

# 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>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	8
1.4 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
3.3 Vista general, especificaciones	11
3.3.1 Menú	11
1. Archivo	11
2. Ejemplos	12
3. Acerca de	12
4. Tablas	12
5. Funcionamiento	13
6. Código Intermedio	13
7. Código Objeto	13
8. Ayuda	13
3.3.2 Botones	14
1. Nuevo	14
2. Abrir	14
3. Guardar	15
4. Guardar como	15
6. Compilar	15
7. Ejecutar	15
8. Aumentar zoom	16
9. Disminuir zoom	16
3.3.3 Área de programación	16
3.3.4 Área de compilación	17
3.3.5 Área de resultados	18
3.3.5.1 Tabla de Tokens	18
3.3.5.2 Tabla de Cuadрупlos	18



3.3.5.3 Tabla de Tripletas	19
3.3.5.4 Tabla de Ensamblador	19
<b>CAPÍTULO 4. CRUD</b>	<b>21</b>
4.1 ¿Qué es el CRUD?	21
4.2 Crear un nuevo proyecto (y compilarlo)	21
4.2.1 Código	21
4.2.2 Compilación (Análisis)	21
4.2.3 Generación de Código Intermedio	21
4.3 Guardar un Proyecto	21
4.4 Abrir un proyecto Existente	21
4.5 Modificar un proyecto Existente	21
<b>Apéndice A. Las que siempre son nombradas</b>	<b>21</b>
A.1 Alfabeto	21
A.1.2 Números	22
A.1.3 Palabras reservadas	22
A.2 Tabla de tokens	23
A.3 Tipos de datos	23
<b>Apéndice B. Ejemplo funcional</b>	<b>24</b>
B.1 Método	24
B.2 Compilación	26
B.2.1 Resultados de compilación	26
B.2.2 Generación de la tabla de tokens	27
B.2.3 Generación de tripletas	27
B.2.4 Generación de cuádruplos	28
B.2.5 Generación de código objeto	28

## Prefacio

**Martin Ernesto Barrón Doroteo,**

**Jesús Alberto Partida Michel,**

**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 MATH3C 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 usuario 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 Ayuda adicional

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

## 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

Al ejecutar el compilador M4TH3C, este mostrará la siguiente pantalla.

