

# M4TH3C (Mathec)



## MATERIA

Lenguajes y Autómatas II

## DOCENTE

Maria Elena Parra Urias

## DESARROLLADORES

Martin Ernesto Barrón Doroteo  
Jesús Alberto Partida Michel  
Marlett Alessandra Marín Bautista  
Jose de Jesús Ramirez Ortega

# Contenido

<b>Contenido</b>	<b>2</b>
<b>Prefacio</b>	<b>4</b>
<b>Agradecimientos</b>	<b>5</b>
<b>Introducción</b>	<b>6</b>
<b>CAPÍTULO 1. Generalidades</b>	<b>7</b>
1.1 Sobre M4TH3C	7
1.2 Objetivo	7
1.3 Requerimientos de sistema	7
1.4 Requerimientos del programa	7
1.5 Ayuda adicional	8
<b>CAPÍTULO 2. M4TH3C, Lenguaje</b>	<b>8</b>
2.1 Sobre el lenguaje	8
2.2 Alfabeto, Palabras reservadas y detalles específicos	8
2.3 Estructura general de un programa en M4TH3C	8
<b>CAPÍTULO 3. M4TH3C, Compilador</b>	<b>9</b>
3.1 Sobre el compilador	9
3.2 Vista general	9
3.2.1 Estructura del proyecto	9
<b>CAPÍTULO 4. Análisis Semántico</b>	<b>12</b>
4.1 ¿Qué es el análisis semántico?	12
4.2 Método semanticAnalysis	12
4.3 Errores Semánticos	15
<b>CAPÍTULO 5. Código intermedio</b>	<b>16</b>
5.1 ¿Qué es el código intermedio?	16
5.2 Generación de código intermedio	16
5.2.1 Tripletas	16
5.2.2 Cuádruplos	20
5.2.3 TabledPanel Cuádruplos y tripletas.	25
<b>CAPÍTULO 6. Optimización</b>	<b>25</b>
6.1 ¿Qué es la optimización?	25
6.2 Generación de la optimización	25
<b>CAPÍTULO 7. Código objeto</b>	<b>25</b>
7.1 ¿Qué es el código objeto?	25
7.2 Generación del código objeto	25
7.3 Resultados	25
<b>Apéndice A. Las que siempre son nombradas</b>	<b>28</b>



A.1 Alfabeto	28
A.1.1 Identificadores	28
A.1.2 Signos de agrupación	29
A.1.3 Números	29
A.1.4 Operador de asignación	30
A.1.5 Delimitador de sentencia	30
A.1.6 Operadores aritméticos	30
A.1.7 Separador	31
A.1.8 Palabras reservadas	31
A.2 Tabla de tokens	33
A.3 Tipos de datos	33
A.4 Errores	34
A.4.1 Errores léxicos	34
A.4.2 Errores sintácticos	35
<b>Apéndice B. Ejemplo funcional</b>	<b>37</b>
B.1 Método	37
B.2 Compilación	38
B.2.1 Generación de la tabla de tokens	38
B.2.2 Generación de tripletas	38
B.2.3 Generación de cuádruplos	38
B.2.4 Generación de código objeto	38
<b>Apéndice C. Para descargar e instalar</b>	<b>38</b>

## Prefacio

**Martin Ernesto Barrón Doroteo**, estudiante del Tecnológico Nacional de México, en el Instituto Tecnológico de Tepic, donde actualmente cursa la carrera de Ingeniería en Sistemas Computacionales. Impulsado por su interés en las ciencias de la computación, espera poder terminar con éxito sus estudios para dedicarse a alguna rama del

**Jesús Alberto Partida Michel**, estudiante del Tecnológico Nacional de México, en el que cursa la carrera de Ingeniería en Sistemas Computacionales, en el campus Instituto Tecnológico de Tepic, ingresado al campus en el año 2019, y permaneciendo vigente hasta la fecha, en ese tiempo logrando participar en varios proyectos de desarrollo de software, donde gracias a la experiencia de dichas participaciones en los proyectos ha adquirido conocimiento en el área de software ayudando a formarse como profesional.

**Marlett Alessandra Marín Bautista**, estudiante del Tecnológico Nacional de México, en el que cursa la carrera de Ingeniería en Sistemas Computacionales, desde el campus ITT, o Instituto Tecnológico de Tepic. Ingreso en el año del 2019, a la carrera en la que continúa en la actualidad, donde se presentaron oportunidades de participación en convocatorias de desarrollo de proyectos, impulsando sus capacidades y habilidades adquiridas a lo largo de su formación superior.

**Jose de Jesús Ramirez Ortega**, estudiante del Tecnológico Nacional de México, donde cursa la carrera de Ingeniería en Sistemas Computacionales. Ha sido partícipe de algunos torneos deportivos en la categoría de fútbol varonil dentro del mismo instituto, además el gusto por la tecnología lo ha motivado a cursar la carrera de ISC en donde ha desempeñado actividades de desarrollo de proyectos a lo largo de la carrera para poder desarrollar sus conocimientos, así como también sus habilidades para trabajar en equipo y demás capacidades a lo largo de su formación en el nivel superior.

## Agradecimientos

M4TH3C agradece a los desarrolladores Martin Ernesto Barrón Doroteo, Jesús Alberto Partida Michel, Marlett Alessandra Marín Bautista y Jose de Jesús Ramirez Ortega por la generación del lenguaje de programación, el software para la compilación del lenguaje y los manuales que se elaboraron.

Se desea también expresar gratitud a la maestra Maria Elena Parra Urias, la cual impartió la materia de Lenguajes y Automatas II en el periodo de Agosto-Diciembre 2022, por aportar sus conocimientos y confianza, así como su capacidad de enseñanza la cual pone en evidencia su grana habilidad para la impartición de los tópicos de la asignatura de manera que esto demuestra lo buen profesora y la responsabilidad que demuestra para las actividades académicas.

## Introducción

Existen actividades que tienen un inicio y un fin, ya que estas pueden tener un tiempo mayor o menor. Durante un año se comenzó la creación del proyecto M4TH3C el cual llega a su culminación, el cual aborda la generación de un lenguaje de programación y el desarrollo de un compilador para el mismo lenguaje enfocado para niños de 12 años los cuales tengan la posibilidad de aprender sobre la programación mediante la resolución de problemas matemáticos empleando aritmética básica.

Los autores presentamos este manual técnico, el cual se enfoca en dar a conocer los elementos, análisis semántico, generación de código intermedio, generación de código objeto que genera el proyecto el cual contempla un lenguaje de programación de modo que también se dan a conocer algunos requisitos e información sobre la intención que tiene el proyecto y el lenguaje de programación que compila.

Con la culminación de este proyecto se pretende dar a conocer todo lo que conlleva generar un producto de este tipo el cual requiere de varias etapas para poder tener una clara organización de los que se genera debido a que son un poco complejas las estructuras que se manejan, además también requiere de un análisis para determinar las funcionalidades de la interfaz y el diseño que se tendrá para poder generar que el manejo por parte del usuario sea lo más eficientemente posible. Deja una sensación satisfactoria la generación de resultados que se han obtenido ya que esto quiere decir que el trabajo y tiempo invertido han valido la pena.

## CAPÍTULO 1. Generalidades

### 1.1 Sobre M4TH3C

El compilador M4TH3C es un programa para el uso exclusivo de un lenguaje de programación enfocado a niños de hasta 12 años para aprender la lógica de programación con la ayuda de resolución de problemas matemáticos básicos en aritmética.

En este contenido se presentan generalidades de las características y detalles sobre las estructuras que se abordan para la generación del análisis semántico y la generación de código intermedio.

### 1.2 Objetivo

Entregar una guía de soporte al programador sobre la generación del software compilador M4TH3C, con el objetivo de sintetizar las instrucciones y permitir el correcto funcionamiento del compilador, así como la aplicación del lenguaje de programación homónimo específico, el cual emplea un lenguaje de programación creado por nosotros mismos.

### 1.3 Requerimientos de sistema

Para el buen funcionamiento del compilador M4TH3C, se presentan los siguientes requerimientos mínimos con los que deberá contar la computadora en la que se desea instalar dicho programa.

Procesador	Ryzen 3 1200 o Intel Core i5-7400
RAM	4 GB
Espacio disponible en disco	300 MB
Sistema Operativo	Windows 7 o superior
JDK(Herramientas de desarrollo)	292 MB

### 1.4 Requerimientos del programa

Para crear el proyecto M4TH3C, es necesario contar con los siguientes elementos instalados en la computadora donde se pretende llevar a cabo la programación. En caso de no contar con dichas herramientas, se necesitará su posterior instalación. En el Apéndice C se proporcionan enlaces para ello.

IDE	NetBeans 13
Lenguaje de programación	Java

Tipo de proyecto	Java Application
Librerías	compilerTools-2.3.7 flatlaf-2.2 jflex-full-1.8.2 JDK 18.0.2.1

## 1.5 Ayuda adicional

El lenguaje de programación M4TH3C cuenta con tres tomos publicados, los cuales se pueden consultar mediante los siguientes enlaces:

[Tomo I: M4TH3C Lenguaje de Programacion enfocado para niños de 10 a 12 años para que aprendan la logica de programacion mediante resolucion de problemas matematicos basados en aritmetica basica](#)

[Tomo II: M4TH3C Manual Tecnico. Lenguaje de Programacion enfocado para niños de 10 a 12 años para que aprendan la logica de programacion mediante resolucion de problemas matematicos basados en aritmetica basica](#)

[Tomo III: M4TH3C Manual de Usuario. Lenguaje de Programacion enfocado para niños de 10 a 12 años para que aprendan la logica de programacion mediante resolucion de problemas matematicos basados en aritmetica basica](#)

## CAPÍTULO 2. M4TH3C, Lenguaje

### 2.1 Sobre el lenguaje

M4TH3C es un lenguaje de programación específico, sin embargo trae consigo algunas limitantes por lo tanto se espera que mediante este lenguaje se generen programas útiles para que los niños de hasta 12 años comprendan la lógica de programación para la resolución de problemas matemáticos, considerando variables, flujos y las base de lo que trae consigo este lenguaje de programación debido que a por muy familiarizado que se pueda estar con el lenguaje se generan casos en los que se podrían presentar problemas con las estructura ya que estas pueden diferir.

### 2.2 Alfabeto, Palabras reservadas y detalles específicos

Las palabras reservadas en programación, o palabras clave, son aquellas palabras que tienen una función ya definida en el lenguaje. Para M4TH3C, existen un total de 22 palabras reservadas. Estas, se encuentran en el **Apéndice A** del presente libro, así como también el alfabeto , los tokens empleados, identificadores y los tipos de datos.



