

# PROYECTO MINOR C



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
Facultad de Ingeniería

## MANUAL DE USUARIO

Luis Alfonso Ordoñez Carrillo  
201603127

Proyecto No.2 Organización de lenguajes y compiladores 2

# Contenido

Introducción .....	2
Objetivos .....	3
General.....	3
Específicos .....	3
Manual de Usuario .....	4
1. Barra de menú.....	4
2. Barra de herramientas .....	8
3. Área de trabajo.....	8
4. Consola .....	9
5. Visor de Imágenes.....	10
Debugeo .....	10
Traducción .....	12
Menú ayuda .....	13
Ver manual de usuario .....	13
Ver manual Técnico .....	13
Acerca de.....	14

# Introducción

La finalidad de este manual enseñar al usuario el funcionamiento del software mediante varias imágenes que le ayudaran con cada uno de los comandos que este programa maneja para la administración de discos.

Augus es un lenguaje de programación, basado en PHP y en MIPS. Su principal funcionalidad es ser un lenguaje intermedio, ni de alto nivel como PHP ni de bajo nivel como el lenguaje ensamblador de MIPS.

El lenguaje tiene dos restricciones: la primera, es que cada instrucción es una operación simple; y la segunda, es que en cada instrucción hay un máximo de dos operandos y su asignación (si la hubiera).

Es un lenguaje débilmente tipado, sin embargo, si se reconocen cuatro tipos de datos no explícitos: entero, punto flotante, cadena de caracteres y arreglo.

Para manejar el flujo de control se proporciona la declaración de etiquetas, sin tener palabras reservadas para ese uso. Es decir, no hay ciclos for, while, ni do-while.

Minor C es un lenguaje de programación, basado en C. Su principal funcionalidad es ser un lenguaje de alto nivel y que para propósitos de este proyecto se va a hacer toda la fase de análisis y síntesis para pasar de este lenguajes a lenguaje Augus

# Objetivos

## General

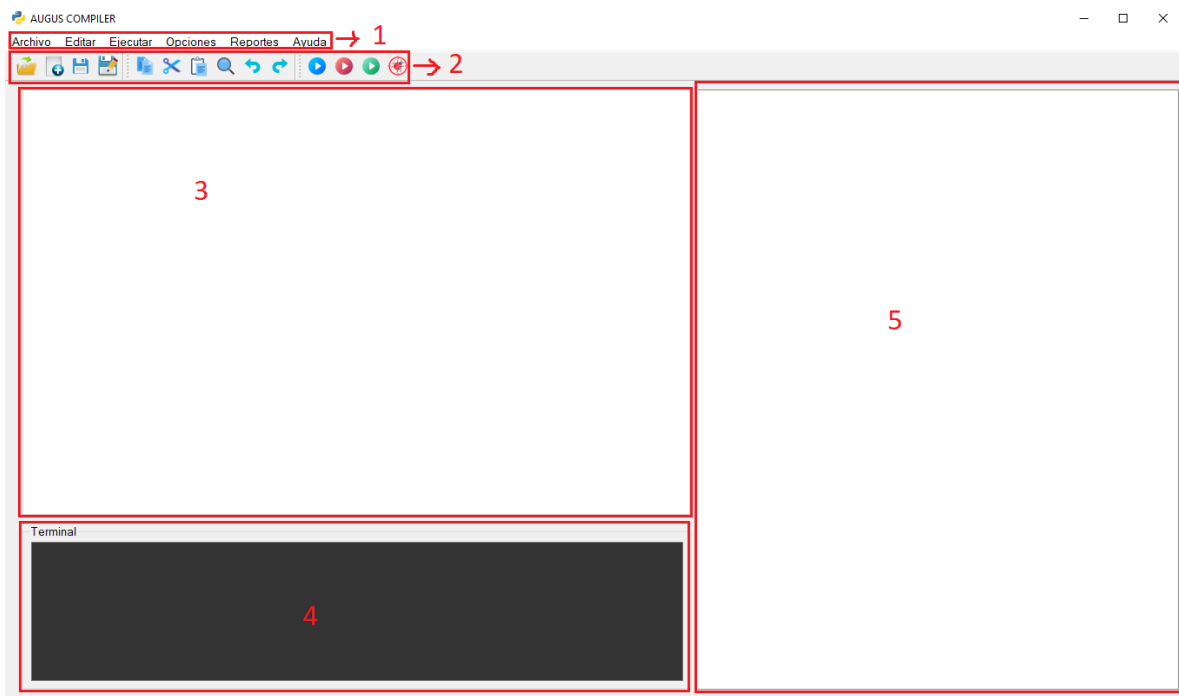
Proporcionar una ayuda al usuario que utiliza el software para que tenga mejor conocimiento sobre la manera correcta en la que tiene que manejar la aplicación.

