**Lenguajes y Autómatas II.**

**Palabras Reservadas.**

public, private, class, String, Int, Boolean, float, double, void, false, true, if, while, do, else, for, Print.

**Operadores Aritméticos.**

Suma, Resta, Multiplicación, División, Incremento, Decremento, Modulo.

**Operadores Relacionales.**

Mayor que, Menor que, Menor o igual que, Mayor o igual que, igual a, diferente a.

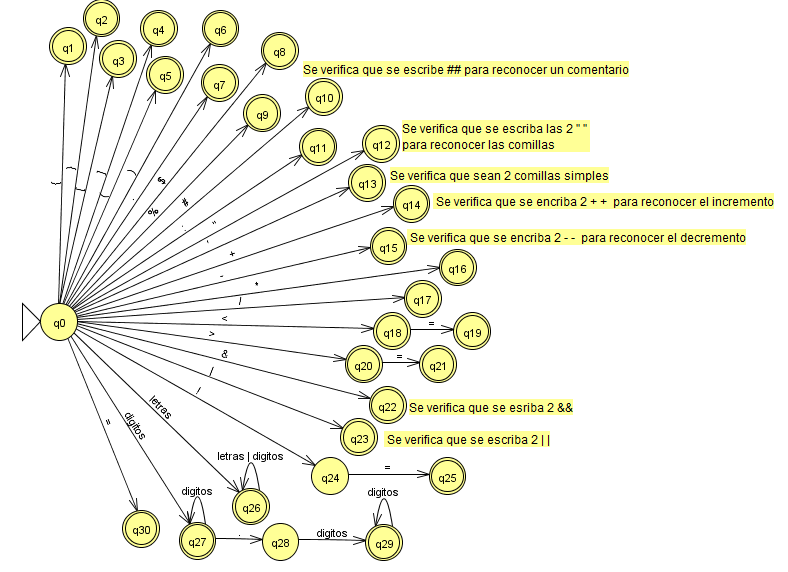
**Operadores Lógicos.**

AND, OR, NOT.

**Lista de Tokens.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Token | Lexema | Expresión Regular |
| 1 | Llave apertura | **{** | **{** |
| 2 | Llave cierre | **}** | **}** |
| 3 | Paréntesis apertura | **(** | **(** |
| 4 | Paréntesis cierre | **)** | **)** |
| 5 | Coma | **,** | **,** |
| 6 | Punto y coma | **;** | **:** |
| 7 | Comilla | **“** | **“** |
| 8 | Delimitador | **$** | **$** |
| 9 | Comentario | **##** | **##** |
| 10 | Suma | **+** | **+** |
| 11 | Resta | **-** | **-** |
| 12 | Multiplicación | **\*** | **\*** |
| 13 | División | **/** | **/** |
| 14 | Modulo | **%** | **%** |
| 15 | Igual | **=** | **=** |
| 16 | Incremento | **++** | **++** |
| 17 | Decremento | **--** | **--** |
| 18 | Menor igual | **<=** | **<=** |
| 19 | Mayor igual | **>=** | **>=** |
| 20 | Igual | **==** | **==** |
| 21 | AND | **&** | **&** |
| 22 | OR | **|** | **|** |
| 22 | NOT | **!=** | **¡=** |
| 23 | Letra | **a, b, c, d, e, f, g ,h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, y, z, A,B,C,D,E,F,G,H,I,J,K**  **,L,M,N,O,P,Q,R,S,T,U,V,W,Y,Z** | **[a-zA-Z]** |
| 25 | Digito | **0, 1, 2, 3, 4, 5, 6, 7, 8, 9** | **[0-9]\*** |
| 26 | Constante Entera | **0, 1, 2, 3, 4, 5, 6, 7, 8, 9** | **Digito Digito\*** |
| 27 | Identificador | **Val1, Val2** | **Letra (Letra|Digito)\*** |
| 28 | Espacio | **\t, \n, \r** | **\t | \n | \r** |
| 29 | class | class | class |
| 30 | String | String | String |
| 31 | Boolean | Boolean | Boolean |
| 32 | Int | Int | Int |
| 33 | float | float | float |
| 34 | double | double | double |
| 35 | void | void | void |
| 36 | public | public | public |
| 37 | private | private | private |
| 38 | false | false | false |
| 39 | true | true | true |
| 40 | if | if | if |
| 41 | do | do | do |
| 42 | while | while | while |
| 43 | for | for | for |
| 44 | Print | Print | Print |
| 45 | else | else | else |

**Autómata Finito Determinista (AFD).**

****

**Estados Finales del AFD**

Q1: Llave de apertura. -> {

Q2: Lave Cierre. -> }

Q3: Corchete apertura. -> [

Q4: Corchete Cierre. -> ]

Q5: Paréntesis apertura. -> (

Q6: Paréntesis cierre. -> )

Q7: Punto. -> .

