

GRAMATICA 1

Expresiones Regulares Utilizadas

```

blancos=[\r\t]+
cadena=[\"][^\n"]*\["
decimal=[0-9]+("[0-9]+)?
variable=[a-zA-Z]([0-9]|[a-zA-Z]|"_")*
comentarioo=("###")(.*\n)
comentarioomu("#" "*" ["^" "*" ]*" "*" "#")

```

Terminales	No Terminales
-generarreporteestadistico	-INICIO
-definirglobales	-FUERA
-compare	-CONTENIDOP
-graficabarras	-VAODE
-graficapie	-VAOCA
-graficalineas	-ESTADISTICO
-puntajepecifico	-LINEAS
-sstring	-CONTENIDOL
-ddouble	-PIE
-titulo	-CONTENIDOB
-valores	-CUERPO2
-ejex	-BARRAS
-cadena	-INSTRUCCIONES
-decimal	-GLOBALES
-variable	-CUERPO
-titulox	-COMPARAR
-tituloy	-COMENTARIO
-archivo	
-comentarioo	
-comentariomu	
-Puntajegeneral	

El inicio de la gramatica es: INICIO

INICIO -> ESTADISTICO;

ESTADISTICO -> FUERA "{" GLOBALES CUERPO;

FUERA ->COMENTARIO FUERA

|COMENTARIO

|generarreporteestadistico;

CUERPO -> COMPARAR CUERPO2

|BARRAS CUERPO

|PIE CUERPO

|COMENTARIO CUERPO

|LINEAS CUERPO;

CUERPO2 -> BARRAS CUERPO2

|PIE CUERPO2

|LINEAS CUERPO2

|COMENTARIO CUERPO2

|"}";

GLOBALES -> definirglobales "{" INSTRUCCIONES;

INSTRUCCIONES->sstring variable "=" cadena ":" INSTRUCCIONES

|ddouble variable "=" decimal2 ":" INSTRUCCIONES

|ddouble variable "=" "\$" "{" puntajepecifico "," cadena "," cadena "," cadena
"}" ":" INSTRUCCIONES

|ddouble variable "=" "\$" "{" Puntajegeneral "}" ":" INSTRUCCIONES

|COMENTARIO INSTRUCCIONES

|"}";

COMPARAR-> compare "(" cadena "," cadena ")"

LINEAS-> graficalineas "{" CONTENIDOL;

CONTENIDOL->titulo ":" VAOCA ":" CONTENIDOL

```
|archivo ":" VAOCA "," CONTENIDOL
|COMENTARIO CONTENIDOL
|"}";
```

```
PIE-> graficapie "{" CONTENIDOP;
CONTENIDOP->titulo ":" VAOCA "," CONTENIDOP
|ejex ":" "(" REPE ")" "," CONTENIDOP
|valores ":" "(" REPE2 ")" "," CONTENIDOP
|COMENTARIO CONTENIDOP
|"}";
```

```
BARRAS-> graficabarras "{" CONTENIDOB;
CONTENIDOB-> titulo ":" VAOCA "," CONTENIDOB
|ejex ":" "(" REPE ")" "," CONTENIDOB
|valores ":" "(" REPE2 ")" "," CONTENIDOB
|titulox ":" VAOCA "," CONTENIDOB
|tituloy ":" VAOCA "," CONTENIDOB
|COMENTARIO CONTENIDOB
|"}";
```

```
REPE-> VAOCA "," REPE
|VAOCA
REPE2->VAODE "," REPE2
|VAODE
VAOCA-> variable:
|cadena
VAODE->decimal
|variable
|PUNTAJE
```

```
PUNTAJE-> "$" "{" puntajepecifico "," cadena "," cadena "," cadena "}"
| "$" "{" Puntajegeneral "}";
```

COMENTARIO->comentarioo

|comentarioomu;

GRAMATICA 2

Expresiones Regulares Utilizadas

```

blancos=[ \r\t]+
cadena=[""][\^\n]*[""]
caracter=[""][\^\n]?[""]
decimal=[0-9]+("."[0-9]+)?
id=[a-zA-ZÀ-ÿ\u00f1\u00d1]([0-9]|[a-zA-ZÀ-ÿ\u00f1\u00d1]|"_")*
comentarioo=("//")(.)*(\n)
comentarioomu=("/"*[^\"]*"/*"/)
"=" acion=("="=")
diferencia=("!""=")
ma "=" =(">""=")
me "=" =("<""=")
andd=("&""&")
orr=("|""|")

```

Terminales	No Terminales
- class	-INICIO
- var	-CLASE
- let	-FUNCIONES
- const	-FUNCION
- true	-VARIABLES
- false	-VARIABLE
- if	-LINEAS
- else	-LINEAS2
- for	-PARAMETROS
- while	-LINEA
- do	-TIPO
- switch	-IF
- case	-ELSE
- break	-ELIF
- default	-IFS
- console	-ELIFS
- log	-FOR
- require	-WHILE
- cadena	-DO
- caracter	-SWITCH
- decimal	-EXP
- comentarioo	-INICIALIZADOR
-id	-CASE
- comentarioomu	-CASES
- "=" acion	-DEFAULT

