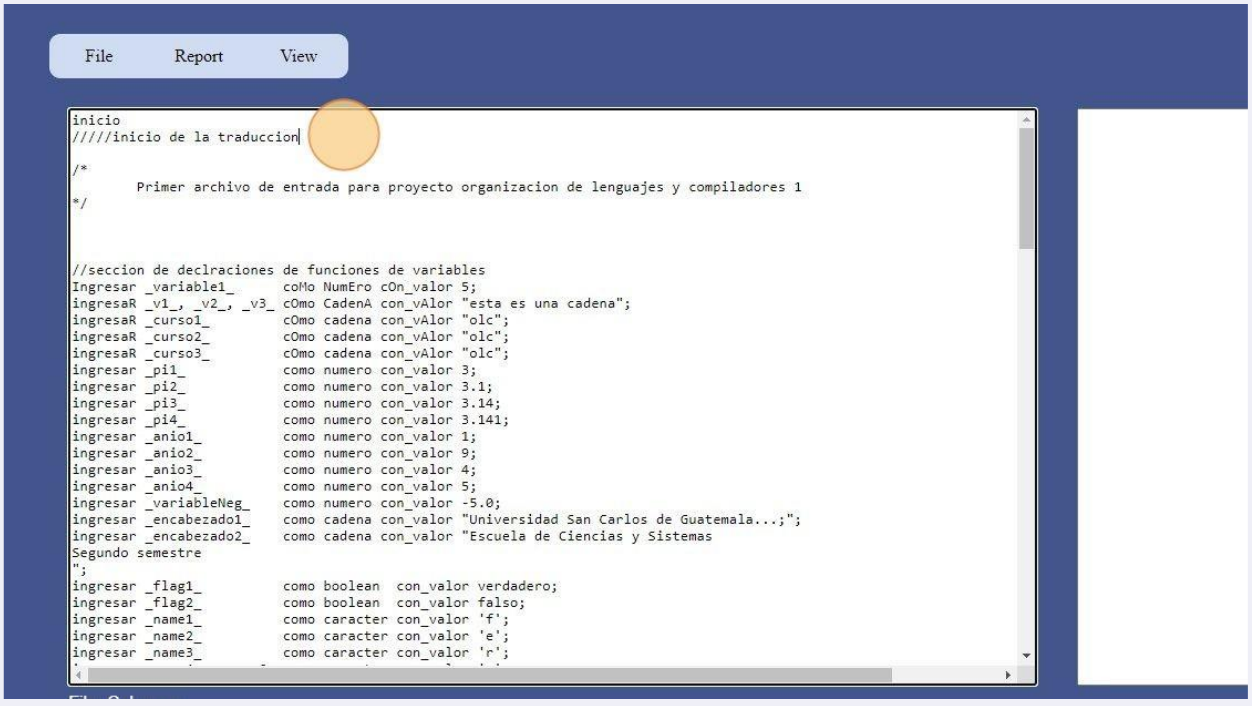
User Manual

1 Open index.html

2 ¿Como abrir archivos?
Click "File

3 Click "Open file"
Le permitirá seleccionar archivos de pseudo código en su PC y verlos en el navegador

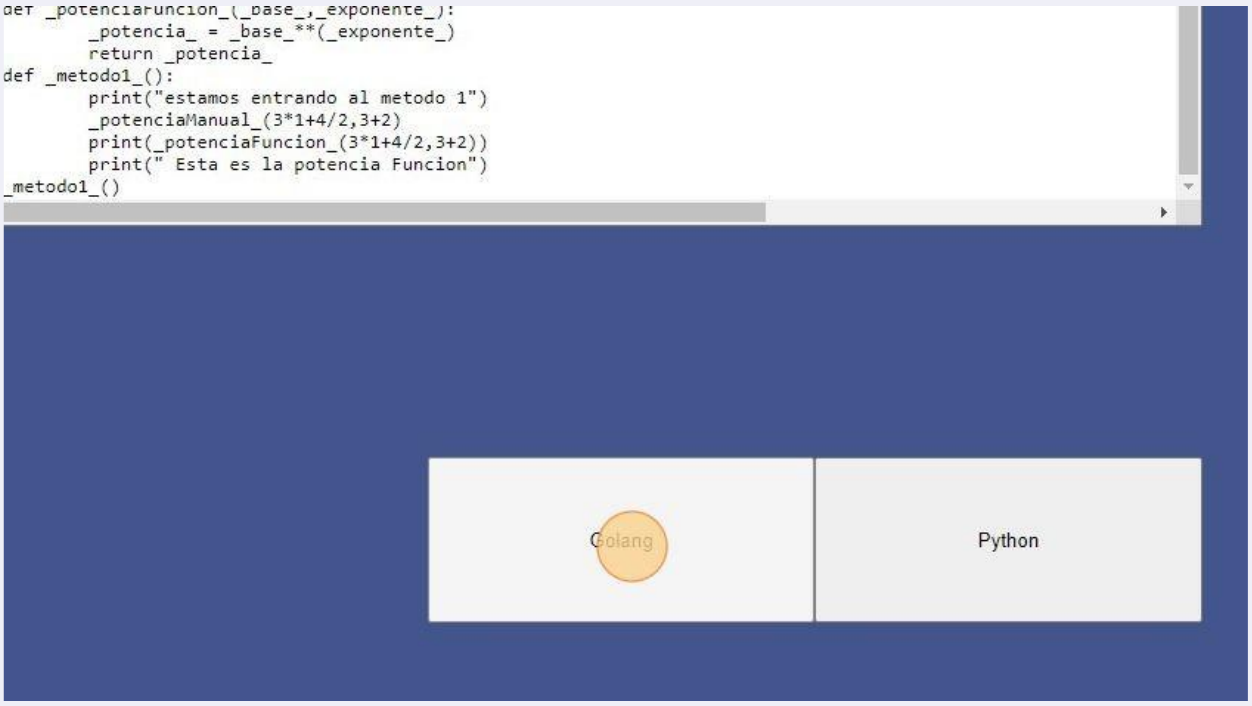
