**ESTRUCTURAS DE CÓDIGO DEL COMPILADOR**

Las estructuras de código son generadas a través de los componentes léxicos, dónde son concatenaciones de los componentes léxicos que siguen un patrón dicha estructura, por ende las estructuras de este compilador también son producciones y las que se pueden generar son las que están en la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| **Producción Declaraciones de puerto (DP)** | | |
| PUERTO TIPO\_PUERTO IDENTIFICADOR FIN\_DE\_LINEA | port\_A1 proximity proximitySensor! | |
| **Producción Declaraciones con valor (DCV)** | | |
| DECLARACION TIPO\_DATO IDENTIFICADOR OPERADOR\_ASIGNACION **VALORES** FIN\_DE\_LINEA | var string holaMundo = 'HolaMundo'! | |
| **Producción Declaraciones sin valor (DSV)** | | |
| DECLARACION TIPO\_DATO IDENTIFICADOR FIN\_DE\_LINEA | const string number3 ! | |
| **Producción Asignaciones (A)** | | |
| IDENTIFICADOR OPERADOR\_ASIGNACION **VALORES** FIN\_DE\_LINEA | number2 = 3.3 ! | |
| **Producción funciones de los motores con valor (FMCV)** | | |
| MOTOR SIGNO\_AGRUPACION\_3 IDENTIFICADOR SIGNO\_PUNTUACION\_1 **VALORES** SIGNO\_PUNTUACION\_1 IDENTIFICADOR SIGNO\_AGRUPACION\_4 FIN\_DE\_LINEA | move(motor1,180,boton1)! | |
| **Producción funciones de los motores sin valor (FMSV)** | | |
| MOTOR SIGNO\_AGRUPACION\_3 IDENTIFICADOR SIGNO\_PUNTUACION\_1 IDENTIFICADOR SIGNO\_AGRUPACION\_4 FIN\_DE\_LINEA | start(motor1,boton1)! | |
| **Producción Método de impresora a consola (MIC)** | | |
| PALABRA\_RESERVADA\_6 SIGNO\_AGRUPACION\_3 PALABRA\_RESERVADA\_7 SIGNO\_PUNTUACION\_1 (**VALORES** | IDENTIFICADOR) SIGNO\_AGRUPACION\_4 FIN\_DE\_LINEA | print(console,'El motor1 se esta moviendo')! | |
| **Producción Método de impresora a LCD (MIL)** | | |
| PALABRA\_RESERVADA\_6 SIGNO\_AGRUPACION\_3 IDENTIFICADOR SIGNO\_PUNTUACION\_1 (**VALORES** | IDENTIFICADOR) SIGNO\_AGRUPACION\_4 FIN\_DE\_LINEA | print(Hola,'El motor1 se esta moviendo')! | |
| **Producción Operaciones (O)** | | |
| PALABRA\_RESERVADA\_3 IDENTIFICADOR OPERADOR\_ASIGNACION (IDENTIFICADOR | **VALORES**) OPERADOR\_ARITMETICO (IDENTIFICADOR | **VALORES**) FIN\_DE\_LINEA | operation result = a + b! | |
| **Producción funciones sin parámetros (FSP)** | | |
| ESTRUCTURA\_DE\_CONTROL\_1 IDENTIFICADOR SIGNO\_AGRUPACION\_3 SIGNO\_AGRUPACION\_4 SIGNO\_AGRUPACION\_1 (**FMCV | FMSV** | **MIC** | **MIL**)\* SIGNO\_AGRUPACION\_2 | function moverMotor(){  move(motor1,180,boton1)!  print(console,'El motor1 se esta moviendo')!  } | |
| **Producción funciones con parámetros (FCP)** | | |
