

Juan Daniel Bogotá Fuentes

Desarrollo orientado por objetos

DOPO

PRE-PARCIAL 1er TERCIO

15/09/2025

PROFESOR: ANGIE TATIANA MEDINA GIL

Contenido

[I. MEMORIA: 3](#_Toc208872363)

[II. CÓDIGO: 3](#_Toc208872364)

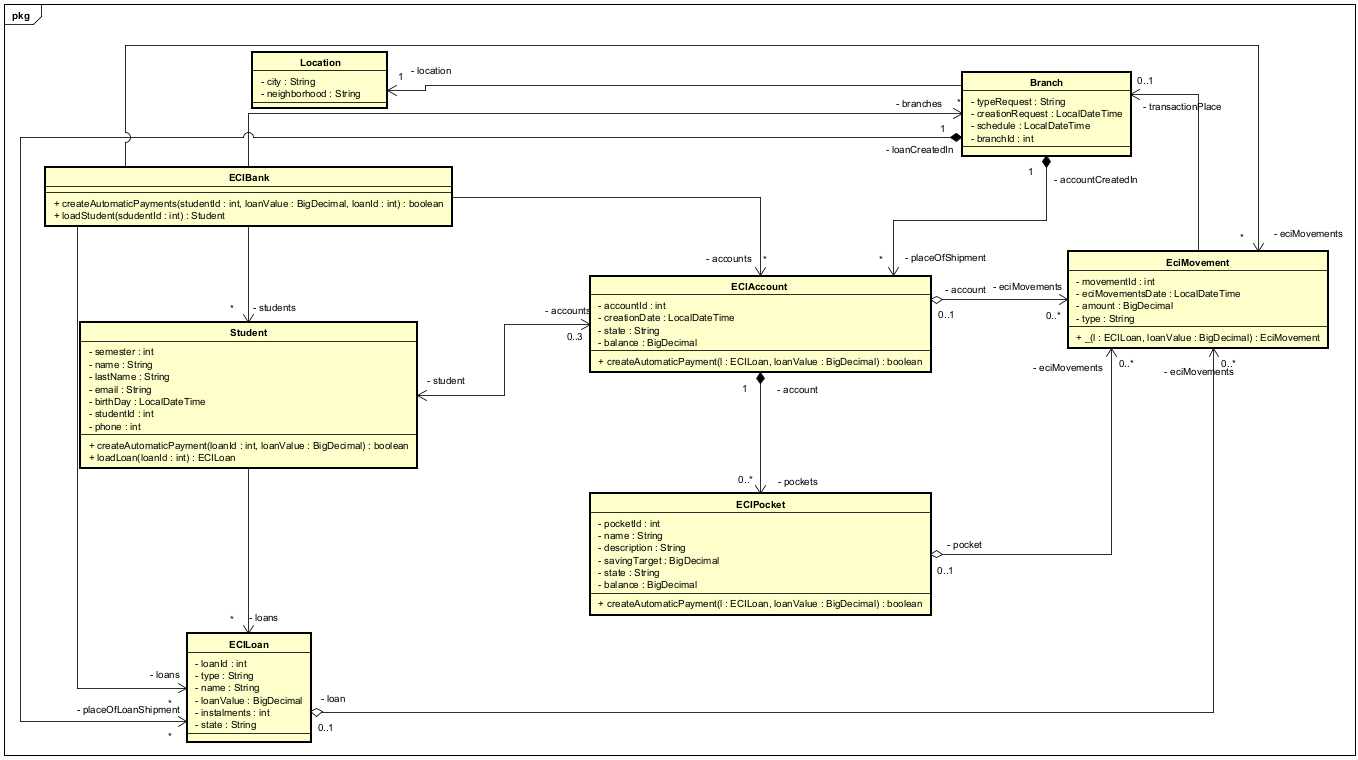
[III. DISEÑO: 5](#_Toc208872365)

[IV. CONCEPTOS: 6](#_Toc208872366)

# image1757985577809.pngMEMORIA:

# CÓDIGO:

1. Diagrama de clases actualizado.



1. Encabezado y atributos de ECIBank, además de la documentación de la invariante.

import java.time.LocalDateTime;

import java.util.TreeMap;

/\*

 \* Class representing a bank with students, branches, accounts, and loans.

 \* Each entity is stored in a TreeMap for efficient access and management.

 \* @author Juan Daniel Bogotá

 \* @version 1.0

 \* @since 2025-09-15

 \*/

 /\* Invariante de clase:

 \* 1) students, branches, accounts y loans son no nulos.

 \* 2) Las llaves en cada TreeMap son únicas.

