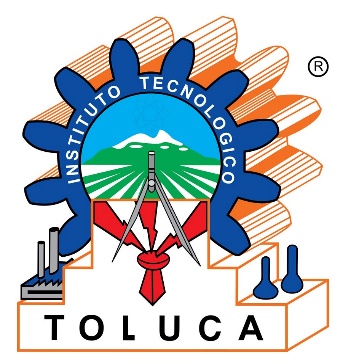
**SEP**

**SNESTM**

**TNM**

**INSTITUTO TECNOLÓGICO DE TOLUCA**



Ingeniería en Sistemas Computacionales

**Lenguajes y Autómatas II**

**Análisis Semántico**

P R E S E N T A:

Muñoz Sánchez René

#### Metepec, Estado de México; septiembre 2017

CONTENIDO

[1. LENGUAJE “CLAPEE” 4](#_Toc486890153)

[Objetivo 4](#_Toc486890154)

[2. PALABRAS RESERVADAS 5](#_Toc486890155)

[3. TABLAS 6](#_Toc486890156)

[Tabla de símbolos 6](#_Toc486890157)

[Tabla de errores 6](#_Toc486890158)

[Tablas de resultados 6](#_Toc486890159)

[Operador Aritmético 6](#_Toc486890160)

[Asignación 7](#_Toc486890161)

[Operador relacional 7](#_Toc486890162)

[Operador lógico 8](#_Toc486890163)

[Tabla de Prioridades 9](#_Toc486890164)

[4. GRAMÁTICA 9](#_Toc486890165)

[Sentencias 9](#_Toc486890166)

[Clase 10](#_Toc486890167)

[Declaración y asignación 10](#_Toc486890168)

[Método 1 con retorno 10](#_Toc486890169)

[Método 2 vacío 11](#_Toc486890170)

[Parámetros 11](#_Toc486890171)

[Asignación 2 11](#_Toc486890172)

[Valor 12](#_Toc486890173)

[SI 12](#_Toc486890174)

[Exp 12](#_Toc486890175)

[Switch 12](#_Toc486890176)

[Caso 13](#_Toc486890177)

[Condición 13](#_Toc486890178)

[Contador 13](#_Toc486890179)

[Mientras 14](#_Toc486890180)

[Para 14](#_Toc486890181)

[5. RUTINA SEMÁNTICA 15](#_Toc486890182)

[Sentencias 15](#_Toc486890183)

[Clase 15](#_Toc486890184)

[Declaración y asignación 15](#_Toc486890185)

[Método 1 con retorno 16](#_Toc486890186)

[Método 2 vacío 17](#_Toc486890187)

[Parámetros 17](#_Toc486890188)

[Asignación 2 17](#_Toc486890189)

[Valor 18](#_Toc486890190)

[SI 18](#_Toc486890191)

[Exp 19](#_Toc486890192)

[Switch 20](#_Toc486890193)

[Caso 20](#_Toc486890194)

[Condición 20](#_Toc486890195)

[Contador 21](#_Toc486890196)

[Mientras 21](#_Toc486890197)

[Para 21](#_Toc486890198)

# LENGUAJE “CLAPEE”

## Objetivo

Diseñar un lenguaje similar a Java en español conservando algunas de sus estructuras como Para, Mientras, Asignación, Switch para realizar un análisis semántico, creando una rutina semántica partiendo de la gramática y posteriormente la rutina de código intermedio, asi como la optimización de código y creación de código objeto correspondientes a la materia de Lenguajes y Autómatas 2, con la finalidad de aprobar la materia.

|  |  |
| --- | --- |
| Tipos de datos | |
| Lenguaje Java | Lenguaje ASDF |
| Int  Dooble  String  Boolean  Arreglo  Matriz  Char | ent 1  dec 1  cad 1  bool 1  arre 1  mat 1  ch 1 |

|  |  |
| --- | --- |
| Operadores relacionales | |
| Lenguaje Java | Lenguaje ASDF |
| Menor que  Mayor que  Menor igual que  Mayor igual que  Igual que  Diferente | < 80  > 80  <= 80  >= 80  == 80  ! 80 |

|  |  |
| --- | --- |
| Operadores aritméticos | |
| Lenguaje Java | Lenguaje ASDF |
| Suma  Resta  Multiplicación  División | + 60  - 2  \* 2  / 2 |

