Universidad San Carlos de Guatemala

Facultad de Ingeniería

Ciencias y Sistemas

Organización de Lenguajes y Compiladores 1

Manual Usuario Compilador

Natthaliee María Molina Cruz

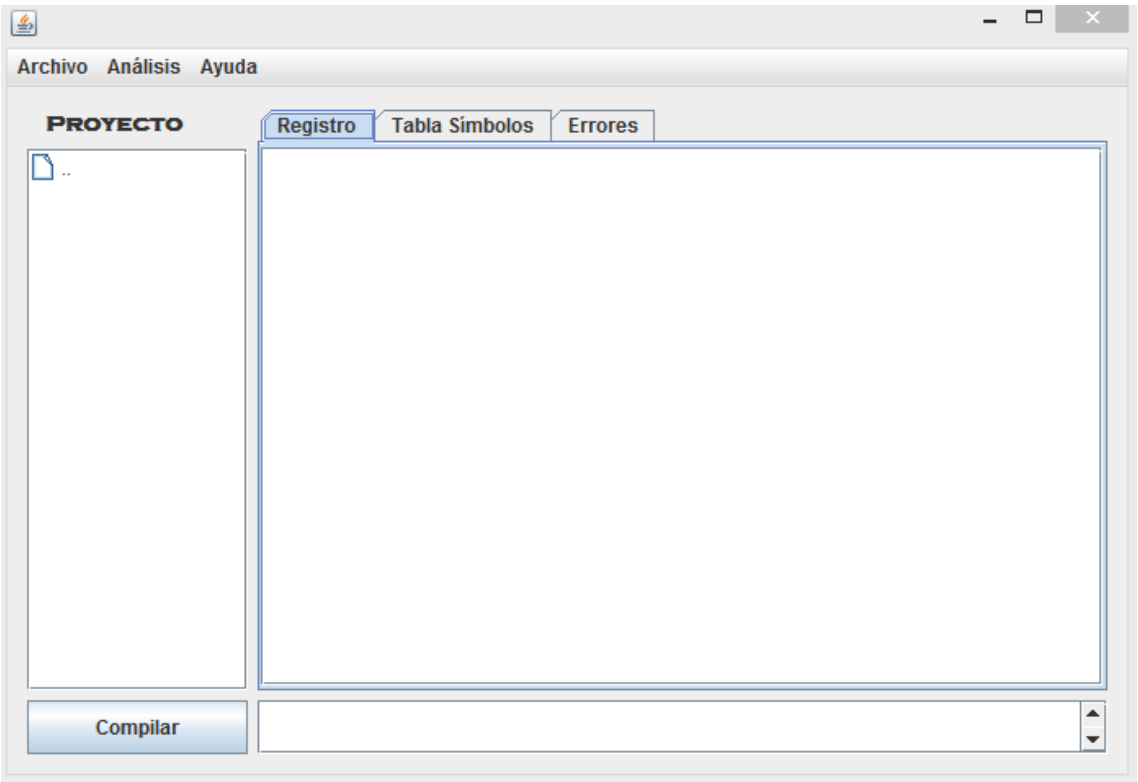
Carnet: 201212501

Guatemala, 10 de Enero 2014

Compilador

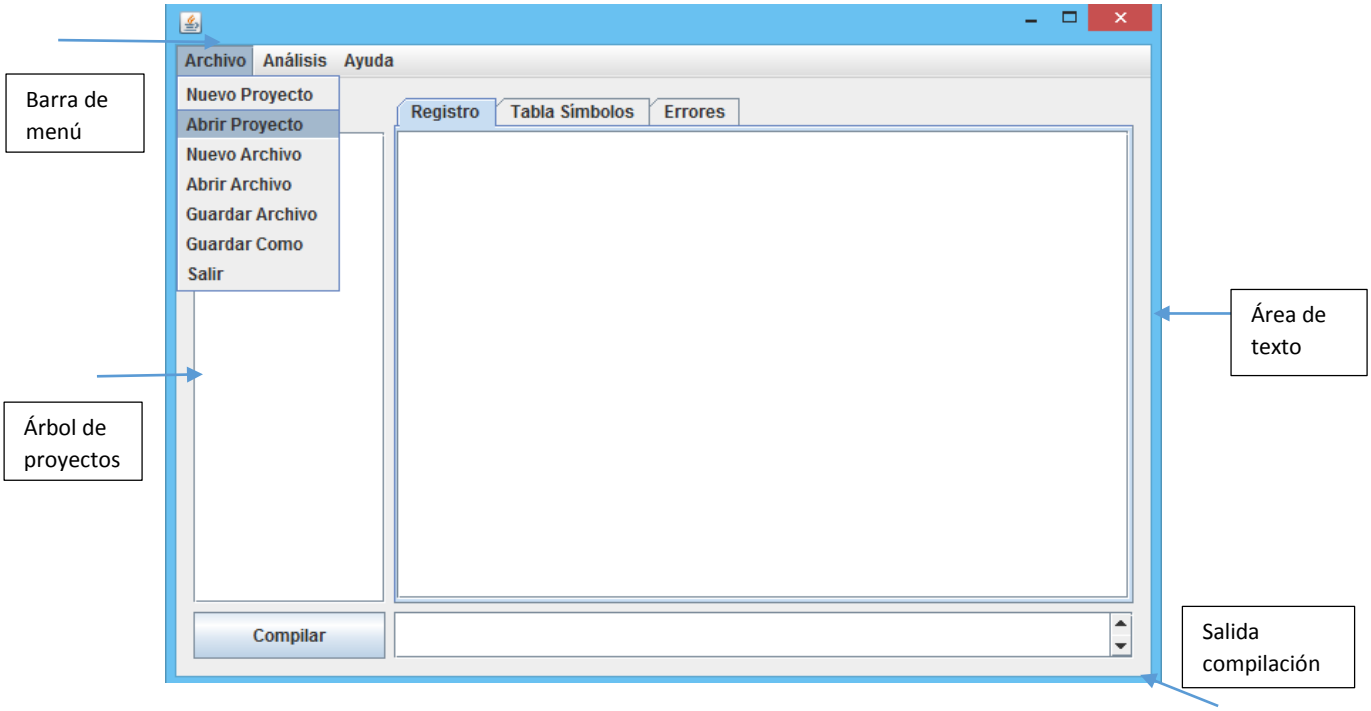
Con la nueva aplicación Compilador se podrán crear interfaces y estructuras, definir funciones, crear registros y exportar a un archivo externo con extensión csv. Si es tu primera experiencia te recomendamos leer el manual completo.

Al iniciar la aplicación se despliega la ventana principal el cual se podrá cargar los archivos de texto y luego proceder a compilarlos para la ejecución de las acciones definidas por el usuario.