## Específicos

- Ayudar al usuario a crear, guardar y abrir sus archivos a ejecutar.
- Enseñarle al usuario el uso adecuado del software.
- Ayudar al usuario con la visualización de los reportes
- Facilitar al usuario la locación y la acción que realizan los botones y menus de la aplicacion

# Manual de Usuario

La aplicación se ve de esta manera

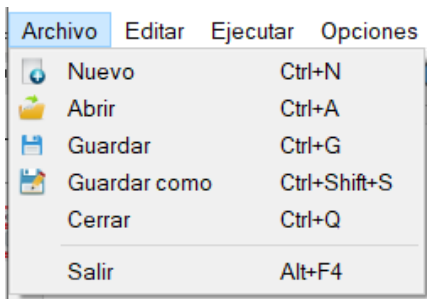


## 1. Barra de menú

En la barra de menú podemos encontrar:

- Menú Archivo
- Menú Editar
- Menú Ejecutar
- Opciones
- Reportes
- Ayuda

### Menú Archivo



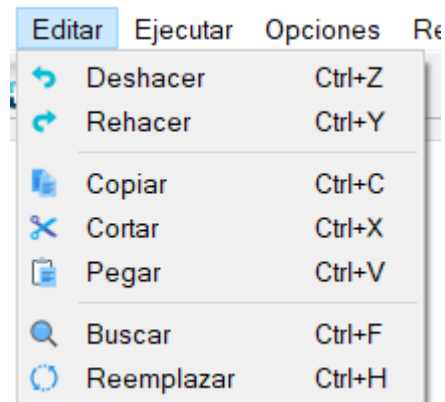
En este menú podemos encontrar las opciones clásicas para crear un nuevo documento, abrir documentos ya existentes, guardar nuestro documento, guardar como, por si quisiéramos guardar lo con otro nombre

Las opciones de cerrar para cerrar alguna de las pestañas y Salir que cierra la aplicación.

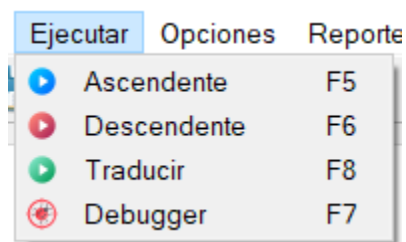
## Menú Editar

En el menú editar tenemos las opciones de deshacer y rehacer, también tenemos el copiar, pegar y cortar, y por último el buscar y reemplazar.

Como podemos darnos ver en el menú no es necesario tener que ir a buscar al menú la opción que deseamos ya que también podemos hacerlo mediante los comandos que están a la derecha de cada opción.

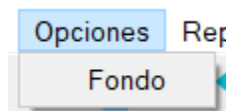


## Menu Ejecutar

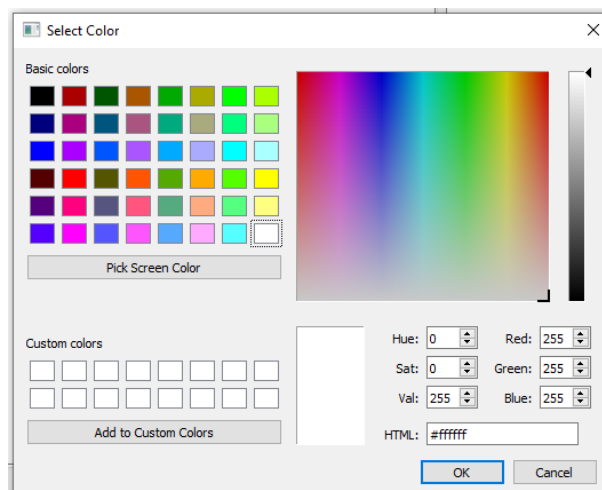


En el menú ejecutar podemos encontrar todo lo relacionado con el análisis y a ejecución del código de AUGUS y Minor C. El primer botón ejecutará utilizando la gramática Ascendente, el segundo ejecutar el código en la pestaña utilizando el análisis descendente, el tercer botón traducirá el código de Minor C a código de AUGUS y el último botón es el debugger este realizará y mostrará los resultados paso a paso en la consola.

## Opciones



En el menú opciones solo tenemos la opción de cambiar el color de fondo al editor. Esta opción desplegará una paleta de colores de la cual podemos elegir el que más nos guste.



## Reportes

En el menu de reportes podemos visualizar 5 opciones estos están divididos, los primeros 3 son reportes de AUGUS que son la tabla de simbolos, el reporte de errores y el AST, y para Minor C tenemos el reporte de simbolos, el AST de MINOR C

Reportes Ayuda

Ver Tabla de simbolos

Ver Reporte de Errores

Ver AST

Ver Reporte Simbolos

Ver AST - mini C

The screenshot shows the AUGUS COMPILER interface. On the left, a code editor displays a C program snippet. Below it, a terminal window shows the output "10". On the right, the "Reporte de simbolos" window is open, displaying the "Tabla de Simbolos".

