



12 DE JUNIO DE 2017

# MANUAL DE USUARIO

c-0

YORLEY AGUILAR  
FERNANDA ALVARADO

## Contenido

1. Introducción .....	2
2. Estructura del programa. ....	3
3. Caracteres no permitidos.....	4
4. Símbolos aritméticos.....	5
5. Interfaz gráfica .....	5
5.1 Interfaz gráfica y sus partes. ....	5
5.2 Abrir o guardar un archivo.....	6
5.3 Cambiar color a IDE.....	7
5.4 Analizar código.....	8
6. Código de ejemplo .....	10

## 1. Introducción

En el curso Compiladores e Interpretes del ITCR se ha desarrollado el compilador C-0, el cuál es una derivación del lenguaje C; con esto se pretendía que los estudiantes pusieran en práctica los conocimientos obtenidos durante el curso.

C-0 cuenta con una serie una interfaz donde puede escribir el código, ejecutarlo y visualizar todos los tokens, errores: léxicos, sintácticos y semánticos.

Este compilador cuenta con una serie de instrucciones que serán detalladas en este documento, además se explicará como utilizar este software. Además; el software no reconoce instrucciones que estén escritas en mayúscula.

## 2. Estructura del programa.

En esta sección se detalla la forma en que se deben escribir las instrucciones para que este funcione.

Instrucción	Descripción	Ejemplo
<b>Int</b>	Este software solo utiliza un tipo de dato, esta instrucción se escribe antes de declarar una variable o declarar una función.	Declaración de variables: int y; int x=3;  Declaración de funciones: Int func_name(opcional parameters){  }
<b>main</b>	Es utilizada para iniciar el programa; normalmente se escribe después de haber declarado una variable.	main(){ instructions; . . . }
<b>puts</b>	Imprime una cadena de strings.	puts("Hola mundo");
<b>putw</b>	Imprime un entero; ya sea de una asignación de variable o un número normal.	putw(x); putw(6);
<b>if</b>	Intrucción que evalúa si se cumple una condición o no.	if (3<5) { instructions; . . . }
<b>else</b>	Instrucción que se realiza si no se cumple la condición del if.	else instructions; . . . }
<b>while</b>	Es una instrucción iterativa; es decir se ejecuta cierta cantidad de veces hasta que la condición no se cumple.	while (3<x) { instructions; . . . }

<b>break</b>	Esta instrucción se encuentra en un while o un switch; lo que hace es finalizar el ciclo. En este caso; si es un break de un while, es necesario utilizar “;” al final de la instrucción si no, no se escribe “;”.	<pre>while (3&lt;x) {     instructions;     .     .     .     break }</pre>
<b>switch</b>	Es implementado para evaluar varios casos.	<pre>switch(x) {     instructions;     .     .     . }</pre>
<b>case</b>	Se escribe dentro del case, indica el caso que se va a evaluar. Al final de cada case se escribe un break.	<pre>switch(x) {     case 1:         instructions;         .         .         .         break }</pre>
<b>default</b>	Instrucción que siempre debe ir en el switch; esta instrucción se ejecuta si no cumple con ninguno de los casos evaluados.	<pre>switch(x) {     case 1:         instructions;         .         .         .         break     default:         break }</pre>
<b>return</b>	Devuelve un valor.	<pre>return x</pre>

### 3. Caracteres no permitidos.

Al igual que en todo lenguaje, C-0 tiene una serie de símbolos que no se pueden utilizar ya que dan error.

Entre ellos están:

á, é, í, ó, ú, Á, É, Í, Ó, Ú, ñ, Ñ, ¿

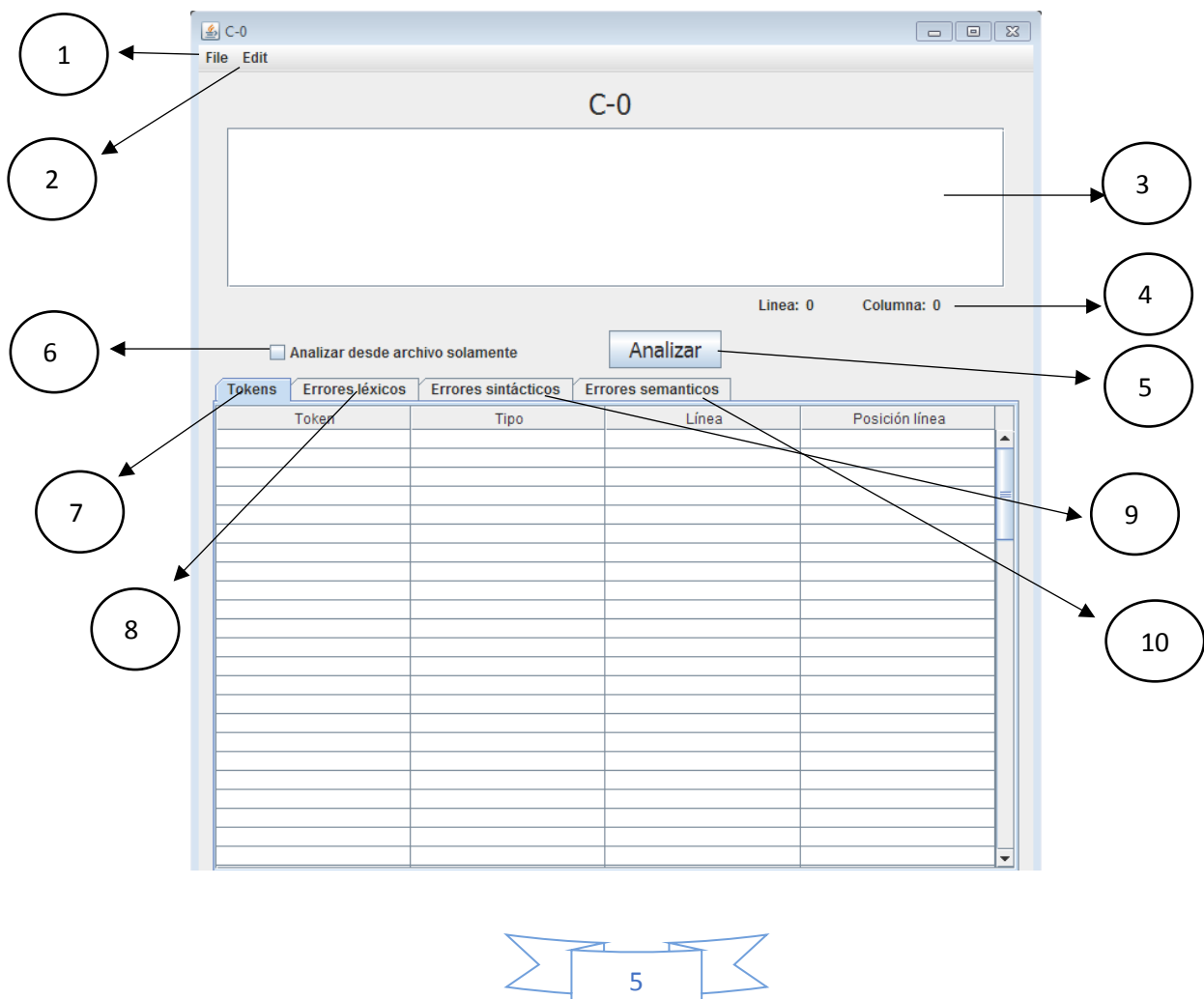
#### 4. Símbolos aritméticos.

Nombre	Símbolo
Suma	+
Resta	-
Multiplicación	*
División	/
Menor	<
Mayor	>
Igual	==
Diferente	!=
And	&&
Or	
Asignación	=

#### 5. Interfaz gráfica

La interfaz gráfica del programa se ha diseñado lo más simple posible; esto para que el usuario no tenga ningún inconveniente.

