

Universidad San Carlos de Guatemala

Facultad de Ingeniería

Ciencias y Sistemas

Organización de Lenguajes y Compiladores 1

Manual Técnico Compiladores

Natthaliee María Molina Cruz

Carnet: 201212501

Guatemala, 10 de Enero 2015

TABLA DE CONTENIDOS

1.	Introducción.....	1
2.	Objetivos.....	1
3.	Descripción.....	1
4.	Requerimientos y Restricciones.....	2
5.	Prácticas de Programación	2
6.	Tipos de Clases.....	2-3
7.	Glosario	4
8.	Interfaz de Usuario.....	5

Compilador

Introducción:

Compilador es una aplicación de un compilador cuyo objetivo es analizar los archivos ingresados por el usuario y ejecutar las acciones contenidas en él, como la creación de funciones, estructuras y el ingreso de registros para ser exportados en un archivo con extensión csv. A continuación se describen las restricciones y las características de cada función que está contenida en la aplicación.

El programa fue diseñado en el lenguaje de programación Java con base a los conocimientos adquiridos en la clase de Organización de Lenguajes y Compiladores 1.

Objetivos:

- Introducir en la construcción e implementación de análisis léxico y sintáctico en las herramientas de jFlex y CUP.
- Aplicar los conocimientos adquiridos en la clase de Organización de Lenguajes y Compiladores 1.
- Comprender y utilizar alguna herramienta de desarrollo para la creación del software.
- Aplicar correctamente el uso de una tabla de símbolos y el manejo de errores.

Descripción:

Compilador es una aplicación de escritorio diseñado en Netbeans utilizando las herramientas de JFlex y CUP. El objetivo es la creación de interfaces, funciones y estructuras que podrán ser ejecutadas a través de un archivo principal. También se pueden crear variables. Estructuras de tipo pila, cola, arreglos y definidas por el usuario.

Requerimientos y Restricciones:

- Cualquier herramienta de desarrollo(IDE).
- jFlex y CUP para el análisis léxico y sintáctico.
- Al leer los comandos la aplicación debe actuar como intérprete, y leyendo carácter por carácter.

Prácticas de Programación:

1. Realizar un análisis estructurado y lógico del problema para lograr resolverlo de manera sencilla y práctica.
2. Construir el algoritmo de la aplicación para evitar errores comunes y encontrar los pasos a seguir para realizar cada tarea, encontrando también los requerimientos y restricciones posibles dentro de la aplicación.
3. Planificar un cronograma de actividades para realizar cada proceso y método en el menor tiempo posible.
4. Realizar la compilación verificando y corrigiendo los errores de programación.

Tipos de Clase:

Cada una de las clases contienen modificadores de tipo público o privado.

- Ingreso.java:
Esta interfaz contiene los menús de archivo, análisis y ayuda y los cuadros de registro que contiene un área de texto para ingresar archivos a compilar, tabla de símbolos y errores que se mostrarán al usuario.
- Notas.java:
Realiza una instancia de la clase “Ingreso” y hace visible la interfaz.

- Analisis_lexico.java:
Contiene los métodos encargados del análisis léxico generado por JFlex.
- Analisis_sintactico.java:
Contiene los métodos encargados del análisis sintáctico generado por CUP.
- Mate.java:
Encargado de las operaciones matemáticas que el usuario ingresará en el archivo con extensión mate.
- Interfaz.java:
Se encarga de crear la interfaz definida por el usuario en el archivo con extensión igu.
- Proyecto.java:
Contiene la estructura del árbol de proyectos que se mostrará en la interfaz principal al usuario.

Tabla de Símbolos y Errores:

Deben ser mostrados dentro de la aplicación con formato HTML. Ejemplo:

No.	Lexema	Token	Posición	Error
1	%	Desconocido	Fila 3 columna 4	Error léxico, token no reconocido.

