Anexo

Los arboles generados por el parse de aquellas pruebas correctas, se encontraran al final, su lectura es de izquierda a derecha y arriba abajo(por ocupar menos espacio), de todas maneras se adjunta el parse y el código usado para VAST.

Prueba 1

Código

```
1. /* Prueba Correcta */
2. /*
3. Sentencias Simples
4. */
5. let number n;
6. let boolean b;
7. let string s;
8.
9. if(b) n=1;
10.  alert(s);
11.  input(n);
12.
13.  n-=2;
```

Tokens

```
<let, >
<number, >
<identificador, 0>
<puntoYcoma, >
<let, >
<boolean, >
<identificador, 1>
<puntoYcoma, >
<let, >
<string, >
<identificador, 2>
<puntoYcoma, >
<if, >
<abrirParentesis, >
<identificador, 3>
<cerrarParentesis, >
<identificador, 4>
<asignacion, >
<cteEntera, 1>
<puntoYcoma, >
<alert, >
<abrirParentesis, >
<identificador, 5>
<cerrarParentesis, >
<puntoYcoma, >
<input, >
<abrirParentesis, >
<identificador, 6>
<cerrarParentesis, >
<puntoYcoma, >
<identificador, 7>
<restaAsignacion, >
<cteEntera, 2>
```

```
<puntoYcoma, >
<EOF, >
```

Parse

Ascendente 27 24 28 24 29 24 40 39 36 33 31 43 39 36 33 31 17 23 40 39 36 33 31 19 25 20 25 43 39 36 33 31 22 25 4 2 2 2 2 2 2 2 1

Tabla de Símbolos

```
CONTENIDO DE LA TABLA TSMAIN #1 :

        * LEXEMA : 'n'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 0
-------------------------------------------------------------
        * LEXEMA : 'b'
        ATRIBUTOS:
        + Tipo : 'BOOLEAN'
        + Despl : 1
-------------------------------------------------------------
        * LEXEMA : 's'
        ATRIBUTOS:
        + Tipo : 'STRING'
        + Despl : 2
-------------------------------------------------------------
```

Errores de Análisis

- Ninguno

Prueba 2

Código

```
1. /* Prueba Correcta */
2.
3. let number a;
4. let number b;
5. let boolean c;
6. do{
7.     if(c) a-=1;
8. }while( (a+b) == 1 && (a-b) != 4 || c );
```

Tokens

```
<let, >
<number, >
<identificador, 0>
<puntoYcoma, >
<let, >
<number, >
<identificador, 1>
<puntoYcoma, >
<let, >
<boolean, >
<identificador, 2>
<puntoYcoma, >
<do, >
<abrirCorchete, >
<if, >
<abrirParentesis, >
```

```
<identificador, 3>
<cerrarParentesis, >
<identificador, 4>
<restaAsignacion, >
<cteEntera, 1>
<puntoYcoma, >
<cerrarCorchete, >
<while, >
<abrirParentesis, >
<abrirParentesis, >
<identificador, 5>
<opAritmetico, 1>
<identificador, 6>
<cerrarParentesis, >
<opRelacional, 1>
<cteEntera, 1>
<opLogico, 1>
<abrirParentesis, >
<identificador, 7>
<opAritmetico, 2>
<identificador, 8>
<cerrarParentesis, >
<opRelacional, 2>
<cteEntera, 4>
<opLogico, 2>
<identificador, 9>
<cerrarParentesis, >
<puntoYcoma, >
<EOF, >
```

Parse

Ascendente 27 24 27 24 28 24 40 39 36 33 31 43 39 36 33 31 22 23 16 15 40 39 40 37 36 33 31 41 39 36 43 39 34 33 40 39 40 38 36 33 31 41 39 36 43 39 35 32 31 40 39 36 33 30 26 4 2 2 2 2 1

Tabla de Símbolos

```
CONTENIDO DE LA TABLA TSMAIN #1 :

        * LEXEMA : 'a'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 0
----------------------------------------------------------------
        * LEXEMA : 'b'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 1
----------------------------------------------------------------
        * LEXEMA : 'c'
        ATRIBUTOS:
        + Tipo : 'BOOLEAN'
        + Despl : 2
----------------------------------------------------------------
```

Errores de Análisis

- Ninguno

Prueba 3

Código

```
1. /* Prueba Correcta */
2.
3. function print (string s, string msg, number f)
4. {
5.   alert (s); alert (msg); alert (f);
6.
7. }
8.
9.
10.  function number Factorial (number n)
11.  {
12.          if (n != 0)     return 1;
13.          return n + Factorial (n - 1);
14.  }
15.
16.  print("Factorial","de 6:" ,Factorial(6));
```

