Manual Técnico

Proyecto 2

ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1

C

Edgar Rolando Alvarez Rodriguez
202001144

MAYO 2023

INTRODUCCION

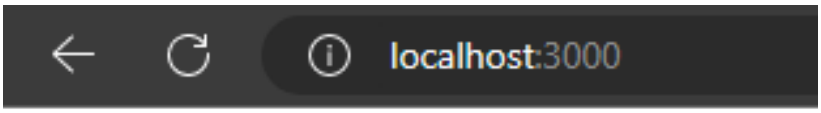
El presente manual abordará el tema del funcionamiento del código de la Proyecto 2, el cual consiste en realizar la lectura de un archivo "tw" así como generar a partir de este diversos gráficos sobre el análisis del sistema.

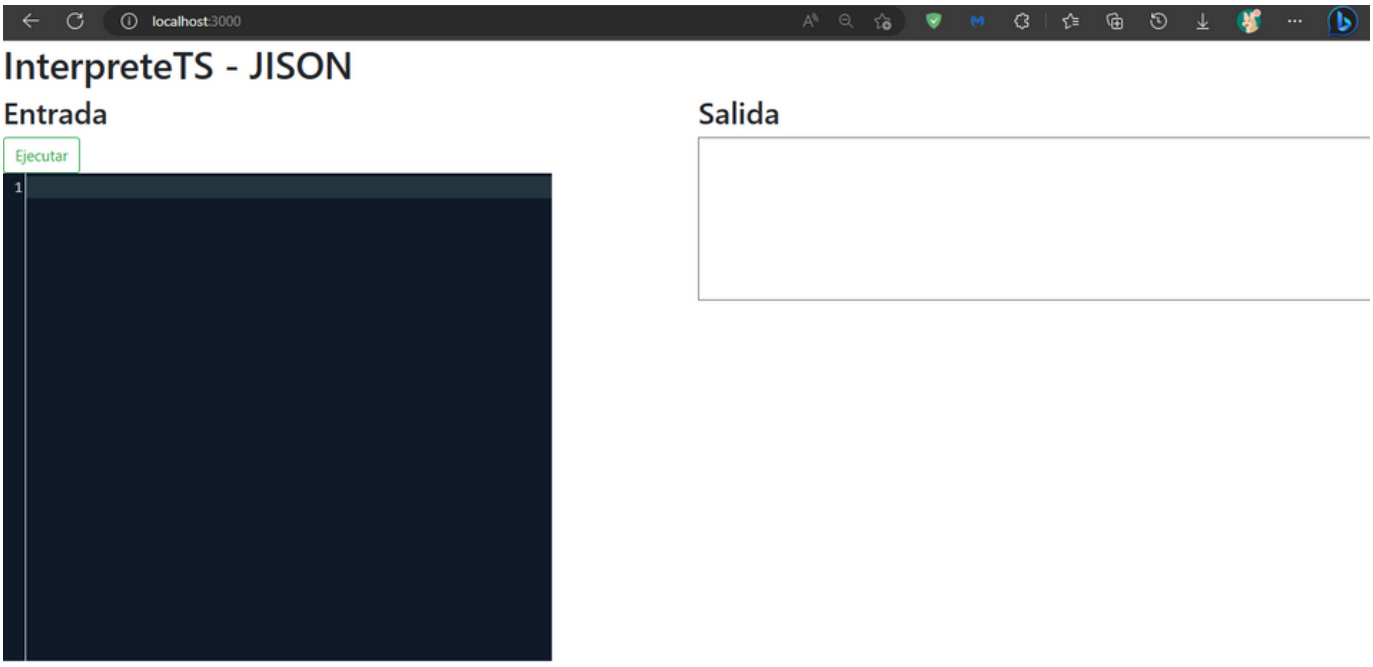
El manual esta dirigido a técnicos y programadores que manejan el soporte del software y puedan darle mantenimiento, añadir funcionalidades al programa o revisar el propósito de los archivos que componen el programa.

FUNCIONAMIENTO DEL SISTEMA

interfaz:

En el localhost:3000 se abre el puerto para colocar el codigo en un caja de texto para ser analizado.





FUNCIONAMIENTO DEL SISTEMA

interfaz (Código):

Se genera una pagina de tipo ejs para poder obtener los datos

```
<!DOCTYPE html>
<html>
<head>
    <title><%= title %></title>

    <!-- ================================== IMPORTACIONES ============================== -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
    <!-- ================================ ESTILOS BOOSTRAP ============================ -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integri
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgy

    <!-- ================================ CODEMIRROR ============================ -->
    <link rel="stylesheet" href="/codemirror/lib/codemirror.css">
    <script src="/codemirror/lib/codemirror.js"></script>

     <script src="/codemirror/lib/codemirror.js"></script>
     <script src="/codemirror/mode/clike/clike.js"></script>
     <link rel="stylesheet" href="/codemirror/lib/codemirror.css"/>
```

```
<body>
    <h1><%= title %></h1>
    <div class="row">
        <div class="col-6"><h2>Entrada</h2></div>
        <div class="col-6"><h2>Salida</h2></div>
    </div>
    <div class="row">
        <div class="col-6">
            <form method="POST" action="/ejecutar">
                <button type="submit" class="btn btn-outline-success">Ejecutar</button>
                <textarea  id="editor" name="codigo"><%= codigo %></textarea>
            </form>
        </div>
        <div class="col-6">
            <div class="row">
                <div class="col-2">
                    <textarea id="consola" name="textarea" rows="7" cols="110"><%= salida %></textarea>
                </div>
            </div>
```

