

*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***

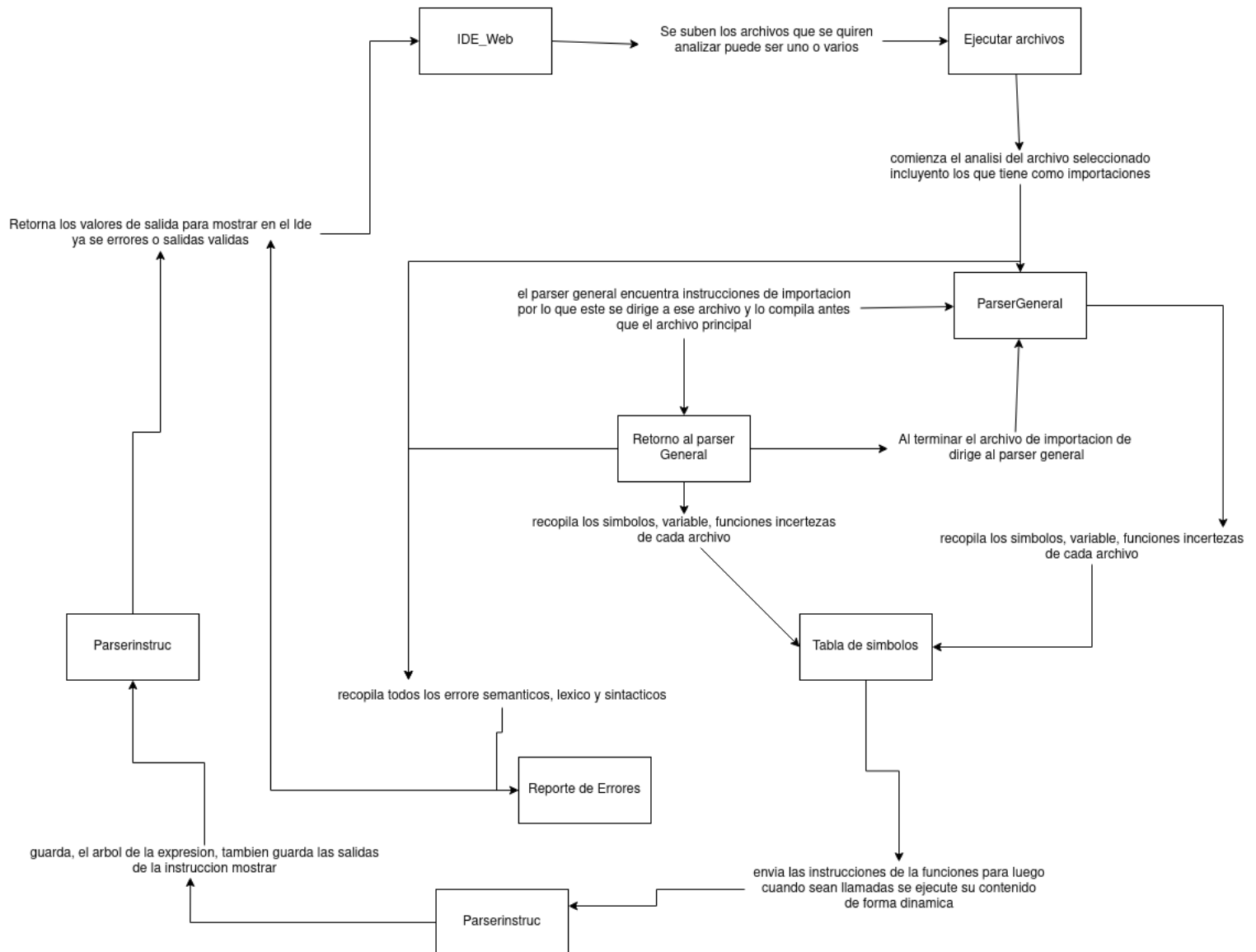
*Elvis Lizandro Aguilar Tax*

***Carnet***

*201930304*

***Manual Tecnico***

## 1. Organización del proyecto



## 2. definición del lenguaje

### Analisi de la gramática de analizador Léxico

#### —Palabras Reservadas:

*Incerteza, Importar, crt, true, false, Double, Boolean, Int, String, Char, Void, Retorno, Principal, Sino, Si, Para, Mientras, Detener, Continuar, Mostrar, DibujarAST, DibujarEXP, DibujarTS*