Tokens

```
<function, >
<identificador, 0>
<abrirParentesis, >
<string, >
<identificador, 1>
<coma, >
<string, >
<identificador, 2>
<coma, >
<number, >
<identificador, 3>
<cerrarParentesis, >
<abrirCorchete, >
<alert, >
<abrirParentesis, >
<identificador, 4>
<cerrarParentesis, >
<puntoYcoma, >
<alert, >
<abrirParentesis, >
<identificador, 5>
<cerrarParentesis, >
<puntoYcoma, >
<alert, >
<abrirParentesis, >
<identificador, 6>
<cerrarParentesis, >
<puntoYcoma, >
<cerrarCorchete, >
<function, >
<number, >
<identificador, 7>
<abrirParentesis, >
<number, >
<identificador, 8>
<cerrarParentesis, >
<abrirCorchete, >
<if, >
```

```
<abrirParentesis, >
<identificador, 9>
<opRelacional, 2>
<cteEntera, 0>
<cerrarParentesis, >
<return, >
<cteEntera, 1>
<puntoYcoma, >
<return, >
<identificador, 10>
<opAritmetico, 1>
<identificador, 11>
<abrirParentesis, >
<identificador, 12>
<opAritmetico, 2>
<cteEntera, 1>
<cerrarParentesis, >
<puntoYcoma, >
<cerrarCorchete, >
<identificador, 13>
<abrirParentesis, >
<cadena, "Factorial">
<coma, >
<cadena, "de 6:">
<coma, >
<identificador, 14>
<abrirParentesis, >
<cteEntera, 6>
<cerrarParentesis, >
<cerrarParentesis, >
<puntoYcoma, >
<EOF, >
```

Parse

Ascendente 10 6 29 29 27 14 13 13 11 7 40 39 36 33 31 19 25 40 39 36 33 31 19 25 40 39 36 33 31 19 25 16 15 15 15 8 5 27 9 6 27 14 11 7 40 39 36 43 39 35 33 31 43 39 36 33 31 49 21 23 40 39 40 39 43 38 36 33 31 48 45 42 37 36 33 31 49 21 25 16 15 15 8 5 44 39 36 33 31 44 39 36 33 31 43 39 36 33 31 48 45 42 39 36 33 31 48 47 47 45 18 25 4 2 3 3 1

Tabla de Símbolos

```
CONTENIDO DE LA TABLA TSMAIN #1 :

        * LEXEMA : 'print'
        ATRIBUTOS:
        + Tipo : 'FUNCTION'
        + numParam : 3
        + TipoRetorno : 'EMPTY'
        + EtiqFuncion : 'eti0'
        + TipoParam1 : 'STRING'
        + TipoParam2 : 'STRING'
        + TipoParam3 : 'NUMBER'
------------------------------------------------------------
        * LEXEMA : 'Factorial'
        ATRIBUTOS:
        + Tipo : 'FUNCTION'
        + numParam : 1
        + TipoRetorno : 'NUMBER'
        + EtiqFuncion : 'eti1'
        + TipoParam1 : 'NUMBER'
------------------------------------------------------------
```

```
CONTENIDO DE LA TABLA FUNCION print #2 :

        * LEXEMA : 'f'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 0
------------------------------------------------------------
        * LEXEMA : 'msg'
        ATRIBUTOS:
        + Tipo : 'STRING'
        + Despl : 1
------------------------------------------------------------
        * LEXEMA : 's'
        ATRIBUTOS:
        + Tipo : 'STRING'
        + Despl : 65
------------------------------------------------------------
CONTENIDO DE LA TABLA FUNCION Factorial #3 :

        * LEXEMA : 'n'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 0
------------------------------------------------------------
```

Errores de Análisis

- Ninguno

Prueba 4

Código

```
1. /* Programa Correcto - Ejemplo DRACO */
2.
3. let string texto;
4. function pideTexto ()
5. {
6.   alert ("Introduce un texto");
7.   input (texto);
8. }
9. function print (string msg)
10.  {
11.          alert (msg);
12.  }
13.  pideTexto();
14.  print (texto);
```

Tokens

```
<let, >
<string, >
<identificador, 0>
<puntoYcoma, >
<function, >
<identificador, 1>
<abrirParentesis, >
<cerrarParentesis, >
<abrirCorchete, >
<alert, >
<abrirParentesis, >
<cadena, "Introduce un texto">
```

```
<cerrarParentesis, >
<puntoYcoma, >
<input, >
<abrirParentesis, >
<identificador, 2>
<cerrarParentesis, >
<puntoYcoma, >
<cerrarCorchete, >
<function, >
<identificador, 3>
<abrirParentesis, >
<string, >
<identificador, 4>
<cerrarParentesis, >
<abrirCorchete, >
<alert, >
<abrirParentesis, >
<identificador, 5>
<cerrarParentesis, >
<puntoYcoma, >
<cerrarCorchete, >
<identificador, 6>
<abrirParentesis, >
<cerrarParentesis, >
<puntoYcoma, >
<identificador, 7>
<abrirParentesis, >
<identificador, 8>
<cerrarParentesis, >
<puntoYcoma, >
<EOF, >
```