|  |  |
| --- | --- |
| Operadores lógicos | |
| Lenguaje Java | Lenguaje ASDF |
| True  False | Verdadero 26  Falso 27 |

|  |  |
| --- | --- |
| Métodos hechos | |
| Lenguaje Java | Lenguaje ASDF |
| System.out.println | leer 28  Imprimir 25 |

|  |  |
| --- | --- |
| Tipos de modificadores de acceso | |
| Lenguaje Java | Lenguaje ASDF |
| Public  Private  Protected | Publico 7  Privado 61  Protegido 62 |

|  |  |
| --- | --- |
| Creación de clases | |
| Lenguaje Java | Lenguaje ASDF |
| class | Clase 6 |

|  |  |
| --- | --- |
| Ciclos | |
| Lenguaje Java | Lenguaje ASDF |
| For  Do  while | Para 8  Do 12  Mientras 15 |

|  |  |
| --- | --- |
| Selección | |
| Lenguaje Java | Lenguaje ASDF |
| If  Else  Case  Switch | Si 9  Sino 10  Caso 20  Switch 24 |

|  |  |
| --- | --- |
| Creación de métodos | |
| Lenguaje Java | Lenguaje ASDF |
| Return  void | Regresa 21  Metodo 22  Vacio 23 |

|  |  |
| --- | --- |
| Separadores especiales | |
| Lenguaje Java | Lenguaje ASDF |
| ;  (  )  {  }  [  ]  :  ,  @52  \  # | ; 35  ( 36  ) 37  { 38  } 39  [ 40  ] 41  : 43  , 44  @ 52  \ 53  # 54 |

|  |  |
| --- | --- |
| Variables | |
|  | Lenguaje ASDF |
|  | 499 entero  500 decimal  501 cadena  502 caracter  503 variable  504 arreglo  505 matriz |

# PALABRAS RESERVADAS

|  |  |
| --- | --- |
| Palabras reservadas con No Token | |
| ent 1  dec 1  cad 1  bool 1  arre 1  mat 1  ch 1  Name 5  Clase 6  Publico 7  Para 8  Si 9  Sino 10  Importar 11  Do 12  Codigo 13  Pack 14  Mientras 15  Nuevo 16  Fin 17  Excepcion 18  Caso 20  Regresa 21  Metodo 22  Vacio 23  Switch 24  Imprimir 25  Verdadero 26  Falso 27 | leer 28  Privado 61  Protegido 62  Ejecuta 63  booleano 64  ; 35  ( 36  ) 37  { 38  } 39  [ 40  ] 41  = 42  : 43  , 44  " 45  < 46  > 46  ? 48  ´ 49  . 50  ! 51  @ 52  \ 53  # 54 |

# TABLAS

## Tabla de símbolos

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | TD | Tipo | Uso | Global | ValorEnt | ValorCad | ValorDec | ValorChar | ValorBol | Apuntador |
|  |  |  |  |  |  |  |  |  |  |  |

## Tabla de errores

|  |  |
| --- | --- |
| No. Error | Tipo de error |
| -100 | Variable repetida |
| -101 | Incompatibilidad de tipo de datos |
| -105 | Variable no declarada |

|  |  |
| --- | --- |
| Error | NoRenglon |
|  |  |

## Tablas de resultados

### Operador Aritmético

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| suma (+) | ent | cad | bool | dec | ch |
| ent | ent | -101 | -101 | dec | -101 |
| cad | -101 | -101 | -101 | -101 | -101 |
| bool | -101 | -101 | -101 | -101 | -101 |
| dec | dec | -101 | -101 | dec | -101 |
| ch | -101 | -101 | -101 | -101 | -101 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| resta (-) | ent | cad | bool | dec | ch |
| ent | ent | -101 | -101 | dec | -101 |
| cad | -101 | -101 | -101 | -101 | -101 |
| bool | -101 | -101 | -101 | -101 | -101 |
| dec | dec | -101 | -101 | dec | -101 |
| ch | -101 | -101 | -101 | -101 | -101 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| multiplicacion (\*) | ent | cad | bool | dec | ch |
| ent | ent | -101 | -101 | dec | -101 |
| cad | -101 | -101 | -101 | -101 | -101 |
| bool | -101 | -101 | -101 | -101 | -101 |
| dec | dec | -101 | -101 | dec | -101 |
| ch | -101 | -101 | -101 | -101 | -101 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| division (/) | ent | cad | bool | dec | ch |
| ent | ent | -101 | -101 | dec | -101 |
| cad | -101 | -101 | -101 | -101 | -101 |
| bool | -101 | -101 | -101 | -101 | -101 |
| dec | dec | -101 | -101 | dec | -101 |
| ch | -101 | -101 | -101 | -101 | -101 |

