Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización de Lenguajes y Compiladores 1

Manual Técnico

Proyecto 2

Gerson Alejandro Beltetón Urbina

201807228

22/05/2020

Detector de Copias

Esta aplicación fue desarrollada en una api rest que conlleva tanto una sección del cliente como un servidor. El cliente fue desarrollado utilizando go, javascript y hmtl, mientras el servidor fue realizado con typescript, express, nodejs.

Servidor:

El servidor se encarga de analizar la cadena de entrada, el cliente se la envía y el servidor la recibe, realiza un análisis léxico y sintáctico utilizando la herramienta jison. Genera una lista tokens, y errores léxicos y sintácticos en caso hubieran. Al mismo tiempo va llenando una sería de listas que conforman el AST, posteriormente con el AST tanto de la entrada original, como de la entrada copia, se comparan en busca de coincidencias en clase, métodos y variables. Se almacena esta información y se la envía al cliente en el momento que este la solicita.

```
%lex
%options case-insensitive
numero [0-9]+;
decimal {numero}("."{numero});
cadena (\"[^"]*\");
caracter ("'")[^"]("'");
comentario("/""/")("/")*(.|{identificador}|{numero}|{decimal}|{cadena})
comentarioML "/*"(.|\n|\r)*?"*/"
identificador ([a-zA-Z_])[a-zA-Z0-9_]*;


%%

\s+                   /* skip whitespace */
{cadena}              return 'cadena';
{caracter}            return 'caracter';
{comentario}          return 'comentario';
{comentarioML}        return 'comentarioML';
"++"                  return '++';
"--"                  return '--';
"=="                  return '=='
"<="                  return '<='
">="                  return '>='
"!="                  return '!='
"<"                   return '<'
">"                   return '>'
"&&"                  return '&&'
"||"                  return '||'
"!"                   return '!'
"*"                   return '*'
"/"                   return '/'
"%"                   return '%'
```

```
INICIO : CLASE EOF {$$ = new Tree($1); return $$;}
;



ASIGNACION : 'identificador' '=' EXPRECION ';' {$$ = new Asignacion($1, $3, _$.first_line, _$.first_colu
           ;
COMENTARIO: 'comentario'{ $$ = new Comentario($1, _$.first_line, _$.first_column); }
        |'comentarioML' { $$ = new Comentario($1, _$.first_line, _$.first_column); }
        ;
CONDICION:EXPNUM '>' EXPNUM      { $$ = new Comparacion($1, $3, '>', _$.first_line, _$.first_column); }
        | EXPNUM  '<' EXPNUM     { $$ = new Comparacion($1, $3, '<', _$.first_line, _$.first_column); }
        | EXPNUM '>=' EXPNUM     { $$ = new Comparacion($1, $3, '>=', _$.first_line, _$.first_column); }
        | EXPNUM '<=' EXPNUM     { $$ = new Comparacion($1, $3, '<=', _$.first_line, _$.first_column); }
        | EXPNUM '==' EXPNUM     { $$ = new Comparacion($1, $3, '==', _$.first_line, _$.first_column); }
        | EXPNUM '!=' EXPNUM     { $$ = new Comparacion($1, $3, '!=', _$.first_line, _$.first_column); }
        | 'true'                 { $$ = new Primitivo(new Type(types.BOOLEAN), true, _$.first_line, _$.fi
        | 'false'               { $$ = new Primitivo(new Type(types.BOOLEAN), false, _$.first_line, _$.fi
        ;

CASE :'case'  EXPRECION ':' SENTENCIASREC  {$$ = new Case('case', $2, $4)}
     |'case'  EXPRECION ':' SENTENCIASREC 'break' ';'{$$ = new Case('case', $2, $4)}
     | 'default' ':' SENTENCIASREC {$$ = new Case('default', null, $3)}
     | 'default' ':' SENTENCIASREC 'break' ';'{$$ = new Case('default', null, $3)}

    ;

DECASIG: DECLARACIONVAR {$$ = $1}
        |ASIGNACION {$$ = $1}
        ;

DECLARACIONVAR: TIPO 'identificador' LISTAID {$$ = new Declaracion($1, $2, $3,  _$.first_line, _$.first_
```

```
        console.log(this.entrada)
    }

    config():void{

        this.router.get('/',(req, res)=> res.send("servidor :D"))
        this.router.get('/token',(req, res)=> res.send(this.lexico()))
        this.router.get('/errlex',(req, res)=> res.send(this.getErrLex()))
        this.router.get('/ast',(req, res)=> res.send(this.sintactico()))
        this.router.get('/errsin',(req, res)=> res.send(this.getErrSin()))
        this.router.get('/copiaclase',(req, res)=> res.send(this.listaClaseToString()))
        this.router.get('/copiametodo',(req, res)=> res.send(this.listaMetodoToString()))
        this.router.get('/copiavariable',(req, res)=> res.send(this.listaVariableToString()))
        this.router.post('/entrada',(req, res)=> {
            var body = req.body
            var dat = {
                ent:body.ent,
                ent2:body.ent2,
            }
            console.log(dat.ent)
            this.entrada = dat.ent
            this.entrada2 = dat.ent2

            res.send('entrada reeecibida')


        })
```

```
copiaVariable(){

    for(var i = 0; i < this.analisis.instructions.length; i++){
        if(this.analisis.instructions[i].tipoInstruccion == "clase"){
            for(var j = 0; j <  this.analisis2.instructions.length; j++){
                if(this.analisis2.instructions[j].tipoInstruccion == "clase"){
                    if(this.analisis.instructions[i].identificador == this.analisis2.instructions[j].identificad
                        for(var k = 0; k < this.analisis.instructions[i].instrucciones.length; k++){

                            if(this.analisis.instructions[i].instrucciones[k].tipoInstruccion == "metodo"){
                                for(var l = 0; l < this.analisis2.instructions[j].instrucciones.length; l++){
                                    if(this.analisis2.instructions[j].instrucciones[l].tipoInstruccion=="metodo"

                                        if(this.analisis.instructions[i].instrucciones[k].identificador == this.
                                            if(this.analisis.instructions[i].instrucciones[k].tipo.type == this.
                                                for(var m = 0; m < this.analisis.instructions[i].instrucciones[k
                                                    if(this.analisis.instructions[i].instrucciones[k].instruccio
                                                        for(var n = 0; n < this.analisis2.instructions[j].instru
                                                            if(this.analisis2.instructions[j].instrucciones[l].i
                                                                if(this.analisis.instructions[i].instrucciones[k
                                                                    if(this. analisis.instructions[i].instruccio
                                                                        console.log("iguales");
                                                                        this.listaVariable.push(new copiaVariabl
                                                                        this.analisis.instructions[i].instruccio
                                                                        this.analisis.instructions[i].identifica
                                                                    }
                                                                }
                                                            }
```