Parse

Ascendente 29 24 10 6 12 7 44 39 36 33 31 19 25 20 25 16 15 15 8 5 10 6 29 14 11 7 40 39 36 33 31 19 25 16 15 8 5 46 18 25 40 39 36 33 31 48 45 18 25 4 2 2 3 3 2 1

Tabla de Símbolos

```
CONTENIDO DE LA TABLA TSMAIN #1 :

        * LEXEMA : 'texto'
        ATRIBUTOS:
        + Tipo : 'STRING'
        + Despl : 0
-------------------------------------------------------------
        * LEXEMA : 'pideTexto'
        ATRIBUTOS:
        + Tipo : 'FUNCTION'
        + numParam : 0
        + TipoRetorno : 'EMPTY'
        + EtiqFuncion : 'eti1'
-------------------------------------------------------------
        * LEXEMA : 'print'
        ATRIBUTOS:
        + Tipo : 'FUNCTION'
        + numParam : 1
        + TipoRetorno : 'EMPTY'
        + EtiqFuncion : 'eti2'
        + TipoParam1 : 'STRING'
-------------------------------------------------------------
CONTENIDO DE LA TABLA FUNCION pideTexto #2 :
```

```
CONTENIDO DE LA TABLA FUNCION print #3 :

        * LEXEMA : 'msg'
        ATRIBUTOS:
        + Tipo : 'STRING'
        + Despl : 0
----------------------------------------------------------------
```

Errores de Análisis

- Ninguno

Prueba 5 – Prueba Completa

Código

```
1. /* prueba correcta - Ejemplo DRACO*/
2.
3. let number entero;
4. let string cadena;
5. let boolean logico;
6. let number abcdefghij________;
7. function FuncionNumero ()
8. {
9.   let number i;
10.
11.    i = entero;
12.    entero=9-i;
13.  }
14.  function string FuncionRetorno (string hola, number x, boolean
   z)
15.  {
16.    let number i;
17.
18.    input (i);
19.    alert (hola);
20.
21.    return hola;
22.  }
23.  function FuncionSentencia (number z_Z, boolean b)
24.  {
25.    do
26.    {
27.          alert (88);
28.    } while (b);
29.    FuncionSentencia(z_Z,b);
30.  }
31.  function Funcion (number x, boolean b)
32.  {
33.         FuncionNumero();
34.    alert
35.      (cadena);return;
36.  }
37.  let number iii;
38.  let boolean bbb;
39.  /* comentario
40.  multi
41.  línea ***/
42.  input(iii);
43.  alert (iii) ;
44.
45.  iii -= iii;
```

```
46.  bbb = bbb;
47.
48.  Funcion
49.    ( iii,
50.      bbb );
```

Tokens

```
<let, >
<number, >
<identificador, 0>
<puntoYcoma, >
<let, >
<string, >
<identificador, 1>
<puntoYcoma, >
<let, >
<boolean, >
<identificador, 2>
<puntoYcoma, >
<let, >
<number, >
<identificador, 3>
<puntoYcoma, >
<function, >
<identificador, 4>
<abrirParentesis, >
<cerrarParentesis, >
<abrirCorchete, >
<let, >
<number, >
<identificador, 5>
<puntoYcoma, >
<identificador, 6>
<asignacion, >
<identificador, 7>
<puntoYcoma, >
<identificador, 8>
<asignacion, >
<cteEntera, 9>
<opAritmetico, 2>
<identificador, 9>
<puntoYcoma, >
<cerrarCorchete, >
<function, >
<string, >
<identificador, 10>
<abrirParentesis, >
<string, >
<identificador, 11>
<coma, >
<number, >
<identificador, 12>
<coma, >
<boolean, >
<identificador, 13>
<cerrarParentesis, >
<abrirCorchete, >
<let, >
<number, >
<identificador, 14>
```

