



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Ingeniería en Ciencias y Sistemas  
Lab. Organización de lenguajes y compiladores

## **MANUAL DE USUARIO**

Justin Josue Aguirre Román  
202004734

## 1) REQUISITOS DEL SISTEMA

Cerciórese de que la computadora satisfaga o supere los siguientes requisitos antes de instalar la aplicación “Invasores Espaciales”.

<b>CPU</b>	Intel Celeron 800 MHz (Intel Core 2 Duo 2 GHz recomendados)
<b>RAM</b>	256 MB (2 GB recomendados)
<b>Espacio disponible en disco</b>	200 MB
<b>Sistema operativo</b>	Windows 7, Windows Vista o Windows XP

## 2) FUNCIONALIDADES

Una vez ejecutado el programa nos mostrara en pantalla el menú del mismo. Dándonos las diferentes opciones con que cuenta el software.

### I) Menú de Archivo

- Cargar archivo
- Guardar Archivo
- Guardar Como
- Generar JSON

### II) Analizar Archivo

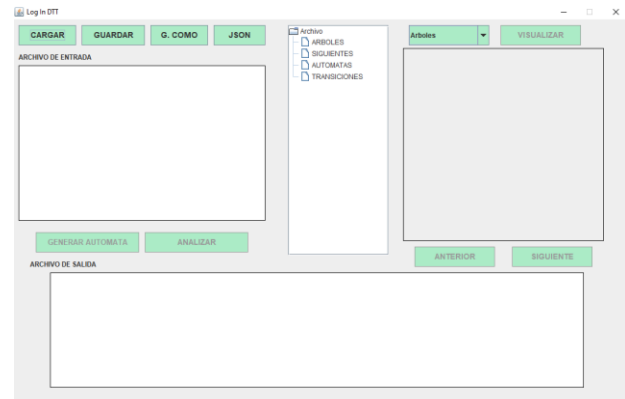
### III) Generar Autómatas

### IV) Visualizar Archivos

- Autómatas
- Arboles
- Tabla de Siguietes
- Tabla de Transiciones

### V) Consola de Salida

### VI) Salir

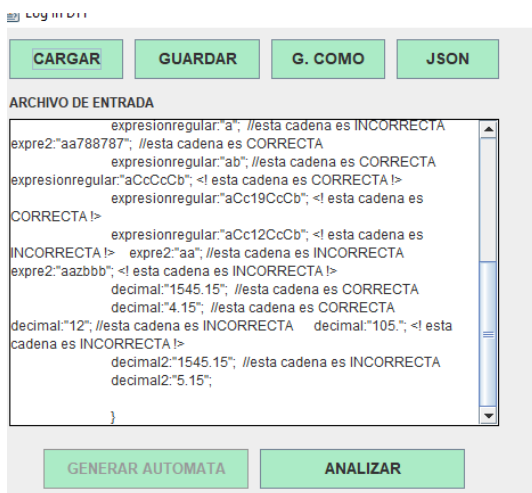


Para seleccionar y acceder a cada una de estas opciones se deberá dar un clic

sobre el botón de opción deseada de las posibilidades expuestas en el menú anterior.

Se requiere un programa que permita la edición de programas fuente a través de un área de texto. Además, que permita la visualización de los resultados y reportes de análisis.

## MENÚ ARCHIVO



- Abrir archivo: Nos permitirá abrir un archivo que esté guardado en disco. Los archivos a abrir tendrán extensión [.exp].

- Guardar: Nos permitirá guardar el archivo actual.

- Guardar Como: Nos permitirá guardar el archivo con otro nombre.

- Generar JSON de salida: Analizará el archivo en base a las

cadena a evaluar con las expresiones regulares

creadas anteriormente, indicando si es correcta o incorrecta su evaluación y guardará el archivo en una ruta definida por el usuario.

## ARCHIVO DE ENTRADA

El archivo para la definición de las expresiones regulares se compone de una sección en la que cada sentencia define el token (identificador) con que el analizador debe reconocer los lexemas ingresados, seguido de la definición de la expresión regulares. Seguido de esto dos símbolos de porcentaje (%%). Por último, se define un identificador y un lexema de entrada, el cual se deberá comparar con la definición que se encuentra en la parte superior.

Cada sentencia se delimita utilizando punto y coma.

Ejemplo:

Para el archivo de entrada pueden venir comentarios de una línea, multilínea, conjuntos, expresiones regulares y expresiones de prueba.

-El lenguaje EXP permitirá el uso de **comentarios de una o varias líneas**. Para comentarios multilínea se utilizará el símbolo “<!” para indicar el inicio del comentario y el símbolo “!>” para indicar el final del comentario. Para comentarios de una sola línea se utilizará la doble barra //.

-**Las expresiones regulares** establecen el patrón que se debe cumplir para representar un token, estas se reconocerán en notación polaca o prefijada.

Notación	Definición
. a b	Concatenación entre a y b
a b	Disyunción entre a y b
* a	0 o más veces
+ a	1 o más veces
? a	0 o una vez

-Para la definición de **conjuntos** se utiliza la palabra reservada “CONJ”. Un conjunto puede utilizarse dentro de una expresión regular, pero no en la definición de otro conjunto.

La definición de conjuntos CONJ puede existir en cualquier parte del archivo y su uso de conjuntos se verá delimitado por { llaves }.