## 2.3 Estructura general de un programa en M4TH3C

La estructura del código del programa para el compilador M4TH3C está constituida por el método principal en la que se declaran las variables y a su vez realiza los métodos tales como Sumar, Restar, Multiplicar, Dividir y Mostrar.

```

1) Método &Prueba{
2) Decimal num3 = - 10.61;
3) Decimal num4;
4) Resultado #res2;
5) Resultado #res3;
6) Resultado #res4;
7) Cadena cad1 = 'Hola';
8) Cadena cad4;
9) Figura fig1 = Circulo;
10) Figura fig2;
11) Color col1 = Azul;
12) Color col2;
13) num4 = + 10.45;
14) cad4 = 'El resultado:.';
15) fig2 = Circulo;
16) col2 = Rojo;
17) Multiplicar(num4,num3)<
18) #res3
19) >;
20) Mostrar(cad4,#res3,fig1,col1,fig2,col2,fig1,col1,fig2,col2,fig1,col1);
21) }

```

## CAPÍTULO 3. M4TH3C, Compilador

### 3.1 Sobre el compilador

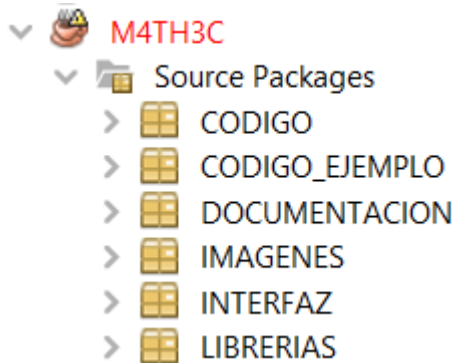
Un compilador es un software que traduce un programa escrito en lenguaje de alto nivel a un programa en lenguaje de bajo nivel, en este caso puede ser lenguaje máquina que sea ejecutable, sin embargo se genera el lenguaje ensamblador y después se traduce a un lenguaje máquina de forma que mediante la traducción se puede generar la optimización del código para que presente mejores características las cuales son la velocidad y los tiempos de ejecución, teniendo como resultado código realmente útil y eliminando el código innecesario,

El compilador M4TH3C traduce un lenguaje de alto nivel también llamado M4TH3C con la posibilidad de traducirlo a lenguaje máquina ejecutable por el emu8086.

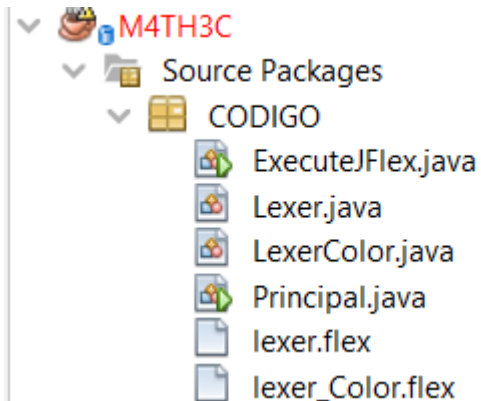
## 3.2 Vista general

### 3.2.1 Estructura del proyecto

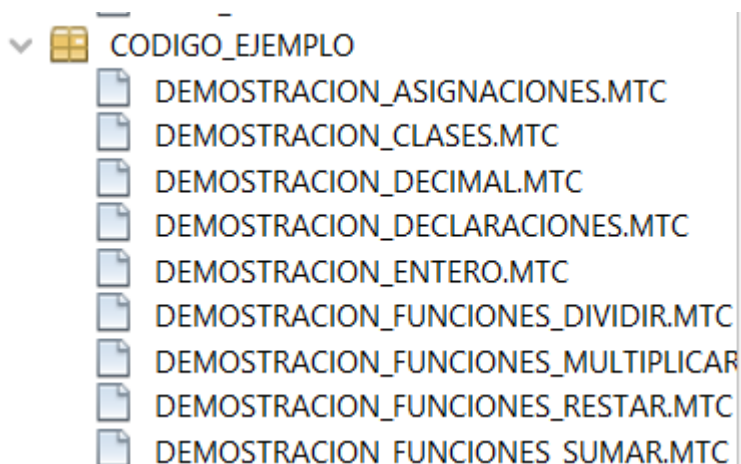
La estructura del proyecto M4TH3C consiste en 6 paquetes ubicados en la carpeta Source Packages, los cuales son los siguientes:



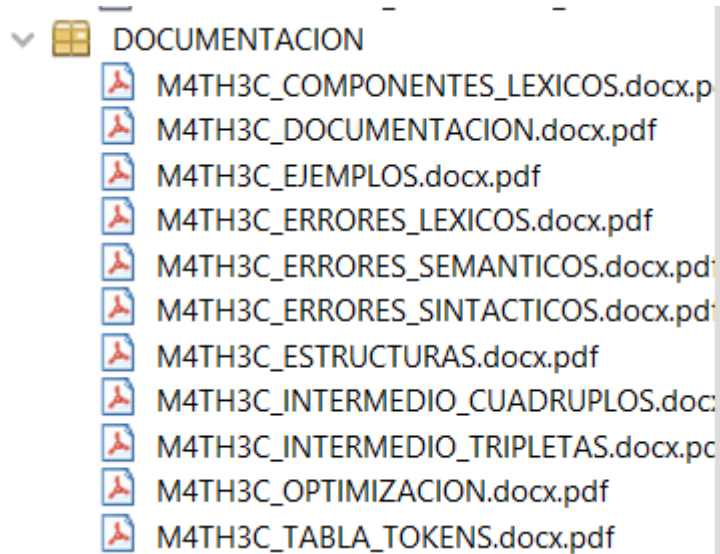
El package “CODIGO” contiene los .java referentes a la parte léxica del compilador.



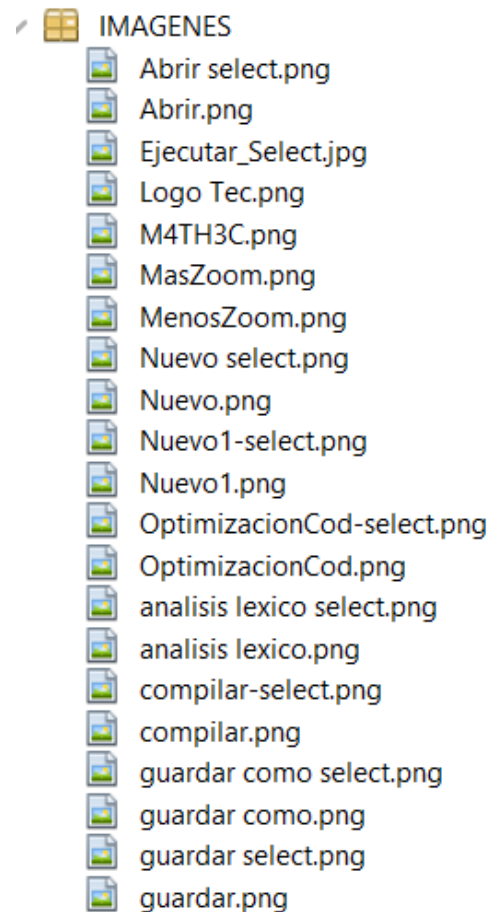
En el package “CODIGO\_EJEMPLO” se encuentran algunos archivos de ejemplo generados con el compilador para servir de apoyo al usuario para realizar declaraciones y operaciones.



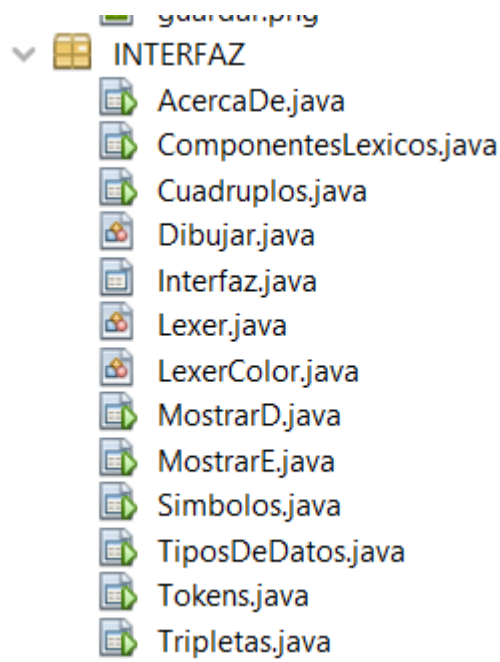
En el package “DOCUMENTACION” se encuentran archivos de documentación referente al compilador con la finalidad de que el usuario tenga la facilidad de poder visualizarlos.



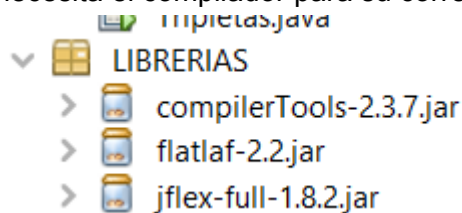
En el package “IMAGENES” se encuentran las imágenes que se utilizan en la interfaz.



En el package “INTERFAZ” se encuentran archivos referentes a las pantallas que contiene la interfaz.



Por último tenemos el package “LIBRERIAS” en la cual se encuentran las librerías que necesita el compilador para su correcto funcionamiento.



## CAPÍTULO 4. Análisis Semántico

### 4.1 ¿Qué es el análisis semántico?

La Semántica de un lenguaje de programación es el conjunto de reglas que especifican el significado de cualquier sentencia sintácticamente válida, en el ámbito de la programación, el análisis semántico es la fase en la cual se comprueba la corrección semántica de un programa, es decir, que aquello que se escribe corresponde a lo que se pretende siguiendo determinadas estructuras. En esta fase, la mayoría de los errores semánticos no dependen del programa, sino del programador o usuario final.

### 4.2 Método semanticAnalysis

El Método semanticAnalysis corresponde al análisis semántico del compilador M4TH3C, En él, se encuentran programadas las sentencias de detección de errores semánticos, el compilador podrá mostrar los errores semánticos que se presenten en el código al momento de que el usuario final se dedique a programarlo con el lenguaje M4TH3C.

A continuación, se muestran las reglas semánticas que se abordan en M4TH3C las cuales corresponden a ciertas estructuras del mismo lenguaje.

### Métodos Declarativos

Estos métodos nos ayudaran para declarar las variables en el lenguaje del cual sigue la siguiente sintaxis dependiendo su tipo, que es muy similar.

<b>Declarar variable tipo Entero</b>
Entero 'Nombre De Variable' = 'Valor De Variable' ;
Entero 'Nombre De Variable' ;
<b>Declarar variable tipo Decimal</b>
Decimal 'Nombre De Variable' = 'Valor De Variable' ;
Decimal 'Nombre De Variable' ;
<b>Declarar variable tipo Resultado</b>
Resultado #'Nombre De Variable' ;
<b>Declarar variable tipo Cadena</b>
Cadena 'Nombre De Variable' = " 'Valor De Variable' " ;
Cadena 'Nombre De Variable' ;
<b>Declarar variable tipo Figura</b>
Figura 'Nombre De Variable' = 'Valor De Variable' ;
Figura 'Nombre De Variable' ;
<b>Declarar variable tipo Color</b>
Color 'Nombre De Variable' = 'Valor De Variable' ;
Color 'Nombre De Variable' ;

### Métodos de asignación

Estos métodos sirven para darle un nuevo valor a la variable del cual esto depende del tipo de dato de la variable que es muy similar su sintaxis.