<puntoYcoma, >
<input, >
<abrirParentesis, >
<identificador, 15>
<cerrarParentesis, >
<puntoYcoma, >
<alert, >
<abrirParentesis, >
<identificador, 16>
<cerrarParentesis, >
<puntoYcoma, >
<return, >
<identificador, 17>
<puntoYcoma, >
<cerrarCorchete, >
<function, >
<identificador, 18>
<abrirParentesis, >
<number, >
<identificador, 19>
<coma, >
<boolean, >
<identificador, 20>
<cerrarParentesis, >
<abrirCorchete, >
<do, >
<abrirCorchete, >
<alert, >
<abrirParentesis, >
<cteEntera, 88>
<cerrarParentesis, >
<puntoYcoma, >
<cerrarCorchete, >
<while, >
<abrirParentesis, >
<identificador, 21>
<cerrarParentesis, >
<puntoYcoma, >
<identificador, 22>
<abrirParentesis, >
<identificador, 23>
<coma, >
<identificador, 24>
<cerrarParentesis, >
<puntoYcoma, >
<cerrarCorchete, >
<function, >
<identificador, 25>
<abrirParentesis, >
<number, >
<identificador, 26>
<coma, >
<boolean, >
<identificador, 27>
<cerrarParentesis, >
<abrirCorchete, >
<identificador, 28>
<abrirParentesis, >
<cerrarParentesis, >
<puntoYcoma, >
<alert, >

```
<abrirParentesis, >
<identificador, 29>
<cerrarParentesis, >
<puntoYcoma, >
<return, >
<puntoYcoma, >
<cerrarCorchete, >
<let, >
<number, >
<identificador, 30>
<puntoYcoma, >
<let, >
<boolean, >
<identificador, 31>
<puntoYcoma, >
<input, >
<abrirParentesis, >
<identificador, 32>
<cerrarParentesis, >
<puntoYcoma, >
<alert, >
<abrirParentesis, >
<identificador, 33>
<cerrarParentesis, >
<puntoYcoma, >
<identificador, 34>
<restaAsignacion, >
<identificador, 35>
<puntoYcoma, >
<identificador, 36>
<asignacion, >
<identificador, 37>
<puntoYcoma, >
<identificador, 38>
<abrirParentesis, >
<identificador, 39>
<coma, >
<identificador, 40>
<cerrarParentesis, >
<puntoYcoma, >
<EOF, >
```

Parse

Ascendente 27 24 29 24 28 24 27 24 10 6 12 7 27 24 40 39 36 33 31 17 25 43 39 40 38 36 33 31 17 25 16 15 15 15 8 5 29 9 6 29 27 28 14 13 13 11 7 27 24 20 25 40 39 36 33 31 19 25 40 39 36 33 31 49 21 25 16 15 15 15 15 8 5 10 6 27 28 14 13 11 7 43 39 36 33 31 19 25 16 15 40 39 36 33 31 26 40 39 36 33 31 40 39 36 33 31 48 47 45 18 25 16 15 15 8 5 10 6 27 28 14 13 11 7 46 18 25 40 39 36 33 31 19 25 50 21 25 16 15 15 15 8 5 27 24 28 24 20 25 40 39 36 33 31 19 25 40 39 36 33 31 22 25 40 39 36 33 31 17 25 40 39 36 33 31 40 39 36 33 31 48 47 45 18 25 4 2 2 2 2 2 2 2 3 3 3 3 2 2 2 2 1

Tabla de Símbolos

```
CONTENIDO DE LA TABLA TSMAIN #1 :

        * LEXEMA : 'entero'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 0
------------------------------------------------------------
        * LEXEMA : 'cadena'
```

```
        ATRIBUTOS:
        + Tipo : 'STRING'
        + Despl : 1
-------------------------------------------------------------
        * LEXEMA : 'logico'
        ATRIBUTOS:
        + Tipo : 'BOOLEAN'
        + Despl : 65
-------------------------------------------------------------
        * LEXEMA : 'abcdefghij________'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 66
-------------------------------------------------------------
        * LEXEMA : 'FuncionNumero'
        ATRIBUTOS:
        + Tipo : 'FUNCTION'
        + numParam : 0
        + TipoRetorno : 'EMPTY'
        + EtiqFuncion : 'eti4'
-------------------------------------------------------------
        * LEXEMA : 'FuncionRetorno'
        ATRIBUTOS:
        + Tipo : 'FUNCTION'
        + numParam : 3
        + TipoRetorno : 'STRING'
        + EtiqFuncion : 'eti5'
        + TipoParam1 : 'STRING'
        + TipoParam2 : 'NUMBER'
        + TipoParam3 : 'BOOLEAN'
-------------------------------------------------------------
        * LEXEMA : 'FuncionSentencia'
        ATRIBUTOS:
        + Tipo : 'FUNCTION'
        + numParam : 2
        + TipoRetorno : 'EMPTY'
        + EtiqFuncion : 'eti6'
        + TipoParam1 : 'NUMBER'
        + TipoParam2 : 'BOOLEAN'
-------------------------------------------------------------
        * LEXEMA : 'Funcion'
        ATRIBUTOS:
        + Tipo : 'FUNCTION'
        + numParam : 2
        + TipoRetorno : 'EMPTY'
        + EtiqFuncion : 'eti7'
        + TipoParam1 : 'NUMBER'
        + TipoParam2 : 'BOOLEAN'
-------------------------------------------------------------
        * LEXEMA : 'iii'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 67
-------------------------------------------------------------
        * LEXEMA : 'bbb'
        ATRIBUTOS:
        + Tipo : 'BOOLEAN'
        + Despl : 68
-------------------------------------------------------------
CONTENIDO DE LA TABLA FUNCION FuncionNumero #2 :
```

```
        * LEXEMA : 'i'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 0
----------------------------------------------------------------
CONTENIDO DE LA TABLA FUNCION FuncionRetorno #3 :

        * LEXEMA : 'z'
        ATRIBUTOS:
        + Tipo : 'BOOLEAN'
        + Despl : 0
----------------------------------------------------------------
        * LEXEMA : 'x'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 1
----------------------------------------------------------------
        * LEXEMA : 'hola'
        ATRIBUTOS:
        + Tipo : 'STRING'
        + Despl : 2
----------------------------------------------------------------
        * LEXEMA : 'i'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 66
----------------------------------------------------------------
CONTENIDO DE LA TABLA FUNCION FuncionSentencia #4 :

        * LEXEMA : 'b'
        ATRIBUTOS:
        + Tipo : 'BOOLEAN'
        + Despl : 0
----------------------------------------------------------------
        * LEXEMA : 'z_Z'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 1
----------------------------------------------------------------
CONTENIDO DE LA TABLA FUNCION Funcion #5 :

        * LEXEMA : 'b'
        ATRIBUTOS:
        + Tipo : 'BOOLEAN'
        + Despl : 0
----------------------------------------------------------------
        * LEXEMA : 'x'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 1
----------------------------------------------------------------
```