| ESTRUCTURA\_DE\_CONTROL\_1 TIPO\_DATO IDENTIFICADOR SIGNO\_AGRUPACION\_3 TIPO\_DATO IDENTIFICADOR (SIGNO\_PUNTUACION\_1 TIPO\_DATO IDENTIFICADOR)\* SIGNO\_AGRUPACION\_4 SIGNO\_AGRUPACION\_1 (**O**)\* PALABRA\_RESERVADA\_5 IDENTIFICADOR FIN\_DE\_LINEA SIGNO\_AGRUPACION\_2 | function int sumaNumerosInt(int a, int b){  operation result = a + b!  return result!  } | |
| **Producción llamadas de funciones con valor (LFCV)** | | |
| PALABRA\_RESERVADA\_4 IDENTIFICADOR SIGNO\_AGRUPACION\_3 **VALORES** (SIGNO\_PUNTUACION\_1 **VALORES**)\* SIGNO\_AGRUPACION\_4 FIN\_DE\_LINEA | call moverMotor(5)! | |
| **Producción llamadas de funciones sin valor (LFSV)** | | |
| PALABRA\_RESERVADA\_4 IDENTIFICADOR SIGNO\_AGRUPACION\_3 SIGNO\_AGRUPACION\_4 FIN\_DE\_LINEA | call moverMotor()! | |
| **Producción Begin (B)** | | |
| ESTRUCTURA\_DE\_CONTROL\_2 SIGNO\_AGRUPACION\_1 (**LFCV** | **LFSV** | **FMCV** | **FMSV**)\* SIGNO\_AGRUPACION\_2 | begin{  call moverMotor()!  call detenerMotor()!  call escribirLCD(5)!  } | |
| **Producción Método de los sensores (MS)** | | |
| PALABRA\_RESERVADA\_4 IDENTIFICADOR OPERADOR\_ASIGNACION IDENTIFICADOR SIGNO\_PUNTUACION\_2 METODO\_SENSOR SIGNO\_AGRUPACION\_3 SIGNO\_AGRUPACION\_4 FIN\_DE\_LINEA | call Distancia = puerto1.distance()! | |
| **Producción Método Delay (MD)** | | |
| PALABRA\_RESERVADA\_2 SIGNO\_AGRUPACION\_3 (IDENTIFICADOR | **VALORES**) SIGNO\_AGRUPACION\_4 FIN\_DE\_LINEA | | delay(1000)! |
| **Producción Método Encender (ME)** | | |
| PALABRA\_RESERVADA\_1 SIGNO\_AGRUPACION\_3 IDENTIFICADOR SIGNO\_PUNTUACION\_1 IDENTIFICADOR SIGNO\_AGRUPACION\_4 FIN\_DE\_LINEA | | ligther(led1,boton1)! |
| **Producción Método Apagar (MA)** | | |
| PALABRA\_RESERVADA\_8 SIGNO\_AGRUPACION\_3 IDENTIFICADOR SIGNO\_PUNTUACION\_1 IDENTIFICADOR SIGNO\_AGRUPACION\_4 FIN\_DE\_LINEA | | turn\_off(led1,boton1)! |
| **Producción If Relacional (IR)** | | |
| ESTRUCTURA\_DE\_CONTROL\_4 SIGNO\_AGRUPACION\_3 (IDENTIFICADOR | **VALORES**) OPERADOR\_RELACIONAL (IDENTIFICADOR | **VALORES**) SIGNO\_AGRUPACION\_4 SIGNO\_AGRUPACION\_1 (**MS** | **MIC** | **MIL** | **A** | **DCV** | **DSV** | **FMCV** | **FMSV** | **LFCV** | **LFSV** | **MD** | **ME** | **MA**)\* SIGNO\_AGRUPACION\_2 | if(number1>number2){  call moverMotor()!  } | |
| **Producción If Lógico con dos valores (IL2V)** | | |
| ESTRUCTURA\_DE\_CONTROL\_4 SIGNO\_AGRUPACION\_3 (IDENTIFICADOR | **VALORES**) OPERADOR\_LOGICO (IDENTIFICADOR | **VALORES**) SIGNO\_AGRUPACION\_4 SIGNO\_AGRUPACION\_1 (**MS** | **MIC** | **MIL** | **A** | **DCV** | **DSV** | **FMCV** | **FMSV** | **LFCV** | **LFSV** | **MD** | **ME** | **MA**)\* SIGNO\_AGRUPACION\_2 | if(number1 and number2){  call moverMotor()!  } | |
| **Producción If Lógico con un valor (IL1V)** | | |
