

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Auxiliar: Maynor Pilo Tuy

Organización de Lenguajes y Compiladores 1

Proyecto 1 Exregan

Lesther Kevin Federico López Miculax

Carnet: 202110897

23/03/2023

DATOS DE DESARROLLADOR

Nombre del desarrollador: Lesther Kevin Federico López Miculax

Correo electrónico: lestherlopez64@gmail.com

Nacionalidad: guatemalteca

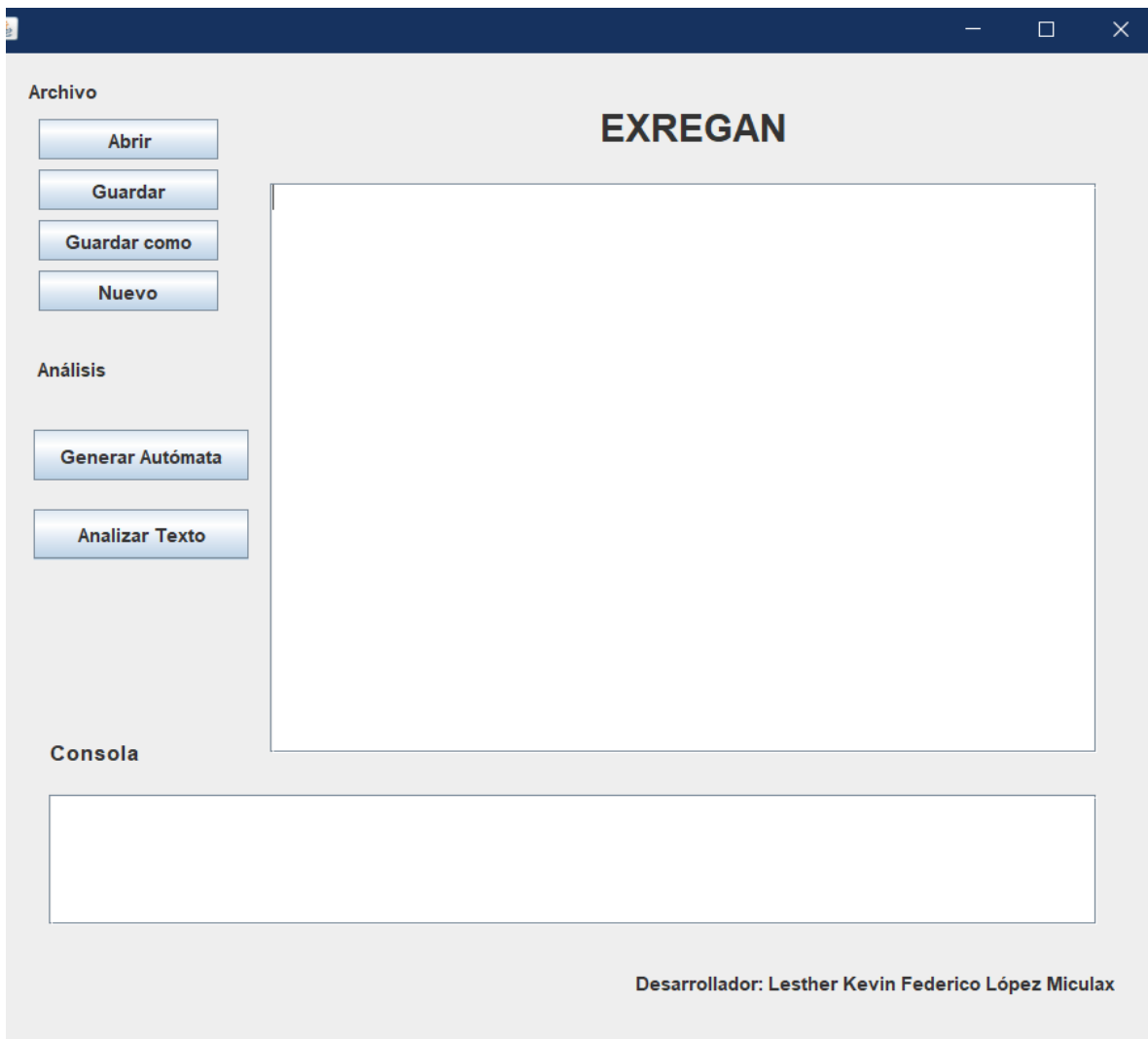
Carné universitario: 202110897

DESCRIPCION GENERAL

El proyecto 1, denominado Exregan es una aplicación altamente funcional que tiene como objetivo general el utilizar los diversos conocimientos acerca del proceso de análisis léxico y sintáctico que tiene un compilador para desarrollar el software de Exregan capaz de producir análisis utilizando la técnica del árbol, de este modo, para brindar una descripción general es importante dar como síntesis que el programa o proyecto es un analizador léxico y Sintáctico, que es capaz de brindar a los estudiantes una comprobación de lo que es el Método del Árbol y Método de Thompson de las expresiones que permite o acepta este lenguaje, con el propósito de apoyar y brindar ayuda para hojas de trabajo o tareas relacionadas a este trabajo, además, otro dato interesante de este proyecto es que las expresiones regulares pueden ingresar en notación polaca o prefija. Por consiguiente, se mencionara a detalles que opciones y funcionalidades tiene el programa y como funciona para que el usuario pueda obtener una mejor percepción de exregan, ya que es importante conocer a profundidad lo que conlleva para una mejor experiencia de usuario. Además, es de vitalidad mencionar que el programa no requiere de mayores recursos necesarios, por lo que logra ser funcional en una gran cantidad de computadoras y funciona de manera correcta sin tantas complicaciones.

MENU PRINCIPAL

