

Manual tecnico de analizador lexico
Centro Universitario de Occidente
CUNOC
Lenguajes de programación 1
David Rodolfo Martinez Miranda
201632145

Requerimientos mínimos

- Para una funcionalidad plena y estable se requiere de un navegador (Mozilla, Chrome, Safari, Explorer, Opera), en la version mas reciente.
- Tener un minimo de ram de 500mb.
- Procesador mayor a core 2 duo.
- Adobe flash, mas reciente instalado en el navegador.
- Conexión a internet (**Obligatoria**)

Instaladores básicos para el servidor

- Tener nodejs en su version 5.6.0.
- Para los ejs una version de npm instalada, version 2.7.1.
- Para nodemon version instalada de 1.19.2.
- Para express una version de 4.17.1.
- El programa del analizador esta en su version 1.

Src

Src contiene carpetas y dos archivos, las carpetas son public, routes, views; y los dos archivos son index y app.

Public

Public contendra, una imagen, que servira para el logo del la pestaña de todo el sistema. Asi mismo contendra un archivo con extencion css y cuatro archivos para la logica, seran de extencion js.

Css

El css contiene dos clases las cuales son para editar texto donde mostrara los titulos principales de cada pagina, tambien de como se mostrara el area para el texto, donde ingresara el usuario lo que desee analizar. Las dos clases son: .a y .textArea.

Js

1. Js Fronted, en el fronted se encontrara la logica que manda, recibe la informacion para mostrarle al usuario, los cuales contiene ayuda de **axios** para poder mandar el texto y asi mismo recuperar todo lo analizado desde el servidor, contara con 7 distintas variables "**let**" las cuales dos de ellas contendran datos como el area de texto y el espacio donde se mostrara lo que el usuario solicite en token. Tambien contara con 4 funciones (**getData, postData, mostrar, borrarTexto**) las cuales la primera recuperara los datos analizados desde el servidor, el siguiente mandara los que este almacenado en el area de texto; mostrar es para ver el token solicitado en la pagina; borrarTexto borrar lo que tenga almacenado el servidor como tambien lo que este en el area de texto.
2. Js Analizador, el en analizador contara con 16 **variables** globales las cuales servian para cada movimiento entre estados, los primeros 16 son del alfabeto aceptado por el automata, las cuales estan como "**fase**" con la subcadena del (1-16). Se cuenta con 20 variables locales "**let**" las cuales estan en distintas funciones y ayuda a que cada funcion cumpla con lo que se requiere. Se tiene un arreglo con cada palabra reservada con la cual se compara la cadena despues des analizar un identificador para saber si es reservada. Se tienen 7 **funciones principales**, las cuales son para cada automata, siendo verificador (**reservada, identificador, entero, flotante, operador, agrupador, signo**), cada una opera como se solicito. Se tienen otras 5 funciones (comparador, salida, token, salidaFinal, analizar), el primero compara un arreglo con un solo

carácter, el cual cumple con ver si se puede **mover dentro del automata** y con que camino; salida servira para saber en que posicion salio; token devuelve la cadena analizada; salida final sera para devolver un arreglo de lo analizado con los detalles; analizar es el automata completo unido.

3. Js tabla, este contiene **6 funciones (getData, separadorSaltos, sepradorComas, nuevaFila, llenar)**. Get data captura lo que analizo el servidor; separadorSaltos es para agrupar cada analisis realizado por el servidor; separadorComas para agrupar los datos necesarios para ser mostrados en pantalla; nuevaFila retornara una cadena la cual agrupa los datos necesarios para anadir una nueva fila en la tabla; llenar anade todas las filas que contiene el texto analizado.
4. Js texto, contiene las funciones para poder leer un archivo y mostrar una ventana emergente para poder seleccionar el archivo y ubicar el contenido en el area de texto.

Routes

1. index. Contiene la configuracion de las rutas, redirecciones, capturas e envio de informacion. Se tienen 5 variables constantes las cuales unas son, una de express para la utilizacion de express. Router para configurar cada ruta soicitada.
Se tienen 2 funciones (**convertidor, verificador**), convertidor se encarga de separa los saltos de linea en el texto entrante, verificador sera para ir avanzando en lo analizado y poder devolver cada **token uno a uno solicitado**. En este se analiza el texto o codigo, importando la funcion de la clase analizador.
Se tienen 6 envios de datos de la configuracion “**get**”, se tienen 3 capturas de datos de la configuracion “**post**”, se tienen 3 “**render**” para cada pagina renderizada.

Views

1. Partials, contiene codigo html5, codigo fijo que sera reutilizado en las 3 paginas.
2. 3 archivos ejs (**entrada, estudiante, tabla**), entrada corresponde al inicio de las paginas donde se podra ingresar el texto y ver token a token analizado con **retaro de 2 segundos**; estudiante contiene la pagina con la informacion del estudiante requerida; tabla contiene lo correspondiente a la tabla con 2 botones (**ver, borrar**), los cuales muestran y limpian los datos en la tabla.

Index y app

index se encarga de inicializar el servidor, importando el modulo de app; App contiene toda la configuracion del servidor (**port, views, view engine, route, public**). Port setea el puerto en que se escuchara; views contiene todo el html visual por el cliente de como se movera en las paginas; view engine setea el motor de vistas con un ejs; route setea las rutas (contenidas en carpeta routes); public contiene todo lo que sera utilizado por el usuario(navegador).

Package

contiene la informacion basica del servidor. En este se seteo el script, para poder utilizar nodemons.