

MANUAL TÉCNICO

OLC1 - PROYECTO 1

Samuel Isaac Pérez Pérez
201902308



Analizador con la herramienta JISON

Podemos colocar una área de importaciones de la siguiente manera:

```
src > analizador > analizador.jison
1  /* ===== IMPORTACIONES ===== */
2  %{
3      const TIPO_OPERACION = require('../controllers/enums/tipoOperacion')
4      const TIPO_VALOR = require('../controllers/enums/tipoValor')
5      const TIPO_DATO = require('../controllers/enums/tipoDato')
6      const INSTRUCCION = require('../controllers/instrucciones/instruccion')
7      const VarStatic = require('../controllers/simbolos/static')
8  %}
9
```

Definición del análisis léxico (Tokens):

```
11 /* ===== LÉXICO ===== */
12
13 /* lexical grammar */
14 %lex
15 %options case-insensitive
16 %%
17
18 \s+                /* skip whitespace */
19 "/*".*            // comentario lineal
20 [/*][^*]*[*](/*)+([/*][^*]*[*])*(/*) // comentario multiple línea
21
22 "int"              return 'Rint'
23 "double"          return 'Rdouble'
24 "bool"            return 'Rboolean'
25 "char"            return 'Rchar'
26 "string"          return 'Rstring'
27
28 "true"            return 'Rtrue'
29 "false"           return 'Rfalse'
30
31 "Console"         return 'Rconsole'
32 "Write"           return 'Rwrite'
33 "if"              return 'Rif'
34 "else"            return 'Relse'
35 "while"           return 'Rwhile'
36 "do"              return 'Rdo'
37 "void"            return 'Rvoid'
38 "main"            return 'Rmain'
39 "return"          return 'Rreturn'
40 "break"           return 'Rbreak'
41 "continue"        return 'Rcontinue'
42 "for"             return 'Rfor'
43 "switch"          return 'Rswitch'
44 "case"            return 'Rcase'
45 "default"         return 'Rdefault'
```

```

47  [0-9]+(".[0-9]+)\b      return 'decimal'
48  "."                    return 'punto'
49  [0-9]+\b               return 'entero'
50  "=="                   return 'igualigual'
51  "!="                   return 'diferente'
52  "<="                   return 'menorIgual'
53  "<"                   return 'menor'
54  ">="                   return 'mayorIgual'
55  "="                    return 'igual'
56  ">"                   return 'mayor'
57  ","                    return 'coma'
58  ";"                    return 'ptcoma'
59  ":"                    return 'dospuntos'
60  "||"                   return 'or'
61  "&&"                   return 'and'
62  "{"                    return 'llaveA'
63  "}"                    return 'llaveC'
64  "*"                    return 'por'
65  "/"                    return 'div'
66  "--"                   return 'menosmenos'
67  "++"                   return 'masmas'
68  "-"                    return 'menos'
69  "+"                    return 'mas'
70  "^"                    return 'exponente'
71  "!"                    return 'not'
72  "%"                    return 'modulo'
73  "("                    return 'parA'
74  ")"                    return 'parC'
75  "["                    return 'corchA'
76  "]"                    return 'corchC'
77
78  ([a-zA-Z])([a-zA-Z0-9_])*      return 'identificador'
79  ["\""](^\"")*["\""]          return 'string'
80  ["\'"](^\'")*["\'"]          return 'char'
81
82  <<EOF>>                  return 'EOF'      // Fin de archivo

```

Manejo de errores léxicos:

```

83  {
84      console.error('Este es un error léxico: ' + yytext + ', en la línea: ' + yylloc.first_line + ', en la columna: ' + yylloc.first_column);
85      VarStatic.listaErrores.push({tipoError: 'Léxico', línea: yylloc.first_line, columna: yylloc.first_column, descripcion: 'El carácter ' + yytext
86  }
87  }
88  /lex

