

Capa de Controller:

RFC2:

1. consultarClienteCompleto(numDocumento)

- Descripción: este metodo maneja solicitudes GET, el cual muestra toda la información del cliente
- Parametros: String numDocumento
- Funcionalidad: se puede observar todos los productos e información de un cliente en específico.

RFC3:

1. extracto(numCuenta)

- descripción: este método maneja una solicitud GET, el cual busca todas las operaciones hechas por una cuenta en específico.
- Parámetros: Integer numCuenta.
- Funcionalidad: se puede observar las operaciones tipo retiros y consignación, además se ven todas las transacciones hechas por la cuenta.

Capa de Servicios: Detalles de Transaccionalidad

RF6:

1. operacionCuentaGuardar(operacionCuenta, idPuntoAtencion, numCuenta)

- Nivel de aislamiento: SERIALIZABLE
- Descripción: recibe una operacionCuenta, el id del punto de atención y el numero de cuenta, lo que hace es hacer un setCuenta y setPuntoAtencion en operacionCuenta, luego revisa si la cuenta esta activa, si es un retiro revisa que el saldo sea suficiente y si es una consignación solo añade el valor a el saldo de la cuenta.

2. OperacionTransferenciaGuardar(operacionTransferencia, idPuntoAtencion, numCuentaOrigen, numCuentaDestino)

- Nivel de aislamiento: SERIALIZABLE
- Descripción: recibe una operacionCuenta, el id del punto de atención, el número de cuenta de origen y el número de cuenta de destino, lo que hace es hacer un setCuenta y setPuntoAtencion en operacionCuenta, luego revisa si la cuenta esta activa, si es un retiro revisa que el saldo sea suficiente y si es una consignación solo añade el valor a el saldo de la cuenta.

RFC4:

1. darOperaciones(numCuenta)

- Nivel de aislamiento: SERIALIZABLE
- Descripción: recibe un numero de cuenta, luego que buscan las operaciones de la cuenta y las transferencias que haya hecho esa cuenta, luego hay un tiempo de espera de 60 segundos y vuelve a buscar las operaciones de la cuenta y las transferencias.
- Uso en tests: precisamente se hace un tiempo de espera para que se puedan hacer otras transacciones y testear el bloqueo de lectura de serializable.

RFC5:

1. darOperacionesCommitted(numcuenta)

- Nivel de aislamiento: READ_COMMITTED
- Descripción: recibe un número de cuenta, luego que buscan las operaciones de la cuenta y las transferencias que haya hecho esa cuenta, luego hay un tiempo de espera de 60 segundos y vuelve a buscar las operaciones de la cuenta y las transferencias.

- Uso en tests: se puede usar para probar las lecturas fantasmas cuando se hace otra transacción al tiempo.

Pruebas de concurrencia:

Serializable:

Pasos:

- **1:** Se corre el RFC4 desde el login como administrador con el botón “Generar Extracto”. Este empieza el tiempo de espera de la transacción.
- **2:** Luego se empiezan los pasos para el RF6, Logeamos como cajero y nos vamos a operacionesCuentas, le damos al botón “Añadir operación”, luego creamos la operación que sea un “retiro” y esto va a tener cierto tiempo de espera.
- **3:** Volvemos a crear una operación Cuenta, pero esta vez va a ser una “consignación” y esto va a tener cierto tiempo de espera.
- **4:** Ahora entramos a operaciones Transferencia y creamos una.

Descripción:

Podemos observar que la transacción de RFC4 se queda bloqueada gracias al tiempo de espera que se puso, además de esto podemos observar que todas las transacciones de RF6 se hicieron con normalidad, pero al terminar de cargar el RFC4 podemos observar como en el extracto de movimientos no podemos observar las operaciones realizadas en RF6 esto es gracias al nivel de Serializable que evita que se vean los cambios de otra transacción ya que se hacen de manera aislada.

En conclusión, no se vieron los cambios hechos en RF6 al terminar la transacción de RFC4 gracias al nivel de aislamiento que hace que las transacciones solo vean los resultados que existían a la hora de empezar la transacción para así evitar “lecturas sucias”, “lecturas no repetibles” y “lecturas fantasmas”.

Read Committed:

Pasos:

- **1:** Se corre el RFC4 desde el login como administrador con el botón “Generar Extracto”. Este empieza el tiempo de espera de la transacción.
- **2:** Luego se empiezan los pasos para el RF6, Logeamos como cajero y nos vamos a operacionesCuentas, le damos al botón “Añadir operación”, luego creamos la operación que sea un “retiro” y esto va a tener cierto tiempo de espera.
- **3:** Volvemos a crear una operación Cuenta, pero esta vez va a ser una “consignación” y esto va a tener cierto tiempo de espera.
- **4:** Ahora entramos a operaciones Transferencia y creamos una.

Descripción:

Podemos observar que la transacción de RFC5 se queda bloqueada gracias al tiempo de espera que se puso, además de esto podemos observar que todas las transacciones de RF6 se hicieron con normalidad no tuvo que esperar a que la consulta se hiciera, pero al terminar de cargar el RFC5 podemos observar como en el extracto de movimientos están las operaciones realizadas en RF6, se esperaba que se vean lecturas fantasmas gracias al nivel de aislamiento “read committed” que se usó.

En conclusión, se vieron los cambios hechos en RF6 al terminar la transacción de RFC5, esto gracias al nivel de aislamiento “Read_Committed” que permite que hayan lecturas fantasmas en la transacción.