### Asignación

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| asignacion | ent | cad | bool | dec | ch |
| ent | ent | -101 | -101 | -101 | -101 |
| cad | -101 | cad | -101 | -101 | cad |
| bool | -101 | -101 | bool | -101 | -101 |
| dec | -101 | -101 | -101 | dec | -101 |
| ch | -101 | -101 | -101 | -101 | ch |

### Operador relacional

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| < | ent | cad | dec | ch | bool |
| ent | bool | -101 | bool | -101 | -101 |
| cad | -101 | -101 | -101 | -101 | -100 |
| dec | bool | -101 | bool | -100 | -100 |
| Ch | -101 | -101 | -101 | -101 | -101 |
| bool | -101 | -101 | -101 | -101 | -101 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| > | ent | cad | dec | ch | bool |
| Ent | bool | -101 | bool | -101 | -101 |
| Cad | -101 | -101 | -101 | -101 | -100 |
| Dec | bool | -101 | bool | -100 | -100 |
| Ch | -101 | -101 | -101 | -101 | -101 |
| Bool | -101 | -101 | -101 | -101 | -101 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| >= | ent | cad | dec | ch | bool |
| ent | bool | -101 | bool | -101 | -101 |
| cad | -101 | -101 | -101 | -101 | -100 |
| dec | bool | -101 | bool | -100 | -100 |
| Ch | -101 | -101 | -101 | -101 | -101 |
| bool | -101 | -101 | -101 | -101 | -101 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| <= | ent | cad | dec | ch | bool |
| ent | bool | -101 | bool | -101 | -101 |
| cad | -101 | -101 | -101 | -101 | -100 |
| dec | bool | -101 | bool | -100 | -100 |
| Ch | -101 | -101 | -101 | -101 | -101 |
| bool | -101 | -101 | -101 | -101 | -101 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| == | ent | cad | dec | ch | bool |
| ent | bool | -101 | -101 | -101 | -101 |
| cad | -101 | bool | -101 | -101 | -101 |
| dec | -101 | -101 | bool | -101 | -101 |
| Ch | -101 | -101 | -101 | bool | -101 |
| bool | -101 | -101 | -101 | -101 | bool |

### Operador lógico

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AND | ent | cad | dec | ch | bool |
| ent | -101 | -101 | -101 | -101 | -101 |
| cad | -101 | -101 | -101 | -101 | -101 |
| dec | -101 | -101 | -101 | -101 | -101 |
| Ch | -101 | -101 | -101 | -101 | -101 |
| bool | -101 | -101 | -101 | -101 | bool |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OR | ent | cad | dec | ch | bool |
| ent | -101 | -101 | -101 | -101 | -101 |
| cad | -101 | -101 | -101 | -101 | -101 |
| dec | -101 | -101 | -101 | -101 | -101 |
| Ch | -101 | -101 | -101 | -101 | -101 |

## Tabla de Prioridades

|  |  |  |
| --- | --- | --- |
| Tipo | Signo | Token |
| aritmético | + | 1 |
| aritmético | - | 1 |
| aritmético | \* | 2 |
| aritmético | / | 2 |
| relacional | < | 3 |
| relacional | > | 3 |
| relacional | >= | 3 |
| relacional | <= | 3 |
| relacional | == | 3 |
| lógico | AND | 4 |
| lógico | OR | 5 |
| lógico | NOT | 6 |
| asignación | = | 7 |

# GRAMÁTICA

## Sentencias

|  |
| --- |
| Gramática |
| <sentencias> → <DecAsig><sentencias> |
| <sentencias> → <Metodo1><sentencias> |
| <sentencias> → <Metodo2><sentencias> |
| <sentencias>→<Asig2><sentencias> |
| <sentencias> → <Contador><sentencias> |
| <sentencias> → <Mientras><sentencias> |
| <sentencias> → <Para><sentencias> |
| <sentencias> → <Inicializacion><sentencias> |
| <sentencias> → <Si><sentencias> |
| <sentencias> →<Switch><sentencias> |

## Clase

|  |  |
| --- | --- |
| Gramática | |
| <clase> → Publico <S1> | <clase> → 7 <S1> |
| <clase> → Privado <S1> | <clase> → 61 <S1> |
| <clase> → Protegido <S1> | <clase> → 62 <S1> |
| <S1> → Clase Variable { <sentencias> } | <S1> → 6 503 38 <sentencias> 39 |

\*No busca la variable en la TS porque no puede haber algo antes que una clase, ni tampoco más de una clase.