<b>Variable tipo Entero</b>
'Nombre De Variable' = 'Valor Nuevo De Variable' ;
<b>Variable tipo Decimal</b>
'Nombre De Variable' = 'Valor Nuevo De Variable' ;

<b>Variable tipo Cadena</b>
'Nombre De Variable' = "Valor Nuevo De Variable" ;
<b>Variable tipo Figura</b>
'Nombre De Variable' = 'Valor Nuevo De Variable' ;
<b>Variable tipo Color</b>
'Nombre De Variable' = 'Valor Nuevo De Variable' ;

### Métodos de Funciones

Estos métodos sirven para realizar las operaciones matemáticas del lenguaje en donde solo se pueden realizar con variables del mismo tipo, y solo pueden recibir máximo 5 variables para los casos de Sumar, Restar, Multiplicar y Dividir;

<b>Método de Sumar</b>
Sumar('Variable Entero 1   Variable Decimal 1' , 'Variable Entero 2   Variable Decimal 2' , ... , 'Variable Entero 5   Variable Decimal 5') < 'Variable Tipo Resultado' >;
<b>Método de Restar</b>
Restar('Variable Entero 1   Variable Decimal 1' , 'Variable Entero 2   Variable Decimal 2' , ... , 'Variable Entero 5   Variable Decimal 5') < 'Variable Tipo Resultado' >;
<b>Método de Multiplicar</b>
Multiplicar ('Variable Entero 1   Variable Decimal 1' , 'Variable Entero 2   Variable Decimal 2' , ... , 'Variable Entero 5   Variable Decimal 5') < 'Variable Tipo Resultado' >;
<b>Método de Dividir</b>
Dividir ('Variable Entero 1   Variable Decimal 1' , 'Variable Entero 2   Variable Decimal 2' , ... , 'Variable Entero 5   Variable Decimal 5') < 'Variable Tipo Resultado' >;
<b>Método de Mostrar</b>
En caso de que la variable resultado sea decimal
Mostrar('Variable Cadena' , 'Variable Resultado' , 'Variable Figura 1' , 'Variable Color 1' , 'Variable Figura 2' , 'Variable Color 2' , 'Variable Figura 3' , 'Variable Color 3' , 'Variable Figura 4' , 'Variable Color 4' , 'Variable Figura 5' , 'Variable Color 5' );
En caso de que la variable resultado sea entero

Mostrar('Variable Cadena' , 'Variable Resultado' , 'Variable Figura' , 'Variable Color' );

### Método de Clase

Este método nos sirve para poder ejecutar un conjunto de funciones que estén definidas dentro del método, solo contamos con un método y es el siguiente, del cual, como máximo acepta un total de 30 funciones, declaraciones o asignaciones.

Método de Clase
Método &Nombre Del Método' { 'Funciones, Declaraciones o Asignaciones del Método'; }

### 4.3 Errores Semánticos

A continuación, se muestran los tipos de errores semánticos contenidos en M4TH3C, así como los mensajes de errores semánticos mostrados al usuario en caso de detectar alguna discordancia con la estructura esperada.

Declarativos	
Nombre del error	Origen del error
Error 55	La variable ya ha sido declarada
Error 56	El valor de la variable declarada no esta permitido
Asignación	
Nombre del error	Origen del error
Error 57	La variable no ha sido declarada
Error 58	El valor que se le asigna a la variable no está permitido
Funciones	
Nombre del error	Origen del error
Error 59	Se están empleando variables de tipos de datos distintos para la función
Error 60	Una de las variables empleadas en la función no ha sido declarada
Error 61	Varias variables empleadas en la función no han sido declaradas

Error 62	Una de las variables empleadas en la función no tiene valor asignado
Error 63	Varias variables empleadas en la función no tienen valor asignado
Error 64	El valor de una variable empleada en la función no está permitido
Error 65	El valor de varias variables empleadas en la función no están permitidos
Error 66	El valor del resultado de la operación realizada en la función no está en el rango permitido
Error 67	La variable ya tiene un valor asignado
<b>Clase</b>	
<b>Nombre del error</b>	<b>Origen del error</b>
Error 68	La clase ya existe

## CAPÍTULO 5. Código intermedio

### 5.1 ¿Qué es el código intermedio?

El código intermedio es un código abstracto independiente de la máquina para la que se generará el código objeto, este debe cumplir 2 requisitos: ser fácil de producir a partir del análisis sintáctico y ser fácil de traducir al lenguaje objeto, la generación del código intermedio es una representación similar al código máquina, generalmente un código intermedio hace referencia a secuencias de tres direcciones en donde se contemplan tres operandos por instrucción, sin embargo es independiente del lenguaje máquina.

### 5.2 Generación de código intermedio

El método *executeCode* que se invoca desde el *actionPerformed* del botón contiene el código necesario para realizar la generación del código intermedio que realiza el compilador MATH3C. En este método se encuentran programadas todas las sentencias para poder generar el código de modo que se generan tanto tripletas como cuádruplas las cuales en este caso las tripletas se agregan a un *textarea* y las cuádruplas a una tabla de modo que en la misma generación del código se realiza la optimización de sentencias que ya se generaron de modo que mediante estas solo se reutiliza su valor para poder generar la optimización del código. El método recibe un *arraylist* de tipo *string* y un dato del tipo entero.



```
private void executeCode(ArrayList<String> blocksOfCode,int repeats){
    int temporal = 0;
    String metodo = "";
    int contador = 0;
    for(int j = 1;j <= repeats; j++){
        int repeatCode = -1;
        for(int i = 0; i < blocksOfCode.size(); i++){
            String blockOfCode = blocksOfCode.get(i);
            if(repeatCode != -1){
                int[] posicionMarcador = CodeBlock.getPositionOfBothMarkers(blocksOfCode, blockOfCode);
                executeCode(new ArrayList<>(blocksOfCode.subList(posicionMarcador[0], posicionMarcador[1])),repeatCode);
            }
        }
    }
}
```

La generación del código intermedio es a través de las estructuras del lenguaje y se genera después de ejecutar el código en el compilador.

### 5.2.1 Tripletas

La generación de tripletas se realiza dependiendo de cuántos operandos tenga como parámetro la función ya sea para sumar, restar, multiplicar y dividir, cabe recordar que las funciones utilizan parámetros del mismo tipo, es decir, pueden ser enteros o decimales pero no se pueden contemplar variables que tengan distintos tipos de valor entero o decimal.

ESTRUCTURA DE DECLARACIONES	
<b>Tipo Entero</b>	
Entero 'Nombre De Variable' = 'Valor De Variable' ;	T1 := 'Nombre de variable' T2 := T1 + 'Valor De Variable' T3 := 1 - 2 'Nombre De Variable' := T2 * T3
	T1 := 'Nombre de variable' T2 := T1 + 'Valor De Variable' 'Nombre De Variable' := T2
Entero 'Nombre De Variable' ;	T1 := 'Nombre de variable'
<b>Tipo Decimal</b>	
Decimal 'Nombre De Variable' = 'Valor De Variable' ;	T1 := 'Nombre de variable' T2 := T1 + 'Valor De Variable' T3 := 1 - 2 'Nombre De Variable' := T2 * T3
	T1 := 'Nombre de variable' T2 := T1 + 'Valor De Variable' 'Nombre De Variable' := T2
Decimal 'Nombre De Variable' ;	T1 := 'Nombre de variable'
<b>Tipo Resultado</b>	

Resultado #‘Nombre De Variable’ ;	T1 := ‘Nombre de variable’
<b>Tipo Cadena</b>	
Cadena ‘Nombre De Variable’ = “ ‘Valor De Variable’ ” ;	T1 := ‘Nombre de variable’ T2 := T1 + ‘Valor De Variable’ ‘Nombre De Variable’ := T2
Cadena ‘Nombre De Variable’ ;	T1 := ‘Nombre de variable’
<b>Tipo Figura</b>	
Figura ‘Nombre De Variable’ = ‘Valor De Variable’ ;	T1 := ‘Nombre de variable’ T2 := T1 + ‘Valor De Variable’ ‘Nombre De Variable’ := T2
Figura ‘Nombre De Variable’ ;	T1 := ‘Nombre de variable’
<b>Tipo Color</b>	
Color ‘Nombre De Variable’ = ‘Valor De Variable’ ;	T1 := ‘Nombre de variable’ T2 := T1 + ‘Valor De Variable’ ‘Nombre De Variable’ := T2
Color ‘Nombre De Variable’ ;	T1 := ‘Nombre de variable’
<b>ESTRUCTURA DE ASIGNACIONES</b>	
<b>Tipo Entero</b>	
‘Nombre De Variable’ = ‘Valor Nuevo De Variable’ ;	T1 := ‘Temporal de la variable’ + ‘Valor De Variable’ T2 := 1 - 2 ‘Nombre De Variable’ := T1 * T2
	T1 := ‘Temporal de la variable’ + ‘Valor De Variable’ ‘Nombre De Variable’ := T1
<b>Tipo Decimal</b>	
‘Nombre De Variable’ = ‘Valor Nuevo De Variable’ ;	T1 := ‘Temporal de la variable’ + ‘Valor De Variable’ T2 := 1 - 2 ‘Nombre De Variable’ := T1 * T2
	T1 := ‘Temporal de la variable’ + ‘Valor De Variable’ ‘Nombre De Variable’ := T1
<b>Tipo Cadena</b>	
‘Nombre De Variable’ = “‘Valor Nuevo De Variable’” ;	T1 := ‘Temporal de la variable’ + ‘Valor De Variable’ ‘Nombre De Variable’ := T1
<b>Tipo Figura</b>	
‘Nombre De Variable’ = ‘Valor Nuevo De Variable’ ;	T1 := ‘Temporal de la variable’ + ‘Valor De Variable’ ‘Nombre De Variable’ := T1

Tipo Color	
'Nombre De Variable' = 'Valor Nuevo De Variable' ;	T1 := 'Temporal de la variable' + 'Valor De Variable' 'Nombre De Variable' := T1
ESTRUCTURA DE FUNCIONES	
De Sumar	
Sumar('Variable Entero 1   Variable Decimal 1', 'Variable Entero 2   Variable Decimal 2', ... , 'Variable Entero 5   Variable Decimal 5') < 'Variable Tipo Resultado' >;	T1 := 'Variable 1' + 'Variable 2' T2 := T1 + 'Variable 3' T3 := T2 + 'Variable 4' T4 := T3 + 'Variable 5' 'Variable resultado' := T4
De Restar	
Restar('Variable Entero 1   Variable Decimal 1', 'Variable Entero 2   Variable Decimal 2', ... , 'Variable Entero 5   Variable Decimal 5') < 'Variable Tipo Resultado' >;	T1 := 'Variable 1' - 'Variable 2' T2 := T1 - 'Variable 3' T3 := T2 - 'Variable 4' T4 := T3 - 'Variable 5' 'Variable resultado' := T4
De Multiplicar	
Multiplicar ('Variable Entero 1   Variable Decimal 1', 'Variable Entero 2   Variable Decimal 2', ... , 'Variable Entero 5   Variable Decimal 5') < 'Variable Tipo Resultado' >;	T1 := 'Variable 1' * 'Variable 2' T2 := T1 * 'Variable 3' T3 := T2 * 'Variable 4' T4 := T3 * 'Variable 5' 'Variable resultado' := T4
De Dividir	
Dividir ('Variable Entero 1   Variable Decimal 1', 'Variable Entero 2   Variable Decimal 2', ... , 'Variable Entero 5   Variable Decimal 5') < 'Variable Tipo Resultado' >;	T1 := 'Variable 1' / 'Variable 2' T2 := T1 / 'Variable 3' T3 := T2 / 'Variable 4' T4 := T3 / 'Variable 5' 'Variable resultado' := T4
De Mostrar	
En caso de que la variable resultado sea decimal	
Mostrar('Variable Cadena', 'Variable Resultado', 'Variable Figura 1', 'Variable Color 1', 'Variable Figura 2', 'Variable Color 2', 'Variable Figura 3', 'Variable Color 3', 'Variable Figura 4', 'Variable Color 4', 'Variable Figura 5', 'Variable Color 5');	T1 := 'Variable Cadena' + 'Variable Resultado' T2 := 'Variable Figura 1' + 'Variable Color 1' T3 := 'Variable Figura 2' + 'Variable Color 2' T4 := T2 + T3 T5 := 'Variable Figura 3' + 'Variable Color 3' T6 := 'Variable Figura 4' + 'Variable Color 4' T7 := T5 + T6 T8 := 'Variable Figura 5' + 'Variable Color 5'

