UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1

SECCIÓN “C”

PRIMER SEMESTRE 2023

MANUAL DE USUARIO

Diego Andrés Huite Alvarez

202003585

18/03/2023

Índice

INTRODUCCIÓN

Bienvenido al manual de usuario de "Exregan", un software diseñado para generar autómatas finitos deterministas y no deterministas a partir de expresiones regulares en notación polaca y usando el método del árbol. Con Exregan, podrá diseñar y analizar expresiones regulares y generar sus respectivos AFD Y AFND.

REQUERIMIENTOS

Software:

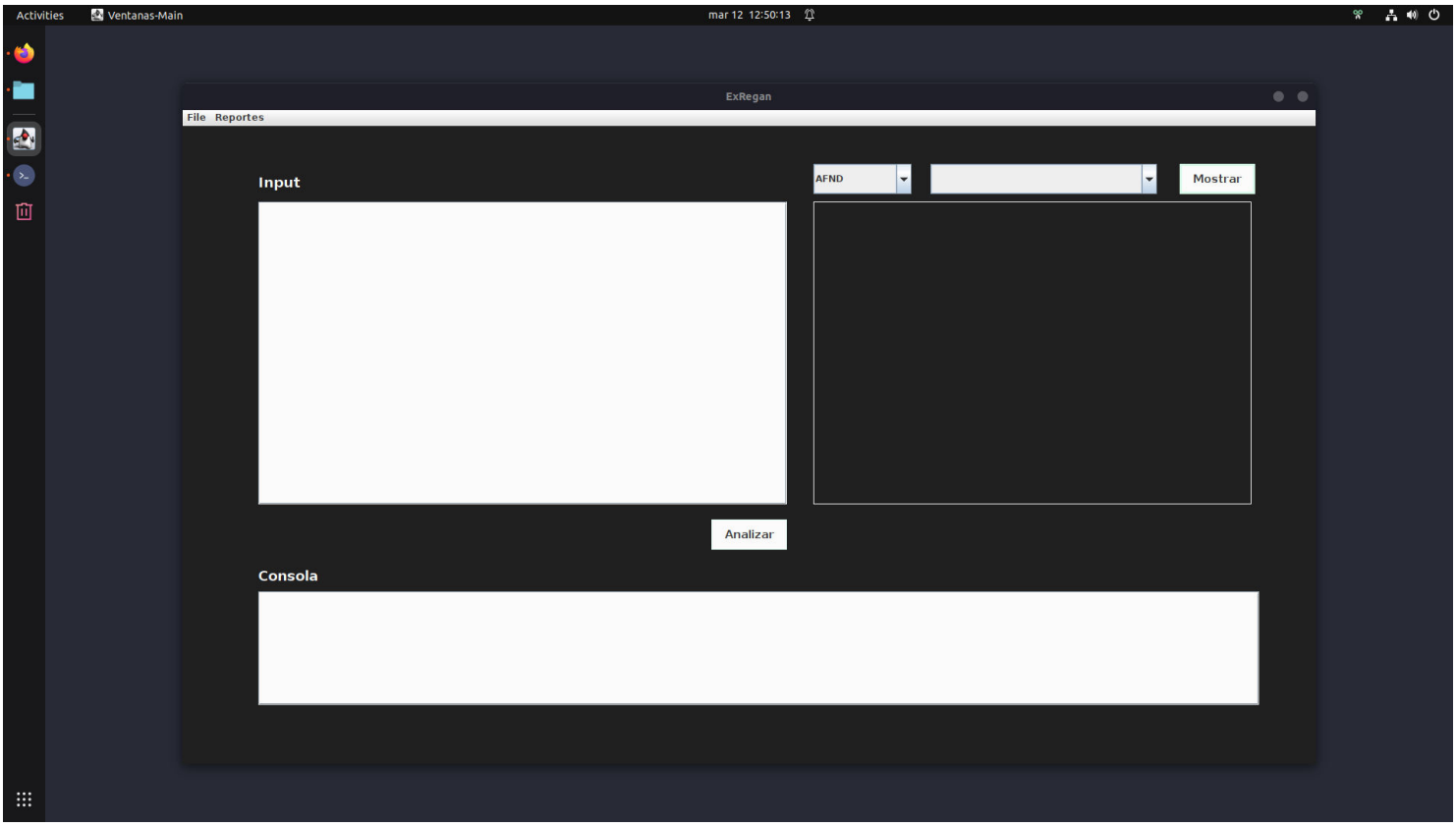
- Sistema operativo windows, linux o macOS
- Java jdk 8
- Visor de imágenes
- Graphviz instalado

