```
85          lecturaTkn += currentChar;
86          if(ex.isNumeric(currentChar) && lecturaTkn.length() == 1){
87              analyzeNumberTkn(texto);
88              saveToken(preliminarType:2, incrDone:false);
89          }
90      //cuando se interrumpe el flujo por un caracter especial
91      } else if (!ex.isIgnoredCharacter(currentChar) && lecturaTkn.length() != 0) {
92          saveToken(preliminarType:1, incrDone:true);
93          columna--;
94          index--;
95      //cuando se inicia por un caracter especial
96      } else if (!ex.isIgnoredCharacter(currentChar) && lecturaTkn.length() == 0) {
97          lecturaTkn += currentChar;
98          if(currentChar == '\"' || currentChar == '\'' || currentChar == '#'){ //cadenas y comentarios
99              readAll = true;
100         }else if(!ex.isCombinable(currentChar)){
101             saveToken(preliminarType:4, incrDone:false);
102         }else if(ex.isCombinable(currentChar)){//cuando es un caracter especial que es pueda combinar
103             analyzeCombinableTkn(texto);
104             saveToken(preliminarType:4, incrDone:false);
105         }
106     //cuando se interrumper por un caracter ignorado
107     } else if (ex.isIgnoredCharacter(currentChar) && lecturaTkn.length() != 0) {
108         saveToken(preliminarType:1, incrDone:true);
109     } //de lo contrario si se trata de otros caracteres ignorados no hace nada
110 }
111 private void analyzeCombinableTkn(String texto){
112     while (true) {
113         if ((index + 1) < texto.length()) {
114             char nextChar = texto.charAt(index + 1);
115             if (ex.isCombinable(nextChar)) {
116                 lecturaTkn += nextChar;
117                 actualizarIndex();
118             } else {
119                 break;
```

Manual de Usuario

Parser-pY

Yennifer María de León Samuc

Registro Académico No. 202231084

Lenguajes Formales y de programación, segundo semestre 2023

ÍNDICE

CONTENIDO **Pág**

Sobre la clasificación de los tokens 1

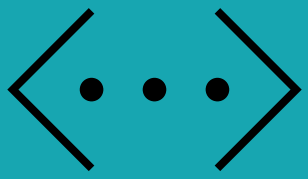
Partes de la aplicación 2

- Barra de opciones 3
- Editor 4
- Botón limpiar 4
- Botón Análisis Léxico 5
- Panel de Errores 5
- Área de Reportes 5
  - Tabla de reporte 6
  - Botón generar gráfica 6
  - Botón guardar reporte 6

Terminología 7

A continuación se muestra una tabla resumida del proceso de clasificación de los tokens del programa (para más información al respecto del programa, véase el manual técnico).