ID	Valor	Tipo	Dimension	Declarada	Referencias
\$t1	5	ENTERO	1	main	None
\$t2	10	ENTERO	1	main	None

The screenshot shows the AUGUS COMPILER interface with the "Reporte de AST" window open. It displays the Abstract Syntax Tree for the same C program snippet. The tree structure is as follows:

```
graph TD
    Inicio --> while
    while --> ifGoto
    while --> Asignacion1
    while --> Asignacion2
    while --> Operacion1
    while --> Operacion2
    ifGoto --> end
    Asignacion1 --> t2["$t2"]
    Operacion1 --> t2
    Operacion1 --> plus["+"]
    Asignacion2 --> t1["$t1"]
    Operacion2 --> t1
    Operacion2 --> t1
```

The terminal window below the code editor shows the output "10".

**AUGUS COMPILER**

Archivo Editar Ejecutar Opciones Reportes Ayuda

basico.mc Documento 2

```

19
20 }
21
22 void operacionesBasicas()
23 {
24     printf("Operaciones Aritmeticas 1: valor esperado: \n)a62 \n)b0 \n)c-19 \n)d)256 \n)resultados:
25     \n");
26     double a = (20-10+8/2*3+10-10-10+50);
27     int a2 = a;
28     double b = (50/50*50+50-100+100-100);
29     int b2 = b;
30     double c = (100/20*9-78+6-7+8-7+7*1*2*3/3);
31     int c2 = c;
32     printf("a) %d\n", a2);
33     printf("b) %d\n", b2);
34     printf("c) %d\n", c2);
35     if (a2==62 && b2==0 && c2 == -19)
36     {
37         printf("Operaciones aritmeticas 1 bien :D\n");
38         punteo = punteo + 5;
39     }
40     else

```

Terminal

**Reporte de Símbolos**

Fila	ID	Tipo	Alcance
2	var1	variable	global
3	punteo	variable	global
5	Declaracion	Metodo	global
8	n4	variable	global_Declaracion
9	db4	variable	global_Declaracion
10	db1	variable	global_Declaracion
11	chr4	variable	global_Declaracion
22	operacionesBasicas	Metodo	global
25	a	variable	global_operacionesBasicas
26	a2	variable	global_operacionesBasicas
27	b	variable	global_operacionesBasicas
28	b2	variable	global_operacionesBasicas
29	c	variable	global_operacionesBasicas
30	c2	variable	global_operacionesBasicas
45	operacionesAvanzadas	Metodo	global
46	aritmética1	variable	global_operacionesAvanzadas
47	aritmética2	variable	global_operacionesAvanzadas
49	aritmética3	variable	global_operacionesAvanzadas

**AUGUS COMPILER**

Archivo Editar Ejecutar Opciones Reportes Ayuda

basico.mc Documento 2

```

20 }
21
22 void operacionesBasicas()
23 {
24     printf("Operaciones Aritmeticas 1: valor esperado: \n)a62 \n)b0 \n)c-19 \n)d)256 \n)resultados:
25     \n");
26     double a = (20-10+8/2*3+10-10-10+50);
27     int a2 = a;
28     double b = (50/50*50+50-100+100-100);
29     int b2 = b;
30     double c = (100/20*9-78+6-7+8-7+7*1*2*3/3);
31     int c2 = c;
32     printf("a) %d\n", a2);
33     printf("b) %d\n", b2);
34     printf("c) %d\n", c2);
35     if (a2==62 && b2==0 && c2 == -19)
36     {
37         printf("Operaciones aritmeticas 1 bien :D\n");
38         punteo = punteo + 5;
39     }
40     else

```

Terminal

**Diagrama de Flujo**

```









graph TD
    Inicio[Inicio] --> Logicas2[Logicas2]
    Logicas2 --> Asignacion[Asignacion =]
    Logicas2 --> Declaracion1[Declaracion =]
    Logicas2 --> Declaracion2[Declaracion =]
    Asignacion --> n1_1[n1]
    Asignacion --> plus[+]
    plus --> n1_2[n1]
    plus --> 1[1]
    Declaracion1 --> condicion1[condicion1]
    Declaracion1 --> not_eq[!=]
    not_eq --> n1_3[n1]
    not_eq --> 2[2]
    Declaracion2 --> aritmetica1[aritmetica1]
    aritmetica1 --> 6[6]
    aritmetica1 --> slash[/]
    slash --> n0[n0]
    slash --> 3[3]

