

Universidad De San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ciencias y Sistemas

Laboratorio organización de lenguajes y compiladores 1  
Sección "C"



# **“MANUAL TÉCNICO”**

Diego André Mazariegos Barrientos

Carné: 202003975

# Objetivos

## General:

Brindar al lector una guía que contenga la información del manejo de clases, atributos, métodos y del desarrollo de la interfaz gráfica para facilitar futuras actualizaciones y futuras modificaciones realizadas por terceros.

## Específicos:

- Mostrar al lector una descripción lo más completa y detallada posible del SO, IDE entre otros utilizados para el desarrollo de la aplicación.
- Proporcionar al lector una concepción y explicación técnica - formal de los procesos y relaciones entre métodos y atributos que conforman la parte operativa de la aplicación.

# Introducción

Este manual técnico tiene como finalidad dar a conocer al lector que pueda requerir hacer modificaciones futuras al software el desarrollo de la aplicación denominada “OLC-P1-202003975” desarrollada durante el transcurso de las semanas de enero y marzo del año 2022, indicando el IDE utilizado para su creación, su versión, requerimientos del sistema, etc...

La aplicación tiene el objetivo de cumplir con los requerimientos solicitados por el curso de Organización de Lenguajes y Compiladores 1, perteneciente a la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala. Dichos requerimientos incluyen el análisis de los datos de entrada con un formato previamente establecido, la generación de reportes entre otras funcionalidades. Así como, desarrollar el método del árbol y el método de Thompson con los datos ingresados y esto con el fin de que los estudiantes del curso puedan verificar que las respuestas de las tareas y exámenes del curso son correctas. Con dichos datos se despliegan varias funcionalidades para el manejo de la información. Como último la aplicación cuenta con una opción de generación de reportes en formato HTML y JSON.

# Descripción de la Solución

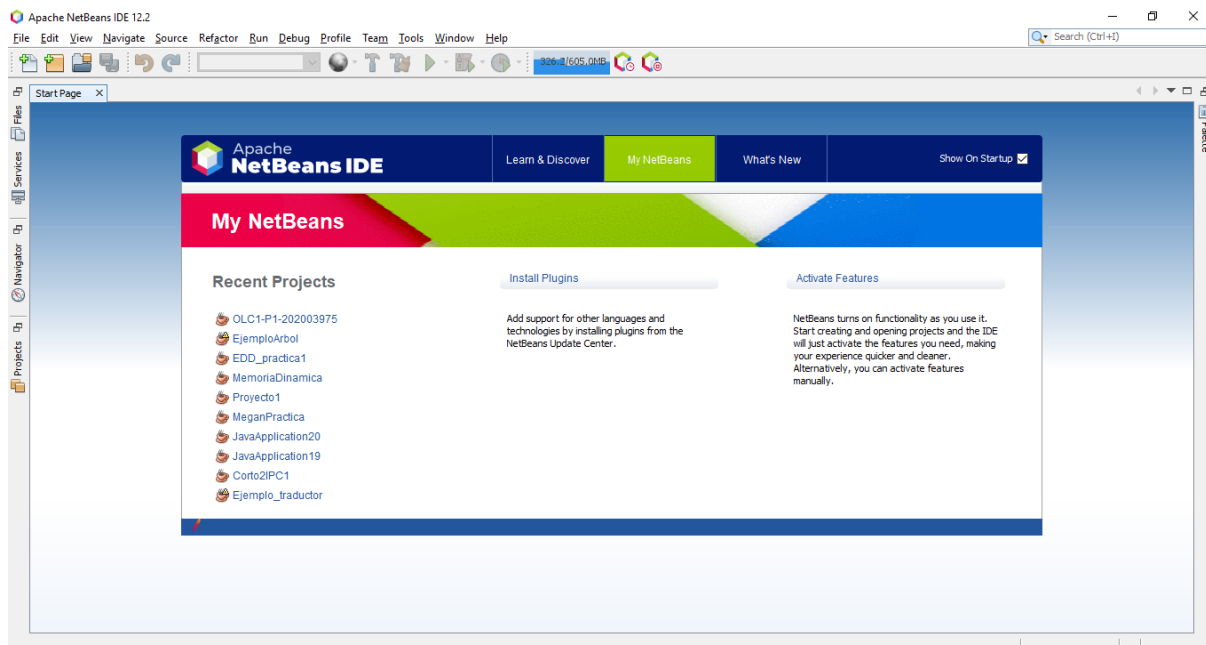
Para poder desarrollar este proyecto se analizó lo que el cliente solicitaba y lo que el cliente realmente necesitaba, sus restricciones tanto humanas, de equipo y financieras del proyecto y empresa; y el ambiente y forma de trabajo de los futuros operadores de la aplicación.

Entre las consideraciones encontramos con mayor prioridad están:

- Realizar la lectura del archivo de entrada con el formato correcto.
- Análisis completo del archivo de entrada una vez verificada que la entrada sea correcta lexicalmente y sintácticamente procediendo a elaborar las respectivas funcionalidades dependiendo de las acciones que tome el usuario.
- Generación de reportes solicitados.
- Presentación de la interfaz de forma agradable y fácil de usar.

