**DOCUMENTACIÓN**

Descripción de componentes (analizador léxico, sintáctico, etc.)

* Analizador léxico RuLexer.

El Lexer es generado a partir de la gramática Ru.g4. Esté se encarga de reconocer los tokens de Ru, como operadores (+,-,\*,/, etc.), palabras reservadas (if, while, etc.), Además de incluir reglas para otros caracteres (OTRO).

* Analizador sintáctico RuParser.

El Parser también generado por Ru interpreta la secuencia de tokens y la organiza en una estructura de árbol.

* RubaseVisitor

Es una clase abstracta generada por ANTLR que proporciona los métodos vacíos para cada nodo en el árbol sintáctico.

* MyVisitor

Es una clase Java creada por el programador la cual hereda las funciones de “RuBaseVisistor”. Donde cada nodo tiene su propia implementación.

* Main

Define la **interfaz de usuario (GUI)** usando Java Swing. Divide en 2 secciones la pantalla en la parte izquierda permite escribir código Ru directamente a través de un botón “Play”; un boton “Cargar archivos” .ru desde el sistema; otro botón “guardar resultados” el cual guarda en un archivo nuevo o sobrescribir un archivo .txt

Lista de clases e instancias utilizadas

Main

JFrame

Clase para la interfaz gráfica construida con Java Swing

Carga y ejecuta código .Ru en el panel izquierdo

Muestra los resultados de la Ejecución .txt

MyVisitor

Extiende de RuBaseVisitor

Define el comportamiento semántico de Ru

Maneja las estructuras de control como if y while

Usa HasMap para almacenar y consultar variable

muestra resultados con “imprime”

Ru.g4

Define la sintaxis

Genera el lexer (analizador léxico), parser(analizador sintáctico)

Genera la interfas para recorrer el arbor (BaseVisitor y Visitor)