	$T9 := T4 + T7$ $T10 := T9 + T8$ $T11 := T1 + T10$
En caso de que la variable resultado sea entero	
Mostrar('Variable Cadena' , 'Variable Resultado' , 'Variable Figura' , 'Variable Color' );	$T1 := \text{'Variable Cadena'} + \text{'Variable Resultado'}$ $T2 := \text{'Variable Figura'} + \text{'Variable Color'}$ $T3 := T1 + T2$
<b>ESTRUCTURA DE CLASES</b>	
<b>De Clase</b>	
Método &Nombre Del Método' { 'Funciones, Declaraciones o Asignaciones del Método'; }	$T1 := \text{'&Nombre Del Método'}$ $\text{'&Nombre Del Método'} := T1$

### 5.2.2 Cuádruplos

La generación de cuádruplas se realiza dependiendo de cuántos operandos tenga como parámetro la función ya sea para sumar, restar, multiplicar y dividir, cabe recordar que las funciones utilizan parámetros del mismo tipo, es decir, pueden ser enteros o decimales pero no se pueden contemplar variables que tengan distintos tipos de valor entero o decimal.

<b>ESTRUCTURA DE DECLARACIONES</b>				
<b>Tipo Entero</b>				
Entero 'Nombre De Variable' = 'Valor De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1
	+	T1	'Valor De Variable'	T2
	-	1	2	T3
	*	T2	T3	'Nombre De Variable'
	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1

	+	T1	'Valor De variable'	T2
	=	T2		'Nombre De Variable'
Entero 'Nombre De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1
<b>Tipo Decimal</b>				
Decimal 'Nombre De Variable' = 'Valor De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1
	+	T1	'Valor De Variable'	T2
	-	1	2	T3
	*	T2	T3	'Nombre De Variable'
	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1
	+	T1	'Valor De variable'	T2
	=	T2		'Nombre De Variable'
Decimal 'Nombre De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1
<b>Tipo Resultado</b>				
Resultado # 'Nombre De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1
<b>Tipo Cadena</b>				
Cadena 'Nombre De Variable' = " 'Valor De Variable' " ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1
	+	T1	'Valor De variable'	T2
	=	T2		'Nombre De Variable'
Cadena 'Nombre De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1
<b>Tipo Figura</b>				

Figura 'Nombre De Variable' = 'Valor De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1
	+	T1	'Valor De variable'	T2
	=	T2		'Nombre De Variable'
Figura 'Nombre De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1
<b>Tipo Color</b>				
Color 'Nombre De Variable' = 'Valor De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1
	+	T1	'Valor De variable'	T2
	=	T2		'Nombre De Variable'
Color 'Nombre De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'Nombre de variable'		T1
<b>ESTRUCTURA DE ASIGNACIONES</b>				
<b>Tipo Entero</b>				
'Nombre De Variable' = 'Valor Nuevo De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	+	'Temporal de la variable'	'Valor De Variable'	T1
	-	1	2	T2
	*	T1	T2	'Nombre De Variable'
	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	+	'Temporal de la variable'	'Valor De Variable'	'Nombre De Variable'
	=	T1		'Nombre De Variable'
<b>Tipo Decimal</b>				
'Nombre De Variable' = 'Valor Nuevo De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	+	'Temporal de la variable'	'Valor De Variable'	T1
	-	1	2	T2
	*	T1	T2	'Nombre De Variable'
	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>

	+	'Temporal de la variable'	'Valor De Variable'	'Nombre De Variable'
	=	T1		'Nombre De Variable'
<b>Tipo Cadena</b>				
'Nombre De Variable' = “Valor Nuevo De Variable” ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	+	'Temporal de la variable'	'Valor De Variable'	'Nombre De Variable'
	=	T1		'Nombre De Variable'
<b>Tipo Figura</b>				
'Nombre De Variable' = 'Valor Nuevo De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	+	'Temporal de la variable'	'Valor De Variable'	'Nombre De Variable'
	=	T1		'Nombre De Variable'
<b>Tipo Color</b>				
'Nombre De Variable' = 'Valor Nuevo De Variable' ;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	+	'Temporal de la variable'	'Valor De Variable'	'Nombre De Variable'
	=	T1		'Nombre De Variable'
<b>ESTRUCTURA DE FUNCIONES</b>				
<b>De Sumar</b>				
Sumar('Variable Entero 1   Variable Decimal 1', 'Variable Entero 2   Variable Decimal 2', ... , 'Variable Entero 5   Variable Decimal 5') < 'Variable Tipo Resultado' >;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	+	'Variable 1'	'Variable 2'	T1
	+	T1	'Variable 3'	T2
	+	T2	'Variable 4'	T3
	+	T3	'Variable 5'	T4
	=	T4		'Variable resultado'
<b>De Restar</b>				
Restar('Variable Entero 1   Variable Decimal 1', 'Variable Entero 2   Variable Decimal 2', ... , 'Variable Entero 5   Variable Decimal 5') < 'Variable Tipo Resultado' >;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	-	'Variable 1'	'Variable 2'	T1
	-	T1	'Variable 3'	T2
	-	T2	'Variable 4'	T3
	-	T3	'Variable 5'	T4

	=	T4		'Variable resultado'
<b>De Multiplicar</b>				
Multiplicar ('Variable Entero 1   Variable Decimal 1', 'Variable Entero 2   Variable Decimal 2', ... , 'Variable Entero 5   Variable Decimal 5') < 'Variable Tipo Resultado' >;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	*	'Variable 1'	'Variable 2'	T1
	*	T1	'Variable 3'	T2
	*	T2	'Variable 4'	T3
	*	T3	'Variable 5'	T4
	=	T4		'Variable resultado'
<b>De Dividir</b>				
Dividir ('Variable Entero 1   Variable Decimal 1', 'Variable Entero 2   Variable Decimal 2', ... , 'Variable Entero 5   Variable Decimal 5') < 'Variable Tipo Resultado' >;	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	/	'Variable 1'	'Variable 2'	T1
	/	T1	'Variable 3'	T2
	/	T2	'Variable 4'	T3
	/	T3	'Variable 5'	T4
	=	T4		'Variable resultado'
<b>De Mostrar</b>				
En caso de que la variable resultado sea decimal				
Mostrar('Variable Cadena', 'Variable Resultado', 'Variable Figura 1', 'Variable Color 1', 'Variable Figura 2', 'Variable Color 2', 'Variable Figura 3', 'Variable Color 3', 'Variable Figura 4', 'Variable Color 4', 'Variable Figura 5', 'Variable Color 5');	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	+	'Variable Cadena'	'Variable Resultado'	T1
	+	'Variable Figura 1'	'Variable Color 1'	T2
	+	'Variable Figura 2'	'Variable Color 2'	T3
	+	T2	T3	T4
	+	'Variable Figura 3'	'Variable Color 3'	T5
	+	'Variable Figura 4'	'Variable Color 4'	T6
	+	T5	T6	T7
	+	'Variable Figura 5'	'Variable Color 5'	T8
	+	T4	T7	T9
	+	T9	T8	T10



	+	T1	T10	T11
En caso de que la variable resultado sea entero				
Mostrar('Variable Cadena' , 'Variable Resultado' , 'Variable Figura' , 'Variable Color' );	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	+	'Variable Cadena'	'Variable Resultado'	T1
	+	'Variable Figura'	'Variable Color'	T2
	+	T1	T2	T3
<b>ESTRUCTURA DE CLASES</b>				
<b>De Clase</b>				
Método &Nombre Del Método' {  'Funciones, Declaraciones o Asignaciones del Método';  }	<b>Op</b>	<b>Arg1</b>	<b>Arg2</b>	<b>Resultado</b>
	=	'&Nombre Del Método'		T1
	=	T1		'&Nombre Del Método'

### 5.2.3 TabledPanel Cuádruplos y tripletas.

En el caso del TabledPanel cuádruplos muestran los cuádruplos generados a manera de tabla, empezando por el operador, seguido de los argumentos y el resultado.

Mientras que TabledPanel tripletas muestra el código de tres direcciones empezando por el temporal, seguido del 1er argumento, operador y el 2do argumento.

Tokens Cuadruplos Tripletas			
Operador	Argumento 1	Argumento 2	Resultado

## CAPÍTULO 6. Optimización

### 6.1 ¿Qué es la optimización?

La optimización se define como la acción de desarrollar una actividad lo más eficientemente posible, es decir, con la menor cantidad de recursos y en el menor tiempo posible.

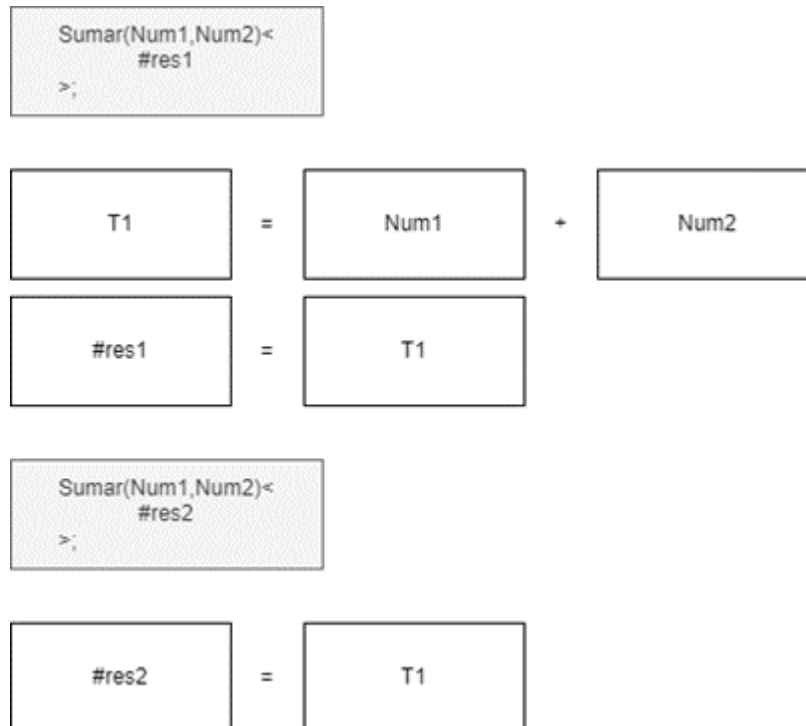
En este caso, la optimización cubre aspectos como la eliminación de código redundante, código muerto, la omisión de operaciones matemáticas innecesarias, etc.

