

Lee cuidadosamente el problema que se presenta a continuación y determina cuales son los requerimientos que se solicitan.

Un banco muy importante desea que se desarrolle un sistema que permita gestionar las transacciones que realizan sus clientes. Como en la mayoría de bancos, los clientes pueden tener diferentes cuentas asociadas. En cada cuenta de cliente se guarda un número de cuenta generado por el sistema, fecha de apertura y el saldo en pesos mexicanos.

De los clientes interesa saber su nombre completo, domicilio, fecha de nacimiento y edad. Con el objetivo de facilitar el uso a sus clientes, el banco decidió que las personas interesadas puedan registrarse y actualizar sus datos personales utilizando el sistema directamente sin necesidad de ir a la sucursal. La eliminación de clientes no está considerada como parte del sistema.

En el caso de las cuentas, una vez registradas, no se podrán eliminar ni actualizar sus datos excepto el saldo disponible. Las cuentas podrán cancelarse en cualquier momento si el cliente así lo desea.

Es muy importante que en el sistema se puedan realizar dos tipos de transacciones y a continuación se explican:

1. Un cliente puede realizar una “transferencia” que consiste en retirar dinero desde alguna de sus cuentas y depositar a otra cuenta, que puede ser una cuenta de otro cliente u otra cuenta que le pertenezca a él.
2. Una persona que no sea cliente del banco podría realizar un “retiro sin cuenta” a nombre de un cliente. La persona podrá retirar el monto solicitado en el sistema del banco utilizando un folio de operación y una contraseña generada por el sistema al momento de solicitarla. El folio es un consecutivo generado por el sistema y la contraseña debe de ser un número aleatorio de 8 dígitos.

Finalmente un cliente debe poder consultar su historial de operaciones con posibilidad de buscar por tipo de operación y rango de fechas mediante un selector de fechas.

Consideraciones generales

1. El sistema tiene que ser una aplicación **sencilla** Java SE que utiliza Maven como gestor de dependencias, con interfaces gráficas de usuario y con el motor MySQL.
2. El sistema tiene que permitir realizar las operaciones tanto de los clientes así como de las personas que no sean clientes.
3. Al ser un sistema bancario, la seguridad es un elemento indispensable para garantizar la calidad por lo tanto las contraseñas deben encriptarse al momento de almacenarse.
4. Se debería llevar un registro de todas las operaciones realizadas en las cuentas de los clientes, así como fechas y horas exactas al momento de su ejecución.
5. Para los “retiros sin cuenta” deberá permitirse la operación en un periodo no superior a 10 minutos. Posterior a ese tiempo, el retiro sin cuenta debería pasar a un estado de “no cobrado”.

6. Incluir todas las validaciones correspondientes para cada interfaz gráfica como pueden ser: validación de datos de entrada (por ejemplo formato y campos vacíos), que no se permita guardar información repetida (por ejemplo que existan 2 clientes o cuentas repetidas), validaciones lógicas (por ejemplo un cliente no puede realizar una transferencia mayor al saldo de su cuenta, que un retiro sin cuenta no pueda realizarse dos veces, o que se utilice la misma cuenta como origen y destino).
7. La usabilidad del sistema es de suma importancia, por lo tanto la información debe presentarse de forma legible incluyendo formatos de fechas y de dinero cuando sea conveniente. El sistema debería informar siempre de los resultados de las operaciones realizadas a los usuarios con mensajes claros y concisos.

Entregables del proyecto obligatorios

- Prototipos de interfaz gráfica que incluyan información de muestra.
- Modelo entidad relación y modelo relacional.
- Script de base de datos que incluya además de las tablas, relaciones y datos de prueba, un procedimiento almacenado, un trigger y utilice transacciones donde sea conveniente.
- Proyecto Java SE de Netbeans con Maven como gestor de dependencias.
- Documentación JavaDoc en formato HTML.

NOTA: En caso de no incluir algún entregable el proyecto no se calificará.