Errores de Análisis

- Ninguno

Prueba 6

El error esta en el operado *, nosotros no lo tenemos implementado

Código

```
1. /* Prueba Incorrecta - Error Lexico*/
2. function number SumaAlCuadrado (number a, number b)
3. {
4.    j= a + b;
5.    return j * j;
6.
7. }
```

Tokens

```
1. <function, >
2. <number, >
3. <identificador, 0>
4. <abrirParentesis, >
5. <number, >
6. <identificador, 1>
7. <coma, >
8. <number, >
9. <identificador, 2>
10.  <cerrarParentesis, >
11.  <abrirCorchete, >
12.  <identificador, 3>
13.  <asignacion, >
14.  <identificador, 4>
15.  <opAritmetico, 1>
16.  <identificador, 5>
17.  <puntoYcoma, >
18.  <return, >
19.  <identificador, 6>
20.  <identificador, 7>
```

Parse

Ascendente 27 9 6 27 27 14 13 11 7 40 39 40 37 36 33 31 17 25

Tabla Símbolos

```
CONTENIDO DE LA TABLA TSMAIN #1 :

        * LEXEMA : 'SumaAlCuadrado'
        ATRIBUTOS:
        + Tipo : 'FUNCTION'
        + numParam : 2
        + TipoRetorno : 'NUMBER'
        + EtiqFuncion : 'eti0'
        + TipoParam1 : 'NUMBER'
        + TipoParam2 : 'NUMBER'
------------------------------------------------------------
        * LEXEMA : 'j'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 0
------------------------------------------------------------
CONTENIDO DE LA TABLA FUNCION SumaAlCuadrado #2 :

        * LEXEMA : 'b'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
```

```
        + Despl : 0
------------------------------------------------------------
        * LEXEMA : 'a'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 1
------------------------------------------------------------
```

Errores de Análisis

Error Lexico:

Caracter No valido en la linea:5

Error Sintactico en la linea:5

@Usuario: Se esperaba: (,||,&&,==,!=,+,- o nada

Prueba 7

El error es la sentencia de declaración con asignación, no la hemos incluido

Código

```
1. /*Prueba incorrecta - Error Sintaxis*/
2. let number var4 = 1+1;
```

Tokens

```
<let, >
<number, >
<identificador, 0>
<asignacion, >
```

Parse

Ascendente 27

Tabla Símbolos

```
CONTENIDO DE LA TABLA TSMAIN #1 :

        * LEXEMA : 'var4'
```

Errores de Análisis

Error Sintactico en la linea:2

@Usuario: Se esperaba: ;

Prueba 8

El error es un “return” fuera del cuerpo de función

Código

```
1. /* Prueba Incorrecta - Error Semantico*/
2.
3. let number b;
4. return b;
```

Tokens

```
<let, >
<number, >
<identificador, 0>
<puntoYcoma, >
<return, >
<identificador, 1>
<puntoYcoma, >
<EOF, >
```

Parse

Ascendente 27 24 40 39 36 33 31 49 21 25 4 2 2 1

Tabla Símbolos

```
CONTENIDO DE LA TABLA TSMAIN #1 :

        * LEXEMA : 'b'
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 0
-----------------------------------------------------------------
```

Errores de Análisis

Error Semantico:

Return Fuera de funcion

Error Semantico:

Revisa los errores que tienes, algo no anda bien

Prueba 9

El error es una Re-Declaracion de variable

Código

```
/* Prueba Incorrecta */
let number b;
let boolean b;
```

Tokens

```
<let, >
<number, >
<identificador, 0>
<puntoYcoma, >
<let, >
<boolean, >
<identificador, 1>
<puntoYcoma, >
<EOF, >
```

Parse

Ascendente 27 24 28 24 4 2 2 1

Tabla Símbolos

```
CONTENIDO DE LA TABLA TSMAIN #1 :

        * LEXEMA : 'b'
```

```
        ATRIBUTOS:
        + Tipo : 'NUMBER'
        + Despl : 0
-------------------------------------------------------------
```

Errores de Análisis

Error Semantico:

Nombre de identificador: "b" ,ya declarado previamente

Error Semantico:

Revisa los errores que tienes, algo no anda bien

Prueba 10

Error semántico en alert e input

Código

```
1. let boolean b;
2. alert(b);
3. input(b);
```

Tokens

```
<let, >
<boolean, >
<identificador, 0>
<puntoYcoma, >
<alert, >
<abrirParentesis, >
<identificador, 1>
<cerrarParentesis, >
<puntoYcoma, >
<input, >
<abrirParentesis, >
<identificador, 2>
<cerrarParentesis, >
<puntoYcoma, >
<EOF, >
```

Parse

Ascendente 28 24 40 39 36 33 31 19 25 20 25 4 2 2 2 1

Tabla Símbolos

```
CONTENIDO DE LA TABLA TSMAIN #1 :

        * LEXEMA : 'b'
        ATRIBUTOS:
        + Tipo : 'BOOLEAN'
        + Despl : 0
-------------------------------------------------------------
```

Errores de Análisis

Error Semantico:

Alert tiene que tener como argumento una cadena o numero@Usuario:

(Contruccion Correcta) >> alert(<cadena|numero>);

(Su construccion) >> alert(<'BOOLEAN'>);

Error Semantico:

input tiene que tener como argumento una cadena o un entero

@Usuario:

(Construccion Correcta) >> input(<cadena|numero>);

(Su construccion) >> input(<'BOOLEAN'>);

Error Semantico:

Revisa los errores que tienes, algo no anda bien

Árbol resultado de:

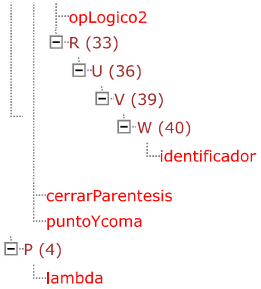
Gramática: D:\OneDrive - Universidad Politécnica de Madrid\Universidad\3º\Procesadores de Lenguajes\Practica\ProcesadoresLenguajes\Documentacion\2.Analizador Sintactico\GramaticaVast.txt

Parse: D:\OneDrive - Universidad Politécnica de Madrid\Universidad\3º\Procesadores de Lenguajes\Practica\ProcesadoresLenguajes\Documentacion\Anexo\allTests\Prueba 01\Parse0.txt

Árbol resultado de:

Gramática: D:\OneDrive - Universidad Politécnica de Madrid\Universidad\3º\Procesadores de Lenguajes\Practica\ProcesadoresLenguajes\Documentacion\2.Analizador Sintactico\GramaticaVast.txt

Parse: D:\OneDrive - Universidad Politécnica de Madrid\Universidad\3º\Procesadores de Lenguajes\Practica\ProcesadoresLenguajes\Documentacion\Anexo\allTests\Prueba 02\Parse1.txt


opLogico2
R (33)
U (36)
V (39)
W (40)
identificador
cerrarParentesis
puntoYcoma
P (4)
lambda

Árbol resultado de:

Gramática: D:\OneDrive - Universidad Politécnica de Madrid\Universidad\3º\Procesadores de Lenguajes\Practica\ProcesadoresLenguajes\Documentacion\2.Analizador Sintactico\GramaticaVast.txt

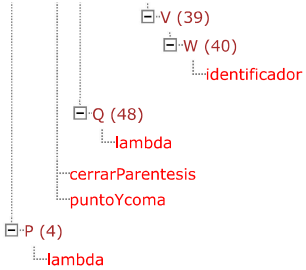
Parse: D:\OneDrive - Universidad Politécnica de Madrid\Universidad\3º\Procesadores de Lenguajes\Practica\ProcesadoresLenguajes\Documentacion\Anexo\allTests\Prueba 03\Parse2.txt


