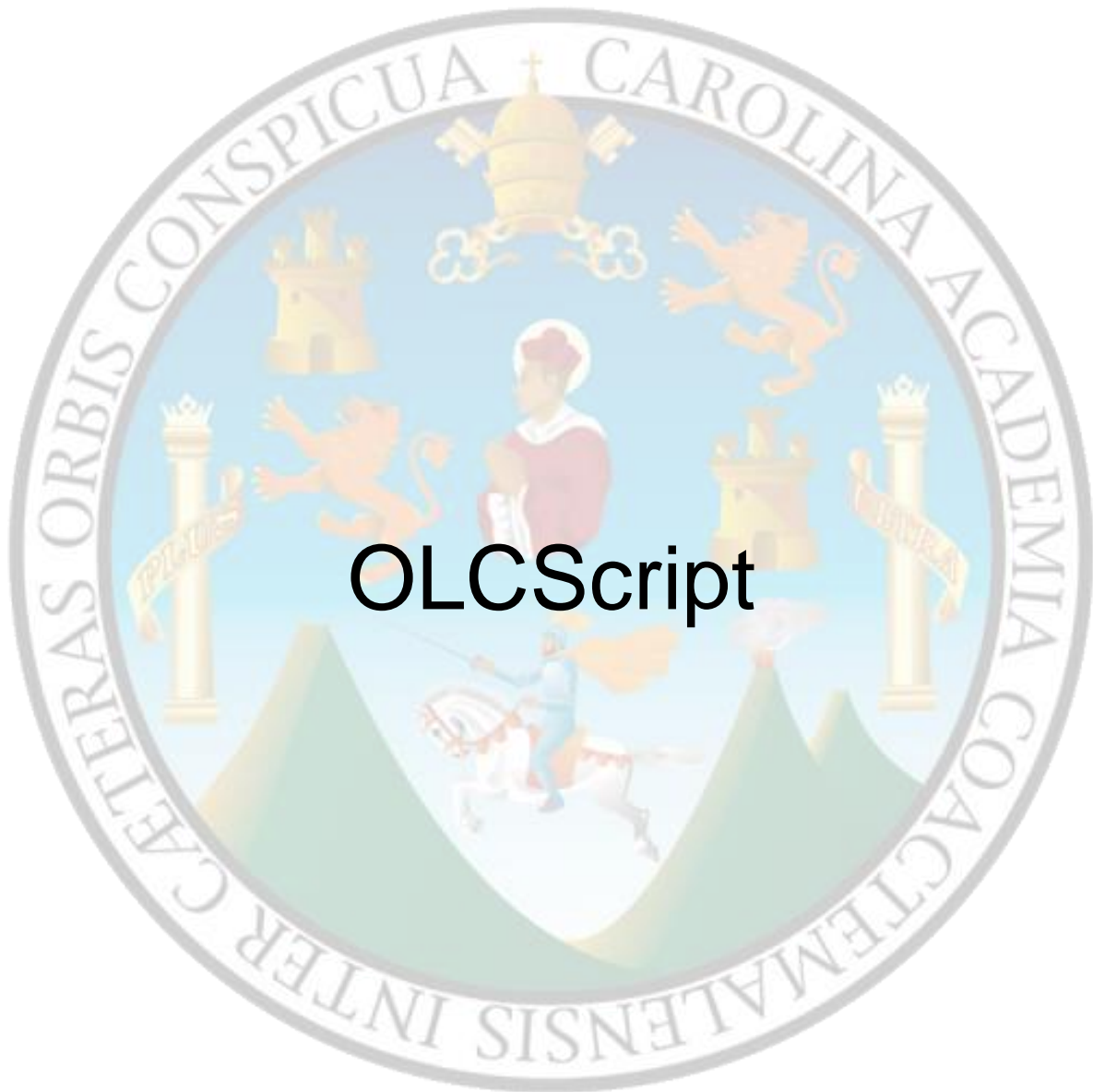


MANUAL DE USUARIO



Contenido

I. Introducción.....	1
1. Objetivo.....	1
2. Requerimiento.....	1
II. Diseño.....	1
III. Opciones del Sistema.....	2
1. Archivo.....	2
1.1. Abrir	2
1.2. Guardar.....	3
1.3. Guardar como	3
2. Ejecutar.....	3
3. Consola.....	3
4. Tabla de Símbolos	4
5. Errores	4
IV. Archivo.....	5
1. Estructura de Archivo.....	5

