

Creación de un compilador

Manual de usuario

Versión: 1

Fecha: 10/10/2020

ÍNDICE

Registro de cambios.....	3
Objetivo.....	3
Acerca del Manual.....	3
Propósito.....	3
Conocimientos necesarios.....	4
Introducción.....	4
Visión global.....	5
Especificaciones.....	5
Conceptos generales.....	5
Instalación.....	5
Acceso.....	5
Pantallas.....	6

Organismo			
Proyecto	Software para analizar gramáticas ascendente		
Entregable	Manual de usuario		
Autor	Jhonny García		
Versión/Edición	01	Fecha Versión	28/08/2020
Aprobado por		Fecha Aprobación	01/09/2020
		Nº Total de Páginas	10

Registro de cambios

Versión	Causa del Cambio	Responsable del Cambio	Fecha del Cambio
01	Versión inicial	Jhonny García Hernández	10/10/2020

Objetivo

Se pretende mostrar de una manera clara y concisa el funcionamiento de la aplicación Software de un acercamiento a la creación de un compilador.

Acerca del Manual.

- **Propósito.**

El presente manual tiene como finalidad ser una guía básica de las operaciones del software; permitiendo al usuario adquirir conocimientos indispensables para una operación adecuada del sistema, y ser una herramienta de consulta de primera mano a la cual puede recurrir el usuario en cualquier momento.

Esta herramienta está diseñada con el fin de entender el funcionamiento de un compilador con sus respectivas partes, desde los diferentes tipos de análisis hasta la ejecución del código.

Se pretende dar una idea más detallada de los alcances y ventajas del sistema, que resulta útil para la comprensión de lenguajes de programación.

- **Conocimientos necesarios.**

- Conocer previamente el funcionamiento de un computador.

Introduction

El presente sistema fue creado con el propósito de ser una herramienta con el cual las diferentes usuarios finales, puedan comprender la parte del análisis hasta el código objeto de un compilador.

Una de las formas simples de mantener al cliente satisfecho es cubrir sus necesidades sin mayores complicaciones y sin quebrantar la confianza; lo que se logra con una automatización de procesos, así como con un control de la mayoría de inconvenientes durante el camino.

El manejo correcto de las computadoras y software brindan la posibilidad de una mejor administración de datos y mejorar la seguridad de datos.

Como usuario de este producto lo aprovechara y obtiene una ventaja en muchos factores.

El desarrollador.

Visión global.

- **Especificaciones.**

Los requisitos básicos del sistema son:

Estación de trabajo, teclado y mouse, y sistema operativo (cualquiera).

Contar con la máquina virtual de java, JDK11.

- **Conceptos generales.**

Instalación

El proceso de instalación es el siguiente:

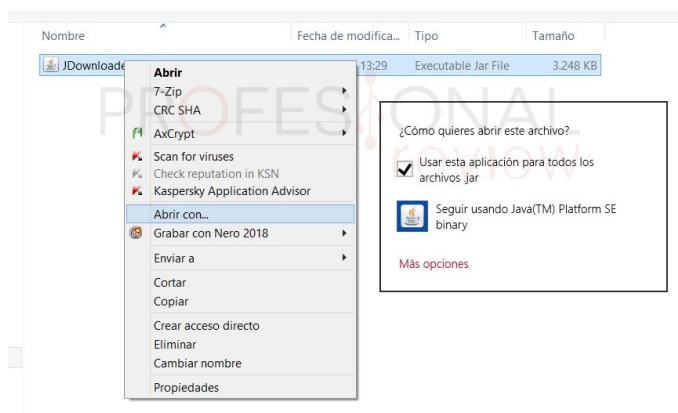
Para instalar la máquina virtual de java basta con seguir el siguiente link:

<https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>

Para sistemas operativos Windows, Linux o Mac.

Acceso.