##### 5.1 Interfaz gráfica y sus partes.

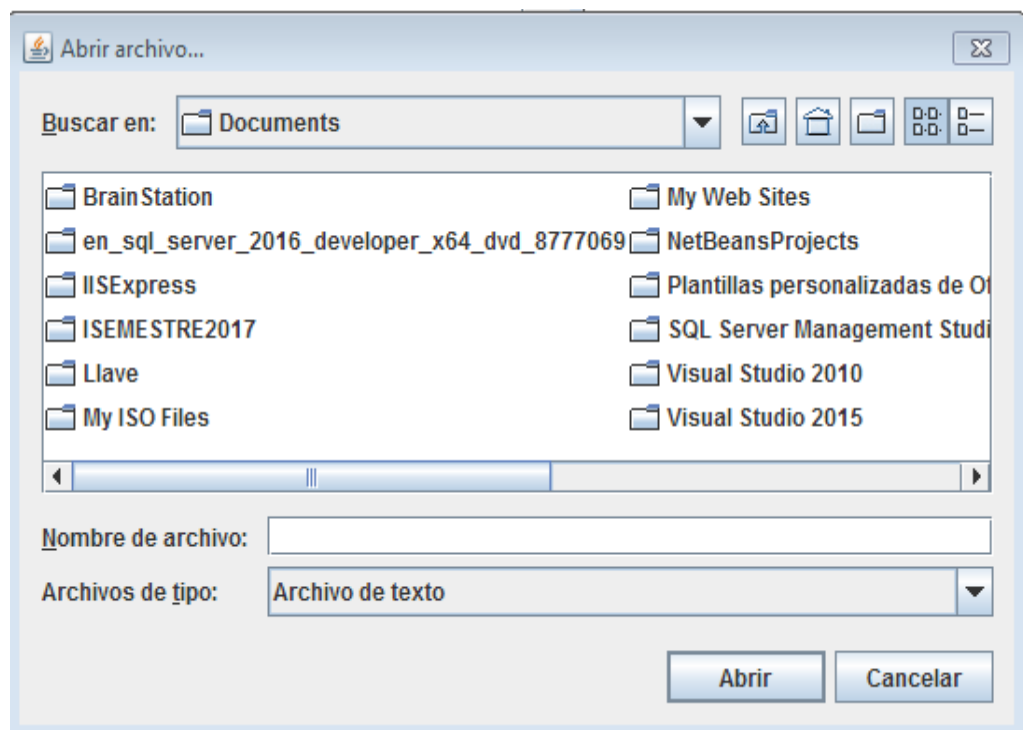


1. Abrir o guardar un archivo desde cualquier directorio.
2. Configurar el color del IDE.
3. Campo para escribir el código.
4. Indica el número de línea y la columna en que está el puntero.
5. Analiza el código escrito en el área de texto.
6. Si se selecciona, lee un archivo predeterminado.
7. Muestra todos los tokens encontrados en el programa.
8. Muestra los errores léxicos encontrados.
9. Muestra los errores sintácticos encontrados.
10. Muestra los errores semánticos encontrados.

## 5.2 Abrir o guardar un archivo

A continuación, se muestra como abrir o guardar un archivo.

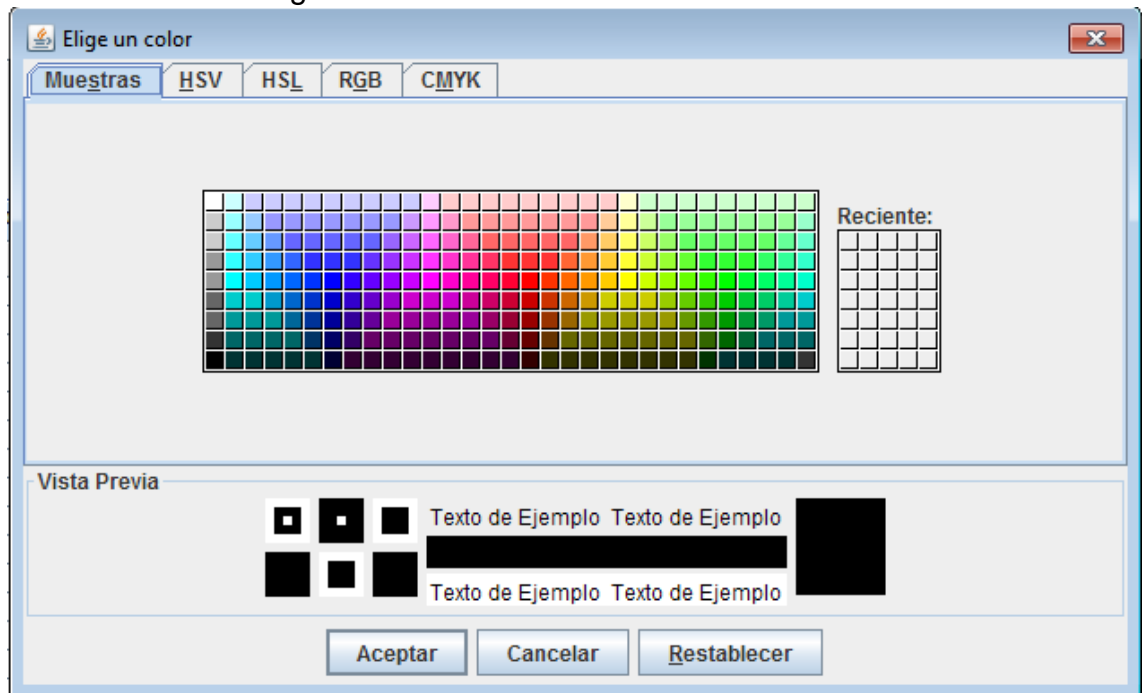
1. Debe seleccionar la opción file.
2. Elige abrir desde o guardar como.
3. Se muestra la siguiente ventana:



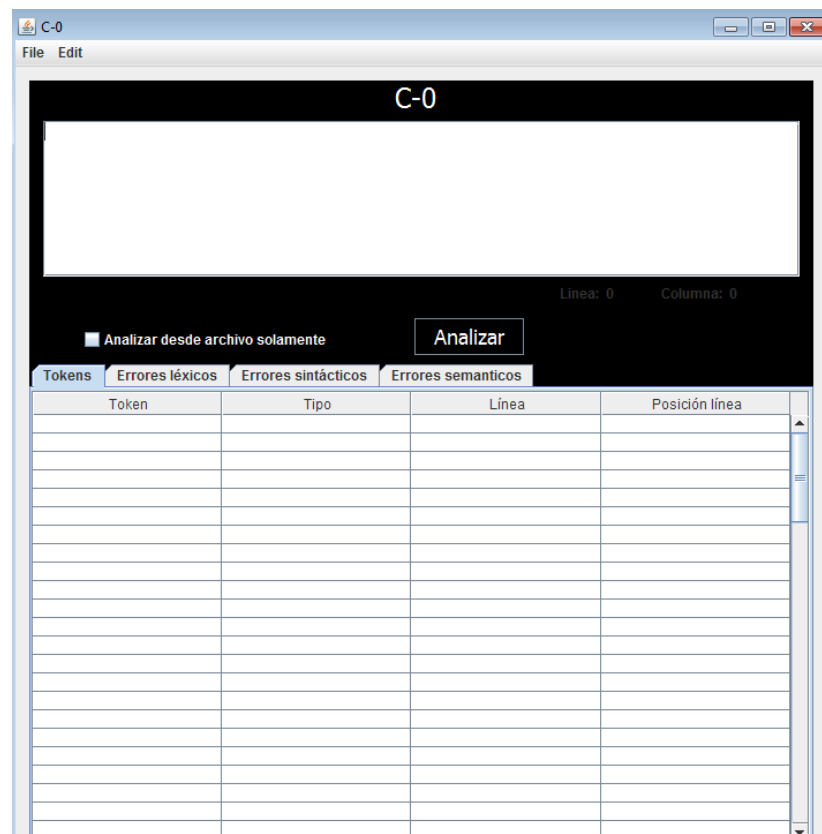
4. Busca el directorio donde desea guardar el archivo.
5. Le da un nombre al archivo.
6. Le da click en abrir.