CARGA DE DATOS

LECTURA DE ARCHIVOS:

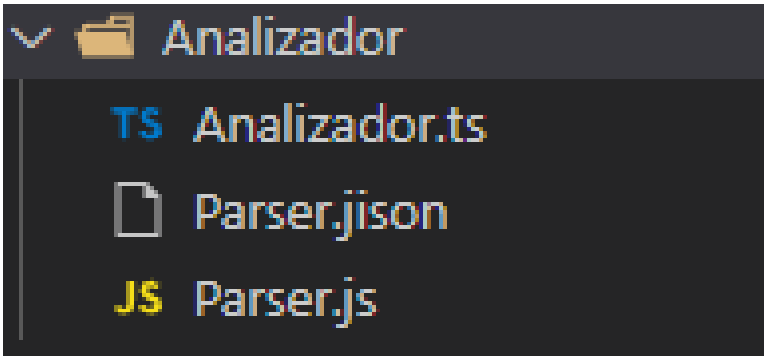
Para ingresar los archivos tw al sistema en esta ocasión se opto por colocar el texto directamente en el panel de entrada del sistema

```
let cadena_codigo = req.body.codigo;
let analizador = new Analizador(cadena_codigo, "editor");
let ast: AST = analizador.Analizar();
// console.log(analizador)
if(ast != undefined) {
    res.render('index.ejs', { title: 'InterpreteTS - JISON', salida: ast.getSalida(), codigo: cadena_codigo});
} else{
    res.render('index.ejs', { title: 'InterpreteTS - JISON', salida: 'ERROR al procesar cadena', codigo: cadena_c
}
```

Analisis de los Datos

Lexico

Una vez ingresado los datos dentro de la caja de texto, se procede a analizar los datos con el boton de "ejecutar", que se mandan al sistema y el archivo Parser.js y Parser.jison son capaces de analizar.

