
 <b>Universidad de los Andes</b> Facultad de Ingeniería	Ingeniería de Sistemas y Computación <b>Pregrado</b> <b>ISIS2304 – Sistemas Transaccionales</b>	
---	---	---

## PROYECTO DEL CURSO – ENTREGA 3 DEL PROYECTO

Esta 3ra entrega del proyecto consta de dos partes, las cuales se describen a continuación:

- La parte A consiste en
  - Completar y hacer funcionar todos los requerimientos funcionales de la aplicación informática desarrollada para la entrega 2. Esto tiene un valor del 30% de la entrega 3
  - Ejecutar una simulación de las transacciones asociadas al RF6 del caso de estudio del Banco de los Andes. Esto tiene un valor del 20% de la entrega 3.
- La parte B consiste en implementar transacciones para algunos de los requerimientos funcionales de la aplicación desarrollada para el Banco de los Andes y en diseñar escenarios de concurrencia.

### OBJETIVOS

- Parte A - Completar la aplicación informática de la entrega 2 e identificar el nivel de aislamiento a utilizar en la operación de transferencia entre cuentas del RF6.
- Parte B - Implementar transacciones sobre dicha aplicación usando niveles de aislamiento y candados.

### HERRAMIENTAS

- Oracle 12c disponible en la infraestructura computacional del Departamento
- SQL Developer, disponible en los laboratorios del Departamento y disponible para descarga en el sitio Web de Oracle
- Debe usar el IDE (Visual Studio Code) y framework (Spring) de desarrollo indicados en el proyecto guiado de Coursera
- Para la implementación transaccional de esta entrega, pueden usar como guía los recursos de ayuda que serán proveídos por la coordinación del curso. Estos son solo una sugerencia y presentan la implementación de la gestión transaccional de algunas funcionalidades adicionales de parranderos a través de una capa de servicios.

### CASO DE ESTUDIO

La iteración se desarrolla utilizando el caso **Banco de los Andes**, que se encuentra en Bloque Neón (BN) - Contenidos.

### ARQUITECTURA DE LA APLICACIÓN

Disponible en la documentación y archivos de base disponibles en BN.

### PARTE A: COMPLETAR LA CONSTRUCCIÓN DE LA APLICACIÓN

Revise el caso de estudio propuesto. Identifique los elementos fundamentales que hacen parte del negocio que se describe: entidades de negocio, funcionalidades principales, y reglas de negocio.

1. **(30%)<sup>1</sup>** Complete la aplicación desarrollada para la entrega 2 usando Java Spring y siguiendo la arquitectura de aplicación de ejemplo (ver proyecto guiado de coursera). El objetivo es que el grupo sea capaz de implementar por

<sup>1</sup> La aplicación debe tener interfaces que permitan al usuario interactuar con los datos desde un navegador web (ver proyecto guiado de coursera). A través de estas interfaces se debe poder poblar la BD y modificar los datos (RFs) y acceder a las consultas (RFCs)..

completo y hacer funcionar los requerimientos funcionales de modificación (RF1 a RF9) y los requerimientos de consulta (RFC1 a RFC3) descritos en el documento marco.

- Evaluación detallada: 2.5% para cada requerimiento funcional de modificación (RF1 a RF9) y para cada requerimiento funcional de consulta (RFC1 a RFC3).
- Los RF y RFC que ya funcionaron en la entrega 2 y en los cuales se sacaron 5.0 son un avance para esta entrega y no necesitan ser reescritos. El grupo ya tendría una nota de 5.0 para esos requerimientos en esta entrega.
- Debe entregar la documentación de los RF y RFC que no funcionaron en la entrega 2 y fueron completados.

2. **(20%) Nivel de aislamiento.** Con la información de datos de prueba que tiene en su Base de datos para el proyecto del Banco de los Andes y usando SQL Developer, haga:

- Habrá dos sesiones en SQL Developer con el mismo usuario de base de datos y asegúrese que el autocommit esté deshabilitado en cada una de las sesiones.
- Suponga que se inician dos transacciones concurrentes (una en cada sesión abierta en developer) para registrar operaciones sobre una misma cuenta. Esto significa, por ejemplo, que los usuarios 1 (Cliente) y 2 (Banco) iniciaron casi al mismo tiempo los registros de operaciones 1 y 2 respectivamente, sobre la CUENTA\_1 y la CUENTA\_2
- En cada sesión ejecute una a una las operaciones de las transacciones concurrentes de manera intercalada según lo descrito abajo con el fin de simular la ejecución concurrente de las dos transacciones.

