PROYECTO 2 - MFMScript

MANUAL TÉCNICO

FACULTAD DE INGENIERÍA, USAC

NOVIEMBRE 2022

YENIFER ESTER YOC LARIOS, 201952336

OBJETIVOS

Objetivo General:

Brindar a los estudiantes del curso Introducción a Programación y computación 1 un intérprete que implemente la ejecución de pseudocodigo, para de este modo comprender las bases de la programación, así como su sintaxis y las propiedades de un lenguaje. Es por ello que se dirigió a ellos este proyecto, para fomentar su aprendizaje como estudiantes y futuros ingenieros en Ciencias y Sistemas.

Objetivos Específicos:

Presentar una interfaz interactiva, por medio de la estructura cliente - servidor y proporcionar una vista fácil de usar e interpretar.

Implementar la herramienta Jison para el análisis léxico y sintáctico.

Generar el árbol sintáctico que muestre la ejecución

Alcances:

Que pueda implementarse el uso y manipulación de la herramienta Jison como analizador léxico y sintáctico, así como interpretar semánticamente el código.

Especificación Técnica:

Server:

- **Typescript**
- **NodeJs**

Front:

- **Angular**
- **NodeJs**

Requisitos del hardware:

- Un ordenador de escritorio o portátil.

Requisitos del Software:

- Navegador web tales como: Mozilla Firefox, Safari (macOS), Microsoft Edge, Avast Secure Browser, Opera.
- Se solicita al usuario que, para poder ejecutar este programa, tenga instalado en su ordenador npm, angular y TS
- Se necesita que la resolución de la pantalla sea a colores para poder apreciar
- la interfaz como es debido.

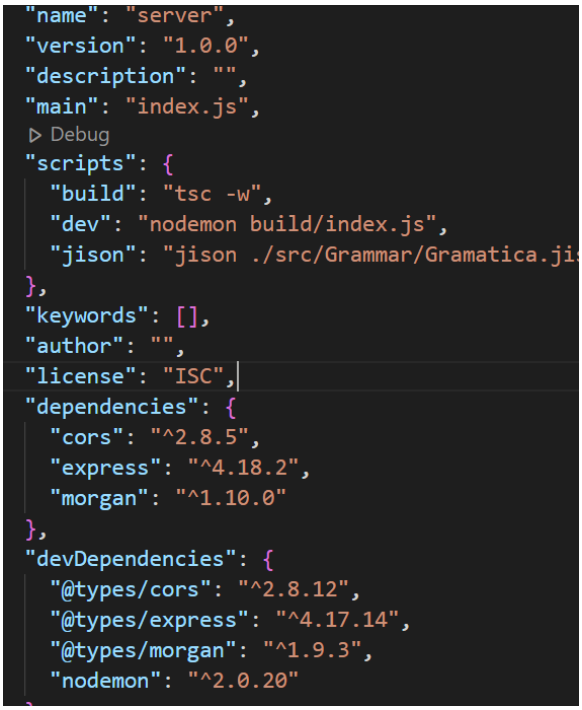
LÓGICA DEL PROGRAMA

El programa se realizó por medio del lenguaje TypeScript para implementar tanto en el servidor como en angulas las condiciones necesarias para su interacción

