

## Expresiones Regulares

### Espacios en Blanco

`[ \r\t]+`

`\n`

`(\\V).*`

`[/][*][^*]*[*]+([^/*][^*]*[*]+)*[/]`

### Simbolos

`","` PCOMA

`"("` PAIZQ

`")"` PADER

`"."` PUNTO

`":"` DOSPUNTOS

`","` COM

`"["` CORIZQ

`"]"` CORDER

`"{"` LLAVEIZQ

`"}"` LLAVEDER

`"=="` IGUALIGUAL

`"<="` MENORIGUAL

`">="` MAYORIGUAL

`"++"` INCREMENTO

`"--"` DECREMENTO

`"="` IGUAL

### //palabras reservadas

`"print"` RPRINT

`"true"` RTRUE

`"false"` RFALSE

"if"	RIF
"else"	RELSE
"while"	RWHILE
"main"	RMAIN
"for"	RFOR
"do"	RDO
"toLowerCase"	RTOLOWER
"toUpperCase"	RTOUPPER
"round"	RROUND
"truncate"	RTRUNCATE
"length"	RLENGTH
"typeof"	RTYPEOF
"toString"	RTOSTRING
"return"	RRETUR
"continue"	RCONTINUE
"break"	RBREAK
"switch"	RSWITCH
"case"	RCASE
"default"	RDEFAULT

#### Tipos de Variable

"int"	RINT
"double"	RDOUBLE
"char"	RCHAR
"string"	RSTRING
"boolean"	RBOOLEAN

#### Tipo para Metodo

"void"	return "RVOID"
--------	----------------

## Operadores Aritmeticos

```
"+"      return "MAS"
"- "     return "MENOS"
"*"      return "POR"
"/"      return "DIVISION"
"^"      return "POTENCIA"
%"       return "MODULO"
```

## Operadores Relacionales

```
"!="      return "DIFERENTE"
"<"      return "MENORQUE"
">"      return "MAYORQUE"
```

## Operadores Lógicos

"&&"	"AND"
"  "	"OR"
"!"	"NOT"

```
[a-zA-Z][a-zA-Z0-9_]* "ID"
[0-9]+(".[0-9]+)\b "DECIMAL"
[0-9]+\b "ENTERO"
\'((\\\'|\\\"|\\n|\\r|\\t|\\f|\\b|\\z|\\s|\\p{Cn})*)\' 'CHARACTER'
["]
<string>[^"\\]+
<string>"\\\\"
```

```

<string>"\\n"
<string>"\\t"
<string>"\\\\\\"
<string>"\\\\"
<string>["]      'CADENA';}

```

**Conjunto de Terminales** = { RPRINT, PAIZQ, PADER, PCOMA, PUNTO, DOSPUNTOS, COMA, CORIZQ, CORDER, LLAVEIZQ, LLAVEDER, IGUALIGUAL, MENORIGUAL, MAYORIGUAL, INCREMENTO, DECREMENTO, IGUAL, RTRUE, RFALSE, RIF, RELSE, RWHILE, RMAIN, RFOR, RDO, RTOLOWER, RTOUPPER, RROUND, RTRUNCATE, RLENGTH, RTYPEOF, RTOSTRING, RRETURN, RCONTINUE, RBREAK, RSWITCH, RCASE, RDEFAULT, RINT, RDOUBLE, RCHAR, RSTRING, RBOOLEAN, RVOID, MAS, MENOS, POR, DIVISION, POTENCIA, MODULO, DIFERENTE, MENORQUE, MAYORQUE, AND, OR, NOT, ID, DECIMAL, ENTERO, CARACTER, CADENA }

**Conjunto de No Terminales** = { INICIO, INSTRUCCIONES, INSTRUCCIÓN, FUNCPRINT, INSDECLARAR, INSASIGNAR, LLAMARFUNCION, GUARDARFUNCION, INSTIF, INSWHILE, INSDOWHILE, INSFOR, INSSWITCH, INCREMENTODECREMENTO, RETORNO, EXPRESION, ASIGNARFOR, CASTEO, CASOS, CASEST, INSSTATEMENT, PARAMETROS, PARAMETRO, ARITMETICA, ARGUMENTOS, RELACIONAL, LOGICO, ACCEDERVARIABLE, TOLOWERUPPER, REDONDEO, TRUNCAR, ELENGTH, ETYPOF, ETOSTRING, PRIMITIVO, TIPO }