### 5.3 Cambiar color a IDE.

1. Selecciona la opción editar.
2. Se muestra la siguiente ventana:



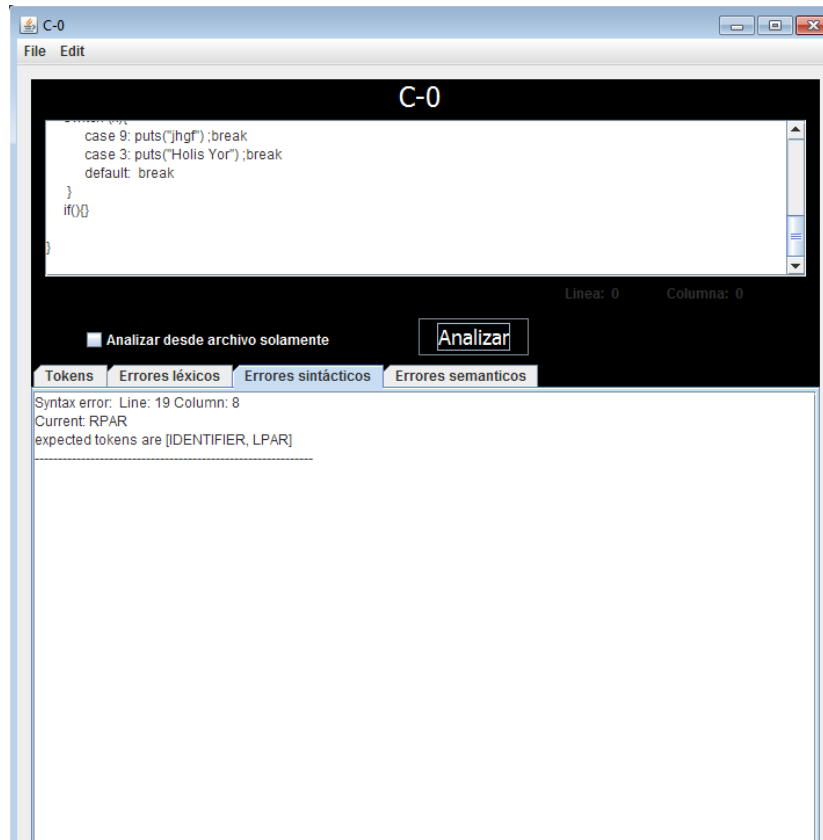
3. Elige un color.
4. Da click en aceptar y se obtiene el siguiente resultado.



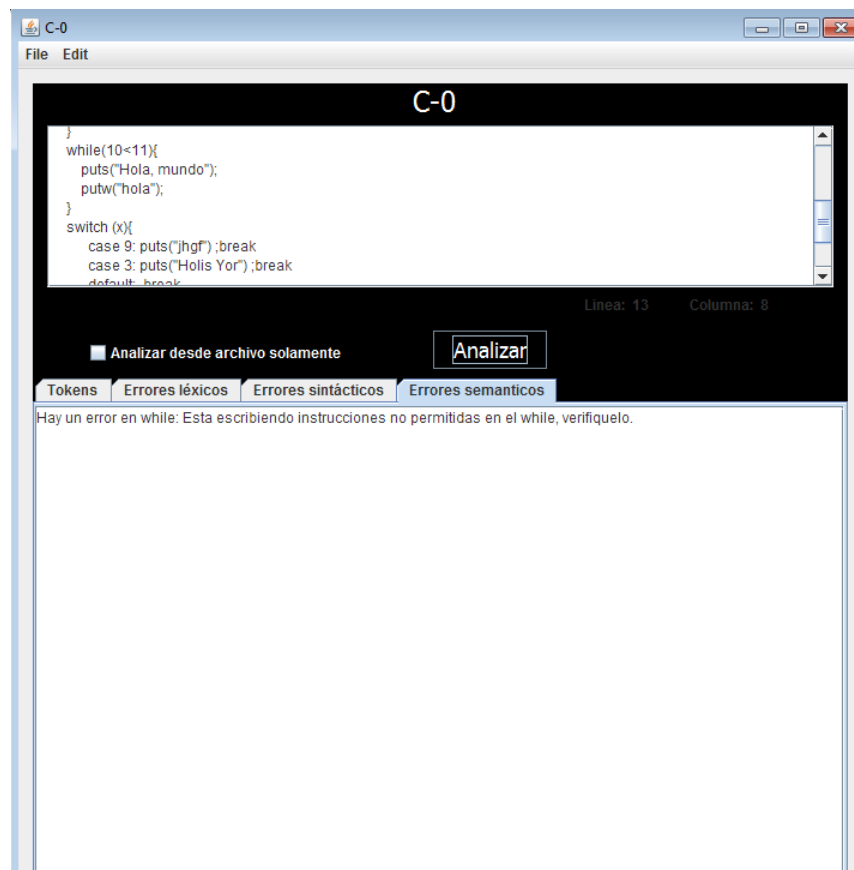




## 2.3 Ver errores sintácticos.



## 2.4 Ver errores semánticos.



## 6. Código de ejemplo

```
int x;  
  
int y;  
  
int c=6;  
  
main(){  
  
    x=6;  
    y=1+2;  
    putw(0);  
    if(9<1){  
        putw(8787);  
    }  
    while(10<11){  
        puts("Hola, mundo");  
    }  
    switch (x){  
        case 9: puts("jhgf") ;break  
        case 3: puts("Holis Yor") ;break  
        default: break  
    }  
  
}
```