## 6.2 Generación de la optimización

La optimización se divide dependiendo de las estructuras del lenguaje.

ESTRUCTURA DE DECLARACIONES
<p>En estas estructuras se genera optimización solamente al declarar una variable con valor de tipo Entero o Decimal, ya que esta genera la siguiente tripleta:</p> $T_n := 1 - 2$ <p>Del cual si ya se generó una vez dicha tripleta, ya no se vuelve a generar y cuando se ocupe, se manda a llamar la tripleta.</p>
ESTRUCTURA DE ASIGNACIONES
<p>En estas estructuras se genera optimización solamente al asignar un valor a una variable de tipo Entero o Decimal, ya que esta genera la siguiente tripleta:</p> $T_n := 1 - 2$ <p>Del cual si ya se generó una vez dicha tripleta, ya no se vuelve a generar y cuando se ocupe, se manda a llamar la tripleta.</p>
ESTRUCTURA DE FUNCIONES

En estas estructuras se genera optimización en todas, del cual en Sumar, Restar, Multiplicar y Dividir se toma en cuenta el valor resultante de la operación, ya sea de dos, tres, cuatro o cinco parámetros, y si dicho valor resultante se vuelve a generar en las tripletas, ya no se genera y cada vez que se ocupe se manda a llamar la primer tripleta de la operación, por ejemplo:



Y en mostrar se optimiza de manera similar, solo que en esto engloba Variable Figura con Variable Color y Variable Cadena con Variable Resultado, y si se genera una tripleta con un valor que ya se haya generado se manda a llamar esa en vez de generarse nuevamente.

## ESTRUCTURA DE CLASES

En estas estructuras no hay optimización, debido a que no es posible generar un código intermedio igual a otro.

## CAPÍTULO 7. Código objeto

### 7.1 ¿Qué es el código objeto?

El código objeto es un conjunto de códigos de instrucciones que una computadora entiende en el nivel de hardware más bajo, el código objeto generalmente es producido por un

compilador que lee algunas instrucciones del lenguaje fuente el cual es de un nivel superior y se traducen a instrucciones de lenguaje máquina equivalentes. Esta clase de código se produce cuando un compilador traduce el código fuente en código de máquina reconocible y ejecutable.

## 7.2 Generación del código objeto

La generación del código objeto es a través de las tripletas generadas de las estructuras del lenguaje y se genera después de ejecutar el código en el compilador.

ESTRUCTURA DE DECLARACIONES	
<b>Tipo Entero</b>	
T1 := 'Nombre de variable' T2 := T1 + 'Valor De Variable' T3 := 1 - 2 'Nombre De Variable' := T2 * T3	'Nombre de variable' dw 'Valor de Variable' Negativo dw "-" MOV AX,'Nombre de Variable' MUL Negativo MOV 'Nombre de Variable',AX
T1 := 'Nombre de variable' T2 := T1 + 'Valor De Variable' 'Nombre De Variable' := T2	'Nombre de variable' dw 'Valor de Variable'
T1 := 'Nombre de variable'	'Nombre de variable' dw 0
<b>Tipo Decimal</b>	
T1 := 'Nombre de variable' T2 := T1 + 'Valor De Variable' T3 := 1 - 2 'Nombre De Variable' := T2 * T3	'Nombre de variable' dw 'Valor de Variable' Negativo dw "-" MOV AX,'Nombre de Variable' MUL Negativo MOV 'Nombre de Variable',AX
T1 := 'Nombre de variable' T2 := T1 + 'Valor De Variable' 'Nombre De Variable' := T2	'Nombre de variable' dw 'Valor de Variable'
T1 := 'Nombre de variable'	'Nombre de variable' dw 0
<b>Tipo Resultado</b>	
T1 := 'Nombre de variable'	'Nombre de variable' dw 0
<b>Tipo Cadena</b>	
T1 := 'Nombre de variable' T2 := T1 + 'Valor De Variable' 'Nombre De Variable' := T2	'Nombre de variable' dw 'Valor de Variable'
T1 := 'Nombre de variable'	'Nombre de variable' dw 0
<b>Tipo Figura</b>	

T1 := 'Nombre de variable' T2 := T1 + 'Valor De Variable' 'Nombre De Variable' := T2	'Nombre de variable' dw 'Valor de Variable'
T1 := 'Nombre de variable'	'Nombre de variable' dw 0
<b>Tipo Color</b>	
T1 := 'Nombre de variable' T2 := T1 + 'Valor De Variable' 'Nombre De Variable' := T2	'Nombre de variable' dw 'Valor de Variable'
T1 := 'Nombre de variable'	'Nombre de variable' dw 0
<b>ESTRUCTURA DE ASIGNACIONES</b>	
<b>Tipo Entero</b>	
T1 := 'Temporal de la variable' + 'Valor De Variable' T2 := 1 - 2 'Nombre De Variable' := T1 * T2	Negativo dw "-" MOV AX,'Nombre de Variable' MUL Negativo MOV 'Nombre de Variable',AX
T1 := 'Temporal de la variable' + 'Valor De Variable' 'Nombre De Variable' := T1	ADD 'Nombre De Variable','Valor De Variable'
<b>Tipo Decimal</b>	
T1 := 'Temporal de la variable' + 'Valor De Variable' T2 := 1 - 2 'Nombre De Variable' := T1 * T2	Negativo dw "-" MOV AX,'Nombre de Variable' MUL Negativo MOV 'Nombre de Variable',AX
T1 := 'Temporal de la variable' + 'Valor De Variable' 'Nombre De Variable' := T1	ADD 'Nombre De Variable','Valor De Variable'
<b>Tipo Cadena</b>	
T1 := 'Temporal de la variable' + 'Valor De Variable' 'Nombre De Variable' := T1	ADD 'Nombre De Variable','Valor De Variable'
<b>Tipo Figura</b>	
T1 := 'Temporal de la variable' + 'Valor De Variable' 'Nombre De Variable' := T1	ADD 'Nombre De Variable','Valor De Variable'
<b>Tipo Color</b>	
T1 := 'Temporal de la variable' + 'Valor De Variable' 'Nombre De Variable' := T1	ADD 'Nombre De Variable','Valor De Variable'
<b>ESTRUCTURA DE FUNCIONES</b>	
<b>De Sumar</b>	
T1 := 'Variable 1' + 'Variable 2'	MOV AX,'Variable 1'

T2 := T1 + 'Variable 3' T3 := T2 + 'Variable 4' T4 := T3 + 'Variable 5' 'Variable resultado' := T4	MOV BX,'Variable 2' ADD AX,BX MOV BX,'Variable 3' ADD AX,BX MOV BX,'Variable 4' ADD AX,BX MOV BX,'Variable 5' ADD AX,BX MOV 'Variable resultado',AX
<b>De Restar</b>	
T1 := 'Variable 1' - 'Variable 2' T2 := T1 - 'Variable 3' T3 := T2 - 'Variable 4' T4 := T3 - 'Variable 5' 'Variable resultado' := T4	MOV AX,'Variable 1' MOV BX,'Variable 2' SUB AX,BX MOV BX,'Variable 3' SUB AX,BX MOV BX,'Variable 4' SUB AX,BX MOV BX,'Variable 5' SUB AX,BX MOV 'Variable resultado',AX
<b>De Multiplicar</b>	
T1 := 'Variable 1' * 'Variable 2' T2 := T1 * 'Variable 3' T3 := T2 * 'Variable 4' T4 := T3 * 'Variable 5' 'Variable resultado' := T4	MOV AX,'Variable 1' MOV BX,'Variable 2' MUL BX MOV BX,'Variable 3' MUL BX MOV BX,'Variable 4' MUL BX MOV BX,'Variable 5' MUL BX MOV 'Variable resultado',AX
<b>De Dividir</b>	
T1 := 'Variable 1' / 'Variable 2' T2 := T1 / 'Variable 3' T3 := T2 / 'Variable 4' T4 := T3 / 'Variable 5' 'Variable resultado' := T4	MOV AX,'Variable 1' MOV BX,'Variable 2' DIV BX MOV BX,'Variable 3' DIV BX MOV BX,'Variable 4' DIV BX MOV BX,'Variable 5' DIV BX MOV 'Variable resultado',AX
<b>De Mostrar</b>	
En caso de que la variable resultado sea decimal	
T1 := 'Variable Cadena' + 'Variable Resultado' T2 := 'Variable Figura 1' + 'Variable Color 1'	MOV AX,'Variable Cadena'

T3 := 'Variable Figura 2' + 'Variable Color 2' T4 := T2 + T3 T5 := 'Variable Figura 3' + 'Variable Color 3' T6 := 'Variable Figura 4' + 'Variable Color 4' T7 := T5 + T6 T8 := 'Variable Figura 5' + 'Variable Color 5' T9 := T4 + T7 T10 := T9 + T8 T11 := T1 + T10	ADD AX,'Variable Resultado' MOV BX,'Variable Figura 1' ADD BX,'Variable Color 1' MOV CX,'Variable Figura 2' ADD CX,'Variable Color 2' ADD BX,CX MOV CX,'Variable Figura 3' ADD CX,'Variable Color 3' MOV DX,'Variable Figura 4' ADD DX,'Variable Color 4' ADD CX,DX MOV DX,'Variable Figura 5' ADD DX,'Variable Color 5' ADD BX,CX ADD BX,DX ADD AX,BX
En caso de que la variable resultado sea entero	
T1 := 'Variable Cadena' + 'Variable Resultado' T2 := 'Variable Figura' + 'Variable Color' T3 := T1 + T2	MOV AX,'Variable Cadena' ADD AX,'Variable Resultado' MOV BX,'Variable Figura' ADD BX,'Variable Color' ADD AX,BX
<b>ESTRUCTURA DE CLASES</b>	
<b>De Clase</b>	
T1 := '&Nombre Del Método' '&Nombre Del Método' := T1	'&Nombre Del Método':

### 7.3 Resultados

Las siguientes imágenes corresponden a un código de prueba en el lenguaje de programación MATH3C, ejecutado en el compilador del mismo nombre. Este ejemplo, se presentan los resultados del análisis (léxico, sintáctico y semántico) así como también la generación de código intermedio mediante tripletas y cuádruplos.