```



## 2. Barra de herramientas

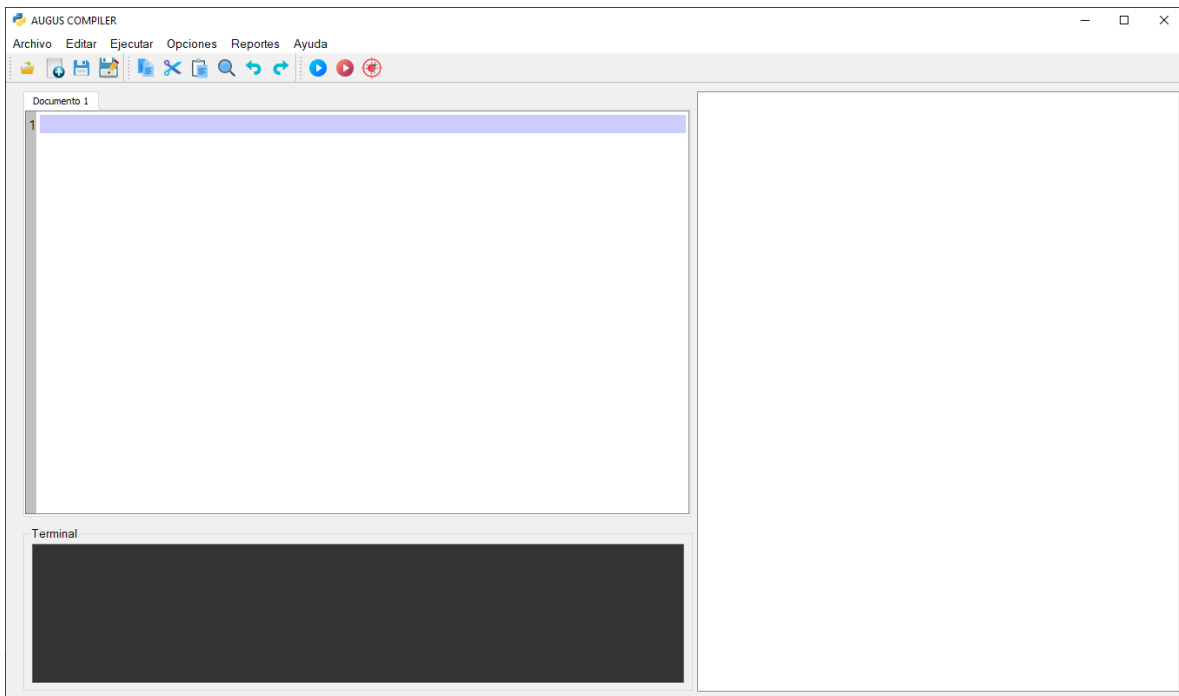
En la barra de herramientas podemos encontrar:

-  Botón abrir
-  Botón guardar
-  Boton guardar como
-  Copiar, pegar y cortar
-  Boton Hacer y rehacer
-  Boton analizar ascendente
-  Boton analizar descendente
-  Boto Debug

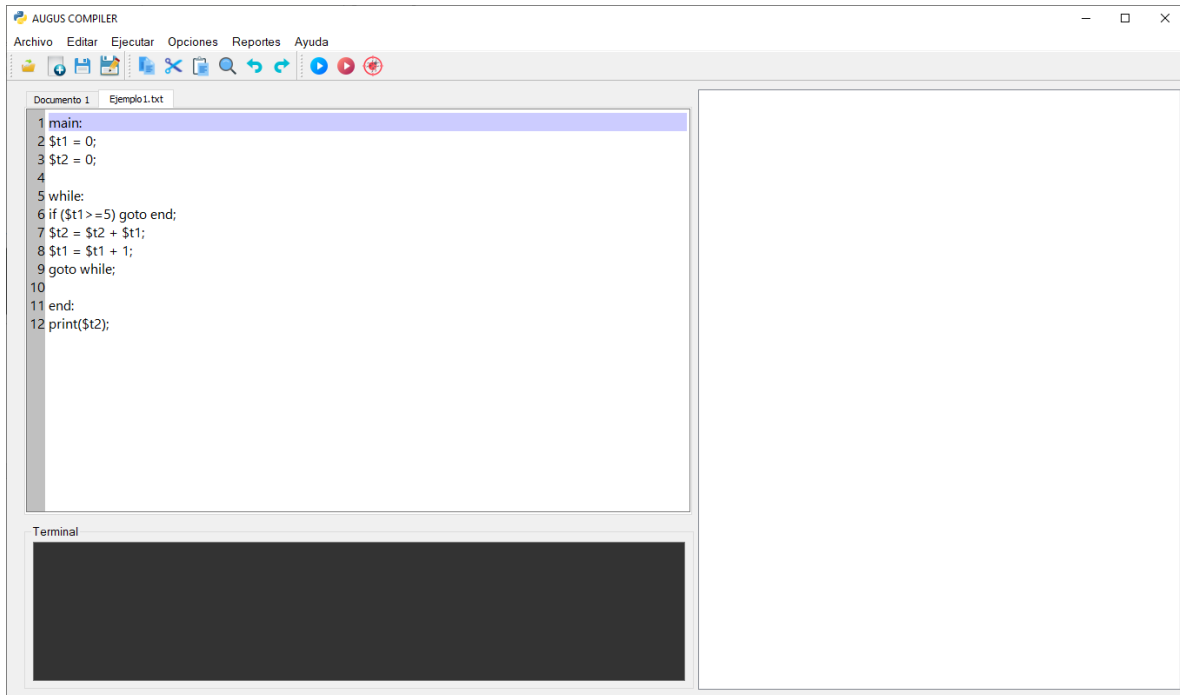
Estos botones funcionan del mismo modo como fueron descritos en la sección de barra de menús.