Cliente:

Este es la conexión directa con el usuario, recibe la cadena de entrada introducida por el usuario y es enviada la información al servidor. Hace una constante llamada al servidor en busca de la información, para levantar el servidor y las vistas se utilizó Go con templates de html, las peticiones fueron realizadas utilizando javascript.

```
var txtEntrada = document.getElementById("txtEntrada")
var txtEntrada2 = document.getElementById("txtEntrada2")

$("#button").click(function () {
  $.ajax({
    url: root + '/entrada',
    method: 'POST',
    dataType: 'json',
    data: {
      ent: txtEntrada.value,
      ent2: txtEntrada2.value
    }
  }).then(function (data) {

  })

  $.ajax({

    url: root + "/copiaclase",
    method: 'GET',
  }).then(function (data) {
    var texto = data
    console.log(texto)
    document.getElementById("cc").innerText = texto

  })

  $.ajax({

    url: root + "/copiametodo",
    method: 'GET',
```

client > main.go

```go
package main


import (
    "fmt"
    "html/template"
    "net/http"
)


func index(w http.ResponseWriter, r *http.Request) {
    t := template.Must(template.ParseFiles("index.html"))
    t.Execute(w, nil)
}

func main() {

    http.HandleFunc("/", index)
    fmt.Println("Servidor corriendo en el puerto: 8100")
    http.ListenAndServe(":8100", nil)
}
```