

Equipo 1

Cuentas

Historial de transacciones

- Añadido endpoint para obtener todas las transacciones asociadas a una cuenta (tanto salientes como entrantes) ordenadas por fecha:
 - o `/api/transfers/history/{accountId}`
- Se podrían añadir funciones de filtrado o de ordenación para mayor personalización en las peticiones

Update de cuentas

- Añadido endpoint para bloquear/desbloquear una cuenta:
 - o `/api/account/update/isBlocked/{accountId}`
- Añadido endpoint para actualizar el tipo de una cuenta:
 - o `/api/account/update/accountType/{accountId}`
- Añadido endpoint para actualizar la fecha de expiración de una cuenta:
 - o `/api/account/update/expirationDate/{accountId}`

*ExpirationDate una llamada al endpoint actualiza la fecha de expiración 5 años solo si faltan menos de 3 meses para que se cumpla la fecha de expiración anterior.

Checkeo de cuentas bloqueadas o en deuda al borrarlas

- Añadidos checkers al borrar una cuenta en los tres endpoints dedicados a ello:
 - o `/api/account/delete/{accountId}`
 - o `/api/account/delete/customer/{customerId}`
 - o `/api/account/delete`
- Estos checkers consisten en:
 - o (Por cada cuenta)
 - o Comprobar si está en deuda, crear un mensaje de “cuenta en deuda” y guardar el estado en un booleano
 - o Comprobar si está bloqueada y, en función del booleano de deuda, crear un mensaje de “cuenta bloqueada” o modificar el ya existente para reflejar ambos estados

- Devolver el mensaje sin eliminar la cuenta y/o lanzar una excepción
- En el catch, imprimir el/cada mensaje de error de eliminación

Usuarios

Diferenciación entre clientes y administradores

- Añadida la columna “role” a la tabla Customer en la base de datos, así como el atributo en la clase Customer.java con los siguientes valores:
 - ADMIN
 - USER
 - ...
- Añadido el rol como propiedad al crear el token de inicio de sesión.
- Para poder comprobar el rol a la hora de utilizar endpoints, añadida una función que obtiene el rol a partir de un token.
- Utilizado filtrado con Spring Security para bloquear endpoints a usuarios en función de su rol.

Equipo 2

Withdraw

- Añadida la entidad Withdraw a la base de datos, con los siguientes campos:
 - withdrawId
 - Amount
 - WithdrawDate
 - CardId
- Los withdraws estan asociados a una tarjeta, que a su vez está asociada a una cuenta.
- Añadidos los siguientes endpoints:
- Realizar un withdraw:
 - `/api/withdraw`
- Obtener todos los withdraws del usuario logueado:
 - `/api/withdraws`
- Obtener todos los withdraws de una tarjeta:
 - `/api/withdraws/card/{cardId}`
- Obtener todos los withdraws asociados a una cuenta:
 - `/api/withdraws/account/{accountId}`
- Eliminar los withdraws
 - `/api/withdraws/delete/card/{cardId}`

Tarjetas

Bloquear y desbloquear tarjetas

- Añadido el parámetro isBlocked, además cuando una tarjeta está bloqueada no se permite realizar withdraws con ella.
- Añadido endpoint para modificar el estado de tarjeta bloqueada

Creación de tarjetas

- Ajustado la creación de tarjetas para que ahora se tome el tipo de tarjeta del cardRequest o sino lo genere aleatorio.
- Limitada la creación de tarjetas, ahora únicamente se puede generar una tarjeta si la cuenta asociada a la tarjeta es la del usuario logueado.

Update de tarjetas, establecimiento límites diarios/mensuales

- Añadido los parámetros DailyLimit y MonthlyLimit, creados dos endpoints para actualizar los valores de estos límites, las tarjetas creadas tienen un límite diario y mensual por defecto.
- `/api/card/update/dailyLimit/{cardId}`
- `/api/card/update/monthlyLimit/{cardId}`

Cientes

Eliminar un customer

- Se ha añadido un nuevo endpoint, que elimina un customer pasando su ID. Además, elimina sus cuentas asociadas. De estas cuentas, también elimina todas las tarjetas asociadas a cada cuenta, además de todas las transferencias y las retiradas de dinero.
- `/public/customer/delete/{customerId}`

Ficheros en los que se han hecho cambios:

- BankAPI
 - CustomerController
 - AccountController
 - WithdrawController
 - CustomerService
 - CustomerRepository
 - CardController
-
- Se ha añadido un fichero de pruebas; deleteCustomer.feature, que hace la prueba de eliminar el customer por el id. Para ello se han hecho cambios en los ficheros GenericSteps, AuthenticationSteps y BankService.