción de la transacción 2 se ejecuta antes que termine la transacción 1.

**PJ: l1(X); e1(X); l1(Y); e1(Y); c1; l2(X); e2(X); c2**

Es serie, la primera instrucción de la segunda transacción se ejecuta después que termina la primera transacción.



**22. Que es un Plan Equivalente**

Es aquel que produce los mismos efectos en la Base de Datos.

**23. Defina el concepto de Planificación Serializable.**

Una planificación **P** (no serie) es serializable si es equivalente a una planificación serie de las mismas **n** transacciones.

**24. Existen dos maneras de definir la Equivalencia entre Transacciones ¿Cuáles son?**

**\*** Una planificación que no es equivalente a ninguna ejecución en serie, es una planificación NO SERIALIZABLE.

\* Toda planificación serializable es CORRECTA.

**25. Explique ¿Cuándo dos operaciones de una Planificación están en conflicto? Cite ejemplos.**

En una planificación dos operaciones están en conflicto si pertenecen a diferentes transacciones. Ejemplo

Tienen acceso al mismo elemento **Leer(X)** y al menos una de ella es **Escribir(X).**

**l1(X); e2(X);**

**26. Dada la siguientes Planificaciones, indicar cuales operaciones están en conflicto, para cada caso explicar por qué.**

**PA: l1(X); l2(X); e1(X); l1(Y); e2(X); c2; e1(Y); c1;**

Esta en conflicto, debido a que la tercera instrucción escribe X para T1 cambiando el dato de X para la T2 que ya ha leído un dato antiguo en la segunda instrucción.

**PB: l1(X); e1(X); l2(X); e2(X); c2; l1(Y); e1(Y); c1;**

No esta en conflicto, las instrucciones se ejecutan de manera intercalada.

