

Evaluador de Expresiones Matemáticas

Manual de Uso y Guía de Instalación

Luciano David Duarte
lucianoduarte.wow@gmail.com

Diciembre 2025

Índice

1. Introducción	3
1.1. ¿Qué puede hacer este programa?	3
2. Requisitos del Sistema	3
2.1. Requisitos de Hardware	3
2.2. Sistemas Operativos Soportados	3
3. Primeros Pasos	3
3.1. Modo Interactivo	3
3.2. Procesar Archivos	4
4. Formato de Archivos	4
4.1. Sintaxis	4
4.2. Ejemplo de Archivo	4
4.3. Reglas del Formato	4
5. Funciones Soportadas	5
5.1. Operadores Aritméticos	5
5.2. Funciones Trigonométricas	5
5.3. Funciones Hiperbólicas	5
6. Ejemplos de Uso	6
6.1. Ejemplo 1: Derivada de un Polinomio	6
6.2. Ejemplo 2: Funciones Trigonométricas	6
6.3. Ejemplo 3: Función Compuesta	6
6.4. Ejemplo 4: Optimización Automática	6
7. Archivos de Ejemplo Incluidos	7
8. Manejo de Errores	7
8.0.1. Error de Dominio	7
8.0.2. División por Cero	7
8.0.3. Variable No Definida	7

1. Introducción

Bienvenido al **Evaluador de Expresiones Matemáticas con Diferenciación Automática**. Este programa permite evaluar expresiones matemáticas complejas y calcular sus derivadas de forma automática y precisa.

1.1. ¿Qué puede hacer este programa?

Capacidades Principales

- Evaluar expresiones matemáticas en un punto específico
- Calcular derivadas automáticamente usando números duales
- Procesar múltiples expresiones desde archivos
- Modo interactivo para experimentación
- Optimizar expresiones algebraicas automáticamente
- Validar dominios matemáticos y detectar errores

2. Requisitos del Sistema

2.1. Requisitos de Hardware

- **Procesador:** Cualquier CPU moderna (x86_64 o ARM)
- **RAM:** 512 MB mínimo, 1 GB recomendado
- **Disco:** 500 MB de espacio libre

2.2. Sistemas Operativos Soportados

- Windows 10/11 (64-bit)
- macOS 10.15 o superior
- Linux (cualquier distribución moderna)

3. Primeros Pasos

3.1. Modo Interactivo

El modo interactivo permite experimentar con expresiones:

```
1 cabal run ALP2025-LCC
```

Ejemplo de Sesión Interactiva

```

=== Evaluador de Expresiones con Derivadas ===
Escribe una expresión matemática:
Por ejemplo: sin(x) + x^2
Escribe 'salir' o 'quit' para finalizar.

>>> x^2 + 2*x + 1
AST generado: Add (Add (Pow (Var "x") (Lit 2.0))
                    (Mul (Lit 2.0) (Var "x"))) (Lit 1.0)
Ingrese el valor de x: 3
Valor f(3.0): 16.0
Derivada f'(3.0): 8.0

>>> salir

```

3.2. Procesar Archivos

Para procesar múltiples expresiones desde un archivo:

```

1 cabal run ALP2025-LCC -- -f archivo.txt
2 # 0 usando la forma larga:
3 cabal run ALP2025-LCC -- --file archivo.txt

```

4. Formato de Archivos

4.1. Sintaxis

Formato de Entrada

expresión @ valor_x

4.2. Ejemplo de Archivo

Cree un archivo ejemplos.txt:

```

1 -- Esto es un comentario
2 x^2 @ 3
3 sin(x) + cos(x) @ 0.5
4 log(x) * exp(x) @ 2
5 sqrt(x) @ 9
6 -- Funciones compuestas
7 sin(x^2) @ 1.5
8 exp(-x) * x @ 1

```

4.3. Reglas del Formato

- Una expresión por línea

- Comentarios comienzan con `--`
- El símbolo `@` separa expresión y valor
- Líneas vacías se ignoran

