

“Proyecto Iteración 3”

Juan Camilo Eduardo Burbano

Daniel Antonio Pulido Nieto

Proyecto

Universidad de los Andes, Bogotá, Colombia

{j.burbanon, [d.pulidon](mailto:d.pulidon@uniandes.edu.co)}@uniandes.edu.co

Fecha de presentación: Noviembre 15 de 2021

Tabla de contenido

1	Introducción	1
2	Análisis	1
3	Diseño.....	4
4	Resultados Logrados	7
5	Resultados no Logrados.....	7
6	Balance Pruebas	8
7	Reglas de Negocio.....	8

1 Introducción

En el presente documento mostramos nuestros resultados de la tercera iteración del proyecto de sistemas transaccionales para el cual se tuvo que hacer uso de los temas y lecturas vistas antes y durante la preparación de este proyecto asimismo como de las iteraciones anteriores del proyecto 1 y 2.

2 Análisis

Para esta iteración, se hizo una modificación en la tabla de préstamo, para asignarle un gerente que abre el préstamo, y creamos las tablas Consigna, que guarda la relación entre un cliente que paga el salario a otro cliente, y Operaciones, que guarda la información de cada transacción que modifique la cantidad de dinero en una cuenta bancaria

PRESTAMO									
ID	TIPO	ESTADO	NOMBRE	MONTO	INTERES	NUMEROCUOTAS	DIAPAGA	VALORCUOTA	GERENTE
PK	NN	NN	FK (USUARIO, LOGIN)	NN	NN	NN	NN	NN	NN