4 Se moverá el texto al cuadro de entrada del lado izquierda, el cual podrá editar el contenido o enviarlo a traducir



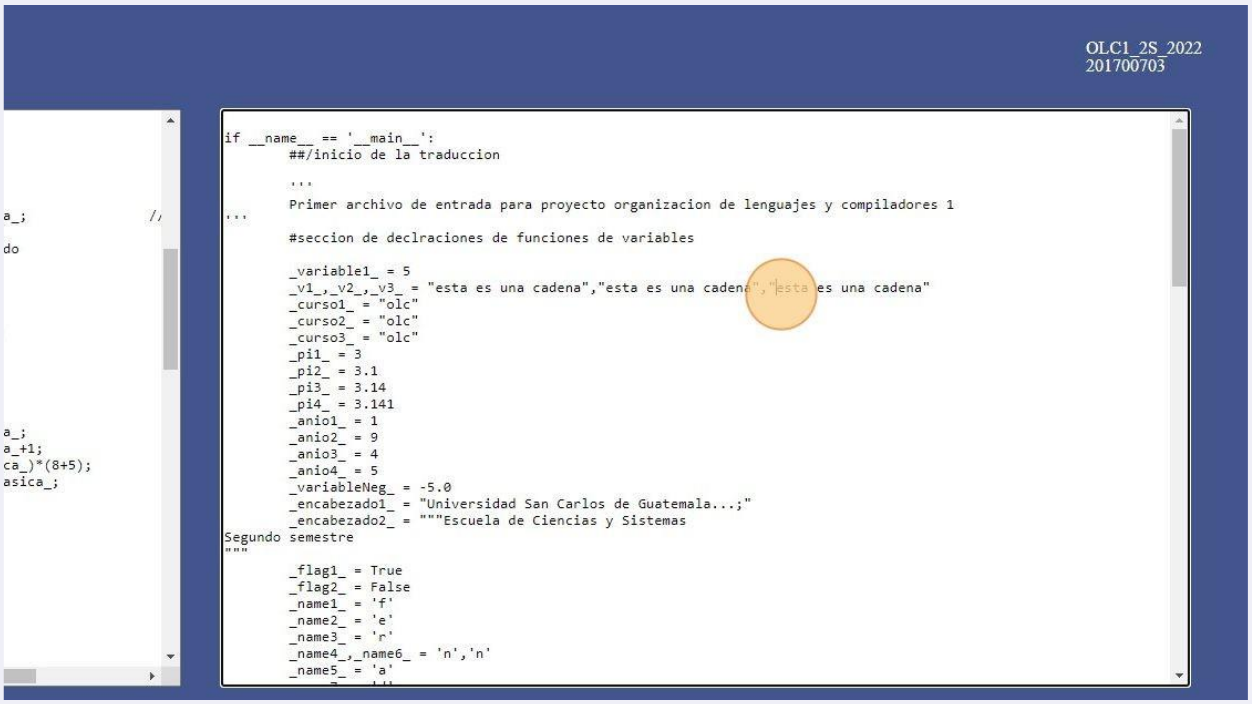
5 ¿Como traducir?
Dependiendo del lenguaje al que desee traducir el pseudocódigo de