| ESTRUCTURA\_DE\_CONTROL\_4 SIGNO\_AGRUPACION\_3 (IDENTIFICADOR | **VALORES**) SIGNO\_AGRUPACION\_4 SIGNO\_AGRUPACION\_1 (**MS** | **MIC** | **MIL** | **A** | **DCV** | **DSV** | **FMCV** | **FMSV** | **LFCV** | **LFSV** | **MD** | **ME** | **MA**)\* SIGNO\_AGRUPACION\_2 | if(number1){  call moverMotor()!  } | |
| **Producción Else Relacional (ER)** | | |
| ESTRUCTURA\_DE\_CONTROL\_4 SIGNO\_AGRUPACION\_3 (IDENTIFICADOR | **VALORES**) OPERADOR\_RELACIONAL (IDENTIFICADOR | **VALORES**) SIGNO\_AGRUPACION\_4 SIGNO\_AGRUPACION\_1 (**MS** | **MIC** | **MIL** | **A** | **DCV** | **DSV** | **FMCV** | **FMSV** | **LFCV** | **LFSV** | **MD** | **ME** | **MA**)\* SIGNO\_AGRUPACION\_2 ESTRUCTURA\_DE\_CONTROL\_5 SIGNO\_AGRUPACION\_1 (**MS** | **MIC** | **MIL** | **A** | **DCV** | **DSV** | **FMCV** | **FMSV** | **LFCV** | **LFSV** | **MD** | **ME** | **MA**)\* SIGNO\_AGRUPACION\_2 | if(number1>number2){  call moverMotor()!  }else{  call detenerMotor()!  } | |
| **Producción Else Lógico con dos valores (EL2V)** | | |
| ESTRUCTURA\_DE\_CONTROL\_4 SIGNO\_AGRUPACION\_3 (IDENTIFICADOR | **VALORES**) OPERADOR\_LOGICO (IDENTIFICADOR | **VALORES**) SIGNO\_AGRUPACION\_4 SIGNO\_AGRUPACION\_1 (**MS** | **MIC** | **MIL** | **A** | **DCV** | **DSV** | **FMCV** | **FMSV** | **LFCV** | **LFSV** | **MD** | **ME** | **MA**)\* SIGNO\_AGRUPACION\_2 ESTRUCTURA\_DE\_CONTROL\_5 SIGNO\_AGRUPACION\_1 (**MS** | **MIC** | **MIL** | **A** | **DCV** | **DSV** | **FMCV** | **FMSV** | **LFCV** | **LFSV** | **MD** | **ME** | **MA**)\* SIGNO\_AGRUPACION\_2 | if(number1 and number2){  call moverMotor()!  }else{  call detenerMotor()!  } | |
| **Producción Else Lógico con un valor (EL1V)** | | |
| ESTRUCTURA\_DE\_CONTROL\_4 SIGNO\_AGRUPACION\_3 (IDENTIFICADOR | **VALORES**) SIGNO\_AGRUPACION\_4 SIGNO\_AGRUPACION\_1 (**MS** | **MIC** | **MIL** | **A** | **DCV** | **DSV** | **FMCV** | **FMSV** | **LFCV** | **LFSV** | **MD** | **ME** | **MA**)\* SIGNO\_AGRUPACION\_2 ESTRUCTURA\_DE\_CONTROL\_5 SIGNO\_AGRUPACION\_1 (**MS** | **MIC** | **MIL** | **A** | **DCV** | **DSV** | **FMCV** | **FMSV** | **LFCV** | **LFSV** | **MD** | **ME** | **MA**)\* SIGNO\_AGRUPACION\_2 | if(number1){  call moverMotor()!  }else{  call detenerMotor()!  } | |
| **Producción Loop (L)** | | |
| ESTRUCTURA\_DE\_CONTROL\_3 SIGNO\_AGRUPACION\_1 (**MS** | **MIC** | **MIL** | **A** | **DCV** | **DSV** | **FMCV** | **FMSV** | **LFCV** | **LFSV** | **IR** | **IL2V** | **IL1V** | **ER** | **EL2V** | **EL1V** | **MD** | **ME** | **MA**)\* SIGNO\_AGRUPACION\_2 | loop{  if(number1>number2){  call moverMotor()!  }else{  call detenerMotor()!  }  call Distancia = puerto1.distance()!  number2 = 25.0 !  } | |