El menú del programa es una interfaz gráfica que permite a los usuarios acceder a las diferentes funciones y opciones disponibles en el programa, de este modo, cuenta con varias opciones las cuales son funcionales y de gran apoyo, por consiguiente, el contenido del menú puede variar dependiendo del software y su finalidad, pero en general puede incluir opciones como guardar, abrir, entre otros, de esta manera, es adecuado especificar que dentro de este menú esta la existencia de varios botones, los cuales cada uno tiene una función en específico las cuales se explicaran a continuación de manera más detallada.



1. Sección Archivo: sector que tiene opciones para abrir, guardar, guardar como y nuevo archivo.

1.1. Nuevo archivo: Esta opción del menú permite al usuario la apertura de un nuevo archivo con la extensión OLC, siendo posible que el usuario elija la ruta donde desea guardar el archivo y el nombre que le desea colocar.

1.2. Abrir archivo: en esta opción el usuario es capaz de puede navegar por el sistema de archivos de su ordenador y seleccionar el archivo que desea abrir. Por consiguiente, una vez que se selecciona el archivo, se tiene que verificar que sea de la extensión permitida la cual es OLC y el programa lo cargará en la ventana del programa, específicamente lo mostrará en la caja de texto ubicada en la interfaz gráfica y el usuario podrá trabajar en él según sea necesario.

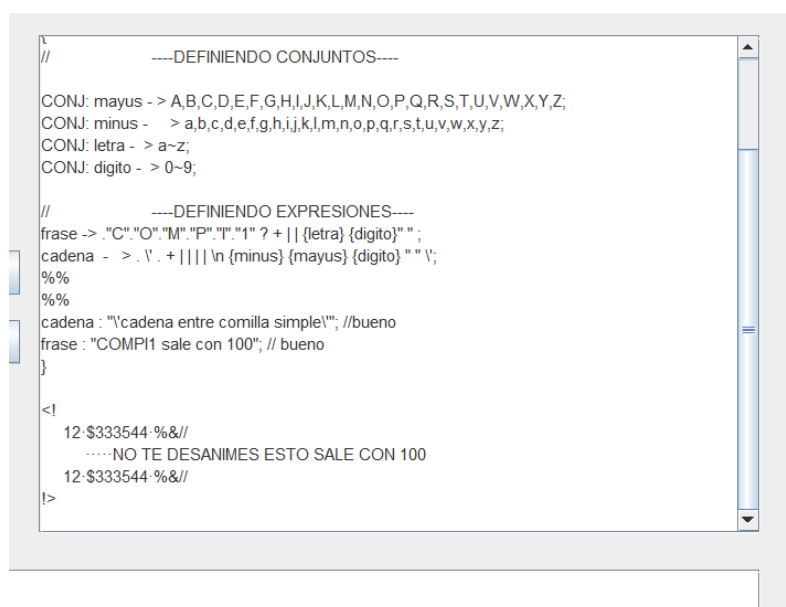
1.3. Guardar: esta opción permite al usuario guardar los cambios realizados en un archivo existente, por lo que es una herramienta funcional que asegura que los cambios más recientes se guarden y se conserven en el archivo, esta opción no genera un archivo, sino que reescribe el documento abierto previamente

1.4. Guardar como: permite al usuario guardar una copia de un archivo con un nombre diferente o en una ubicación diferente en el disco duro u otro dispositivo de almacenamiento, por consiguiente, el usuario puede especificar un nombre diferente para el archivo siempre manteniendo la extensión OLC, así como elegir una carpeta o ubicación diferente en la que guardar el archivo



