

MANUAL DE USUARIO



Contenido

I. Introducción.....	1
1. Objetivo.....	1
2. Requerimiento.....	1
II. Opciones del Sistema.....	2
1. Archivo.....	2
1.1. Nuevo.....	2
1.2. Abrir.....	2
1.3. Guardar.....	3
2. Ejecutar.....	3
3. Reporte.....	3
3.1 Tokens.....	4
3.2 Errores.....	4
3.3 Tabla de Símbolos.....	4
3.4 Árbol Sintáctico.....	4
III. Archivo.....	4
1. Estructura de Archivo.....	4

I.Introducción

1. Objetivo

Brindar asistencia al usuario de este programa informático (QueryCrypter 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 Javascript
- Exploradores: Internet Explorer 9 y superior

II. Opciones del Sistema

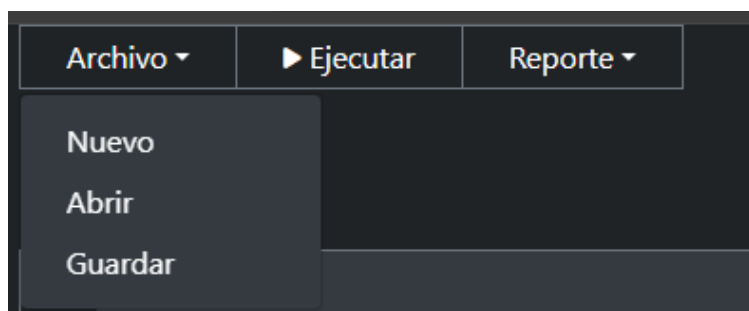
El presente manual está organizado de acuerdo con el menú de opciones de la siguiente manera:

1. Archivo
 2. Ejecutar
 3. Reportes
-

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. Nuevo

Cuando se presiona el botón "Nuevo", se generará una nueva pestaña vacía en el editor de código. Esto te permitirá comenzar a escribir código desde cero en esa pestaña sin que haya contenido preexistente en ella.

1.2. 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.

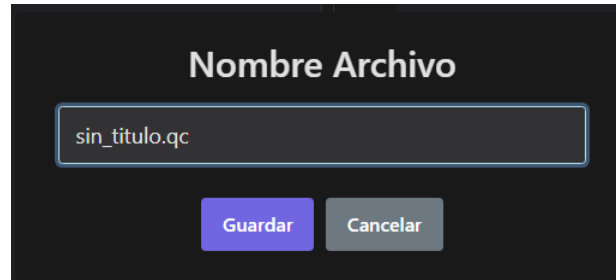
El programa soporta únicamente archivos con la extensión .qc y pueden ser abiertos varios archivos en la misma ejecución.

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.

1.3. Guardar

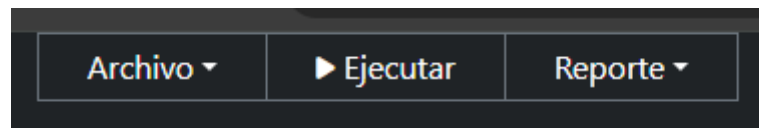
Al presionar el botón “Guardar” el programa guardará el contenido ingresado en el área de texto.

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.



2. Ejecutar

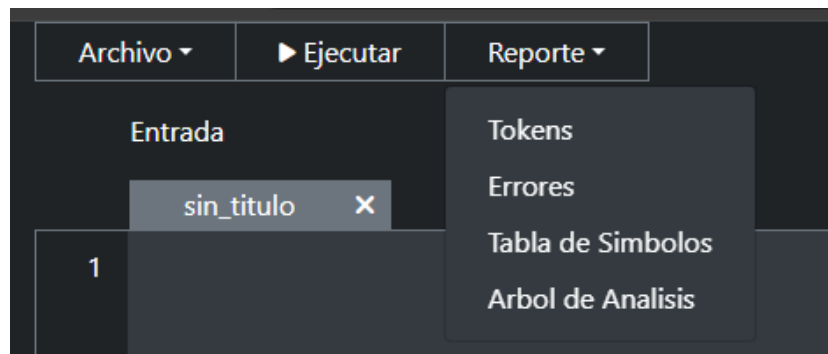
Al hacer clic sobre este botón el programa analizará todo el contenido que encuentre en el área de texto. Debe existir contenido en el área de texto pues de no ser así se mostrará un mensaje de error en la pantalla.



3. Reporte

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.



3.1 Tokens

Al hacer clic sobre el botón “Tokens” se redireccionará a una página html en la cual se encuentra una tabla que contiene todos los tokens que el analizador léxico reconoció.

Token	Valor	Línea	Columna
TK_CREATE	CREATE	1	1
TK_TABLE	TABLE	1	8
TK_ID	Empleados	1	14
TK_LEFT_PARENTHESIS	(1	24
TK_ID	id	2	2
TK_INT	INT	2	5
TK_COMMA	,	2	8
TK_ID	nombre	3	2

3.2 Errores

Al hacer clic sobre el botón “Errores” se redireccionará a una página html en la cual se encuentra una tabla que contiene todos los errores que el analizador léxico encontró.

Tipo de Error	Descripción	Línea	Columna
Lexico	Error lexico, token '!' no reconocido.	30	1
Lexico	Error lexico, token '"' no reconocido.	31	1
Lexico	Error lexico, token '&' no reconocido.	32	1
Sintactico	Error sintactico, token 'CREAAATE' no esperado.	4	1
Sintactico	Error sintactico, token 'TABLE' no esperado.	4	9
Sintactico	Error sintactico, token 'Empleados' no esperado.	4	15
Sintactico	Error sintactico, token '(' no esperado.	4	25
Sintactico	Error sintactico, token 'id' no esperado.	5	2

3.3 Tabla de Símbolos

Al hacer clic sobre el botón “Tokens” se redireccionará a una página html en la cual se encuentra una tabla que contiene todos los tokens que el analizador léxico reconoció.

Reporte de Símbolos				
Identificador	Tipo	Tipo	Línea	Columna
nombre	variable	VARCHAR	1	1
edad	variable	VARCHAR	2	1
pais	variable	VARCHAR	3	1
nombre	variable	VARCHAR	5	1
edad	variable	VARCHAR	6	1
pais	variable	VARCHAR	7	1
				Cerrar

3.4 Árbol Sintáctico

Al hacer clic sobre el botón “Tokens” se redireccionará a una página html en la cual se encuentra una tabla que contiene todos los tokens que el analizador léxico reconoció.

III. 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 .qc de entrada:

```
Entrada
archivo.qc X
1  --Bloque de codigo
2  BEGIN
3      DECLARE @cadena VARCHAR DEFAULT "OLC1";
4      PRINT cadena;
5
6      SET @cadena = "Hola mundo";
7      prINT cadena;
8
9      DECLARE @numero INT DEFAULT 5;
10
11     --string cadena2 = "Mi numero es: " + numero;
12     DECLARE @cadena2 VARCHAR DEFAULT "Mi numero es: " + numero;
13
14     print cadena2;
15
16 END;
17
18 --Instruccion DDL
19 CREATE TABLE Clientes
20 ( ID_Cliente INT, Nombre VARCHAR, Email VARCHAR );
```


Salida

Salida

```
1  OLC1
2  Hola mundo
3  cadena2
```