

PROYECTO 1 - TRADUCTOR

MANUAL TÉCNICO

FACULTAD DE INGENIERÍA, USAC

SEPTIEMBRE 2022

YENIFER ESTER YOC LARIOS, 201952336

OBJETIVOS

Objetivo General:

Brindar al nuevo personal de la empresa de nuestro estimado cliente un traductor que implemente los lenguajes Python y Go, puesto que a los mismos no les es conocido y para agilizar las labores se nos ha solicitado la creación de este programa traductor.

Objetivos Específicos:

Presentar una interfaz interactiva, que permita una interacción entre el usuario y el programa.

Implementar las herramientas JFlex y Cup para los analizadores de código.

Generar archivos con extensión .py y .go con las traducciones al respectivo lenguaje.

Alcances:

Que pueda implementarse el uso y manipulación de las herramientas JFlex y Cup para la traducción de determinado texto escrito en pseudocódigo.

Especificación Técnica: Lenguaje Java

Requisitos del hardware:

- Un ordenador de escritorio o portátil.

Requisitos del Software:

- Navegador web tales como: Mozilla Firefox, Safari (macOS), Microsoft Edge, Avast Secure Browser, Opera.
- Se solicita al usuario que, para poder ejecutar este programa, tenga instalado en su ordenador, el IDE Eclipse, y el JDK respectivo.
- Se necesita que la resolución de la pantalla sea a colores para poder apreciar la interfaz como es debido.

LÓGICA DEL PROGRAMA

El programa se realizó por medio del lenguaje JavaScript para implementar estructuras de datos lineales y visualizarlas en la página web.

Las clases creadas fueron las siguientes:

La clase principal Main()

```
public class Main {
    public static LinkedList<Mi_error> errores;
    public static LinkedList<Mi_error> errores_Lex;

    public static void main(String[] args) {

        // TODO Auto-generated method stub
        Analizador_int Principal_v = new Analizador_int();
        Principal_v.setVisible(true);
        Principal_v.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

La interfaz:

```
public Analizador_int() {
    //Frame mi_fr = new Frame();

    setTitle("Proyecto 1 OLC1");
    setSize(950,550);
    setLocationRelativeTo(null);
    prin_aut = new JPanel();
    prin_aut.setBackground(new Color(0, 0, 128));
    prin_aut.setBorder(new EmptyBorder(5, 5, 5, 5));
    prin_aut.setLayout(null);
    setContentPane(prin_aut);

    JMenuBar mi_menu=new JMenuBar();
    mi_menu.setFont(new Font("Broadway", Font.PLAIN, 14));
    JMenu m_file = new JMenu("File");
    JMenu m_report=new JMenu("Report");
    JMenu m_view = new JMenu("View");
    JMenuItem open_file=new JMenuItem("Open File");
    JMenuItem save_as=new JMenuItem("Save_as");
    JMenuItem flow=new JMenuItem("Flowchart");
```

Clase Instrucción

```

1 package proyecto_1;
2
3 public interface Instruccion {
4
5     public int contador=0;
6
7     public Object Codigo_python();
8     public Object Codigo_golang();
9     public String CodigoDot();
10    public String dot_flu();
11 }
12

```

Demás clases que heredan de ella:

```

1 package proyecto_1;
2
3 import java.util.LinkedList;
4
5 public class Declaracion_var implements Instruccion{
6     private String tipo_dato;
7     private LinkedList<String> identificadores;
8     private Expresion valor;
9     public static int contador =0;
10    public static int contador2 =0;
11    public Declaracion_var(LinkedList<String> identificadores, String
12        super();
13        this.tipo_dato = tipo_dato;
14        this.identificadores = identificadores;
15        this.valor = valor;
16    }
17

```

Asignación, Declaracion_var, Caso, Si, Según, Mientras, Imprimir, Ejecutar, etc.

Los analizadores:

```
package proyecto_1;
import java_cup.runtime.*;

import java.util.LinkedList;

%%

//directrices
//java -jar jflex-full-1.7.0.jar A_flex.jflex

%public
%class Analizador_Lexico
%cupsym Simbolos
%cup
%char
%column
%full
%ignorecase
%line
%unicode

L=[a-zA-Z]
D=[0-9]
COMENTUNILINEA      =  ("//".*\r\n)|("//".*\n)|("//".*\r)
COMENTMULTILINEA    =  "/*"/*"([^\r\n]|"/*"/*"[^\r\n])"/*"/*"/"
espacio=[ \r\t\n]+
comentario_linea = ("//".*)
cad = ("\"\"\"[^\r\n\"]+\"\"")
asc_ = "\\'$\"\"\\{ "[0-9]+"\\}\\' | '{L}'

%{
    public LinkedList<Mi_erroror> errores = new LinkedList<>();
}%

%%

//EXPRESIONES REGULARES
```

```

1 package proyecto_1;
2 import java_cup.runtime.Symbol;
3
4 import java.util.LinkedList;
5
6
7 parser code
8 {
9     //java -jar java-cup-11b.jar -parser Analizador_sintactico -symbols Simbolos A_cup.cup
10     public LinkedList<Instruccion> AST;
11     public LinkedList<Mi_error> errores_sintac = new LinkedList<>();
12     public void syntax_error(Symbol s){
13         System.out.println("ERROR RECUPERABLE ");
14         System.err.println("Error De Sintaxis en la línea " + (s.right) + " Columna "+s.left+ ". El error es: "+s.value);
15         errores_sintac.add(new Mi_error((String)s.value, s.left, s.right));
16         for(int i = 0; i<errores_sintac.size(); i++){
17             System.out.println(i+"ERROR SINTACTICO: "+errores_sintac.get(i).lexema+" fila "+errores_sintac.get(i).linea);
18             System.out.println("\n");
19         }
20
21     public void unrecovered_syntax_error(Symbol s) throws java.lang.Exception{
22         System.out.println("ERROR NO RECUPERABLE ");
23
24         System.err.println("Error sintactico en la línea " + (s.right)+ " Columna "+s.left+ ". Componente no reconocido: "+s.value);
25         errores_sintac.add(new Mi_error((String)s.value, s.left, s.right));
26         for(int i = 0; i<errores_sintac.size(); i++){
27             System.out.println(i+"ERROR SINTACTICO: "+errores_sintac.get(i).lexema+" fila "+errores_sintac.get(i).linea);
28             System.out.println("\n");
29         }
30
31     public LinkedList<Instruccion> getAST() {
32         return AST;
33     }
34 }
35
36
37 terminal String tipo dato, identificado, cadena, caracter, numero, es inicio, es fin, es set, es mod, es

```

Anexos

<https://github.com/YeniferYoc/OLC1-201952336.git>