## Declaración y asignación

|  |  |
| --- | --- |
| Gramática | |
| <DecAsig> → Publico  <S2> | <DecAsig> → 7<S2> |
| <DecAsig> → Privado  <S2> | <DecAsig> → 61<S2> |
| <DecAsig> → Protegido  <S2> | <DecAsig> → 62<S2> |
| <S2> → TD variable  <S4> | <S2> → 1 50<S4> |
| <S3>→ ; | <S3>→ 35 |
| <S3> → , variable <S4> | <S3> → 44 50<S4> |
| <S4> → = <valor><s3> | <S4> → 42<valor>  <s3> |
| <S4> → <S3> | <S4> → <S3> |

\*En nuestra gramática no asignamos expresiones directamente.

## Método 1 con retorno

|  |  |
| --- | --- |
| Gramática | |
| <Metodo1> → Metodo <S5> | <Metodo1> → 22 <S5> |
| <S5> → Privado <S6> | <S5> → 61 <S6> |
| <S5> → Publico <S6> | <S5> → 7 <S6> |
| <S5> → Protegido <S6> | <S5> → 62 <S6> |
| <S6> → <S6'> | <S6> → <S6'> |
| <S6> → <S10> | <S6> → <S10> |
| <S6'> → TD variable ( <S7> | <S6> → 1 503 36 <S7> |
| <S7> → <parametro> ) {<sentencias> Regresa <S8> | <S7> → <parametro> 37 38<sentencias> 21 <S8> |
| <S7> → ) { <sentencias>  Regresa <S8> | <S7> → 37 38<sentencias> 21 <S8> |
| <S8> → <exp> ; } | <S8> → <exp> 35 39 |
| <S8> → <Valor>;} | <S8> → <Valor> 35 39 |

## Método 2 vacío

|  |  |
| --- | --- |
| Gramática | |
| <S10> → Vacio  variable (  <S11> | <S10> → 23 503 36 <S11> |
| <S11> → <parametro>  ) { <sentencias> } | <S11> →<parametro>37 38 <sentencias> 39 |
| <S11> → ) {  <sentencias> } | <S11> → 37 38 <sentencias> 39 |

## Parámetros

|  |  |
| --- | --- |
| Gramática | |
| <parametro> → TD variable<p> | <parametro> → 1 503 <p> |
| <p> → ,<parametros> | <p> → 44 <parametros> |
| <p> →λ | <p> → puede ser terminal |

## Asignación 2

|  |  |
| --- | --- |
| Gramática | |
| <Asig2> → variable = <S12> | <Asig2> → 503 42 <S12> |
| <S12> → <exp><S13> | <S12> → <exp>  <S13> |
| <S12> → <leer><S13> | <S12> → <leer>  <S13> |
| <S12> → variable <S13> | <S12> → 503 <S13> |
| <S12> → <valor><S13> | <S12> → <valor>  <S13> |
| <S13> → ; | <S13> → 35 |

## Valor

|  |  |
| --- | --- |
| Gramática | |
| <valor> → CADENA | <valor> → 501 |
| <valor> → ENTERO | <valor> → 499 |
| <valor> → DECIMAL | <valor> → 500 |
| <valor> → CARÁCTER | <valor> → 502 |
| <valor> → BOOLEANO(TRUE) | <valor> → 26 |
| <valor> → BOOLEANO(FALSE) | <valor> → 26 |

## SI

|  |  |
| --- | --- |
| Gramática | |
| <Si> → Si <S20> | <Si> → 9 <S20> |
| <S20>→ ( <condición> ) { <sentencia>} <S21> | <S20>→ 36 <condición> 37 38 <sentencia>39 <S21> |
| <S21>→ <SinoSi> | <S21>→ <SinoSi> |
| <S21>→<Sino> | <S21>→<Sino> |
| <Sino> → { <sentencia>} | <Sino> → 38 <sentencia>39 |
| <Sino>→ λ | <Sino>→ λ |
| <SiNoSi>→ Sino <Si> | <SiNoSi>→ 10 <Si> |

