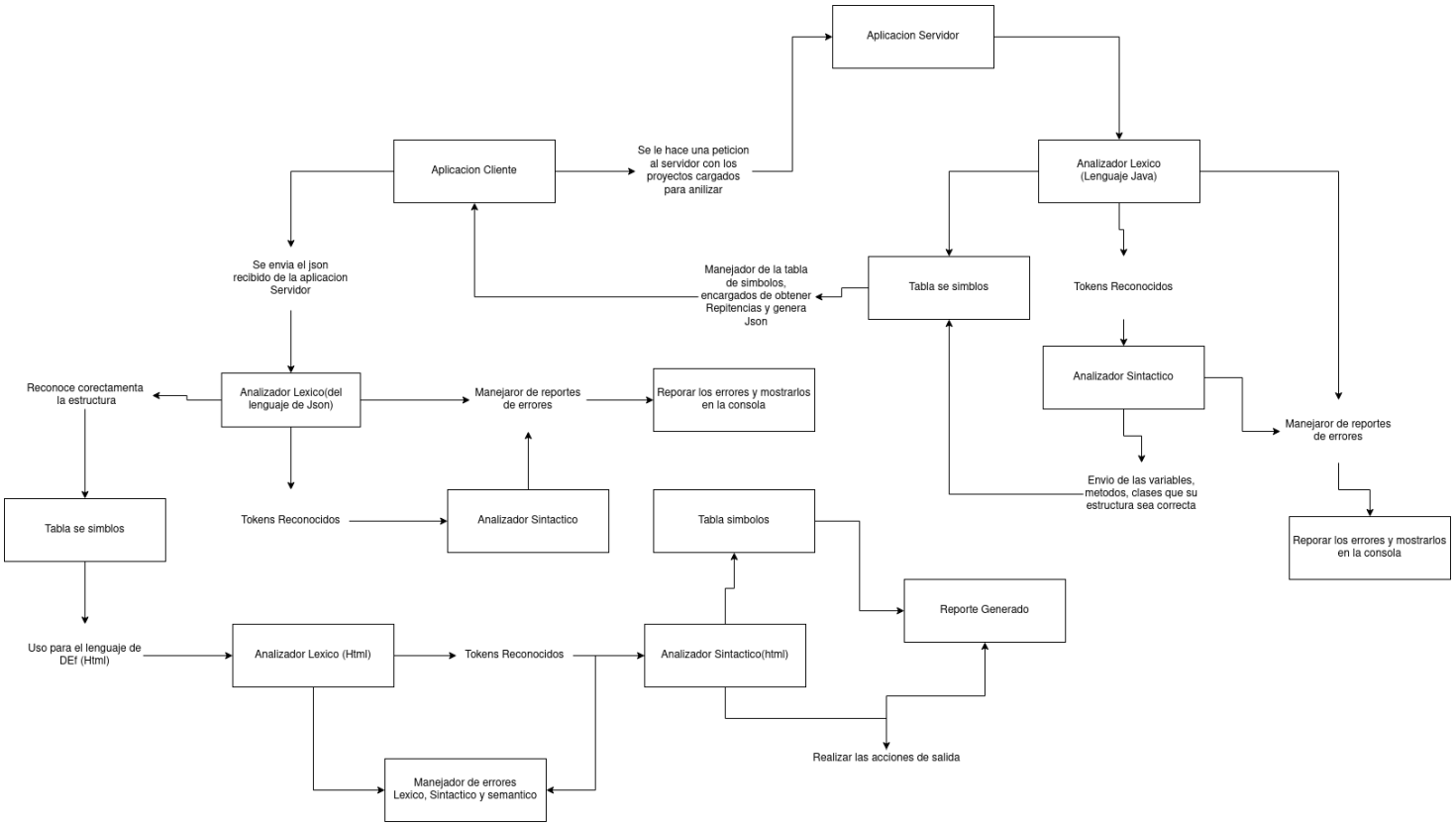
Universidad de San Carlos de Guatemala USAC.
División de Ciencias de la Ingeniería.
Centro Universitario de Occidente CUNOC.
Organización de lenguajes y compiladores
Ing. Moises

Estudiante ***Carnet***
Elvis Lizandro Aguilar Tax *201930304*

Manual Tecnico

1. *Organización del proyecto*



2. *Lenguaje Java*

Analisi de la gramática de analizador Léxico

—Palabras Reservadas:

IMPORT="import", PRIVATE = "private", PUBLIC = "public", PROTECTED = "protected"
FINAL = "final", CLASS="class", INT = "int", BOOLEAN = "boolean", STRING = "String"
CHAR = "char", DOUBLE = "double", IF = "if", ELSE = "else", FOR = "for", WHILE = "while", DO = "do , SWITCH = "switch", BREAK="break" ,RETURN = "return"
NUEVO="new", DEFAULT = "default", CASO = "case", TRUE = "true", FALSE = "false"

–Identificadores:

L*os identificadores fueros reconocido válidos como las siguientes expresiones:*
IDD = ({LETRA}|{DIAGONAL})({LETRA}|{ENTERO}|)({LETRA}|{ENTERO}))|{LETRA})*

—OperadoreAlgebraicos

signos conocidos como operadores de forma — (+|-||/)*

—Signos de Puntuación

signos conocidos como puntuación de forma — (;|,|:|.)

—Signo de agrupación

signos conocidos como agrupación de forma — ([|]|{|})

—Comentario Bloque
*signos conocidos como operadores de forma — (// o /**/ seguida de cualquier expresión)*

Gramática para el analizador sintáctico:
El analizador sintáctico, conocido como parser, o en el lenguaje Java la Herramienta de Parseo(CUP), se encarga de plantear la validez sintáctica por medio de revisión de gramáticas que representan un una secuencia de tokens enviadas por el analizador léxico, revisando el orden de las mismas y que la expresión sea reconocida de acuerdo con las reglas gramaticales que se estén implementando.

Gramática (G)
G(N,T,P,S)
- *N = No terminales*
- *T = Terminales*
- *P = Reglas de Producción*
- *S = Símbolo Inicial*

N —- Símbolos no Terminales:
- *inicio, item_punto, importacion, def_clase, def_metodo, def_variable, variables, items_coma, items, fin_asignar, asignacion*
- *tipo, visibilidad, sentencias, sentencia, sentencias_global, sentencia_global, def_if, def_if_else, def_for, acceso_termin_var_metod;*
- *def_while, def_do_while, def_switch, cases, caso_sw, factor_casos, terminal_casos, salid, retorno, condition, logica, sentencias_switch;*
- *agrupation, def_and, def_or, condition_for, operation_for, constructor, parametros, llamada_metod, parametros_llamada;*
- *recurcion_new, increment, mas_igual, var_llamad_asigan, iddOption, siguiente, var_iterador, e_inicial, t_production, e_pri, t_pri, f_terminal;*

T —- Símbolos Terminales:
- *ENTERO, DECIMAL, LLAVEA, LLAVEC, PARENTESISA, PARENTESISC, PUNTO,COMA,PUNTOCOMA, DEFAULT;*
- *DOPUNTO, MENOS, MAS, POR, DIVISION, IGUAL,EQUALS,OR,AND,MENORQ,MAYORQ,MAYOROI,MENOROI, CASO;*
- *NOTEQUALS, IMPORT, IDD, PRIVATE, PUBLIC, PROTECTED, FINAL, CLASS, INT, BOOLEAN, TRUE, CADENA;*
- *STRING, NEGATION, CHAR, CARACTER, DOUBLE, IF, ELSE, FOR, WHILE, DO, SWITCH, BREAK, RETURN, NUEVO, FALSE;*

