**Tecnológico de Costa Rica**

**1° Proyecto**

**Analizador Léxico**

**Curso: Compiladores**

**Profesor**

**Allan Rodríguez Davila**

**Estudiantes**

**Aarón Piñar Mora**

**Cristopher Zúñiga Jiménez**

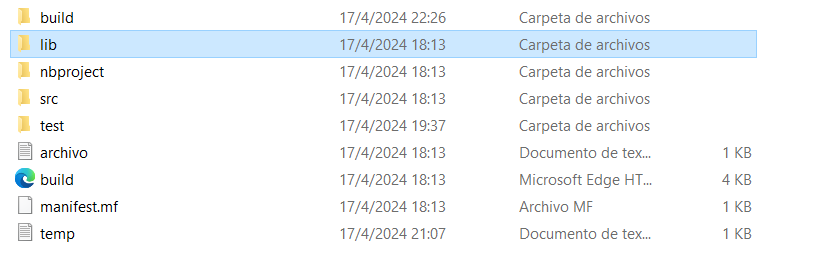
**2023**

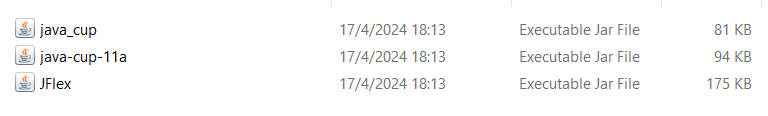
# Descripción del Problema

Un grupo de desarrolladores desea crear un nuevo lenguaje imperativo, ligero, que le permita realizar operaciones básicas para la configuración de chips, ya que esta es una industria que sigue creciendo constantemente, y cada vez estos chips necesitan ser configurados por lenguajes más ligeros y potentes. Es por esto por lo que este grupo de desarrolladores requiere desarrollar su propio lenguaje para el desarrollo de sistemas empotrados.

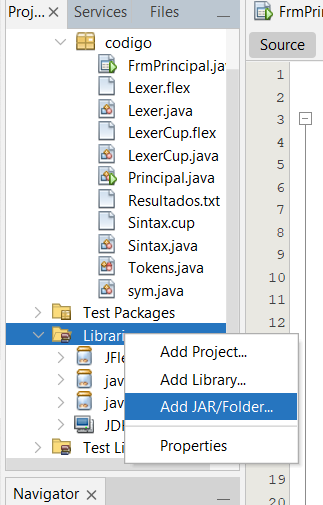
# Manual de Usuario

Se debe de tener instalado Apache Net Beans para poder correr el programa y las librerías JFlex yJCup. Las cuales fueron agregadas en la carpeta lib donde se encuentra este proyecto.

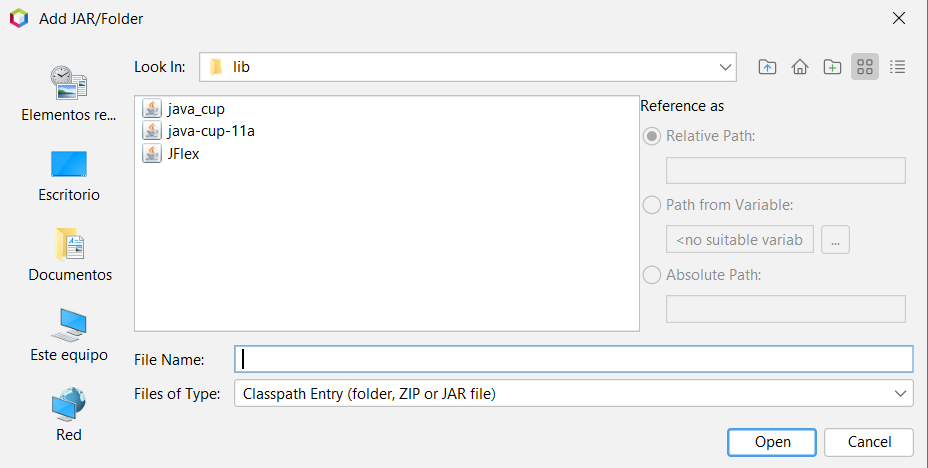




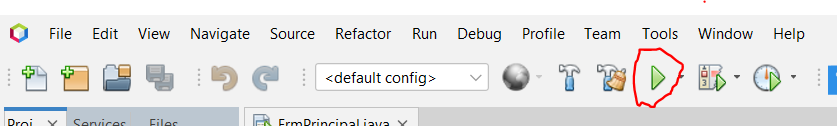
Este paso debe hacerse únicamente si neatBeans tira error en reconocimiento de las librerías. Se debe de cargar las librerías en la carpeta lib de la siguiente forma y deben de cargarse en el siguiente orden para que no existan conflictos a la hora de funcionar el programa.



