**Documentación**

**Parte Léxica**

**Expresiones Regulares**

digito = [0-9]

letra = [a-zñ]

cadena = [\"][^\"\n]+[\"]

double = [+-]? {digito}\* [\.]? {digito}+

id = {letra}({letra}|{ digito }|\_)\*

LineTerminator = \r|\n|\r\n

InputCharacter = [^\r\n]

comentariosimple = "!" {InputCharacter}\* {LineTerminator}?

comentarioMulti = "<!" [^/] ~"!>"

**Palabras Reservadas**

Program

End Program

Var

End

Double

Char

Arr

SUM

RES

MUL

DIV

MOD

Media

Mediana

Moda

Varianza

Max

Min

Console

Print

Column

EXEC

graphBar

graphPie

graphLine

Histogram

values

label

Titulo

ejeX

ejeY

tituloX

tituloY

**Simbolos**

“:”

“<”

“>”

“!”

“-”

“;”

“[”

“]”

“@”

“.”

“,”

“””

“(”

“)”

**Parte Sintáctica**

**Terminales**

DosPuntos, menor, mayor, guion, PuntoComa, AbreC, CierraC, arroba, Coma, AbreP, CierraP, Igual, Comillas

Program, endProgram, var, end, doubleee, charr, arr

SUM, RES, MUL, DIV, MOD, Media, Mediana, Moda, varianza, max, min, Console, print

column, exec, graphBar, graphPie, graphLine, Histogram, values, label, titulo, ejeX, ejeY, tituloX, tituloY

cadena, doublee, id

**No terminales**

String INICIO

Imprimir, Operaciones

DeclaracionDou, DeclaracionCha, DeclaracionAr, TipoAr, TipoArD, TipoArC

ImpresionP, TipoImpresionP

GraphPie

TodoTipo

Empieza con en no terminal, INICIO

**INICIO** -> Program TodoTipo endProgram

**TodoTipo** -> DeclaracionDou TodoTipo

| DeclaracionCha TodoTipo

| DeclaracionAr TodoTipo

| ImpresionP

| GraphPie

**DeclaracionDou** -> var DosPuntos doubleee DosPuntos DosPuntos id menor guion doublee end PuntoComa

**DeclaracionCha** -> var DosPuntos charr AbreC CierraC DosPuntos DosPuntos id menor guion cadena end PuntoComa

**DeclaracionAr** -> arr DosPuntos TipoAr end PuntoComa

**TipoAr** -> doubleee DosPuntos DosPuntos arroba id menor guion AbreC TipoArD CierraC

| charr DosPuntos DosPuntos arroba id menor guion AbreC TipoArC CierraC

**TipoArD** -> doublee Coma TipoArD

| doublee

**TipoArC** -> cadena Coma TipoArC

| cadena

**ImpresionP** -> Console DosPuntos DosPuntos print Igual TipoImpresionP end PuntoComa

**TipoImpresionP** -> doublee Coma TipoImpresionP

| cadena Coma TipoImpresionP

| id Coma TipoImpresionP

| doublee

| cadena

| id