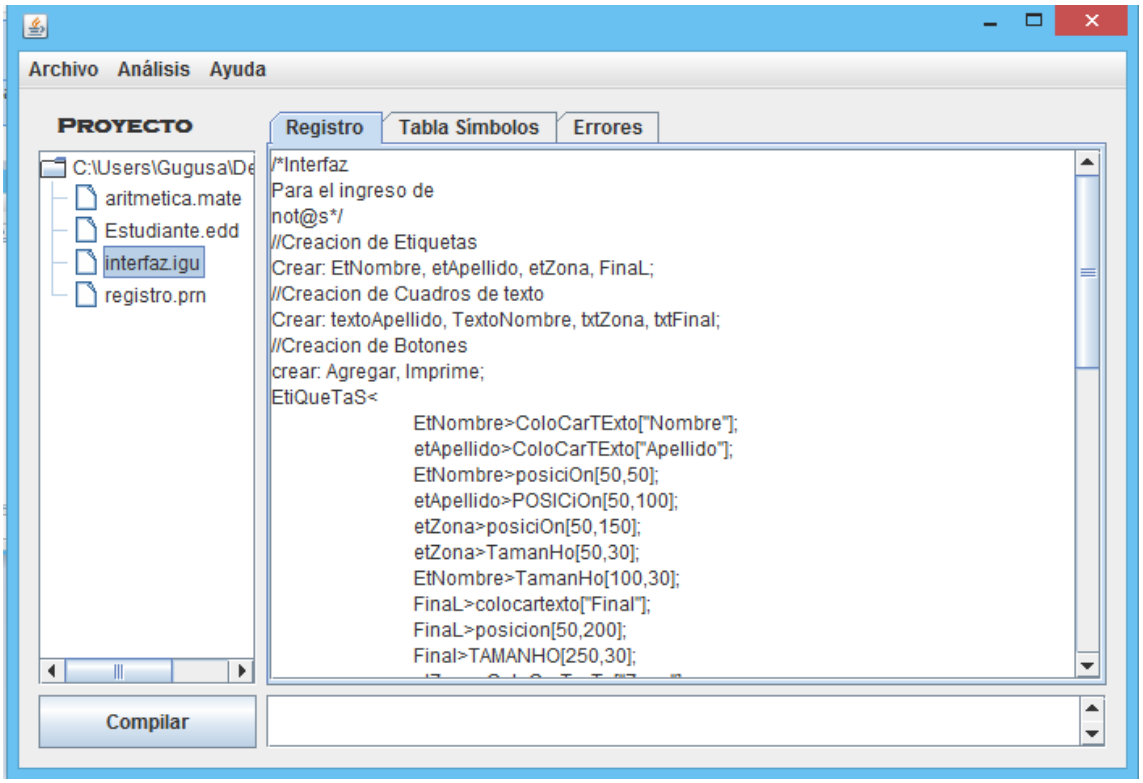


La aplicación abre archivos con extensión “igu” que realiza la interfaz gráfica que se mostrará al usuario, extensión “mate” el cual se encarga de realizar operaciones matemáticas, con extensión “edd” donde se definen las estructuras tipo pila, cola, arreglos o creadas por el usuario, y extensión “rpn” el cual contiene el archivo principal encargado de ejecutar todos los métodos.

Al abrir un proyecto todos los archivos que contiene la raíz se ejecutar al mismo tiempo en el menú de análisis/compilar, y con el botón de compilar se compilará lo q se encuentre dentro del área de texto de registro.