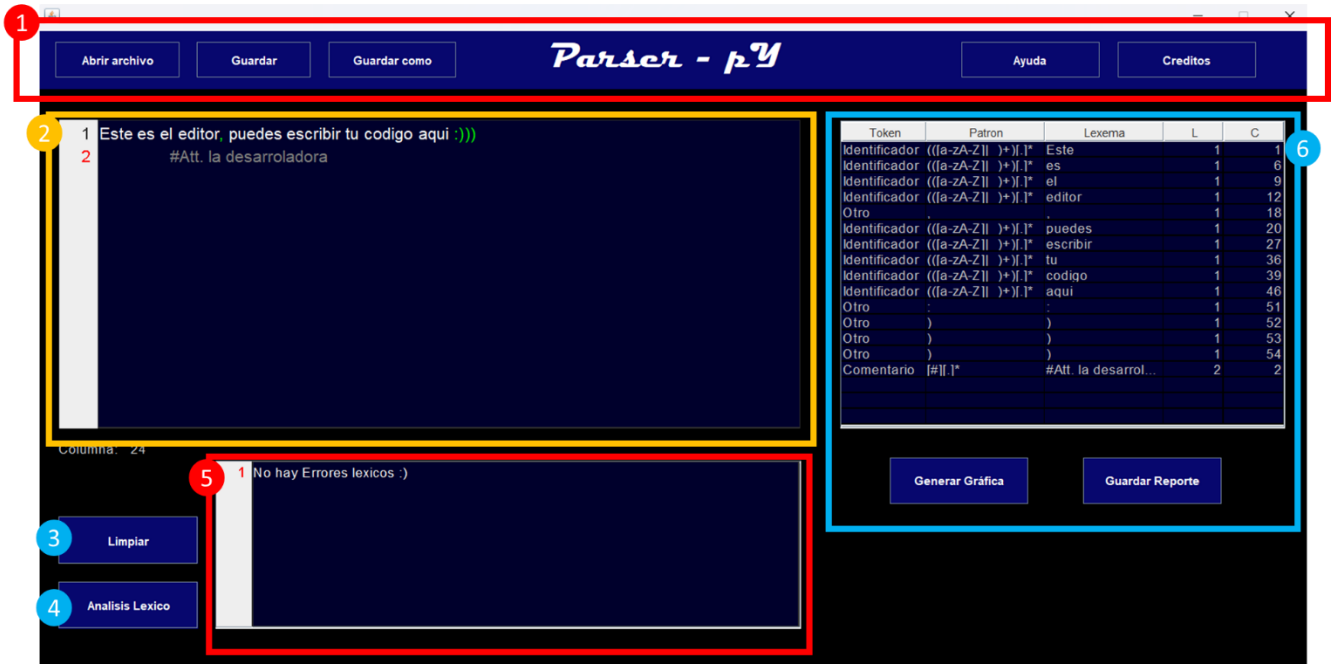
Esta incluye la categoría del token que se detecta, el patrón (mediante expresiones regulares o una explicación con palabras) y algunas observaciones u ejemplos.



SOBRE LA CLASIFICACIÓN DE TOKENS

Token	Patrón	Observaciones
Aritméticos	Signos aritméticos predefinidos	Por ejemplo: Suma (+), resta (-).
Comparación	Signos comparativos predefinidos	Por ejemplo: Diferente(!=)
Lógicos	Palabras con patrón lógico predefinidos	Son: and, or, not
Asignación	= [Aritmético][=]	
Palabras clave	Palabras clave predefinidas	El patrón debe coincidir exactamente
Constantes **-Entero** **-Decimal** **-Cadena** **-booleanas**	 - [\d]+ - [\d]+[“.”]+[\d]+ - [“][.]*[“] - booleanas predefinidas	 [/d] = digito [“.”]= un punto [.] = cualquier carácter Cerradura + Indica que al menos se debe ingresar un carácter de esa categoría Cerradura * Indica que se puede ingresar uno o más caracteres de esa categoría, pero se puede no ingresar nada
Comentario	[#][.]*	