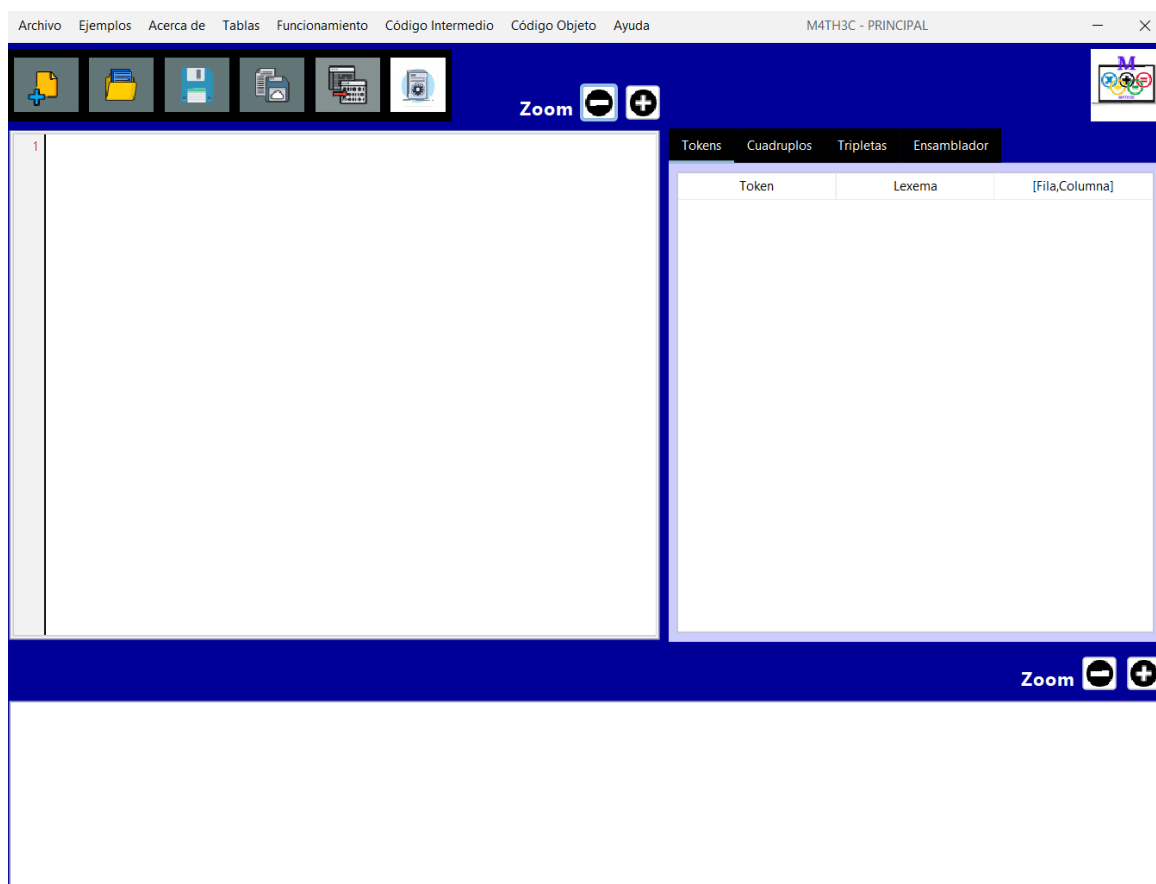


Figura 1. Vista general de M4TH3C

Esta cuenta con 5 secciones visibles, más una adicional que corresponde a las ventanas que determinados botones o acciones permiten visualizar.

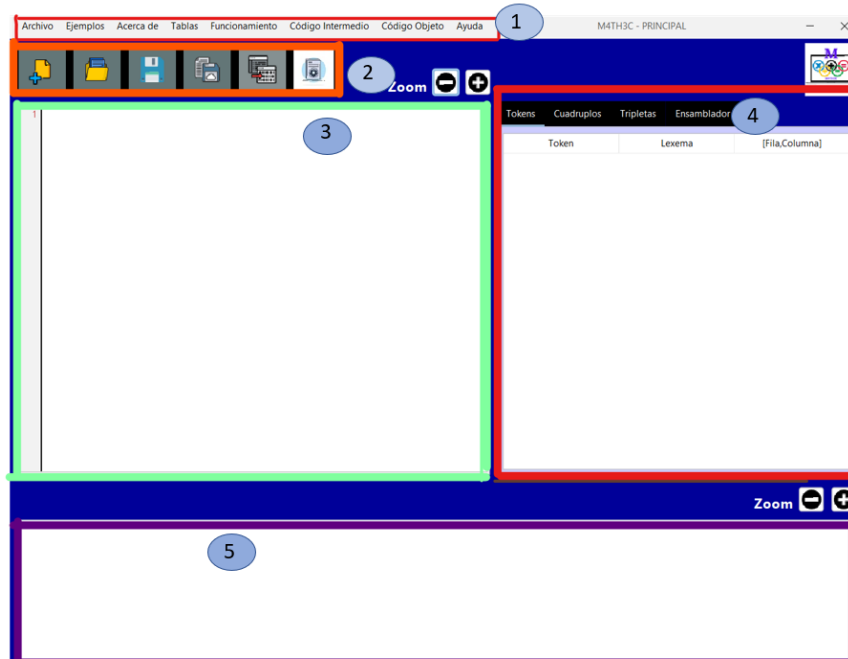


Figura 2. Vista general de MATH3C anotada

### 3.3 Vista general, especificaciones

#### 3.3.1 Menú

El menú de pestañas es la primer sección del entorno visual del compilador. Estas, corresponden a ocho botones en la parte superior izquierda de la pantalla principal, los cuales despliegan una serie de listas con botones de acciones, ya sea referentes al archivo que se va ejecutar o sobre la información de ayuda del compilador.

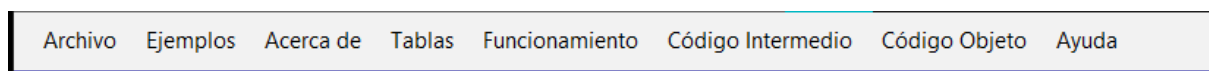


Figura 3. Sección 1, menú.

#### 1. Archivo

- A. Este botón despliega opciones sobre el archivo. Estas son: Nuevo, Abrir, Guardar, Guardar Como, Compilar y Ejecutar.

