UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERIA

ESCUELA DE SISTEMAS

ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1

Ing. Kevin Lajpop

Aux. Moises Gonzales

PROYECTO 1

Pseudocodigo -> Python/Golang

JOSUE ROLANDO

GRAMAJO ROLDAN

202000895

3021021080101

INDICE

Introducción

Un cliente ha solicitado a usted un Pseudo-Parser para que el nuevo personal que no conoce los lenguajes de Python y Golang, se solicita que implemente de las primeras 2 fases de un compilador y ejecute una traducción con la entrada de pseudocódigo a Python y Golang.

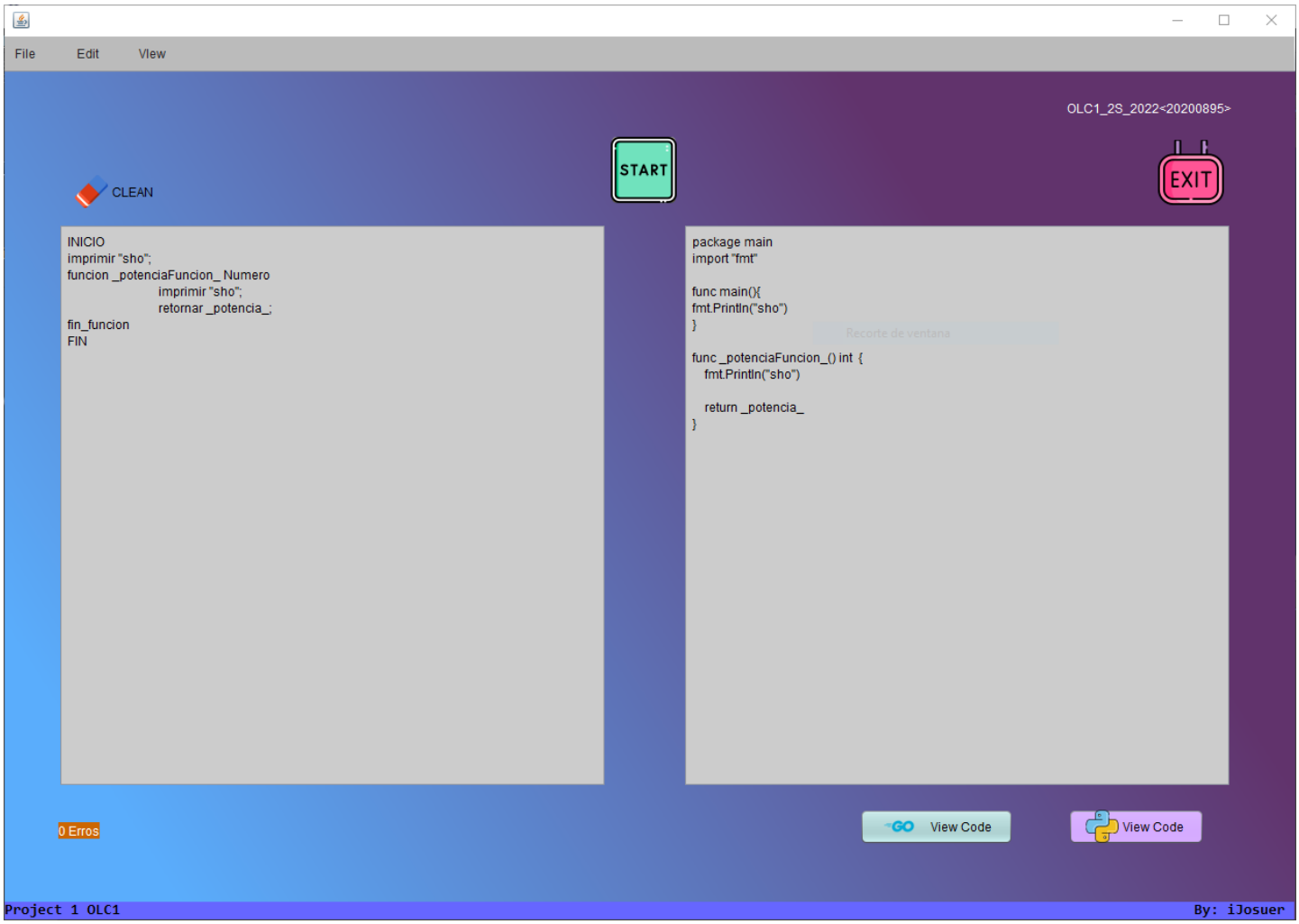
Objetivos

Guiar al lector en la manipulación del programa, para un uso correcto y fluido, de esa manera se evitaran bugs o posibles errores en la aplicación.

Requerimientos mínimos para ejecutar en Neatbens 8.2

- Java 1.8
- Procesador a 1.6 GHz o superior
- 512 MB de RAM
- 750 MB de espacio disponible en el disco duro.
- Tarjeta de vídeo compatible con DirectX 9
- windows 7,8,superior

Interfaz de Usuario



Botón CLEAN: Este botón es el que limpia toda el área de texto para ingresar el Pseudocodigo.

