

Universidad de San Carlos de Guatemala -USAC-

Facultad de Ingeniería

Ingeniería en Ciencias y Sistemas

Ing. Manuel Castillo

Aux. Kevin López

Organización de Lenguajes y Compiladores 1

**PROYECTO No. 1
ExRegan USAC
(Manual de Usuario)**

Robin Omar Buezo Díaz

Carné 201944994

Sábado 18 de marzo de 2023.

El presente documento tiene como finalidad mostrar al usuario la funcionalidad del software y guiarlo durante el uso de este para que pueda usarlo de una forma correcta y obtener el mayor beneficio de este.

Se explican el flujo y las diferentes opciones que como usuario podrá encontrar en el programa, en cada ventana se dará una guía de cada componente que se incluya y como debemos de interactuar con este para que el sistema nos sea de gran ayuda.

DESCRIPCIÓN Y ALCANCE DEL SOFTWARE

La finalidad de este software es poder ayudar al usuario a revisar los AFD y AFND en conjunto con sus agregados, de las expresiones regulares que el usuario desee. Esto con la ayuda de un lenguaje que se construyó para que el usuario puede allí ingresar sus expresiones regulares y las cadenas a evaluar para que el sistema pueda retornar las diferentes respuestas siempre y cuando el código ingresado cumpla con las reglas del lenguaje, de igual forma este se encargará de guiar al usuario con el encuentro de los diferentes errores que se puedan cometer.

El sistema funcionará casi como un editor de texto en donde podremos desde cargar un archivo, editarlo, guardarlo o bien crearlo desde cero desde la misma aplicación, luego tendremos las opciones de poder generar los autómatas correspondientes a nuestras expresiones y analizarlas para que el sistema determine si existen errores o no y al mismo tiempo poder realizar las operaciones y reportes solicitados.

Dado lo anterior podemos determinar que el alcance del software está delimitado para uso principalmente de las personas que necesiten manipular archivos del lenguaje propio que se está evaluando.

CARACTERÍSTICAS PRINCIPALES

- Poder generar tanto el AFND y el AFD con sus archivos necesarios para cada una de las expresiones regulares ingresadas al sistema.
- Realizar el estudio de si las cadenas ingresadas para evaluar con las diferentes expresiones regulares coinciden o no con las reglas que esta expresión regular tiene.
- Poder permitir al usuario la facilidad de modificar, crear y analizar archivos en dicho lenguaje desde la misma aplicación y poder mostrar los errores que se tengan.

FUNCIONALIDADES

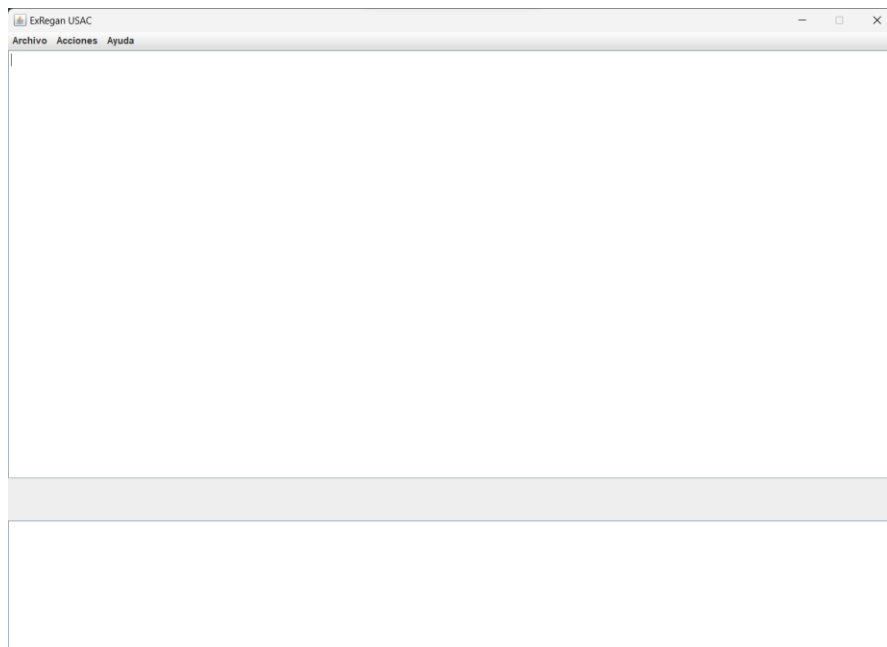
El software necesitará que ya sea que le carguemos un archivo con las expresiones a las cuales se les van a generar los autómatas y las cadenas a analizar o bien crearlo desde el editor que el programa tiene para luego poder realizar estas acciones.