Seleccionar la aplicación proporcionada por el desarrollador, clic derecho y abrir con la aplicación anteriormente instalada, aconsejable dar checko en la opción de usar esta aplicación para todos los archivos.jar;



Pantallas

Pantalla principal con todos los menús:

Archivo:

1. Abrir un archivo: Abre un archivo, a través de un buscador, con extensión mlg.
2. Guardar un archivo: Toma el documento abierto y seleccionado, y guarda los datos.
3. Guardar como un archivo: Toma el documento abierto y seleccionado, luego solicita la ruta donde se guardara el archivo
4. Nuevo archivo: crea un archivo en blanco
5. Salir: Cierra la aplicación

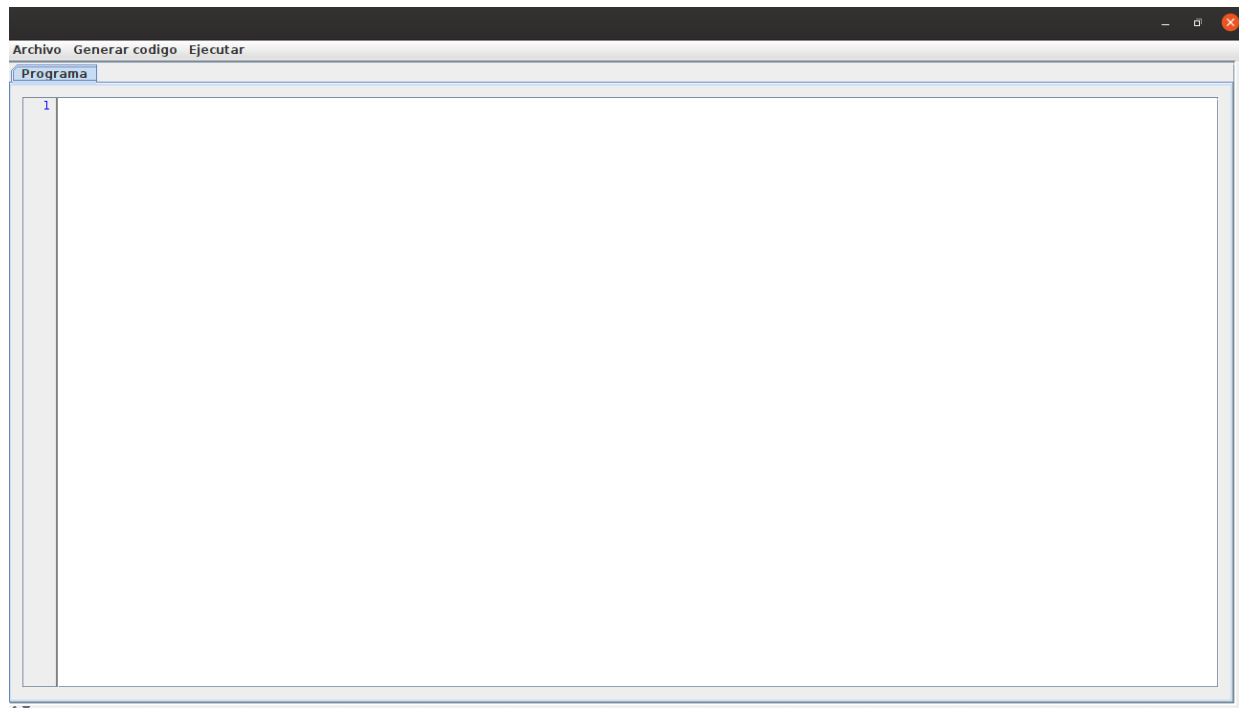
Generar código:

1. Compilar: Compila la entrada de texto abierto.
2. Generar código de 3 direcciones: Si la compilación esta completa, genera el código intermedio y lo muestra.
3. Código Optimizado: Optimiza el código intermedio y lo muestra con los respectivos cambios.
4. Código Assembler: genera el código assembler.