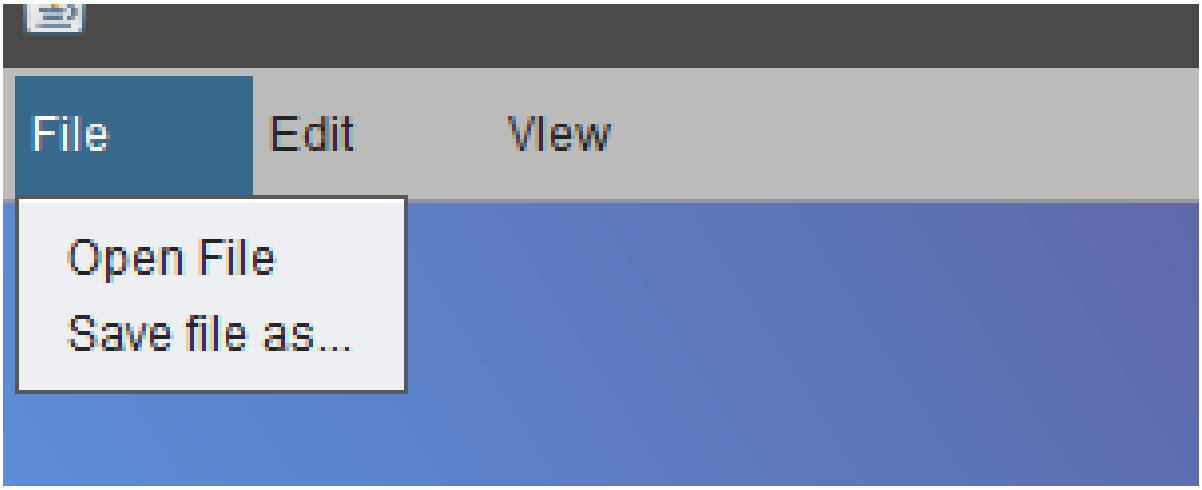
Botón START: Este botón función a de “Backend”, llama a nuestro “compilador” y procede a realizar la traducción.

Botón EXIT: Este botón finaliza la ejecución del programa.

Botón PYTHON: Con este botón podremos visualizar el código Python generado.

Botón GOLANG: Con este botón podremos visualizar el código Golang generado.

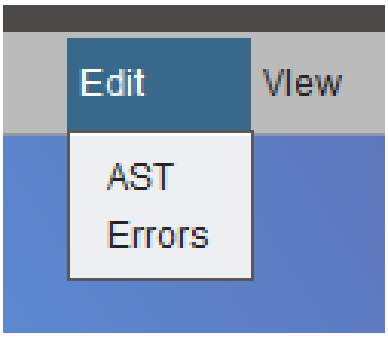
Submenú File



Open File: Este botón nos permite escoger un archivo dentro de nuestra carpeta principal para después ingresarlo en el textArea del seudocódigo.

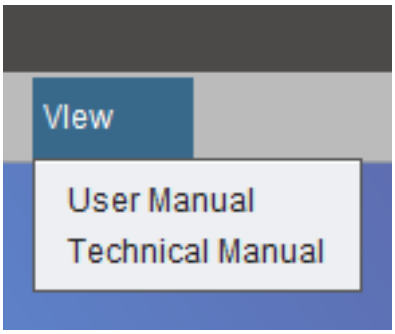
Save File: Este botón nos permite guardar el archivo con código Python/Golang con un nombre específico para después guardarlo en la carpeta principal.

Submenú Edit.



AST: Este botón nos permite visualizar el árbol de análisis sintáctico generado al momento de hacer la traducción.

Errors: Este botón nos abre un archivo HTML para poder visualizar de forma ordenada todos los erros encontrados al momento de hacer la traducción.



User Manual: Este botón nos permite visualizar el manual de usuario que estás viendo en este momento.

Technical Manual: Este botón nos permite visualizar el manual técnico de la aplicación.

Ejemplo archivo de entrada

```
@ /*otro error lexico************/

//seccion de asignaciones
_v1_                        -> "esta es la cadena numero 1";
_v2_, _v3_                  -> "estas cadenas deben ser diferentes";
_curso1_ , _curso2_ ,_curso3_  -> "Organizacion de lenguajes y compiladores 1";
_curso1_ , _curso2_ ,_curso3_  "Organizacion de lenguajes y compiladores 1" ;   /******error sintactico, le falta la flecha*/


$ //otro error lexico

imprimir_nl _encabezado1_;
imprimir_nl _encabezado2_;
imprimir  "...";
imprimir _anio1_ ;
imprimir _anio2_ ;
imprimir _anio3_ ;
imprimir _anio4_ ;
imprimir_nl ".";
imprimir_nl (_v3_);

si _v1_ es_igual _v2_ entonces
        imprimir_nl "Al parecer no funciona la asignacion";
        mientras not (_variable1_ mayor_o_igual 5*5+8/2) hacer
                imprimir _variable1_;
                _variable1_ -> _variable1_ + 1;
        fin_mientras
fin_si

si _v1_ es_igual _v2_ entonces
        imprimir_nl "no tiene que imprimir este mensaje";
        de_lo_contrario
            imprimir "este print es un ejemplo";
fin_si
```

```
/*Ahora empezamos con las funciones y procedimientos*/

metodo _potenciaManual_ con_parametros (_base_ Numero, _exponenete_ Numero)
        ingresar _i_ como Numero con_valor 0;
        ingresar _acumulado_ como Numero con_valor 0;
        para _i_->0 hasta _exponente_-1 hacer
                _acumulado_ -> _acumulado_ + _acumulado_;
        fin_para
        imprimir _acumulado_;

fin_metodo

funcion _potenciaFuncion_ Numero con_parametros (_base_ Numero, _exponente_ Numero)
        ingresar _potencia_ como Numero con_valor _base_ potencia [_exponente_];
        retornar _potencia_;
fin_funcion

metodo _metodo_1_
        imprimir_nl "estamos entrando al metodo 1";
        ejecutar _potenciaManual_(3*1+4/2, 3+2);
        imprimir ejecutar _potenciaFuncion_(3*1+4/2, 3+2);
        imprimir_nl " Esta es la potencia Funcion";
fin_metodo

ejecutar _metodo_1_();
```