La siguiente tabla indica lo que debe realizar en cada sesión y en cada momento de tiempo, teniendo en cuenta que debe hacerlo para los niveles de aislamiento vistos en clase y soportados por la versión de Oracle que tenemos, es decir, el primer caso de simulación debe definir es un esquema de aislamiento: Serializable – Serializable (sesión 1 y sesión 2) y el segundo caso Read Committed - Read Committed (sesión 1 y sesión 2).

En la siguiente tabla, en la sesión S1 la transacción T1 va a realizar dos operaciones: la primera para consignar 1 millón de pesos en la CUENTA\_1 y la segunda para retirar 50.000 pesos de la CUENTA\_2.

De manera concurrente, en la sesión S2, la transacción T2 va a realizar igualmente dos operaciones: la primera para retirar 30.000 pesos de cuota de manejo de la cuenta CUENTA\_1 y la segunda para consignar 5.000 de intereses sobre la CUENTA\_2.

Cada una de estas 4 operaciones requiere dos sentencias SQL, una que ajusta el saldo de la cuenta y otra que agrega el respectivo registro al log de operaciones de cuentas.

Sesión 1 (S1)	T	Sesión 2 (S2)
Defina nivel de Aislamiento para iniciar Bloque de Transacción T1 en S1	t1	
<b>S1T1 - Operación 1: modificar el saldo</b> 1. Seleccione una cuenta que llamemos CUENTA_1 específica de su Base de datos sobre la cual quiere hacer esta consignación y elabore la sentencia SQL respectiva que modifica el saldo de esta cuenta sumándole 1 millón de pesos.	t2	
<b>S1T1 - Operación 1: registro en el log</b> 2. Para la misma CUENTA_1 específica de su Base de datos sobre la cual quiere hacer esta consignación de 1 millón de pesos, elabore la sentencia SQL para insertar en el log de operaciones sobre cuentas el registro de la consignación realizada.	t3	

<b>S1T1 – Operación 2: modificar el saldo</b> 1. Seleccione una cuenta que llamemos CUENTA_2 ( <i>diferente a la cuenta CUENTA_1</i> ) específica de su Base de datos y elabore la sentencia SQL respectiva que modifica el saldo de la cuenta restándole 50.000.	t4	Defina nivel de Aislamiento para iniciar Bloque de Transacción T2 en la sesión S2
<b>S1T1 - Operación 2: registro en el log</b> 2. Para la misma CUENTA_2 específica de su Base de datos sobre la cual quiere hacer este retiro, elabore la sentencia SQL para insertar en el log de operaciones sobre cuentas el registro del retiro de 50.000 pesos.	t5	
	t6	<b>S2T2 - Operación 1: modificar el saldo</b> Para la misma CUENTA_1 (que escogió en la Sesión 1, tiempo t2), elabore la sentencia SQL respectiva que modifica el saldo de esta cuenta específica de su Base de datos, para descontar 30.000 pesos del saldo correspondientes a la cuota de manejo de la cuenta CUENTA_1.
Cierre exitosamente este primer Bloque de Transacción S1T1(se confirma el registro de las operaciones)	t7	
	t8	<b>S2T2 - Operación 1: registro en el log</b> Para la misma CUENTA_1 específica de su Base de datos elabore la sentencia SQL para insertar en el log de operaciones sobre cuentas el registro del retiro de 30.000 pesos correspondientes a la cuota de manejo de la CUENTA_1.
	t9	<b>S2T2 - Operación 1: modificar el saldo</b> Para la misma CUENTA_2 específica de su Base de datos, elabore la sentencia SQL respectiva que modifica el saldo de esta cuenta sumándole 5.000 de intereses sobre la CUENTA_2.
	t10	<b>S2T2 - Operación 1: registro en el log</b> Para la misma CUENTA_2 específica de su Base de datos, elabore la sentencia SQL para insertar en el log de operaciones sobre cuentas el registro de la consignación de 5.000 de intereses realizada.
Consulte el saldo de la cuenta CUENTA_1 y CUENTA_2, para ello haga la consulta SQL respectiva	T1 1	
	T1 2	Cierre exitosamente este primer Bloque de Transacción S2T2 (se confirma el registro de las operaciones)
Consulte el saldo de la cuenta CUENTA_1 y CUENTA_2, para ello haga la consulta SQL respectiva	t13	Consulte el saldo de la cuenta CUENTA_1 y CUENTA_2, para ello haga la consulta SQL respectiva