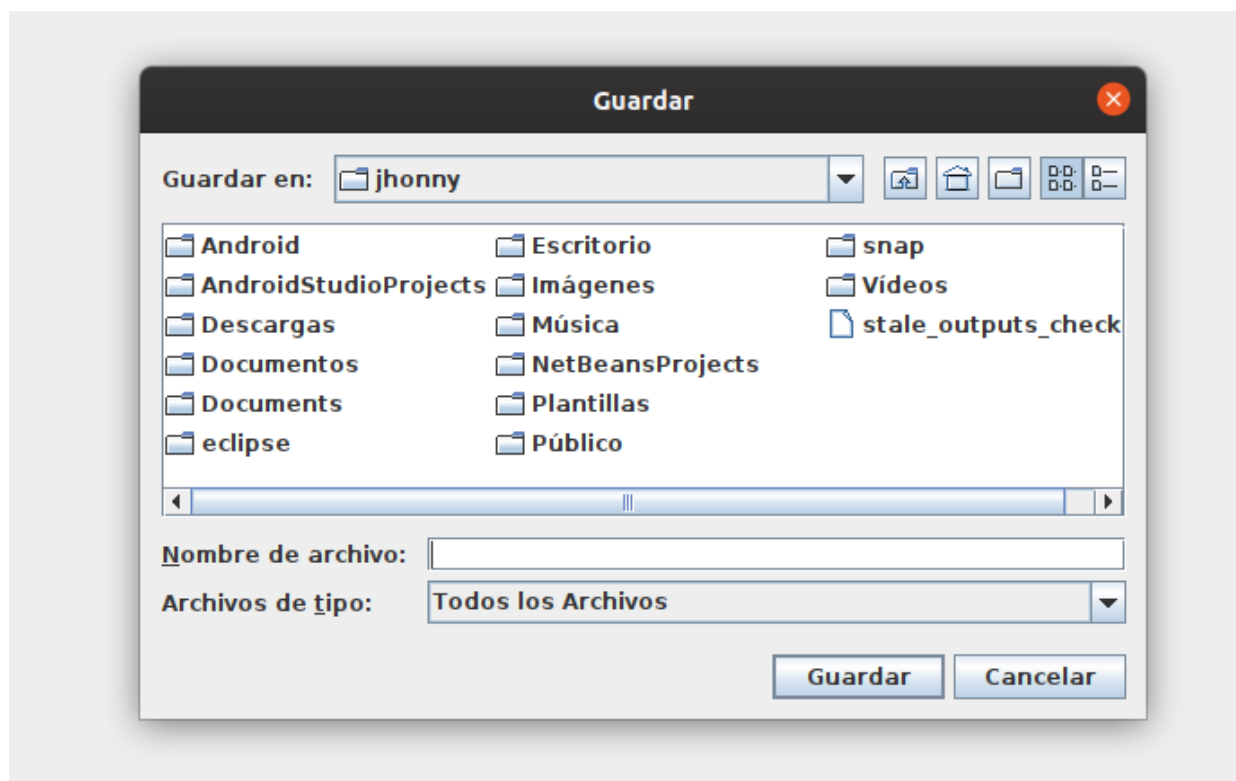
Ejecutar:

1. Ejecutar código intermedio: Ejecuta el código generado.
2. Ejecutar código optimizado: Ejecuta el código intermedio optimizado.
3. Ejecutar assembler: Ejecuta el código assembler.

Nuevo documento: crea un archivo nuevo con extension mlg.



Guardar como y abrir archivo : con la variante de que para abrir un archivo permite seleccionar archivos y guardar como solo directorios.



Editor de texto, para modificar o visualizar la entrada .mlg


```

1 %%VB
2 Public Function Incremento(ByVal Valor As Integer) As Integer
3     Valor = Valor + 1
4     Return Valor
5 End Function
6
7
8
9
10 %%JAVA
11
12 %%PY
13 def Mensaje(tipo):
14     if tipo == 1:
15         print("Arreglo antes de ordenarse")
16     elif tipo == 2:
17         print("Arreglo depues de ordenarse")
18     else:
19         print("default")
20
21 def Mostrar(indice, valor):
22     print("arreglo[, indice, "] = ",valor)
23
24 %%PROGRAMA
25 /* -----
26             Seccion de Librerias
27 ----- */
28 #include "VB"
29 #include "PY"
30
31 /* -----
32     Declaracion de Constantes
33 ----- */
34
35 int tamano;
36
37 /* -----
38     Declaracion de Variables Globales
39 ----- */
40
41 int i;

```

Codigo Intermedio:

```

1 void VB_Incremento_int_int(){
2   op_1=punteroInt+0;
3   valTmp1_Int=stack_Int[op_1];
4   int_num0=valTmp1_Int+1;
5   p1=punteroInt + 0;
6   stack_Int[p1]=int_num0;
7   op_1=punteroInt+0;
8   valTmp1_Int=stack_Int[op_1];
9   op_1=punteroInt-1;
10  stack_Int[op_1]=valTmp1_Int;
11  goto E1;;
12  E1;;
13  };
14  void PY_Mensaje_1(){
15    p2=punteroDouble + 0;
16    valTmp1_Double=stack_Double[p2];
17    num_tmp_float=valTmp1_Double;
18    if ( num_tmp_float == 1 ) goto E2;
19    p2=punteroDouble + 0;
20    valTmp1_Double=stack_Double[p2];
21    num_tmp_float=valTmp1_Double;
22    if ( num_tmp_float == 2 ) goto E3;
23    goto E5;;
24    E2;;
25    printf("Arreglo antes de ordenarse");
26    goto E6;;
27    E3;;
28    printf("Arreglo depues de ordenarse");
29    goto E6;;
30    E5;;
31    printf("default");
32    goto E6;;
33    E6;;
34  };
35  void PY_Mostrar_2(){
36    printf("arreglo");
37    p1=punteroDouble + 0;
38    printf("%f",stack_Double[p1]);
39    printf("] = ");
40    p1=punteroDouble + 1;
41    printf("%f",stack_Double[p1]);
42  }.

