

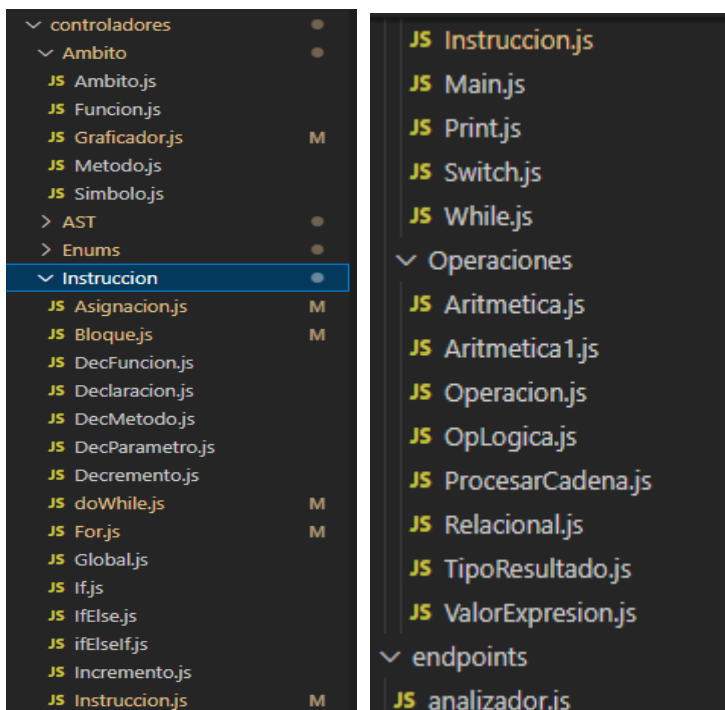
Universidad de San Carlos de Guatemala.
Facultad de ingeniería
Ingeniería en Ciencias y Sistemas
Organización de compiladores y lenguajes 1.
Aux. Kevin López.

Proyecto 2//Manual Técnico.

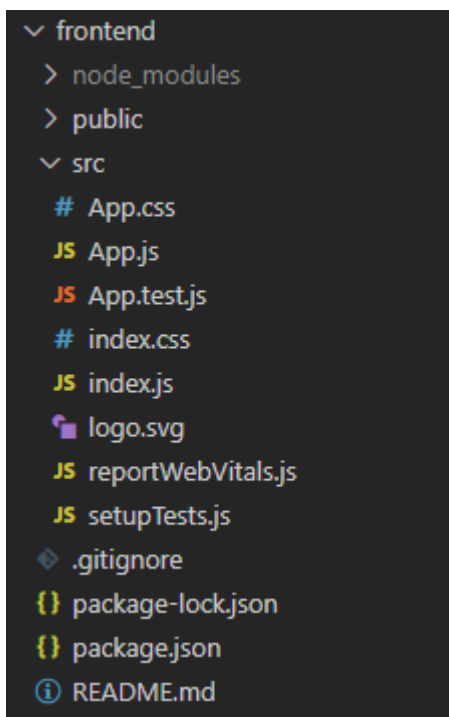
Kevin Estuardo
Secaida Molina
201602404

A continuación se muestran a grandes rasgos como esta conformado la estructura del proyecto.

Backend desarrollado en javascript.



Frontend desarrollado en node.js



Debido a la cantidad se explicara a grandes rasgos las carpetas.

Controladores, contiene todo el backend, Ambito maneja la conexión previa a la conexión back-front, Instrucción contiene constructores para realizar las diferentes operaciones, Operación contiene todo lo que es de Operaciones, aritmética, lógica, relacional, mientras que en instrucciones se manejan todo lo que son sentencias y ejecución del programa, vemos una carpeta llamada AST que es el la cual se almacenan las imágenes creadas.

