Tabla de errores

|  |  |
| --- | --- |
| Nombre | Número |
| Variable incorrecta | 100 |
| Mensaje/Cadena o carácter no valido/a | 101 |
| Número entero erróneo | 102 |
| Número doble erróneo | 103 |
| Símbolo desconocido | 104 |

|  |  |  |
| --- | --- | --- |
| Lexema | Nombre | Número |
| inicio | Inicio | 1 |
| ( | Paréntesis que abre | 2 |
| ) | Paréntesis que cierra | 3 |
| { | Llave que abre | 4 |
| } | Llave que cierra | 5 |
| int | Tipo de dato | 6 |
| double |
| string |
| caracter |
| ; | Punto y coma | 7 |
| = | Igual | 8 |
| read | Leer | 9 |
| put | Imprime | 10 |
| : | Dos puntos | 11 |
| while | Repetición | 12 |
| < | Operador relacional | 13 |
| > |
| <= |
| >= |
| == |
| ¡= |
| AND | Operador lógico | 14 |
| OR |
| + | Operador aritmético | 15 |
| - |
| \* |
| / |
| , | Coma | 16 |
| for | Para | 17 |
| do | Hacer | 18 |
| # | Inicio/fin de programa | 19 |
| call | Llamada a función | 20 |

Tipo de dato

|  |  |
| --- | --- |
| Nombre | Numero |
| Variable | 50 |
| Mensaje/Cadena | 51 |
| Número entero | 52 |
| Número doble | 53 |
| Caracter | 54 |

Separadores Naturales

|  |  |
| --- | --- |
| Separadores | Nombre |
| \b | Blanco |
| \n | Nueva línea |
| \r | Enter |
| \t | tabulación |

4.- REGLAS SINTÁCTICAS:

<inicio> 🡪inicio ( ) {<sentencias>}.

<sentencias>🡪<acciones> | <declaraciones> | λ

<acciones> 🡪 variable = <acciones’>;|<escritura>|<repetición>|<op mat >

<acciones’> 🡪<opción><op mat><sentencias> | read ( tipo de dato ) <sentencias>

<op mat> 🡪 opA <op mat’> | λ

<Tipo de Dato>🡪int| double| string

<escritura> 🡪put: (mensaje); <sentencias>

<declaraciones>🡪<tipo de dato> variable ; <sentencias>

<Repetición> 🡪while ( <condición> ) {<sentencias>} <sentencias>

<Condición> 🡪<opción >operador relacional <opción´>

<Con´> 🡪opLog <Condición>

<opción> 🡪 variable | numero entero | numero doble

<Op mat> 🡪<Opción><Op mat’>

<Op Mat’> 🡪<opción><Op mat>

T:{inicio, (,),{,},variable, ; , int, double, string, =, read, put, : , while, operador relacional, Op Arit, numero, letra, Op Log}

NT:{inicio, declaraciones, acciones, tipo de dato, asignación, lectura, escritura, repetición, condición, op mat, opción, con´}

**Gramática**

<s0>-----------------------------> <inicio>

1. <inicio>-----------------------------> 19 1 2 3 <sentencias> 5 <función> 19
2. <sentencias1>-----------------> <declaraciones>
3. <sentencias2>-----------------> <acciones>
4. <sentencias->-----------------> -
5. <acciones1>-------------------> 50 8 acciones'> 7 <sentencias>
6. <acciones2>-------------------> escritura>
7. <acciones3>-------------------> repeticion
8. <VACIO>-----------------------> VACIO
9. <acciones'1>------------------> <opción> <opmat> <sentencias>
10. <acciones'2>------------------> 9 2 6 3 <sentencias>
11. <OM>---------------------------> 15 <opmat'>
12. <OM->--------------------------> -
13. <escritura>--------------------> 10 11 2 51 3 7 <sentencias>
14. <declaracion>----------------> 6 50 7 <sentencias>
15. <repeticion>------------------> 12 2 <condición> 3 4 <sentencias> 5 <sentencias>
16. <condicion>------------------> <opción> 13 <opción> con'>
17. <con'->------------------------> -
18. <con'>-------------------------> 14 <condicion>
19. <opcion1>--------------------> 50
20. <opcion2>--------------------> 52
21. <opcion3>--------------------> 53
22. <OM'>------------------------> <opción> <opmat>
23. <funcion>-----------------------------> 50 2 <par> 3 4 <sentencias> 5 <fun’>
24. <fun’>-----------------------------> <funcion>
25. <fun’>-----------------------------> -
26. <par>----------------> 6 50 <par’>
27. <par’>-------------------- > 16 <par>
28. <par’>-------------------- > -
29. <repeticion2>------------------> 17 2 <for> 3 4 <sentencias> 5 <sentencias>
30. <for> --- > 6 50 8 <opcion> 7 <condición> 7 50 8 <opcion> <opmat>
31. <repeticion3>------------------> 18 4 sentencias 5 2 <condicion > 3 7 <sentencias>
32. <funcion>-----------------------------> -
33. <par>-----------------------------> -
34. <opcion4>--------------------> 51