## 3. Área de trabajo

Para poder comenzar se debe de agregar una nueva hoja, esto se hace utilizando el botón de nuevo o con el comando Ctrl + N, o abriendo algún archivo que ya se han creado



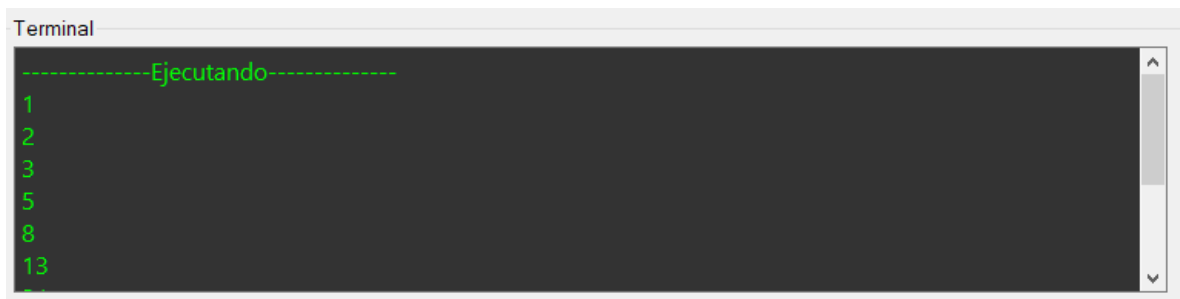
Pestaña creada con el botón nuevo



Pestaña creada a partir de un archivo existente (abrir)

## 4. Consola

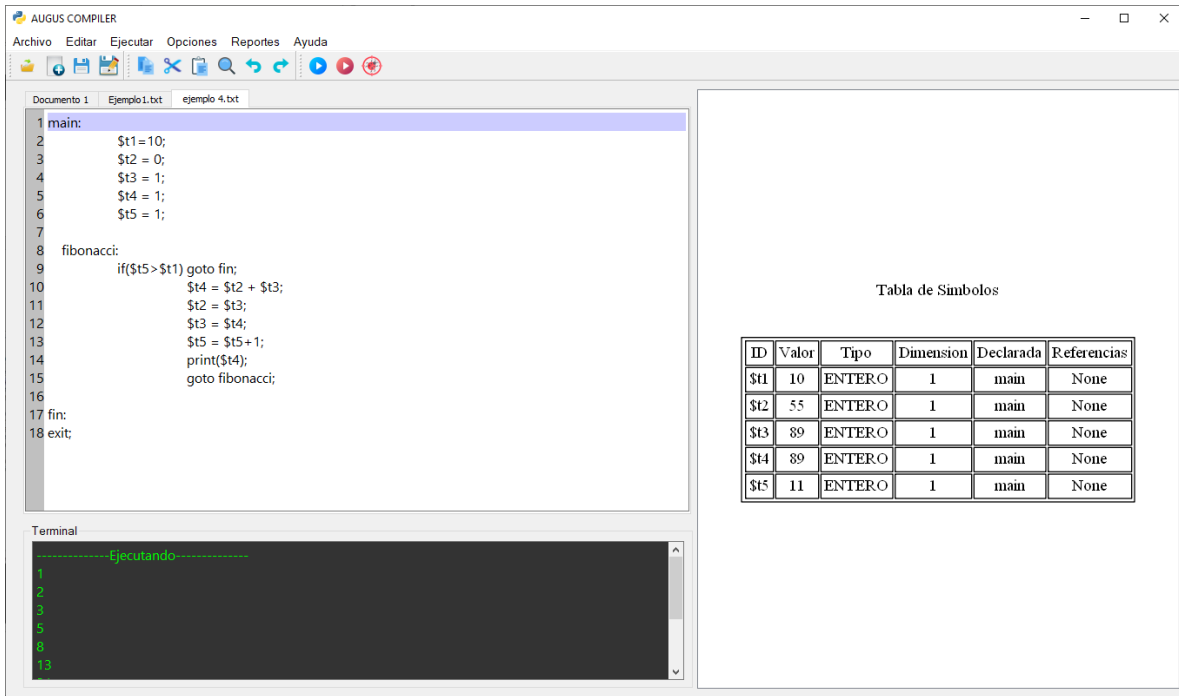
La consola se ubica en la parte inferior de la pantalla y nos ayuda a saber cuales fueron las salidas de los prints o de los mensajes que nos manda el programa