```

Definición del análisis sintáctico (Precedencia y Gramática):

```
91  /* ===== SINTÁCTICO ===== */
92
93  /* operator associations and precedence */
94  %left 'or'
95  %left 'and'
96  %right 'not'
97  %left 'igualigual' 'menor' 'menorIgual' 'mayor' 'mayorIgual' 'diferente'
98  %left 'mas' 'menos'
99  %left 'por' 'div' 'modulo'
100 %nonassoc 'exponente'
101 %left umenos
102
103 %start INICIO
104
105 %% /* language grammar */
106
107 INICIO: INSTRUCCIONES EOF{return $1;}
108 ;
109
110
111 INSTRUCCIONES: INSTRUCCIONES INSTRUCCION {$$ = $1; $1.push($2);}
112 | INSTRUCCION {$$ = [$1];}
113 ;
```

Instrucciones del lenguaje y recuperación de errores sintácticos:

```
119 INSTRUCCION: DEC_VAR {$$ = $1;} // DECLARACION DE VARIABLE
120 | ASIG_VAR {$$ = $1;} // ASIGNACION DE VARIABLE
121 | PRINT {$$ = $1;}
122 | IF {$$ = $1;}
123 | WHILE {$$ = $1;}
124 | DOWHILE {$$ = $1;}
125 | METODO {$$ = $1;}
126 | FUNCION {$$ = $1;}
127 | MAIN {$$ = $1;}
128 | RETURN {$$ = $1;}
129 | BREAK {$$ = $1;}
130 | CONTINUE {$$ = $1;}
131 | FOR {$$ = $1;}
132 | SWITCH {$$ = $1;}
133 | error DEC_VAR { console.error('DEC_VAR - Este es un error sintáctico: ' + yytext + ', en la línea: ' + this._$.first_line + ', en la columna: ' + this._$.first_column + '); }
134 | error ASIG_VAR { console.error('ASIG_VAR - Este es un error sintáctico: ' + yytext + ', en la línea: ' + this._$.first_line + ', en la columna: ' + this._$.first_column + '); }
135 | error PRINT { console.error('PRINT - Este es un error sintáctico: ' + yytext + ', en la línea: ' + this._$.first_line + ', en la columna: ' + this._$.first_column + '); }
136 | error IF { console.error('IF - Este es un error sintáctico: ' + yytext + ', en la línea: ' + this._$.first_line + ', en la columna: ' + this._$.first_column + '); }
137 | error WHILE { console.error('WHILE - Este es un error sintáctico: ' + yytext + ', en la línea: ' + this._$.first_line + ', en la columna: ' + this._$.first_column + '); }
138 | error DOWHILE { console.error('DOWHILE - Este es un error sintáctico: ' + yytext + ', en la línea: ' + this._$.first_line + ', en la columna: ' + this._$.first_column + '); }
139 | error METODO { console.error('METODO - Este es un error sintáctico: ' + yytext + ', en la línea: ' + this._$.first_line + ', en la columna: ' + this._$.first_column + '); }
140 | error FUNCION { console.error('FUNCION - Este es un error sintáctico: ' + yytext + ', en la línea: ' + this._$.first_line + ', en la columna: ' + this._$.first_column + '); }
141 | error MAIN { console.error('MAIN - Este es un error sintáctico: ' + yytext + ', en la línea: ' + this._$.first_line + ', en la columna: ' + this._$.first_column + '); }
142 | error FOR { console.error('FOR - Este es un error sintáctico: ' + yytext + ', en la línea: ' + this._$.first_line + ', en la columna: ' + this._$.first_column + '); }
143 | error SWITCH { console.error('SWITCH - Este es un error sintáctico: ' + yytext + ', en la línea: ' + this._$.first_line + ', en la columna: ' + this._$.first_column + '); }
144 ;
```

Para compilar el analizador se debe abrir una terminal en la carpeta de la API y ejecutar el comando **npm start**.