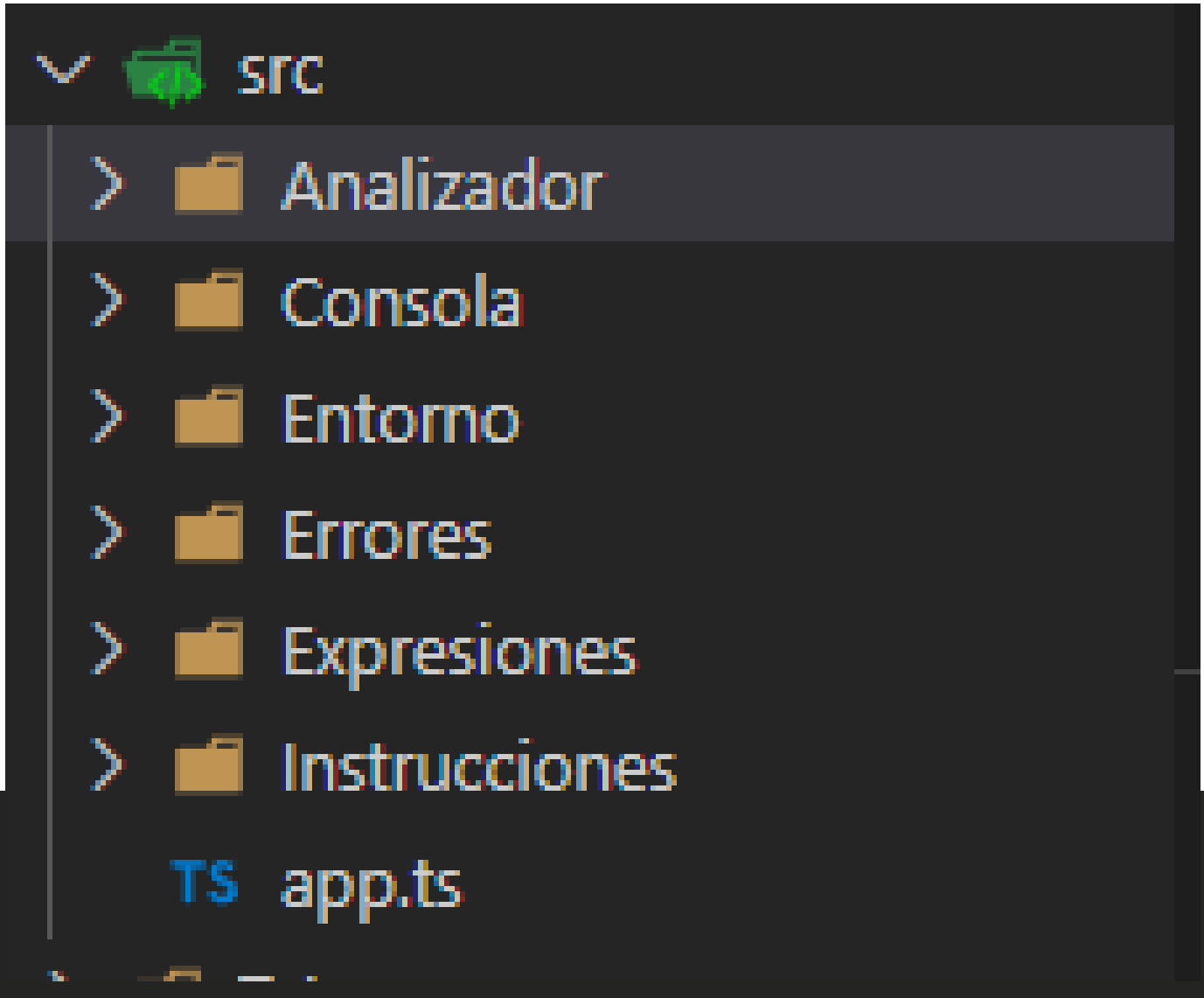



```
var Parser = (function(){
var o=function(k,v,o,l){for(o=o||{},l=k.length;l--;o[k[l]]=v);return o},$V0=[1,16],$V1=[1,21],$V2=[1,20],$V3=[1,23],$V4=[1,24],$V5=[1,25],$V6=[1,15],$V7=[1,26],$V8=[1,27],$V9=[1,28],
var parser = {trace: function trace () { },
yy: {},
symbols_: {"error":2,"INICIO":3,"SENTENCIAS":4,"EOF":5,"SENTENCIA":6,"BLOQUE_SENTENCAS":7,"{":8,"}":9,"DECLARACION":10,";":11,"FUNCION":12,"ASIGNACION":13,"SENTENCIAS_TRANSFERENCIA":
terminals_: {2:"error",5:"EOF",8:"{",9:"}",11:";",22:"id",23:"[",25:"]",27:"=",28:"tnew",30:"(",32:")",33:"tswitch",37:"tcase",38:":",39:"tdefault",40:"++",41:"--",42:"tif",44:"telse
productions_: [0,[3,2],[4,2],[4,1],[7,3],[7,2],[6,2],[6,1],[6,2],[6,1],[6,1],[6,1],[6,2],[6,1],[6,1],[6,1],[21,4],[10,10],[10,8],[10,4],[10,2],[10,7],[16,8],[16,7],[16,7],[34,2],[34,
performAction: function anonymous(yytext, yyleng, yylineno, yy, yystate /* action[1] */, $$ /* vstack */, _$ /* lstack */) {
/* this == yyval */

var $0 = $$.length - 1;
switch (yystate) {
case 1:

        console.log("Parser de Jison entrada corriendo");
        let raiz = new Raiz($$[$0-1]);
        this.$ = raiz;
        return raiz;

break;
case 2:

            $$[$0-1].push($$[$0]);
            this.$ = $$[$0-1];

break;
case 3:

            let lstsent = [];
            lstsent.push($$[$0]);
            this.$ = lstsent;

break;
case 4:
```

Analisis de los Datos

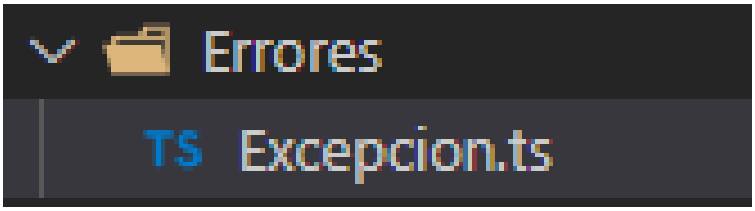
Sintáctico

Luego de que el análisis léxico terminase y enviase los datos de los errores al sistema, el programa procede a ejecutar el archivos dentro de src para el analisis sintanctico, el cual se encarga de recolectar las variables y expresiones regulares encontradas en el archivo para luego guardarlos en listas para ser analizados.



Errores

Los errores se manejan por medio un objeto que recibe los parámetros del tipo de error, ya sea léxico o sintáctico y la fila y columna donde esta el error, para luego guardar este mismo en una lista y plasmarla en un archivo "html".




```
export class Excepcion {
    /**
     *
     * @param titulo Clasifica el tipo de Excepcion
     * @param descripcion Detalle del Excepcion que
     * @param linea Linea en donde se encontro el Exc
     * @param columna Columna en donde se encontro el
     */
    constructor(
        public titulo: string,
        public descripcion: string,
        public linea: number,
        public columna: number
    ) { }
}
```

GENERAR LOS GRAFICOS

Para generar los gráficos de cada una de las expresiones regulares encontradas se utilizo Graphviz que es una librería que nos permite generar gráficos por medio de su lenguaje.

Se generan 1 graficas por medio de Graphviz.

1. La grafica del Árbol Sintáctico se genera en el archivo llamado "ASTgrafo.svg" en la carpeta "Txt"