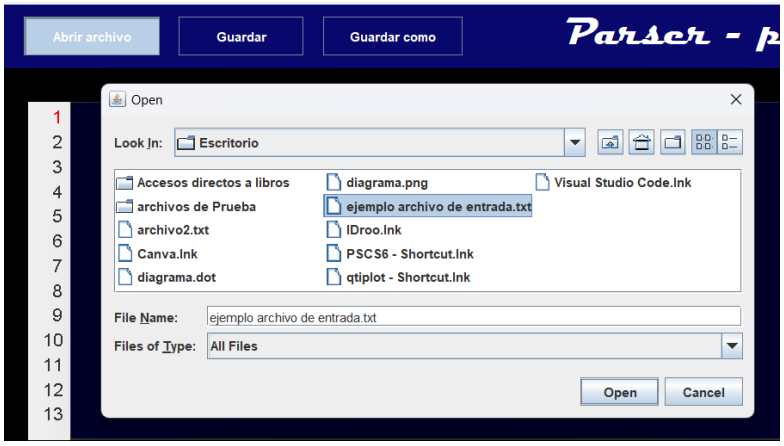
PARTES DE LA APLICACIÓN



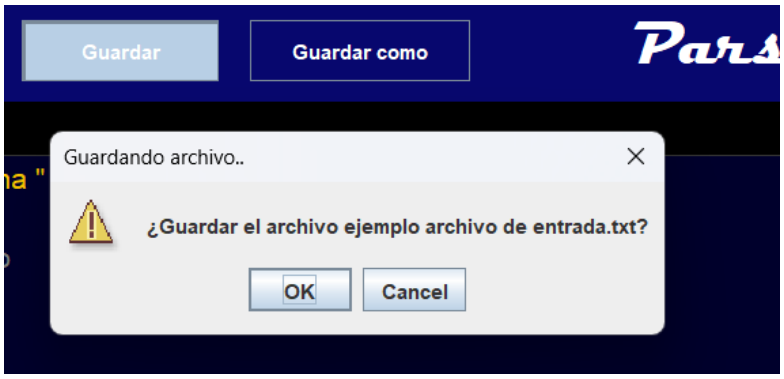
1. **Barra de opciones:** contiene opciones útiles.
2. **Editor:** área en donde se escribe el código, este cuenta con una visualización preliminar de los tokens, coloreandolos para una más comoda visualización del código.
3. **Boton limpiar:** limpia el editor, el panel de errores y la tabla de reporte (véase el área de reporte).
4. **Boton Análisis Léxico:** ejecuta el análisis léxico.
5. **Panel de errores:** al realizar el análisis léxico, en este panel se muestran los errores si se encontraran.
6. **Área de reportes:** contiene opciones útiles para visualizar y guardar reportes de análisis léxicos.

Barra de Opciones

1. **Botón Abrir Archivo:** permite abrir un archivo que puede contener código, este se mostrará en el editor para poder simplemente visualizarlo o editarlo.



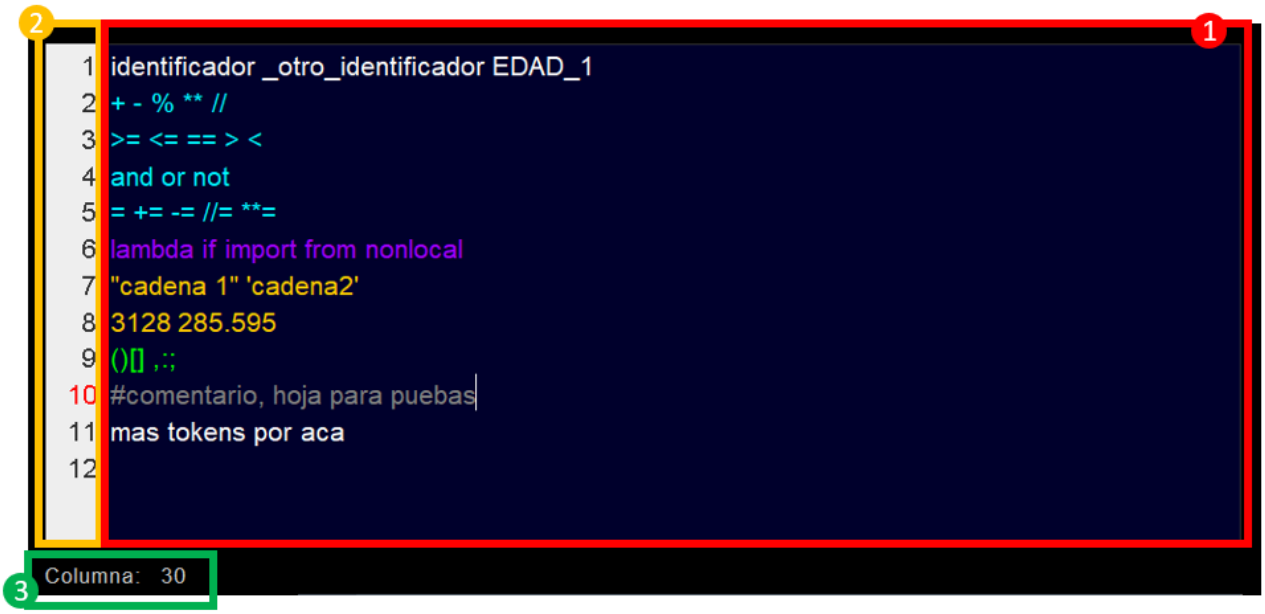
2. **Botón Guardar:** permite guardar el código del editor en un archivo previamente abierto, de lo contrario, permite crear uno nuevo.