P — Reglas de Producción:
inicio ::= importacion def_clase ;
importacion ::= IMPORT IDD PUNTO item_punto

```
        | IMPORT IDD PUNTOCOMA importacion                    |
        | error PUNTOCOMA                          ;


item_punto ::= IDD PUNTO item_punto
        | IDD PUNTOCOMA importacion
        | POR PUNTOCOMA importacion
        | error PUNTOCOMA                          ;

def_clase ::= visibilidad comodin_Class LLAVEA sentencias_global LLAVEC
        | FINAL comodin_Class LLAVEA sentencias_global LLAVEC
        | comodin_Class LLAVEA sentencias_global LLAVEC
        ;

comodin_Class ::= CLASS IDD:nomClase
                ;

items ::= IGUAL asignacion PUNTOCOMA
        | IGUAL asignacion COMA items_coma
        | COMA items_coma
        | PUNTOCOMA
        | error  IDD
        ;

fin_asignar ::= FINAL
        |
        ;

asignacion ::= condition
        | NUEVO IDD PARENTESISA parametros_llamada PARENTESISC
        | NUEVO IDD PARENTESISA  PARENTESISC
        | error  PUNTOCOMA
        ;

tipo ::= INT
        | BOOLEAN
        | DOUBLE
        | STRING
        | CHAR
        ;

visibilidad ::= PUBLIC fin_asignar
                | PROTECTED fin_asignar
```

```
                | PRIVATE fin_asignar
                ;
 sentencias ::= sentencia sentencias
        |
        | error  sentencias
        ;

 sentencias_global ::= sentencia_global sentencias_global
        |
        ;

 sentencia_global ::= constructor
        | def_variable
        | def_metodo
        ;

sentencia ::=  def_if
        | def_if_else
        | def_while
        | def_do_while
        | def_for
        | FINAL variables
        | salid
        | variables
        | IDD iddOption
        | IDD:tipoVar items_coma
        | def_switch
        | FINAL IDD:tipoVar items_coma
        ;

 def_if_else ::= def_if ELSE LLAVEA sentencias LLAVEC
        ;

 def_for ::= FOR PARENTESISA condition_for PARENTESISC  LLAVEA sentencias LLAVEC
        ;

 def_while ::= WHILE PARENTESISA condition PARENTESISC  LLAVEA sentencias
LLAVEC
        ;

 def_do_while ::= DO LLAVEA sentencias LLAVEC WHILE PARENTESISA condition
PARENTESISC PUNTOCOMA
        ;
```

def_switch ::= SWITCH PARENTESISA IDD PARENTESISC LLAVEA cases LLAVEC
| SWITCH PARENTESISA IDD llamada_metod PARENTESISC LLAVEA cases LLAVEC
;

cases ::= caso_sw salid cases
| caso_sw cases
|
| error CASO
;

3. ***Lenguaje Json***

Analisi de la gramática de analizador Léxico

—Palabras Reservadas:

SCORE="score, CLASES="clases", VARIBALES="variables"
METODOS="metodos", COMENTARIOS="comentarios"
FUNCION="funcion", TIPO="tipo", NOMBRE="nombre", PARAMTETROS="parametros"
TEXTO="texto"

–Identificadores:

L*os identificadores fueros reconocido válidos como las siguientes expresiones:*
IDD = ({LETRA}|{DIAGONAL})({LETRA}|{ENTERO}|)({LETRA}|{ENTERO}))|{LETRA})*

—OperadoreAlgebraicos

signos conocidos como operadores de forma — (+|-||/)*

—Signos de Puntuación

signos conocidos como puntuación de forma — (;|,|:|.)

—Signo de agrupación

signos conocidos como agrupación de forma — ([|]|{|})

Gramática para el analizador sintáctico:

El analizador sintáctico, conocido como parser, o en el lenguaje Java la Herramienta de Parseo(CUP), se encarga de plantear la validez sintáctica por medio de revisión de gramáticas que representan un una secuencia de tokens enviadas por el analizador léxico, revisando el orden de las mismas y que la expresión sea reconocida de acuerdo con las reglas gramaticales que se estén implementando.