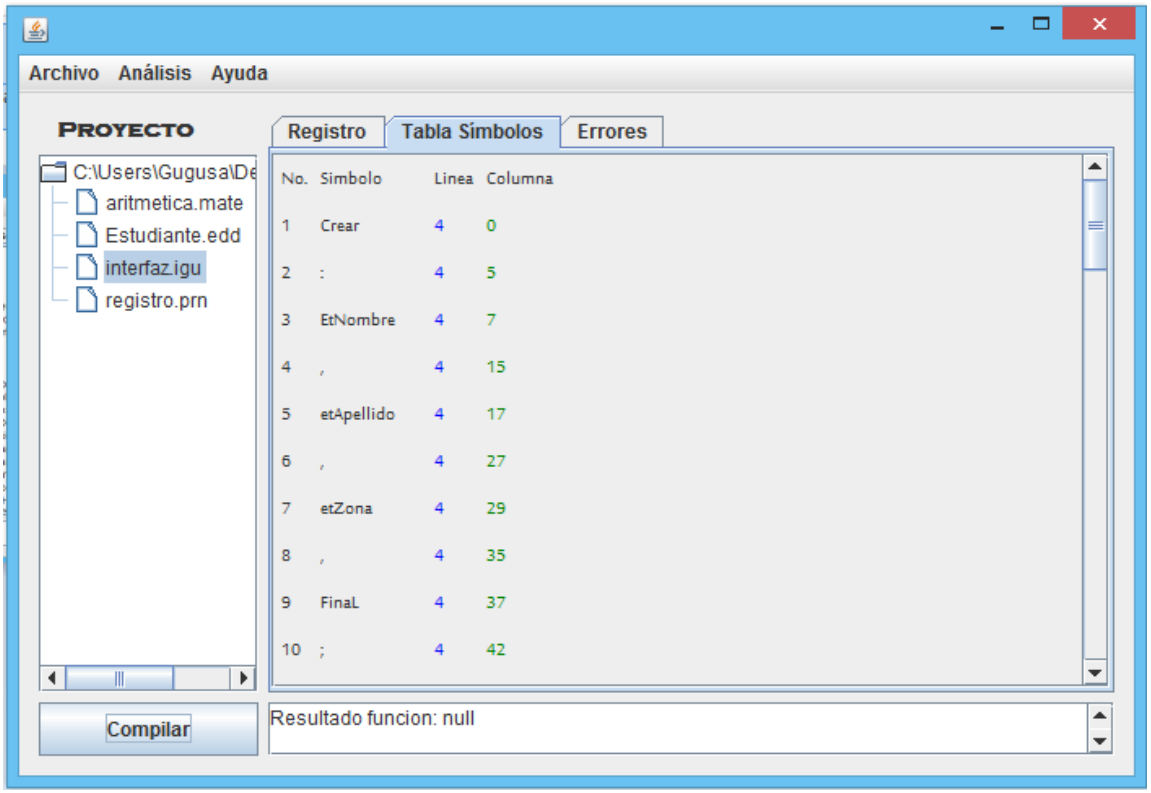


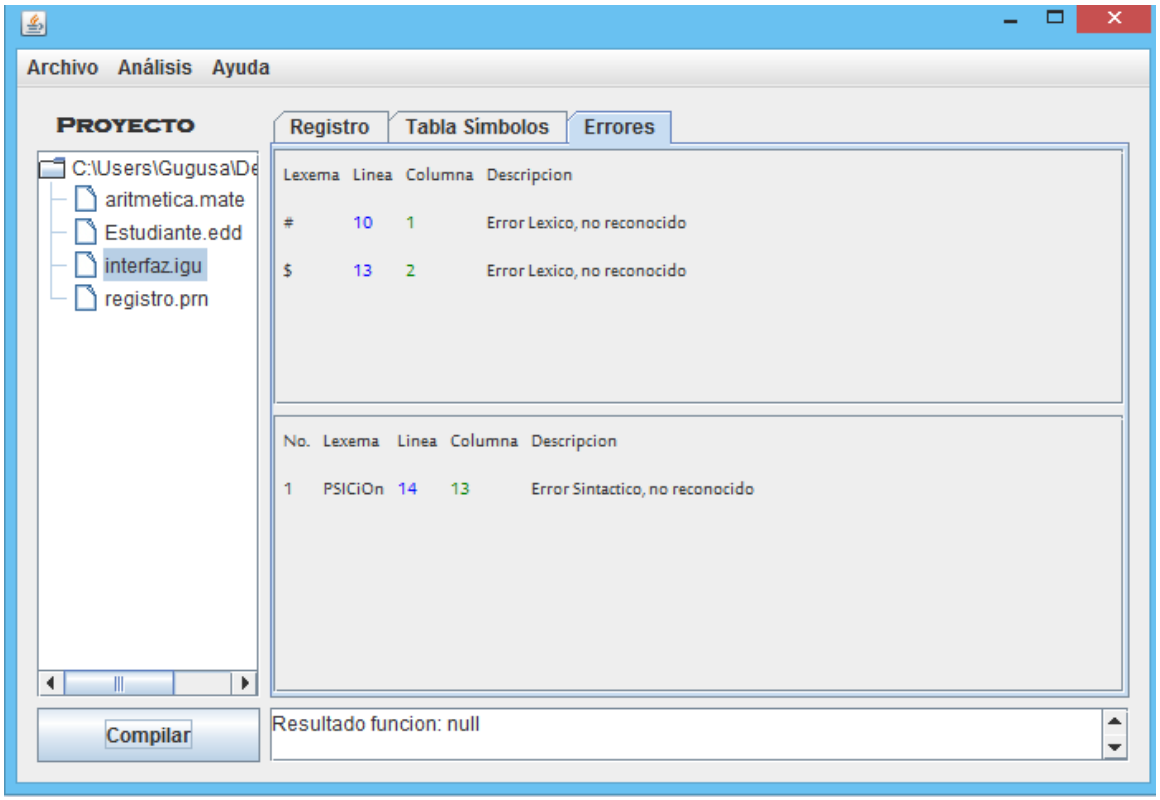
La aplicación contiene un menú de archivo para crear un nuevo proyecto, abrir un proyecto anteriormente contruido, agregar archivos al proyecto, guardarlos y salir de la aplicación.



