

PROYECTO 1



“Analizador Léxico”

LUIS EDUARDO GUILLÉN RUIZ

UNIVERSIDAD SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE OCCIDENTE “CUNOC”
DIVISIÓN DE CIENCIAS DE LA INGENIERÍA
INGENIERÍA EN CIENCIAS Y SISTEMAS
SEGUNDO SEMESTRE
2,019
Ing. OLIVER ERNESTO SIERRA

“ID Y ENSEÑAD A TODOS”

Contenido del Manual

Contenido del Manual	2
Presentación	3
Objetivos del Programa	3
Generales	3
Específicos	3
Instrucciones de Programa	4
Requerimientos Técnicos	4
Herramientas de Construcción	4
Como Acceder al Programa	5
Home	5
Analizador	6
Diagrama de Moore	6
Autómata General	7
Conjunto de Estados del Autómata:	7
Estado Inicial	7
Alfabeto Σ	7
Estados de Aceptación F	7
Función de Transición ∂ (delta)	7
Tabla de Transición	8
Estado Inicial S0	8

“Manual Técnico, Analizador Léxico”

Presentación

El presente documento es el manual Técnico del analizador léxico implementando la metodología Cliente-Servidor en la cual se le dará acceso al cliente de mandar archivos para un análisis en la parte lógica del lado del servidor la cual será la encargada de manejar todos los datos que el cliente desee analizar.

La aplicación está diseñada en Nodejs por medio de la sintaxis JavaScript para poder dar la respectiva visualización por medio de cualquier navegador que posea el motor de plantillas para levantar el archivo generado, así mismo se incluye el funcionamiento básico del mismo y una guía para su primera ejecución y posteriormente el uso del mismo.

Objetivos del Programa

Generales

- Familiarizar al estudiante con el lenguaje JavaScript.
- Conocer el proceso de análisis léxico realizado por un compilador.
- Aplicar conceptos de creación de Autómatas.
- Elaborar la lógica para la solución del problema planteado.

Específicos

- Construcción de algoritmos para los requerimientos de la actividad.
- Ampliar el conocimiento de Programación en JavaScript.
- Desarrollar una aplicación web con JavaScript utilizando el Backend como el Frontend.
- Desarrollar toda la lógica del analizador léxico en el lado del Backend.
- Desarrollar en el Frontend la comunicación con el Backend para realizar el análisis léxico.
- Desarrollar diagramas de Moore.
- Desarrolló de manual técnico.

Instrucciones de Programa

1. Ejecutar Visual Studio Code.
2. Instalar dependencias Npm.
3. Levantar el Server.
4. Ejecutar el puerto de visualización.
5. Elegir entre las rutas de acceso.
6. Buscar Archivo de Texto a Analizar.
7. Analizar
8. Solicitar Tokens
9. Refrescar
10. Salir

Requerimientos Técnicos

- SO que soporte JavaScript (Ejemp.: Linux, Windows, IOs, Android).
- Java 1.8.0_201 o compatibles.
- Visual Studio Code (creación: v. 1.39.1) o cualquier editor de lenguaje.
- Entender lenguaje JavaScript (para lógica del programa).
- Entender lenguaje HTML (para estructura del programa).
- Entender lenguaje CSS (para diseño del programa).
- Experiencia Diseño Web.

Herramientas de Construcción

- Visual Studio Code
- NodeJs
- BootStrap
- Axios
- Navegador Web
- PC funcional
- Conocimientos de JavaScript
- Control de Versiones (GitHub)

Como Acceder al Programa

Se debe tener el repositorio Online del Analizador descargar por medio de https://github.com/LGuilln/201744792_Luis_Guillen_P1. Luego clonar en repositorio local, instalar dependencias de npm y Nodemon así como Axios y Express luego levantar el servidor para poder visualizar desde el navegador de preferencia. Analizar archivos de texto.

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

luedu@DESKTOP-DL33U4M MINGW64 ~/Documents/PrimerProyectoL_F (master)
$ npm run dev

> primerproyecto1_f@0.0.1 dev C:\Users\luedu\Documents\PrimerProyectoL_F
> nodemon src

[nodemon] 1.19.3
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src`
C:\Users\luedu\Documents\PrimerProyectoL_F\src\views
escuchando en el puerto 3000
```

Cargar y Ejecutar Proyecto

Home

Cargar Archivo X Analizador Lexico X GitHub - LGuilln/201744792_L... X Analizador Lexico X +

localhost:3000

Autor Analizador Diagramas Moore

Centro Universitario de Occidente "CUNOC"

Analizador Léxico de un Compilador

Primer Proyecto de Lenguajes Formales Formales y de Programación

Autor

Luis Eduardo Guillen Ruiz

201744792

Implementado Control de Versiones

• [Manejador Github](#)

Analizador

Cargar Archivo Cargar Archivo GitHub - LGuilln/201744792_Lu X Analizador Lexico

localhost:3000/analizador

Autor Analizador Diagramas Moore

Selección de Documento TXT

[Examinar...](#) entrada.txt

Contenido del archivo:

```
variable a = 5;
variable b = "hola mundo";

mientras a >= 1 hacer {
  si (a == 3) {
    b = "abcedfg";
  }
  sino {
    b = b + " - " + a;
  }
  a--;
}

