

**INSTITUTO TECNOLÓGICO DE TOLUCA**

Ingeniería en Sistemas Computacionales

Lenguajes y Autómatas II

Código Intermedio

Integrantes:

Hernández Carmona Cristopher

Muñoz Sánchez René

#### Metepec, Estado de México, noviembre de 2017

**S.E.P S.E.S.T.N.M. T.N.M**

Contenido

[Gramática y Rutinas Semánticas 3](#_Toc497807716)

[Programa ejemplo 5](#_Toc497807717)

[Resultado del programa ejemplo 6](#_Toc497807718)

# Gramática y Rutinas Semánticas

|  |  |
| --- | --- |
| **Sentencia** | **Rutina** |
| <Sentencia>🡪IF(<CondicionLogica>){<Sentencia>} | EV=Nueva Etiqueta  EF=Nueva Etiqueta  ESig=Nueva Etiqueta  <sentencia>.codigo= <CondicionLogica>.codigo ||genera (EV”:”)|| <sentencia>.codigo || genera (“goto” ESig) || genera( EF ”:”) || <sentencia>.codigo |
| <Sentencia>🡪ELSE{<Sentencia>} | <Sentencia>.codigo=<Sentencia>.codigo || generar(ESig “:”)|| |
| <Secuencia>🡪WHILE(<CondicionLogica>) {Secuencia} | EI=Nueva Etiqueta  EV=Nueva Etiqueta  EF =Nueva  <sentencia>.codigo= genera(EI”:”)||  <condicionLogica>.codigo|| genera(EV”:”)|| <sentencia>.codigo || genera(“goto” EI)|| genera(EF “:”) |
| <CondicionLogica>🡪<ExpLogica>l | <CondicionLogica>.codigo= Crear pila de lexemas y pasar a posfijo pila de lexemas || <ExpLogica>. Código |
| <ExpLogica> 🡪 <ExpLogica> AND <ExpLogica> | Ev1 = Nueva etiqueta  EF1= EF  Ev2= EV  EF2= EF  <ExpLogica>.codigo= <ExpLogica>.codigo || genera(EV1 “:”) || <ExpLogica>.codigo |
| <ExpLogica> 🡪 <ExpLogica> OR <ExpLogica> | EV1=EV  EF1=Nueva etiqueta  EV2=EV  EF2=EF  <ExpLogica>.codigo= <ExpLogica>.codigo || genera(EF “:”) || <ExpLogica>.codigo |
| <ExpLogica> 🡪 NOT <ExpLogica> | EV1=EF  EF1=EV  <ExpLogica>.codigo=<ExpLogica>.codigo |
| <ExpLogica> 🡪 <exp1>OpeRelacional<exp2> | <ExpLogica>.codigo=<exp1>.codigo || <exp2>.codigo || genera(“IF” ret exp1 OpeRelacional ret exp2 “goto” EV)|| genera(“goto” EF) |
| <sentencia>🡪 TipoDato id=<exp> | <sentencia>.codigo= Formar pila de lexema y enviar a posfijo ||<exp>.codigo ||generar(TipoDato id retorno exp) |
| <exp>🡪 id |  |
| <exp>🡪 (<exp>) | <exp>.codigo=<exp>.codigo |
| <exp>🡪<exp>OA<exp2> | <exp>.codigo= Temporal=Nuevo||  <exp>.codigo|| genera(retorno exp OA retorno exp2) |
| <sentencia>🡪 TipoDato id; | <sentencia>.codigo= generar(TipoDato id) |
| <sentencia>🡪} <sentencia> | <sentencia>.codigo= retorno |
| <sentencia>🡪 FOR (<S1>:<CondicionLogica>:<S2>{<sentencia>} | <sentencia>.codigo=  EI=Nueva etiqueta  EV= Nueva Etiqueta  EF= Nueva Etiqueta ||  <S1>.codigo||<CondicionLogica>.codigo|| genera(EV “:”) ||<sentencia>.codigo || <S2>.codigo  || genera (“goto” EI) || genera(EF “:”)|| <S2>.codigo |
| <S1>🡪TipoDato id=<exp> | <S1>.codigo= Formar pila de lexema y enviar a posfijo ||  generar(TipoDato id “= “retorno <exp>.codigo)|| genera(EI”:”) |
| <S2>🡪id ++ | <S2>.codigo= generar(id”=”id+1) |
| <S2>🡪id -- | <S2>.codigo= generar(id”=”id-1) |

# Programa ejemplo

#INICIO

int a = 2;

float b = 1;

int c = 3;

float d = a \* c;

switch (5+8\*6/4) {

case 1 {

msj "uno";

for (int i=0:i<10:i++) {

msj "el resultado es" + i;

if (a>d) {

msj "entra";

}

else {

msj "no entra";

}

}

break;

}

case 2 {

msj "dos";

while (a < 10) {

msj "entra while";

a = a + 1;

if (d<5) {

msj "entra";

}

else {

msj "no entra";

}

}

break;

}

}

#

# Resultado del programa ejemplo

Al momento de correr nuestro programa con el código fuente mostrado anteriormente nos dará como resultado el siguiente código intermedio:

a = 2

b = 1

c = 3

T1 = a \* c

d = T1

T2 = 8 \* 6

T3 = T2 / 4

T4 = 5 + T3

IF T4 == 1 GOTO 20

GOTO 30

20:

PRINT "uno "

i = 0

40:

IF i < 10 GOTO 50

GOTO 60

50:

PRINT "el resultado es "

PRINT i

IF a > d GOTO 70

GOTO 80

70:

PRINT "entra "

GOTO 90

80:

PRINT "no entra "

90:

i + 1

GOTO 40

60:

GOTO 0

30:

IF T4 == 2 GOTO 110

GOTO 120

110:

PRINT "dos "

130:

IF a < 10 GOTO 140

GOTO 150

140:

PRINT "entra while "

T5 = a + 1

a = T5

IF d < 5 GOTO 160

GOTO 170

160:

PRINT "entra "

GOTO 180

170:

PRINT "no entra "

180:

GOTO 130

150:

GOTO 0

120: