

3) Diseñe una clase de C++

Diseñe desde cero una clase llamada CuentaBancaria que contenga:

- I. Un atributo privado saldo (tipo float)
- II. Un atributo numeroCuenta (tipo std::string)
- III. Un constructor que reciba ambos atributos
- IV. Método depositar(float) que sume al saldo
- V. Método retirar(float) que reste del saldo si hay fondos suficientes
- VI. Método bloquearRetiros(). ¿Se requiere algo adicional en la clase?
- VII. Método float obtenerSaldo()

VIII. BONO: Documente los métodos usando estilo Doxygen

La parte de diseño incluye únicamente generar el contenido del archivo .h con la estructura de los atributos y funciones.

Adicional al diseño, debe escribir a mano la implementación de alguno de los métodos de libre elección, el resto se implementan en el siguiente punto.

4) Implemente en un archivo .cpp, los métodos descritos en el archivo .h y desarrolle en otro archivo .cpp una aplicación que use la clase CuentaBancaria para registrar depósitos y retiros de usuarios. Para la aplicación tener en cuenta los siguientes criterios y estructura sugerida del programa.

- I. Use std::vector para almacenar varias cuentas
- II. Simule lectura de datos y registro depósitos, retiros y bloqueos
- III. Genere un reporte en formato .txt que muestre el saldo de cada usuario

```
#include "contador_energia.h"

int main() {
    std::vector<ContadorEnergia> medidores;

    // Agregue 3 objetos al vector con valores iniciales

    // Simule registro de consumo para cada medidor

    // Genere archivo de salida "reporte.txt" con ID y consumo

    return 0;
}
```

b

NOTA: La imagen es un ejemplo de un flujo similar al que se debe implementar con la clase CuentaBancaria. Organice el proyecto siguiendo las buenas prácticas en cuanto al manejo de folders y archivos auxiliares como CMake y Doxygen. Subir la solución completa a un repositorio de github publico cuya dirección o enlace indicado en la primera hoja