I.Introducción

1. Objetivo

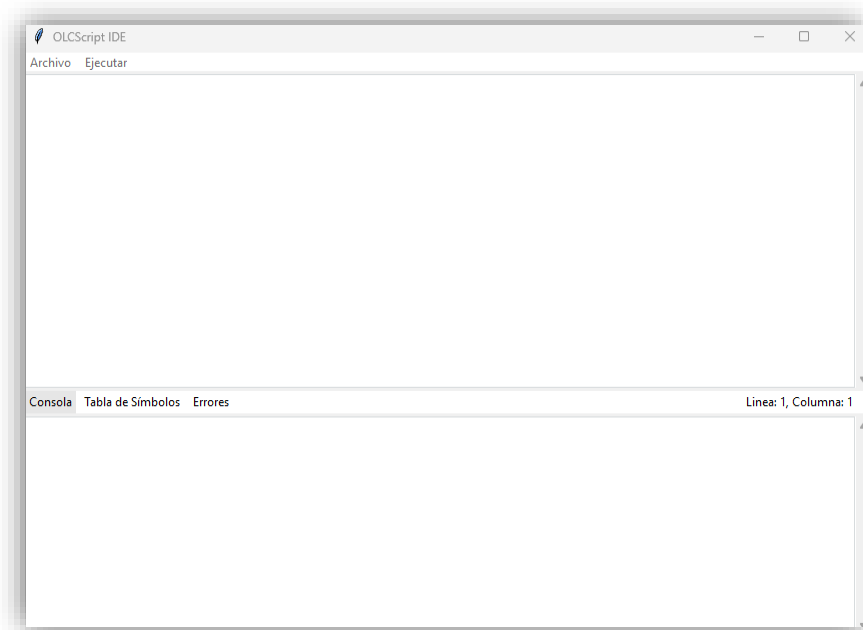
Brindar asistencia al usuario de este programa informático (OLCScript Interpreter), describiendo las opciones o el funcionamiento del proceso que se muestra en cada una de las pantallas conforme el usuario vaya avanzando en dicho programa.

2. Requerimiento

- Sistema Operativo: Windows 7 o superior
- Procesador mínimo Intel Pentium (800 MHz Intel Pentium)
- Mínimo 1GB en RAM
- IDE Visual Studio Code, o compatible con Python
- Exploradores: Internet Explorer 9 y superior

II.Diseño

A continuación, se muestra una imagen de la interfaz de la aplicación.



III. Opciones del Sistema

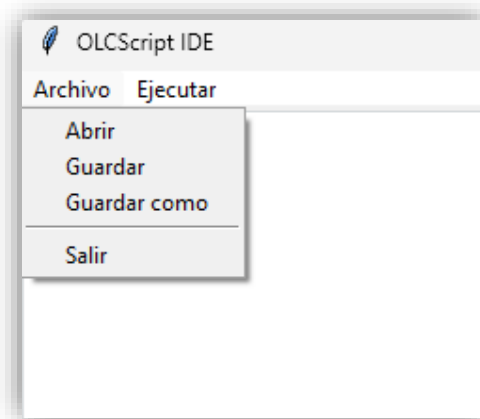
El presente manual está organizado de acuerdo con el menú de opciones de la siguiente manera:

1. Archivo
 2. Ejecutar
 3. Consola
 4. Tabla de Símbolos
 5. Errores
-

1. Archivo

Al presionar este botón se desplegará una lista con todas las opciones que el usuario puede elegir.

Cada opción es representada por un botón el cual el usuario puede presionar para realizar una acción.



1.1. Abrir

Al hacer clic sobre el botón "Abrir" este mostrará el explorador de archivos para que el usuario seleccione el archivo que desee ejecutar y/o editar.

El programa soporta únicamente archivos con la extensión .qc y pueden ser abiertos varios archivos en la misma ejecución.

Si se intenta abrir un archivo con la extensión incorrecta el programa mostrará un mensaje de advertencia en pantalla y no leerá el archivo seleccionado.

1.2. Guardar

Al presionar el botón “Guardar” el programa guardará el contenido ingresado en el área de texto.

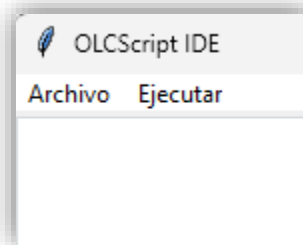
Si existiese un archivo abierto el contenido del área de texto será almacenado en dicho archivo, de no ser así el contenido será almacenado en un archivo nuevo si el usuario así lo desea.

1.3. Guardar como

Al presionar el botón “Guardar como” el programa desplegará el explorador de archivos para que el usuario pueda guardar el archivo con el nombre y la ruta que él desee.

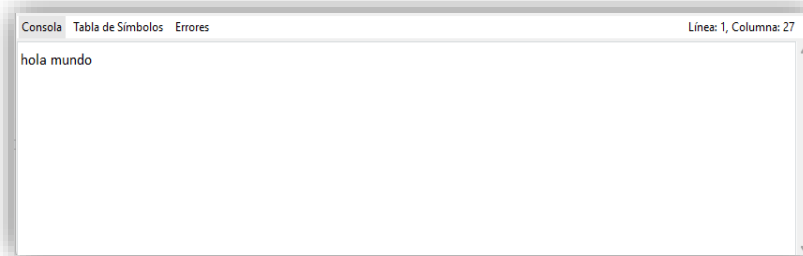
2. Ejecutar

Al hacer clic sobre este botón el programa analizará todo el contenido que encuentre en el área de texto. Debe existir contenido en el área de texto pues de no ser así se mostrará un mensaje de error en la pantalla.