 \* 3) Toda cuenta referencia a un Student existente.

 \* 4) Todo préstamo referencia a un Student existente.

 \* 5) No deben existir cuentas con identificadores negativos ni préstamos con montos negativos.

 \* 6) No deben existir estudiantes con identificadores negativos.

 \*/

public class ECIBank {

    private TreeMap<Integer, Student> students;

    private TreeMap<Integer, Branch> branches;

    private TreeMap<Integer, ECIAccount> accounts;

    private TreeMap<Integer, ECILoan> loans;

    public ECIBank() {

        students = new TreeMap<>();

        branches = new TreeMap<>();

        accounts = new TreeMap<>();

        loans = new TreeMap<>();

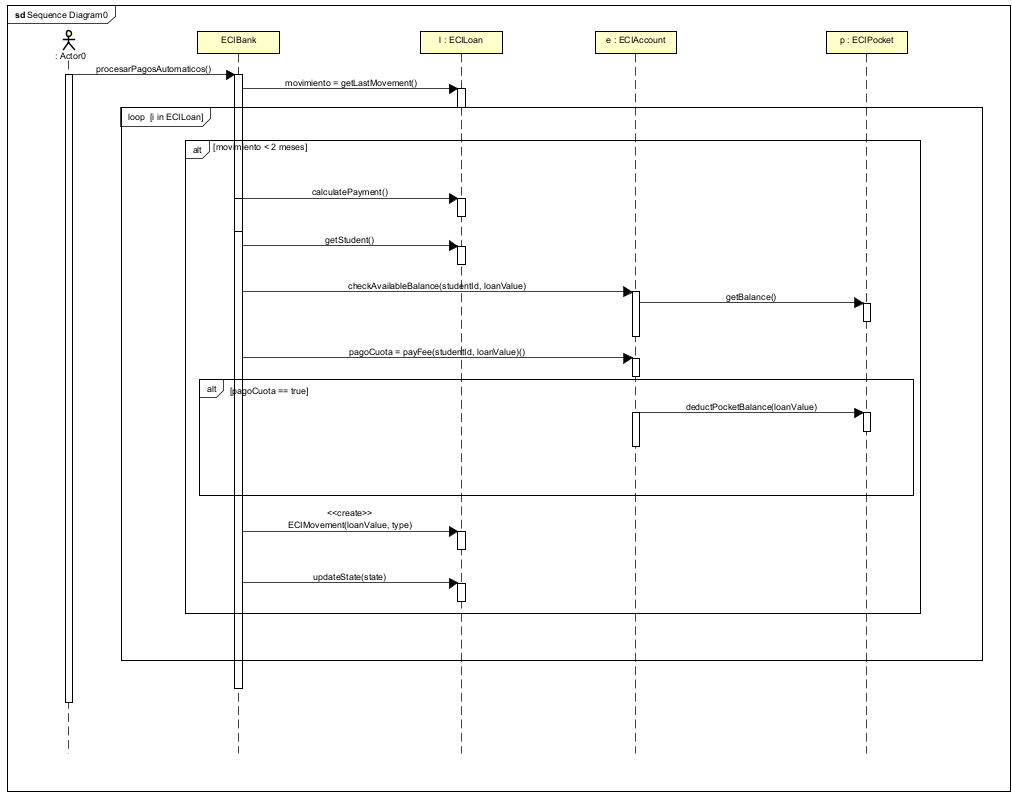
    }

}

1. Implementación de cada uno de los métodos.

Está en la carpeta .zip, una carperta llamada eciBank, ahí se encuentra todo el proyecto BlueJ.

# DISEÑO:



# CONCEPTOS:

* Mencione algunos mecanismos o buenas prácticas para la implementación del código en JAVA.
  + Encapsulamiento: Mantener los atributos como private, así proteger la integridad de los datos y tener acceso controlado.
  + camelCase: para las variables y método, ejemplo: sumarNumeros(int a, int b).
  + Pruebas unitarias: validar que los métodos funcionan según lo esperado y asegurar mantenibilidad del código.
  + Respetar principios SOLID, por ejemplo, el principio de responsabilidad única, cada clase debe tener un único propósito, por ejemplo, ECIMovement solo registra movimientos, no debería calcular préstamos.
* ¿Por qué se dice que Java posee una arquitectura neutral y cómo influye esto en su portabilidad?
  + Java usa una maquina virtual (JVM), esta disponible para muchas plataformas dando compatibilidad, por ejemplo, para Windows, macOs, Android, etc.
  + Es una maquina virtual porque compila el mismo tipo de archivo independiente de la máquina.