Ejemplo de la salida del factorial de 10

## 5. Visor de Imágenes

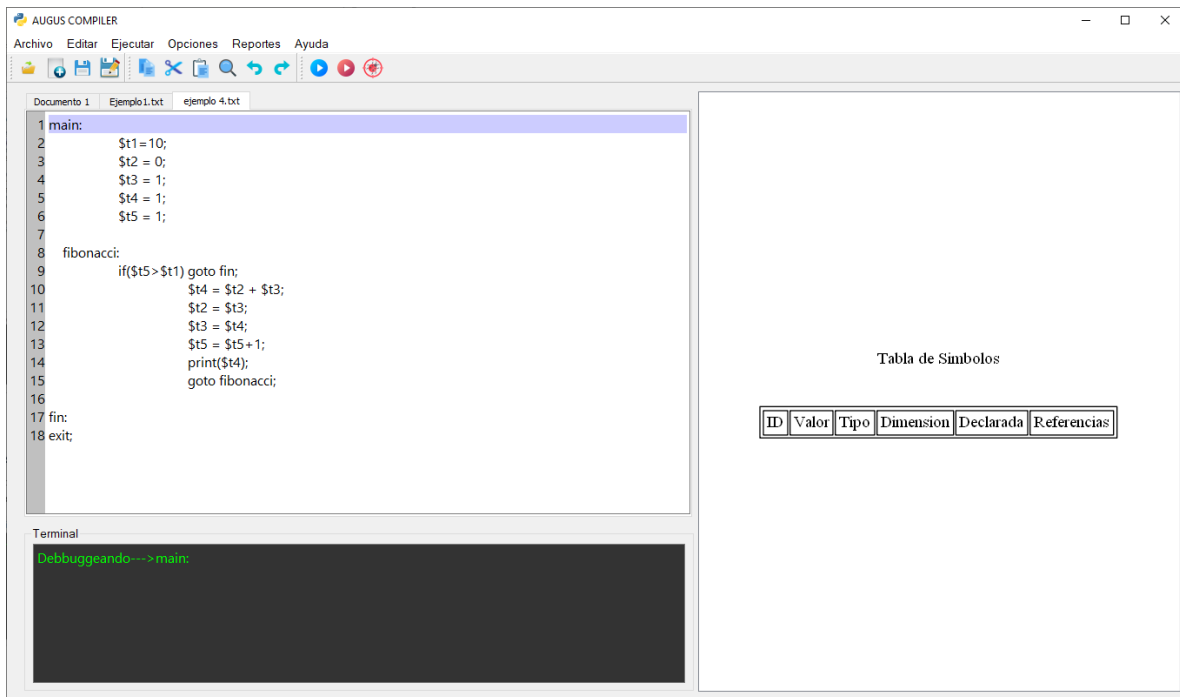
El visor de imágenes sirve para mostrar los reportes dentro de la aplicación, en el se mostrarán los reportes que el usuario quiera hacer.



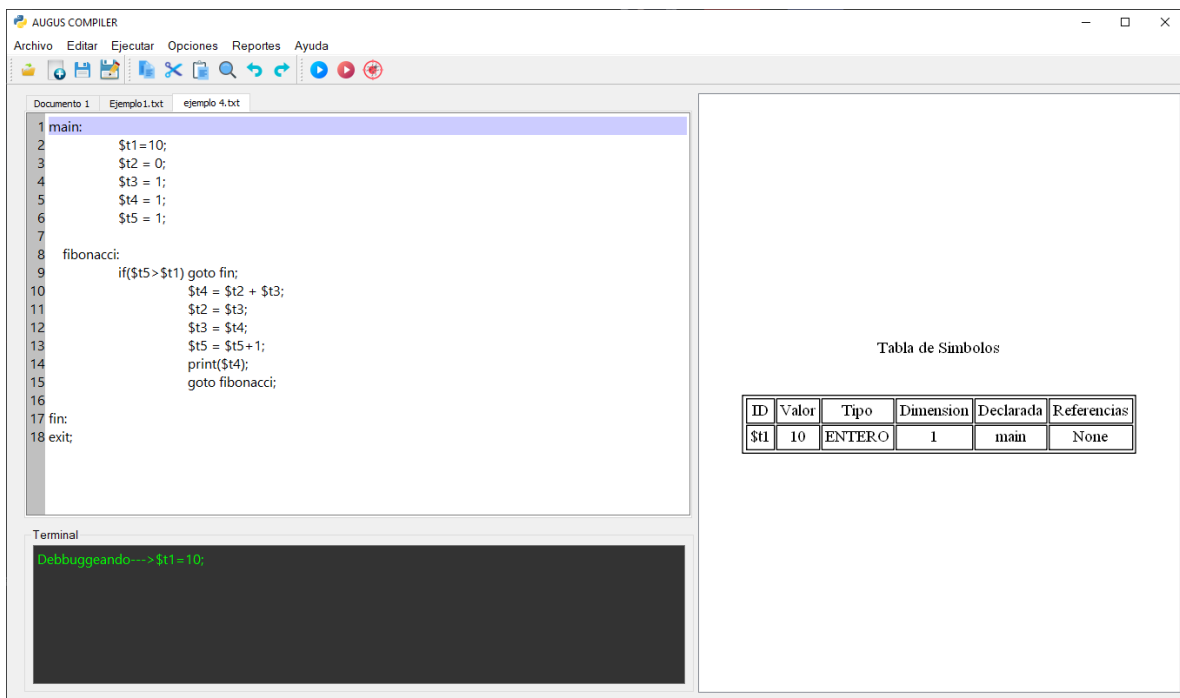
Aquí en el ejemplo podemos como después de haber dado click en el botón de reporte tabla de simbolos generara en el visor la imagen correspondiente a lo analizado.