CONSIGNA						
JEFE	IDJEFE	EMPLEADO	IDEMPLEADO	MONTO	FECHA	FRECUENCIA
NN	FK(CUENTA, ID)	NN	FK(CUENTA, ID)	NN	NN	NN

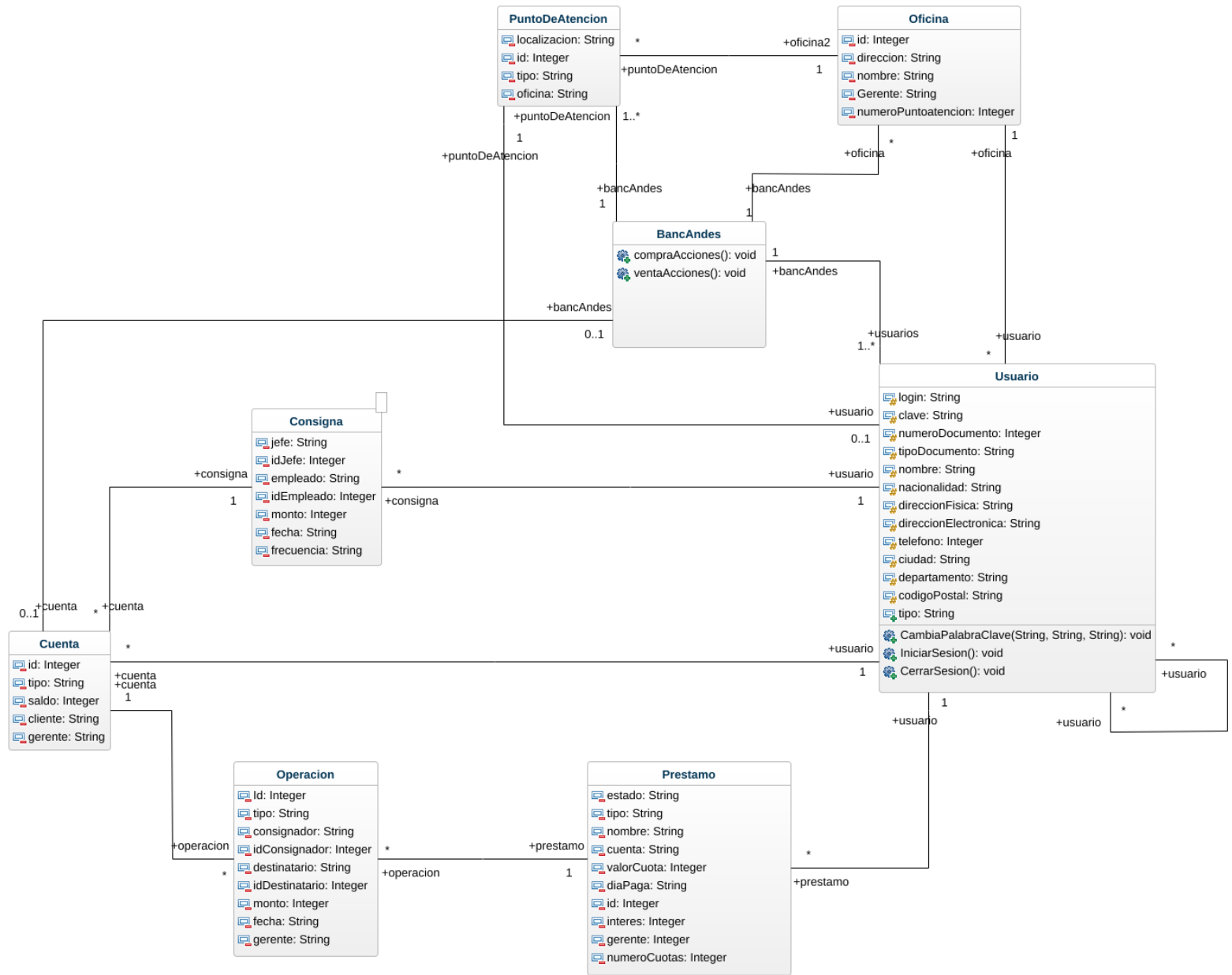
OPERACIONES								
ID	TIPO	CONSIGNADOR	IDCONSIGNADOR	DESTINATARIO	IDDESTINATARIO	MONTO	FECHA	GERENTE
FK	CK (PRESTAMO, CUENTA)	FK (USUARIO, LOGIN)	FK (CUENTA, ID)	FK (USUARIO, LOGIN)	FK (CUENTA, ID)	NN		NN

Oficina				
id	direccion	nombre	Gerente	numeroPuntoatencion
PK	NN,NC	NN,NC	NN	NN

Usuarios											
Login	Clave	Numero Documento	Tipo Documento	nombre	nacionalidad	Direccion Fisica	dirección Electrónica	telefono	ciudad	departamento	Codigo Postal
PK,UA	NN,UA	NN,NC	NN	NN	NN	NN	NN	NN	NN	NN	NN

Cuenta				
Id	Tipo	Saldo	Cliente	Gerente
PK	NN	NN	FK(Usuario , id),NN	NN

Punto de Atención			
id	tipo	localización	Oficina
FK	CK (Físico, virtual),NN	CK(Bancandes), NN	



3 Diseño

A nivel conceptual el impacto que tuvieron los requerimientos fue la integración de una nueva tabla “consigna” para cubrir los nuevos requerimientos que no se presentaban en la iteración anterior y que resultaba vital su creación para la solución del proyecto además de la modificación de algunas tablas que puedan cumplir con las actualizaciones a los requerimientos

Debido a que ya logramos implementar las tablas y columnas previstas para la construcción de los requerimientos solicitados se confirma que las tanto las tablas modificadas, así como creadas son BCNF además de que estas no presentan ningún problema frente a las operaciones transaccionales como inserción, borrado o actualización debido a las múltiples pruebas que se llevaron a cabo para verificar el funcionamiento de los requerimientos.