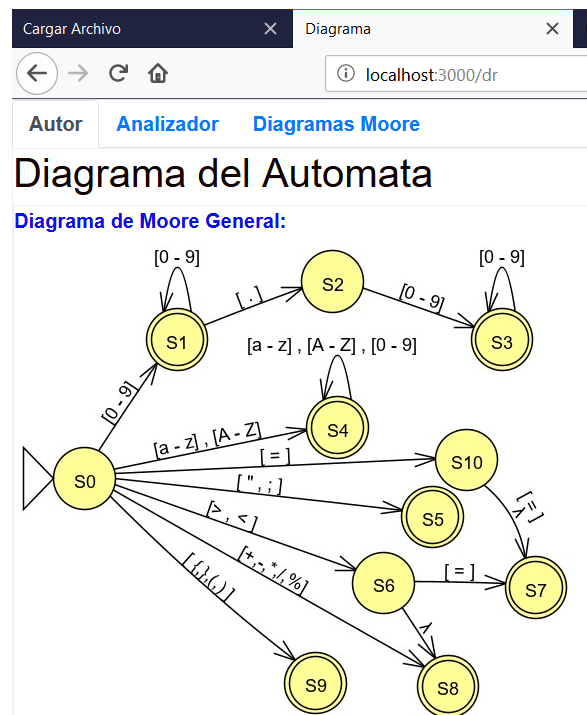
variable c = b.toString();

a = (5+6/(25*4)-1)+1;

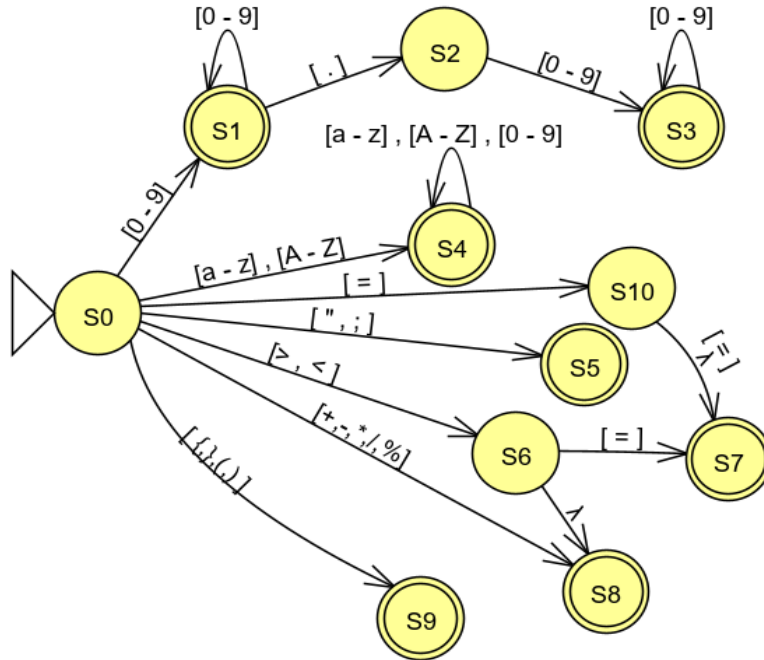
console.log(c);
```

Enviar

Diagrama de Moore



Autómata General



Conjunto de Estados del Autómata:

R// $Q = \{S0, S1, S2, S3, S4, S5, S6, S7, S8, S9, S10\}$

Estado Inicial

R// S0

Alfabeto Σ

R// $\{[0-9] \wedge [A-Z] \wedge [a-z] \wedge [+,-,*,/,%,<,>] \wedge [",;] \wedge [{,},(,)]\}$

Estados de Aceptación F

R// $F = \{S1, S3, S4, S5, S7, S8, S9\}$

Función de Transición ∂ (delta)

R//

$\partial(S0, [0-9]) = S1$
 $\partial(S0, [A-Z]) = S4$
 $\partial(S0, [a-z]) = S4$
 $\partial(S0, [",;]) = S5$
 $\partial(S0, [<,>]) = S6$
 $\partial(S0, [+,-,*,/,%]) = S8$
 $\partial(S0, [{,},(,)] = S9$
 $\partial(S0, [=]) = S10$
 $\partial(S1, [0-9]) = S1$

$\partial(S1, \cdot) = S2$
 $\partial(S2, [0-9]) = S3$
 $\partial(S3, [0-9]) = S3$
 $\partial(S4, [0-9]) = S4$
 $\partial(S4, [A-Z]) = S4$
 $\partial(S4, [a-z]) = S4$
 $\partial(S6, [=]) = S7$
 $\partial(S6, \lambda) = S8$
 $\partial(S10, [\lambda, =]) = S7$

Tabla de Transición

	[0-9]	[A-Z]	[a-z]	“, ;	< , >	+, -, *, /, %	{, }, (,)	=	.	λ
S0	S1	S4	S4	S5	S6	S8	S9	S10	-	-
S1	S1	-	-	-	-	-	-		S2	-
S2	S3	-	-	-	-	-	-		-	-
S3	S3	-	-	-	-	-	-		-	-
S4	S4	S4	S4	-	-	-	-		-	-
S6	-	-	-	-	-	-	-	S7	-	S8
S10	-	-	-	-	-	-	-	S7	-	S7

Estado Inicial S0

Cadena	[0-9]	[A-Z]	[a-z]	“, ;	< , >	+, -, *, /, %	{, }, (,)	=	.	λ
Estado	S1 S3 S4	S4	S4	S5	S6	S8	S9	S7 S10	S2	S7 S8