V (39)
W (40)
identificador
opRelacional2
V (39)
W (43)
cteEntera
cerrarParentesis
S (21)
return
X (49)
E (31)
R (33)
U (36)
V (39)
W (43)
cteEntera
puntoYcoma
C (15)
B (25)
S (21)
return
X (49)
E (31)
R (33)
U (36)
V (37)
V (39)
W (40)
identificador
opAritmetico1
W (42)
identificador
abrirParentesis
L (45)
E (31)
R (33)
U (36)
V (38)
V (39)
W (40)
identificador
opAritmetico2
W (43)
cteEntera
Q (48)
lambda
cerrarParentesis
puntoYcoma
C (16)
lambda
cerrarCorchete
P (2)
B (25)
S (18)
identificador
abrirParentesis
L (45)
E (31)
R (33)
U (36)
V (39)
W (44)
cadena
Q (47)
coma
E (31)
R (33)
U (36)
V (39)
W (44)
cadena
Q (47)
coma
E (31)
R (33)
U (36)
V (39)
W (42)
identificador
abrirParentesis
L (45)
E (31)
R (33)
U (36)
V (39)
W (43)
cteEntera
Q (48)
lambda
cerrarParentesis
Q (48)
lambda
cerrarParentesis
puntoYcoma
P (4)
lambda

Árbol resultado de:

Gramática: D:\OneDrive - Universidad Politécnica de Madrid\Universidad\3º\Procesadores de Lenguajes\Practica\ProcesadoresLenguajes\Documentacion\2.Analizador Sintactico\GramaticaVast.txt

Parse: D:\OneDrive - Universidad Politécnica de Madrid\Universidad\3º\Procesadores de Lenguajes\Practica\ProcesadoresLenguajes\Documentacion\Anexo\allTests\Prueba 04\Parse3.txt

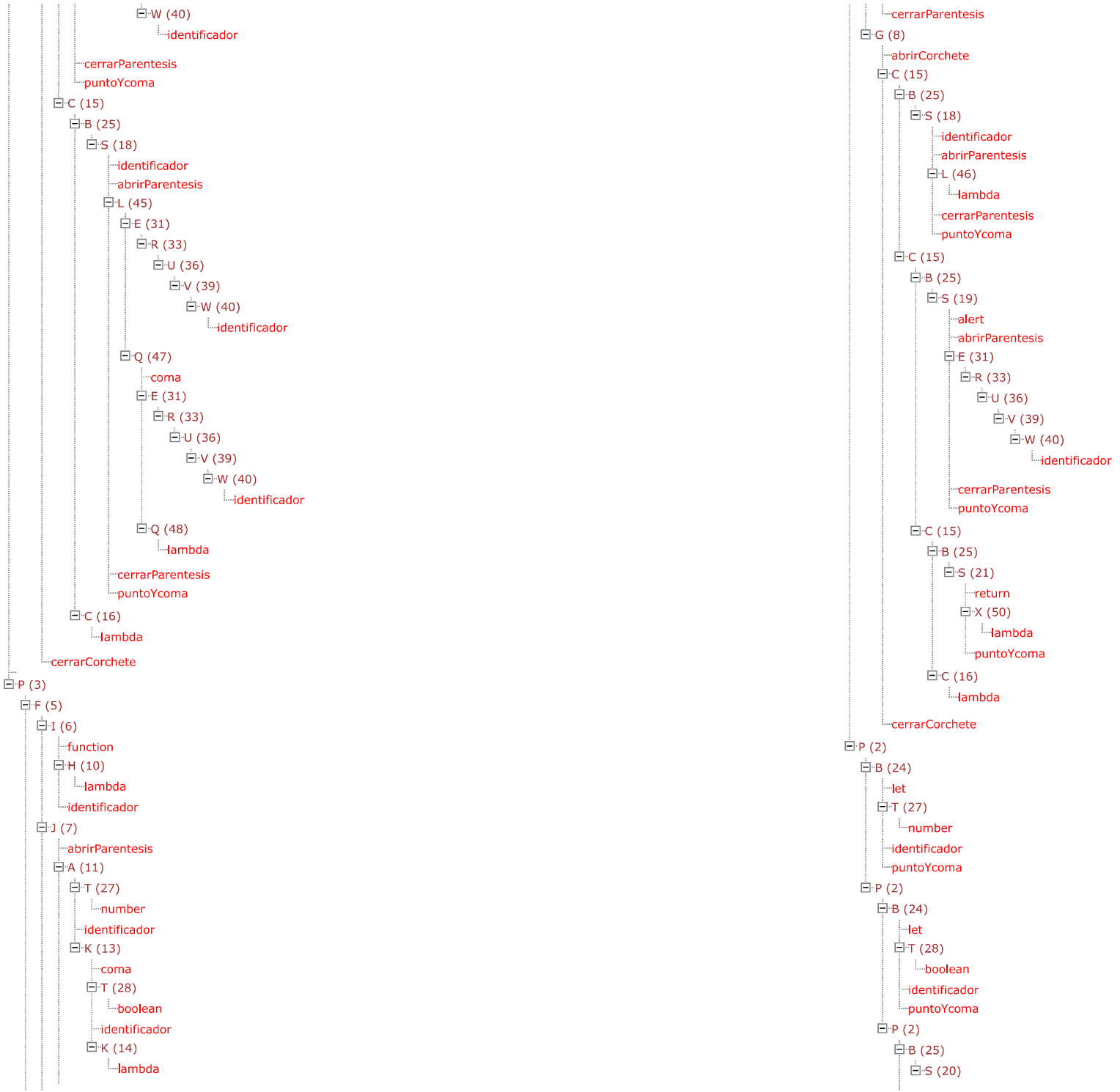

V (39)
W (40)
identificador
Q (48)
lambda
cerrarParentesis
puntoYcoma
P (4)
lambda

