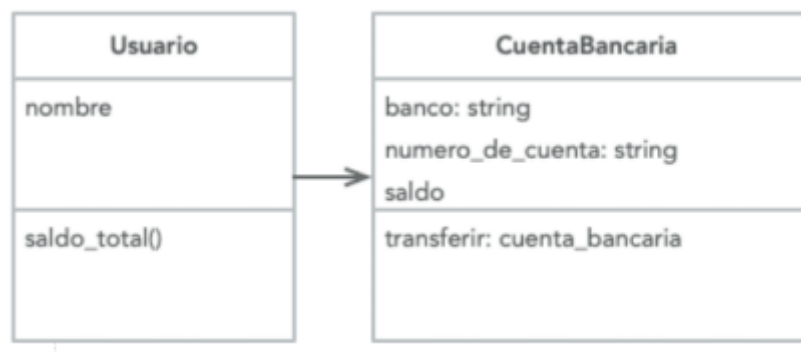


Desafío - Cuentas bancarias y baraja de cartas

- Para realizar este desafío debes haber estudiado previamente todo el material disponible correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el .zip en el LMS.
- Puntaje total: **10 puntos**.
- Desarrollo desafío:
 - El desafío se debe desarrollar de manera Individual/Grupal.

Ejercicio 1

Dado el siguiente Diagrama UML, se pide en un proyecto desde cero:



1. Crear las clases Usuario y CuentaBancaria. **(1 Punto)**
2. Crear el constructor de la clase CuentaBancaria que reciba el nombre del banco, el número de cuenta y el saldo (el saldo por defecto será cero). **(1 Punto)**
3. Crear el método `transferir` en la clase Cuenta que reciba un monto y otra cuenta. Este método restará del saldo actual el monto y aumentará el saldo de la otra cuenta en el mismo monto. **(1 Punto)**



Es necesario que exista el `attr_writer` o `attr_accessor` del saldo para acceder al saldo de la otra cuenta.

4. Probar creando dos cuentas cada una con un saldo de 5000 y transferir el total de una cuenta a la otra.
5. Crear el constructor de Usuario que reciba el nombre del usuario y un arreglo con al menos 1 cuenta bancaria. **(1 Punto)**

6. Crear el método saldo total que devuelva la suma de todos los saldos del usuario. (1 Punto)

Ejercicio 2

Baraja de cartas

- Crear la clase carta con los atributos `numero` y `pinta`. (1 Punto)
- Agregar los getters y setters a ambos atributos.
- Crear el constructor de la clase carta que le permita recibir un número del 1 al 13 y la pinta que está indicada por una sola letra. Puede ser Corazón: 'C', Diamante: 'D', Espada: 'E' o Trébol: 'T'. (1 Punto)
- Utilizar el ejercicio Carta creado en la primera parte y dejarlo en el archivo carta.rb.
- Crear la clase baraja en el archivo barajas.rb con el atributo cartas, el cual será un arreglo. (1 Punto)
- Al crear una baraja (constructor) se deben generar todas las combinaciones de números y pinta y guardarse dentro del arreglo cartas.
- Crear el método barajar que debe desordenar el arreglo de cartas Tip: Utilizar `.shuffle`.
- Crear el método sacar que permita retirar la primera carta de la baraja Tip: Utilizar `.pop`. (1 Punto)
- Crear el método repartir mano que devolverá un arreglo con las primeras 5 cartas de la baraja. (1 Punto)