

# Gramática

ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1

# Tokens

Expresión regular	Token
"//".*	comentario lineal
[/][^/]*([/][^/]*+)([^\n/][^/]*+)*[/]	comentario multiple líneas
"int"	Rint
"double"	Rdouble
"boolean"	Rboolean
"char"	Rchar
"string"	Rstring
"list"	Rlist
"if"	Rif
"else"	Relse
"switch"	Rswitch
"case"	Rcase
"do"	Rdo
"while"	Rwhile
"for"	Rfor
"break"	Rbreak
"continue"	Rcontinue
"return"	Rreturn
"default"	Rdefault
"void"	Rvoid
"new"	Rnew
"add"	Radd

# Tokens

Expresión regular	Token
"setvalue"	Rsetvalue
"print"	Rprint
"true"	Rtrue
"false"	Rfalse
"main"	Rmain
"getvalue"	Rgetvalue
"tochararray"	Rtochararray
"tolower"	Rtolower
"toupper"	Rtoupper
"length"	Rlength
"truncate"	Rtruncate
"round"	Rround
"typeof"	Rtypeof
"tostring"	Rtostring
[0-9]+("."[0-9]+)\b	decimal
"."	punto
[0-9]+\b	entero
"=="	igualigual
"!="	diferente
"<="	menorlgual
"<"	menor
">="	mayorlgual

# Tokens

Expresión regular	Token
"="	igual
">"	mayor
","	coma
","	ptcoma
":"	dospuntos
"  "	or
"&&"	and
"{"	llaveA
"}"	llaveC
"*"	multi
"/"	div
"--"	menosmenos
"++"	masmas
"-"	menos
"+"	suma
"^"	exponente
"!"	not
"%"	modulo
"("	parA
")"	parC
"["	corA
"]"	corC

# Tokens

Expresión regular	Token
<code>([a-zA-Z])([a-zA-Z0-9_])*</code>	identificador
<code>"*["]</code>	string
<code>"*["]</code>	char

# Gramática

- Los terminales: Rint, Rdouble, Rchar, Rboolean, Rstring, Rlist, Rvoid, Rmain, Rnew, Rtochararray, Rif, Relse, Rwhile, Rdo, Rfor, Rswitch, Rcase, Rdefault, Rprint, Radd, Break, Continue, Return, identificador, parA, parC, corA, corC, llaveA, llaveC, punto, coma, igual, dospuntos, menor, mayor, ptcoma.
- Los no terminales son: INICIO, OPCIONESCUEPO, CUERPO, METODOS, LLAMADAMETODO, LLAMADAFUNCION, LISTAVALORES, FUNCIONES, LPARAMETROS, PARAMETROS, MAIN, DEC\_VAR, ASIG\_VAR, TIPO, INSTRUCCIONES, INSTRUCCION, PRINT, DEC\_VECTOR, ASIG\_VECTOR, DEC\_LISTA, ADD\_LISTA, ACCEDER\_LISTA, CHARARRAY, IF, ELSEIFS, CONELSEIF, SWITCH, CSWITCH, CONSWITCH, DEF, WHILE, DOWHILE, FOR, INSTRUCCIONFOR, BREAK, CONTINUE, RETURN.

```
INICIO:  OPCIONESCUEPO EOF
        | error ptcoma
```

```
OPCIONESCUEPO: OPCIONESCUEPO CUERPO
               | CUERPO
```

```
CUERPO: DEC_VAR ptcoma
        | ASIG_VAR ptcoma
        | METODOS
        | FUNCIONES
        | MAIN
        | DEC_VECTOR
        | ASIG_VECTOR
        | DEC_LISTA
        | ADD_LISTA
        | ACCEDER_LISTA
        | CHARARRAY
        | error
```

# Gramática

METODOS: Rvoid identificador parA parC llaveA INSTRUCCIONES llaveC  
| Rvoid identificador parA LPARAMETROS parC llaveA INSTRUCCIONES llaveC

LLAMADAMETODO: identificador parA parC ptcoma  
| identificador parA LISTAVALORES parC ptcoma

LLAMADAFUNCION: identificador parA parC  
| identificador parA LISTAVALORES parC

LISTAVALORES: LISTAVALORES coma EXPRESION  
| EXPRESION

FUNCIONES: TIPO identificador parA parC llaveA INSTRUCCIONES llaveC  
| TIPO identificador parA LPARAMETROS parC llaveA INSTRUCCIONES llaveC

LPARAMETROS: LPARAMETROS coma PARAMETROS  
| PARAMETROS

PARAMETROS: TIPO identificador  
| TIPO menor TIPO mayor identificador  
| TIPO TIPO identificador

MAIN: Rmain identificador parA parC ptcoma  
| Rmain identificador parA LISTAVALORES parC ptcoma

