



Tecnológico Nacional de México
Instituto Tecnológico de Minatitlán
Ingeniería en Sistemas Computacionales

Alumno:

-Mathey Jimenez Joshep Arturo-



-PROYECTO INTEGRADOR-

HORARIO DEL GRUPO: 3:00pm - 4:00pm

Docente: KEVIN DAVID MOLINA GOMEZ

Ciudad y Fecha de entrega:

MINATITLÁN VERACRUZ, VIERNES 30 DE MAYO DEL 2025

1. instrucciones para ejecutar el proyecto.

Mi mini IDE se trata de poder hacer operaciones aritméticas con un compilador este funciona de tal manera que si pones el código (la operación que quieres hacer) ya sea de suma, resta, multiplicación, división, potencia, raíz cuadrada todas estas operaciones aritméticas para poder cumplir la gramática y tener una estructura correcta y completa pondremos una "Y" este servirá como nuestro conector de nuestra operación y son las que pueden ser analizadas con diferentes análisis como lo es el sintáctico que este funciona de tal manera que hace ver si lo que ponemos como código este dentro de la gramática, que lo hayamos escrito de la forma correcta y que este dentro del diccionario que nosotros mismos le vamos a proporcionar a la hora de hacer el análisis léxico y es ahí donde entra el otro análisis que es el léxico este funciona para asignar el diccionario de palabras con las cuales vas a poder meter el código al compilador y así poder cumplir con los análisis tanto como léxico como sintáctico.

Por ultimo también tenemos la función de la maquina de Turing esta funciona de la mejor manera colocando el mismo numero de 0s y de 1s como cadenas (0011,0101,1100,000111, etc.) mientras sea la misma cantidad de estos mismos la cadena será aceptada y si no cumple con que sean las mismas cantidades esta cadena no será aceptada.

2. EXPLICACION DEL LENGUAJE PERSONALIZADO(tokens,gramática,errores)

//tokens

Tokens de Operaciones Aritméticas

- suma X y Y - Suma dos números
- resta X y Y - Resta dos números
- mult X y Y - Multiplica dos números
- div X y Y - Divide dos números
- pot X y Y - Eleva X a la potencia Y
- raiz X - Calcula la raíz cuadrada de X

Tokens Auxiliares

OPERADOR → "y" (conecta dos operandos)

Tokens de Datos

NUM → [0-9]+(\\. [0-9]+)? (números enteros o decimales)
VAR → [a-zA-Z][a-zA-Z0-9]* (identificadores de variables)



Tokens Especiales

DESCONOCIDO → cualquier palabra no reconocida

// Gramática soportada

Gramática Soportada:

Operaciones: [operación] [número] y [número]

Raíz: raíz [número]

//errores

Tipos de Errores Implementados

1.- Errores Léxicos

- **Tokens Desconocidos:** Palabras no reconocidas por el analizador
- **Resaltado Visual:** Las palabras desconocidas se marcan con fondo rojo

sumar 5 y 3 ✗ "sumar" no reconocido (debería ser "suma")

2.- Errores Sintácticos

- **Estructura Incorrecta:** No sigue las reglas gramaticales
- **Parámetros Faltantes:** Operaciones sin suficientes operandos

suma 5 ✗ Falta el segundo operando
resta y 3 ✗ Primer operando no es número
mult 2 con 3 ✗ "con" no es válido (debería ser "y")

3. Ejemplos de entrada válidos e inválidos.

validos

Suma 5 y 1, resta 5 y 1, mult 5 y 1, div 6 y 2, pot 3 y 4, raiz 16

invalidos

sum 5 y 1, suma 5 , suma 3 , 4, SUMA 5 y 3, Suma 3 y 4, suma 5 + 3, suma 5y6, suma 5 y6, 5 mas 6.

rest 5 y 2, 5 menos 4, restar 5 - 1, Resta 5 y 6.



multi 5 y 6, multiplicacion 5 y 6, multiplicación 5 y 6, $5 * 6$, Mult 5 y 6, mult $5 * 6$.

dividir 8 y 2, división 4 y 2, Div 3 y 4, div 8.

potencia 3 y 4, potenciacion 3 y 4, Pot 4 y 3, pot 3.

Raiz6, Raiz de 5, rais de 10, raiz y 6.