Inicio de la Gramática → start with INICIO

INICIO

: INSTRUCCIONES EOF

INSTRUCCIONES

: INSTRUCCIONES INSTRUCCION

| INSTRUCCION

## INSTRUCCION

: FUNCPRINT

| INSDECLARAR

| INSASIGNAR

| LLAMARFUNCION PCOMA

| GUARDARFUNCION

| INSTIF

| INSWHILE

| INSDOWHILE

| INSFOR

| INSSWITCH

| INCREMENTODECREMENTO PCOMA

| RETORNO PCOMA

| RCONTINUE PCOMA

| RBREAK PCOMA

| RMAIN LLAMARFUNCION PCOMA

| error

## FUNCPRINT

: RPRINT PAIZQ EXPRESION PADER PCOMA

## INSDECLARAR

: TIPO ID PCOMA

| TIPO ID IGUAL EXPRESION PCOMA

| TIPO ID IGUAL CASTEO PCOMA

## INSASIGNAR

: ID IGUAL EXPRESION PCOMA

| ID IGUAL CASTEO PCOMA

ASIGNARFOR

: ID IGUAL EXPRESION

CASTEO

: PAIZQ TIPO PADER EXPRESION

GUARDARFUNCION

: TIPO ID PAIZQ PADER INSSTATEMENT

| TIPO ID PAIZQ PARAMETROS PADER INSSTATEMENT

INSTIF

:RIF PAIZQ EXPRESION PADER INSSTATEMENT INSTELSE

INSTELSE

:RELSE INSSTATEMENT

|RELSE INSTIF

|

INSWHILE

: RWHILE PAIZQ EXPRESION PADER INSSTATEMENT

INSDOWHILE

: RDO INSSTATEMENT RWHILE PAIZQ EXPRESION PADER PCOMA

INSFOR

: RFOR PAIZQ INSDECLARAR EXPRESION PCOMA EXPRESION PADER INSSTATEMENT

| RFOR PAIZQ INSASIGNAR EXPRESION PCOMA EXPRESION PADER INSSTATEMENT  
| RFOR PAIZQ INSDECLARAR EXPRESION PCOMA ASIGNARFOR PADER INSSTATEMENT  
| RFOR PAIZQ INSASIGNAR EXPRESION PCOMA ASIGNARFOR PADER INSSTATEMENT

## INSSWITCH

: RSWITCH PAIZQ EXPRESION PADER LLAVEIZQ CASOS LLAVEDER

## CASOS

:RCASE EXPRESION DOSPUNTOS CASEST  
|CASOS RCASE EXPRESION DOSPUNTOS CASEST  
|RDEFAULT DOSPUNTOS CASEST  
|CASOS RDEFAULT DOSPUNTOS CASEST

## CASEST

:INSTRUCCIONES  
|

## RETORNO

: RRETURN  
| RRETURN EXPRESION

## INSSTATEMENT

: LLAVEIZQ INSTRUCCIONES LLAVEDER

## PARAMETROS

: PARAMETROS COMA PARAMETRO  
| PARAMETRO

## PARAMETRO

: TIPO ID

## EXPRESION

: PRIMITIVO

| RELACIONAL

| LOGICO

| ACCEDERVARIABLE

| INCREMENTODECREMENTO

| ARITMETICA

| LLAMARFUNCION

| TOLOWERUPPER

| REDONDEO

| TRUNCAR

| ELENGTH

| ETYPEOF

| ETOSTRING

| PAIZQ EXPRESION PADER

## LLAMARFUNCION

: ID PAIZQ PADER

| ID PAIZQ ARGUMENTOS PADER

## ARGUMENTOS

: ARGUMENTOS COMA EXPRESION

| EXPRESION



## ARITMETICA

- : EXPRESION MAS EXPRESION
- | EXPRESION MENOS EXPRESION
- | EXPRESION POR EXPRESION
- | EXPRESION DIVISION EXPRESION
- | EXPRESION POTENCIA EXPRESION
- | EXPRESION MODULO EXPRESION
- | MENOS EXPRESION

## RELACIONAL

- : EXPRESION IGUALIGUAL EXPRESION
- | EXPRESION DIFERENTE EXPRESION
- | EXPRESION MENORQUE EXPRESION
- | EXPRESION MENORIGUAL EXPRESION
- | EXPRESION MAYORQUE EXPRESION
- | EXPRESION MAYORIGUAL EXPRESION