De la anterior simulación de flujo concurrente y para cada nivel de aislamiento: Caso 1 Serializable – Serializable (sesión 1 y sesión 2) y Caso 2: Read Committed - Read Committed (sesión 1 y sesión 2) haga:

Un informe (según planilla adjunta) que explique en cada tiempo y sesión (1 y 2) que pasa con las operaciones que se hacen en cada momento, explicando los resultados obtenidos y el porqué de estos (por ejemplo, la operación se pudo hacer, hubo bloqueo, abrazos mortales, etc), puede adjuntar pantallazos del proceso.

Archivo .SQL, que contiene las sentencias sql de las diferentes operaciones sugeridas para cada sesión y tiempo definido y que permita cada sesión y nivel de aislamiento.

## PARTE B: IMPLEMENTACIÓN DE TRANSACCIONES EN LA APLICACIÓN

3. **(15%) Nuevo requerimiento funcional de consulta RFC4 – CONSULTA DE OPERACIONES REALIZADAS SOBRE UNA CUENTA - SERIALIZABLE:** como entrada, la aplicación recibe el número de la cuenta. El resultado debe ser la lista de operaciones realizadas (consignaciones, retiros, y transferencias), en los últimos 30 días. Así, por ejemplo, si la consulta se ejecuta el día 12 de abril, el reporte debe traer todas las operaciones existentes en el log de operaciones para la cuenta especificada desde el 13 de marzo hasta el momento exacto de ejecución de la consulta. Algunas de las características de esta consulta son:
  - Debe estar implementada a través de una transacción que tenga el máximo nivel de aislamiento (SERIALIZABLE), con el fin de evitar las anomalías vistas en clase (lecturas sucias, lecturas no repetibles y fantasmas).
  - La transacción debe contener un rollback que se ejecuta en caso en que no se pueda realizar la consulta. En ese caso, un mensaje informando al usuario debe ser presentado.
  - La transacción debe tener un temporizador de 30 segundos (ver función thread.sleep en Spring) que fuerce su tiempo de ejecución en esa longitud con el fin de poder hacer pruebas de concurrencia.
4. **(5%) Nuevo requerimiento funcional de consulta RFC5 – CONSULTA DE OPERACIONES REALIZADAS SOBRE UNA CUENTA – READ COMMITTED:** este requerimiento funcional es similar al anterior. Así, como entrada la aplicación recibe el número de la cuenta y el resultado debe ser la lista de operaciones realizadas (consignaciones, retiros, y transferencias), en los últimos 30 días. La diferencia consiste en que:
  - Debe estar implementado a través de una nueva transacción que tenga el nivel de aislamiento READ COMMITTED, el cual evita las lecturas sucias, pero permite las lecturas no repetibles y los fantasmas.
5. **(15%) Manejo transaccional para RF6:** Modifique los componentes de la aplicación que implementan el requerimiento **RF6 - REGISTRAR OPERACIÓN SOBRE CUENTA**, para implementar una gestión transaccional del mismo:
  - RF6 - REGISTRAR OPERACIÓN SOBRE CUENTA: este requerimiento funcional involucra 3 operaciones: retirar dinero de una cuenta, consignar dinero de una cuenta y transferir dinero de una cuenta a otra del mismo banco. El grupo debe crear una transacción diferente para cada una de estas 3 operaciones. Recuerden que cada operación involucra dos sentencias: la primera ingresa un registro en el log de operaciones sobre cuentas y la segunda actualiza el saldo de la cuenta (ver detalles en el caso de estudio Banco de los Andes).
  - En cada transacción, se hace commit únicamente si las dos sentencias pueden realizarse de manera efectiva o rollback de lo contrario.nivel
6. **(7.5%) Escenario de prueba de concurrencia 1:** escriba un escenario de prueba en el cual se ejecuta primero el RFC4 – CONSULTA DE OPERACIONES REALIZADAS SOBRE UNA CUENTA – **SERIALIZABLE**, y antes de que pasen los 30 segundos de ejecución de esta consulta, el usuario ejecuta de manera concurrente el componente que implementa RF6 - REGISTRAR OPERACIÓN SOBRE CUENTA (versión transaccional, solicitado en esta entrega), para hacer una consignación. Este escenario de prueba debe contener:
  - Los pasos para la ejecución concurrente de RFC4 y RF6
  - Descripción de lo sucedido: ¿acaso el componente que implementa RF6 debió esperar a que terminara la ejecución de la consulta RFC4 para poder registrar la consignación?
  - El resultado presentado por RFC4: presente el resultado de las operaciones para la cuenta ingresada por el usuario en el último mes. Diga si allí apareció la consignación ingresada por el usuario al ejecutar el componente que implementa RF6 de manera simultánea.
7. **(7.5%) Escenario de prueba de concurrencia 2:** escriba un escenario de prueba en el cual se ejecute primero el RFC5 – CONSULTA DE OPERACIONES REALIZADAS SOBRE UNA CUENTA – **READ COMMITTED**, y antes de que pasen los 30 segundos de ejecución de esta consulta, el usuario ejecuta de manera concurrente el componente