3. **Botón Guardar Como:** permite crear un nuevo documento que contendrá el código del editor.

4. **Botón Ayuda:** muestra un enlace a esta guía.
5. **Botón Créditos:** muestra los créditos de la realización de este proyecto.

Editor



Área en donde se puede escribir el código **(1)**, dependiendo del tipo de token se coloreará para una mejor visualización, esto es según la siguiente tabla (para más detalles véase la sección “Sobre la clasifiación de los tokens”):

TOKEN	*COLOR*
Identificadores	Blanco
Operadores Aritmeticos, comparativos, lógicos, asignación	Celeste
Palabras clave o reservadas	Morado
Constantes(numeros, cadenas, booleanos)	Anaranjado
Comentarios	Gris
Otros (corchetes, paréntesis, etc.)	Verde
Errores (carácter irreconocible, cadena que no cierra)	Rojo

Se muestra además una sencilla visualización de la línea **(2)** y la columna **(3)** en donde se encuentra el puntero.

Botón Limpiar

Limpia el editor, el panel de errores y la tabla de reporte.

Botón Análisis Léxico

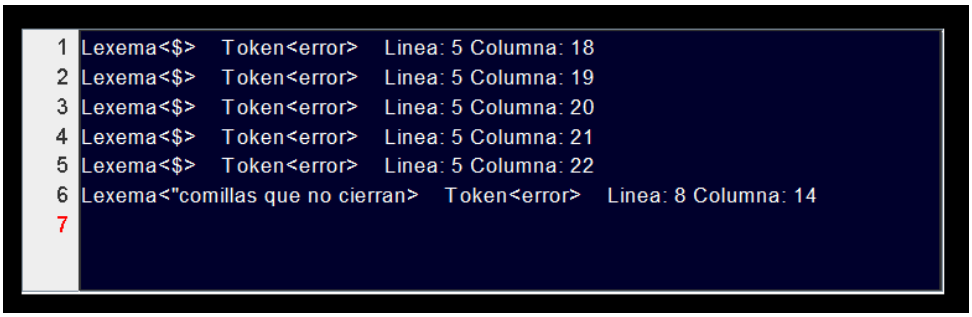
Ejecuta el análisis léxico del texto contenido en el editor; el análisis incluye mostrar los errores en el panel de errores, mostrar un reporte en la tabla de reporte.

La siguiente captura muestra un análisis léxico realizado correctamente:

Panel de errores

Al realizar el análisis léxico, en este panel se muestran los errores si se encontraran.

Se muestran algunos errores léxicos detectados, de estos se especifican la línea y columna en donde se encuentran para su mejor ubicación en el editor:



Nota: *si no hubiera ningún error léxico, en este apartado se especifica que no se encontraron errores*

Área de Reportes

1. **Tabla de reporte:** al realizar el análisis léxico, en este apartado se muestran los tokens encontrados, así como información útil sobre los mismos.
2. **Botón Generar Gráfica:** genera la gráfica de un toquen previamente seleccionado.
3. **Botón Guardar Reporte:** realiza un análisis léxico y guarda el reporte del mismo en un documento de texto.