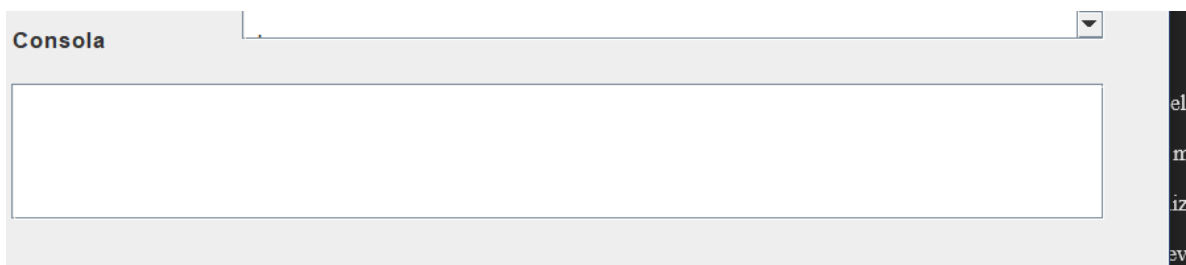
CAJA DE TEXTO

Como se menciona anteriormente, los archivos abiertos mostraran su contenido en una caja de texto ubicada en la interfaz gráfica, esta es una parte fundamental que se caracteriza por ser una zona rectangular en donde podrá escribir y realizar los cambios necesarios o deseados en archivos para posteriormente guardarlos. Por último, es importante mencionar que esta caja de texto contiene un scroll en donde se podrá bajar y subir, sin embargo, esta opción es funcional únicamente cuando el contenido escrito en ella es demasiado amplio.



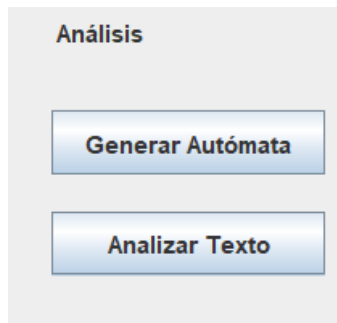
CONSOLA

La interfaz gráfica del proyecto también contiene una consola, la cual es una interfaz en la que se puede observar la salida correspondiente, luego del proceso de evaluación de cadena, es decir, en este apartado se mostrara el resultado final, de este modo, se convierte en una parte esencial de la interfaz. En este apartado también se podrá observar si existe algún error dentro del análisis y se mostrará un mensaje de error.



ANALIZAR ENTRADA

El botón analizar entrada es el alma de este programa y la parte mas fundamental, siendo este el botón que contiene la mayor parte funcional del programa, de este modo, se puede especificar que este botón realiza una serie de comprobaciones y procesos en el texto que ha sido ingresado o escrito previamente en la caja de texto, realizando un análisis sintáctico y léxico, por consiguiente, si este proceso del análisis resulta correcto sin ningún tipo de error se procederá a crear los diagramas y tablas que se explicaran a continuación de lo contrario, de existir algún tipo de error en la entrada no se ejecutara nada y mostrara error, por lo que el análisis fracasara de manera rotunda.



ARCHIVO DE ENTRADA

1. Extensión

El programa tiene ciertas restricciones y especificaciones que es importante hacer mención en este manual de usuario y, uno de estos aspectos a tomar en cuenta es que el proyecto únicamente recibirá archivos con la extensión OLC, por lo que si en dado caso se metiera un documento con otra extensión como html, csv, py, entre otros, el programa no lo ejecutara y se dará un error evitando seguir con los otras funciones del programa.

2. Estructura.

- a. Comentario multilínea: El archivo de entrada será aceptado si los comentarios multilíneas inician con “<!” y culmina con “!>”, de lo contrario se dará un error léxico.

A screenshot of a code editor with a light gray background. It shows a multilined comment in green text. The comment starts with '<!' on the first line, followed by 'Este es un comentario' on the second line, 'en nuestro programa' on the third line, and ends with '!>' on the fourth line.

- b. Comentario simple: el documento de entrada podrá tener comentarios de una sola línea siempre y cuando inicien con doble slash, es decir, “//”.

A screenshot of a code editor with a light gray background. It shows a single-line comment in gray text that starts with '////' followed by the word 'CONJUNTOS'.

- c. Declaración de conjuntos: los conjuntos se declaran y se aceptan de manera que al inicio venga la palabra reservada del lenguaje “CONJ”, posterior a eso el nombre del conjunto, luego un guion, signo de mayor, la notación y finalmente un punto y coma.

```
CONJ: digito -> 0~9;
```