Hardware:

- Mouse
- Teclado
- Monitor
- 2gb de ram

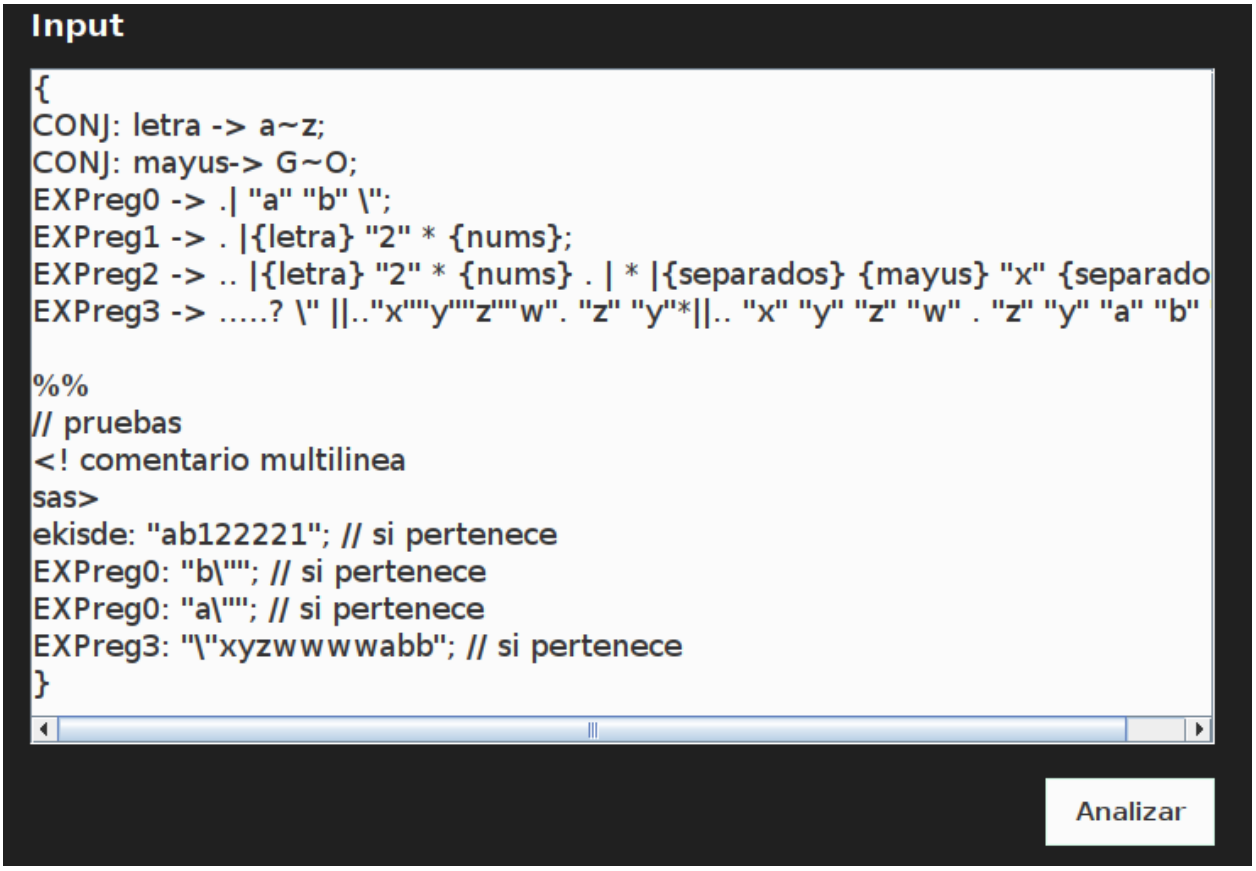
EJECUCIÓN INICIAL DEL PROGRAMA

Abra la carpeta donde tenga el archivo .jar del programa y en consola escriba el siguiente comando: java -jar NombreDelArchivo.jar. Cuando lo ingrese automáticamente se abrirá una ventana con el siguiente aspecto:



Se crearán carpetas en el directorio donde esté el jar para guardar los reportes en una manera más ordenada.

ARCHIVO DE ENTRADA



El archivo de entrada que acepta el programa luce así:

Donde todo el código debe de ir entre llaves, y antes del %% usted declarará conjuntos de caracteres. Estos pueden ir de letra-letra, caracter-caracter (entre el 32 y 125 en ASCII) y listas de caracteres como por ejemplo CONJ: vocales-> a,e,i,o,u;

También puede declarar las expresiones regulares a analizar (en notación polaca). Por ejemplo si usted quisiera analizar la expresión regular “a”|”b”, deberá escribirla así: | “a” “b”. Cabe recalcar que usted también puede usar los conjuntos previamente definidos, pero estos deben de ir entre llaves de la siguiente manera: {NombreConjunto}

Todo lo que venga después del %% serán solamente evaluaciones de cadenas con la expresión regular.

Expresión_Regular_con_la_que_comparar: “cadena_a_comparar”;

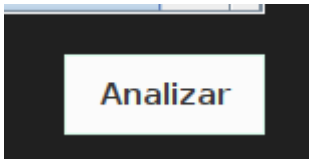
Usted también puede agregar comentarios de una línea, cuando escriba antes //Cualquier cosa.

y comentarios multilínea:

