

EVALUACION T2

Asignatura	ESTRUCTURA DE DATOS	Código	SIST1201A
Modalidad:	PRESENCIAL	Ciclo:	IV
Docente:	Dr. Eric Gustavo Coronel Castillo	Semestre:	2025-2
Tema	Arboles binarios		

DESAFÍO

SISTEMA DE EXPRESIONES MATEMÁTICAS JERÁRQUICAS

Contexto

Una academia de matemáticas en línea requiere un sistema que permita a los estudiantes ingresar expresiones matemáticas complejas y visualizar su estructura jerárquica. El sistema debe ayudar a los estudiantes a comprender el orden de evaluación de operadores, identificar subexpresiones y calcular resultados paso a paso.

Problema

Los estudiantes tienen dificultades para:

- Entender la precedencia de operadores en expresiones complejas
- Visualizar cómo se agrupan las operaciones matemáticas
- Identificar errores en la estructura de expresiones mal formadas
- Evaluar expresiones siguiendo el orden correcto de operaciones

Ejemplo de expresión: $((5 + 3) * 2) - (8 / (4 - 2))$

Solución Solicitada

Desarrollar un sistema en C# que utilice un Árbol Binario de Expresiones donde:

- Los **nodos terminales** (hojas) contienen valores numéricos (operandos).
- Los **nodos no terminales** (con hijos) representan operadores matemáticos (+, -, *, /, ^).

Funcionalidades obligatorias

1. **Parsear expresiones** aritméticas en notación infija y construir el árbol
2. **Validar expresiones** y detectar errores de sintaxis
3. **Visualizar el árbol** de forma gráfica o textual indentada
4. **Evaluar la expresión** mediante recorrido del árbol
5. **Convertir entre notaciones:**
 - De Infija a Prefija (mediante recorrido PreOrden)
 - De Infija a Postfija (mediante recorrido PostOrden)
6. **Identificar subexpresiones** críticas o caminos específicos
7. **Calcular la altura** del árbol (complejidad de la expresión)
8. **Contar operadores** de cada tipo en la expresión

Tipo de aplicación requerida

- Aplicación de Consola en C# (Console Application)
- Justificación: Permite enfocarse en la implementación del árbol binario y sus algoritmos sin complejidad adicional de interfaces gráficas

RECOMENDACIONES

Menú Principal Interactivo

SISTEMA DE ÁRBOLES DE EXPRESIONES MATEMÁTICAS

- [1] Ingresar nueva expresión
- [2] Visualizar árbol de expresión actual
- [3] Evaluar expresión
- [4] Convertir a notación Prefija
- [5] Convertir a notación Postfija
- [6] Mostrar altura del árbol
- [7] Contar operadores por tipo
- [8] Ver historial de expresiones
- [9] Ayuda - Operadores y sintaxis válida
- [0] Salir

Seleccione una opción: _

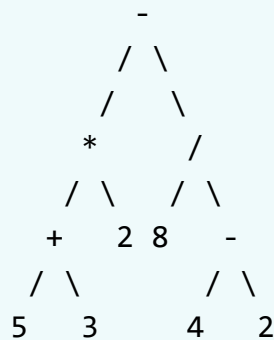
Visualización del Árbol (formato ASCII art)

Usar representación indentada jerárquica.

Ejemplo de salida esperada:

Expresión: $((5 + 3) * 2) - (8 / (4 - 2))$

Árbol de Expresión:



Profundidad: 4 niveles

Validación y Manejo de Errores

Mensajes claros cuando la expresión es inválida.

Indicar la posición exacta del error si es posible.

Ejemplos de mensajes:

- ✗ Error: Paréntesis desbalanceados en la posición 8
- ✗ Error: Operador inválido '%' no reconocido
- ✗ Error: Expresión vacía. Por favor ingrese una expresión válida
- ✓ Expresión válida y parseada correctamente

Salida de Conversiones

Mostrar claramente cada tipo de notación.

Conversiones de: $(5 + 3) * 2$

Notación Infija: $(5 + 3) * 2$

Notación Prefija: $* + 5 3 2$

Notación Postfija: $5 3 + 2 *$

Evaluación Paso a Paso (opcional, para destacar):

Evaluando: $((5 + 3) * 2) - (8 / (4 - 2))$

Paso 1: Evaluar $(5 + 3) = 8$

Paso 2: Evaluar $8 * 2 = 16$

Paso 3: Evaluar $(4 - 2) = 2$

Paso 4: Evaluar $8 / 2 = 4$

Paso 5: Evaluar $16 - 4 = 12$

✓ Resultado final: 12

Estadísticas del Árbol

ESTADÍSTICAS DE LA EXPRESIÓN

Altura del árbol:	4
Total de nodos:	9
Nodos operadores:	4
Nodos operandos:	5

Conteo por operador:

Suma (+):	1
Resta (-):	2
Multiplicación (*):	1
División (/):	1

Ayuda integrada

AYUDA - SINTAXIS DE EXPRESIONES

Operadores soportados:

- + Suma
- Resta
- * Multiplicación
- / División
- ^ Potencia (opcional)

Reglas:

- Use paréntesis para agrupar operaciones
- Los números pueden ser enteros o decimales
- Respete espacios entre operadores

Ejemplos válidos:

$(5 + 3) * 2$

$10 / (2 + 3)$
 $((8 - 2) * 3) + 5$

Presione cualquier tecla para continuar...

ENTREGABLE

- **Informe Académico (formato PDF) que incluya:**
 - ✓ Fundamentos teóricos
 - ✓ Explicación del algoritmo de construcción del árbol desde la expresión
 - ✓ Justificación del uso de cada tipo de recorrido
 - ✓ Comparación de notación Infija, Prefija y Postfija con ejemplos
 - ✓ Análisis de casos especiales (paréntesis, precedencia de operadores)
 - ✓ Diagramas de árboles para al menos 5 expresiones de diferente complejidad
 - ✓ Capturas de pantalla de la aplicación de consola funcionando
 - ✓ Conclusiones
 - ✓ Recomendaciones
- **Código fuente en C# con:**
 - ✓ Implementación completa del árbol binario de expresiones
 - ✓ Analizador sintáctico (parser) robusto
 - ✓ Manejador de errores con mensajes descriptivos
 - ✓ Interfaz de consola siguiendo las recomendaciones especificadas
 - ✓ Batería de pruebas con expresiones válidas e inválidas
 - ✓ Comentarios explicativos en el código
 - ✓ Archivo README.txt con instrucciones de compilación y ejecución

CRITERIO DE EVALUACIÓN

ITEM	ITEM	PUNTAJE MAXIMO
1	Informe académico	20
2	Desarrollo del proyecto	20
3	Video en YouTube de la exposición del trabajo, todos los integrantes del grupo deben participar y con cámara activa.	20
4	Exposición en aula y respuesta a preguntas.	20

La nota final es el promedio de los 4 ítems.