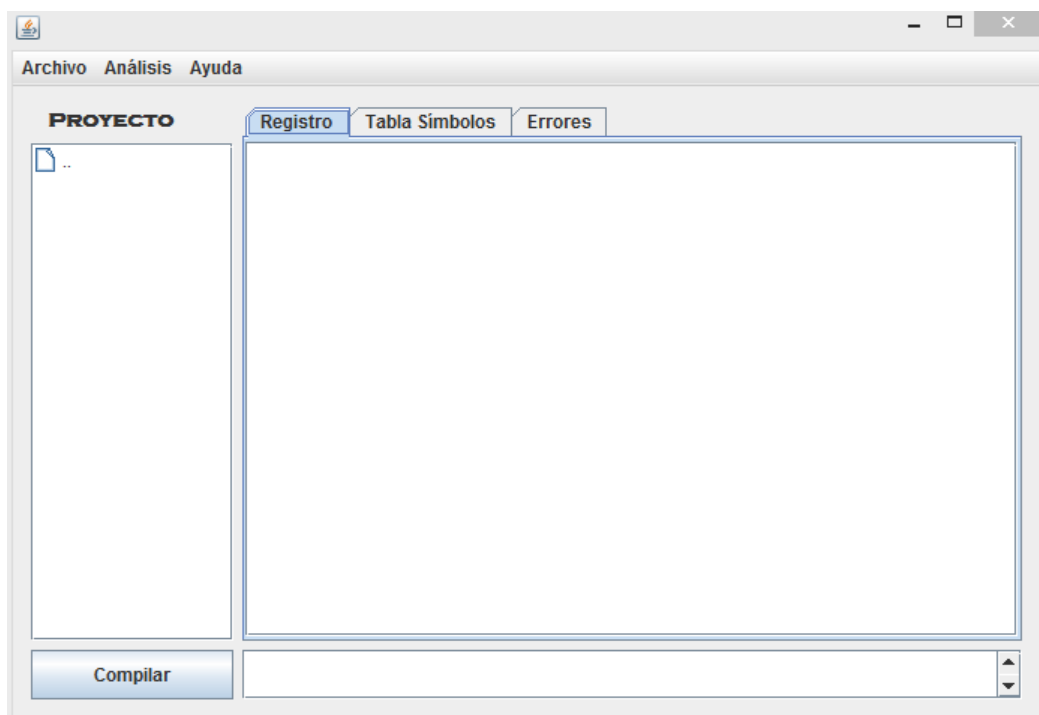
Glosario:

- La gramática es un ente formal para especificar, de una manera finita, el conjunto de símbolos que constituye un lenguaje, así tenemos la gramática del español, del inglés, del francés, etc.
- Gramática regular: Formada por cuatro elementos, y son $G = \{\Sigma, N, P, S_0\}$

- Elemento:

	Nombre	Descripción
Σ	alfabeto (Terminales)	Símbolo o carácter permitido en el lenguaje específico, siempre entre apóstrofes ('). También se permiten conjuntos de terminales, previamente definidos (se explica la sintaxis posteriormente).
N	No terminales	Símbolos auxiliares que no son parte de los caracteres permitidos.
P	Reglas de producción	Las producciones para las gramáticas regulares pueden ser de la siguiente forma: $A \rightarrow 'a'B$ $A \rightarrow B'a'$ $A \rightarrow 'a'$ $A \rightarrow \epsilon;$ Dónde: No terminales = {A,B} y Terminales = {'a'}
So	Inicio	So es un No terminal.