## Exp

|  |  |
| --- | --- |
| Gramática | |
| <exp> → <exp> OA <exp> | <exp> → <exp> OA <exp> |
| <exp> → ( <exp> ) | <exp> → 36 <exp> 37 |
| <exp> → variable | <exp> → 503 |
| <exp> → <valor> | <exp> → <valor> |

## Switch

|  |  |
| --- | --- |
| Gramática | |
| <Switch>→ Switch (<S25> ){<caso>} | <Switch>→ 24 36 <S25> 37 38 <caso>39 |
| <S25> → <valor> | <S25> → <valor> |
| <S25> → variable | <S25> → 503 |

## Caso

|  |  |
| --- | --- |
| Gramática | |
| <Caso> → Caso( <S26>) : <sentencia> Fin <S27> | <Caso> → 20 36 <s26>37 43 <sentencia> 17 <s27> |
| <S26> → <valor> | <S26> → <valor> |
| <S26> → variable | <S26> → 503 |
| <S27> → <Caso> | <s27> → <Caso> |
| <S27>→λ | <s27> λ |

## Condición

|  |  |
| --- | --- |
| Gramática | |
| <Condicion> → <exp>OR<exp><S14> | <Condicion> → <exp>80<exp><S14> |
| <S14> → OL <Condicion> | <S14> → 03 <Condicion> |
| <S14> → λ | <S14> → λ |

## Contador

|  |  |
| --- | --- |
| Gramática | |
| <Contador> → variable <s15> | <Contador> → 503 <s15> |
| <s15> → + + | <s15> → + + |
| <s15> → - - | <s15> → - - |

## Mientras

|  |  |
| --- | --- |
| Gramática | |
| <Mientras> → Mientras (<condición>){<sentencias>} | <Mientras> → 15 36<condición>37 38<sentencias>39 |

## Para

|  |  |
| --- | --- |
| Gramática | |
| <Para> → Para (<Inicialización>;<condición>;<contador>){<sentencias>} | <Para> → 8 36<Inicialización>35<condición>35<contador>37 38 <sentencias>39 |
| <Inicializacion> → TD<S16> | <Inicializacion> → 1 <S16> |
| < Inicializacion > → variable<s18> | < Inicializacion > → 503 <s18> |
| <s16> → variable<s17> | <s16> → 503 <s17> |
| <s18> → =<s19> | <s18> → 42 <s19> |
| <s19> → <valor> | <s19> → <valor> |
| <s19> → variable | <s19> → 503 |

# RUTINA SEMÁNTICA

## Sentencias

|  |  |
| --- | --- |
| Gramática | Rutina semántica |
| <sentencias> → <DecAsig><sentencias> | <sentencias>.codigo = <DecAsig>.codigo  <sentencias>.codigo |
| <sentencias> → <Metodo1><sentencias> | <sentencias>.codigo =<Metodo1>.codigo  <sentencias>.codigo |
| <sentencias> → <Metodo2><sentencias> | <sentencias>.codigo = <Metodo2>.codigo  <sentencias>.codigo |
| <sentencias>→<Asig2><sentencias> | <sentencias>.codigo = <Asig2>.codigo  <sentencias>.codigo |
| <sentencias> → <Contador><sentencias> | <sentencias>.codigo = <Contador>.codigo  <sentencias>.codigo |
| <sentencias> → <Mientras><sentencias> | <sentencias>.codigo = <Mientras>.codigo  <sentencias>.codigo |
| <sentencias> → <Para><sentencias> | <sentencias>.codigo = <Para>.codigo  <sentencias>.codigo |
| <sentencias> → <Inicializacion><sentencias> | <sentencias>.codigo = <Inicializacion>.codigo  <sentencias>.codigo |
| <sentencias> → <Si><sentencias> | <sentencias>.codigo = <Si>.codigo  <sentencias>.codigo |
| <sentencias> →<Switch><sentencias> | <sentencias>.codigo = <Switch>.codigo  <sentencias>.codigo |

## Clase

|  |  |  |
| --- | --- | --- |
| Gramática | | Rutina Semántica |
| <clase> → Publico <S1> | <clase> → 7 <S1> | <Clase>.codigo = <S1>.codigo |
| <clase> → Privado <S1> | <clase> → 61 <S1> | <Clase>.codigo = <S1>.codigo |
| <clase> → Protegido <S1> | <clase> → 62 <S1> | <Clase>.codigo = <S1>.codigo |
| <S1> → Clase Variable { <sentencias> } | <S1> → 6 503 38 <sentencias> 39 | <S1>.codigo = Inserta TS(variable)  <sentencias>.codigo |

\*No busca la variable en la TS porque no puede haber algo antes que una clase, ni tampoco más de una clase.

