***Diagrama de Classes***

Aqui estão as **principais práticas recomendadas**:

### **1. Atributos sempre privados (-)**

* **Por quê?** Garante que os dados só possam ser acessados e modificados por métodos da própria classe.
* **Boa prática:** Use **getters e setters** públicos para expor o acesso controlado aos atributos.

- saldo: double

+ getSaldo(): double

+ setSaldo(valor: double): void

**2. Métodos públicos apenas quando necessário**

* **Evite expor métodos que são usados apenas internamente.**
* Se um método serve de apoio a outros métodos e **não deve ser usado fora da classe**, torne-o privado (-).

**3. Use visibilidade protegida (#) apenas com herança**

* **Quando usar:** Se você sabe que a classe terá **subclasses** que precisarão acessar diretamente os membros.
* **Evite:** Usar # sem ter hierarquia definida.

**4. Prefira o princípio do "menor privilégio"**

* Dê o **nível mínimo de acesso necessário** para cada membro.  
  + Se só a classe precisa acessar → -
  + Se precisa ser acessado por subclasses → #
  + Se qualquer um pode usar → +

**5. Use abstração: exponha comportamentos, não dados**

* Em vez de deixar atributos públicos, **ofereça métodos que expressem intenções**.

Ruim:

+ saldo: double

Melhor:

- saldo: double

+ consultarSaldo(): double

+ sacar(valor: double): boolean

### **6. Separe responsabilidades**

* Classes devem ser **coesas**. Não exponha métodos ou atributos que não façam parte da responsabilidade principal da classe.

### **7. Documente e justifique visibilidades incomuns**

* Se usar algo fora do padrão (como ~ ou #), deixe claro **por quê** — por exemplo, se estiver modelando pacotes Java ou prevendo extensibilidade via herança.