Teniendo los datos necesarios entonces el sistema podrá analizar si nuestras indicaciones están desarrolladas de forma correcta y si es así procederá a realizar las instrucciones, de lo contrario también podremos observar cuales son los errores que hemos cometido.

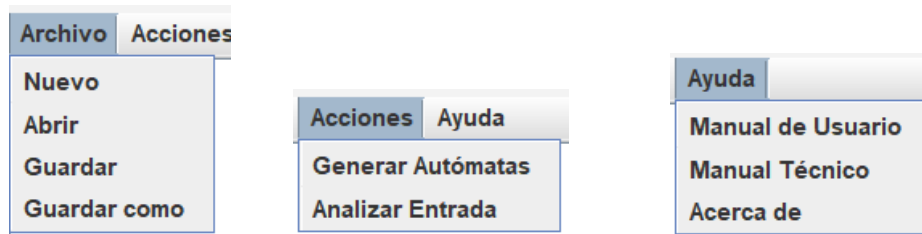
Si no tenemos errores y nuestras instrucciones se realizaron de forma correcta podremos observar los archivos finales con nuestros autómatas y así mismo un archivo JSON que nos mostrará el resultado del estudio de las cadenas con las expresiones regulares.

USABILIDAD Y NAVEGABILIDAD

El primer paso para el uso de nuestro software será tener listo nuestro archivo a cargar o bien podremos editarlo desde el editor que nos muestra la siguiente pantalla.



En esta pantalla podemos observar nuestro editor de texto, una pequeña consola en la parte inferior que nos guiará mediante mensajes y también una cinta de opciones en donde vamos a encontrar las siguientes opciones.



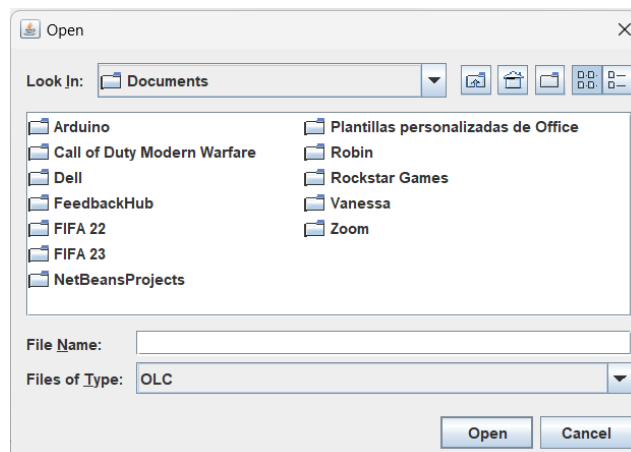
Nuevo

Esta opción nos limpiará nuestro editor para poder empezar con un nuevo archivo, antes de esto el sistema nos preguntará si queremos guardar los datos que ya tenemos en el editor.



Abrir

Esta opción nos abrirá una ventana emergente desde donde podremos buscar y seleccionar el archivo que queremos cargar (extensión .OLC).

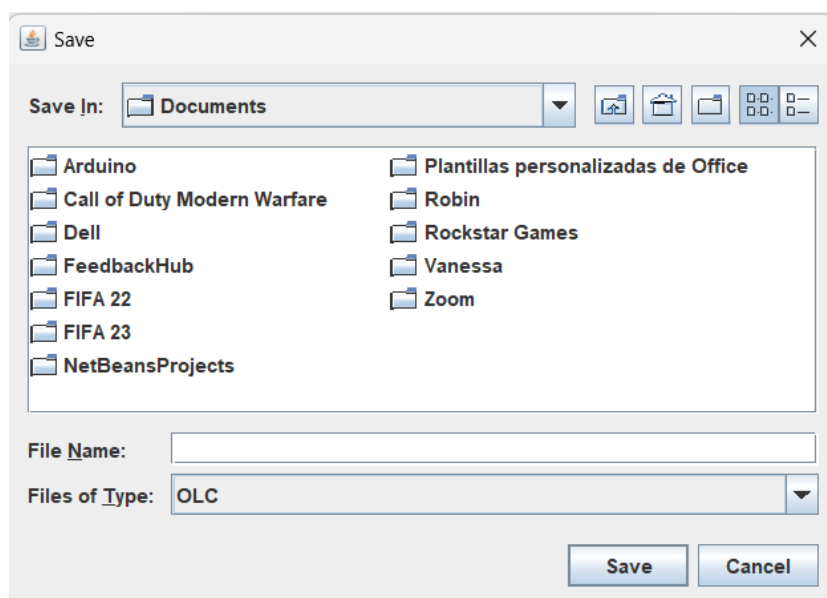


Guardar

Este botón tendrá la funcionalidad de sobrescribir el archivo que estemos editando con los cambios que le hayamos realizado, de lo contrario, si aún no tenemos ningún archivo abierto entonces funcionará como el botón de “Guardar Como”.