**Gramática corregida**

<s0>-----------------------------> <inicio>

1. <inicio>-----------------------------> # inicio ( { <sentencias> } <función> #
2. <sentencias>-----------------> <declaraciones>
3. <sentencias>-----------------> <acciones>
4. <sentencias->-----------------> -
5. <acciones>-------------------> variable = <acciones'> ; <sentencias>
6. <acciones>-------------------> <escritura>
7. <acciones>-------------------> <repetición>
8. <VACIO>-----------------------> VACIO
9. <acciones'>------------------> <opción> <opmat> <sentencias>
10. <acciones'>------------------> read ( tipoDato ) <sentencias>
11. <OM>---------------------------> opMat <opmat'>
12. <OM->--------------------------> -
13. <escritura>--------------------> put : ( <salida> ) ; <sentencias>
14. <declaracion>----------------> tipoDato variable <declaración’> ; <sentencias>
15. <repeticion>------------------> while ( <condición> ) { <sentencias> } <sentencias>
16. <condicion>------------------> <opción> opRel <opción> con'>
17. <con'->------------------------> -
18. <con'>-------------------------> opLog <condicion>
19. <opcion>--------------------> Variable
20. <opcion>--------------------> Entero
21. <opcion>--------------------> Doble
22. <OM'>------------------------> <opción> <opmat>
23. <funcion>-----------------------------> Variable ( <par> ) { <sentencias> } <fun’>
24. <fun’>-----------------------------> <funcion>
25. <fun’>-----------------------------> -
26. <par>----------------> tipoDato Variable <par’>
27. <par’>-------------------- > Coma <par>
28. <par’>-------------------- > -
29. <repeticion2>------------------> for ( <for> ) { <sentencias> } <sentencias>
30. <for> --- > tipoDato variable = <opcion> ; <condición> ; variable = <opcion> <opmat>
31. <repeticion>------------------> do { sentencias } ( <condicion > ) ; <sentencias>
32. <funcion>-----------------------------> -
33. <par>-----------------------------> -
34. <opcion>--------------------> Cadena
35. **<declaraciones’>**----------------> -
36. **<declaraciones’>**----------------> **, variable <declaraciones’’>**
37. **<declaraciones’’>-------------------**→ **, variable <declaraciones’>**
38. **<declaraciones’’>**----------------> -
39. <sentencias>----------> <llamada>
40. <llamada>----------> call variable ( <par> ) ; < sentencias>
41. <salida>--------------------> cadena
42. <salida>--------------------> variable

<inicio>-----------------------------> 1 2 3 4 <sentencias> 5 <función>

<sentencias>-----------------> <declaraciones> | <acciones>

<acciones>-------------------> 50 8 <acciones'> 7 <sentencias> | <escritura> | <repetición>