Gramática (G)
G(N,T,P,S)

- *N = No terminales*
- *T = Terminales*
- *P = Reglas de Producción*

❖ *S = Símbolo Inicial*

N —- Símbolos no Terminales:

- *inicio, def_clase, clases, clase, def_score, def_variable, variable, variables, nom_Var, tipo_Var, funtions, def_coment, comodinComent,metodo;*
- *funcion_padre, componentes, fun_oblig, fun_Items, def_metodo, metods, parametros, nom_Metod, tipo_Metod, idd_Coment, text, coments, comodinItem;*

T —- Símbolos Terminales:

- *terminal Token DOSPUNTO, COMA, ENTERO, DECIMAL, COMILLAS, CORCHETEA, CORCHETEC, LLAVEA, LLAVEC;*
- *terminal Token SCORE, CLASES, VARIBALES, METODOS, COMENTARIOS, FUNCION, TIPO, NOMBRE, PARAMTETROS, TEXTO,IDD, ALGO ;*

P — Reglas de Producción:

inicio ::= LLAVEA componentes LLAVEC
;

componentes ::= def_score def_clase def_variable def_metodo def_coment

def_clase ::= CLASES DOSPUNTO CORCHETEA clases CORCHETEC COMA
| CLASES DOSPUNTO CORCHETEA CORCHETEC COMA
| error def_variable
;

clases ::= LLAVEA clase LLAVEC COMA clases
| LLAVEA clase LLAVEC
| error LLAVEC
;

clase ::= NOMBRE DOSPUNTO COMILLAS IDD:nomClase COMILLAS
| error LLAVEC
;

def_variable ::= VARIBALES DOSPUNTO CORCHETEA variables CORCHETEC COMA
| VARIBALES DOSPUNTO CORCHETEA CORCHETEC COMA
| error def_metodo
;

variables ::= LLAVEA variable LLAVEC COMA variables
| LLAVEA variable LLAVEC
| error LLAVEC

;

variable ::= nom_Var COMA tipo_Var COMA funtions
| error LLAVEC
;

nom_Var ::= NOMBRE DOSPUNTO COMILLAS IDD:nomVar COMILLAS
| error COMA
;

tipo_Var ::= TIPO DOSPUNTO COMILLAS IDD:tipoVAr COMILLAS
| error COMA
;

funtions ::= FUNCION DOSPUNTO COMILLAS funcion_padre COMILLAS
| error LLAVEC
;

funcion_padre ::= fun_oblig fun_Items
;

def_metodo ::= METODOS DOSPUNTO CORCHETEA metods CORCHETEC COMA
| METODOS DOSPUNTO CORCHETEA CORCHETEC COMA
| error def_coment
;

metods ::= LLAVEA metodo LLAVEC COMA metods
| LLAVEA metodo LLAVEC
| error LLAVEC
;

4. ***Lenguaje DEF***

Analisi de la gramática de analizador Léxico

—***Palabras Reservadas:***

HTMABRE="<html>", HTMCIERRE="</html>", H1ABRE="<h1>", H1CIERRE="</h1>"
H2ABRE="<h2>", H2CIERRE="</h2>", TABLAABRE="<table>",
TABLACIERRE="</table>", TRABRE="<tr>", TRCIERRE="</tr>", THABRE="<th>"
*THCIERRE="</th>", TDABRE="<td>", TDCIERRE="</td>", SALTO="
"*
FORABRE="<for", FORCIERRE="</for>", PUNTO = ".", DOSPUNTO = ":"
HASTA="hasta", ITERADOR = "iterador" INTEGER="integer"
STRING="string"

–***Identificadores:***

L*os identificadores fueros reconocido válidos como las siguientes expresiones:*

IDD = ({LETRA}|{DIAGONAL})({LETRA}|{ENTERO}|)({LETRA}|{ENTERO}))|{LETRA})*

—OperadoreAlgebraicos
signos conocidos como operadores de forma — (+|-||/)*

—Signos de Puntuación
signos conocidos como puntuación de forma — (;|,|:|.)