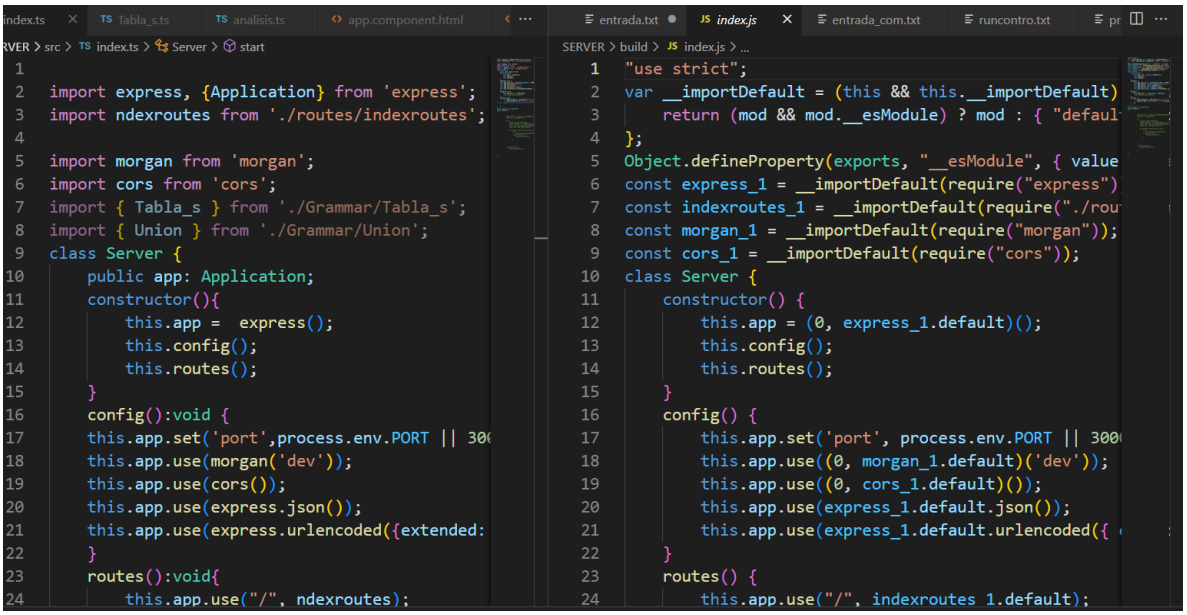
Las Estructura fue realizada de la siguiente manera:

Server:

Se realizó por medio de Node.Js para de este modo montar un servidor localmente, además de ello se utilizó TS para realizar las demás clases y por medio de la dependencia se efectuó la conversión del código TS a Js.



Por medio de la declaración “npm run dev” se llevó a cabo la escucha y ejecución de la carpeta build y conversión de código



Clases abstractas

Se crearon clases abstractas con el fin de que todas las clases que hereden de ellas sean ejecutables tanto para interpretar como para la generación del AST.

Una de estas clases fue:

```
import { Tabla_s } from "./Tabla_s"

export abstract class Instruccion {

    constructor(public linea: number, public columna: number) {
        this.linea = linea
        this.columna = columna + 1
    }

    public abstract ejecutar(tabla_sim: Tabla_s): any
    public abstract ast(): string

}
```

Estructura Cliente:

Como ya se mencionó anteriormente, la estructura del cliente se ralo con Angular y NodeJs,

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { LeerComponent } from './components/leer/leer.component';
import { ObtenerComponent } from './components/obtener/obtener.component

const routes: Routes = [{
  path : '',
  redirectTo: '/analizar',
  pathMatch: 'full'
},
{
  path : 'analizar',
  component: ObtenerComponent
},
{
  path : 'leer',
  component: LeerComponent
},



];
```

Contiene múltiples componentes entre los cuales se encuentran.

```
<header id="header" class="fixed-top ">
    <div class="container d-flex align-items-center justify-content-betw

      <h1 class="logo"><a href="index (2).html">MFMScript</a></h1>
      <!-- Uncomment below if you prefer to use an image logo -->
      <!-- <a href="index.html" class="logo"><img src="assets/img/logo.p

      <nav id="navbar" class="navbar">
        <ul>
          <li><a class="nav-link scrollto active" href="#hero">Inicio</a
          <li><a class="nav-link scrollto" href="#about">Entrada</a></li
          <li><a class="nav-link scrollto" href="#about">Consola</a></li
          <li><a class="nav-link scrollto" href="#team">Arbol</a></li>
          <li class="dropdown"><a href="#"><span>Reportes</span> <i clas
            <ul>
              <li><a href="#features">Tabla Simbolos</a></li>
              <li><a href="#features">Errores</a></li>
           <!--    <li><a href="#teamd">Manual usuario</a></li>
              <li><a href="#teamd">Manual Tecnico</a></li> -->
            </ul>
          </li>