Guardar Como

Este botón nos permite guardar nuestro archivo, pero creando otro archivo sin modificar el archivo cargado originalmente. Para ello también tendremos una ventana emergente.



Generar Autómatas

Este botón será el encargado de hacer que nuestro programa pueda realizar el estudio de nuestro código y si tenemos errores el sistema nos lo mostrará a través de la consola y nos creará un archivo HTML en la carpeta ERRORES_201944994 que nos mostrará nuestros errores.

¡Análisis finalizado!
ERRORES EN EL ANÁLISIS

Tabla de Errores

#	Tipo de Error	Descripción	Línea	Columna
1	Léxico	El carácter "~" no pertenece al lenguaje.	6	2
2	Léxico	El carácter "●" no pertenece al lenguaje.	19	32
3	Léxico	El carácter "Z" no pertenece al lenguaje.	30	2

en caso de que el código ingresado sea correcto automáticamente generará los reportes del Árbol, Tabla de Siguietes, Tabla de Transiciones, AFD, AFND cada uno en su respectiva carpeta y la consola únicamente nos mostrará el mensaje de “¡Análisis finalizado!”.

```

ExRegan USAC - pruebaf.olg
Archivo Acciones Ayuda

{
<!
    #####Este es el archiv facil para el proyecto 1#####
    NOTA: no se permite modificar
>

//AGREGANDO CONJUNTOS
CONJ: abecedarioMinus -> a-z; // declarando conjunto de letras desde a hasta z en minusculas
CONJ: digito -> 0-5; // creamos el conjunto de digitos solo para 0, 1, 2, 3, 4 y 5

//AGREGANDO EXPRESIONES REGULARES
identificador -> . {abecedarioMinus} * | "_" | {abecedarioMinus} {digito}; //INFLUJO: abecedarioMinus ("_"abecedarioMinus{digito})
decimales -> . +{digito} . "." + {digito}; //INFLUJO: digito+ "." digito+

%%
%%
//ESTAS SON LAS PRUEBAS PARA LAS EXPRESIONES REGULARES
identificador : "soy_un_id_1"; //correcto
identificador : "ComPi1"; //incorrecto
decimales : "301.59"; //incorrecto
decimales : "1505.55"; //correcto
}

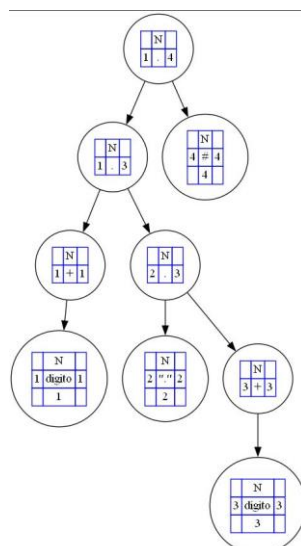
<! YA s4llo cOMPI 1
    sale con 100
    el proyecto 1 #4&$2=?
    #####RESULTADO SATISFACTORIO PARA ESTE COMENTARIO#####
>

¡Análisis finalizado!

```

Este es un ejemplo de los archivos generados.

- **Árbol**



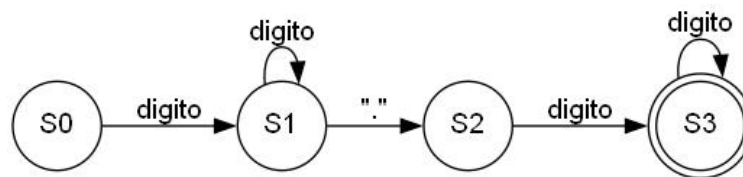
- **Siguientes**

Nodo	Valor	Siguiente
1	digito	[1, 2]
2	"."	[3]
3	digito	[3, 4]