```

Codigo en ejecución:

```

jhonny@jhonny-Inspiron-14-3467:~/Escritorio/6To.Semestre/Compi2/LabCompi/Proyecto_2/compilador$ ./codigo
Arreglo antes de ordenarsearreglo[0.000000] = 7.000000arreglo[1.000000] = 14.000000arreglo[2.000000] = 18.000000arreglo[3.000000] = 19.000000
arreglo[4.000000] = 21.000000arreglo[5.000000] = 2.000000arreglo[6.000000] = 9.000000arreglo[7.000000] = 24.000000arreglo[8.000000] = 29.000000
arreglo[9.000000] = 47.000000Arreglo depues de ordenarsearreglo[0.000000] = 47.000000arreglo[1.000000] = 29.000000arreglo[2.000000] = 24.000000
arreglo[3.000000] = 21.000000arreglo[4.000000] = 19.000000arreglo[5.000000] = 18.000000arreglo[6.000000] = 14.000000arreglo[7.000000] = 9.000000
arreglo[8.000000] = 7.000000arreglo[9.000000] = 2.000000

```

Codigo optimizado, en representacion de cuartetos para hacerlo más identificable.

El dato en la columna “codigo optimizado” el que se encuentra en color rojo es el que sera removido.

#	Codigo Intermedio	Codigo Optimizado	Dato
1	void VB_Incremento_int_int() {	void VB_Incremento_int_int() {	
2	int_num0=Valor+1;	Valor=Valor+1;	
3	Valor=int_num0;	Valor=int_num0;	La asignacion puede asignarse de una vez, a la instruccion anterior
4	return Valor	return Valor	
5	goto E1;	goto E1;	
6	E1:	E1:	
7	;}	;}	
8	void PY_Mensaje_1() {	void PY_Mensaje_1() {	
9	num_tmp_float=tipo;	num_tmp_float=tipo;	
10	if(num_tmp_float == 1)goto E2;	if(num_tmp_float == 1)goto E2;	
11	num_tmp_float=tipo;	num_tmp_float=tipo;	
12	if(num tmp float == 2)goto E3;	if(num tmp float == 2)goto E3;	

De la misma manera se puede ver el codigo ejecutable del codigo optimizado:

```
jhonny@jhonny-Inspiron-14-3467:~/Escritorio/6To.Semestre/Compi2/LabCompl/Proyecto_2/compilador$ ./codigo
Arreglo antes de ordenarsearreglo[0.000000] = 7.000000arreglo[1.000000] = 14.000000arreglo[2.000000] = 18.000000arreglo[3.000000] = 19.000000a
rreglo[4.000000] = 21.000000arreglo[5.000000] = 2.000000arreglo[6.000000] = 9.000000arreglo[7.000000] = 24.000000arreglo[8.000000] = 29.000000
arreglo[9.000000] = 47.000000Arreglo despues de ordenarsearreglo[0.000000] = 47.000000arreglo[1.000000] = 29.000000arreglo[2.000000] = 24.000000
arreglo[3.000000] = 21.000000arreglo[4.000000] = 19.000000arreglo[5.000000] = 18.000000arreglo[6.000000] = 14.000000arreglo[7.000000] = 9.000
000arreglo[8.000000] = 7.000000arreglo[9.000000] = 2.00000010
jhonny@jhonny-Inspiron-14-3467:~/Escritorio/6To.Semestre/Compi2/LabCompl/Proyecto_2/compilador$
```

Si existe algun error (lexico, sintactico o semantico) se despliega una ventana indicando ciertas características.

Fila	Columna	Token	Extra	Tipo error
76	10	j	La variable no ha sido ini...	Semantico
79	16	j	La variable no ha sido ini...	Semantico
79	29	j	La variable no ha sido ini...	Semantico
81	35	j	La variable no ha sido ini...	Semantico
82	42	j	La variable no ha sido ini...	Semantico