- RF10: Para cumplir con la propiedad de atomicidad se verifico en la persistencia que, tras una sentencia de SQL ejecutada, se tendrían que ejecutar todas antes de poder hacer commit, de lo contrario se haría rollback. Para cumplir con la propiedad de coherencia en este requerimiento se utilizó el atributo de fecha en la tabla de consignación para determinar después de registrar un día de pago, cuando sería el siguiente. Para cumplir con la propiedad de aislamiento se registró cada tupla de la tabla de manera que solo se pudieran hacer cambios sobre esta desde otro dispositivo cuando esta finalizara la transacción. Y finalmente para garantizar la durabilidad tras cada transacción se guarda la información en la base de datos con un commit y se registran estos cambios en los logs

RF11: Para cumplir con la propiedad de atomicidad se verifico en la persistencia que, tras una sentencia de SQL ejecutada, se tendrán que ejecutar todas antes de poder hacer commit, de lo contrario se haría rollback. Para cumplir con la propiedad de coherencia en este requerimiento se verifico que se restara exitosamente la misma cantidad de dinero de la cuenta del consignador que se le sumara a la cuenta del destinatario. Para cumplir con la propiedad de aislamiento se registró cada tupla de la tabla de manera que solo se pudieran hacer cambios sobre esta desde otro dispositivo cuando esta finalizara la transacción. Y finalmente para garantizar la durabilidad tras cada transacción se guarda la información en la base de datos con un commit y se registran estos cambios en los logs.

RF12:

Para cumplir las propiedades ACID en este requerimiento y su complejidad se tuvo que hacer uso de requerimientos anteriormente implementados en la iteración anterior por lo cual para cumplir con la Atomicidad se tuvo que realizar modificaciones que permitieron a todas las operaciones de la transacción llegar a un punto a salvo commit en caso de que todas las operaciones se realizaran o rollback en caso de haber un error en el cualquier punto de la transacción para la coherencia se verifico en el programa que en todo momento el dinero que saliera de la cuenta fuera el mismo que se descontaba del préstamo , en aislamiento se logra aislar los elementos (tuplas) que formaban parte de la transacción de manera que solo se pudieran hacer cambios desde otro dispositivo cuando esta finalizara tomando esta como la única transacción aunque se realizaran otras y finalmente para la durabilidad se tomó en cuenta guardar tanto las tuplas actualizadas en las tablas cuenta y préstamo así como la transacción en operaciones para mantener la información.

RF13: Para cumplir con la propiedad de atomicidad se verifico en la persistencia que, tras una sentencia de SQL ejecutada, se tendrán que ejecutar todas antes de poder hacer commit, de lo contrario se haría rollback. Para cumplir con la propiedad de coherencia en este requerimiento se registró en la tabla de Consigna la fecha en la que se tendrá que hacer la siguiente consigna a una cuenta, y tras una consignación, se determinaría la siguiente fecha añadiendo 15 o 30 días a la fecha. Para cumplir con la propiedad de aislamiento se registró cada tupla de la tabla de manera que solo se pudieran hacer cambios sobre esta desde otro dispositivo cuando esta finalizara la transacción. Y finalmente para garantizar la durabilidad tras cada transacción se guarda la información en la base de datos con un commit y se registran estos cambios en los logs.

RF14: Para cumplir las propiedades ACID en este requerimiento y su complejidad se tuvo que hacer uso de RF10, aunque esto complico el requerimiento debido a que se necesitó de muchas más operaciones para la transacción se logró que la transacción llegara a un commit en caso de que todas las operaciones se realizaran sin problema de manera que todas las operaciones se tuvieran que realizar para llegar a un commit o de lo contrario a un rollback ,en la coherencia se verifico mediante operaciones que tanto las cuentas que el usuario ingresara así como la como la consigna fueran propiedad de este , en aislamiento se logra aislar los elementos (tuplas) que formaban parte de la transacción de manera que solo se pudieran hacer cambios desde otro dispositivo cuando esta finalizara tomando esta como la única transacción, aunque se realizaran otras y finalmente para la durabilidad se cumplió tanto con guardar tanto la cuenta cerrada del cliente corporativo, así como la nueva cuenta y cambios a la consigna esto para mantener la información.