Click "Python"

6 o Click "Golang"

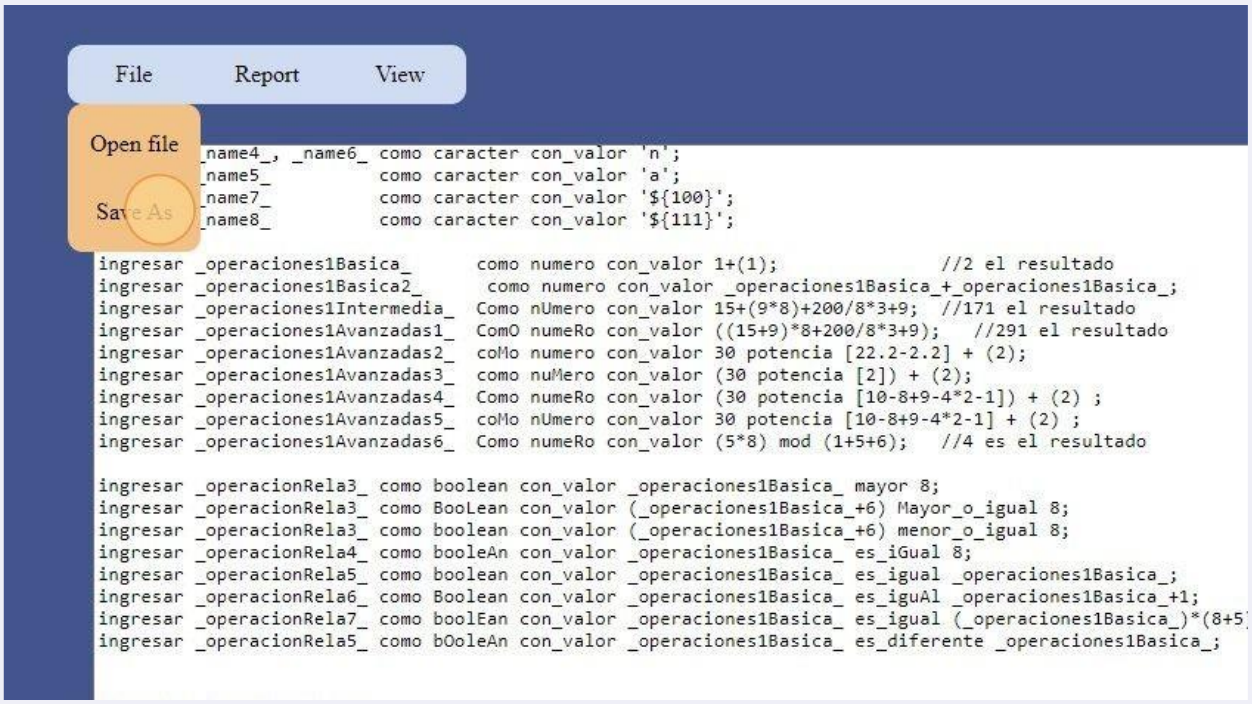


7 Podrá observar la traducción en el lado derecho



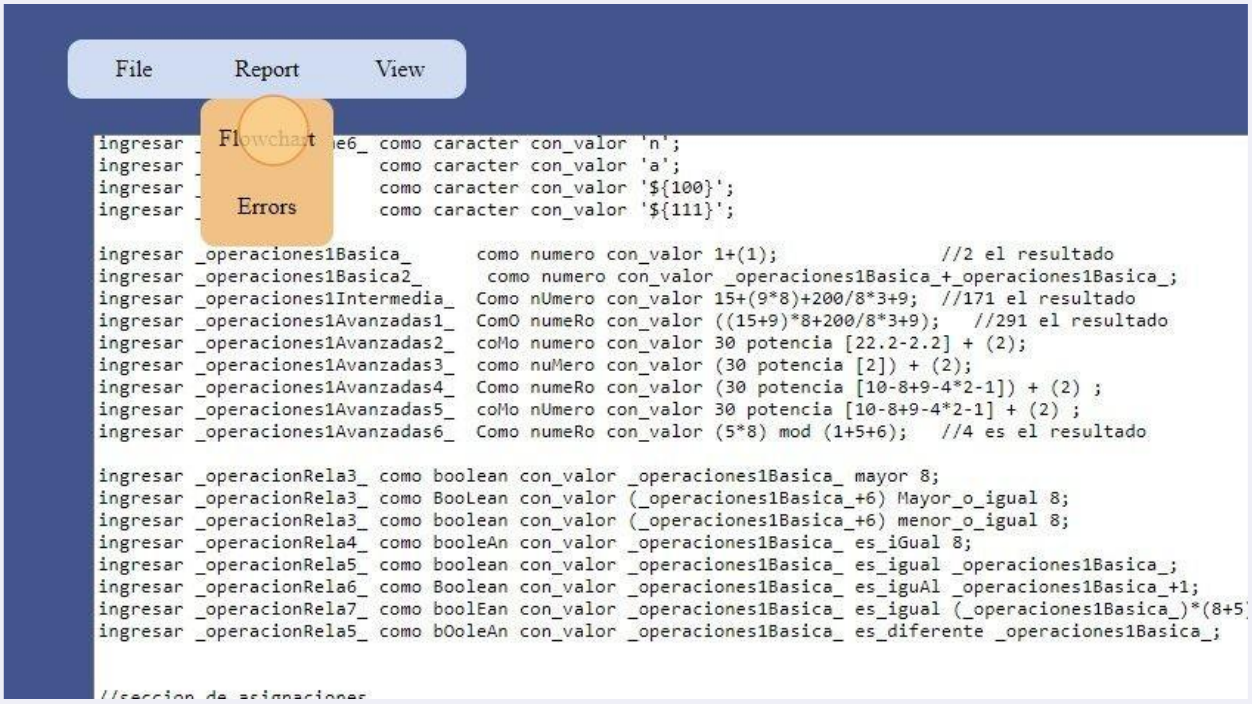
8 En dado caso el texto ingresado tuviera errores, mostrara los errores en el siguiente recuadro indicando que tipo de tokens/simbolos estaba esperando