## Declaración y asignación

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina Semántica | Error |
| <DecAsig> → Publico  <S2> | <DecAsig> → 7<S2> | <DecAsig>.codigo = <S2>.codigo |  |
| <DecAsig> → Privado  <S2> | <DecAsig> → 61<S2> | <DecAsig>.codigo = <S2>.codigo |  |
| <DecAsig> → Protegido  <S2> | <DecAsig> → 62<S2> | <DecAsig>.codigo = <S2>.codigo |  |
| <S2> → TD variable  <S4> | <S2> → 1 50<S4> | <S2>.codigo = TD1=TD var1=variable  <S4>.codigo |  |
| <S3>→ ; | <S3>→ 35 | <S3>.codigo = Busca TS(var1)  Si existe manda error(-100)  SINO Agrega a a TS(var1) | -100 |
| <S3> → , variable <S4> | <S3> → 44 50<S4> | <S3>.codigo = Inserta TS(var1,TD1)  var1=variable  <S4>.codigo | -100 |
| <S4> → = <valor><s3> | <S4> → 42<valor>  <s3> | <S4>.codigo = Busca TS (var1)  Si existe manda error(-100)  SINO TDval = <valor>.codigo  Compara TS(TDval=TD1)  Si son diferentes manda error (-101)  <S3>.codigo | -100  -101 |
| <S4> → <S3> | <S4> → <S3> | <S4>.codigo = <S3>.codigo |  |

\*En nuestra gramática no asignamos expresiones directamente.

## Método 1 con retorno

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina Semántica | Error |
| <Metodo1> → Metodo <S5> | <Metodo1> → 22 <S5> | <Metodo1>.codigo = <S5>.codigo |  |
| <S5> → Privado <S6> | <S5> → 61 <S6> | <S5>.codigo = <S6>.codigo |  |
| <S5> → Publico <S6> | <S5> → 7 <S6> | <S5>.codigo = <S6>.codigo |  |
| <S5> → Protegido <S6> | <S5> → 62 <S6> | <S5>.codigo = <S6>.codigo |  |
| <S6> → <S6'> | <S6> → <S6'> | <S6>.codigo = <S6'>.codigo |  |
| <S6> → <S10> | <S6> → <S10> | <S6>.codigo = <S10>.codigo |  |
| <S6'> → TD variable ( <S7> | <S6> → 1 503 36 <S7> | <S6>.codigo = Busca TS(variable)  Si existe manda error (-100)  SINO TD1=TD var1=variable  Inserta TS (var1, TD1)  <S7>.código | -100 |
| <S7> → <parametro> ) {<sentencias> Regresa <S8> | <S7> → <parametro> 37 38<sentencias> 21 <S8> | <S7>.codigo = <parametros>.codigo  <sentencias>.codigo  <S8>.codigo |  |
| <S7> → ) { <sentencias>  Regresa <S8> | <S7> → 37 38<sentencias> 21 <S8> | <S7>.codigo = <sentencias>.codigo  <S8>.codigo |  |
| <S8> → <exp> ; } | <S8> → <exp> 35 39 | <S8>.codigo = <exp>.codigo |  |
| <S8> → <Valor>;} | <S8> → <Valor> 35 39 | <S8>.codigo = TD1=<Valor>.codigo  Comparar(TD = TD1)  Si son diferentes manda error(-101) | -101 |

## Método 2 vacío

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina Semántica | Error |
| <S10> → Vacio  variable (  <S11> | <S10> → 23 503 36 <S11> | <S10>.codigo =Busca TS(variable)  Si existe manda error (-100)  SINO TD1=TD var1=variable  Inserta TS(var1,TD1)  <S11>.codigo | -100 |
| <S11> → <parametro>  ) { <sentencias> } | <S11> →<parametro>37 38 <sentencias> 39 | <S11>.codigo = <parametro>.codigo  <sentencias>.codigo |  |
| <S11> → ) {  <sentencias> } | <S11> → 37 38 <sentencias> 39 | <S11>.codigo = <sentencias>.codigo |  |

