

**Universidad de San Carlos de Guatemala**  
**Centro universitario de Occidente**  
**División de ciencias de la Ingeniería**  
**Ingeniería en ciencias y sistemas**

Curso: Lenguajes Formales y de Programación  
Catedrático: Ing. Oliver Sierra  
Sección: "A"



Proyecto 1

Daniel Eduardo Fernández Ovando ..... 201731816

Quetzaltenango, 11 de Octubre del 2019.

# **Manual** **Técnico del** **“Analizador** **Léxico”**

**Lenguajes Formales y de programación**

**Sección “A”**

**Nombre: Daniel Eduardo Fernández Ovando**

**Carné: 201731816**

## **Manual Técnico del Analizador Léxico:**

### **INSTRUCCIONES DE USO (inicio, objetivos, instrucciones sobre el analizador):**

Este Manual Técnico se hace con el fin de darle al cliente o usuario una mejor idea o forma de poder enseñarle como utilizar o que pueda ver cómo funciona este Analizador. El objetivo de este analizador es tratar de ver diferentes tipos de patrones y así este lenguaje que reconozca que hemos ingresado a la hora de ingresarlo al Analizador un texto. Cada persona podrá ingresar una palabra, número, signo, o lo que se pide para visualizar a la hora de ingresarlo que nos regrese ya sea una palabra reservada, identificador, números o signos y etc.

Dentro de este proyecto deberán aparecer en el Frontend tres páginas las cuales una será para que aparezca en la primera la información del desarrollador o estudiante el cual haya realizado este proyecto, luego la segunda página que nos deberá aparecer será el analizador con su botón para poder ingresar un texto y recibir tokens, y que luego en un listado vaya apareciendo los tokens. Y en la última página deberán aparecer los diagramas de Moore en los cuales irán nuestros autómatas los cuales son los que fueron de ayuda para poder realizar la lógica del analizador y el proyecto.

### **Partes que componen al Analizador:**

- Tres páginas Web
- Frontend
- Backend
- Información del estudiante o desarrollador
- Analizador Léxico (conformado por botones, y un frontend atractivo)
- Historial o listado
- Diagramas de Moore
- Autómatas
- Tabla de Transición
- Imágenes de los autómatas

- Utilización de Node.js
- Utilización de Axios

## **Elementos del Manual Técnico:**

### **Objetivos Generales:**

- Familiarizar al estudiante con el lenguaje JavaScript.
- Elaborar la lógica para la solución del problema planteado.
- Aplicar conceptos de programación con JavaScript y Autómatas recibidos en clase y laboratorio.
- El realizar buenas prácticas.

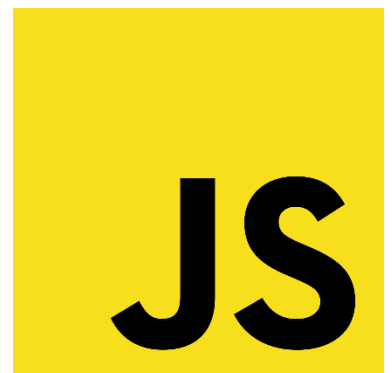
### **Objetivos Específicos:**

- Elaborar un Diagrama de Moore para mejor comprensión del programa y del autómata.
- Poder ampliar el conocimiento en JAVASCRIPT
- Poder ampliar conocimiento en programación con Autómatas
- Implementar un Frontend y Backend entendibles
- Implementación de node.js
- Implementación de Axios

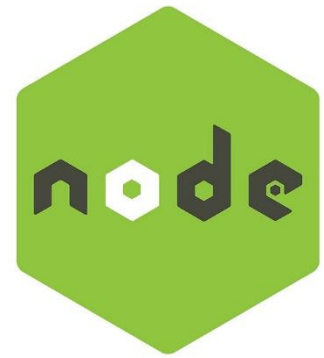
## **Elementos de requerimiento para la funcionalidad del programa (Analizador):**

### **\*\*\* Requiere PC con los Sistemas operativos siguientes**

- Sistema Operativo (Windows o Linux)
- Java 1.8.0\_201 o compatibles.