9 Si desea guardar el pseudocódigo a un archivo, puede dar Clic en el botón Save As, lo que descargara un archivo con el código automáticamente

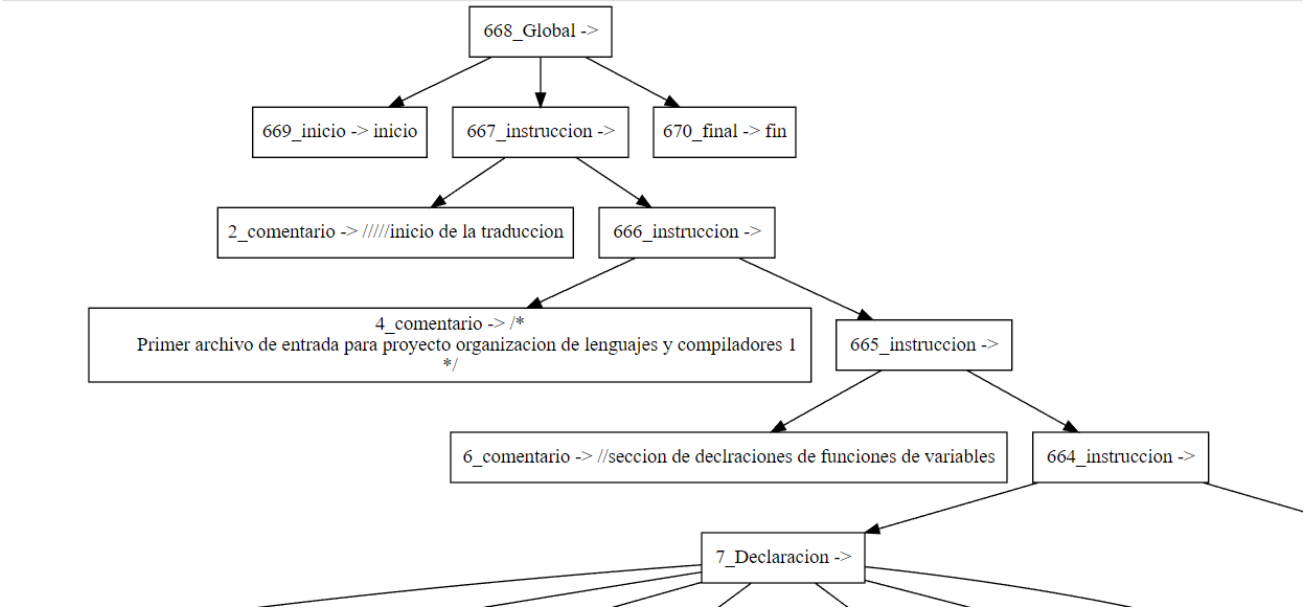


10 Click "Flowchart"

Le llevará a ver el árbol del código ingresado y como este fue estructurado en una nueva ventana