Q8: $.

Q9: Modulo. -> %.

Q10: Comentario simple. -> # # , se verifica que se escriba 2 # #para reconocer un comentario.

Q11: Punto y coma. -> ;

Q12: Comillas. -> “ ” se verifica que se escriban 2 “ ” para reconocer las comillas.

Q13: Comillas simples - > se verifica que se escriba 2 ‘ ’ para ser reconocido.

Q14: Operador Suma. -> +, se verifica que se escriba 2 ++ para reconocer el incremento.

Q15: Operador Menos. -> -, se verifica que se escriban 2 – para reconocer el decremento.

Q16: Operador multiplicación. -> \*

Q17: Operador división. -> /

Q18: Menor que. -> <

Q20: Mayor que. -> >

Q19: Menor igual que. -> <=

Q21: Mayor igual que. -> >=

Q22: Operador And. -> Se verifica que se introduzca 2 && para reconocer.

Q23: Operador OR. -> Se verifica que se introduzca 2 || para reconocer.

Q27: Operador Not. -> ¡=

Q29: Se Verifica si pertenece a la lista de palabras reservadas. -> Int, String, public

Q30: Identificador. -> Variable1, Variable 2

Q32: Constante Entera. -> 12

Q34: Número real. -> 324.567

Q35: Igual. -> =

**Tabla de Transiciones.**

**Alfabeto.**

{A-Za-z0-9,<,>,=!,+,-,\*,/,,,%,.,$,#,{,},(,),\_,class, String, int, boolean, float, char, doublé, void, protected, private, public, false, true, if, do while, for, print}

* **GRAMATICA.**

**PROGRAMA** 🡪 *Inicio\_programa cuerpo\_codigo*

**INICIO\_PROGRAMA 🡪** *disponibilidad* **class** **identificador** **{**

**DISPONIBILIDAD** 🡪 **public | private**

**CUERPO\_CODIGO 🡪** *declaración| método }*

**FUNCIÓN** 🡪 cond\_*if | cond\_while | cod\_for*

**COND\_IF** 🡪 **if** **(***condición***) {***impresión***} else{** *impresión***}**

**COND\_WHLE 🡪 while** **(***condición*) {*impresión*}

**CONDICION 🡪** *condición\_lógica | condición\_AND | condición\_OR*

**CONDICION AND** 🡪 *condición\_lógica && condición\_lógica*

**CONDICION OR** 🡪 *condición\_lógica | | condición\_lógica*

**CONDICIÓN\_LÓGICA** 🡪 Exp > Exp | Exp < Exp | Exp >= Exp | Exp <= Exp | Exp == Exp | Exp!= Exp

**Exp** 🡪 Exp **+** Term | Exp **–** Term | Term

**Term** 🡪 Term **\*** Factor | Term **/** Factor | Factor

**Factor 🡪 digito | identificador | (Exp)**

**IMPRESIÓN**  🡪 **Print** ( “ *texto* ”)*$*

**TEXTO 🡪** *texto* **letras | espacio | números**

**METODO 🡪** *disponibilidad* *identificador () {función}*

**DECLARACION 🡪**  *declVE declaración | declVD declaración | declVB declaración | declVE | declVD | declVB*

**declVE 🡪 Int identificador $ | Int identificador = num $ | Int identificador =** *Exp$*

**declVD 🡪 float identificador $ | float identificador = num.num $**

**declVB 🡪 bool identificador $ | bool identificador =** *valorbool* **$**

**valorbool 🡪 true | false**

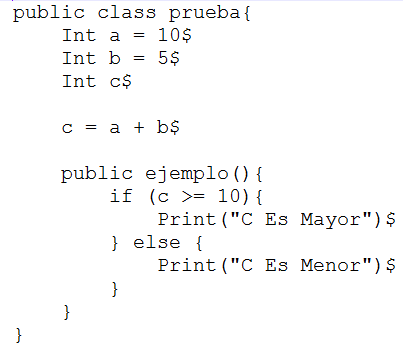


Figura 1 Ejemplo de Código.