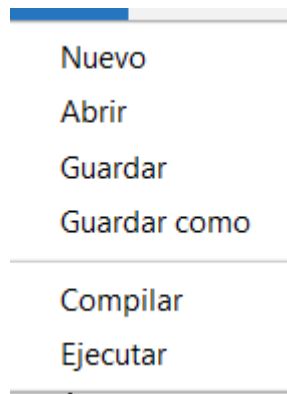


Figura 4. Botón Archivo

## 2. Ejemplos

- A. Este botón despliega opciones sobre programas de ejemplo. Estas son las siguientes:

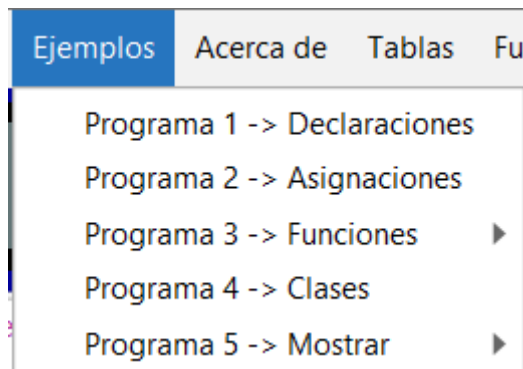


Figura 5. Botón Ejemplos

## 3. Acerca de

- a. Este botón despliega información respectiva sobre M4TH3C.

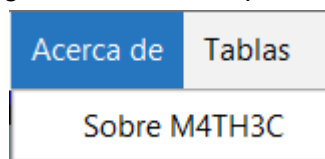
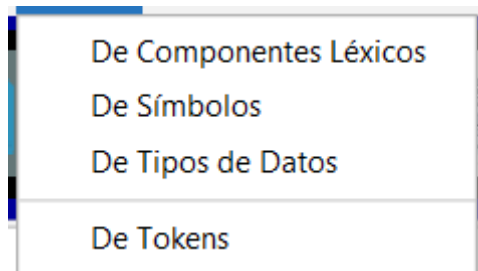


Figura 6. Botón Acerca de

## 4. Tablas

- a. Este botón despliega opciones sobre de apoyo a modo de tabla de los siguientes temas:



*Figura 7. Botón Tablas*

## 5. Funcionamiento

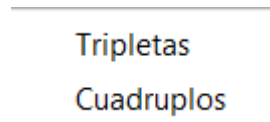
- a. Este botón despliega las siguientes opciones que al darle click nos aparecerá un mensaje que nos explicará información al respecto.

Declaraciones  
Asignaciones  
Funciones <Primera parte>  
Funciones <Segunda parte>  
Clases

*Figura 8. Botón Funcionamiento*

## 6. Código Intermedio

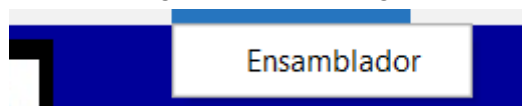
- a. Este botón despliega las opciones de tripletas y cuádruplos, que nos despliega la ventana con la tabla de la opción seleccionada.



*Figura 9. Botón Código intermedio*

## 7. Código Objeto

- a. Este botón despliega la opción de Ensamblador que nos permite ver en otra ventana el código ensamblador generado.



*Figura 10. Botón Código Objeto*

## 8. Ayuda

- a. Este botón despliega opciones de ayuda tales como los siguientes, que al darle click abrirán el documento pdf con la información que se seleccionó.

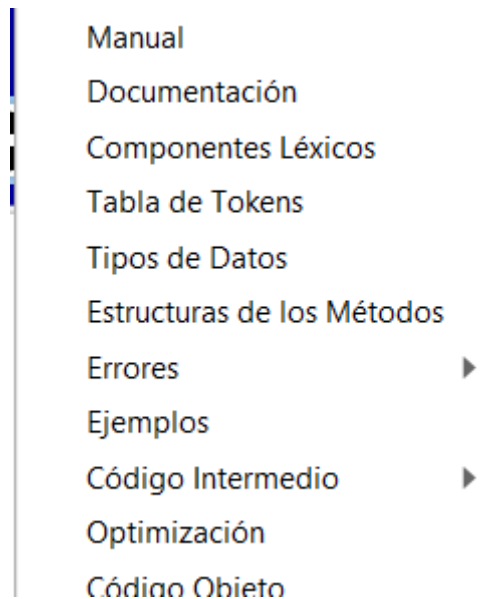


Figura 11. Botón Ayuda

### 3.3.2 Botones

El menú de botones corresponde a la segunda sección del entorno visual del compilador. Estos ocho botones, posicionados en la parte superior izquierda de la pantalla, debajo del menú anterior, permiten accesos directos a varias acciones presentadas en el menú de pestañas, además de los botones necesarios para la compilación del código creado por el usuario.