## Debugeo

Para debugear dentro de la aplicación es fácil, solo necesitamos que nuestro código no tenga errores léxicos ni sintácticos. Para empezar con el debugeo basta con dar clic una vez al botón rojo que tiene un insecto y el programa empezara debugear



Aquí podemos ver que una vez que le damos debuggear, en la consola nos aparecerá el texto de debuggeo y nos mostrara la línea que esta ejecutado en ese momento. También nos ira mostrando la tabla de símbolos en tiempo real.

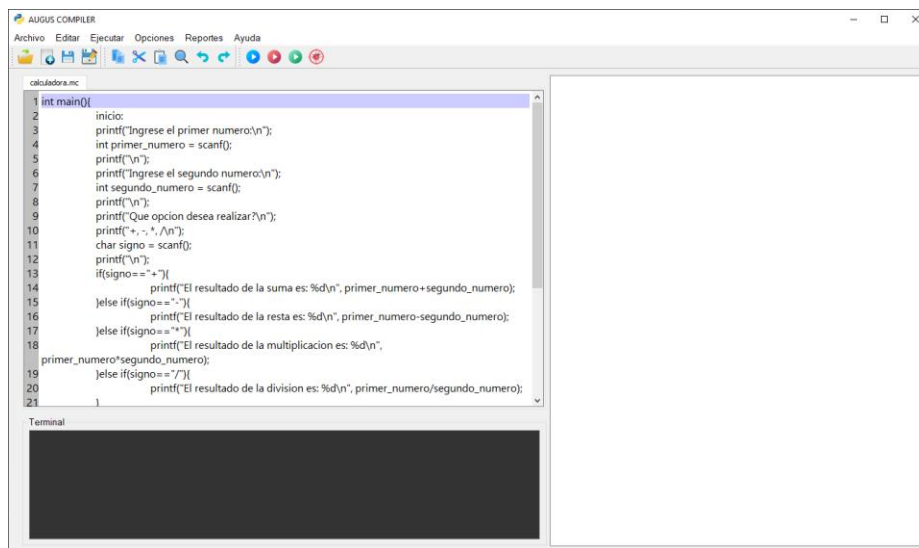


Para continuar con el debuggeo solo se tiene que volver a presionar el botón con el insecto para que continúe con la siguiente instrucción. Si quisiera iniciar una

ejecución con el analizador descendente o ascendente no hace falta para el depugeo, solo tiene que hacer clic en el botón de la ejecución que desee y automáticamente ya

## Traducción

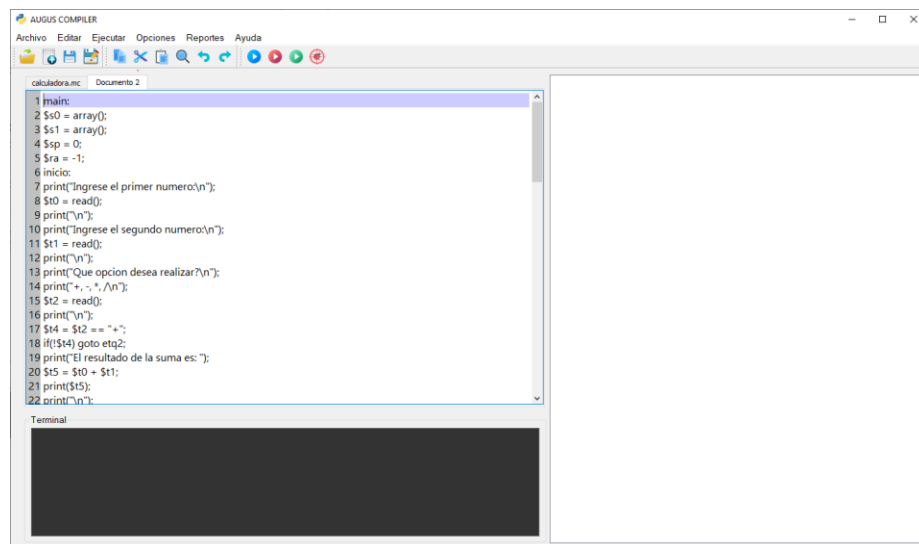
Para realizar una traducción solo necesitamos abrir un documento con algunas sentencias de Minor C, luego podemos ir nos al menú de ejecutar y dar le clic a la opción de traducir, o también podemos simplemente darle clic al botón verde para iniciar la traducción de código. El programa automáticamente empezara a traducir y nos mostrara el resultado de la salida en un nuevo documento.