**27. ¿Cuándo un Plan No Serie es equivalentes por conflictos?**

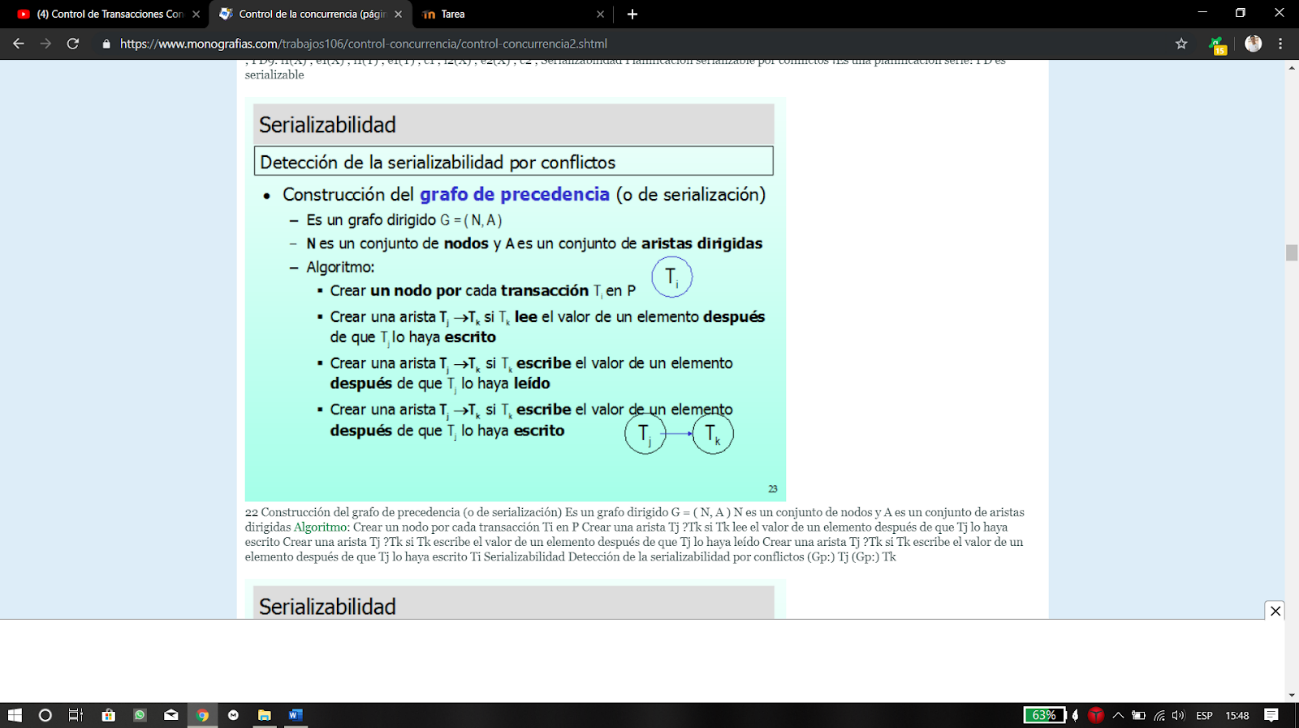
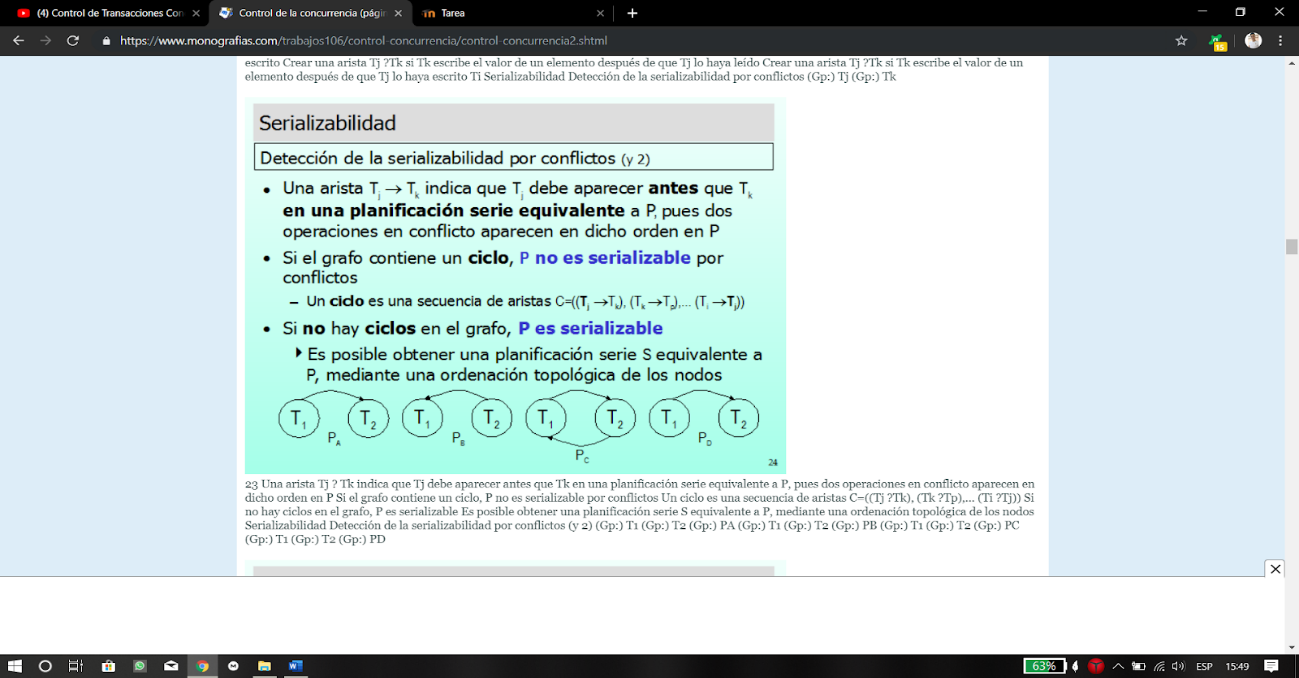
Dos planes son equivalentes por conflictos si el orden de cualesquiera dos operaciones en conflicto es el mismo en ambos planes.

**28. Dada la siguientes Planificación No Serie PD, validar paso a paso si la misma es equivalente por Conflicto.**

**PD: l1(X); e1(X); l2(X); e2(X); c2; l1(Y); e1(Y); c1;**

No es equivalente por conflicto, las operaciones se realizan de forma intercalada.

**29. Describa los pasos del Algoritmo (usando grafos) que permite determinar si un Plan es serializable por conflictos.**

**30. Dados los Planes PA, PB y PC dibujar los grafos para probar la serializabilidad por conflictos usando el Algoritmo.**

**PA: l1(X); e1(X); l1(Y); e1(Y); c1; l2(X); e2(X); c2;**

**PB: l1(X); e1(X); l2(X); e2(X); c2; l1(Y); e1(Y); c1;**

**PC: l1(X); l2(X); e1(X); e2(X); c2; l1(Y); e1(Y); c1**