<! hola

que tal !>

ANALIZAR ENTRADA

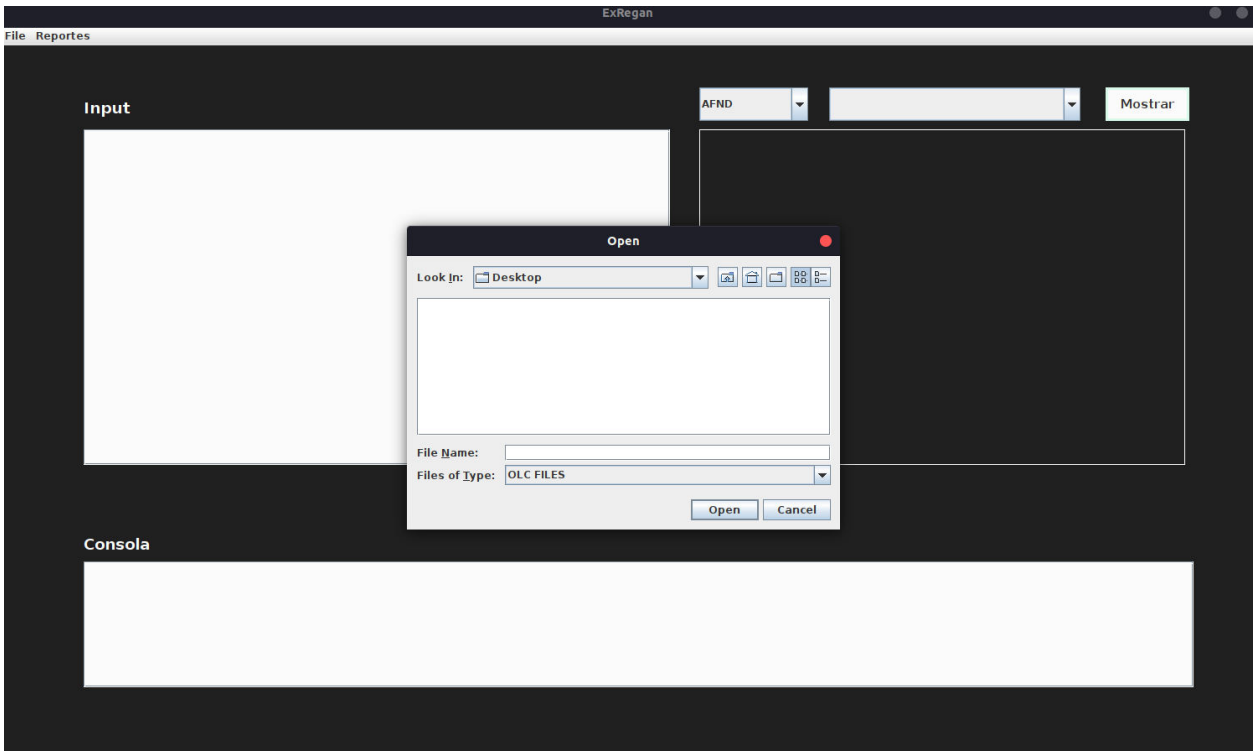


Usted podrá analizar la entrada en la caja de texto con ese botón. Cuando usted analice la entrada entonces se crearán los árboles de las expresiones regulares, las tablas de siguientes y las tablas de transiciones. Para así usted poder solicitar los reportes del menú Reportes que se explicará a continuación.

MENÚ FILE

Este menú File o archivo, le permite a usted hacer las siguientes cosas:

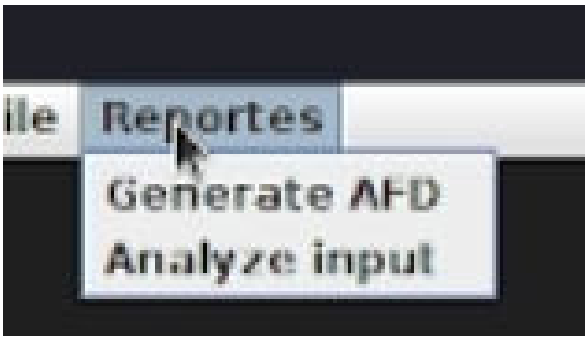
- New: permite limpiar la consola “input” para que pueda escribir de nuevo su código.
- Open: permite leer archivos con extensión .olc
