

Manual de Usuario

Analizador léxico y sintáctico

Descripción

Sistema desarrollado para el análisis léxico y sintáctico de un lenguaje formal. Consta de dos componentes principales: un **analizador léxico (lexer)**, encargado de identificar y clasificar los tokens presentes en la entrada, y un **analizador sintáctico predictivo LL(1)**, que valida la estructura gramatical de la cadena conforme a una gramática libre de contexto.

Requerimientos de software

Openjdk 21 o JDK 21

Instalación

Ejecutar el archivo analizador.jar desde la interfaz grafica, o desde la terminal usando 'java -jar rutaAlArchivo/analizador.jar'

Funciones

Funciones del editor

Abrir

Puedes abrir un archivo de texto y visualizar su contenido en el área de edición. Si realizaste cambios sin guardar, el sistema te pedirá confirmación para continuar, ofreciéndote la opción de guardar o descartar los cambios antes de proceder.

Guardar

Puedes guardar los cambios realizados en el mismo archivo. Si no has abierto un archivo previamente, se mostrará un cuadro de diálogo para guardar el contenido actual en una ubicación de tu preferencia.

Guardar como

Tienes la posibilidad de guardar el contenido en un nuevo archivo, eligiendo la ubicación y el nombre a través de un cuadro de diálogo.

Nuevo

Al crear un nuevo archivo, si hay cambios sin guardar, se te pedirá confirmación para guardar o desechar los cambios antes de comenzar con un documento en blanco.

Copiar / Pegar

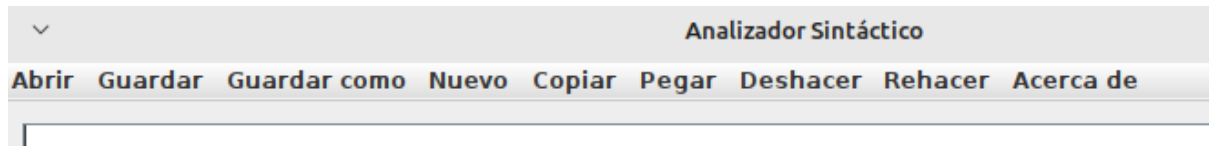
Puedes seleccionar el texto deseado para copiarlo. También puedes pegar texto en cualquier parte del editor.

Deshacer / Rehacer

Cuentas con las funciones clásicas de deshacer y rehacer cambios, similares a cualquier editor de texto. Esto te permite corregir o repetir acciones recientes fácilmente.

Acerca de

Puedes acceder a una ventana que muestra información sobre el autor de la aplicación, incluyendo nombre y otros detalles relevantes del estudiante que la desarrolló.



Reglas léxicas

Números Enteros: Se aceptan números enteros con o sin signo. No se aceptan ceros a la izquierda, salvo que el número sea 0.

Identificadores: Todo identificador debe comenzar con el símbolo \$, seguido de letras, dígitos, guiones medios (-) o guiones bajos (_).

Palabras Reservadas: Son palabras con un significado especial en el lenguaje. Estas no pueden usarse como identificadores.

PRINT, END, REPEAT, INIT, IF, TRUE, FALSE, THEN

Literales (Cadenas de texto): Se pueden utilizar cadenas de texto encerradas entre comillas dobles ("). No se permiten saltos de línea dentro de las comillas.

Operadores Aritméticos: Los operadores disponibles son +, -, *, ^, y /. También puedes usar paréntesis para modificar el orden de evaluación.

Operador de Asignación: El operador de asignación es el símbolo "=".

Comentario de una sola línea: Comienza con "#" y termina al final de la línea.

Comentarios en bloque: Comienzan con "/*" y terminan con "*/". Pueden incluir varias líneas.

Estructuras sintácticas aceptadas

Estructura PRINT:

Comienza con la palabra reservada **PRINT**.

Sigue un **literal**, **número** o **identificador**.

Termina con la palabra reservada **END**.

Estructura REPEAT:

Comienza con la palabra reservada **REPEAT**.

Sigue un **número entero positivo** o un **identificador**.

Continúa con la palabra reservada **INIT**.

Después puede tener **cero o muchas** estructuras PRINT.

Termina con la palabra reservada **END**.

Estructura CONDICIONAL:

Comienza con la palabra reservada **IF**.

Sigue una de las palabras reservadas **TRUE** o **FALSE**.

Luego viene la palabra reservada **THEN**.

Puede tener **cero o una** estructura PRINT.

Termina con la palabra reservada **END**.

Estructura EXPRESIÓN:

Compuesta por **operaciones aritméticas** entre números enteros e identificadores.

Se puede utilizar **agrupación con paréntesis**.

Estructura ASIGNACIÓN:

Comienza con un **identificador**.

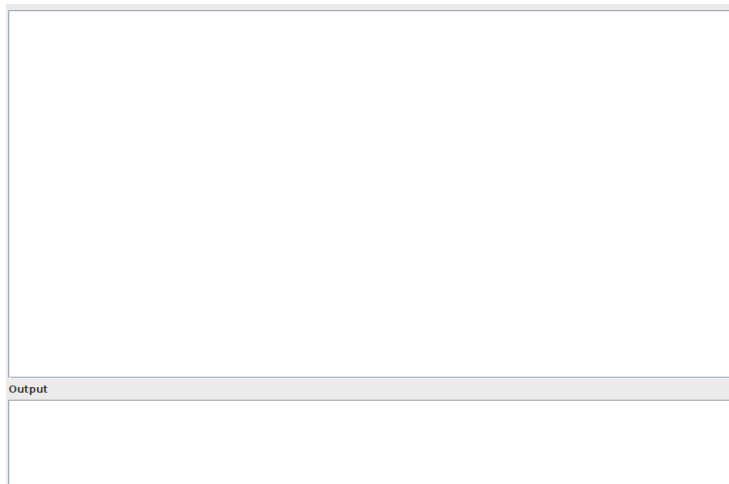
Sigue un signo **igual (=)**.

Luego una **estructura EXPRESIÓN**.

Termina con la palabra reservada **END**.

Entrada y salida

Áreas designadas para ingresar texto y ver los errores generados



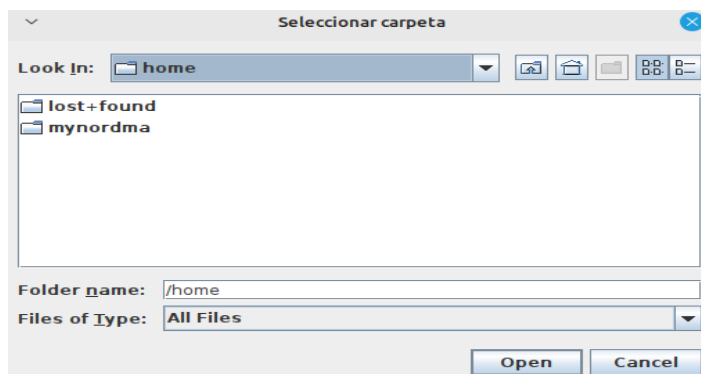
Tokens

Permite visualizar los tokens generados por el lexer media vez no haya errores.

Analizar

Analiza los tokens y genera un árbol sintáctico, y si hay errores, los muestra en el apartado output.

Se tiene la opción de guardar el archivo en donde se imprimirán las estructuras print.



Árbol Sintáctico

Muestra el árbol generado por el parser, media vez no hayan errores.