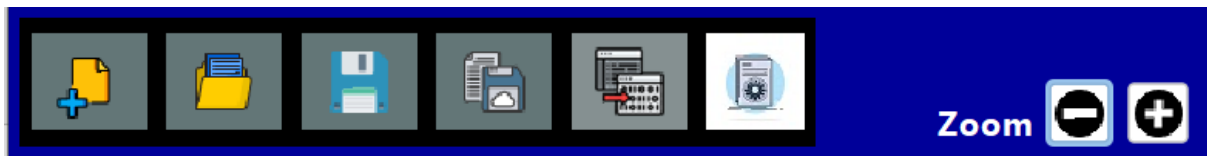


Figura 12. Sección 2, Botones

#### 1. Nuevo

- a. Este botón permite crear un nuevo archivo para un nuevo código de M4TH3C.



Figura 13. Botón Nuevo

#### 2. Abrir

- a. Este botón permite abrir un nuevo archivo para un nuevo código de M4TH3C.



*Figura 13. Botón Abrir*

### 3. Guardar

- a. Este botón permite guardar el archivo.



*Figura 14. Botón Guardar*

### 4. Guardar como

- a. Este botón permite guardar como el archivo que se está utilizando.



*Figura 15. Botón Guardar como*

### 6. Compilar

- a. Este botón permite compilar el archivo.



*Figura 16. Botón Compilar*

### 7. Ejecutar

- a. Este botón permite ejecutar el archivo.



*Figura 17. Botón Ejecutar*

## 8. Aumentar zoom

- a. Este botón permite aumentar el zoom del archivo que se está utilizando.



*Figura 18. Botón Aumentar Zoom*

## 9. Disminuir zoom

- a. Este botón permite disminuir el zoom del archivo que se está utilizando.



*Figura 19. Botón Disminuir Zoom*

### 3.3.3 Área de programación

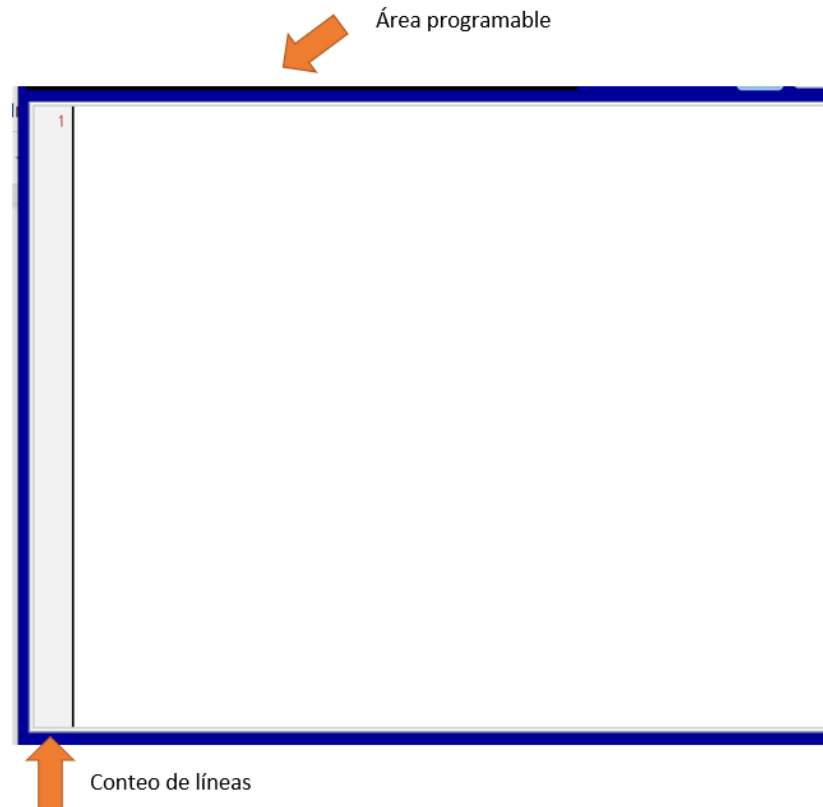
El área de programación, corresponde al tercer espacio del entorno visual del compilador. Se encuentra al centro a la izquierda de la pantalla y abarca la mitad de la pantalla. Aquí, es donde el programador escribe el código del programa que realiza.

```

1  Metodo &Declaraciones{
2
3      Entero num1 = + 500;
4      Entero num2;
5
6      Decimal num3 = + 30.40;
7      Decimal num4;
8
9      Resultado #res1;
10
11     Cadena cad1 = 'Prueba 1';
12     Cadena cad2;
13
14     Figura fig1 = Circulo;
15     Figura fig2;
16
17     Color col1 = Verde;
18     Color col2;
19
20 }
21 |
    
```