## LOGICO

- : EXPRESION AND EXPRESION
- | EXPRESION OR EXPRESION
- | NOT EXPRESION

## ACCEDERVARIABLE

- :ID

INCREMENTODECREMENTO

: ID INCREMENTO

| ID DECREMENTO

TOLOWERUPPER

: RTOLOWER PAIZQ EXPRESION PADER

| RTOUPPER PAIZQ EXPRESION PADER

REDONDEO

: RROUND PAIZQ EXPRESION PADER

TRUNCAR

: RTRUNCATE PAIZQ EXPRESION PADER

ELENGTH

: RLENGTH PAIZQ EXPRESION PADER

ETYPEOF

: RTYPEOF PAIZQ EXPRESION PADER

ETOSTRING

: RTOSTRING PAIZQ EXPRESION PADER

## PRIMITIVO

: ENTERO

| DECIMAL

| CADENA

| CHARACTER

| RTRUE

| RFALSE

## TIPO

: RINT

| RDOUBLE

| RCHAR

| RSTRING

| RBOOLEAN

| RVOID

### Descripción de las Producciones

No Terminal	Descripción de Producción
<b>INICIO</b>	Es el estado inicial y con el que se empieza el análisis sintáctico, su producción es instrucciones.
<b>INSTRUCCIONES</b>	Este estado se encarga de formar la lista de instrucciones produciendo aun mas de su mismo estado o pasando a un estado INSTRUCCIÓN
<b>INSTRUCCIÓN</b>	Este estado es el encargado de la producción de cualquier sentencia de tipo instrucción como por ejemplo la instrucción para print, la instrucción para poder realizar un ciclo condicional if o la asignación y declaración para nuestras variables.
<b>FUNCPRINT</b>	Este estado se encarga de la producción para el manejo de la instrucción Print la cual utiliza las palabras reservadas como RPRINT o tokens como los paréntesis y el punto y coma además de utilizar una siguiente producción de tipo EXPRESION.
<b>INSDECLARAR</b>	Esta producción se encarga de manejar las diferentes formas en las que una variable puede llegar a ser declarada, como por ejemplo una variable que solo tenga tipo id punto y coma. O una variable a la cual se le asigna un valor desde el principio como por ejemplo tipo id igual expresión punto y coma.
<b>INSASIGNAR</b>	Esta producción se encarga de que podamos manejar el asignar un nuevo valor a una variable existente.
<b>ASIGNARFOR</b>	Esta producción nos permite manejar de una forma mas sencilla la asignación que puede llegar a suceder dentro del ciclo for.
<b>CASTEO</b>	Esta producción nos permite manejar el casteo de las variables.
<b>GUARDARFUNCION</b>	Esta producción nos permite guardar las diferentes funciones y métodos que creemos durante el programa.
<b>INSTIF</b>	Esta producción nos permite manejar la creación de la sentencia if
<b>INSTESE</b>	Esta producción nos permite manejar la creación de la sentencia else la cual puede ser solo un else o un else if.
<b>INSWHILE</b>	Esta producción nos permite manejar la creación de la sentencia while.
<b>INSFOR</b>	Esta producción nos permite manejar la creación de la sentencia for.
<b>INSSWITCH</b>	Esta producción nos permite manejar la sentencia switch case.
<b>CASOS</b>	Esta producción permite manejar el arreglo de casos de la sentencia switch.
<b>CASEST</b>	Esta producción nos permite crear el área donde se encuentran las instrucciones de los casos.
<b>RETORNO</b>	Esta instrucción nos permite el manejo de la sentencia return ya sea return y un valor o solamente return.
<b>INSSTATEMENT</b>	Esta producción nos permite el fácil manejo del área entre llaves donde las funciones y sentencias cíclicas y el if por ejemplo expresan la serie de instrucciones a realizar.

<b>PARAMETROS</b>	Esta producción nos permite manejar fácilmente los parámetros de las funciones viendo si son muchos parámetros o solo un parametro.
<b>PARAMETRO</b>	Esta producción nos permite la creación del parámetro de una función.
<b>EXPRESION</b>	Esta producción nos permite el fácil manejo de todas las producciones que sean de tipo expresión es decir que deben retornar alguna cosa. Por ejemplo, una operación o una función nativa.
<b>LLAMARFUNCION</b>	Esta producción nos permite realizar la llamada a una función por medio de su id los paréntesis y sus parámetros si es que se tuvieran.
<b>ARGUMENTOS</b>	Esta producción nos permite hacer el envío de los parámetros en la llamada de una función.
<b>ARITMETICA</b>	Esta producción se encarga del manejo de todas las operaciones entre expresiones que sean de tipo aritméticas es decir suma, resta, multiplicación, división, entre otros.
<b>RELACIONAL</b>	Esta producción se encarga del manejo de todas las operaciones entre expresiones de tipo relacional como mayor o menor que, o diferente.
<b>LOGICO</b>	Esta producción se encarga del manejo de las operaciones lógicas entre expresiones un ejemplo seria la operación lógica Or
<b>ACCEDERVARIABLE</b>	Esta producción nos permite acceder al valor que tenga una variable por medio de su id.
<b>INCREMENTODECREMENTO</b>	Esta producción se encarga del maneja de la expresión para el incremento y decremento del valor de una variable de tipo numérico por ejemplo i++
<b>TOLOWERUPPER</b>	Esta producción se encarga del manejo de las funciones nativas to lower y to upper.
<b>REDONDEO</b>	Esta producción se encarga del manejo de la función nativa round.
<b>TRUNCAR</b>	Esta producción se encarga del manejo de la función nativa truncate.
<b>ELENGTH</b>	Esta producción se encarga de la función nativa length.
<b>ETYPEOF</b>	Esta producción se encarga de la función nativa typeof.
<b>ETOSTRING</b>	Esta producción se encarga de la función nativa toString.
<b>PRIMITIVO</b>	Esta producción se encarga del manejo de todos los tipos primitivos para las variables como el entero, decimal o carácter.
<b>TIPO</b>	Esta producción se encarga de manejar el tipo de las variables o funciones por ejemplo int, double, char entre otros.