- Visual Studio Code o cualquier otro editor de lenguaje JAVASCRIPT.



- Node.js o cualquier otro.
- Conocimientos en el lenguaje JAVASCRIPT
- Repositorio en GITHUB o el proyecto completo.
- Entender lenguaje JAVASCRIPT (para ver creación de programa).
- Experiencia previa en páginas web.

#### - **Elementos para la construcción del programa**

- Proyecto JAVASCRIPT
- Packages
- Package.json
- Gitignore
- Backend
- Métodos de los atributos
- Express
- Frontend
- Routes
- Matrices
- Autómatas

## **Descripción del Programa**

Este Analizador consiste en tratar de ver si el texto que se va a ingresar en el analizador lo va a reconocer y nos dirá que es lo que estamos ingresando, con esto podemos ir analizando todo lo que vayamos a ir ingresando y así lo que ingresemos se analice y así poder decir si es ya sea una palabra reservada, un identificador, un número, un signo o un Error y etc.

Ya el cliente una vez habrá el servidor en algún navegador ya podrá ver todo lo que está en nuestro Frontend que es o se utiliza de una manera sencilla y es muy entendible, acá podrá ver las tres páginas las cuales una o la primera mostrara la información del estudiante o desarrollador, luego en la segundo mostrara lo que es el analizador el cual es donde veremos todo que podemos ingresar y lo que nos regresara de respuesta, y por último en la última página veremos los diagramas de Moore donde irán nuestros respectivos autómatas los cuales son los que nos ayudaron a ver de manera óptima a la hora de programar el analizador.

## Analizador:

Primer proyecto.pdf

Información

localhost:2000/in

traductor - Buscar c...

Información

Analizador

Diagramas Moore

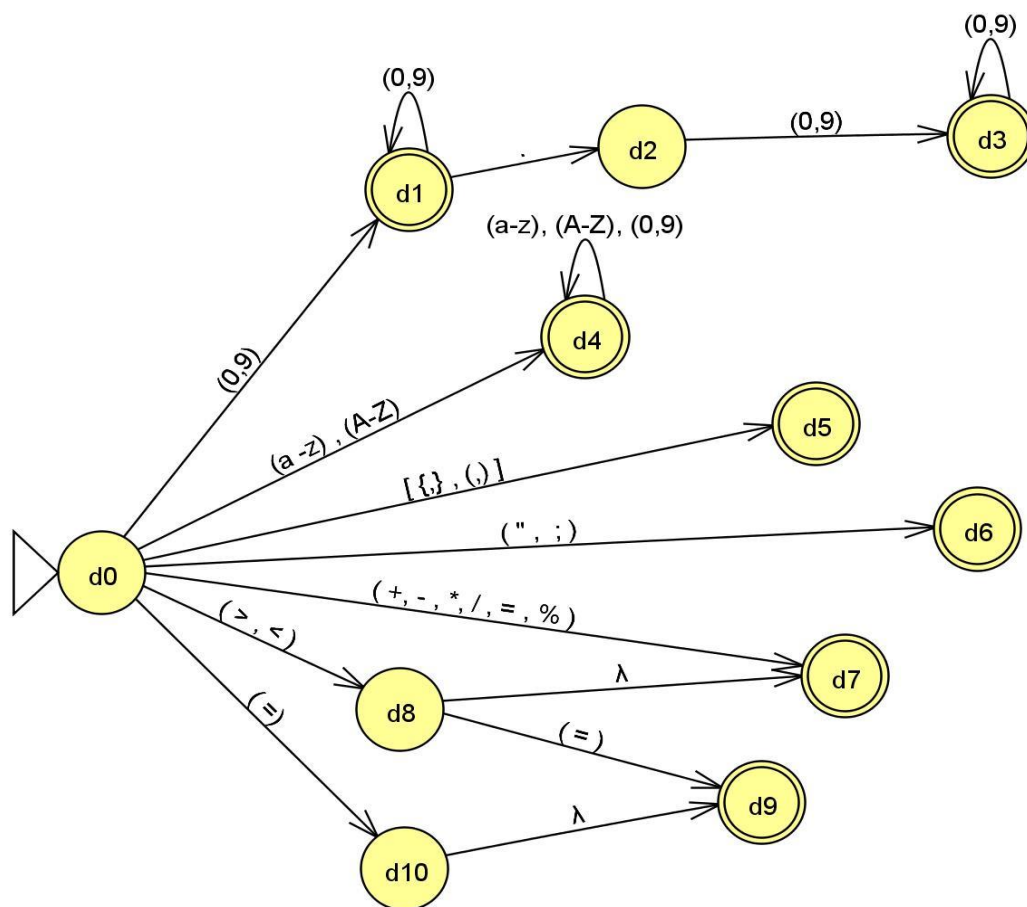
----- Universidad de San Carlos -----  
"Centro Unversitario de Occidente"