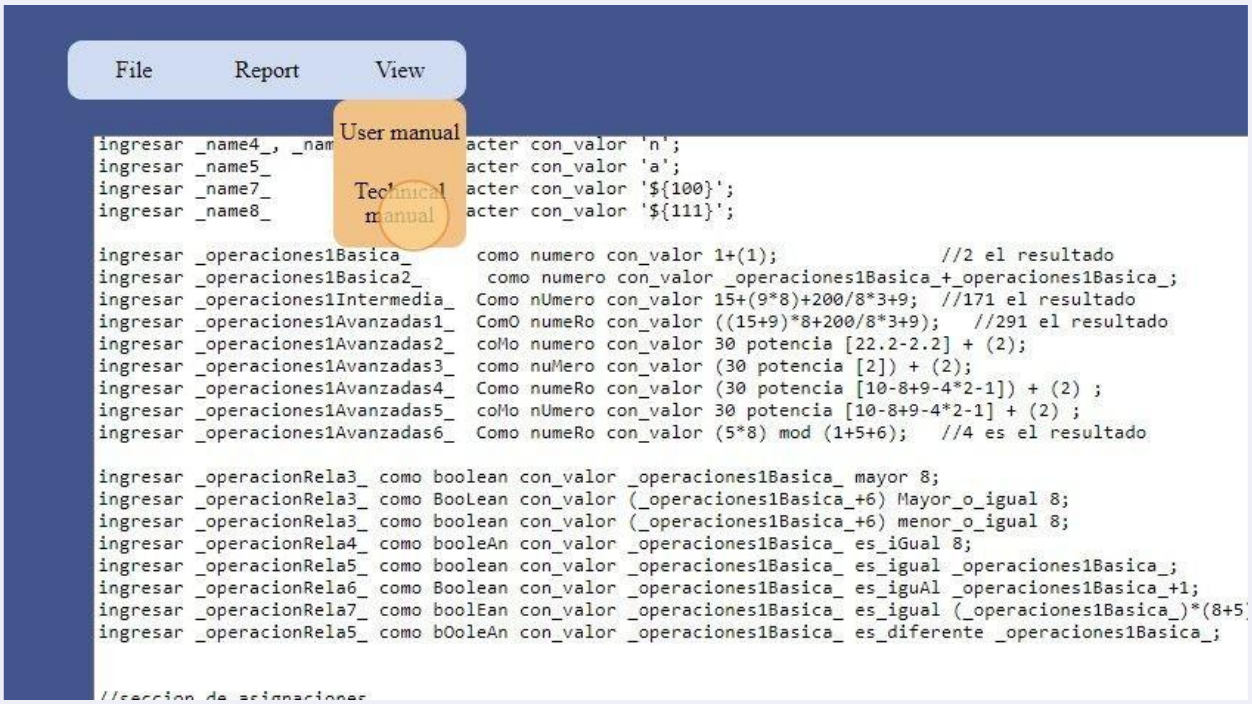


11 Switch to tab

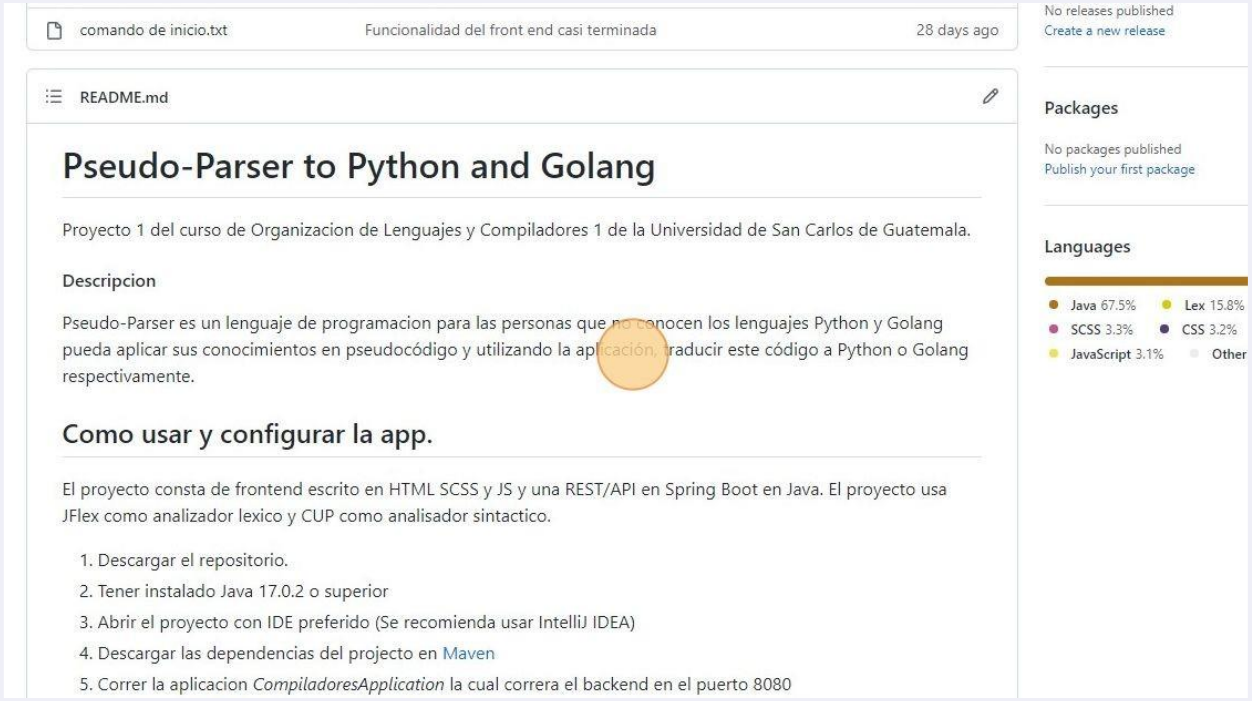


12 Click "Technical manual"

Lo llevará a ver la documentación técnica en GitHub



13 Documentacion tecnica



14

pueda aplicar sus conocimientos en pseudocódigo y utilizando la aplicación, traducir este código a Python o Golang respectivamente.

JavaScript 3.1%

Como usar y configurar la app.

El proyecto consta de frontend escrito en HTML SCSS y JS y una REST/API en Spring Boot en Java. El proyecto usa JFlex como analizador lexico y CUP como analisador sintactico.

1. Descargar el repositorio.
2. Tener instalado Java 17.0.2 o superior
3. Abrir el proyecto con IDE preferido (Se recomienda usar IntelliJ IDEA)
4. Descargar las dependencias del projecto en Maven
5. Correr la aplicacion *CompiladoresApplication* la cual correra el backend en el puerto 8080
6. Abrir el *index.html* para observar la interfaz. Dentro de *script.js* existe una variable que indica la direccion ip del backend, cambiarla a la direccion ip de su PC de lo contrario no funcionaran las peticiones al backend.
7. Listo para usarse.

Esctructura del proyecto

Backend

src/main/

- /cup

15

Esctructura del proyecto

Backend

src/main/

- /cup

  En esta carpeta encontraras dos archivos, *parser.cup* y *golang_helper.cup*

  parser.cup: Archivo encargado de generar el parser para el analisis sintactico del pseudo codigo.

  golang_helper.cup: Archivo encargado de generar un parser llamado *golang_helper* el cual ayuda al traductor de golang a reacomodar la sintaxis al momento de la traduccion. Este nuevo parser es utilizado en la clase *Traductor_Go* mas adelante.

- 🔗 java/com/backend/compiladores

  mainController: Encargado de recibir las peticiones POST del front para iniciar la traduccion a traves del *compilerController*

  CompiladoresApplication: Clase principal para correr Spring boot

- /services

  compilerController: Encargado de crear el lexer, parser, el AST (Abstract Sintax Tree), traducir el codigo y determinar los errores lexicos o sintacticos que puedan surgir, anexando toda la informacion generada al objeto response, el arbol generado se guarda en la carpeta resources.

  - /parserPackage

    Clases para la creacion de arboles y nodos

  - /response

16 Click "User manual"

Lo llevara a ver este manual