

El sistema computacional de una entidad bancaria chilena permite realizar transferencias bancarias solo entre cuentas corrientes originadas en Chile. Para ello, la aplicación en cuestión define la clase “CuentaCorriente”, la cual define el atributo “pesos”, el cual se puede leer y escribir libremente. Toda transferencia se hace a través de una instancia de la clase “Banco”, la cual define el método *transferir(fuente, destino, monto)*, donde *fuente* y *destino* son instancias de “CuentaCorriente” y *monto* es un entero que representa el monto de la transferencia en pesos. Este método permite traspasar el monto especificado como parámetro desde la cuenta *fuente* hacia la cuenta *destino*.

Usted ha sido contratado para desarrollar la versión 2.0 de este sistema computacional, el cual entre sus múltiples funcionalidades debe permitir transferir dinero desde y hacia cuentas extranjeras que operen con dólares americanos. Los sistemas con los que usted debe interoperar son afortunadamente muy similares al sistema chileno, pues estos definen una clase “ForeignAccount” que también define un único atributo que se puede leer y escribir libremente, solo que en este caso el atributo se llama “dollars” y representa el monto de la cuenta en dólares americanos.

Durante la fase de análisis de su solución, su equipo de desarrollo se da cuenta de un detalle que puede ser importante: los montos en pesos de “CuentaCorriente” se guardan como enteros, mientras que los montos en dólares de “ForeignAccount” se guardan como números decimales con dos dígitos de exactitud. Sin embargo, al transformar pesos a dólares, suele ocurrir que los montos resultantes definen 3 dígitos decimales o más. Después de consultar con el cliente, se optó por programar el sistema para que cada vez que ocurra un caso similar durante una transferencia, el monto a setear en la cuenta extranjera se debe truncar a solo 2 dígitos decimales, y la diferencia, por pequeña que sea, se debe depositar en una cuenta del banco.

Dado lo anterior, utilizando el lenguaje Ruby, usted debe:

- Escribir la clase “Banco”.
- Escribir un adaptador que permita que el método *transferir(fuente, destino, monto)* funcione con instancias de “ForeignAccount” y que incorpore la funcionalidad de redondeo aquí descrita.
- Escribir un breve script que emule una operación de transferencia de 50.000 pesos desde una “CuentaCorriente” con 100.000 pesos hacia una “ForeignAccount” con 1.000 dólares, a una tasa de cambio de 715,53 pesos/dólar.

Puede asumir que “Banco” guarda una referencia a la cuenta corriente del banco y que las clases “CuentaCorriente” y “ForeignAccount” vienen dadas (no tiene que implementarlas). Se le recuerda que Ruby define el método *floor(n)*, que permite truncar números decimales a ‘n’ cifras significativas.