

## Tokenización:

El proceso de tokenización es la primera etapa del funcionamiento de la calculadora, donde la expresión de entrada se divide en unidades más pequeñas llamadas *tokens*. El código en Flex se encarga de este proceso. La función principal del lexer es identificar diferentes tipos de elementos en la expresión, como números, operadores matemáticos, paréntesis, y otros, para que el parser pueda procesarlos correctamente.

- **Números:** El lexer reconoce cualquier secuencia de dígitos ( $[0-9]^+$ ) como un token NUMBER. Por ejemplo, en la expresión  $5 + 3$ , los números 5 y 3 serían tokens NUMBER.
- **Operadores:** Los símbolos  $+$ ,  $-$ ,  $*$ ,  $/$  se identifican como tokens ADD, SUB, MUL, DIV, respectivamente. Estos tokens indican al parser qué tipo de operación matemática se debe realizar.
- **Función ABS:** La cadena ABS se convierte en el token ABS, que se usa para calcular el valor absoluto de un número.
- **Paréntesis:** Los paréntesis ( y ) se convierten en los tokens LPAREN y RPAREN, que ayudan a gestionar la precedencia de las operaciones en las expresiones.
- **Espacios en Blanco:** Los espacios y tabulaciones ( $[\ \backslash t]$ ) son ignorados, ya que no afectan la interpretación de la expresión.
- **Errores:** Cualquier secuencia de letras no reconocida ( $[a-zA-Z]^+$ ) o cualquier carácter desconocido se maneja como un error, generando un mensaje adecuado.

Una vez que el lexer ha convertido toda la expresión en tokens, estos se envían al parser (implementado en Bison) para su análisis.

## Análisis:

El análisis es el proceso en el que la secuencia de tokens generada por el lexer se utiliza para construir una representación estructurada de la expresión, llamada *árbol de sintaxis*. El parser, implementado en Bison, utiliza reglas gramaticales para construir este árbol.

- **Expresiones:** El parser reconoce que una expresión puede ser un número (NUMBER), o una combinación de términos conectados por operadores de suma o resta (ADD, SUB). Por ejemplo, en la expresión  $5 + 3$ , el parser identifica 5 y 3 como términos, y  $+$  como un operador que los conecta.
- **Términos:** Un término es un componente de una expresión que puede ser un número, o una combinación de factores conectados por operadores de multiplicación o división (MUL, DIV). Por ejemplo, en  $4 * 2$ , 4 y 2 son factores, y  $*$  es el operador.

- **Factores:** Un factor puede ser un número, una expresión entre paréntesis, un número negativo, o el valor absoluto de una expresión. Por ejemplo, en  $-5$ ,  $5$  es el factor y  $-$  indica que se debe cambiar su signo.
- **Manejo de Errores:** Si el parser encuentra un error, como una expresión incompleta o una división por cero, se genera un mensaje de error. El parser tiene mecanismos para recuperarse de ciertos errores, lo que permite al usuario ingresar una nueva expresión sin que el programa se detenga.

El árbol de sintaxis resultante de este análisis refleja la estructura de la expresión original, organizando las operaciones según su precedencia y asociación.

### **Evaluación:**

Una vez que se ha construido el árbol de sintaxis, el siguiente paso es evaluarlo para calcular el resultado final de la expresión.

- **Evaluación de Operaciones:** El árbol de sintaxis se recorre para realizar las operaciones matemáticas. Cada nodo del árbol representa una operación o un valor. Por ejemplo, un nodo que representa la suma (ADD) tomará los valores de sus dos nodos hijos y los sumará.
- **Precedencia de Operaciones:** El orden en que se realizan las operaciones está determinado por la estructura del árbol. Por ejemplo, en la expresión  $3 + 4 * 2$ , la multiplicación se realiza antes que la suma, porque el nodo de multiplicación está más cerca de las hojas del árbol que el nodo de suma.
- **Funciones y Paréntesis:** Las funciones como ABS y los paréntesis modifican la estructura del árbol de modo que se respeten las reglas matemáticas correspondientes. Por ejemplo, en la expresión  $ABS(-5)$ , el nodo ABS tomará el valor de su hijo (que representa  $-5$ ) y lo convertirá en  $5$ .
- **Manejo de Errores Durante la Evaluación:** Si durante la evaluación se encuentra un error, como una división por cero, se genera un mensaje de error y se evita que el programa falle.

Finalmente, el resultado de la evaluación se imprime al usuario, y el programa queda listo para recibir una nueva expresión.