Notación	Definición
a~c	Conjunto {a, b, c}.
a~z	Conjunto de la a hasta la z en minúsculas.
A~Z	Conjunto de la A hasta la Z en mayúsculas.
0~7	Conjunto del 0 al 7.
0,2,4,6,8	Conjunto {0, 2, 4, 6, 8}
A,b,C,d	Conjunto {A, b, C, d}
!~&	Conjunto de signos entre ! (33 en código ascii) y & (38 en código ascii). Nota: el rango valido será desde el ascii 32 hasta 125 omitiendo los ascii de las letras y dígitos.

Los conjuntos vistos anteriormente son ejemplos de las diferentes variantes que

éstos pueden tomar (Ej: también puede existir a~d o 0~9).

Dentro del lenguaje pueden utilizarse estos caracteres especiales:

Notación	Definición
\n	Salto de línea
\'	Comilla Simple
\"	Comilla Doble

Ejemplo

```
{ //definicion de conjuntos
  CONJ:impar -> 1,3,5,7,9;
  CONJ:grupo_1 -> 7~8;
  CONJ:letrac -> c,C;
  CONJ:digito -> 0~9; //definicion de expresiones regulares

  expresionregular-> .."a" * | {impar} {letrac} "b";
  expre2->.."a"."a"+{grupo_1};
  //esta expresion regular esta en carpeta: simple
  decimal-> . + {digito} . "." + {digito};
  decimal2-> . {digito} . "." + {digito};
  %%
  %% //ahora toca evaluar las cadenas
  expresionregular:"a"; //esta cadena es INCORRECTA
  expre2:"aa788787"; //esta cadena es CORRECTA
  expresionregular:"ab"; //esta cadena es CORRECTA
  expresionregular:"aCcCcCb"; <! esta cadena es CORRECTA !>
  expresionregular:"aCc19CcCb"; <! esta cadena es CORRECTA !>
  expresionregular:"aCc12CcCb"; <! esta cadena es INCORRECTA !>
  expre2:"aa"; //esta cadena es INCORRECTA
  expre2:"aazbbb"; <! esta cadena es INCORRECTA !>
  decimal:"1545.15"; //esta cadena es CORRECTA
  decimal:"4.15"; //esta cadena es CORRECTA
  decimal:"12"; //esta cadena es INCORRECTA
  decimal:"105."; <! esta cadena es INCORRECTA !>
  decimal2:"1545.15"; //esta cadena es INCORRECTA
  decimal2:"5.15";

}
```

## ANALIZAR ARCHIVO

Al seleccionar esta opción el software interpretara los conjuntos las expresiones regulares y las pruebas del archivo de salida, ordenando todo para facilitar la generación de autómatas.

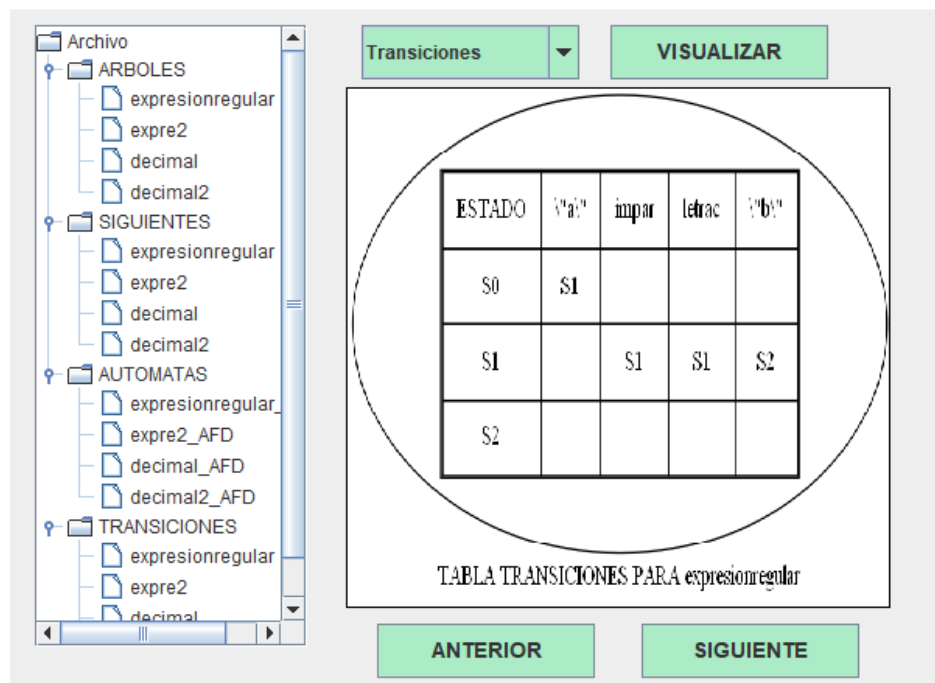
## GENERACIÓN DE AUTÓMATAS

Esta opción realizara el método completo de generación de autómatas AFD por medio del método del árbol, generando imágenes de la tabla del árbol, tabla de siguientes, tabla de transiciones y el autómata AFN de cada expresión regular dentro del archivo de salida.

Además, habilitara las opciones de visualización de cada una de estas salidas dentro del software. Estos archivos generados se guardarán en carpetas predefinidas que llevan el nombre del paso generado.

## VISUALIZAR ARCHIVOS

Se podrá seleccionar únicamente después de haber cargado, analizado y generado según contenga el archivo de entrada. Pudiendo seleccionar con un combo box los diferentes pasos del método del árbol a cada expresión regular, moviéndose a la siguiente o anterior con los botones que contienen su respectiva palabra comando.



## ARCHIVO DE SALIDA

Parte del sistema que muestra los resultados de la validación de pruebas con su

expresión regular, esto obteniendo la información de los archivos JSON de cada expresión regular.

