



Desafío - Cuenta bancaria

- Para realizar este desafío debes haber estudiado previamente todo el material disponibilizado en la sesión online correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta y sube el `.zip` con la siguiente estructura

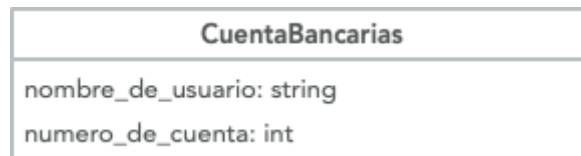
```
|— carta.rb
|— cuenta_bancaria.rb````
```

Ejercicio 1

- Crear la clase carta con los atributos `numero` y `pinta` (o color) (**1punto**)
- Agregar los getters y setters a ambos atributos. (**1punto**)
- Crear el constructor de la clase carta que le permita recibir un numero 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' (**2 puntos**)
 - Tip 1: Para escoger un número al azar ocupar `Random.rand(rango_inferior, rango_superior)`
 - Tip 2: Agregar las pintas posibles en un arreglo y ocupar el método `.sample`
- Probar la clase creando un arreglo con 5 cartas. (**1punto**)

Ejercicio 2

Dado el siguiente código UML



1. Crear la clase con el nombre CuentaBancaria (**1punto**)
 - Los métodos getter y setter para el atributo nombre_de_usuario (**1punto**)
 - Crear el constructor que reciba el nombre_usuario y numero_de_cuenta y lo asigne a los atributos(**1punto**)
 - Levantar una excepción del tipo RangeError si el atributo numero_de_cuenta tiene un número de dígitos distinto a 8 (se puede ocupar el método .digits para obtener los dígitos y .count para contarlos) (**1punto**)
 - Agregar un tercer parámetro opcional al constructor que permita establecer si una cuenta es VIP (1pto), este valor puede ser 1 o 0. Por defecto será 0.
 - Crear un método llamado numero_de_cuenta que devuelva con el número de cuenta con un prefijo '1-' si es vip y '0-' si no lo es. Ejemplo: si la cuenta es VIP y el número 00112233, el método debería devolver '1-00112233' (**1punto**)