

# Evaluador y Validador Postfijo con Autómata de Pila

## Proyecto Final - Teoría de la Computación

Diego Sotelo   Alexis Gonzales   Paolo Villavicencio   Álvaro Salazar

**Universidad Nacional Mayor de San Marcos**  
Facultad de Ingeniería de Sistemas e Informática

*Docente: Víctor Hugo Bustamante Olivera*

1 de diciembre de 2025

# Agenda

- 1 Introducción
- 2 Objetivos
- 3 Marco Teórico
- 4 Desarrollo
- 5 Resultados
- 6 Conclusiones

## Notación Infija (Habitual)

- Requiere paréntesis para prioridad.
- Ambigüedad en el orden de operaciones.
- Complejo de evaluar algorítmicamente.
- Ej:  $3 + 4 * 2$

## Notación Postfija (RPN)

- No usa paréntesis.
- Orden lineal de evaluación.
- Ideal para estructuras LIFO (Pilas).
- Ej:  $3\ 4\ 2\ *\ +$

## Propuesta

Implementar un **Autómata de Pila (AP)** que evalúe expresiones RPN y valide su sintaxis, integrando lectura y escritura de archivos.

# Objetivos del Proyecto

## Objetivo General

Desarrollar un evaluador de expresiones postfijas basado en Autómata de Pila, capaz de calcular resultados y rechazar entradas inválidas.

## Objetivos Específicos:

- Formalizar el modelo matemático  $(E, A, P, \delta, e_0, Z_0, F)$ .
- Implementar lógica de validación (división por cero, falta de operandos).
- Desarrollar módulo de persistencia (Lectura/Escritura de .txt).
- Generar reportes automáticos de la evolución de la pila.

# Autómata de Pila (AP)

Un modelo matemático que extiende al autómata finito con una memoria auxiliar (pila) de tipo LIFO (Last In, First Out).

$$APD = \langle E, A, P, \delta, q_0, Z_0, F \rangle$$

- **E**: Conjunto de estados.
- **A**: Alfabeto de entrada.
- **P**: Alfabeto de pila ( $P \cap A = \emptyset$ ).
- $\delta$ : Función de transición.
- $Z_0$ : Símbolo de fondo de pila.

## Criterio de Aceptación:

- 1 Cadena consumida totalmente.
- 2 Pila vacía
- 3 Estado final alcanzado.

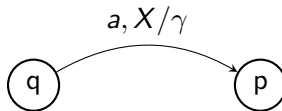
La función de transición define el comportamiento:

$$\delta(q, a, X) = (p, \gamma)$$

**Lee:** Símbolo  $a$  de la entrada.

**Extrae:** Símbolo  $X$  de la cima de la pila.

**Escribe:** Cadena  $\gamma$  en la pila.



*Transición de estado  $q$  a  $p$  leyendo  $a$ , sacando  $X$  y apilando  $\gamma$ .*

# Formalización del Autómata Diseñado

Nuestro autómata para RPN se define como:

$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \emptyset)$$

- $Q = \{q_1\}$  (Un solo estado principal).
- $\Sigma = \{V, +, -, *, /\}$  (Donde  $V$  es un valor numérico).
- $\Gamma = \{X, Z_0\}$  (Alfabeto de pila).
- Aceptación por **Pila Vacía**.

**Regla de Oro:**

- **Operando:** Push  $X$ .
- **Operador:** Pop  $X$ , Pop  $X$ , Push  $X$ .

# Tabla de Transiciones $\delta$

#	Entrada	Cima Pila	Acción	Semántica
1	V (Valor)	$Z_0$	Push X	Primer operando
2	V (Valor)	X	Push X	Operandos sucesivos
3	+	XX	Pop 2, Push 1	Suma
4	-	XX	Pop 2, Push 1	Resta
5	*	XX	Pop 2, Push 1	Multiplicación
6	/	XX	Pop 2, Push 1	División
7	$\varepsilon$	$XZ_0$	Pop X, $Z_0$	<b>ACEPTACIÓN</b>

Cuadro: Mapa de transiciones para evaluación RPN



El sistema no es solo interactivo, permite procesamiento por lotes:

❶ **Entrada** (`entrada.txt`):

- Lectura de la cadena cruda.
- Tokenización y limpieza.

❷ **Procesamiento:**

- Ejecución del AP.
- Captura de excepciones (`ZeroDivisionError`, `StackUnderflow`).

❸ **Salida:**

- `evolucion_entrada.txt`: Traza paso a paso de la pila.
- `resultado_entrada.txt`: Valor final o mensaje de `ERROR`.

# Casos de Prueba

Se validó la robustez con diferentes escenarios:

Entrada (RPN)	Salida Esperada	Estado
3 4 + 2 *	14	Éxito
10 2 /	5	Éxito
5 0 /	ERROR (Div/0)	Detectado
3 +	ERROR (Sintaxis)	Detectado
3 4 5 +	ERROR (Pila > 1)	Detectado

\* El sistema genera correctamente los archivos de reporte ante fallos sintácticos o semánticos.

- ❶ **Eficacia del Modelo:** El Autómata de Pila es la estructura natural y más eficiente para procesar lenguajes libres de contexto como las expresiones aritméticas sin paréntesis (RPN).
- ❷ **Integridad de Datos:** La implementación diferencia correctamente entre errores de estructura (sintaxis RPN inválida) y errores matemáticos (división por cero).
- ❸ **Automatización:** El módulo de archivos eleva la utilidad del proyecto, permitiendo validar lotes de expresiones y generando trazas de auditoría (evolución de la pila) automáticamente.

# ¡Muchas Gracias!

¿Preguntas?

*Universidad Nacional Mayor de San Marcos  
Facultad de Ingeniería de Sistemas e Informática*