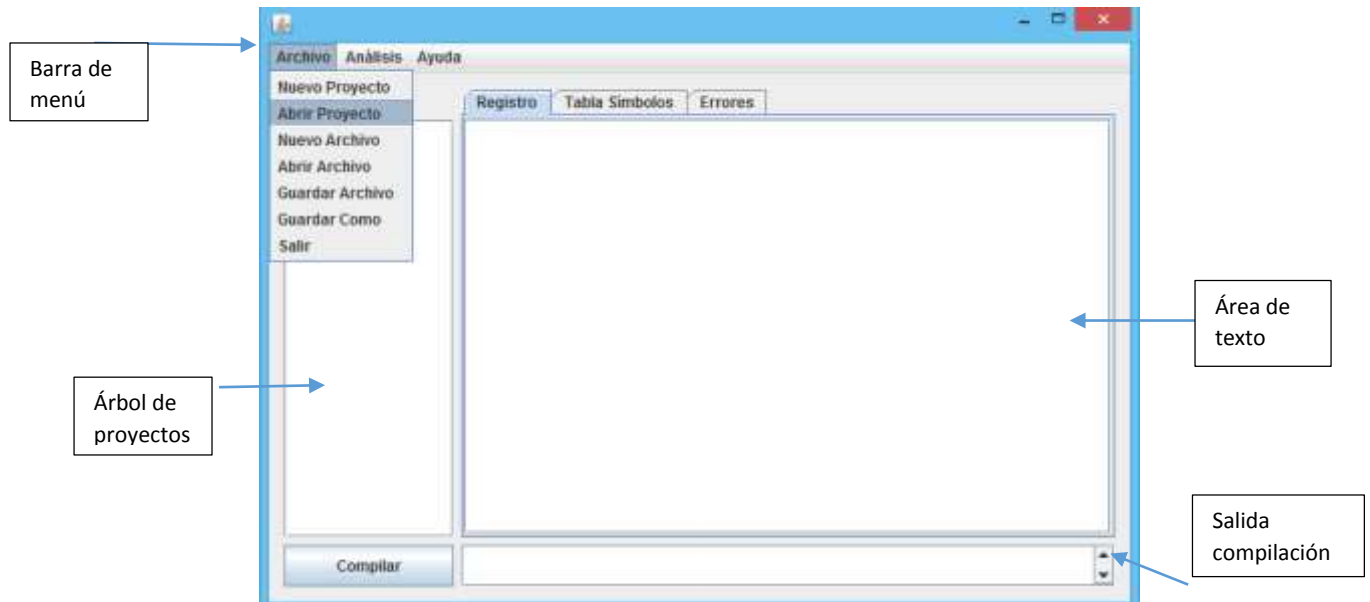
Interfaz de Usuario:



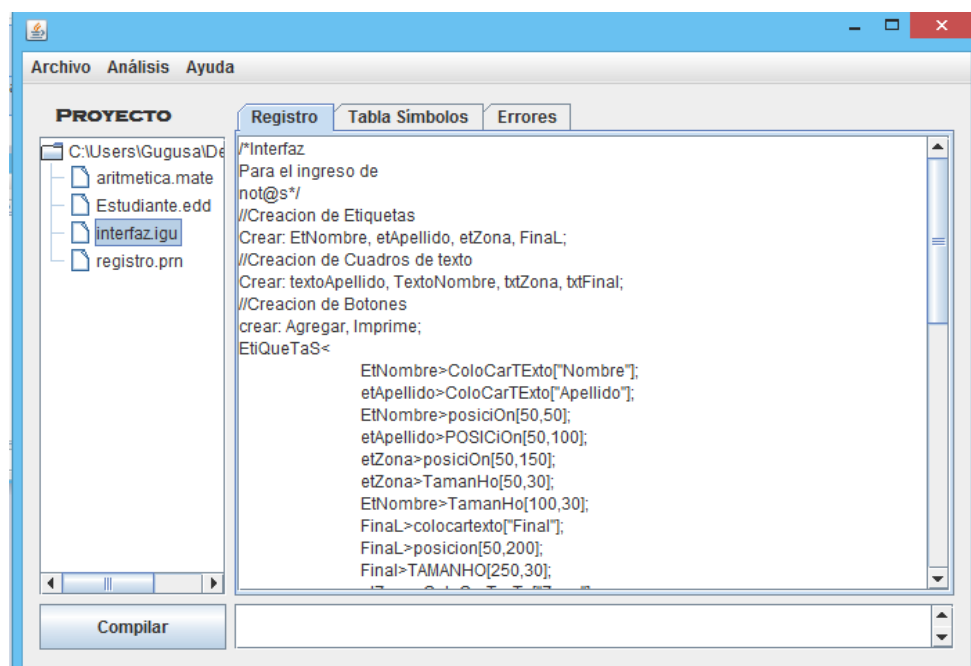
Secuencia de pantallas:

La aplicación contiene un menú de archivo para crear nuevos proyectos, abrir proyectos ya definidos, agregar archivos al proyecto, los cuales serán agregados al árbol de proyectos.

También incluye las funciones de guardar y salir de la aplicación.



El cuadro de "Registro" se utilizará para agregar los archivos y compilarlos independientemente al hacer click en el botón compilar. La aplicación abre archivos con extensión "igu" que realiza la interfaz gráfica, extensión "mate" el cual se encarga de realizar operaciones matemáticas, con extensión "edd" donde se definen las, y extensión "rpn" el cual contiene el archivo principal encargado de ejecutar todos los métodos.



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.