En dicho ejemplo se puede apreciar cómo se generan las cuádruplas para las 4 operaciones aritméticas básicas en donde se contemplan 2 operandos del tipo entero.

```

1  Metodo &Prueba{
2      Entero n1 = + 5;
3      Entero n2 = + 8;
4      Resultado #res1;
5      Resultado #res2;
6      Resultado #res3;
7      Resultado #res4;
8      Sumar(n1,n2)<
9          #res1
10     >;
11     Restar(n1,n2)<
12         #res2
13     >;
14     Multiplicar(n1,n2)<
15         #res3
16     >;
17     Dividir(n1,n2)<
18         #res4
19     >;
20 }
    
```

*Ilustración . Resultados del código de ejemplo*

Token	Lexema	[Fila,Columna]
Palabra_Reservada_22	Metodo	[Linea 1, Columna 1]
Identificador_Metodo	&Prueba	[Linea 1, Columna 8]
Signo_De_Agrupacion_5	{	[Linea 1, Columna 15]
Palabra_Reservada_6	Entero	[Linea 2, Columna 2]
Identificador_Variable	n1	[Linea 2, Columna 9]
Operador_De_Asignacion	=	[Linea 2, Columna 12]
Operador_Aritmetico_Mas	+	[Linea 2, Columna 14]
Numero_Entero	5	[Linea 2, Columna 16]
Delimitador_De_Sentenci...	;	[Linea 2, Columna 17]
Palabra_Reservada_6	Entero	[Linea 3, Columna 2]
Identificador_Variable	n2	[Linea 3, Columna 9]
Operador_De_Asignacion	=	[Linea 3, Columna 12]
Operador_Aritmetico_Mas	+	[Linea 3, Columna 14]
Numero_Entero	8	[Linea 3, Columna 16]
Delimitador_De_Sentenci...	;	[Linea 3, Columna 17]
Palabra_Reservada_8	Resultado	[Linea 4, Columna 2]
Identificador_Resultado	#res1	[Linea 4, Columna 12]
Delimitador_De_Sentenci...	;	[Linea 4, Columna 17]
Palabra_Reservada_8	Resultado	[Linea 5, Columna 2]
Identificador_Resultado	#res2	[Linea 5, Columna 12]

*Ilustración . Resultados, léxico y sintáctico.*

Compilación terminada...

La compilacion termino sin errores...

```

--> Se ha creado una Clase llamada &Prueba.....
--> Se ha declarado una variable de tipo Entero llamada n1 con valor +5.....
--> Se ha declarado una variable de tipo Entero llamada n2 con valor +8.....
--> Se ha declarado una variable de tipo Resultado llamada #res1.....
--> Se ha declarado una variable de tipo Resultado llamada #res2.....
--> Se ha declarado una variable de tipo Resultado llamada #res3.....
--> Se ha declarado una variable de tipo Resultado llamada #res4.....
    
```

*Ilustración . Resultados, semántico*



Tokens	Cuadрупlos	Tripletas	Ensamblador
Operador	Argumento 1	Argumento 2	Resultado
=	&Prueba		T1
=	T1		&Prueba
=	n1		T2
+	T2	5	T3
=	T3		n1
=	n2		T4
+	T4	8	T5
=	T5		n2
=	#res1		T6
=	#res2		T7
=	#res3		T8
=	#res4		T9
+	n1	n2	T10
=	T10		#res1
-	n1	n2	T11
=	T11		#res2
*	n1	n2	T12
=	T12		#res3
/	n1	n2	T13
=	T13		#res4

Ilustración . Resultados, cuadрупlos

Tokens	Cuadрупlos	Tripletas	Ensamblador
T1 := &Prueba			
&Prueba := T1			
T2 := n1			
T3 := T2 + 5			
n1 := T3			
T4 := n2			
T5 := T4 + 8			
n2 := T5			
T6 := #res1			
T7 := #res2			
T8 := #res3			
T9 := #res4			
T10 := n1 + n2			
#res1 := T10			
T11 := n1 - n2			
#res2 := T11			
T12 := n1 * n2			
#res3 := T12			
T13 := n1 / n2			
#res4 := T13			

Ilustración . Resultados, tripletas

Tokens
Cuadрупlos
Tripletas
Ensamblador

Inicio:

```

MOV AX,@DATA
MOV DS,AX
;Inicio de las estructuras de Metodos,Asignacion
&Prueba:
MOV AX,n1
MOV BX,n2
ADD AX,BX
MOV #res1,AX
MOV AX,n1
MOV BX,n2
SUB AX,BX
MOV #res2,AX
MOV AX,n1
MOV BX,n2
MUL BX
MOV #res3,AX
MOV AX,n1
MOV BX,n2
DIV BX
MOV #res4,AX
;Fin de las estructuras de Metodos,Asignacion
Fin:
MOV AX,4C00H
INT 21H
END

```

Zoom



Guardar
Guardar Como
Abrir

Ilustración . Resultados, Ensamblador

## Apéndice A. Las que siempre son nombradas

### A.1 Alfabeto

<b>Letra</b>	a,b,c,d,e,f,g,h,i,j,k,l,m,n,ñ,o,p,q,r,s,t,u,v,w,x,y,z,A,B,C,D,E,F,G,H,I,J,K,L,M,N,Ñ,O,P,Q,R,S,T,U,V,W,X,Y,Z
<b>Dígito</b>	0,1,2,3,4,5,6,7,8,9
<b>Símbolo</b>	. , = espacio ( ) < > # { } + - ; : &
<b>Comilla</b>	‘
<b>Gato</b>	#
<b>Ampersand</b>	&
<b>Agrupación</b>	(){}<>
<b>Punto</b>	.
<b>Coma</b>	,
<b>Asignación</b>	=
<b>Delimitador</b>	;
<b>Operador</b>	+ -

#### A.1.1 Identificadores

Son las variables, nombres de métodos, resultados y cadenas que se formarán en nuestro lenguaje, las cuales se forman de la siguiente manera:

**Variables:** Palabras formadas por una letra seguido de 0 o más letras o dígitos

Variable = letra (letra U dígito)\*

**Cadenas:** Palabras formadas por comilla seguido de letras o dígitos seguido de 0 o mas letras o dígitos o símbolos y finaliza con comilla

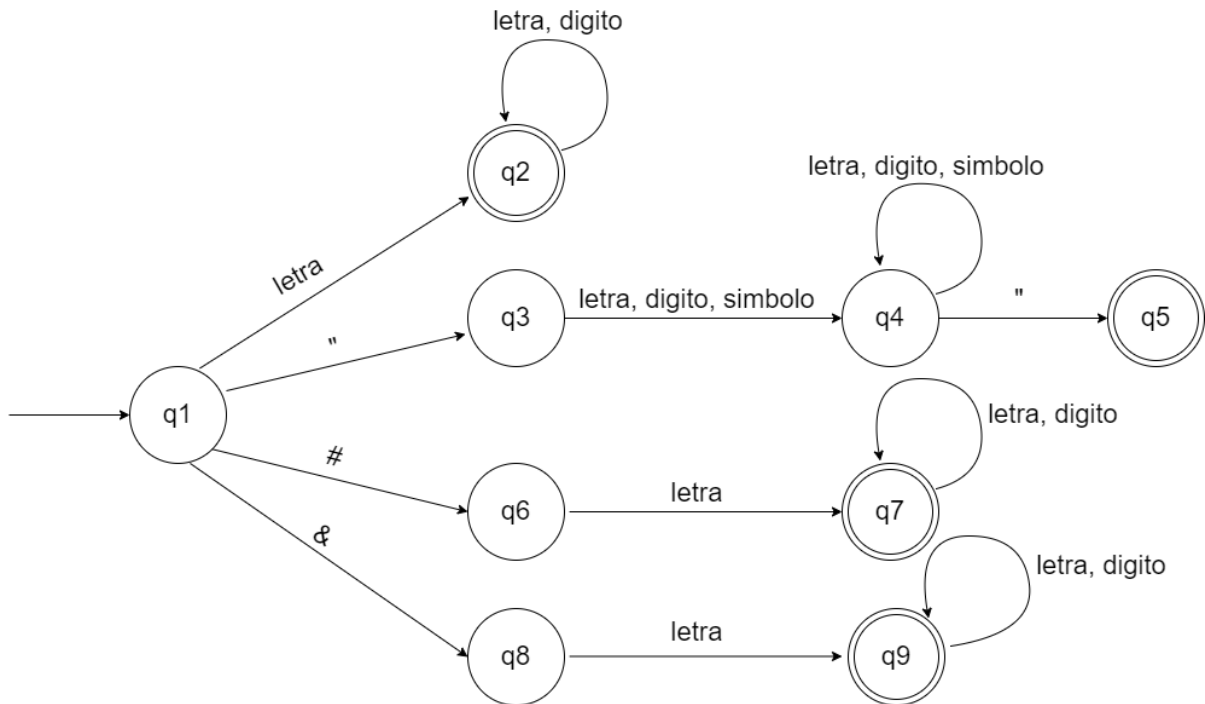
Cadena = comilla (letra U dígito U espacio) (letra U dígito U símbolo)\* comilla

**Resultados:** Palabras formadas por gato seguido de 1 letra seguido de 0 o más letras o dígitos

Resultado = gato letra (letra U dígito)\*

**Metodos:** Palabras formadas por Ampersand seguido de 1 letra seguido de 0 o más letras o dígitos

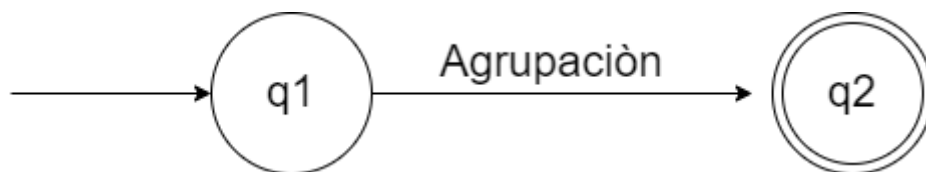
Método = ampersand letra (letra U dígito)\*



### A.1.2 Signos de agrupación

Es un grupo de símbolos que nos ayudarán a agrupar información dentro de los mismos

De los cuales los símbolos son los de agrupación



### A.1.3 Números

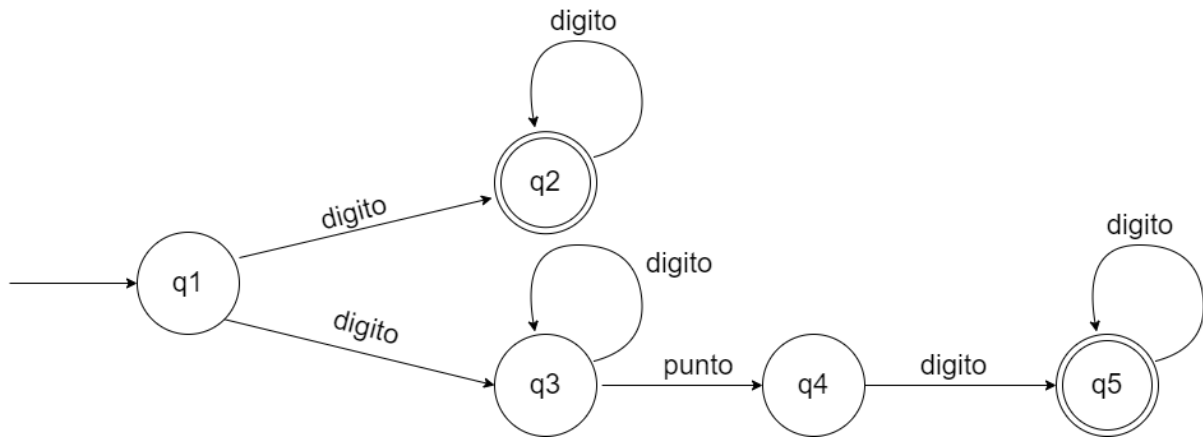
Son aquella combinación de dígitos que se formaran en nuestro lenguaje, de tal manera que contamos con los enteros y decimales en nuestro lenguaje