→Tabla de reporte

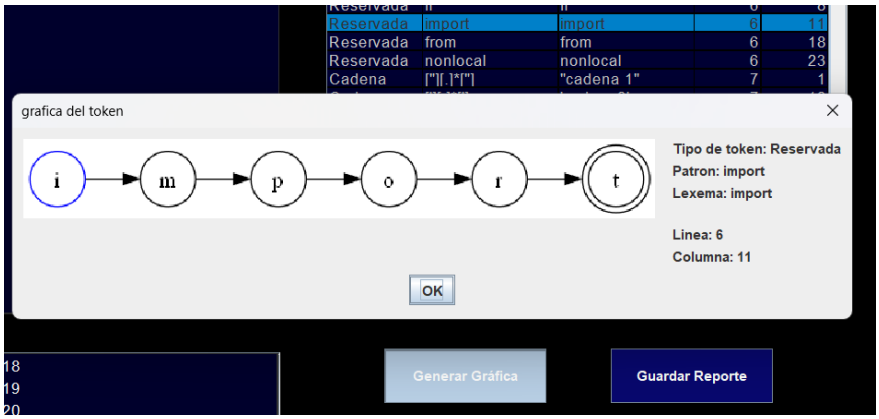
Al realizar el análisis léxico, en este apartado se muestra la lista de tokens analizados, mostrando la siguiente información:

1. **Type:** qué tipo de token es, a que categoría pertenece.
2. **Token:** el tipo de token especifico.
3. **Patrón:** representado con expresiones regulares, muestra el patrón que cumplió el lexema para ser considerado de algún tipo de token en específico
4. **Lexema:** el contenido del token.
5. **L:** abreviatura para Linea, muestra el numero de linea en donde se encuentra el token analizado.
6. **C:** abreviatura para Columna, muestra el numero de la columna en donde inicial el token analizado

→Botón "Generar gráfica"

Genera la gráfica de un toquen previamente seleccionado.

Para seleccionar un token se debe de hacer clic en la tabla de reporte, se iluminará la celda seleccionada, posteriormente se debe de dar clic a este botón, y nos aparecerá la gráfica del token seleccionado.



El circulo pintado de azul (también puede ser más de uno), representa el inicio del token

El circulo con un margen doble (puede ser más de uno) representa el fin de un token

Notas:

- ✓ *Si se selecciona más de una fila, al presionar este botón se graficará unicamente el primer token seleccionado.*
- ✓ *No se generarán gráficas de los errores léxicos.*

→Botón "Guardar Reporte"

Realiza un análisis léxico y guarda la información presentada en la Tabla de reporte en un documento de texto que podemos nombrar, se muestra un ejemplo de una parte de un reporte:

```
Lexema<identificador>    Token<Identificador>    Linea: 1 Columna: 1
Lexema<_otro_identificador>    Token<Identificador>    Linea: 1 Columna: 15
Lexema<EDAD_1>    Token<Identificador>    Linea: 1 Columna: 35
```

TERMINOLOGÍA

- **Analisis Léxico:** Es la primera fase de un compilador, toma el programa fuente de los pre-procesadores que está escrito en forma de declaraciones. Este proceso desglosa el código en una serie de tokens, deshaciéndose primero de todos los comentarios en el código y los espacios en blanco.

- **Analizador Léxico:** programa que realiza un análisis léxico.

- **Errores léxicos:** Son aquellos que se detectan cuando el analizador léxico intenta reconocer componentes léxicos y la cadena de caracteres de la entrada no encaja con ningún patrón. Algunos ejemplos son:
  - Se escribe un carácter inválido
  - No se cierran comillas

- **Expresión Regular**: son patrones utilizados para encontrar una determinada combinación de caracteres dentro de una cadena de texto. Las expresiones regulares proporcionan una manera muy flexible de buscar o reconocer cadenas de texto.

- **Lexema:** secuencia de caracteres en el programa fuente, que coinciden con el patrón para un token y que el analizador léxico identifica como una instancia de ese token.

- **Patrón:** sucesión de elementos que se construyen siguiendo una regla.

- **Token:** El token es una referencia (un identificador) que regresa a los datos sensibles a través de un sistema de tokenización.