#### —Identificadores:

*Los identificadores fueron 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 Logicos**

signos conocidos como agrupación de forma (|&,||,&&)

### **—Signo de Relacionales**

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, comodinImport, def\_metodo, def\_variable, variables, items\_coma, items, fin\_asignar, asignacion, saltos, identador, identadorRecu, inserteza
- tipo, visibilidad, sentencias, sentencia, sentencias\_global, sentencia\_global, def\_Si, def\_sino, def\_para
- parametros, llamada\_metod, parametros\_llamada, sentenciasGlobales
- def\_Mientras, def\_Mostrar, parametroMostrar, Graficando, parametros, terminalrOp operation

### **$T$ — Símbolos Terminales:**

- ENTERO, DECIMAL, PARENTESISIA, PARENTESISIC, PUNTO, COMA, PUNTOCOMA;
- DOPUNTO, MENOS, MAS, POR, DIVISION, IGUAL, EQUALS, OR, AND, MENORQ, MAYORQ, MAYOROI, MENOROI, CASO;
- NOTEQUALS, IMPORT, IDD, INT, BOOLEAN, TRUE, CADENA;

- *STRING, NEGATION, CHAR, CHARACTER, DOUBLE, SI, SINO, PARA, SWITCH, RETORNO, FALSE, SALTOS, TABULADORES*

***P — Reglas de Producción:***

*inicio:*

*saltos importacion insertez sentenciasGlobales*  
*| importacion insertez sentenciasGlobales*  
*;*

*importacion :*

*comodinImpor importacion {}*  
*|*  
*;*

*comodinImpor:*

*IMPORT IDD PUNTO EXTENSIONCLR saltos*  
*;*

*saltos :*

*SALTO saltos*  
*| SALTO*  
*| EOF*  
*;*

*insertez :*

*INSERTEZA DECIMAL saltos*  
*| INSERTEZA ENTERO saltos*  
*|*  
*;*

*variables:*

*tipo items\_coma*  
*;*

*items\_coma:*

*comodinItems items*  
*| error saltos*  
*;*

*comodinItems:*

*IDD*  
*;*

*items :*

*IGUAL asignacion*  
*| COMA items\_coma*  
*| saltos*  
*| error saltos*  
*;*

*tipo:*

*DOUBLE*  
*| INT*  
*| BOOLEAN*  
*| CHAR*  
*| STRING*  
*;*

*asignVar:*

*IDD IGUAL*  
*;*

*asignacion :*

*operation saltos*  
*| error saltos*  
*;*

*funtions :*

*comodinIDFun saltos sentenciasFuncion*  
*;*

*comodinIDFun:*

*comss PARENTESIS C DOPUNTO*  
*| comss parametros PARENTESIS C DOPUNTO*  
*;*

*comss:*

*tipo IDD PARENTESIS A*  
*;*

*retornoFuntion :*

*RETURN operation saltos*  
*| RETURN saltos*  
*| error saltos*  
*;*

*parametros:*

*param COMA parametros*  
*| param*

| *error saltos*  
;

*param:*  
    *tipo IDD*  
;

*sentenciasFuncion:*  
    *sentenciaFn sentenciasFuncion*  
    | *sentenciaFn*  
    | *error saltos*  
;

*sentenciaFn:*  
    *identadorRecu variables*  
    | *identadorRecu defSi*  
    | *identadorRecu defMientras*  
    | *identadorRecu defPara*  
    | *identadorRecu defMostrar*  
    | *identadorRecu defSino*  
    | *identadorRecu retornoFuntion*  
    | *identadorRecu llamadaFun*  
    | *identadorRecu graficando*  
    | *identadorRecu escapes*  
    | *identadorRecu graficandoATS*  
    | *identadorRecu graficandoTS*  
;