RFC5: Debido a que este es un requerimiento de consulta fue sencillo cumplir la atomicidad debido a que constaba de una única operación, para la coherencia se verificó en múltiples ocasiones que el resultado de la consulta fuera el mismo de los datos obtenidos o guardados anteriormente en él, aislamiento se logró aislar los elementos (tuplas) que formaban parte de la transacción de manera que solo se pudieran hacer cambios desde otro dispositivo cuando esta finalizara tomando esta como la única transacción, aunque se realizaran otras y finalmente para la durabilidad la información se mantenía para futuras investigaciones.

RFC6. Cumple con la propiedad de la atomicidad porque al ser una única consulta, no es necesario tener en cuenta cambios hechos desde la sesión actual a la base de datos. Para cumplir con la coherencia de tuplas consultadas en este requerimiento se verificó el estado de la cuenta loggada actualmente, para garantizarle permiso en la consulta de las tuplas que se le permitiría ver como se describe en el enunciado.

4 Resultados Logrados

En esta iteración se lograron todos los objetivos planteados en el enunciado con respecto a los requerimientos funcionales RF10-RF14, y los requerimientos funcionales de consulta RFC5 y RFC6

5 Resultados no Logrados

Todos los requerimientos para esta interacción se realizaron por lo tanto no hay resultados no logrados.

6 Balance Pruebas

Para esta iteración se hicieron muchas pruebas la mayoría de ella basadas en los casos que se nos recomendaban en el documento del proyecto además de realizar otras pruebas específicas que demostraran los posibles errores que pudiera tener la aplicación además de probar todos los requerimientos de la aplicación en distintas condiciones como

Realizar las pruebas desde distintos computadores, registrando toda información en la base de datos desde la aplicación de java y verificando en la terminal de SQL los resultados obtenidos.

Y aunque al principio las pruebas demostraron varios errores tanto en la aplicación como en los requerimientos se logró que para la entrega el programa funcionara correctamente según lo previsto

7 Reglas de Negocio

Para esta iteración se tienen en cuenta reglas de negocio para el perfecto funcionamiento de la aplicación.

Principalmente, es necesario estar iniciado sesión en algún usuario que sea del tipo del que se requiere para la ejecución de un requerimiento. Por ejemplo, iniciar sesión en un usuario de admin en bancandes para ejecutar sus requerimientos como añadir oficinas o usuarios.

También es necesario que al hacer cambios sobre una cuenta que puede hacer un cliente, se tiene que estar iniciado sesión en el cliente dueño de dicha cuenta.

En el requerimiento de consultar operaciones, para los campos que requieren de valores numéricos, si no se quiere hacer un filtro sobre estos se digita un 0 y entre, y en los campos que son de Sting, para no filtrar no se digita nada y se presiona entre.

Cuando se utilice el requerimiento de buscar prestamos el usuario tiene una gran variedad de filtros para obtener la información esperada sin embargo debe seguir un formato para llenar los filtros dependiendo si es para un caso numérico deberá llenarlo con 3 números los cuales serán “mayor, menor, igual” cada uno separado con “,” y pegados el usuario debe para que la aplicación funcione llenarla con 3 números el filtro mayor y menor es para crear un rango e igual para buscar un numero específico.

Ejemplo

Mayor= 5

Menor= 8

Igual=20

En este caso el usuario visualizara los números entre 5 y 8 sin incluirlos y 20.

Para el caso de cadenas deberá llenar 2 campos los cuales serán “igual, esta” pegado y separado por “,”

Ejemplo

Igual=abierta

Esta=c

En este caso el usuario obtendrá resultados que sean iguales a abierta o palabras que contengan la letra c

Por ultimo si el usuario no quiere usar un filtro deberá dejarlo vacío y darle “ok”