

# Proyecto Final: Gestión de Cuentas Bancarias con Herencia

## Descripción del Proyecto

Debes desarrollar un sistema para gestionar cuentas bancarias utilizando herencia para diferentes tipos de cuentas. Los usuarios podrán agregar clientes, asignar cuentas bancarias a los clientes y realizar transferencias entre cuentas previamente registradas. ***Usted es libre de agregar todos los detalles que desee para el correcto funcionamiento de su proyecto.***

## Requisitos del Proyecto

### 1. Clase CuentaBancaria:

- Atributos privados o protected: NumeroCuenta, Titular, Saldo.
  - ***Todos los atributos privados van encapsulados (setters y getters)***
- Métodos públicos:
  - Constructor para inicializar una cuenta con el número de cuenta, el titular y el saldo.
  - Métodos para depositar y retirar dinero.
  - Método para transferir dinero a otra cuenta.
  - Método ToString para representar la cuenta como una cadena.
    - ***(recordar si estos métodos se van a heredar para sobreescribir o darle otro comportamiento es necesario ponerle public virtual)***

### 2. Clases Derivadas:

- **Clase CuentaAhorro:** Hereda de CuentaBancaria, añade un atributo para la tasa de interés.
- **Clase CuentaCorriente:** Hereda de CuentaBancaria, añade un atributo para el límite de sobregiro.

### 3. Clase Cliente:

- Atributos privados: Nombre, Cuentas (lista de CuentaBancaria).
  - ***Todos los atributos privados van encapsulados (setters y getters)***

- Métodos públicos:
  - Constructor para inicializar el nombre del cliente.
  - Métodos para agregar una cuenta bancaria y mostrar información del cliente.
  - Método ToString para representar al cliente como una cadena.

#### 4. Clase GestorCuentas:

- Atributos privados: Clientes (lista de Cliente).
  - *Todos los atributos privados van encapsulados (setters y getters)*
- Métodos públicos:
  - Métodos para agregar clientes y cuentas bancarias.
  - Métodos para realizar transferencias entre cuentas.
  - Método MostrarClientes para mostrar la información de todos los clientes.

#### 5. Clase Program o Tester:

- Punto de entrada del programa.
- Interacción con el usuario a través de una interfaz de texto o menu.
- Funcionalidades para agregar clientes y cuentas, realizar transferencias y mostrar la información.