- Proyecto 1 -

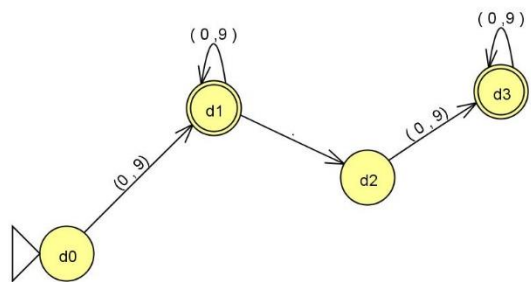
-----

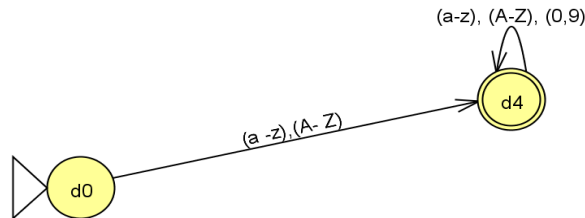
- Proyecto 1
- DANIEL EDUARDO FERNÁNDEZ OVANDO
- Carné: 201731816

## Autómatas:









---

## **Alfabeto:**

**Autómata:** Un autómata es un modelo matemático para una máquina de estado finito (FSM sus siglas en inglés). Una FSM es una máquina que, dada una entrada de símbolos, "salta" a través de una serie de estados de acuerdo a una función de transición (que puede ser expresada como una tabla).

**JavaScript:** es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, 3 basado en prototipos, imperativo, débilmente tipado y dinámico.

**Node:** Node es un programa de servidor.

**CSS:** Las siglas CSS (Cascading Style Sheets) significan «Hojas de estilo en cascada» y parten de un concepto simple pero muy potente: aplicar estilos (colores, formas, márgenes, etc...) a uno o varios documentos (generalmente documentos HTML, páginas webs) de forma masiva.

**EJS:** permite procesar paginas HTML (cuya extensión será .ejs) en el servidor antes de servir las al cliente, de forma que éstas podrán tener partes dinámicas.

**Axios:** es un cliente HTTP basado en Promesas para Javascript, el cual puede ser utilizado tanto en tu aplicación Front-end, como en el Back-end con Nodejs.

**Front-end:** es la parte de una web que conecta e interactúa con los usuarios que la visitan. Es la parte visible, la que muestra el diseño, los contenidos y la que permite a los visitantes navegar por las diferentes páginas mientras lo deseen.

**Back-end:** es la parte del software que interactúa con los usuarios y el back-end es la parte que procesa la entrada desde el front-end.

**Página Web:** es un documento o información electrónica capaz de contener texto, sonido, vídeo, programas, enlaces, imágenes y muchas otras cosas, adaptada para la llamada World Wide Web (WWW) y que puede ser accedida mediante un navegador web.