

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	print(console,'El motor1 se esta moviendo')!

IDENTIFICADOR) SIGNO_AGRUPACION_4 FIN_DE_LINEA	
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	if(number1 and number2){ call moverMotor()!

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	}
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	<pre>if(number1){ call moverMotor()! }</pre>
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	<pre>if(number1>number2){ call moverMotor()! }else{ call detenerMotor()! }</pre>
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	<pre>if(number1 and number2){ call moverMotor()! }else{ call detenerMotor()! }</pre>
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	<pre>if(number1){ call moverMotor()! }else{ call detenerMotor()! }</pre>

DCV DSV FMCV FMSV LFCV LFSV MD ME MA)* SIGNO_AGRUPACION_2	
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	<pre> loop{ if(number1>number2){ call moverMotor()! }else{ call detenerMotor()! } call Distancia = puerto1.distance()! number2 = 25.0 ! } </pre>