**Enteros:** Números formados por 1 dígito seguido de 0 o más dígitos

Entero = dígito (dígito)\*

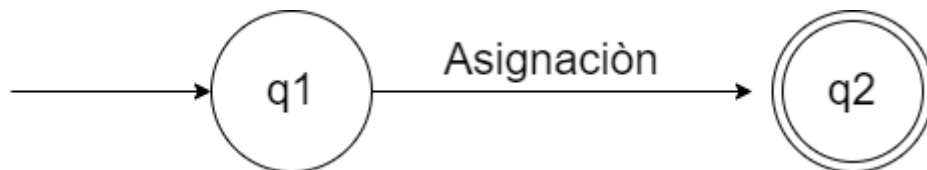
**Decimales:** Números formados por 1 dígito seguido de 0 o más dígitos, seguido de punto, seguido de uno o más dígitos

Decimal = dígito (dígito)\* punto (dígito)+



#### A.1.4 Operador de asignación

Es aquel símbolo que le dará valor a las variables, el símbolo es el de asignación



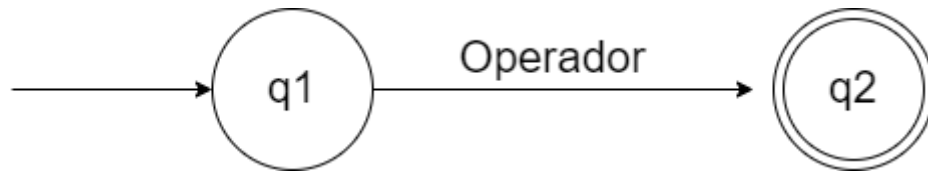
#### A.1.5 Delimitador de sentencia

Este símbolo nos ayuda a delimitar las sentencias en el lenguaje y está representado por el símbolo delimitador



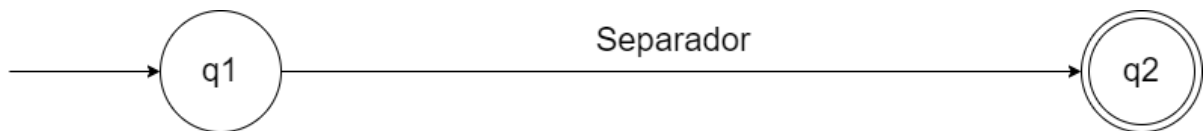
#### A.1.6 Operadores aritméticos

Es aquel símbolo que nos ayudará a darle un estado a los números, este se representa por el símbolo operador.



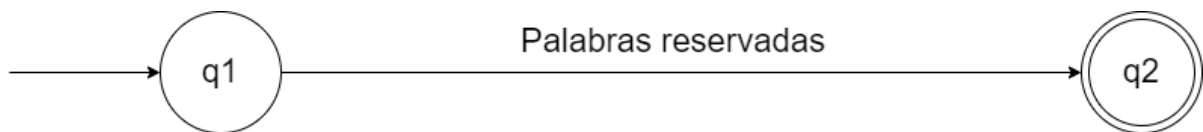
### A.1.7 Separador

Es aquel símbolo que nos ayudará a separar las variables dentro de una función, y está representado por el símbolo coma

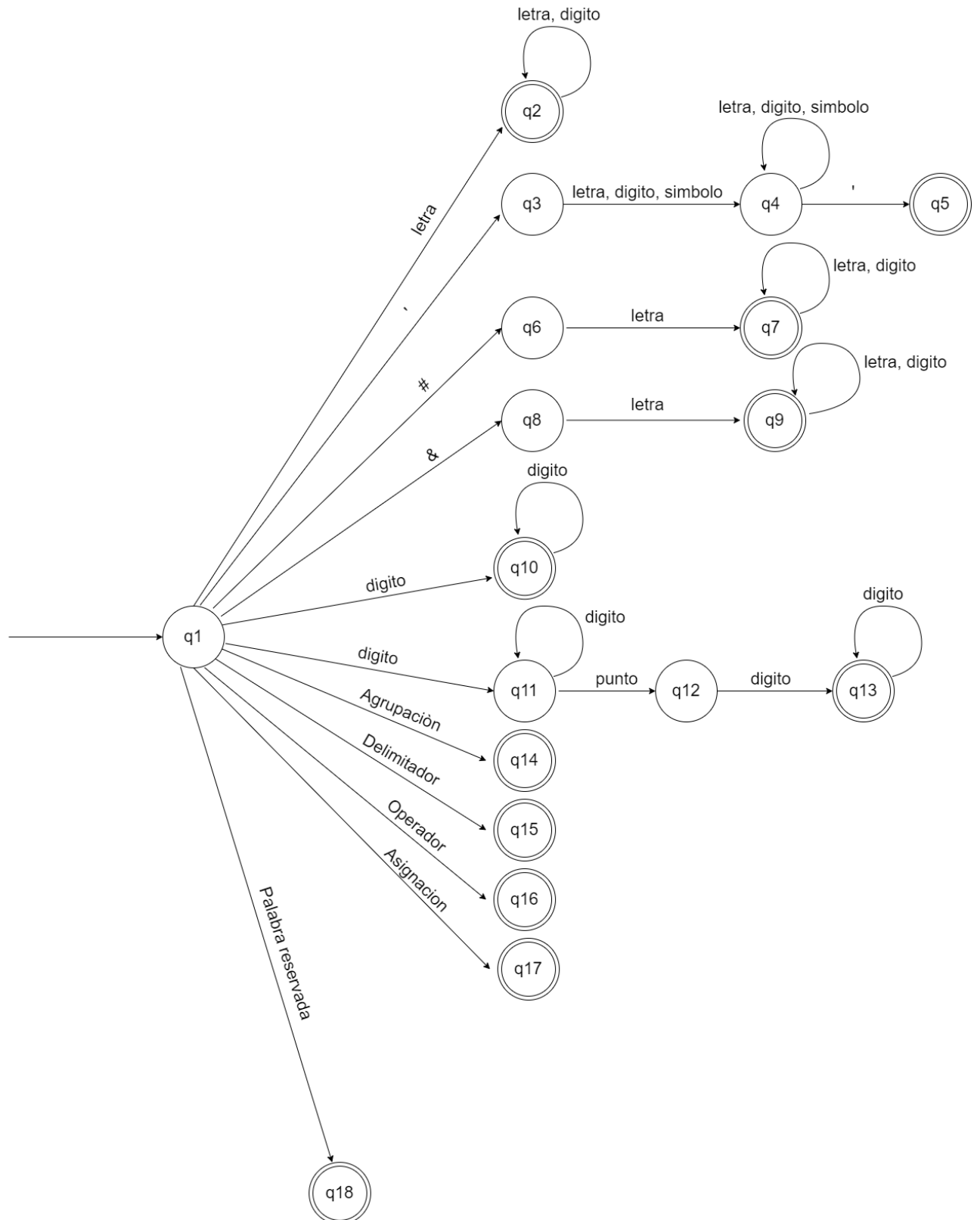


### A.1.8 Palabras reservadas

Sumar, Restar, Multiplicar, Dividir, Entero, Decimal, Resultado, Cadena, Figura, Color, Mostrar, Rojo, Azul, Verde, Blanco, Negro, Cuadrado, Triángulo, Rectángulo, Rombo, Circulo, Metodo



El autómata del lenguaje completo es el siguiente:



## A.2 Tabla de tokens

Componente léxico	Lexemas ejemplo	Descripción del patrón
Identificador	Arbol1, Boton7, Valor4	letra (letra U dígito)*
Identificador	'Una cadena', 'El valor '	comilla (letra U dígito U símbolo)+ comilla
Identificador	#Resultado, #Variable	gato letra (letra U dígito)*
Identificador	&Arbol, &Volcan	ampersand letra (letra U dígito)*
Signo de agrupación	(, ), {, }	agrupación
Número	23, 45, 12, 56	dígito (dígito)*
Número	48.23, 38.12, 3.56, 47.12	dígito (dígito)* . (dígito)+
Operador de asignación	=	asignación
Signo delimitador de sentencia	;	delimitador
Operador aritmético	+, -	operador
Separador	,	separador
Palabra reservada	Sumar, Restar, Decimal, Cadena, Cuadrado	Sumar U Restar U Multiplicar U Dividir U Entero U Decimal U Resultado U Cadena U Figura U Color U Mostrar U Rojo U Azul U Verde U Blanco U Negro U Cuadrado U Triángulo U Rectángulo U Rombo U Circulo U Metodo

## A.3 Tipos de datos

Tipo de dato	Descripción
Entero	Tipo de dato que sirve para almacenar un número del rango de -100,000 a +100,000
Decimal	Tipo de dato que sirve para almacenar un número fraccionario del rango de -99.9999 a +99.9999

Resultado	Tipo de dato que almacena un espacio para mostrar el resultado de una operación aritmética de tipo Entero o Decimal, de la cual en el Entero tiene un rango de valores de -200,000 a +200,000 y en el decimal es de -199.9999 a +199.9999
Cadena	Tipo de dato que sirve para almacenar caracteres definidos por el usuario
Figura	Es el tipo de dato que mostrará una figura de acuerdo a un número del cual solo están disponibles las siguientes figuras: Cuadrado, Triángulo, Rectángulo, Rombo, Círculo
Color	Es aquel tipo de dato que sirve para dar color al tipo de dato figura, del cual solo se puede definir de la siguiente manera: Rojo, Azul, Verde, Blanco y Negro

## A.4 Errores

### A.4.1 Errores léxicos

Generales	
Nombre del error	Origen del error
Error 0	El carácter no pertenece al alfabeto del lenguaje
Identificadores	
Nombre del error	Origen del error
Error 1	La variable no inicia con una letra
Error 2	Un carácter del nombre de la variable no pertenece a un símbolo válido
Error 3	Un carácter de la cadena no pertenece a un símbolo válido
Error 4	No hay ningún elemento entre ' '
Error 5	Un carácter del nombre del resultado no pertenece a un símbolo válido
Error 6	No hay por lo menos una letra después de #
Error 7	Un carácter del nombre del método no pertenece a un símbolo válido
Error 8	No Hay por lo menos una letra después de &
Números	
Nombre del error	Origen del error



Error 9	El número contiene un caracter no valido o cuenta con más de un punto
Error 10	No contiene un dígito como mínimo después del punto en el número
Error 11	El número inicia con punto decimal
Error 12	El número inicia con punto decimal y tiene más de uno
Error 13	El número no inicia con un dígito

#### A.4.2 Errores sintácticos

Declarativos	
Nombre del error	Origen del error
Error 14	Falta la palabra reservada en la declaración
Error 15	Falta la variable en la declaración
Error 16	Falta el = en la declaración
Error 17	Falta el ; en la declaración
Error 18	Falta el valor de la variable en la declaración
Error 19	Falta signo “+” o “-” antes del número en la declaración
Error 20	Faltan varios elementos en la declaración
Error 21	El tipo de dato no se encuentra en un método declarado o de asignación
Error 22	La variable no se encuentra en un método declarativo, de asignación o de función
Error 23	El signo de “+” o “-” no se encuentran en un método declarativo o de asignación
Error 24	El = no se encuentran en un método declarativo o de asignación
Error 25	El ; no se encuentran en un método declarativo , de asignación o de función
Error 26	La palabra reservada no se encuentran en un método declarativo
Asignación	
Nombre del error	Origen del error
Error 27	Falta la variable en la asignación