Descripción de la API

Se utilizó Node.js con el framework Express para crear un servidor, con el cual podemos servir el resultado del análisis de la entrada de texto, en formato JSON a cualquier cliente por medio de peticiones HTTP.

```
JS app.js X
src > JS app.js > ...
1  const express = require('express')
2  const morgan = require('morgan')
3  const cors = require('cors')
4
5  const app = express()
6
7
8  //Middlewares
9  app.use(express.json())
10 app.use(express.urlencoded({extended: false}))
11 app.use(morgan('dev'))
12 app.use(cors())
13
14
15 app.get('/', (req, res) => {
16   res.send('API - OLC1 Proyecto 1')
17 })
18
19
20 //Routes API
21 app.use(require('./routes/analisis.route'))
22
23
24 app.listen(3000, () => {
25   console.log('Server on port 3000')
26 })
```

Para levantar el servidor, se debe abrir una terminal en la carpeta de la API y ejecutar el comando **npm run dev**.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\samue\OneDrive\Documentos\GitHub\OLC1_Proyecto1\Backend> npm run dev

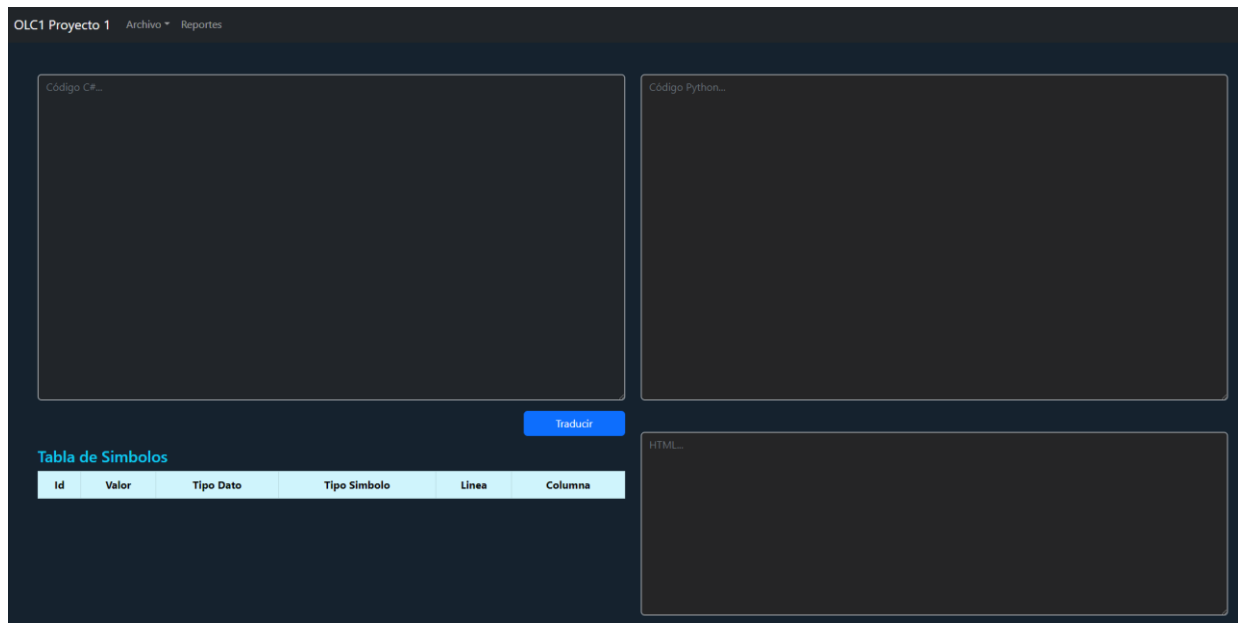
> backend@1.0.0 dev
> nodemon src/app.js

[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/app.js`
Server on port 3000
```

Descripción del Frontend

Se utilizó el framework Angular para crear la interfaz gráfica y poder realizar peticiones HTTP a la API. Solicitando como entrada código C# y mostrando como salida su traducción a código Python.

Para levantar el frontend, se debe abrir una terminal en la carpeta de la raíz del proyecto y ejecutar el comando **ng serve**.



Componentes. Se utilizó un único componente para mostrar los elementos necesarios e interactuar con el usuario.