—Signo de agrupación
signos conocidos como agrupación de forma — ([|]|{|})

Gramática para el analizador sintáctico:
El analizador sintáctico, conocido como parser, o en el lenguaje Java la Herramienta de Parseo(CUP), se encarga de plantear la validez sintáctica por medio de revisión de gramáticas que representan un una secuencia de tokens enviadas por el analizador léxico, revisando el orden de las mismas y que la expresión sea reconocida de acuerdo con las reglas gramaticales que se estén implementando.

Gramática (G)
G(N,T,P,S)
- ❖ *N = No terminales*
- ❖ *T = Terminales*
- ❖ *P = Reglas de Producción*

—- Símbolos no Terminales:
- *def_variable,variable_integer,variable_String,strin,items_coma,items,asignacion,iddVarComodin, titleRescursiver;*
- *inte,items_int,items_coma_B,asignacionInt,inicio,var_Asign,iddItemsComodin,iddComodinInt,operationaux;*
- *String operationString, concatenacion,subTerminalTabla,def_var_Globla, terminalTabla,operationAsig;*
- *Integer index, parametros,operation;*
- *title1, itemHtml, comodinIddHtml, usoVarGlobal, title2,columTitle,comodinH1H2, columDatos, fila_columna, validosTR, forAnidado;*
- *comodintitle, def_table, comodinTable, def_html_General, sentencias, sentencia,def_for, sentenciasFro,sentenciaFor,fila_columnaRecur;*

T —- Símbolos Terminales:
- *HTMABRE,ENTERO,DECIMAL,HTMCIERRE,H1ABRE,H1CIERRE,H2ABRE,H2CIERRE,TABLAABRE,TABLACIERRE,TRABRE,TRCIERRE,THABRE;*
- *THCIERRE,TDABRE,TDCIERRE,SALTO,FORABRE,FORCIERRE,PUNTO,DOSPUNTO,HASTA,MAYOR,MENOR,POSABERTURA,POSCERRADURA,COMA;*

- *PUNTOCOMA, IDD,COMODIN,PARENTESISA,PARENTESISC,IGUAL,MENOS,MAS,POR,DIVISION,CORCHETEA,CORCHETEC,INTEGER,STRING;*
- *SCORE, CADENA, RESULT, VARIABLES, CLASES, NOMBRE,TIPO,FUNCION,METODOS,COMENTARIOS,TEXTO,PARAMTETROS, ITERADOR;*

P — Reglas de Producción:

inicio ::= def_variable def_html_General
;

def_variable ::= variable_integer def_variable
| variable_String def_variable
| var_Asign def_variable
|

variable_String ::= strin items
;

strin ::= STRING IDD:identificador
;

items_coma ::= iddItemsComodin items
| error PUNTOCOMA
;

iddItemsComodin ::= IDD:nomVar
;

items ::= asignacion PUNTOCOMA
| asignacion COMA items_coma
| COMA items_coma
| PUNTOCOMA
;

inte ::= INTEGER IDD:identificador
;

items_int ::= asignacionInt PUNTOCOMA
| asignacionInt COMA items_coma_B
| COMA items_coma_B{::}
;

```
 items_coma_B ::= iddComodinInt items_int

        ;

fila_columna ::= TRABRE validosTR  TRCIERRE    {:actionSalid.capturarTR();:}
        | TRABRE   TRCIERRE
        | error TABLACIERRE
        ;

validosTR ::= columDatos validosTR
        | columTitle validosTR
        | columDatos
        | columTitle
        | comodintitle validosTR
        | comodintitle
        ;

 fila_columnaRecur ::= fila_columna fila_columnaRecur
                | fila_columna
                | error FORCIERRE
                ;

forAnidado ::= FORABRE ITERADOR DOSPUNTO IDD HASTA DOSPUNTO IDD
PUNTOCOMA MAYOR fila_columnaRecur FORCIERRE
        | FORABRE ITERADOR DOSPUNTO IDD HASTA DOSPUNTO IDD PUNTOCOMA
MAYOR  FORCIERRE
        ;
```

Diagramas de Clases: Imágenes en la carpeta :)