*identadorRecu :*  
    *comodinIdentado identadorRecu*  
    | *comodinIdentado*  
;

*comodinIdentado:*  
    *IDENTADOR*  
;

*metod :*  
    *comodinVerificador DOPUNTO saltos sentenciasMetod*  
;

*comodinVerificador:*  
    *comodinMetod PARENTESISISC*

| *comodinMetod parametros PARENTESISC*  
;

*comodinMetod:*  
    *VOID IDD PARENTESISA*  
;

*sentenciasMetod :*  
    *sentenciaFn sentenciasMetod*  
    | *sentenciaFn*  
    | *error saltos*  
;

*funPrincipal :*  
    *comodinFunPrim saltos sentenciasFuncion*  
;

*comodinFunPrim:*  
    *VOID PRINCIPAL PARENTESISA PARENTESISC DOPUNTO*  
;

*llamadaFunOP:*  
    *IDD PARENTESISA PARENTESISC*  
    | *IDD PARENTESISA parametrosLlamada PARENTESISC*  
;

*parametrosLlamada :*  
    *comoidnOP COMA parametrosLlamada*  
    | *comoidnOP*  
    | *error saltos*  
;

*comoidnOP:*  
    *operation*  
;

*sentenciasGlobales :*  
    *sentenciaGlobales sentenciasGlobales*  
    |  
;

*sentenciaGlobales :*  
    *funtions*

```

| asignVar
| variables
| metod          {}
| funPrincipal
;

```

```

defSi :
    SI PARENTESIS operation PARENTESISC DOPUNTO saltos
;

```

```

defSino :
    SINO DOPUNTO saltos          {}
;

```

```

defPara:
    PARA PARENTESIS comodinVar PUNTOCOMA operation PUNTOCOMA
incremDecrem PARENTESISC DOPUNTO saltos
;

```

```

comodinVar:
    INT IDD IGUAL operation
;

```

```

incremDecrem :
    MASMAS
    | MENOSMENOS
;

```

```

defMientras :
    MIENTRAS PARENTESIS operation PARENTESISC DOPUNTO saltos
;

```

```

defMostrar :
    MOSTRAR PARENTESIS parametroMostrar PARENTESISC saltos
;

```

```

parametroMostrar :
    stringOidd COMA parametroMostrar
    | stringOidd
    | error saltos
;

```

```

stringOidd :

```



*operation*  
;

*graficando :*

*DIBUJAREXP PARENTESISA operation PARENTESISC saltos*  
;

*escapes:*

*DETENER saltos*  
| *CONTINUAR saltos*  
;

*graficandoATS:*

*DIBUJARAST PARENTESISA IDD PARENTESISC saltos*  
;

*graficandoTS:*

*DibujarTS PARENTESISA PARENTESISC saltos*  
;

*terminalsOP :*

*TRUE*  
| *FALSE*  
| *DECIMAL*  
| *ENTERO*  
| *CARACTER*  
| *CADENA*  
| *IDD*  
| *llamadaFunOP*  
;

*operation :*

*operation MAS operation*                      *{ \$\$ = Parser.yy.opCast.suma(\$1,\$3); }*  
| *operation MENOS operation*                      *{ \$\$ =*  
*Parser.yy.opCast.resta(\$1,\$3); }*  
| *operation POR operation*                      *{ \$\$ = Parser.yy.opCast.multiplicacion(\$1,\$3); }*  
  
| *operation DIVISION operation*  
| *operation ELEVADO operation*  
| *operation MOD operation*  
| *MENOS operation %prec UMINUS*  
| *NEGADO operation %prec UMINUS*  
| *operation EQUALS operation*  
| *operation NOEQUALS operation*  
| *operation MENORQ operation*

| *operation MAYORQ operation*  
| *operation MENOROI operation*  
| *operation MAYOROI operation*  
| *operation SIGINSERTEZA operation*  
| *operation OR operation*  
| *operation AND operation*  
| *operation XOR operation*  
| *PARENTESISIA operation PARENTESISIC*  
| *terminalsOP*

;