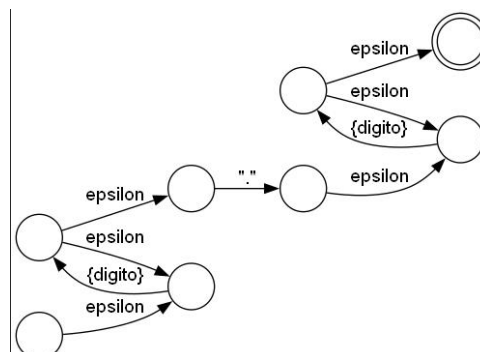
- **Transiciones**

Estado	Nodo	Terminal	Siguiente
S0[1]	1	digito	S1
S1[1, 2]	1	digito	S1
S1[1, 2]	2	"."	S2
S2[3]	3	digito	S3
S3[3, 4]	3	digito	S3

- **AFD**



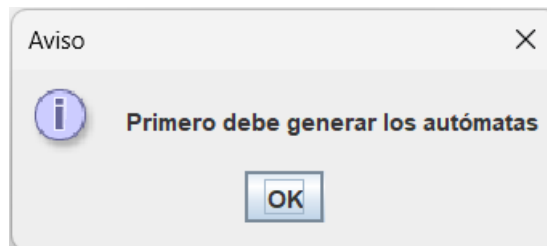
- **AFND**



Analizar Entrada

Esta opción nos ayudará si queremos analizar las entradas con sus respectivas Expresiones Regulares en caso de que ya se hayan generado los autómatas, en caso contrario el sistema nos mostrará un mensaje.

- **Mensaje en caso de no haber generado los autómatas previamente.**



- **Mensaje de consola en caso de que todo esté correcto.**

```
¡Análisis finalizado!  
La expresión: 301.59 no es válida con la Expresión Regular decimales  
La expresión: 1505.55 es válida con la Expresión Regular decimales  
La expresión: soy_un_id_1 es válida con la Expresión Regular identificador  
La expresión: ComPi1 no es válida con la Expresión Regular identificador
```

- **Archivo JSON de salida en caso de que todo esté correcto.**

```
{  
  {  
    "ExpresionRegular": "decimales",  
    "Resultado": "Cadena Inválida",  
    "Valor": "301.59"  
  },  
  {  
    "ExpresionRegular": "decimales",  
    "Resultado": "Cadena Válida",  
    "Valor": "1505.55"  
  },  
  {  
    "ExpresionRegular": "identificador",  
    "Resultado": "Cadena Válida",  
    "Valor": "soy_un_id_1"  
  },  
  {  
    "ExpresionRegular": "identificador",  
    "Resultado": "Cadena Inválida",  
    "Valor": "ComPi1"  
  }  
}
```

Manual de Usuario

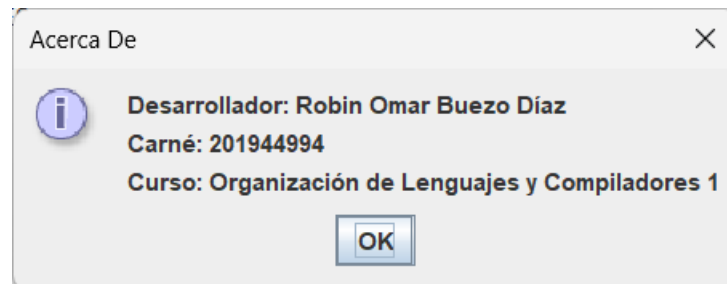
Este botón nos permite abrir el manual de usuario desde la aplicación.

Manual Técnico

Este botón nos permite abrir el manual técnico desde la aplicación.

Acerca de

Este botón nos mostrará la información de la aplicación y su desarrollador.



GLOSARIO

HTML: Es un lenguaje de marcado que nos permite indicar la estructura de nuestro documento mediante etiquetas.

AFD: Un autómata finito determinista (abreviado AFD) es un autómata finito que además es un sistema determinista; es decir, para cada estado en que se encuentre el autómata, y con cualquier símbolo del alfabeto leído, existe siempre no más de una transición posible desde ese estado y con ese símbolo.

AFND: Un autómata finito no determinista (abreviado AFND) es un autómata finito que, a diferencia de los autómatas finitos deterministas (AFD), posee al menos un estado $q \in Q$, tal que para un símbolo $a \in \Sigma$ del alfabeto, existe más de una transición $\delta(q,a)$ posible. Todo AFND puede ser convertido en un AFD equivalente.

JSON: JSON (acrónimo de JavaScript Object Notation, 'notación de objeto de JavaScript') es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a

su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.