En el cuadro de “Registro” se ingresar el texto que se desea compilar y se procede a hacer click en el botón compilar.

Incluye también un cuadro con la tabla de símbolos que se generará al momento de compilar un archivo o lo que se encuentre en el área de texto y un cuadro de errores que muestra al usuario los error léxicos y sintácticos en caso de ser encontrados durante el análisis de los archivos ejecutados.





Estructura de los archivos que pueden ser ejecutados por el usuario

Archivo Interfaz:

```
/*Interfaz
Para el ingreso de
not@s*/
//Creacion de Etiquetas
Crear: EtNombre, etApellido, etZona, FinaL;
//Creacion de Cuadros de texto
Crear: textoApellido, TextoNombre, txtZona, txtFinal;
//Creacion de Botones
crear: Agregar, Imprime;
EtiQueTaS<
        EtNombre>ColoCarTExto["Nombre"];
        etApellido>ColoCarTExto["Apellido"];
        EtNombre>posiciOn[50,50];
        etApellido>POSICiOn[50,100];
        etZona>posiciOn[50,150];
        etZona>TamanHo[50,30];
        EtNombre>TamanHo[100,30];
        FinaL>colocartexto["Final"];
        FinaL>posicion[50,200];
        Final>TAMANHO[250,30];
        etZona>ColoCarTexTo["Zona"];
>
Cuadros_TEXTO<
        TextoNombre>ColoCarTExto[""];
        TextoNombre>posicion[110,50];
        TextoNombre>TamanHo[200,30];
        textoApellido>ColoCarTExto[""];
        textoApellido>POSICION[110,100];
        textoApellido>TamanHo[200,30];
        txtZona>ColoCarTExto[""];
        txtZona>POSICION[110,150];
        txtZona>TamanHo[200,30];
        txtFinal>ColoCarTExto[""];
        txtFinal>POSICION[110,200];
        txtFinal>TamanHo[200,30];
>


Botones<
Agregar>ColoCarTExto["Registrar"];
Imprime>ColoCarTExto["Imprime"];
Imprime>POSICION[200,250];
Agregar>posicion[50,250];
>

Ventana<
TITULo["Registro"];
Tamanho[400,400];
>
```

Archivo Mate:

```
InCluir<<Estudiante.edd>>;
VariablEs:{
	anho :: 2013; //Asignación de variable
}
Funciones:{
	Def Suma(num1,num2,num3)
		RetOrnO :: (num1 (num2 num3 +) +);
	FIn;

	//Polimorfismo
	Def Suma(num1,num2)
		ReTorno :: (num1 num2 +);
	FiN;
}
```

Archivo Estructura:

```
/*Creación de variables*/
/*Estructura definida por el usuario de nombre "Estudiante"*/
ESTrucTura::persona{
EnTero edad<
CadeNa nombre<
cadena apellido
Entero anho<
EnterO zona<
ENTeRO final<
}
/*Arreglo de estudiantes de 50 posiciones*/
Arreglo<persona> stdnts[50]<
/*Definicion de Funciones*/
FunCIon::vacio insertarEstudiante(ENterO indice, cadena nombre, cadena apellido, entero zona, entero
final):entero{
stdnts[índice]@nombre -> nombre<
stdnts[índice]@apellido -> apellido<
stdnts[índice]@zona -> zona<
stdnts[índice]@final -> final<
}
Funcion::entero obtenerAnho(entero indice):entero{
retorno -> prsns[índice]@edad<
}
Entero c1,c2,c3<
FunCIOn::cadena obtenerNombre(entero índice){
retorno -> prsns[índice]@nombre<
}
FuncIOn::cadena obtenerApellido(entero índice){
retorno -> prsns[índice]@apellido<
}
fuNcIOn::entero obtenerZona(entero índice){
retorno -> prsns[índice]@zona<
}
fUncIOn::entero obtenerFinal(entero índice){
retorno -> prsns[índice]@final<
}
```