*Figura 20. Sección 3: Área de Programación*

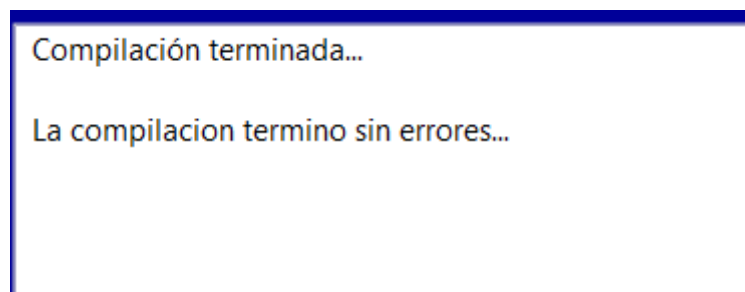
Se divide en dos partes, el conteo de líneas, en la parte lateral izquierda, y el área programable, en el resto de la sección.



*Figura 21. Área de Programación anotada*

### 3.3.4 Área de compilación

El área de compilación corresponde a la quinta sección del entorno visual del compilador. Se encuentra en la parte inferior del compilador, y es del largo de la pantalla. En esta sección, son mostrados los errores que pudiese presentar el código escrito en el área programable. Dichos errores describirán el tipo, la línea donde se encuentra y su descripción asociada, en caso de no existir error, solo se mostrará un mensaje diciendo que se compiló sin errores.



*Figura 22. Sección 5: Área de Compilación*



### 3.3.5 Área de resultados

El área de resultados corresponde a la cuarta sección del entorno visual del compilador. Se encuentra en la parte derecha del compilador, y abarca la mitad de la pantalla. En esta sección, son mostrados los resultados de la ejecución tales como la tabla de tokens, cuádruplos, tripletas y ensamblador.

#### 3.3.5.1 Tabla de Tokens

En esta, se muestran los tokens empleados en el código que se compilo y ejecuto.

Tokens	Cuádruplos	Tripletas	Ensamblador
Token	Lexema	[Fila,Columna]	
Palabra_Reservada_22	Metodo	[Linea 1, Columna 1]	
Identificador_Metodo	&Declaraciones	[Linea 1, Columna 8]	
Signo_De_Agrupacion_5	{	[Linea 1, Columna 22]	
Palabra_Reservada_6	Entero	[Linea 3, Columna 2]	
Identificador_Variable	num1	[Linea 3, Columna 9]	
Operador_De_Asignacion	=	[Linea 3, Columna 14]	
Operador_Aritmetico_Mas	+	[Linea 3, Columna 16]	
Numero_Entero	500	[Linea 3, Columna 18]	
Delimitador_De_Sentenci...	;	[Linea 3, Columna 21]	
Palabra_Reservada_6	Entero	[Linea 4, Columna 2]	
Identificador_Variable	num2	[Linea 4, Columna 9]	
Delimitador_De_Sentenci...	;	[Linea 4, Columna 13]	
Palabra_Reservada_7	Decimal	[Linea 6, Columna 2]	
Identificador_Variable	num3	[Linea 6, Columna 10]	
Operador_De_Asignacion	=	[Linea 6, Columna 15]	
Operador_Aritmetico_Mas	+	[Linea 6, Columna 17]	
Numero_Decimal	30.40	[Linea 6, Columna 19]	
Delimitador_De_Sentenci...	;	[Linea 6, Columna 24]	
Palabra_Reservada_7	Decimal	[Linea 7, Columna 2]	
Identificador_Variable	num4	[Linea 7, Columna 10]	

Figura 23. Sección 4: Tabla Tokens

#### 3.3.5.2 Tabla de Cuádruplos

En esta, se muestran los cuádruplos que se generaron en el código que se compilo y ejecuto.

