

Gestión Bancaria

PARTE 1:

1. Define una clase llamada **Cliente** con los siguientes atributos:
 - **__nombre**: el nombre del cliente.
 - **__dni**: el número de documento del cliente.
2. Crea una clase llamada **Cuenta** con los siguientes atributos:
 - **numero_cuenta**: un número único para identificar la cuenta.
 - **saldo**: el saldo actual en la cuenta.
 - **propietario**: una instancia de la clase **Cliente** que representa al dueño de la cuenta.
3. Implementa un método en la clase **Cuenta** llamado **mostrar_informacion** que imprima el número de cuenta, el saldo y la información del propietario (nombre y dni).
4. Define una clase llamada **Banco** que tenga como atributos:
 - **nombre**: el nombre del banco.
 - **cuentas**: una lista para almacenar instancias de la clase **Cuenta**.
5. Agrega un método en la clase **Banco** llamado **agregar_cuenta** que reciba una instancia de **Cuenta** y la añada a la lista de cuentas del banco.
6. Crea al menos dos instancias de la clase **Cliente**.
7. Crea al menos dos instancias de la clase **Cuenta**, asociando cada cuenta con un cliente.
8. Crea una instancia de la clase **Banco** y agrega las cuentas creadas a través del método **agregar_cuenta**.
9. Utiliza el método **mostrar_informacion** de las cuentas para imprimir los detalles de cada cuenta en el banco.

PARTE 2:

1. Modifica la clase **Cuenta** para que:
 - El saldo no pueda ser negativo. Si se intenta realizar una operación que cause un saldo negativo, lanza una excepción **SaldoNegativoError**.
 - El número de cuenta sea único. Si se intenta crear una nueva cuenta con un número que ya existe, lanza una excepción **NumeroCuentaExistenteError**.
2. Define las excepciones **SaldoNegativoError** y **NumeroCuentaExistenteError** como subclases de la excepción base **Exception**. Personaliza los mensajes de error.
3. Actualiza el método **mostrar_informacion** de la clase **Cuenta** para incluir el manejo de excepciones y mostrar mensajes informativos en caso de error.
4. Añade control de errores en el método **agregar_cuenta** de la clase **Banco**. Si se intenta agregar una cuenta con un número que ya existe, lanza la excepción **NumeroCuentaExistenteError**.
5. En el código principal:
 - Intenta realizar operaciones que generen **SaldoNegativoError** en al menos una cuenta.
 - Intenta crear cuentas con números que ya existen para generar **NumeroCuentaExistenteError**.
6. Asegúrate de que el código maneje estas excepciones de manera adecuada.