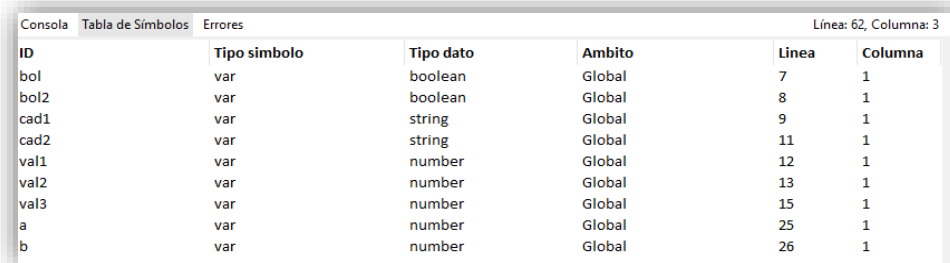
3. Consola

Al presionar el botón “Consola” muestra todos los resultados obtenidos luego de interpretar el contenido del archivo de entrada.



4. Tabla de Símbolos

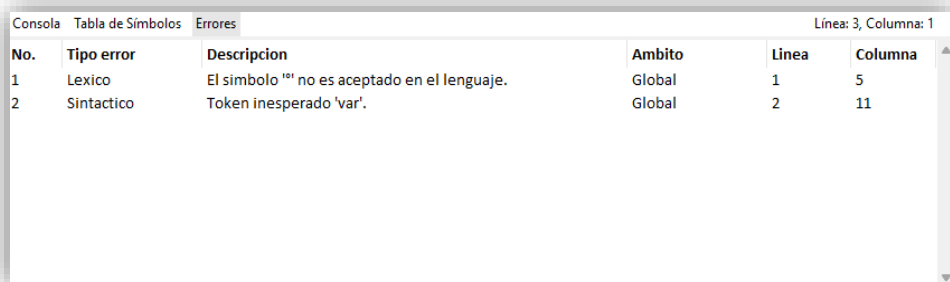
Al hacer clic sobre el botón “Tabla de símbolos” se mostrará una tabla con todos los símbolos encontrados al momento de interpretar el contenido del archivo de entrada.



ID	Tipo símbolo	Tipo dato	Ambito	Linea	Columna
bol	var	boolean	Global	7	1
bol2	var	boolean	Global	8	1
cad1	var	string	Global	9	1
cad2	var	string	Global	11	1
val1	var	number	Global	12	1
val2	var	number	Global	13	1
val3	var	number	Global	15	1
a	var	number	Global	25	1
b	var	number	Global	26	1

5. Errores

Al presionar el botón “Errores” se mostrará una tabla que contiene todos los errores tanto léxicos, sintácticos y semánticos encontrados al momento de ejecutar el contenido del archivo de entrada.



No.	Tipo error	Descripción	Ambito	Linea	Columna
1	Lexico	El símbolo '19' no es aceptado en el lenguaje.	Global	1	5
2	Sintactico	Token inesperado 'var'.	Global	2	11

IV. Archivo

En esta sección se llevará a cabo cuál es la estructura adecuada que un archivo de entrada debe tener.

1. Estructura de Archivo

Es importante conocer cuál es la estructura correcta que debe tener un archivo de entrada para que el programa no tenga problemas al ejecutarlo.

Si el archivo de entrada no está escrito de forma correcta es posible que el programa lance un error o en caso contrario sea ejecutado pero el resultado puede no ser el que el usuario espera.

Es también de vital importancia que al momento de llamar valores de variables estas sean llamadas con la estructura correcta.

Ejemplo de archivo .olc de entrada:

```
var val1: number = 7 - (5 + 10 * (2 + 4 * (5 + 2 * 3)) - 8 * 3 * 3) + 50 * (6 * 2);
var val2: number = (2 * 2 * 2 * 2) - 9 - (8 - 6 + (3 * 3 - 6 * 5 - 7 - (9 + 7 * 7 * 7) + 10) - 5) + 8 - (6 - 5 * (2 * 3));
var val3: number = val1 + ((2 + val2 * 3) + 1 - ((2 * 2 * 2) - 2) * 2) - 2;

console.log("El valor de val1 es:", val1);
console.log("El valor de val2 es:", val2);
console.log("El valor de val3 es:", val3);
```

Consola:

Consola Tabla de Símbolos Errores Línea: 1, Columna: 1

```
El valor de val1 es: 214
El valor de val2 es: 412
El valor de val3 es: 1439
```

Símbolos

Consola Tabla de Símbolos Errores Línea: 1, Columna: 1					
ID	Tipo símbolo	Tipo dato	Ambito	Línea	Columna
val1	var	number	Global	2	1
val2	var	number	Global	3	1
val3	var	number	Global	5	1