- Save: permite guardar el archivo actual.
- Save as: permite la opción guardar como



MENÚ REPORTES

Este menú tiene las siguientes opciones:

- Generate AFD: genera los autómatas afnd y afd (debe de haber analizado primero el texto en consola).
- Analyze input: analiza las cadenas que le haya dado usted de entrada en el input(ya deben de estar generados los autómatas).



EJEMPLO

Una vez usted analice la entrada, tendrá los árboles, tabla de siguientes y tabla de transiciones generadas en las carpetas en el mismo directorio donde usted tiene el .jar

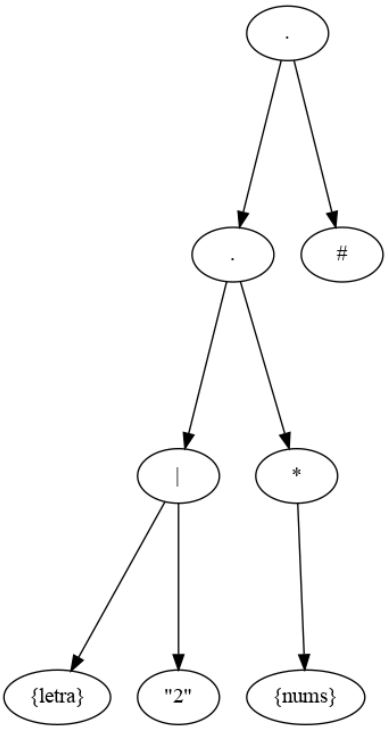
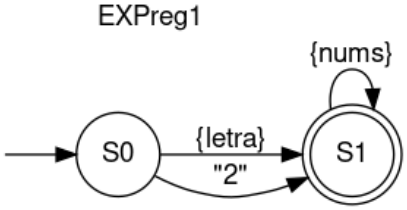
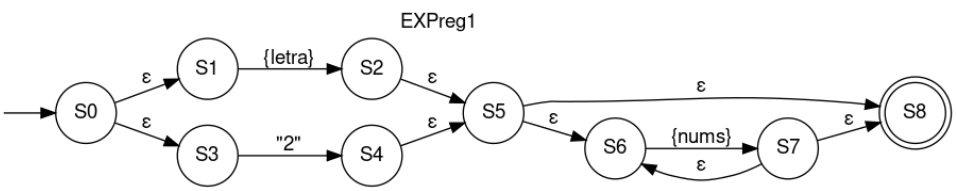


