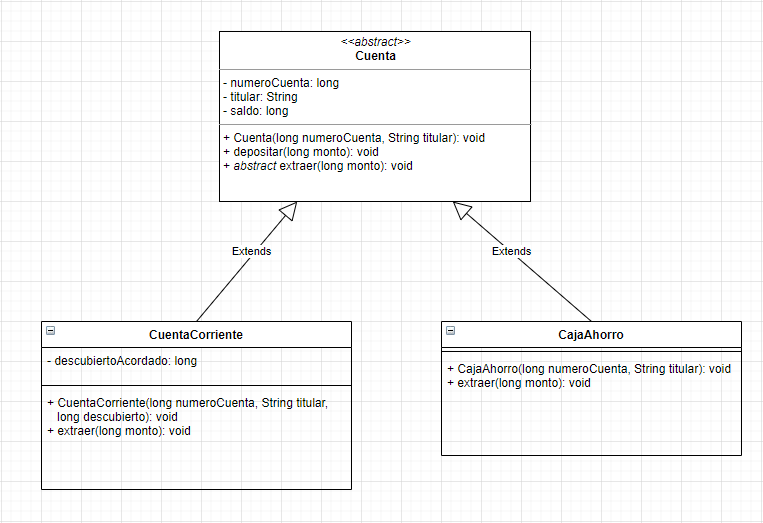
# ATIX LABS \_

# EVALUACIÓN Developer SR \_ Semi SR

1. Resolución de problemas de programación y diseño

**A) Tome el código que sigue – que representa una cuenta bancaria – y responda las siguientes preguntas:**

* ¿Le parece que está bien diseñada la clase? Si no lo está, pero funciona, ¿la cambiaría o agregaría clases nuevas?
* No me parece que esté bien diseñada la clase, aunque a simple vista pareciera compilar y funcionar. Sí, le agregaría una jerarquía de clases polimórficas, poniendo como clase base abstracta a ***Cuenta***y que las demás clases concretas hereden de ésta y extiendan su comportamiento específico. En este caso sería **CuentaCorriente** y **CajaAhorro**.
* Si cambiara el diseño, exponga todo lo que cambiaría con un diagrama de clases de UML. Si algún método debe cambiarlo mucho, escríbalo.
* [Link a diagrama UML](https://drive.google.com/file/d/1gExV8CIjVyOmnmcpyc0bsr_dTq2UmWpo/view?usp=sharing)



* ¿Es seguro hacer estos cambios? ¿Por qué? ¿Qué precauciones tomaría?
* Es seguro si se detectan todos los puntos de impacto que trae este refactor en el proyecto. Tendría tests unitarios con coverage alto como precaución, para luego correrlos y verificar si alguno falla, como primera instancia.
* ¿Agregaría getters y setters? ¿Cuáles? ¿Por qué?
* Sí, todos getters y setters públicos salvo el setter del saldo que es *protected* para la jerarquía de clases de ***Cuenta***. Agregué estos getters y setters porque el código base de la clase ***Cuenta*** tenía todos sus atributos como *private.*
* ¿Su solución usa algún patrón de diseño? ¿Cuál?
* No necesité, solo utilicé herencia.