Árbol resultado de:

Gramática: D:\OneDrive - Universidad Politécnica de Madrid\Universidad\3º\Procesadores de Lenguajes\Practica\ProcesadoresLenguajes\Documentacion\2.Analizador Sintactico\GramaticaVast.txt

Parse: D:\OneDrive - Universidad Politécnica de Madrid\Universidad\3º\Procesadores de Lenguajes\Practica\ProcesadoresLenguajes\Documentacion\Anexo\allTests\Prueba 05\Parse4.txt


number
identificador
K (13)
coma
T (28)
boolean
identificador
K (14)
lambda
cerrarParentesis
G (8)
abrirCorchete
C (15)
B (24)
let
T (27)
number
identificador
puntoYcoma
C (15)
B (25)
S (20)
input
abrirParentesis
identificador
cerrarParentesis
puntoYcoma
C (15)
B (25)
S (19)
alert
abrirParentesis
E (31)
R (33)
U (36)
V (39)
W (40)
identificador
cerrarParentesis
puntoYcoma
C (15)
B (25)
S (21)
return
X (49)
E (31)
R (33)
U (36)
V (39)
W (40)
identificador
puntoYcoma
C (16)
lambda
cerrarCorchete
P (3)
F (5)
I (6)
function
H (10)
lambda
identificador
J (7)
abrirParentesis
A (11)
T (27)
number
identificador
K (13)
coma
T (28)
boolean
identificador
K (14)
lambda
cerrarParentesis
G (8)
abrirCorchete
C (15)
B (26)
do
abrirCorchete
C (15)
B (25)
S (19)
alert
abrirParentesis
E (31)
R (33)
U (36)
V (39)
W (43)
cteEntera
cerrarParentesis
puntoYcoma
C (16)
lambda
cerrarCorchete
while
abrirParentesis
E (31)
R (33)
U (36)
V (39)


W (40)
identificador
cerrarParentesis
puntoYcoma
C (15)
B (25)
S (18)
identificador
abrirParentesis
L (45)
E (31)
R (33)
U (36)
V (39)
W (40)
identificador
Q (47)
coma
E (31)
R (33)
U (36)
V (39)
W (40)
identificador
Q (48)
lambda
cerrarParentesis
puntoYcoma
C (16)
lambda
cerrarCorchete
P (3)
F (5)
I (6)
function
H (10)
lambda
identificador
J (7)
abrirParentesis
A (11)
T (27)
number
identificador
K (13)
coma
T (28)
boolean
identificador
K (14)
lambda
cerrarParentesis
G (8)
abrirCorchete
C (15)
B (25)
S (18)
identificador
abrirParentesis
L (46)
lambda
cerrarParentesis
puntoYcoma
C (15)
B (25)
S (19)
alert
abrirParentesis
E (31)
R (33)
U (36)
V (39)
W (40)
identificador
cerrarParentesis
puntoYcoma
C (15)
B (25)
S (21)
return
X (50)
lambda
puntoYcoma
C (16)
lambda
cerrarCorchete
P (2)
B (24)
let
T (27)
number
identificador
puntoYcoma
P (2)
B (24)
let
T (28)
boolean
identificador
puntoYcoma
P (2)
B (25)
S (20)


input
abrirParentesis
identificador
cerrarParentesis
puntoYcoma
P (2)
B (25)
S (19)
alert
abrirParentesis
E (31)
R (33)
U (36)
V (39)
W (40)
identificador
cerrarParentesis
puntoYcoma
P (2)
B (25)
S (22)
identificador
restaAsignacion
E (31)
R (33)
U (36)
V (39)
W (40)
identificador
puntoYcoma
P (2)
B (25)
S (17)
identificador
asignacion
E (31)
R (33)
U (36)
V (39)
W (40)
identificador
puntoYcoma
P (2)
B (25)
S (18)
identificador
abrirParentesis
L (45)
E (31)
R (33)
U (36)
V (39)
W (40)
identificador
Q (47)
coma
E (31)
R (33)
U (36)
V (39)
W (40)
identificador
Q (48)
lambda
cerrarParentesis
puntoYcoma
P (4)
lambda