# IDE

El IDE con el que se desarrolló el proyecto “OLC-P1-202003975” fue Apache Netbeans IDE 12.2, debido a su apoyo al desarrollador gracias a su asistente que detecta errores semánticos, sintácticos del código por lo cual ayudan y hacen que la duración de la fase de programación sea más corta, además posee una interfaz muy agradable y fácil de entender en el modo debugging. Debido a la experiencia utilizada se recomienda utilizar este IDE para el desarrollo de aplicaciones Java con interfaz gráfica.



## Requerimientos de IDE:

- Los paquetes NetBeans de PHP y C / C ++ solo requieren que Java Runtime Environment (JRE) 8 se instale y ejecute.
- Las características de Java en las características IDE y JavaFX 8 requieren JDK 8.
- Se requiere Java SE 8 para instalar y ejecutar NetBeans IDE 12.2.

# Requisitos del programa

Sistema operativo	Memoria RAM mínima	Memoria RAM recomendada	Espacio en disco mínimo	Espacio en disco recomendado
El programa puede ser instalado en cualquier sistema operativo.	512 MB	8 GB	4.21 MB	1 GB

## Máquina en la cual fue desarrollado el programa

### Especificaciones del dispositivo

HP Laptop

Nombre del dispositivo

Procesador

RAM instalada

Id. del dispositivo

Id. del producto

Tipo de sistema

Lápiz y entrada táctil

Copiar

Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz  
1.19 GHz

8.00 GB (7.70 GB utilizable)

Sistema operativo de 64 bits, procesador x64

Compatibilidad con entrada manuscrita

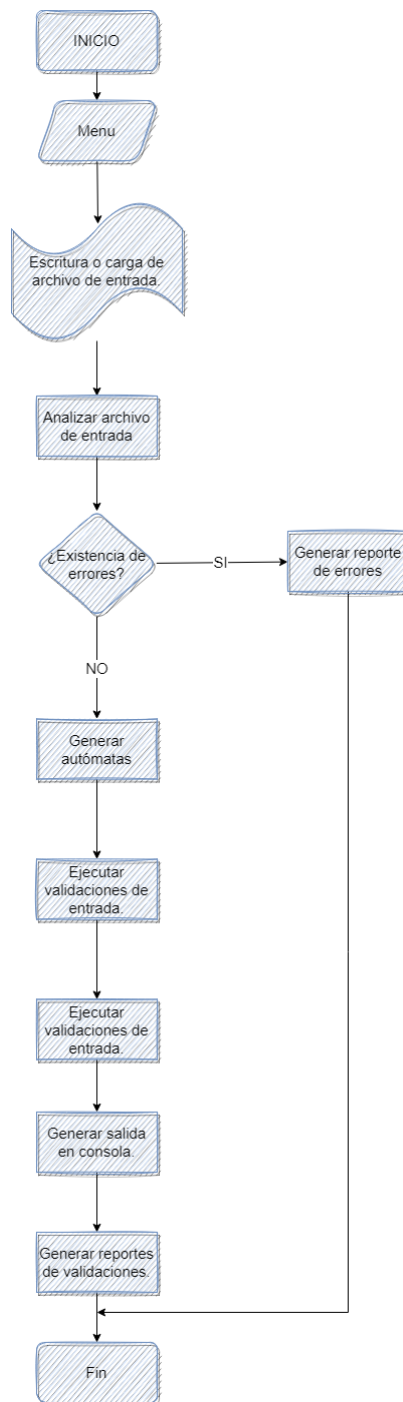
# Librerías Utilizadas

Las librerías utilizadas para el desarrollo de este proyecto fueron:

```
java.awt.Desktop;  
java.io.*;  
org.json  
java cup runtime  
java cup  
jflex full
```

En el caso de la librería `awt` y `io`, se utilizaron para poder acceder al archivo de entrada fácilmente ya que estas despliegan el buscador de archivos y también para el desarrollo de la interfaz gráfica. La librería `org.json` fueron utilizadas para poder escribir ficheros de tipo `Json` con mayor facilidad. La librería `JFlex full` se utilizó para realizar el análisis léxico del archivo de entrada. Y la librería se utilizó para realizar el análisis sintáctico del archivo de entrada. Cabe recalcar que estas dos últimas librerías pueden ser utilizadas conjuntamente.

# Diagrama flujo





# Clases del programa

En el programa se utilizó una gran cantidad de clases, pero entre ellas las principales son:

- **Menu.java**

Es básicamente la clase donde se manejó el control de las variables globales, así como el manejo de la interfaz gráficas y todas sus funcionalidades.

- **TablaSimbolos.java**

Es una clase que se diseñó para poder llevar un registro de todas las variables que aparecieran en el código y así cuando dichas variables fuesen buscadas en algún otro momento esta tabla sirviera como base de datos a la cual hacer la consulta de dicha variable y verificar su existencia y el valor que esta representa.

- **Arbol.java**

Es una clase diseñada para formar la estructura principal del método del árbol, básicamente funciona como cualquiera otra estructura de árbol y en esta se operan ciertos pasos del método del árbol aprovechando de la recursividad y del rendimiento de esta estructura.

- **TablaTransiciones.java**

Clase utilizada para almacenar todos los estados y consigo todas las transiciones realizadas en el método del árbol y de este modo llevar un mejor control del mismo.

- **Instrucción.java**

Es una interfaz diseñada para poder abstraer todas las instrucciones del código de entrada y poder manejar mejor las acciones de dichas instrucciones, esta interfaz posee las siguientes clases derivadas: Asignacion.java, Conjunto.java, Verificacion.java, Porcentajes.java, Operacion.java.