

## METODOS:

\*\*\*\*\*manejo de vista\*\*\*\*\*

### 1. Private Sub obtener\_tokens(texto As String)

En este metodo obtenemos todos los tokens de la cadena ingresada en el editor.

### 2. Private Function estado\_transicion(ByVal n As Integer) As Integer

En esta funcion es el encargado de hacer las transiciones entre estados.

### 3. Private Sub analizador()

En este método se verifica que venga una palabra reservada.

### 4. Private Sub declarar()

En este método se hacen todas las funciones para declarar variables (VAR).

### 5. Private Function operar()

En esta fucion se hacen las funciones de la palabra reservada OPERA, se reutilizan los métodos SUMA, RESTA, DIVI, MULT.

### 6. Private Function sumar() As Double

En esta funcion se hacen todas las funciones de la palabra reservada SUMA y retorna como valor double la suma de todos los atributos ingresados.

### 7. Private Function restar() As Double

En esta funcion se hacen todas las funciones de la palabra reservada RESTA y retorna como valor double la resta de los dos atributos ingresados.

### 8. Private Function multiplicar() As Double

En esta funcion se hacen todas las funciones de la palabra reservada MULT y retorna como valor double la multiplicación de todos los atributos ingresados.

### 9. Private Function divir() As Double

En esta funcion se hacen todas las funciones de la palabra reservada DIVI y retorna como valor double la division de los dos atributos ingresados.

### 10. Private Sub asignar()

En este método se hacen las asignaciones a las variables es uno a uno, es decir solo se puede asignar uno a la vez, puede asignarse un numero o bien otra variable declarada anteriormente.

### 11. Private Sub salida()

En este método sirve para poder imprimir en consola cualquier cosa que venga de la siguiente forma SALIDA("Cualquier cosa", identificador);

### 12. Private Sub comentario\_linea()

En este método se validan los comentarios de solo una línea y se agregan a la lista de comentarios

**13. Private Sub comentario\_multilinea()**

En este método se validan los comentarios de varias líneas y se agregan a la lista de comentarios

**14. Private Sub obtener\_lista()**

En este método se obtienen la lista de n cantidad de identificadores o números a la hora de hacer una suma o una multiplicación ej SUMA(var1, var2, ..., varn);

**15. Private Sub consumir ()**

En este método se incrementa la variable entera index\_global para recorrer la lista de tokens.

**16. Private Function validar\_numero\_variable()**

En esta función se valida si el valor ingresado en cualquier operación es un identificador, un entero o un decimal.

**17. Private Sub capturar\_error()**

En este método se capturan los errores léxicos y sintácticos (en proceso). Recibe el lexema y lo agrega a una lista de errores.

**18. Private Function es\_solo\_mayuscula()**

Esta función valida que sea solo mayúscula mediante un rango de la tabla ascii y retorna un valor de verdad.

**19. Private Function es\_solo\_minuscula ()**

Esta función valida que sea solo minúscula mediante un rango de la tabla ascii y retorna un valor de verdad.

**20. Private Function es\_solo\_numero()**

Esta función valida que sea solo número mediante un rango de la tabla ascii y retorna un valor de verdad.

**21. Private Function es\_parentesis\_inicio ()**

Esta función valida que sea solo paréntesis inicio mediante un index de la tabla ascii y retorna un valor de verdad.

**22. Private Function es\_parentesis\_fin ()**

Esta función valida que sea solo paréntesis fin mediante un index de la tabla ascii y retorna un valor de verdad.

**23. Private Function es\_punto\_coma()**

Esta función valida que sea solo punto y coma mediante un index de la tabla ascii y retorna un valor de verdad.

**24. Private Function es\_coma ()**

Esta función valida que sea solo coma mediante un index de la tabla ascii y retorna un valor de verdad.

**25. Private Function es\_doble\_comilla ()**

Esta función valida que sea solo doble comilla mediante un index de la tabla ascii y retorna un valor de verdad.

**26. Private Function es\_espacios ()**

Esta función valida que sean espacios(salto de línea, tab, o espacio normal) mediante un rango de la tabla ascii y retorna un valor de verdad.

**27. Private Function es\_punto ()**

Esta función valida que sea solo punto mediante un index de la tabla ascii y retorna un valor de verdad.

**28. Private Function es\_palabra\_reservada ()**

Esta función valida que sea palabra reservada comparando con el arreglo de palabras reservadas y retorna el numero de token.

**29. Private Function es\_diagonal ()**

Esta función valida que sea solo diagonal mediante un index de la tabla ascii y retorna un valor de verdad.

**30. Private Function es\_asterisco ()**

Esta función valida que sea solo asterisco mediante un index de la tabla ascii y retorna un valor de verdad.

\*\*\*\*\*manejo de vista\*\*\*\*\*

**31. Private Function crear\_ventana ()**

Esta función sirve para crear una nueva ventana con un txt dentro para poder ser analizado. El botón nueva ventana manda a llamar este método al igual que el botón de abrir.

**32. Private Function guardar ()**

Esta función despliega una ventana de donde guardar el archivo de texto en formato .lfp

**33. Private Function cerrar ()**

Esta función sirve para cerrar la ventana actual.  
En los botones cerrar todas o guardar todas se llama este método.

\*\*\*\*\*creación de html\*\*\*\*\*

**34. Function html\_resultados ()**

Esta función recorre las listas de tokens, errores y comentarios. Ya recorridos los agrega a tablas en formato html.

**35. Function generar\_archivo ()**

Esta función se despliega otra ventana donde pide la ubicación para guardar el archivo html.

Numero de token	Lexema
1	VAR
2	ASIG
3	OPERA
4	SUMA
5	RESTA
6	MULTI
7	DIVI
8	SALIDA
9	"identificador"
10	"numero entero"
11	(
12	)
13	;
14	,
15	"
16	/
17	*
18	.
19	"decimal"
20	"caracter extraño"

$[A-Z].[A-Z](0-9)^*$  |  $[0-9].[0-9](0-9)^*$   
 $(/).[0-9](*)[?].[cualquier cosa].(*)?(/)?$

