

Trabajo Final. Tema 1

Objetivo

Extender el sistema bancario para incluir la gestión de transacciones bancarias utilizando Spring Boot.

Se deberá aclarar toda decisión de diseño que se tome ya sea explicitando en un documento, o en comentarios del código. Si existen validaciones de negocio que se crean pertinentes y que no fueron explicitadas, pueden agregarlas, siempre y cuando lo documenten.

Cabe aclarar que **TODOS** deberán tener corriendo el endpoint de cliente y crear cuenta (que fue realizado en clase durante el cuatrimestre).

Descripción

Crear Endpoint para Realizar Transferencias:

- Implementar un endpoint para realizar transferencias entre cuentas del mismo banco o no.
- El endpoint debe recibir como parámetros el número de cuenta origen, número de cuenta destino, moneda y el monto a transferir.
- Verificar que ambas cuentas existan en el banco, y si la cuenta destino no existe se debe invocar un servicio que centraliza las transferencias entre distintos bancos (Banelco). Este servicio **simulará** la invocación a un servicio externo al banco que puede dar cómo resultado la transferencia exitosa o no.
- Se debe verificar que la moneda de ambas cuentas sea la misma, y en caso de ser transferencia en pesos, el banco cobrará un cargo del 2% de la transferencia si supera \$1000000 (o caso de ser dólares cobra un 0.5% si son más de U\$S5000).
- Se deberá verificar que la cuenta origen tenga fondos suficientes para realizar la transferencia y que la moneda de ambas cuentas sea la misma.
- Actualizar el saldo de ambas cuentas en consecuencia (o la cuenta de origen si fue transferencia a cuenta de distinto banco)

Especificación del endpoint

- URL: `/api/transfer`
- Método HTTP: POST
- Input (JSON)

```
{
  "cuentaOrigen": long,
  "cuentaDestino": long,
  "monto": double,
  "moneda": "string",
}
```

- Output (JSON)

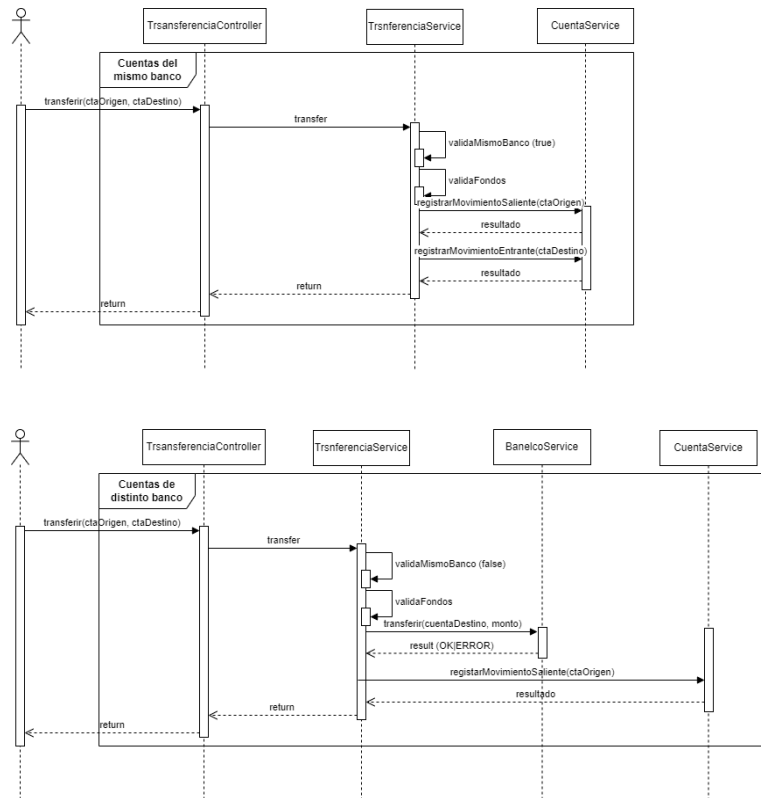
```
{  
  "estado": "string",  
  "mensaje": "string"  
}
```

Ejemplos input y output

```
{  
  "cuentaOrigen": 12345678,  
  "cuentaDestino": 87654321,  
  "monto": 5678.91,  
  "moneda": "dolares"  
}
```

```
{  
  "estado": "EXITOSA|FALLIDA",  
  "mensaje": "Transferencia exitosa"  
}
```

Secuencia de invocaciones para transferencia (mismo y distinto banco)



Crear un endpoint para consultar historial de transacciones:

- Implementar un endpoint para consultar el historial de transacciones de una cuenta.
- El endpoint debe recibir como parámetro el número de cuenta y devolver una lista de transacciones, incluyendo fecha, tipo (crédito, débito, transferencia entrante o transferencia saliente), y monto.
- En caso que durante la cursada no hubieran implementado agregar transacciones (movimientos) a una cuenta, deberá responder con la lista de transferencias entrantes y salientes.

Especificación del endpoint

- URL: `/api/cuenta/{cuentaId}/transacciones`
- Método HTTP: **GET**
- Input (Path Variable)
 cuentaId: long
- Output (JSON)

```

{
  "numeroCuenta": "string",
  "transacciones": [
    {
      "fecha": "string",
      "tipo": "string",
      "descripcionBreve": "string",
      "monto": double
    }
  ]
}
  
```

```
]
}
```

Ejemplo output

```
{
  "numeroCuenta": 12345678,
  "transacciones": [
    {
      "fecha": "2024-07-01",
      "tipo": "CREDITO",
      "descripcionBreve": "Transferencia entrante",
      "monto": 1500.00
    },
    {
      "fecha": "2024-07-03",
      "tipo": "DEBITO",
      "descripcionBreve": "Compra Cooperativa Obrera",
      "monto": 500.00
    }
  ]
}
```