que implementa RF6 - REGISTRAR OPERACIÓN SOBRE CUENTA (versión transaccional, solicitado en esta entrega), para hacer una consignación. Este escenario de prueba debe contener:

- Los pasos para la ejecución concurrente de RFC5 y RF6
- Descripción de lo sucedido: ¿acaso el componente que implementa RF6 debió esperar a que terminara la ejecución de la consulta RFC5 para poder registrar la consignación?
- El resultado presentado por RFC5: presente el resultado de las operaciones para la cuenta ingresada por el usuario en el último mes. Diga si allí apareció la consignación ingresada por el usuario al ejecutar el componente que implementa RF6 de manera simultánea.

**Durante la sustentación, los estudiantes deberán hacer una demostración en tiempo real sobre la aplicación corriendo de los escenarios de prueba de concurrencia 1 y 2 entregados, esto hará parte de la nota de los entregables 6 y 7 y además servirá para ayudar a evaluar la implementación de los RF y RFC de esta entrega (ver rúbrica de la entrega.)**

## ENTREGA

Fecha y hora límite: **29 de abril de 2024**

- En la actividad correspondiente en BN, envíe la URL del *commit* de la entrega en el repositorio GitHub asignado a su grupo.
- Contenido de la entrega (*commit* de GitHub):
  - Documento de informe, debe estar en formato `.pdf`. Debe incluir:
    - Documentación de las clases de Java Spring para los RF y RFC completados
    - Planilla de informe del ejercicio de aislamiento y script SQL respectivo.
    - Documentación de las clases de Java Spring para los RF6 (versión transaccional solicitado para esta entrega) y los nuevos RFC4 y RFC5.
    - Documentación de los escenarios de prueba de concurrencia 1 y 2. Adicionalmente, los estudiantes deberán hacer una demostración en tiempo real sobre la aplicación de los escenarios de prueba de concurrencia 1 y 2 documentados y entregados, esto hará parte de la nota de los entregables 6 y 7 y ayudará a evaluar la implementación de los RF y RFC solicitados en este enunciado.
  - Proyecto Java con el código fuente y la documentación del proyecto de software descrita antes.
  - La documentación del proyecto debe quedar en una carpeta `docs`, en la estructura principal de directorios del proyecto java.
  - Archivo `readme.txt`, con todos los pasos a seguir para instalar la aplicación.
- El no seguimiento de los lineamientos de formato de entrega tiene una penalización de 0.5/5.0 puntos en la nota, por cada elemento faltante.

## DOCUMENTACIÓN

Tutoriales de acceso a los servicios de Oracle en el sitio Web del curso

Documentación en línea de Oracle 12c:  
<http://www.oracle.com/technetwork/database/enterprise-edition/documentation/index.html>