DEC\_VAR: TIPO identificador  
| TIPO identificador igual EXPRESION

ASIG\_VAR: identificador igual EXPRESION



# Gramática

TIPO: Rint

|Rdouble

|Rchar

|Rboolean

|Rstring

|Rlist

|corA corC

INSTRUCCIONES: INSTRUCCIONES INSTRUCCION

|INSTRUCCION

INSTRUCCION: LLAMADAFUNCION

|LLAMADAMETODO

|DEC\_VAR ptcoma

|ASIG\_VAR ptcoma

|PRINT

|DEC\_VECTOR

|ASIG\_VECTOR

|DEC\_LISTA

|ADD\_LISTA

|ACCEDER\_LISTA

|CHARARRAY

|CASTEO

|IF

|SWITCH

|WHILE

|DOWHILE

|FOR

|BREAK

|CONTINUE

|RETURN

|INSTRUCCIONFOR ptcoma

| error ptcoma



# Gramática

PRINT: Rprint parA EXPRESION parC ptcoma

DEC\_VECTOR: TIPO corA corC identificador igual Rnew TIPO corA EXPRESION corC ptcoma  
| TIPO corA corC identificador igual llaveA LISTAVALORES llaveC ptcoma

ASIG\_VECTOR: identificador corA EXPRESION corC igual EXPRESION ptcoma

DEC\_LISTA: Rlist menor TIPO mayor identificador igual Rnew Rlist menor TIPO mayor ptcoma

ADD\_LISTA: identificador punto Radd parA EXPRESION parC ptcoma

ACCEDER\_LISTA: identificador corA corA EXPRESION corC corC

CHARARRAY: Rlist menor TIPO mayor identificador igual Rtochararray parA EXPRESION parC ptcoma

IF: Rif parA EXPRESION parC llaveA INSTRUCCIONES llaveC  
| Rif parA EXPRESION parC llaveA INSTRUCCIONES llaveC Relse llaveA INSTRUCCIONES llaveC  
| Rif parA EXPRESION parC llaveA INSTRUCCIONES llaveC ELSEIFS  
| Rif parA EXPRESION parC llaveA INSTRUCCIONES llaveC ELSEIFS Relse llaveA INSTRUCCIONES  
llaveC

ELSEIFS: ELSEIFS CONELSEIF  
|CONELSEIF

CONELSEIF: Relse Rif parA EXPRESION parC llaveA INSTRUCCIONES llaveC

# Gramática

```
SWITCH: Rswitch parA EXPRESION parC llaveA CSWITCH DEF llaveC
      | Rswitch parA EXPRESION parC llaveA CSWITCH llaveC
      | Rswitch parA EXPRESION parC llaveA DEF llaveC
```

```
CSWITCH: CSWITCH CONSWITCH
      | CONSWITCH
```

```
CONSWITCH: Rcase EXPRESION dospuntos INSTRUCCIONES
```

```
DEF: Rdefault dospuntos INSTRUCCIONES
```

```
WHILE: Rwhile parA EXPRESION parC llaveA INSTRUCCIONES llaveC
```

```
DOWHILE: Rdo llaveA INSTRUCCIONES llaveC Rwhile parA EXPRESION parC ptcoma
      | Rdo llaveA llaveC Rwhile parA EXPRESION parC ptcoma
```

```
FOR: Rfor parA CUERPOFOR ptcoma EXPRESION ptcoma INSTRUCCIONFOR parC llaveA INSTRUCCIONES llaveC
    | Rfor parA CUERPOFOR ptcoma EXPRESION ptcoma INSTRUCCIONFOR parC llaveA llaveC
```

```
CUERPOFOR: DEC_VAR
          | ASIG_VAR
```

```
INSTRUCCIONFOR: INCREMENTO
              | DECREMENTO
```

```
INCREMENTO: identificador masmas
```

```
DECREMENTO: identificador menosmenos
```

# Gramática

BREAK: Rbreak ptcoma

CONTINUE: Rcontinue ptcoma

RETURN: Rreturn ptcoma  
| Rreturn EXPRESION ptcoma

CASTEO: parA TIPO parC VAL\_CAST

VAL\_CAST: decimal  
| entero  
| string  
| char

# Gramática

```
EXPRESION: EXPRESION suma EXPRESION
| EXPRESION menos EXPRESION
| EXPRESION multi EXPRESION
| EXPRESION div EXPRESION
| EXPRESION exponente EXPRESION
| EXPRESION modulo EXPRESION
| EXPRESION menor EXPRESION
| EXPRESION mayor EXPRESION
| EXPRESION menorIgual EXPRESION
| EXPRESION mayorIgual EXPRESION
| EXPRESION diferente EXPRESION
| EXPRESION and EXPRESION
| EXPRESION or EXPRESION
| menos EXPRESION umenos
| not EXPRESION
| parA EXPRESION parC
| EXPRESION igualigual EXPRESION
| decimal
| entero
| Rtrue
| Rfalse
| string
| char
| Rtoupper parA EXPRESION parC
| Rtolower parA EXPRESION parC
| Rlength parA EXPRESION parC
| Rtruncate parA EXPRESION parC
| Rround parA EXPRESION parC
| Rtypeof parA EXPRESION parC
| Rtostring parA EXPRESION parC
| identificador corA EXPRESION corC
| Rgetvalue parA identificador coma EXPRESION parC
| identificador
| LLAMADAFUNCION
| ACCEDER_LISTA
| CASTEO
```

# Gramática

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Proyecto 2 [COMPI1]

Primer semestre

Oscar Eduardo Morales Girón

201603028