**Aplicación de los conceptos de Programación Orientada a Objetos (POO) en la Calculadora**

**Cristian David Hincapie Montes**

**Adán Beltrán Gómez**

**Uniempresarial, Bogota**

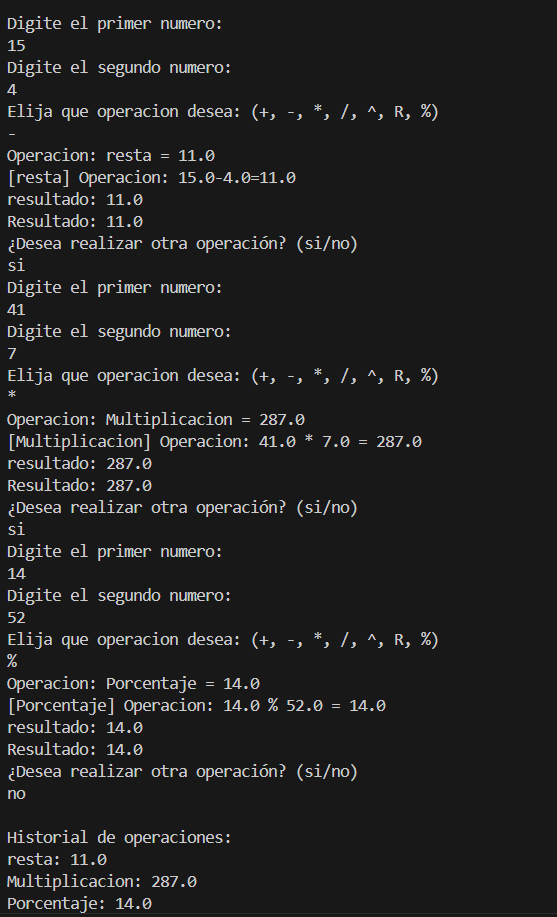
**Ingeniería de software**

**07/04/2025**

# 1. Introducción

En el siguiente documento se muestra cómo se han aplicado los pilares básicos de la Programación Orientada a Objetos (POO) en la construcción de una calculadora en el lenguaje Java, la cual permite realizar las operaciones básicas de suma, resta, multiplicación, división, potenciación, raíz cuadrada y módulo (porcentaje). Así, hemos implementado el histórico y la persistencia en un archivo.

**Capturas de pantalla de la ejecución.**



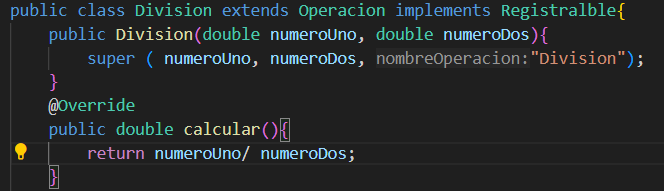
1. Clases y Objetos

El diseño del programa se basa en múltiples clases que representan entidades y responsabilidades específicas:

* Operacion: clase abstracta base para todas las operaciones matemáticas.
* Suma, Resta, Multiplicacion, Division, Potenciacion, Raiz, Porcentaje: subclases que heredan de Operacion.
* Calculadora: se encarga de ejecutar cualquier operación.
* Historial: registra las operaciones realizadas y las guarda en un archivo.
* UI: gestiona la entrada/salida del usuario.
* Main: clase principal que coordina la ejecución del programa.

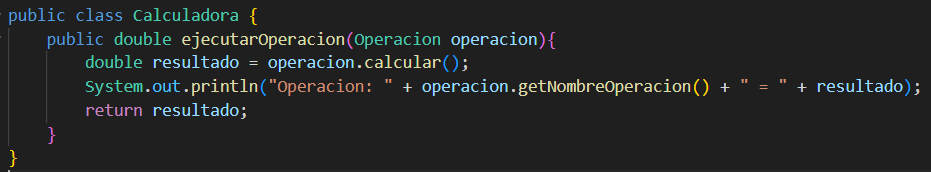
2. Herencia

Se aplica herencia obligando a todas las clases de operación (Suma, Resta, etc.) a heredar de la clase abstracta Operacion, de manera que todas comparten los mismos atributos (numeroUno, numeroDos, nombreOperacion), y también garantizando que todas las clases proporcionan la implementación del método calcular.



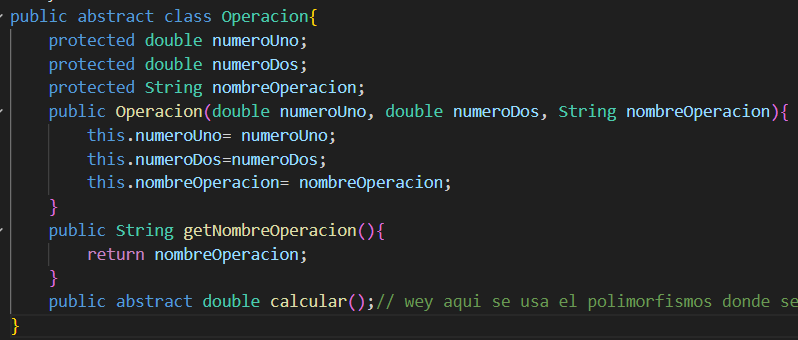
3. Polimorfismo

El polimorfismo se produce en el momento que el método ejecutarOperacion() de la clase Calculadora recibe como parámetro una instancia de Operacion con las clases Suma, Resta, etc. Extensibilidad y reutilización del código.



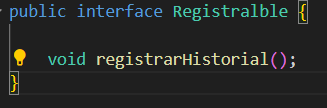
4. Abstracción

La clase Operación representa una abstracción de una operación matemática. Ésta no es instanciable y representa el método abstracto calcular() para forzar a las subclases a implementar este método, para permitir manejar solo lo necesario y ocultar el resto.



5. Interfaces

La interfaz Registrable representa un contrato que van a tener que cumplir todas las operaciones al momento de registrar el historial. Todas las clases de operación que implementan la interfaz Registrable son las encargadas de implementar el método registrarHistorial().



6. Separación de responsabilidades

Cada una de las clases del programa tiene definido un rol, y esto va a facilitar el mantenimiento del código:

* Operacion y sus subclases: lógica de cada operación matemática.
* Calculadora: ejecución operaciones.
* Historial: gestión y grabación del historial de operaciones.
* UI: combinación con el usuario.
* Main: orquestar la ejecución y la validación.

### 2 Archivo historial.txt con las operaciones realizadas