Y como ultimo punto del backend explicaremos el analizador.json el cual fue el archivo base para leer todo el código generado por el backend, cabe mencionar que hay otra carpeta llamada endpoints que contiene un archivo llamado analizador.js este es el encargado de conectar el backend con el frontend.

```
1  /* lexical grammar */
2  %lex
3  %options case-insensitive
4  %%
5
6  \s+          /* skip whitespace */
7  "/*".*       //comentario lineal
8  [\/][^"]*(?![\/])["]*+([\/][^"]*(?![\/])["]*+)*[\/] // comentario multiple líneas
9
10
11
12  [0-9]+(("[0-9]+")?\b) return 'NUMBER'
13  "clase"              return 'clase'
14  "double"             return 'decimal'
15  "string"             return 'cadena'
16  "boolean"            return 'booleano'
17  "true"               return 'Rtrue'
18  "false"              return 'Rfalse'
19  "int"                return 'intpr'
20  "char"               return 'Rchar'
21  "print"              return 'Rprint'
22  "while"              return 'while'
23  "for"                return 'for'
24  "main"               return 'Rmain'
25  "if"                 return 'if'
26  "new"                return 'Rnew'
27  "break"              return 'break'
28  "continue"           return 'continue'
29  "return"             return 'return'
30  "else"               return 'else'
```

```
"|"          return 'or'          ([a-zA-Z])([a-zA-Z0-9_])*          return 'identificador'
"&&"         return 'and'          ["\""]([^\"])*["\""]          return 'string'
"=="         return 'igual'        ["'"]([^\"])*["'"]          return 'char'
"!="         return 'diferente'
"<="         return 'menorigual'
">="         return 'mayorigual'
">"          return 'mayor'
"<"          return 'menor'
","          return 'coma'
";"          return 'ptcoma'
"{"          return 'llaveA'
"}"          return 'llaveC'
"*"          return 'multi'
"/"          return 'div'
"--"         return 'decremento'
"--"         return 'menos'          /* operator associations and precedence */
"++"         return 'incremento'
"+"          return 'suma'          %left 'interrogacion'
"^"          return 'exponente'    %left 'or'
"!"          return 'not'          %left 'and'
%"          return 'modulo'        %right 'not'
"("          return 'parA'          %left 'igual' 'diferente' 'menor' 'menorigual' 'mayor' 'mayorigual'
")"          return 'parC'          %left 'suma' 'menos'
"?"          return 'interrogacion' %left 'multi' 'div' 'modulo'
":"          return 'dospuntos'    %left 'exponente'
"."          return 'corA'          %left umenos
"["          return 'corC'          %right 'incremento' 'decremento'
```

```

INICIO: OPCIONESCUERPO EOF{return $1;}
;

OPCIONESCUERPO: OPCIONESCUERPO CUERPO {$1.push($2); $$=$1;}
| CUERPO {$$=[ $1];}
;

CUERPO: DEC_VAR {$$=$1}
| DEC_MET {$$=$1}
| AS_VAR ptcoma {$$=$1}
| MAIN {$$=$1}
| DEC_FUN {$$=$1}
| DEC_ESTRUCTURA {$$=$1}
| MODVECTOR {$$=$1}
;

```

Debido a que el archivo es demasiado extenso se deja a discreción del siguiente desarrollador leerlo.

Analizador.js (controlador del backend)

```

module.exports=(parser, app)=>{

  var prueba;
  var ast;
  var grafica;
  app.post('/analizar',(req,res)=>{
    prueba = req.body.entrada
    //console.log(prueba)
    ast = parser.parse(prueba);
    grafica = new Graficador(ast);
    //console.log(ast)
    const AmbitoGlobal= new Ambito(null,"Global")
    var cadena = Global(ast, AmbitoGlobal)

    var resultado= {
      arbol: ast,
      resultado:cadena
    }

    res.send(resultado)

  });

  app.get('/CrearAST',(req,res)=>{
    dot = grafica.graficar()

    fs.writeFile('./controladores/AST/AST.dot', dot, function (error) {
      if (error) {
        console.log(error);
      } else {
        console.log("Archivo creado");
      }
    });

    const { exec } = require('child_process');
    exec('dot -Tpng ./controladores/AST/AST.dot -o ./controladores/AST/AST.png', (error, stdout, stderr) => {
      if (error) {
        console.log(`error: ${error.message}`);
        res.send(error.message)
      }
      if (stderr) {
        console.log(`stderr: ${stderr}`);
      }
    });
  });
};

```

Parte del código responsable del manejo del frontend.

```
function App() {
  const [code, setCode] = useState('');
  const [resultado, setResultado] = useState('');
  function analizar(){
    axios.post('http://localhost:5000/analizar', JSON.stringify({
      entrada: code
    }},{headers:{"Content-Type" : "application/json"}})
    .then(function (response) {
      console.log(response);
      setResultado(response.data.resultado);
    })
    .catch(function (error) {
      console.log(error);
    });
  }

  function crearAst(){
    axios.get('http://localhost:5000/CrearAST')
    .then(function (response) {
      console.log(response);
    })
    .catch(function (error) {
      console.log(error);
    });
  }

  function SubirArchivo(){
    var SubirArch = document.getElementById('btnSubArch');
    var jsonFileInput = document.getElementById('json-file-input');

    SubirArch.addEventListener('click', function () {
      jsonFileInput.click();
    });

    jsonFileInput.addEventListener('change', function () {
      var file = jsonFileInput.files[0];
      var reader = new FileReader();
      reader.onload = function (e) {
        var fileContent = e.target.result;
        // Hacer lo que necesites con el contenido del archivo
        //console.log(fileContent,"hola");
        setCode(fileContent);
      };
      reader.readAsText(file);
      //console.log("hiloo", file);
    });
  }
}
```