

DESARROLLO DE SOFTWARE

EJERCICIO N° 8

C#

1. Realizar una bifurcación (*fork*) del repositorio:
<https://github.com/ing-software-frt-utn/dsw2025ej8>
2. Crear una rama de larga duración development
3. Clonar el repositorio bifurcado y trabajar sobre la rama development
4. Refactorizar el código aplicando herencia según el caso
5. Reemplazar los métodos *getters* y *setters*, y campos por propiedades, tener en cuenta la accesibilidad en cada caso
6. Respetar que al crear una cuenta bancaria se reciba el número y el saldo en el constructor
7. La tasa de interés se debe indicar al inicializar la instancia de cuenta, pero no mediante el constructor
8. El límite de descubierto se debe indicar al inicializar la instancia de cuenta, pero no mediante el constructor
9. Agregar las siguientes reglas:
 - a. El monto recibido por cualquier operación no puede ser menor o igual a 0, de lo contrario generar una excepción del tipo `MontoNoValido`
 - b. Cualquier operación se debe realizar si la cuenta está activa, en cualquier otro caso generar una excepción del tipo `CuentaNoActiva`
 - c. Se debe contar con saldo para realizar un retiro, caso contrario debe generar una excepción `SaldoInsuficiente` y la cuenta debe quedar suspendida. Tener en cuenta el límite de descubierto si corresponde
10. Instanciar 4 cuentas (dos de cada tipo) y realizar diferentes operaciones que permitan comprobar todas las funciones posibles.
11. Recorrer las 4 cuentas creadas y mostrar por consola un resumen de cada una, que incluya número, tipo y saldo (utilizar una clase anónima)

Consideraciones:

- Las excepciones deben incluir los siguiente mensajes:
 - `MontoNoValido` -> El monto ingresado no es válido para la operación solicitada
 - `CuentaNoActiva` -> No se puede operar con la cuenta {estado} (reemplazar por el estado en el que se encuentra)
 - `SaldoInsuficiente` -> La cuenta no cuenta con saldo para la operación solicitada. Fue suspendida.
- La aplicación no debe interrumpir su funcionamiento si se produce una excepción.