## Parámetros

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina Semántica | Error |
| <parametro> → TD variable<p> | <parametro> → 1 503 <p> | <parametro>.codigo = Busca TS(variable)  Si existe manda error (-100)  SINO TD1=TD var1=variable  Inserta TS(var1,TD1)  <p>. codigo | -100 |
| <p> → ,<parametros> | <p> → 44 <parametros> | <p>.codigo = <parametros>.codigo |  |
| <p> →λ | <p> → puede ser terminal | <p>.codigo = retorna(λ) |  |

## Asignación 2

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina Semántica | Error |
| <Asig2> → variable = <S12> | <Asig2> → 503 42 <S12> | <Asig2>.codigo = Busca TS(variable)  SINO existe manda error(-105)  <S12>.codigo | -105 |
| <S12> → <exp><S13> | <S12> → <exp>  <S13> | <S12>.codigo = <exp>.codigo // <S13>.codigo |  |
| <S12> → <leer><S13> | <S12> → <leer>  <S13> | <S12>.codigo = <leer>.codigo // <S13>.codigo |  |
| <S12> → variable <S13> | <S12> → 503 <S13> | <S12>.codigo = Busca TS(variable)  SINO existe manda error(-105)  <S13>.codigo | -105 |
| <S12> → <valor><S13> | <S12> → <valor>  <S13> | <S12>.codigo = TD = <valor>.codigo  BuscaTS(variable)  TD1 = TDvar  Compara (TD1 == TDvar)  Si no son iguales manda error(-101)  <S13>.codigo | -101 |
| <S13> → ; | <S13> → 35 | <S13>.codigo = Retorna(;) |  |

## Valor

|  |  |  |
| --- | --- | --- |
| Gramática | | Rutina Semántica |
| <valor> → CADENA | <valor> → 501 | <valor>.codigo = Regresa valor Regresa TD=cad |
| <valor> → ENTERO | <valor> → 499 | <valor>.codigo = Regresa valor Regresa TD=ent |
| <valor> → DECIMAL | <valor> → 500 | <valor>.codigo = Regresa valor Regresa TD=dec |
| <valor> → CARÁCTER | <valor> → 502 | <valor>.codigo = Regresa valor Regresa TD=ch |
| <valor> → BOOLEANO(TRUE) | <valor> → 26 | <valor>.codigo = Regresa valor Regresa TD=bool |
| <valor> → BOOLEANO(FALSE) | <valor> → 26 | <valor>.codigo = Regresa valor Regresa TD=bool |

## SI

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina semántica | Error |
| <Si> → Si <S20> | <Si> → 9 <S20> | <Si>.codigo= <S20>.codigo |  |
| <S20>→ ( <condición> ) { <sentencia>} <S21> | <S20>→ 36 <condición> 37 38 <sentencia>39 <S21> | <S20>.codigo=<condición>.codigo<sentencia>.codigo <S21>.codigo |  |
| <S21>→ <SinoSi> | <S21>→ <SinoSi> | <S21>.codigo= <SinoSi>.codigo |  |
| <S21>→<Sino> | <S21>→<Sino> | <S21>.codigo=<Sino>.codigo |  |
| <Sino> → { <sentencia>} | <Sino> → 38 <sentencia>39 | <Sino>.codigo= <sentencia>.codigo |  |
| <Sino>→ λ | <Sino>→ λ | <Sino>.codigo=Retorna |  |
| <SiNoSi>→ Sino <Si> | <SiNoSi>→ 10 <Si> | <SiNoSi>.codigo=<Si>.codigo |  |

## Exp

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina semántica | Error |
| <exp> → <exp> OA <exp> | <exp> → <exp> OA <exp> | <exp>.codigo= <exp>.codigo CICLO[ si es de igual o menor prioridad sacar P2(OA anterior),insertar P1(OA anterior sacado de P2) ], si es de mayor prioridad insertar P2(OA nuevo) insertar P2(OA) si es de menor prioridad sacar P2(OA anterior),insertar(OA anterior en P1),insertarP2(OA nuevo),si llega parentesis abierto insertarlo en P2,si llega parentesis cerrado sacar P2(todos los OA) e irlos metiendo P1(todos los OA) hasta encontrar un parentesis abierto y Eliminar los parentesis e insertarlos en P1(<s31> <exp>.codigo |  |
| <exp> → ( <exp> ) | <exp> → 36 <exp> 37 | <exp>.codigo = <exp>.codigo |  |
| <exp> → variable | <exp> → 503 | <exp>.codigo = busca TS(variable) sino existe manda error 105 si existe insertar P1 | -105 |
| <exp> → <valor> | <exp> → <valor> | <S23>.codigo= <valor>.codigo |  |

