

---

# Sistema Bancario

## Descripción del Escenario:

Un cliente del banco puede tener múltiples cuentas. Cada cliente tiene un nombre, dirección y una lista de cuentas bancarias. Pueden agregar o eliminar cuentas y consultar el saldo total de sus cuentas. De las cuentas bancarias se tiene la información de número de cuenta, titular y saldo. Cada cuenta tiene un número único y un titular. También tiene métodos para depositar, retirar y consultar el saldo. Tenemos dos tipos de cuentas bancarias :

- Cuenta de ahorro : Las cuentas de ahorro pagan intereses a una tasa específica. Deberás implementar un método para calcular y agregar intereses al saldo.
- Cuenta corriente : Las cuentas corrientes pueden tener un límite de sobregiro y deben manejar retiros que excedan el saldo disponible.

El sistema principal es el Banco, el cual contiene una lista de clientes. Los clientes pueden registrarse en el banco. El banco ofrece los servicios como son abrir cuentas y gestionar a los clientes.

El sistema debe tener una funcionalidad que permite exportar en un csv las cuentas bancarias del banco. En las columnas del csv se debe ver :

- Número único del titular
- Nombre de titular
- Saldo
- Tipo

El listado del archivo tiene que estar ordenado por número único del titular y por saldo.

**Recordar que un usuario (Un número único) puede tener varias cuentas.**

---

## Requerimientos:

- A. Investigar cómo funcionan las cuentas de ahorro y cuenta corriente, en base a esto realizar la implementación de las funcionalidades del sistema.
- B. Aplicar los principios de POO en base a lo investigado.
- C. Realizar una descripción de lo investigado.
- D. En base a los conocimientos de POO, la investigación realizada, identificar las entidades del sistema y representarlo en un diagrama de clases.
- E. Implementarlo en java usando como gestor del proyecto a MAVEN.

## Plus:

- Crear una base de datos con las tablas de las entidades detectadas, relacionarlas e insertar los datos que se exportan en el csv.

## Desarrollo

- A) Investigar cómo funcionan las cuentas de ahorro y cuenta corriente, en base a esto realizar la implementación de las funcionalidades del sistema.

---

## Cuentas de ahorro y Cuentas corrientes

### Cuenta de Ahorro:

La cuenta de ahorro es una herramienta financiera diseñada para que los clientes depositen y acumulen fondos a lo largo del tiempo. Estas cuentas ofrecen un interés sobre los saldos, lo que significa que los clientes pueden ganar dinero adicional mientras mantienen su dinero en la cuenta. La tasa de interés suele ser más alta que la de una cuenta corriente, aunque generalmente hay restricciones en la cantidad de transacciones que se pueden realizar.

#### Características Principales:

- Pagan una tasa de interés sobre el saldo, usualmente baja pero garantizada.
- Permiten un número ilimitado de depósitos.
- Limitan la cantidad de retiros, usualmente 6 al mes.
- No cobran comisiones si se cumple un saldo mínimo mensual.
- Se calculan intereses mensualmente sobre el saldo promedio y se abonan a la cuenta.

### Cuenta Corriente:

La cuenta corriente es una cuenta de depósito que permite a los clientes realizar diversas transacciones financieras, como depósitos, retiros, cheques y transferencias. A diferencia de las cuentas de ahorro, las cuentas corrientes generalmente no generan intereses. Una característica distintiva de las cuentas corrientes es la posibilidad de tener un límite de sobregiro, lo que permite a los titulares de cuentas realizar transacciones incluso si no tienen fondos suficientes, hasta el límite acordado.

#### Características Principales:

- No pagan intereses.
- Permiten un número ilimitado de depósitos y retiros.
- Cobran comisiones por mantenimiento y otras operaciones.
- Pueden tener sobregiro bancario pre-acordado, con un límite y cobro de intereses.

**Tomando en cuenta estos puntos, se pueden implementar funcionalidades como:**

### En CuentaAhorro:

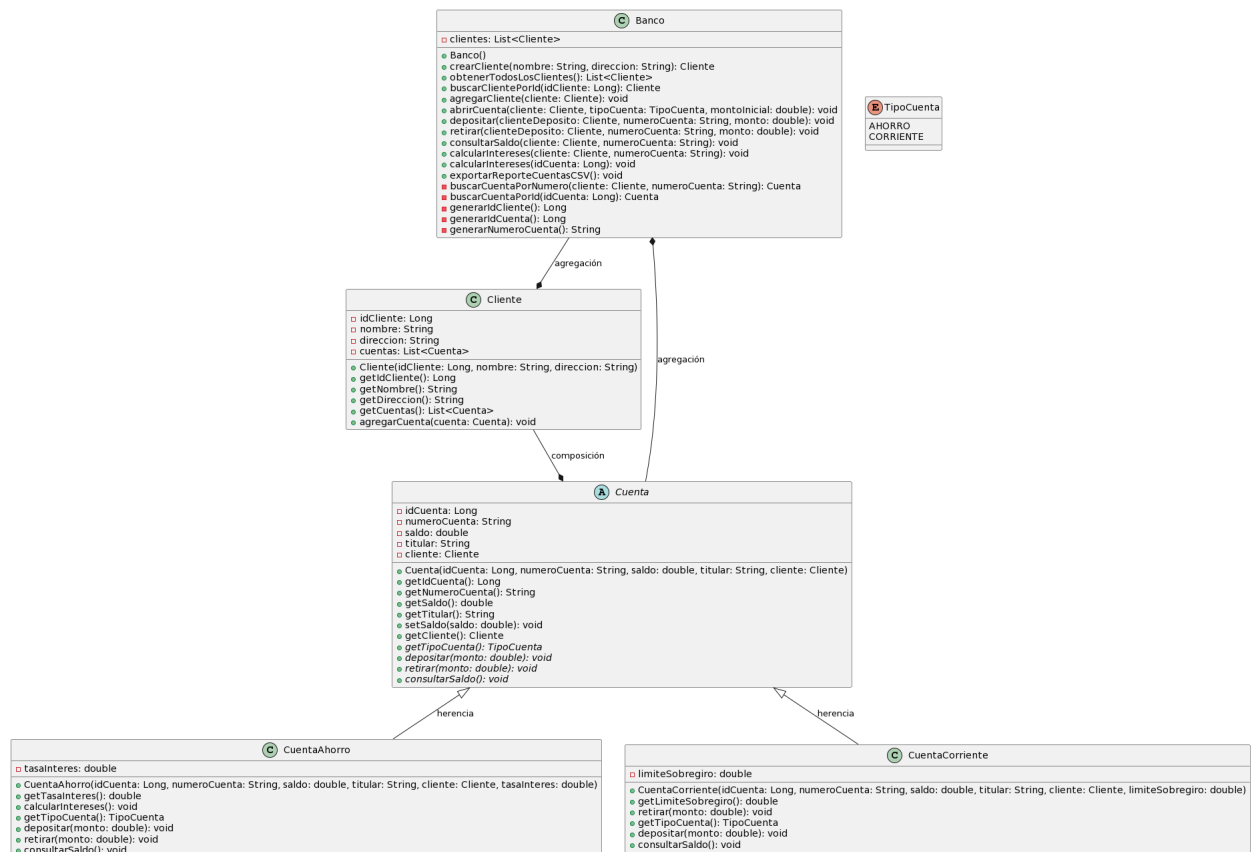
- Implementar un método `calcularInteres()` que calcule la tasa a aplicar al momento de hacer el deposito, sobre el monto depositado.
- Validar que los retiros no excedan el límite.

### En CuentaCorriente:

- método `retirar()` considerando el límite de sobregiro.
- métodos para cobrar comisiones por operaciones.

## Descripción del esquema UML

En el caso que no se pueda visualizad adecuadamente el esquema UML, se encuentra dentro de la carpeta Documents, del proyecto que se subio al repositorio, o puede acceder directamente desde [este link](#)



---

### **Relaciones:**

- **Cliente-Cuenta:**
  - Relación de composición: Un cliente tiene una lista de cuentas. Si el cliente se elimina, también se eliminan sus cuentas asociadas.
- **CuentaAhorro-CuentaCorriente:**
  - Relación de herencia: CuentaAhorro y CuentaCorriente heredan de la clase base Cuenta.
- **Banco-Cliente:**
  - Agregación: El banco tiene una lista de clientes. Si el banco se cierra, los clientes todavía existen independientemente.
- **Cliente-Cuenta (en Banco):**
  - Agregación: Un cliente tiene una lista de cuentas en el contexto del banco. Si el cliente se elimina del banco, sus cuentas siguen existiendo independientemente.

*Nota\*: Para el modelado de los datos y el diagrama UML se utilizó [PlantUML](#), es libre y gratuito. Los archivos generados con este software los puede encontrar [aquí](#)*