TABLA DE SIGUIENTES		
Num_hoja	Simbolo	Siguientes
0	{letra}	[2, 3]
1	"2"	[2, 3]
2	{nums}	[2, 3]

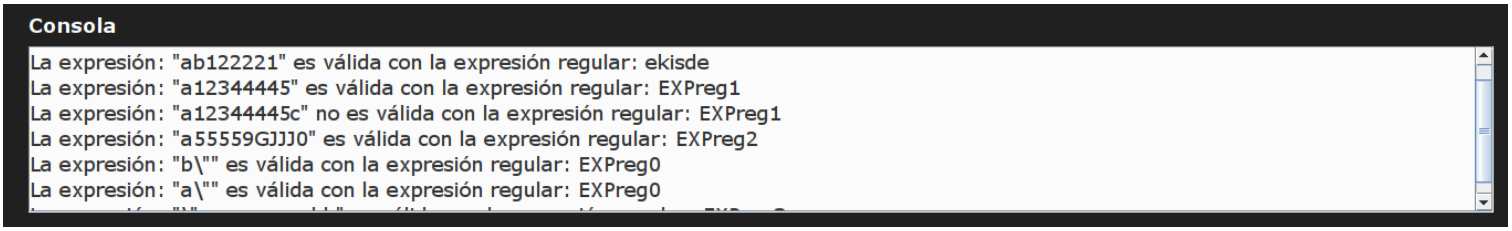
Estados/Terminales	{nums}	"2"	{letra}
S0 {0,1}	-	S1	S1
S1* {2,3}	S1	-	-

Generando los autómatas con el menú reportes, estos lucen así:





Analizando las inputs con la opción analyze input del menú repotres obtenemos la



siguiente información:

y el siguiente reporte en json:


```
[
  {
    "ExpresionRegular": "ekisde",
    "Valor": "\"ab122221\"",
    "Resultado": true
  },
  {
    "ExpresionRegular": "EXPreg1",
    "Valor": "\"a12344445\"",
    "Resultado": true
  },
  {
    "ExpresionRegular": "EXPreg1",
    "Valor": "\"a12344445c\"",
    "Resultado": false
  },
  {
    "ExpresionRegular": "EXPreg2",
    "Valor": "\"a55559GJJJ0\"",
    "Resultado": true
  },
  {
    "ExpresionRegular": "EXPreg0",
    "Valor": "\"b\\\"\"",
    "Resultado": true
  },
  {
    "ExpresionRegular": "EXPreg0",
    "Valor": "\"a\\\"\"",
    "Resultado": true
  },
  {
    "ExpresionRegular": "EXPreg3",
    "Valor": "\"\\\"xyzwwwwabb\"",
    "Resultado": true
  }
]
```

Cabe recalcar que se llegara a encontrar un error en la entrada, se generará un html con los datos de los errores encontrados y usted no podrá generar los autómatas pero se creará un html en la carpeta de errores luciendo así:

Reporte de errores

Tipo	Lexema	Fila	Columna
Léxico	"þ"	19	0