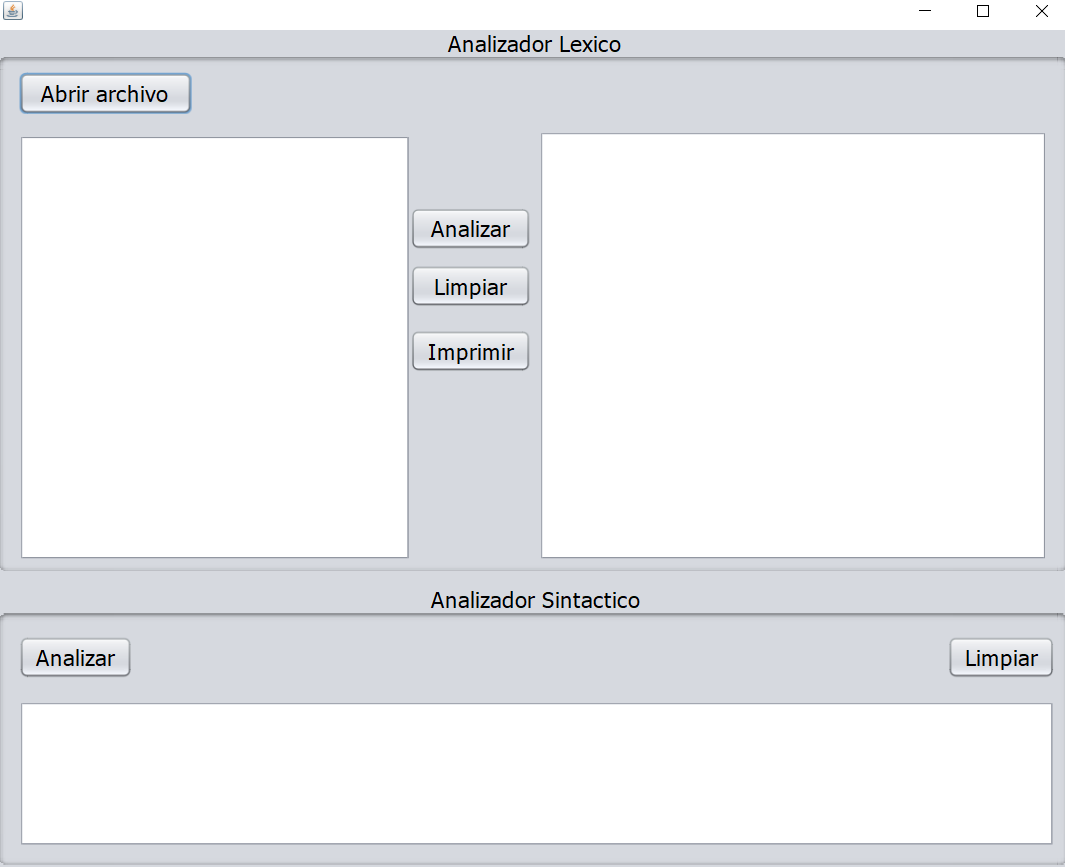
Una vez dado click derecho en add JAR/Folder aparecerá la siguiente ventana emergente donde se deberá cargar primero el JFlex luego java cup 11 y por último java cup. Para ello, buscar carpeta lib y cargar los archivos con el botón Open.



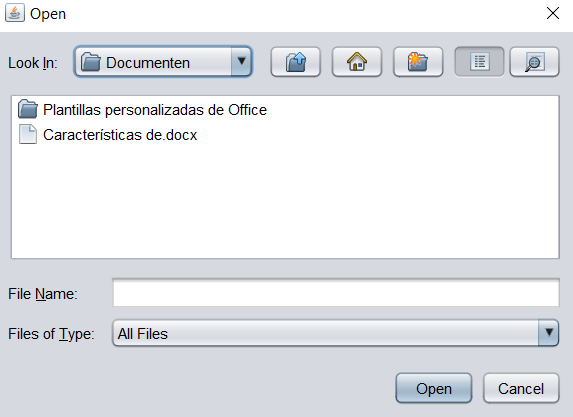
Una vez instalado lo anterior mencionado se puede correr la aplicación utilizando el siguiente botón encerrado en un círculo rojo.



A continuación, se abrirá la siguiente ventana emergente.



Se puede seleccionar un archivo, donde abrirá una ventana emergente y pedirá buscar la ruta del archivo. Como se visualiza en la siguiente imagen.



Una vez puesto se le da al botón del medio que dice analizar y hará el análisis correspondiente y generando un archivo nuevo con resultados del análisis.

# Pruebas de Funcionalidad

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Muestra de análisis de expresiones

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Botón de limpiar Hoja para reseter el analizador léxico.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Archivo generado a partir del análisis léxico previo donde se muestra lo solicitado.

# Diseño del Programa

* Búsqueda correlacionada de los empaquetados para mostrar los tokens y su símbolo en base a la tabla.
* Búsqueda individual para corrección de precedencias. Lo cual, dependiendo del bloque lograba liberar gran cantidad de estas. Lo cual, se buscaba tratar de arreglar aquellas que anidaban más.
* Desarrollo de interfaz para la extracción del archivo mas cómoda para el usuario.
* Se creó en base a la gramática desarrollada en la tarea y se realizaron los cambios propuestos por el profesor. Además, se hizo replanteamiento por problemas por la precedencia en algunos casos.
* Desarrollo de un switch case para la interfaz grafica dando mensajes personalizables a la hora de mostrar lexemas y sus símbolos en pantalla.

# Librerías Usadas

* JFlex
* JCup

# Análisis de Resultados

Se llega a la conclusión de que realiza el análisis léxico correspondiente y demarca aquellos que no corresponden a la gramática propuesta. Pero, se presentaron problemas con la precedencia de algunos lexemas y la ambigüedad del código, eso llevando a una gran cantidad de warnings a la hora de tratar de ejecutar el cup, múltiple de los errores vienen por parte de las expresiones matemáticas debido a que su producción es similar una con otra, se intentó la corrección de estos warnings mas no se lograron solucionar todos por lo tanto el analizar sintáctico no se encuentra funcional. El analizador léxico y tabla de símbolos funciona como corresponde.

# Bitácora

GitHub: [*16pixar/Proyecto-Compi (github.com)*](https://github.com/16pixar/Proyecto-Compi)