Error 28	Falta el = en la asignación
Error 29	Falta el nuevo valor de la variable en la asignación
Error 30	Falta el ; en la asignación
Error 31	El signo de “+” o “-” no se encuentran en la asignación
Error 32	Faltan varios elementos en la asignación
<b>Funciones</b>	
<b>Nombre del error</b>	<b>Origen del error</b>
Error 33	Falta la palabra reservada en la función
Error 34	Falta el paréntesis al inicio “(“ o al final “)” en la función
Error 35	Variables fuera de los rangos en la función (Min. 2 y Máx. 5)
Error 36	Falta signo “<” al inicio o al final “>” en la función
Error 37	Falta la variable tipo resultado en la función
Error 38	Falta ; al final de la función
Error 39	Faltan varios elementos en la función
Error 40	La , no se encuentra en una función
Error 41	El “(”, “)”, “<” o “>” no se encuentran en una función
Error 42	La palabra reservada no se encuentra en una función
Error 43	Variables fuera de los rangos en la función (Min. 4 y Máx. 4)
Error 44	Variables fuera de los rangos en la función (Min. 12 y Max. 12)
Error 45	Falta signo “<” al inicio y al final “>” en la función
Error 46	Falta el paréntesis al inicio “(“ y al final “)” en la función
<b>Clases</b>	
<b>Nombre del error</b>	<b>Origen del error</b>
Error 47	Falta la palabra reservada en la clase
Error 48	Falta el nombre de la clase
Error 49	Falta signo “{“ al inicio o al final “}” en la clase
Error 50	Falta signo “{“ al inicio y al final “}” en la clase
Error 51	Las funciones, las asignaciones o declaraciones no pueden estar fuera de una clase

Error 52	La palabra reservada no se encuentra en una clase
Error 53	La variable no se encuentra en una clase
Error 54	El signo "{" o "}" no se encuentra en una clase

## Apéndice B. Ejemplo funcional

El ejemplo presentado en este apéndice, se basa en la estructura presentada en la sección 2.3 de este mismo documento. Sirviendo de ejemplo para verificar el correcto funcionamiento del compilador.

### B.1 Método

Primeramente se colocarán las variables a emplear, que como bien sabemos las variables son aquellos objetos que serán modificados, con diversos propósitos, a lo largo del programa.

---

```

Metodo &Prueba{
    Entero num1 = - 100;
    Entero num2;
    Entero num5;
    Decimal num3 = - 10.61;
    Decimal num4;
    Resultado #res1;
    Resultado #res2;
    Resultado #res3;
    Resultado #res4;
    Cadena cad1 = 'Hola';
    Cadena cad4;
    Figura fig1 = Circulo;
    Figura fig2;
    Color col1 = Azul;
    Color col2;
    Color col3;
    num2 = + 300;
    num4 = + 10.45;
    cad4 = 'El resultado: ';
    fig2 = Circulo;
    col2 = Rojo;
    
```

*Ilustración . Ejemplo a compilar*

Para este ejemplo, contamos con dieciséis variables, tres de tipo entero, dos de tipo decimal, cuatro de tipo resultado ya que en estas variables se almacenará el resultado de las operaciones a realizar, dos variables de tipo cadena, dos de tipo figura para representar

el resultado de la operación con diversas figuras, siendo de este modo más entendible por los niños, y tres variables de tipo color. Algunas de las variables mencionadas anteriormente son declaradas sin asignarles un valor y otras con valor asignado.

Posteriormente se programan las funciones de operaciones a realizar.

```

23      Sumar(num2,num2,num2,num2,num2)<
24          #res1
25      >;
26      Restar(num3,num3)<
27          #res2
28      >;
29      Multiplicar(num4,num3)<
30          #res3
31      >;
32      Dividir(num3,num4)<
33          #res4
34      >;
35      Mostrar(cad4,#res3,fig1,col1,fig2,col2,fig1,col1,fig2,col2,fig1,col1);
36  }
```

Para este ejemplo, se utilizaron las 4 operaciones que maneja el compilador y el método de mostrar.

## B.2 Compilación

Una vez terminado con la codificación podemos compilarlo dando click en el siguiente botón.



*Ilustración .Botón Ejecutar*

Nos generará las siguientes salidas en la sección de resultados de compilación y las generaciones de la tabla de tokens, tabla de tripletas, tabla de cuádruplas y el código objeto.

## B.2.1 Resultados de compilación

Compilación terminada...

La compilación terminó sin errores...

```
--> Se ha creado una Clase llamada &Prueba.....
--> Se ha declarado una variable de tipo Entero llamada n1 con valor +5.....
--> Se ha declarado una variable de tipo Entero llamada n2 con valor +8.....
--> Se ha declarado una variable de tipo Resultado llamada #res1.....
--> Se ha declarado una variable de tipo Resultado llamada #res2.....
--> Se ha declarado una variable de tipo Resultado llamada #res3.....
--> Se ha declarado una variable de tipo Resultado llamada #res4.....
```

*Ilustración .Resultado de compilación 1*

```
--> Se ha declarado una variable de tipo Resultado llamada #res4.....
--> Se ha utilizado la función de Sumar, que tiene como parametros a:  n1 = +5 , n2 = +8
Donde el resultado de la operación es: 13 , y se guardó en: #res1.....
--> Se ha utilizado la función de Restar, que tiene como parametros a:  n1 = +5 , n2 = +8
Donde el resultado de la operación es: -3 , y se guardó en: #res2.....
--> Se ha utilizado la función de Multiplicar, que tiene como parametros a:  n1 = +5 , n2 = +8
Donde el resultado de la operación es: 40 , y se guardó en: #res3.....
--> Se ha utilizado la función de Dividir, que tiene como parametros a:  n1 = +5 , n2 = +8
Donde el resultado de la operación es: 0 , y se guardó en: #res4.....
```

*Ilustración .Resultado de compilación 2*

## B.2.2 Generación de la tabla de tokens

Tokens	Cuadros	Tripletas	Ensamblador
Token	Lexema	[Fila,Columna]	
Palabra_Reservada_22	Metodo	[Linea 1, Columna 1]	
Identificador_Metodo	&Prueba	[Linea 1, Columna 8]	
Signo_De_Agrupacion_5	{	[Linea 1, Columna 15]	
Palabra_Reservada_6	Entero	[Linea 2, Columna 2]	
Identificador_Variable	n1	[Linea 2, Columna 9]	
Operador_De_Asignacion	=	[Linea 2, Columna 12]	
Operador_Aritmetico_Mas	+	[Linea 2, Columna 14]	
Numero_Entero	5	[Linea 2, Columna 16]	
Delimitador_De_Sentenci...	;	[Linea 2, Columna 17]	
Palabra_Reservada_6	Entero	[Linea 3, Columna 2]	
Identificador_Variable	n2	[Linea 3, Columna 9]	
Operador_De_Asignacion	=	[Linea 3, Columna 12]	
Operador_Aritmetico_Mas	+	[Linea 3, Columna 14]	
Numero_Entero	8	[Linea 3, Columna 16]	
Delimitador_De_Sentenci...	;	[Linea 3, Columna 17]	
Palabra_Reservada_8	Resultado	[Linea 4, Columna 2]	
Identificador_Resultado	#res1	[Linea 4, Columna 12]	
Delimitador_De_Sentenci...	;	[Linea 4, Columna 17]	
Palabra_Reservada_8	Resultado	[Linea 5, Columna 2]	
Identificador_Resultado	#res2	[Linea 5, Columna 12]	

*Ilustración .Resultado de tokens*

### B.2.3 Generación de tripletas

Tokens	Cuádruplos	Tripletas	Ensamblador
<pre> T1 := &amp;Prueba &amp;Prueba := T1 T2 := n1 T3 := T2 + 5 n1 := T3 T4 := n2 T5 := T4 + 8 n2 := T5 T6 := #res1 T7 := #res2 T8 := #res3 T9 := #res4 T10 := n1 + n2 #res1 := T10 T11 := n1 - n2 #res2 := T11 T12 := n1 * n2 #res3 := T12 T13 := n1 / n2 #res4 := T13                     </pre>			

*Ilustración .Resultado de tripletas*

### B.2.4 Generación de cuádruplos

Tokens	Cuádruplos	Tripletas	Ensamblador
Operador	Argumento 1	Argumento 2	Resultado
=	&Prueba		T1
=	T1		&Prueba
=	n1		T2
+	T2	5	T3
=	T3		n1
=	n2		T4
+	T4	8	T5
=	T5		n2
=	#res1		T6
=	#res2		T7
=	#res3		T8
=	#res4		T9
+	n1	n2	T10
=	T10		#res1
-	n1	n2	T11
=	T11		#res2
*	n1	n2	T12
=	T12		#res3
/	n1	n2	T13
=	T13		#res4

*Ilustración .Resultado de cuádruplos*

## B.2.5 Generación de código objeto

```

Inicio:
    MOV AX,@DATA
    MOV DS,AX
    ;Inicio de las estructuras de Metodos,Asignaci
&Prueba:
    MOV AX,n1
    MOV BX,n2
    ADD AX,BX
    MOV #res1,AX
    MOV AX,n1
    MOV BX,n2
    SUB AX,BX
    MOV #res2,AX
    MOV AX,n1
    MOV BX,n2
    MUL BX
    MOV #res3,AX
    MOV AX,n1
    MOV BX,n2
    DIV BX
    MOV #res4,AX
    ;Fin de las estructuras de Metodos,Asignacion
Fin:
    MOV AX,4C00H
    INT 21H
END
    
```

*Ilustración .Resultado de código objeto*

## Apéndice C. Para descargar e instalar

Para que el programa realizado funcione correctamente se requieren algunas librerías y un software los cuales son necesarios, por lo tanto se pone a disposición del programador los enlaces para la descarga de los requerimientos antes mencionados.

Requerimiento	Enlace de Descarga
NetBeans IDE 15	<a href="https://www.apache.org/dyn/closer.cgi/netbeans/netbeans-installers/15/Apache-NetBeans-15-bin-windows-x64.exe">https://www.apache.org/dyn/closer.cgi/netbeans/netbeans-installers/15/Apache-NetBeans-15-bin-windows-x64.exe</a>
CompilerTools 2.3.7	<a href="https://github.com/YisusTecFBI/CompilerTools-2.3.7.git">https://github.com/YisusTecFBI/CompilerTools-2.3.7.git</a>
flatlaf 2.2	<a href="https://github.com/JFormDesigner/FlatLaf.git">https://github.com/JFormDesigner/FlatLaf.git</a>
jflex 1.8.2	<a href="https://www.jflex.de/">https://www.jflex.de/</a>
JDK 18.0.2.1	<a href="https://www.oracle.com/java/technologies/java-se/jdk18-archive-downloads.html">https://www.oracle.com/java/technologies/java-se/jdk18-archive-downloads.html</a>