```
TS app.component.ts M X app.component.html
src > app > TS app.component.ts > AppComponent > limpiar
1 import { Component, ViewChild, ElementRef } from '@angular/core';
2 import { TraductorService } from './services/traductor.service';
3
4 @Component({
5   selector: 'app-root',
6   templateUrl: './app.component.html',
7   styleUrls: ['./app.component.css']
8 })
9
10 export class AppComponent {
11
12   @ViewChild('textEntrada') textEntrada: ElementRef;
13   @ViewChild('textSalidaPython') textSalidaPython: ElementRef;
14   @ViewChild('textSalidaHTML') textSalidaHTML: ElementRef;
15
16   tablaSimbolos = []
17   listaErrores = []
18
19   constructor(private traductorService: TraductorService){
20
21   }
22
23   traducir(){
24     const entrada = this.textEntrada.nativeElement.value
25     if(entrada == "") return
26
27     this.traductorService.createTraduccion({entrada}).subscribe(
28       res => {
29         console.log(res)
30         this.textSalidaPython.nativeElement.value = res.salidaPython
31         this.textSalidaHTML.nativeElement.value = res.salidaHTML
32         this.tablaSimbolos = res.tablaSimbolos
33         this.listaErrores = res.listaErrores
34       },
35       error => console.log(error)
36     )
37   }
38 }
```

Funcionalidad Principal

Al darle click al botón traducir, se envía el contenido del textArea del código de C# a la API y se pinta la respuesta brindada en los textArea correspondientes, así como también en la tabla de símbolos y errores.

OLC1 Proyecto 1 Archivo Reportes

```
int a =4;

string v_1="Marcos Alberto Santos Aquino";

bool v_2= true;

bool v_3 = false;

char v_5= '%';
double l=45.53232432999999;

void id (int a, int b , int c){

if(a>b && b<c){

Console.WriteLine("B esta en los limites exactos: ->" + b);

}
```

Traducir

Tabla de Símbolos

| Id | Valor | Tipo Dato | Tipo Símbolo | Línea | Columna |
|-----|--------------------------------|-----------|--------------|-------|---------|
| a | 4 | ENTERO | DEC_VARIABLE | 1 | 1 |
| v_1 | "Marcos Alberto Santos Aquino" | CADENA | DEC_VARIABLE | 4 | 1 |
| v_2 | true | BOOLEANO | DEC_VARIABLE | 6 | 1 |
| v_3 | false | BOOLEANO | DEC_VARIABLE | 8 | 1 |
| v_5 | '%' | CHAR | DEC_VARIABLE | 10 | 1 |

```
a = 4
v_1 = "Marcos Alberto Santos Aquino"
v_2 = true
v_3 = false
v_5 = '%'
l = 45.53232432999999
def id(a, b, c):
    if a > b and b < c:
        print("B esta en los límites exactos: ->", b)
    else:
        print("B no esta aceptada")

    return

def main():
    def switch(valor):
        switcher = {
            1: precio = 55
        }
        if precio > 30 or precio < 100 and precio >= 0:
            print("Datos aceptado", " si si", " seguimos")
```

```
B esta en los limites exactos: ->  b
B no esta aceptada
Datos aceptado  si si  seguimos
El precio es 0:
No válido. Escoja 1, 2, o 3.
el valor de a es:  a
el valor de a es:  a
el valor de a es:  a
el valor de a es:  a
```

Tabla de Errores

| Tipo Error | Línea | Columna | Descripción |
|------------|-------|---------|-------------------------------|
| Sintáctico | 33 | 1 | Error en Instrucción: FUNCION |