- diferencia - ma “=” - me “=” - orr - andd	-CONSOLA -COMENTARIOS -LLAMADA -ENVIAR -PARAMETROS2 -IMPORTAR -LINFUERA -FUERA -FUERA2 -FUERA3
---	---

El inicio de la gramatica es: INICIO

INICIO -> CLASE;

CLASE -> FUERA id “{” FUNCIONES FUERA2

FUERA -> FUERA COMENTARIOS

|COMENTARIOS class

| class ;

FUERA2 -> “}” FUERA3

| “}” ;

FUERA3 -> FUERA3 COMENTARIOS

|COMENTARIOS;

LINFUERA -> TIPO id “=” EXP “,”

|id “=” EXP “,”

|TIPO id “=” EXP

|id “=” EXP

|TIPO id “,”

|TIPO id

|COMENTARIOS;

FUNCIONES -> FUNCIONES FUNCION

|FUNCION

|FUNCIONES LINFUERA

|LINFUERA;

FUNCION -> idi “(“ PARAMETROS “{“ LINEAS “}”

PARAMETROS -> VARIABLES “)”

| “)” ;

VARIABLES -> VARIABLES “,” id

|id

LINEAS2 -> LINEAS2 LINEA

|LINEA

|LINEAS2 break

|LINEAS2 break “,”

| break “,”

| break;

LINEAS -> LINEAS LINEA

|LINEA;

LINEA -> VARIABLE

|IFS

|FOR

|WHILE

|DO

|SWITCH

|CONSOLA

|COMENTARIOS

|LLAMADA

|IMPORTAR;

VARIABLE -> TIPO id “=” EXP “,”

|id “=” EXP “,”

|TIPO id “=” EXP

|id “=” EXP

|TIPO id “,”

|TIPO id

|TIPO id “*” “=” EXP

|TIPO id “/” “=” EXP

|TIPO id “+” “=” EXP

|TIPO id “-” “=” EXP

|TIPO id “%” “=” EXP

|id “*” “=” EXP

```

|id "/" "=" EXP
|id "-" "=" EXP
|id "%" "=" EXP
|TIPO id "*" "=" EXP ","
|TIPO id "/" "=" EXP ","
|TIPO id "+" "=" EXP ","
|TIPO id "-" "=" EXP ","
|TIPO id "%" "=" EXP ","
|id "*" "=" EXP ","
|id "/" "=" EXP ","
|id "+" "=" EXP ","
|id "-" "=" EXP ","
|id "%" "=" EXP ","
|id "+" "+"
|id "+" "+" ","
|id "-" "-"
|id "-" "-" "," ;

```

TIPO -> var

```

|let
|const;

```

IMPORTAR -> TIPO id "=" require "(" cadena ")" ";"

```

|TIPO id "=" require "(" cadena ")"
|id "=" require "(" cadena ")"
|id "=" require "(" cadena ")" ";" ;

```

LLAMADA -> id "(" PARAMETROS2 ","

```

|id "(" PARAMETROS2;

```

PARAMETROS2 -> ENVIAR ")"

```

| ")" ;

```

ENVIAR -> ENVIAR "," EXP

```

|EXP;

```


COMENTARIOS -> comentario

|comentarioomu

CONSOLA -> console punto log "(" EXP ")" ","

|console punto log "(" EXP ")" ;

SWITCH -> switchh "(" id ")" "{" CASES DEFAULT;

CASES -> CASES CASE

|CASE;

CASE -> case EXP "," LINEAS break ","

| case EXP "," LINEAS break;

DEFAULT -> default "," LINEAS2 "}"

| "}" ;

DO -> do "{" LINEAS2 "}" while("(" EXP ")" ","

|do "{" LINEAS2 "}" while("(" EXP ")" ;

WHILE -> while("(" EXP ")" "{" LINEAS2 "}" ;

FOR -> for "(" INICIALIZADOR "," EXP "," LINEA ")" "{" LINEAS2 "}" ;

INICIALIZADOR -> TIPO id "=" EXP

|id "=" EXP;

ELIFS -> ELIFS ELIF

|ELIF;

IFS -> IF

|IF ELSE

|IF ELIFS ELSE

|IF ELIFS

| if "(" EXP ")" LINEA;

IF -> if "(" EXP ")" "{" LINEAS2 "}" ;

ELSE -> else "{" LINEAS2 "}" ;

ELIF -> else if "(" EXP ")" "{" LINEAS2 "}" ;

EXP -> EXP suma EXP

|EXP "-" EXP

|EXP "*" EXP

|EXP "/" EXP

|EXP "*" "*" EXP

|EXP "%" EXP

|EXP "*" "=" EXP

|EXP "+" "=" EXP

|EXP "<" EXP

|EXP "<" "=" EXP

|EXP ">" EXP

|EXP ">" "=" EXP

|EXP "!="EXP

|EXP "="EXP

|EXP "|" EXP

|EXP "& &" EXP

| "(" EXP ")"

| "-" EXP

| "!" EXP

|decimal

|cadena

|id

|character

| true

| false;