Tokens	Cuadрупlos	Tripletas	Ensamblador
Operador	Argumento 1	Argumento 2	Resultado
=	&Declaraciones		T1
=	T1		&Declaraciones
=	num1		T2
+	T2	500	T3
=	T3		num1
=	num2		T4
=	num3		T5
+	T5	30.40	T6
=	T6		num3
=	num4		T7
=	#res1		T8
=	cad1		T9
+	T9	'Prueba 1'	T10
=	T10		cad1
=	cad2		T11
=	fig1		T12
+	T12	Circulo	T13
=	T13		fig1
=	fig2		T14
=	col1		T15

Figura 24. Sección 4: Tabla Cuadрупlos

### 3.3.5.3 Tabla de Tripletas

En esta, se muestran las tripletas que se generaron en el código que se compilo y ejecuto.

Tokens	Cuadрупlos	Tripletas	Ensa
<pre> T1 := &amp;Declaraciones &amp;Declaraciones := T1 T2 := num1 T3 := T2 + 500 num1 := T3 T4 := num2 T5 := num3 T6 := T5 + 30.40 num3 := T6 T7 := num4 T8 := #res1 T9 := cad1 T10 := T9 + 'Prueba 1' cad1 := T10 T11 := cad2 T12 := fig1 T13 := T12 + Circulo fig1 := T13 T14 := fig2 T15 := col1 T16 := T15 + Verde col1 := T16 T17 := col2                     </pre>			

Figura 25. Sección 4: Tabla Tripletas

### 3.3.5.4 Tabla de Ensamblador

En esta, se muestra el código ensamblador del código M4TH3C que se desarrolló.

```

*****
;*      LENGUAJES Y AUTOMATAS II
;*      COMPILADOR - M4THEC
;*      DESARROLLADORES:
;* NL - 14 MARLETT ALESSANDRA MARIN BAUT
;* NL - 21 JESUS ALBERTO PARTIDA MICHEL
;* NL - 22 JOSE DE JESUS RAMIREZ ORTEGA
;* NL - 1  MARTIN ERNESTO BARRON DOROTEC
*****
.model small
.stack
.data
    num1 dw "5","0","0"
    num2 dw 0,0,0,0,0
    num3 dw "3","0",".", "4","0"
    num4 dw 0,0,0,0,0,0
    res1 dw 0,0,0,0,0,0,0
    cad1 dw "P","r","u","e","b","a"," ", "1"
    fig1 dw "C","i","r","c","u","l","o"
    fig2 dw 0,0,0,0,0,0,0,0,0
    col1 dw "V","e","r","d","e"
    col2 dw 0,0,0,0,0,0
.code
Inicio:
    MOV AX,@DATA
    MOV DS,AX

```

Figura 26. Sección 4: Ensamblador 1

```

.code
Inicio:
    MOV AX,@DATA
    MOV DS,AX
;Inicio de las estructuras de Metodos,Asignaci
&Declaraciones:
;Fin de las estructuras de Metodos,Asignacion
Fin:
    MOV AX,4C00H
    INT 21H
END

```

Figura 27. Sección 4: Ensamblador 2

## CAPÍTULO 4. CRUD

### 4.1 ¿Qué es el CRUD?

CRUD (Create, Read, Update, Delete) es un acrónimo para las maneras en las que se puede operar sobre información almacenada. Es un nemónico para las cuatro funciones del almacenamiento persistente (Crear, Leer, Actualizar, Eliminar). CRUD usualmente se refiere a operaciones llevadas a cabo en una base de datos o un almacén de datos, pero también puede aplicar a funciones de un nivel superior de una aplicación como soft deletes donde la información no es realmente eliminada, si no que es marcada como eliminada a través de un estatus.

### 4.2 Crear un nuevo proyecto (y compilarlo)

Para programar en M4TH3C necesitamos trabajar en un proyecto, por defecto, al abrir el programa, estamos ya en un proyecto vacío, en él podemos escribir código y, posteriormente, guardar el mismo, o, en su defecto, podemos cargar un proyecto que tengamos guardado en nuestra computadora.

#### 4.2.1 Código

Para crear un nuevo proyecto contamos con dos opciones, crearlo desde el menú Archivo, o desde el ícono de Nuevo Proyecto.

Para crear un nuevo proyecto desde el menú Archivo, primero entramos al menú archivo, el cual se encuentra en la esquina superior izquierda.

#### 4.2.2 Compilación (Análisis)

#### 4.2.3 Generación de Código Intermedio

### 4.3 Guardar un Proyecto

### 4.4 Abrir un proyecto Existente

### 4.5 Modificar un proyecto Existente

## 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.1.2 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.1.3 Palabras reservadas

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

## 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, Circulo
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

## 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;
}
```

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)<
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 compilar*

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 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 .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	Cuadros	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,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
    
```

*Ilustración .Resultado de código objeto*