## Switch

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina semántica | Error |
| <Switch>→ Switch (<S25> ){<caso>} | <Switch>→ 24 36 <S25> 37 38 <caso>39 | <Switch>.codigo= <S25>.codigo<caso>.codigo |  |
| <S25> → <valor> | <S25> → <valor> | <S25>.codigo= <valor>.codigo |  |
| <S25> → variable | <S25> → 503 | <S25>.codigo=Revisa TS (variable) si no existe mandar error 105 | -105 |

## Caso

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina semántica | Error |
| <Caso> → Caso( <S26>) : <sentencia> Fin <S27> | <Caso> → 20 36 <s26>37 43 <sentencia> 17 <s27> | <Caso>.codigo=<s26>.codigo <sentencia>.codigo <s27>.codigo |  |
| <S26> → <valor> | <S26> → <valor> | <s26>.codigo= <valor>.codigo |  |
| <S26> → variable | <S26> → 503 | <S26>.codigo=Buscar TS(variable) sino existe manda error 105 | -105 |
| <S27> → <Caso> | <s27> → <Caso> | <s27>.codigo=<Caso>.codigo |  |
| <S27>→λ | <s27> λ | <s27>.codigo=Retorna |  |

## Condición

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina semántica | Error |
| <Condicion> → <exp>OR<exp><S14> | <Condicion> → <exp>80<exp><S14> | <Condicion> .codigo= <exp>.codigo <exp>.codigo <S14>.codigo |  |
| <S14> → OL <Condicion> | <S14> → 03 <Condicion> | <S14>.codigo= <Condicion>.codigo |  |
| <S14> → λ | <S14> → λ | <S14>.codigo = retorna |  |

## Contador

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina semántica | Error |
| <Contador> → variable <s15> | <Contador> → 503 <s15> | <Contador>.codigo= Buscar TS(variable), si no se encuentra mandar erro, de lo contrario seguir <s15>.codigo | -105 |
| <s15> → + + | <s15> → + + | <s15>.codigo= retorna |  |
| <s15> → - - | <s15> → - - | <s15>.codigo= retorna |  |

## Mientras

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina semántica | Error |
| <Mientras> → Mientras (<condición>){<sentencias>} | <Mientras> → 15 36<condición>37 38<sentencias>39 | <Mientras> .codigo= <condición>.codigo <sentencias>.codigo |  |

## Para

|  |  |  |  |
| --- | --- | --- | --- |
| Gramática | | Rutina semántica | Error |
| <Para> → Para (<Inicialización>;<condición>;<contador>){<sentencias>} | <Para> → 8 36<Inicialización>35<condición>35<contador>37 38 <sentencias>39 | <Para>.codigo=<Inicialización>.codigo <condición>.codigo <contador>.codigo <sentencias>.codigo |  |
| <Inicializacion> → TD<S16> | <Inicializacion> → 1 <S16> | <Inicializacion>.codigo = guardar TD en una variable nueva Td1= TD <S16>.codigo |  |
| < Inicializacion > → variable<s18> | < Inicializacion > → 503 <s18> | < Inicializacion >.codigo = Buscar TS(variable ) si no existe enviar error si la encuentra seguir <s18>.codigo | -105 |
| <s16> → variable<s17> | <s16> → 503 <s17> | <s16>.codigo = Buscar en TS(variable) si existe lanzar error si no agregar TS(variable ) e insertar TS(Td1 ) en el mismo renglon <s17>.codigo | -100 |
| <s18> → =<s19> | <s18> → 42 <s19> | <s17>.codigo = <s19>.codigo |  |
| <s19> → <valor> | <s19> → <valor> | <s19>.codigo= <valor>.codigo asignar valor a TS en el mismo renglon de Td1 validando tipo de dato | -101 |
| <s19> → variable | <s19> → 503 | <s19>.codigo= Buscar en TS(variable) si no se encuentra lanzar error, si existe asignar variable a el valor en TS | -105 |