

ESCUELA COLOMBIANA DE INGENIERÍA PROGRAMACIÓN ORIENTADA A OBJETOS

S06: 2025-01

G01

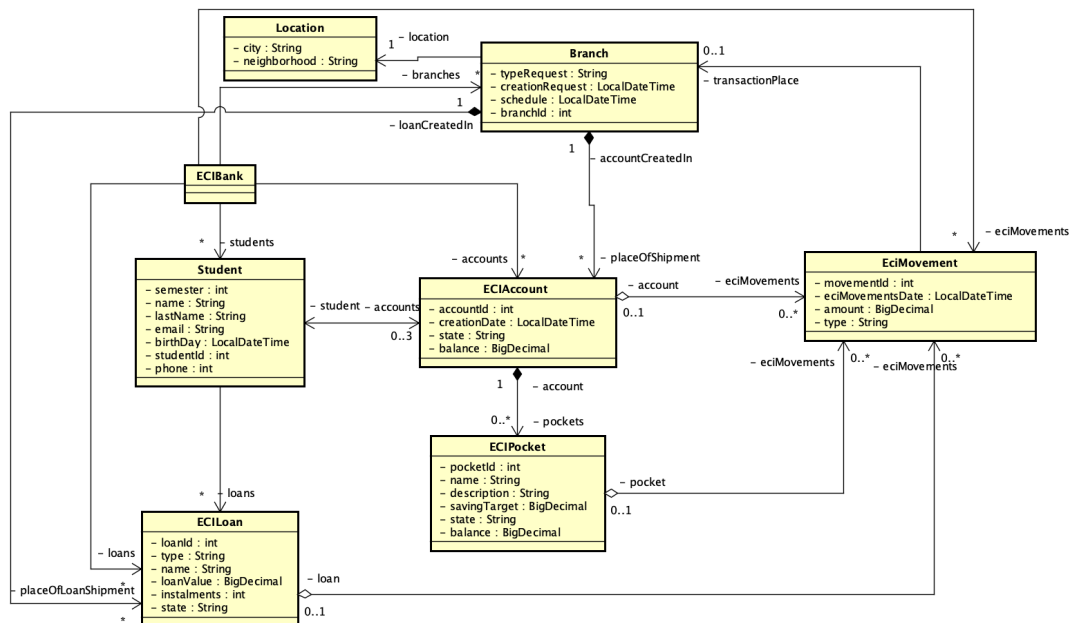
Parcial primer tercio. Nota esperada: ____

ECIBank – Banco para estudiantes

La ESCUELA está creando un banco para sus estudiantes, llamado **ECIBank**, que permitirá gestionar un sistema bancario interno dentro de la universidad. Este banco ofrecerá a los estudiantes cuentas de ahorro para que puedan depositar, retirar, consultar sus movimientos e incluso ahorrar en sus **ECIPockets** dentro de la universidad. Además, se planea crear crédito, **ECILoans**, para que los estudiantes puedan administrar los diversos pagos que deben realizar a lo largo del semestre. Sin embargo, el personal de Osiris ha estado ocupado, por lo que se dirigieron a la Decanatura de Sistemas para designar a los estudiantes **POOB** de este semestre, con el fin de construir la solución que haga posible este proyecto; **ECIBank** tiene entre sus prioridades estas funciones:

1. **Creación y cierre de cuentas de ahorros:** El estudiante que desee abrir su **ECIAccount** en **ECIBank** debe asegurarse de que esté inscrito en un semestre superior al tercero de su carrera. No se permitirá a un estudiante crear la cuenta más de tres veces. Además, es fundamental contar con el número de contacto o el correo electrónico del estudiante para poder comunicarse con él.
2. **Pedir un crédito para el pago de su matrícula:** La universidad puede otorgar un **ECILoan** al estudiante para el pago de su matrícula semestral. Sin embargo, el estudiante no podrá tener más de dos créditos activos a la vez. En caso contrario, no se podrá otorgar un nuevo préstamo hasta que se libere alguna de las deudas.
3. **Crear ahorros en sus bolsillos:** Para incentivar un buen hábito de ahorro, la universidad permite a los estudiantes tener hasta diez **ECIPockets**. En cada uno de estos bolsillos, el estudiante debe especificar el nombre del ahorro, su propósito y el monto destinado. Además, es importante que cada bolsillo tenga movimiento al menos una vez cada seis meses, de lo contrario, se considerará inactivo y se cerrará automáticamente.
4. **Depositar, retirar y consulta de su ECIAccount:** El estudiante solo puede hacer depósitos mayores a \$10000, y retiros mayores y múltiplos de \$20000 debido a que tenemos afiliación con la red verde de servibanca, cuando un estudiante hace cualquiera de estos dos movimientos, el estudiante podrá verlos en sus **ECIMovements**.
5. **ECIMovements** son transacciones que hizo una **ECIAccount**, es importante para el equipo de backoffice que los movimientos tenga la fecha de creación, el monto por el cual se hizo la transacción y el tipo de la misma.
6. **ECIBank** cuenta con dos métodos para gestionar las peticiones de sus estudiantes. Uno es a través de la página que se desarrollará por parte de Osiris, y el otro mediante diferentes módulos disponibles en la universidad. Cada módulo tiene su especialización y horario: algunos se encargan de temas relacionados con cuentas, mientras que otros se enfocan en los préstamos.
7. Todos los productos de **ECIBank** cuentan con tres estados fundamentales: **PROCESSING**, el estado en proceso, que indica que un producto en el banco está en proceso de creación; **ACTIVE**, un estado que señala que el producto está activo; y **CLOSED**, el estado de cierre del producto.

Diseño



(Todos los contenedores son TreeMap, excepto los movimientos, las keys de los mapas son los identificadores de cada clase). (25%) MEMORIA

Presente el mapa de memoria correspondiente a:

- Jairo Gil es un estudiante de octavo semestre que posee una cuenta en ECIBank desde el 10 de enero del presente año. En dicha cuenta, sus padres le han realizado depósitos mensuales de \$400.000 pesos, sumando un total de \$800.000 pesos hasta la fecha. Estos depósitos, destinados a sus gastos de alimentación, se efectúan el día 12 de cada mes

El correo institucional de Jairo en la Escuela es jairo.gil@mail.escuelaing.edu.co.

- ## II. (35%) CÓDIGO

1. Estudie el diagrama de secuencia y la especificación (documentación + encabezado) del método
2. Actualice el diagrama de clase con los nuevos elementos
3. Escriba el código de la clase responsable inicial (encabezado y atributos). Documente el invariante.
4. Implemente cada uno de los métodos correspondientes a la solución. Incluya la documentación.

- studentID: Identificador del estudiante.
- loanValue: valor que se desea pagar.
- loanId: Identificador del préstamo.

[illegible]

- Un crédito se considera en mora si no ha tenido movimientos en los últimos 2 meses.
- Se creará un movimiento con el valor de la cuota adeudada para cubrir el saldo en mora siempre y cuando el estudiante tenga saldo para cubrir la deuda en sus cuentas o bolsillos.
- El método retornará la lista de estudiantes que se encuentran en mora y no se les pudo debitar el valor adeudado por falta de fondos.
- Para calcular la cuota se divide el valor del préstamo en el plazo que corresponde al atributo instalments

- Mencione algunos mecanismos o buenas prácticas para la implementación del código en JAVA.
- ¿Por qué se dice que Java posee una arquitectura neutral y cómo influye esto en su portabilidad?