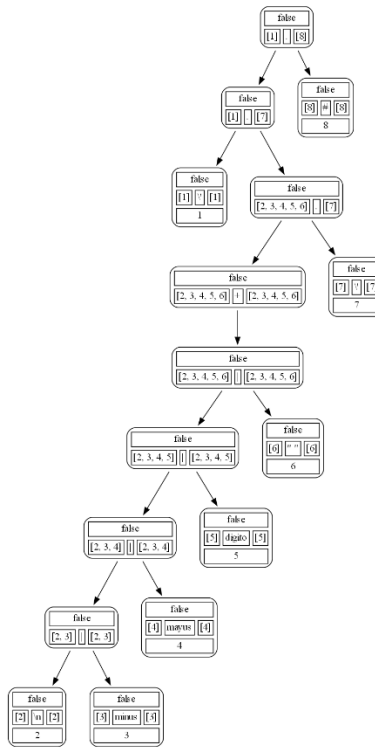
- d. Declaración expresiones regulares: Por otro lado, el programa también aceptará expresiones regulares siempre y cuando venga con una estructura en donde preceda el identificador de la expresión regular, guion, signo de mayor, expresiones regulares y finalmente el punto y coma.

```
ExpReg1 -> . {letra} * | "_" | {letra} {digito};
```

ARCHIVOS DE SALIDA O REPORTES

El programa luego de realizar el análisis de la entrada generara una diversa cantidad de reportes dependiendo la cantidad de expresiones ingresadas, estos reportes varían y tienen diferentes componentes, estructuras y diseños, los cuales se expandirá la información de estos a continuación.

1. Salida de árbol: el programa tiene la lógica necesaria para realizar un árbol binario, el cual contiene nodos con sus respectivos hijos, además cada hoja contiene anulabilidad, la cual depende de los componentes, también se puede observar en general el carácter del nodo, los primeros y últimos de cada nodo. Por consiguiente, toda esta información será plasmada en una imagen con la extensión JPG y estará almacenada en la carpeta de ARBOLES_20211089



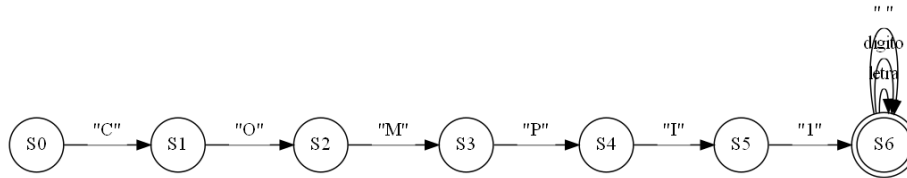
2. Salida de tabla de siguientes: Por otro lado, la capacidad del programa también se permite generar otro archivo JPG que contendrá dentro de ello la tabla de siguientes del árbol analizado, el cual es una parte fundamental del análisis sintáctico ,de este modo, se podrá observar en la tabla el carácter, numero de hoja y los siguientes de los nodos, siendo esta información plasmada en una tabla para que se observe de manera legible.

Hoja		Siguientes
"C"	1	[2]
"O"	2	[3]
"M"	3	[4]
"P"	4	[5]
"T"	5	[6]
"1"	6	[7, 8, 9, 10]
letra	7	[7, 8, 9, 10]
digito	8	[7, 8, 9, 10]
" "	9	[7, 8, 9, 10]
#	10	-

3. Salida de tabla de transiciones: en base a el árbol binario y la tabla de siguientes también se obtendrá otra tabla, la cual es la tabla de transiciones, siendo esta una que nos brinda el comportamiento del automada finito determinista, por lo que dentro de la tabla que se plasmara en un archivo JPG se verá plasmada los estados y terminales correspondientes al AFD. La imagen producida se mostrará en la carpeta TRANSICIONES_202110897

Estado	Terminales					
	\'	\n	minus	mayus	digito	" "
S0[1]	S1	-	-	-	-	-
S1[2, 3, 4, 5, 6]	-	S2	S2	S2	S2	S2
S2[2, 3, 4, 5, 6, 7]	S3	S2	S2	S2	S2	S2
S3[8]	-	-	-	-	-	-

4. Salida de AFD: En otra instancia, el proyecto creará un AFD o autómeta finito determinista, este elemento estará inspirado en la tabla de transiciones que nos dará una mejor perspectiva del comportamiento del AFD, por lo que luego de producirse se mostrará en una imagen de tipo JPG dentro de la carpeta AFD_202110897



5. SALIDAS: existirá otra carpeta denominada SALIDAS_202110897, donde se almacenara archivos de extensión JSON que contendrán la información precisa y necesaria para conocer si la evaluación de las entradas fue correcta o incorrecta
6. Salida de ERRORES: Finalmente, si existiera algún tipo de inconveniente que evite el analisis lexico del archivo de entrada, se creara un archivo con la extensión HTML, donde vendrá dentro de ella una tabla que tendrá la información necesaria para conocer de donde proviene el error, esta información es: No. Error, tipo de error, descripción, línea y columna del error, de este modo, resulta efectivo este tipo de reporte para corregir el error con tiempo.

Errores				
#	Tipo de error	Descripcion	Linea	Columna
1	Léxico	No se esperaba el componente /	4	4