5. Funciones Soportadas

5.1. Operadores Aritméticos

Operador	Descripción
<code>+</code>	Suma
<code>-</code>	Resta
<code>*</code>	Multiplicación
<code>/</code>	División
<code>^</code>	Potenciación

Cuadro 1: Operadores aritméticos

5.2. Funciones Trigonométricas

Función	Descripción
<code>sin(x)</code>	Seno
<code>cos(x)</code>	Coseno
<code>tan(x)</code>	Tangente

Cuadro 2: Funciones trigonométricas

5.3. Funciones Hiperbólicas

Función	Descripción
<code>sinh(x)</code>	Seno hiperbólico
<code>cosh(x)</code>	Coseno hiperbólico
<code>tanh(x)</code>	Tangente hiperbólica
<code>arsinh(x)</code>	Arcoseno hiperbólico
<code>arcosh(x)</code>	Arcocoseno hiperbólico ($x \geq 1$)
<code>artanh(x)</code>	Arcotangente hiperbólica ($ x < 1$)

Cuadro 3: Funciones hiperbólicas

6. Ejemplos de Uso

6.1. Ejemplo 1: Derivada de un Polinomio

Polinomio Cuadrático

Expresión: $f(x) = x^2 + 2x + 1$

Comando:

```
1 x^2 + 2*x + 1 @ 3
```

Resultado:

- $f(3) = 16,0$
- $f'(3) = 8,0$ (correcto: $f'(x) = 2x + 2$)

6.2. Ejemplo 2: Funciones Trigonómicas

Identidad Trigonométrica

Expresión: $f(x) = \sin^2(x) + \cos^2(x)$

Comando:

```
1 sin(x)^2 + cos(x)^2 @ 0.5
```

Resultado:

- $f(0,5) \approx 1,0$ (verifica la identidad)
- $f'(0,5) \approx 0,0$ (derivada nula)

6.3. Ejemplo 3: Función Compuesta

Composición de Funciones

Expresión: $f(x) = e^{\sin(x)}$

Comando:

```
1 exp(sin(x)) @ 0
```

Resultado:

- $f(0) = 1,0$ (ya que $\sin(0) = 0$ y $e^0 = 1$)
- $f'(0) = 1,0$ (por la regla de la cadena)

6.4. Ejemplo 4: Optimización Automática

El programa optimiza automáticamente:

```
1 x + 0 @ 5
2 # Optimizado a: x
```

```
3 # Resultado: f(5) = 5, f'(5) = 1
```

7. Archivos de Ejemplo Incluidos

El proyecto incluye varios archivos de ejemplo en la carpeta `examples/`:

Archivo	Contenido
<code>basico.txt</code>	Funciones básicas
<code>trigonometricas.txt</code>	Funciones trigonométricas
<code>hiperbolicas.txt</code>	Funciones hiperbólicas
<code>compuestas.txt</code>	Funciones compuestas
<code>constantes.txt</code>	Uso de π y e
<code>numeros_negativos.txt</code>	Números negativos
<code>nuevas_caracteristicas.txt</code>	sqrt y optimizaciones

Cuadro 4: Archivos de ejemplo

8. Manejo de Errores

8.0.1. Error de Dominio

Ejemplo de Error

Input: `log(-1) @ 0`

[X] Error al evaluar: `DomainError "log requires positive argument"`

Solución: Verificar que el argumento cumpla el dominio de la función.

8.0.2. División por Cero

Ejemplo de Error

Input: `1/0 @ 0`

[X] Error al evaluar: `DivideByZero`

Solución: Evitar divisiones por cero en las expresiones.

8.0.3. Variable No Definida

Ejemplo de Error

Input: `y + 2 @ 1`

[X] Error al evaluar: `UndefinedVariable "y"`

Solución: Usar solo la variable `x`, o las constantes `pi` y `e`.