Logotipo, nombre de la empresa

Descripción generada automáticamente

Manual del programador para el sistema Compilador con JAVA

UNIREMINGTON

**Edwin Rivera Ballesteros**

Corporación Universitaria Remington

Programa de Ingeniería de Sistemas

Manizales, Colombia

2024

**Manual del programador para el sistema Compilador con JAVA**

**Versión 1.0**

**Edwin Rivera Ballesteros**

**Asesor:**

Francisco Cardona

Corporación Universitaria Remington

Programa de Ingeniería de Sistemas

Manizales, Colombia

2024

INDICE

[INTRODUCCION 3](#_Toc167738391)

[ARQUITECTURA DEL SISTEMA 4](#_Toc167738392)

[Diagrama de arquitectura 4](#_Toc167738393)

[Tecnologías utilizadas 5](#_Toc167738394)

[REQUISITOS DEL SISTEMA 6](#_Toc167738395)

[INSTALACION 6](#_Toc167738396)

[ESTRUCTURA DEL CODIGO 6](#_Toc167738397)

[Organización del proyecto 6](#_Toc167738398)

[Nombramiento de variables 8](#_Toc167738399)

[Indentación y espacios 8](#_Toc167738400)

# INTRODUCCION

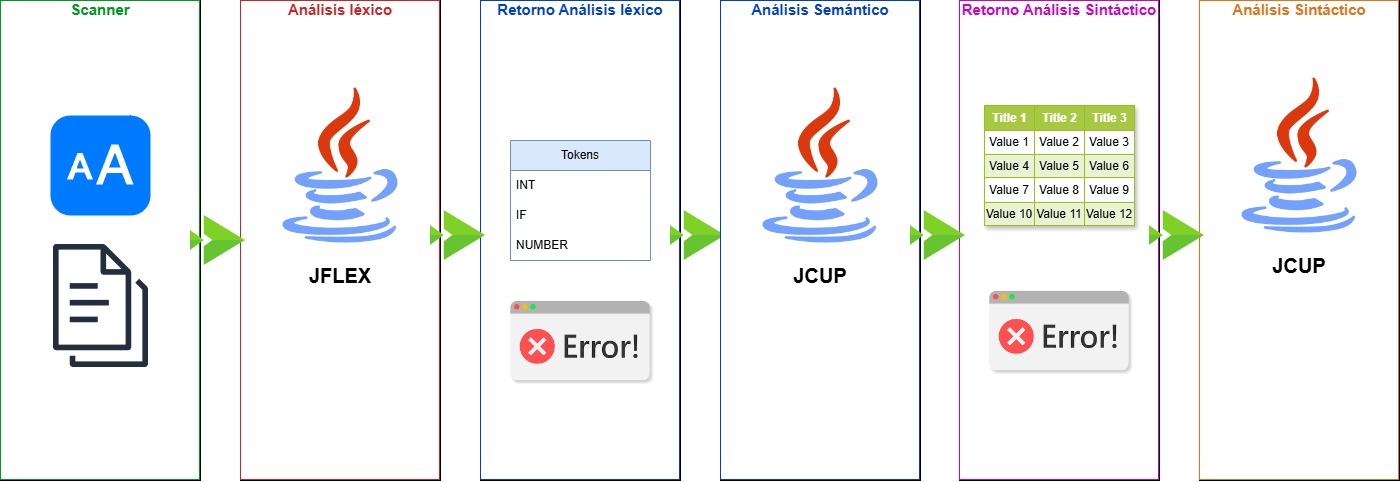
**Este manual está diseñado para proporcionar a los administradores del sistema una comprensión completa y detallada sobre la instalación, configuración y mantenimiento de un compilador o analizador léxico y semántico desarrollado con Java, utilizando las librerías JFlex y JCup. Y tiene como analizar y compilar código fuente escrito en un lenguaje de programación específico, proporcionando información detallada sobre la estructura léxica y semántica de este. A través de este proceso, se garantiza que el código cumple con las reglas sintácticas y semánticas definidas para el lenguaje. Pero mas importante que eso es entender el proceso desde que se inserta una cadena de texto hasta la creación de un programa objeto.**

# **ARQUITECTURA DEL SISTEMA**

## Diagrama de arquitectura

El siguiente diagrama ha sido creado para respaldar la arquitectura de componentes de la solución, diseñada para facilitar el uso del producto por parte de los usuarios.

El Compilador tiene la capacidad de permitirle al usuario la carga de archivos de texto y/o escribir texto en un campo el cual será tomado y analizados léxica y semánticamente y le retornaran al usuario el resultado del proceso de análisis.



## Tecnologías utilizadas

Para el proyecto utilizaremos:

* JDK 22 o superior.
* JAVA Flex (Librería).
* JAVA Cup(Libreria).

Para mayor información de las librerías y dependencias usadas en los proyectos se puede consultar el archivo **Readmi.MD** del proyecto**.**

# REQUISITOS DEL SISTEMA

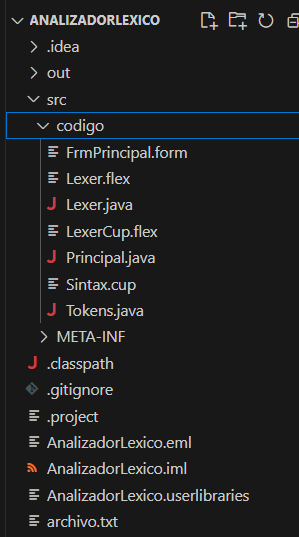
* JDK – JAVA Development Kit (22) o superior.
* JRE - JAVA Runtime Environment (Opcional).
* Windows 10 o superior.

# INSTALACION

Para instalar el compilador primero validar que se cumple con el apartado anterior de requisitos del sistema.

# ESTRUCTURA DEL CODIGO

## Organización del proyecto



* **Src/codigo/FrmPrincipal :** Formulario del proyecto o front end donde el usuario podrá interactuar con el sistema.
* **Src/codigo /Lexer.flex**: Es el archivo que contiene la lista de tokens y los lexemas del analizador léxico.
* **Src/codigo /Lexer.java:** Es el archivo que luego de ser sometido a el proceso de autómata del JFlex contiene todas las reglas para la validación de los tokens del analizador léxico.
* **Src/codigo /Lexer.cup:** al igual que el lexer.flex contiene los tokens del léxico pero con un ajuste necesario para el análisis semántico.
* **Src/codigo Principal.java:** es el archivo principal del proyecto el cual ejecuta las instrucciones que son tomadas por el autómata para construir las reglas del analizador.
* **Src/codigo Sintax.cup:** Es el archivo que contiene la estructura del análisis sintáctico que será evaludado.
* **Src/codigo Tokens.java:** este archivo contiene los tokens en una clase enum.
* **.gitignore:** Directorio en el cual se indica los archivos que se ignorar al subir los cambios al gestor de versiones GitHub.
* **READMI.md :** Archivo donde se almacena información pertinente al proyecto.

### Nombramiento de variables

* Se usa la notación de camelCase para nombrar variables y métodos. Por ejemplo, miVariable o miMetodo().

### Indentación y espacios

* Se aplica una sangría de 2 espacios.
* Se utiliza un espacio después de los dos puntos en las propiedades de los objetos.