The screenshot shows the AUGUS COMPILER window with the menu bar (Archivo, Editar, Ejecutar, Opciones, Reportes, Ayuda) and a toolbar. The editor displays a C program named 'calculadora.mc' with the following code:

```
1 int main()
2 {
3     inicio:
4     printf("Ingrese el primer numero\n");
5     int primer_numero = scanf();
6     printf("\n");
7     printf("Ingrese el segundo numero\n");
8     int segundo_numero = scanf();
9     printf("\n");
10    printf("Que opcion desea realizar?\n");
11    printf("+, -, *, /\n");
12    char signo = scanf();
13    printf("\n");
14    if(signo == "+"){
15        printf("El resultado de la suma es: %d\n", primer_numero+segundo_numero);
16    }
17    else if(signo == "-"){
18        printf("El resultado de la resta es: %d\n", primer_numero-segundo_numero);
19    }
20    else if(signo == "*"){
21        printf("El resultado de la multiplicacion es: %d\n",
22            primer_numero*segundo_numero);
23    }
24    else if(signo == "/"){
25        printf("El resultado de la division es: %d\n", primer_numero/segundo_numero);
26    }
27 }
```

Below the editor is a terminal window, and to the right is a large empty area for the output.



The screenshot shows the AUGUS COMPILER window with the menu bar and toolbar. The editor displays the translated assembly code for the program, with the following code:

```
1 main:
2 $s0 = array();
3 $s1 = array();
4 $sp = 0;
5 $ra = -1;
6 inicio:
7 print("Ingrese el primer numero\n");
8 $t0 = read();
9 print("\n");
10 print("Ingrese el segundo numero\n");
11 $t1 = read();
12 print("\n");
13 print("Que opcion desea realizar?\n");
14 print("+, -, *, /\n");
15 $t2 = read();
16 print("\n");
17 $t4 = $t2 == "+";
18 if($t4) goto eto2;
19 print("El resultado de la suma es: ");
20 $t5 = $t0 + $t1;
21 print($t5);
22 print("\n");
```

Below the editor is a terminal window, and to the right is a large empty area for the output.

## Menú ayuda

Dentro de este menú tendremos las siguientes opciones:

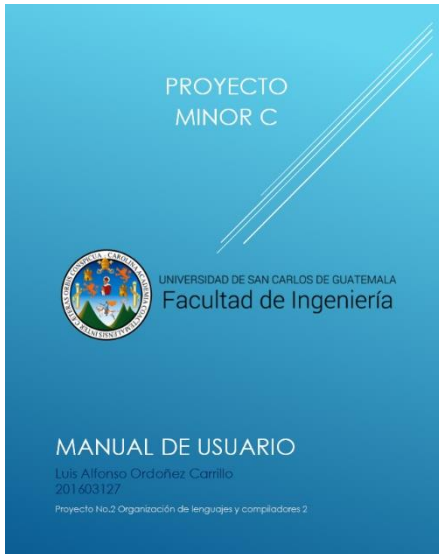
- Ver manual Usuario
- Ver manual Tecnico
- Acerca de....

Ayuda

Ver Manual Usuario

Ver Manual Tecnico

Acerca de...

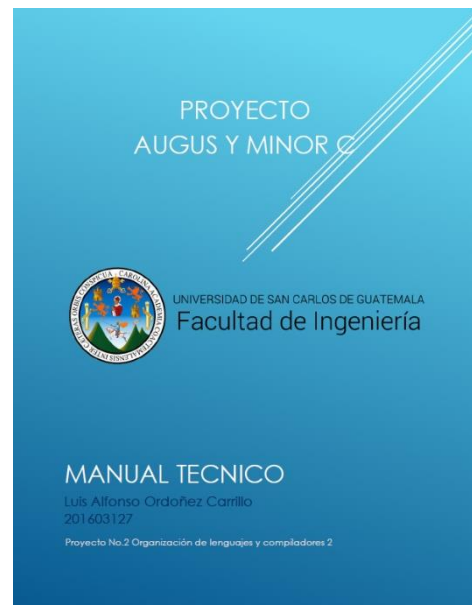


### Ver manual de usuario

Al seleccionar esta opción le desplegará este mismo documento para que pueda leerlo

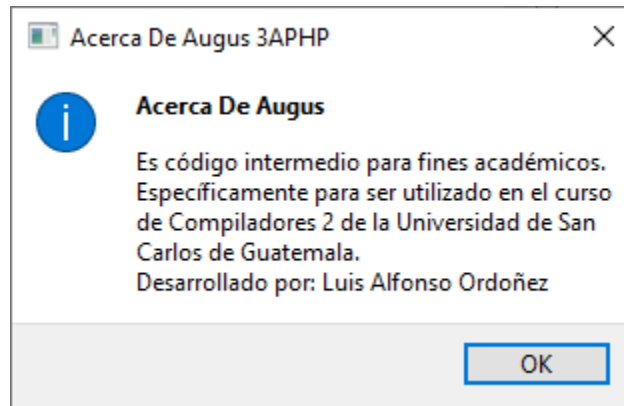
### Ver manual Técnico

Al seleccionar esta opción, el programa desplegará el manual técnico para poder ayudar al usuario a entender más sobre la lógica utilizada para crear



## Acerca de...

Este desplegará una pequeña ventana sobre información del software y del desarrollador



# Contacto

Para contactar al desarrollador para soporte técnico o para servicio personalizado puede visitar la página [\*\*www.facebook.com/lordonezcarrillo\*\*](https://www.facebook.com/lordonezcarrillo).