

Portfolio Bancario 2: Transferencias y Reportes

Ahora que el banco posee la funcionalidad de tener portfolios, quiere empezar a ofrecer más opciones a sus clientes. Para ello decidió agregar un nuevo tipo de transacción: la transferencia entre cuentas. Además, sumará la posibilidad de sacar reportes sobre las operaciones realizadas.

Una **transferencia** es una transacción que tiene "dos patas". La pata de la extracción, de donde se saca la plata, y la pata del depósito a donde se deposita la plata. Una transferencia se realiza entonces entre dos cuentas y por un valor. Algo importante a tener en cuenta es que se tiene que poder navegar desde la pata de la extracción a la transferencia y desde esta última a la pata del depósito y viceversa, aunque evitando tener redundancia de información. Es decir, el valor de la transacción debe estar en un solo lugar y no repetido en cada pata.

Respecto de los **reportes**, se deben poder ejecutar sobre una *ReceptiveAccount* o un *Portfolio* de manera indistinta. Se espera tener dos reportes inicialmente:

1. El resumen de cuentas (Account Summary).
2. El neto de transferencias (Transfer Net).

El **resumen de cuenta** debe generar una línea por cada transacción realizada en una cuenta con el siguiente formato:

Depósito por 100.
Extracción por 50.
Salida por transferencia de 20.
Entrada por transferencia de 30.
Balance = 60

Este sería el resumen de cuenta esperado de una cuenta a la cual se le realizó un depósito por 100, una extracción por 50, se le sacó 20 por una transferencia y recibió 30 por otra transferencia.

El **reporte de neto de transferencias** debe devolver el resultado de sumar todos los depósitos por transferencias y restarle todas las extracciones por transferencias. Para el ejemplo anterior, el neto de transferencias sería 10.

El banco prevé agregar muchos reportes nuevos en un tiempo inmediato, por lo tanto el modelo final para sacar reportes debe cumplir con los siguientes requerimientos de extensibilidad:

1. Al crear nuevos reportes no se tiene que volver a modificar la jerarquía de cuentas.
2. Al crear nuevos reportes no se tiene que volver a modificar la jerarquía de transacciones.
3. Crear nuevos reportes debe implicar crear clases nuevas únicamente y no modificar ninguna existente.

Resolver este ejercicio por medio de TDD.

Bonus track

El CEO del banco nos premiará si logramos tener 2 nuevos reportes de resumen de cuenta para portfolios. El primero deberá mostrar la estructura de árbol completa del portfolio. El segundo, un reporte de cuenta especial que muestre las transacciones indentadas de acuerdo a la profundidad de cada cuenta del portfolio.

Dado un portfolio como se muestra a continuación:

```
johnsAccount := ReceptiveAccount named: 'Cuenta de Juan'.
angiesAccount := ReceptiveAccount named: 'Cuenta de Angeles'.
childrenPortfolio := Portfolio named: 'Portfolio de hijos' with:
johnsAccount with: angiesAccount.
myAccount := ReceptiveAccount named: 'Cuenta mia'.
familyPortfolio := Portfolio named: 'Portfolio de la familia' with:
myAccount with: childrenPortfolio.
```

El reporte de la estructura de árbol de dicho portfolio debería ser:

```
Portfolio de la familia
  Cuenta Mia
  Portfolio de hijos
    Cuenta de Juan
    Cuenta de Angeles
```

Se espera que el resumen de cuenta especial se muestre de la siguiente manera:

```
Portfolio de la familia
  Cuenta Mia
    Depósito por xxx
    Extracción por yyy
    Balance = bbb
  Portfolio de hijos
    Cuenta de Juan
      Depósito por zzz
      Extracción por nnn
      Balance = bbb
    Cuenta de Angeles
      Salida por extracción de qqq
      Balance = bbb
    Balance = bbb
  Balance = bbb
```

Al igual que los reportes anteriores, la diseño final para resolverlos debe permitir agregar reportes sobre portfolios sin tener que modificar nada.