        </ul>
        <i class="bi bi-list mobile-nav-toggle"></i>
S   OUTPUT   DEBUG CONSOLE   TERMINAL   JUPYTER
```

Este se encarga de la navegación del usuario en la página.

Y este componente se encarga de la petición al servidor y así mismo de hacer las inserciones necesarias en el documento HTML

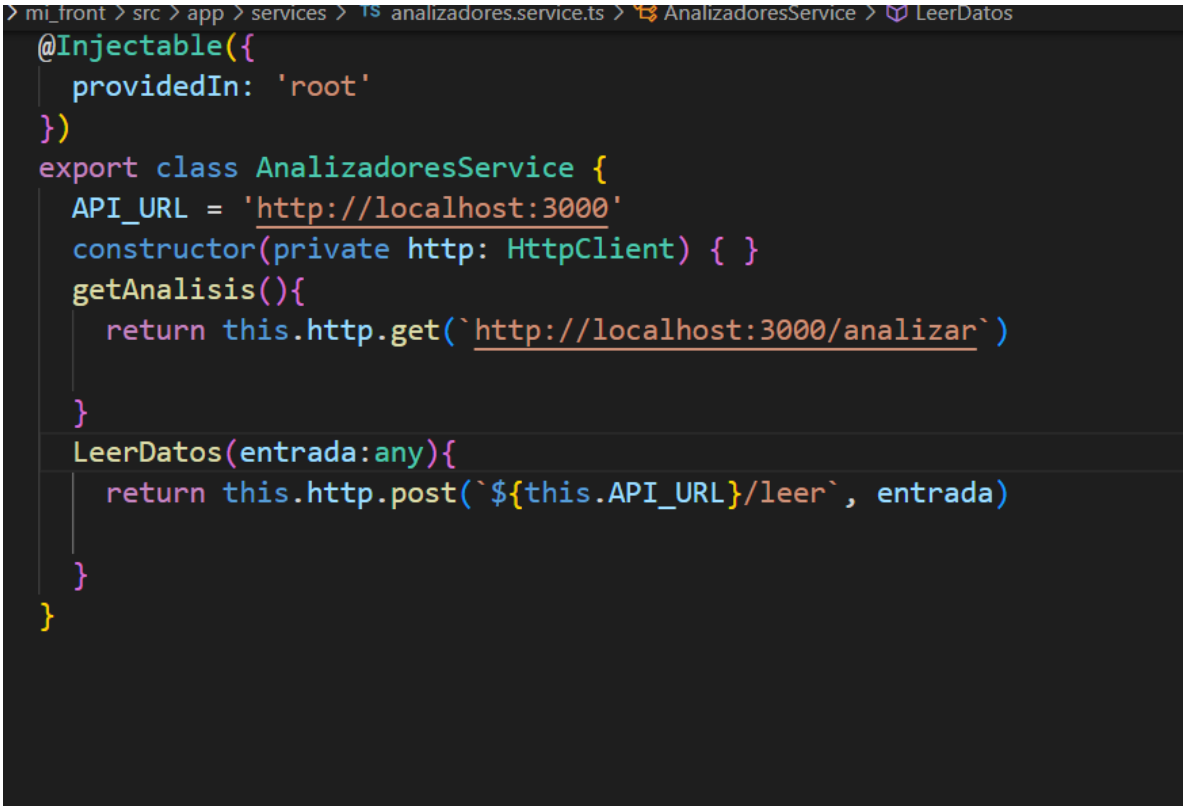
```
<main id="main">

    <!-- ======= About Section ======= -->
    <section id="about" class="about">
      <div class="container">

        <div class="section-title">

          <h3>Cargar <span>Entrada</span></h3>
          <p>El curso de Organización de Lenguajes y Compiladores 1, ha
            proyecto, requerido por la Escuela de Ciencias y Sistemas de
            consiste en crear un lenguaje de programación para los estud
            Introducción a la Programación y Computación 1</p>
        </div>
        <div class="col-sm-12">
            <div class="card">
                <div class="card-body">
                    <h5 class="card-title">CARGA</h5>
                    <div class="mb-3">
                        <label for="formFile" class="form-label" id="pru
                            formato OLC</label>
                        <input class="form-control" type="file" id="user
                    </div>
                    <button type="button" class="btn btn-primary" id = "
```

Para las peticiones y manejo de medios, así como la comunicación entre el Servidor y el Cliente, se realizó un Servicio llamado Analizadores.


mi_front > src > app > services > analizadores.service.ts > AnalizadoresService > LeerDatos
@Injectable({
providedIn: 'root'
})
export class AnalizadoresService {
API_URL = 'http://localhost:3000'
constructor(private http: HttpClient) { }
getAnalisis(){
return this.http.get(`http://localhost:3000/analizar`)
}
LeerDatos(entrada:any){
return this.http.post(`${this.API_URL}/leer`, entrada)
}
}

El analizador:

```
//palabras reservadas
'if"          {console.log("reconoci pr_si "); return 'pr_si'}
'else"          {console.log("reconoci pr_contrario"); return 'pr_contrario'}
'elif"          {console.log("reconoci pr_elif"); return 'pr_elif'}
'print"          {console.log("reconoci pr_imprimir"); return 'pr_imprimir'}
'println"          {console.log("reconoci pr_imprimir_ln"); return 'pr_imprimir_ln'}
'switch"          {console.log("reconoci pr_segun"); return 'pr_segun'}
'case"          {console.log("reconoci pr_case"); return 'pr_case'}
'default"          {console.log("reconoci pr_default"); return 'pr_default'}
'break"          {console.log("reconoci pr_break"); return 'pr_break'}
'while"          {console.log("reconoci pr_while"); return 'pr_while'}
'for"          {console.log("reconoci pr_for"); return 'pr_for'}
'do"          {console.log("reconoci pr_do"); return 'pr_do'}
'until"          {console.log("reconoci pr_until"); return 'pr_until'}
'return"          {console.log("reconoci pr_retorno"); return 'pr_retorno'}
'continue"          {console.log("reconoci pr_continue"); return 'pr_continue'}
'tolower"          {console.log("reconoci pr_minuscula"); return 'pr_minuscula'}
'toupper"          {console.log("reconoci pr_mayuscula"); return 'pr_mayuscula'}
'round"          {console.log("reconoci pr_redondear"); return 'pr_redondear'}
'void"          {console.log("reconoci pr_void"); return 'pr_void'}
'length"          {console.log("reconoci pr_longi"); return 'pr_longi'}
'typeof"          {console.log("reconoci pr_tipo"); return 'pr_tipo'}
'tostring"          {console.log("reconoci pr_a_cadena"); return 'pr_a_cadena'}
'tochararray"          {console.log("reconoci pr_arreglo"); return 'pr_arreglo'}
'push"          {console.log("reconoci pr_push"); return 'pr_push'}
'pop"          {console.log("reconoci pr_pop"); return 'pr_pop'}
'run"          {console.log("reconoci pr_run"); return 'pr_run'}
```

```
INSTRUCCIONES
    : INSTRUCCIONES INSTRUCCION  { $1.push($2); $$ = $1; }
    | INSTRUCCION              { $$ = [$1];           }
;

INSTRUCCION
    : BLOQUE               {  $$ = $1;  }
    | DECLARACION            punto_c {  $$ = $1;  }
    | CASTEO             punto_c {  $$ = $1;  }
    | ASIGNACION        punto_c {  $$ = $1;  }
    | INCREDECRE        punto_c {  $$ = $1;  }
    | PRINT punto_c {  $$ = $1;  }
    | ARRAY          punto_c {  $$ = $1;  }
    | MATRIZ          punto_c {  $$ = $1;  }
    | Asignar_arr          punto_c {  $$ = $1;  }
    | IF                {  $$ = $1;  }
    | SEGUN           {  $$ = $1;  }
    | MIENTRAS              {  $$ = $1;  }
    | PARA                  {  $$ = $1;  }
    | DOWHILE           {  $$ = $1;  }
    | DOUNTIL           {  $$ = $1;  }
    | OP_TERNARIO       punto_c {  $$ = $1;  }
    | FUNCION               {  $$ = $1;  }
    | LLAMADA       punto_c {  $$ = $1;  }
    | RETORNO       punto_c {  $$ = $1;  }
    | ARRAY_PQ      punto_c {  $$ = $1;  }
    | RUN          punto_c {  $$ = $1;  }
```

Anexos

https://github.com/YeniferYoc/OLC1-201952336/tree/master/Mi_proyecto2