

## Objetivos

1. Aprender a aplicar un patrón "de comportamiento" a un contexto de diseño específico.
2. Comprender los detalles del patrón "Método Plantilla".

## Planteamiento del problema

Diseñe una pequeña librería con clases que representen "operandos" en general, sobre los cuales se pueden aplicar las cuatro operaciones básicas: +, -, \* y /. Agregue la evaluación de expresiones posfijas para al menos tres tipos de operandos:

1. Números "dobles", use un Adaptador o "Wrapper" del tipo primitivo "double".
2. Fracciones: 1/2, 3/5, etc, se deberá crear la clase Racional o Fraccion.
3. Conjuntos: se deberá extender <set> de la STL sobrecargando los operadores. En este caso el operador de suma representará la unión de conjuntos, el operador de multiplicación la intersección, el de resta la diferencia y el de división la diferencia simétrica.

NOTA: para todos los tipos de operandos, la evaluación de expresiones posfijas deberá retornar un valor del mismo tipo de los operandos que se podrá visualizar por consola mediante "toString()", como una tira de caracteres.

## Características de la solución:

1. Debe usar el patrón Método Plantilla de manera que el código para la evaluación de expresiones pos-fijas sea código polimórfico que permita el enlace dinámico y que pueda funcionar con cualquier otro tipo de operando que se agregue en el futuro.
2. Es claro que se ocupa una clase para cada tipo de operando.
3. Debe basarse en el algoritmo para evaluar expresiones pos-fijas en: [https://es.wikipedia.org/wiki/Pila\\_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Pila_(inform%C3%A1tica))
4. Simplifique al máximo el análisis de la hilera posfija suponiendo que está bien construida. Se sugiere que el código evaluador de posfijas se base en expresiones basadas en variables únicamente, de manera que no haya que hacer un análisis diferente para cada tipo de operando.
5. El controlador o main() debe mostrar que: 1) se pueden crear operandos de los tres tipos, 2) se pueden procesar correctamente expresiones posfijas para cada tipo de operando, 3) al menos una de las expresiones posfijas para cada tipo de operando incluye los cuatro operadores. Para el caso particular de los conjuntos, el main demostrará su uso especializando < set > con el tipo "int".
6. El Método Plantilla NO puede ser un método que reciba un parámetro de tipo, ni una función con parámetro de tipo ni pertenecer a una clase emplatillada, que recibe un parámetro de tipo.

## Evaluación:

Rubro	Peso
Aplicación correcta de principios SOLID	30%
Aplicación correcta de Constructor	40%
Funcionamiento correcto del código	20%
Uso correcto de UML en el modelo de clases	10%