<acciones'>------------------> <opcion <opmat sentencias| 9 2 6 3 sentencias>

<OM>---------------------------> 15 <opmat'>

<escritura>--------------------> 10 11 2 51 3 7 <sentencias>

<declaracion>----------------> 6 50 7 <sentencias>

<repeticion>------------------> 12 2 <condición> 3 4 <sentencias> 5 <sentencias>

<condicion>------------------> <opcion 13 <opcion> <con'>

<con'>-------------------------> 14 <condición>

<opcion>--------------------> 50 | 52 | 53

<OM'>------------------------> <opción> <opmat>

**Tabla de transiciones**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **50** | **51** | **52** | **53** |
| s0 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 |  |  |  |  |  |
| <inicio> | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <declaraciones> |  |  |  |  |  | 14 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <acciones> |  |  |  |  |  |  |  |  |  | 6 |  | 7 |  |  |  |  | 7 | 7 |  |  | 5 |  |  |  |
| <acciones’> |  |  |  |  |  |  |  |  | 10 |  |  |  |  |  |  |  |  |  |  |  | 9 | 9 | 9 | 9 |
| <sentencias> |  |  |  |  | 4 | 2 | 4 |  |  | 3 |  | 3 |  |  |  |  | 3 | 3 |  |  | 3 |  |  |  |
| <escritura> |  |  |  |  |  |  |  |  |  | 13 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <repetición> |  |  |  |  |  |  |  |  |  |  |  | 15 |  |  |  |  | 29 | 31 |  |  |  |  |  |  |
| <condición> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 16 |  | 16 | 16 |
| <con’> |  |  | 17 |  |  |  | 17 |  |  |  |  |  |  | 18 |  |  |  |  |  |  |  |  |  |  |
| <opción> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 19 | 34 | 20 | 21 |
| <op mat> |  |  | 12 |  |  | 12 | 12 |  |  | 12 |  | 12 |  |  | 11 |  |  |  |  |  | 12 |  |  |  |
| <op mat’> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 22 |  | 22 | 22 |
| <función> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 32 |  | 23 |  |  |  |
| <fun'> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 25 |  | 24 |  |  |  |
| <par> |  |  | 33 |  |  | 26 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <par'> |  |  | 28 |  |  |  |  |  |  |  |  |  |  |  |  | 27 |  |  |  |  |  |  |  |  |
| <for> |  |  |  |  |  | 30 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <declaraciones’> |  |  |  |  |  |  | 35 |  |  |  |  |  |  |  |  | 36 |  |  |  |  |  |  |  |  |
| <declaraciones2’> |  |  |  |  |  |  | 38 |  |  |  |  |  |  |  |  | 37 |  |  |  |  |  |  |  |  |
| <llamada> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 40 |  |  |  |  |
| <salida> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 42 | 41 |  |  |

|  |  |  |
| --- | --- | --- |
| No Terminal | Primeros | Siguientes |
| <S0> | # | $ |
| <inicio> | # | $ |
| <sentencias> | variable put while tipoDato for do call λ | } ; |
| <declaraciones> | tipoDato | } ; variable put while tipoDato for do λ |
| <acciones> | variable put while for do | } ; |
| <escritura> | put | } ; |
| <repetición> | while do for | } ; |
| <acciones’> | variable int double string read | ; |
| <opción> | variable int double string | ) ; opMat and opRel put while tipoDato |
| <OM> | opMat λ | ) ; variable put while tipoDato |
| <condición> | variable int double string | ) ; |
| <con’> | and or λ | ) ; |
| <op mat’> | variable int double string | ; variable put while tipoDato |
| <funcion> | variable λ | } ; # |
| <fun’> | variable λ | # |
| <par> | tipoDato λ | ) |
| <par’> | , λ | ) |
| <for> | tipoDato | ) ; |
| <declaraciones’> | , λ | ; |
| <declaraciones’2> | , λ | ; |
| <llamadas> | call | } ; |
| <salida> | variable cadena | ) |

|  |  |  |
| --- | --- | --- |
| **No Terminal** | **Primeros** | **Siguientes** |
| **<S0>** | **19** | **$** |
| **<inicio>** | **19** | **$** |
| **<sentencias>** | **50 10 12 6 17 18 λ** | **5 7** |
| **<declaraciones>** | **6** | **5 7 50 10 12 6 17 18 λ** |
| **<acciones>** | **50 10 12** | **5 7** |
| **<escritura>** | **10** | **5 7** |
| **<repetición>** | **12 17 18** | **5 7** |
| **<acciones’>** | **50 52 53 9** | **7** |
| **<opción>** | **50 52 53** | **3 7 15 14 13 3 10 12 6** |
| **<op mat>** | **15 λ** | **3 7 50 10 12 6** |
| **<condición>** | **50 52 53** | **3 7** |
| **<con’>** | **14 λ** | **3 7** |
| **<op mat’>** | **50 52 53** | **7 50 10 12 6** |
| **<funcion>** | **50 λ** | **5 7 19** |
| **<fun’>** | **50 λ** | **19** |
| **<par>** | **6 λ** | **3** |
| **<par’>** | **16 λ** | **3** |
| **<for>** | **6** | **3 7** |
| **<declaraciones’>** | **16 λ** | **7** |
| **<declaraciones’’>** | **16 λ** | **7** |
| **<llamada>** | **20** | **5 7** |
| **<salida>** | **50 51** | **3** |

X = read(int);

put : (“hola” | variable);

función sin número

Ejemplo programa:

#

inicio() {

int x;

string w;

x = 2 + 2;

string w = "hola mundo";

}

funcionA() {

y = read(int);

put:("HOLA");

}

funcionB(int x, double y) {

int w;

double z;

while(x > y) {

z = z + 1;

}

do {

y = y + 1;

} while(x > y);

for(int i = 1; z < x; i = i + 1) {

z = z + 1;

}

}

#

COSAS POR HACER:

* **llamadas a funciones**
* **declaración en misma línea**
* **condición llama a expresión no a sentencia**
* **car**
* condición en acciones
* separar condiciones
* expresiones en condiciones