

Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Ciencias y Sistemas  
Organización de Lenguajes y Compiladores 1

## **Manual Técnico**

Bryan Gerardo Páez Morales

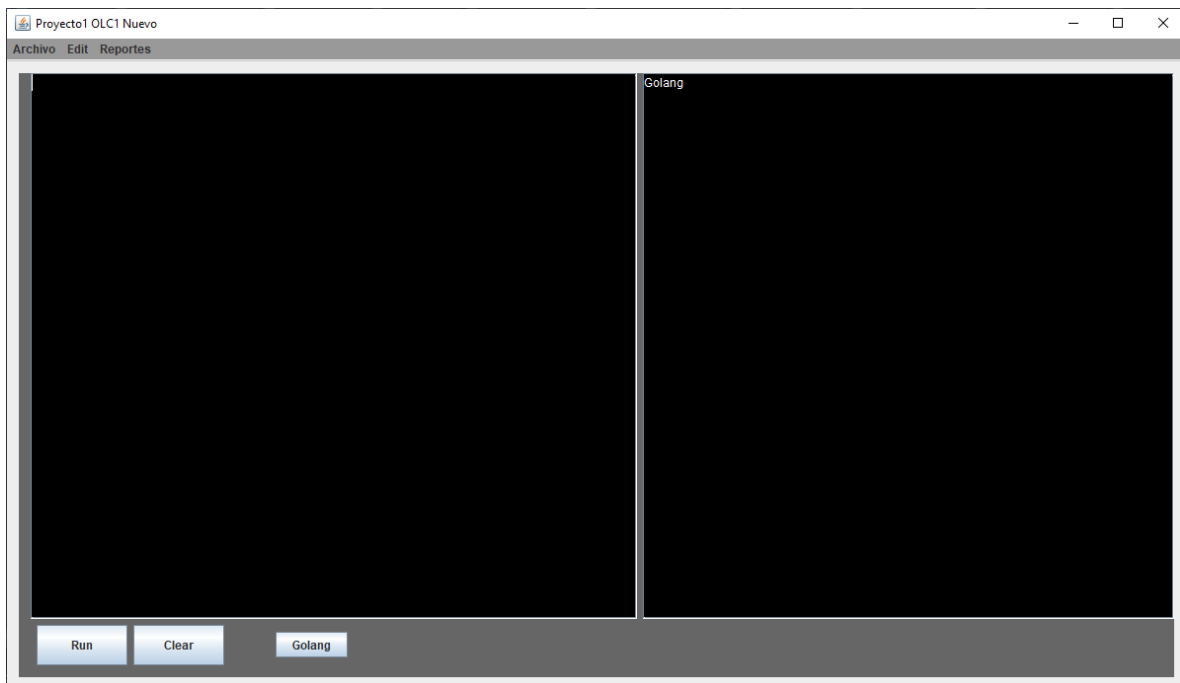
Carné: 201700945

# Manual Técnico

## Clases Utilizadas

- ArbolAST.java
- NodoAST.java
- ArchivoError.Java
- TraduccionGo.java
- TraduccionPy,java
- Datos.java
- Parser.java
- Generar.java

## Interface Grafica



## Código Cup

Código de cup utilizado para el análisis sintactico y el almacenamiento de nodos en un arbol AST

```
start with INICIO;

> INICIO ::= inicio:varin CODIGO:lista_codigo fin:varfin { : ...
> COMIENZO ::= CONTENIDO:lista_contenido{ : ...
> CONTENIDO ::= METODO:lista_metodo{ : ...
> CODIGO ::= INSTRUCCIONES:lista_instrucciones { : ...
> INSTRUCCIONES ::= DECLARACION:lista_dec puntocomas:var_pyc{ : ...
> PRINT ::= imprimir:vari1 EXPRESION:lista_expresiones puntocomas:vari2{ : ...
> PRINTLN ::= imprimir_nl:vari1 EXPRESION:lista_expresiones puntocomas:vari2{ : ...
> DECLARACION ::= ingresar:var_ing NOMBRES como:var_como TIPODATO con_valor:var_con_valor EXPRESION:lista_expresiones{ : //corregir lista de nombres...
> NOMBRES ::= NOMBRE(:lista_auxNodo3.add(nodo_aux_td);) : ...
> NOMBRE ::= var_identificador:var_id coma:var_coma{ : ...
> TIPODATO ::= tcadena:var_tipod{ : ...
> EXPRESIONVAL ::= digito:var_val{ : ...
> ASIGNACION ::= NOMBRES flecha:var_fecha EXPRESION:lista_expresiones { : ...
> EXPRESION ::= EXPRESION:lista_expresionesI suma:vari EXPRESION:lista_expresionesD{ : ...
> INST_SI ::= si:vari1 EXPRESION:lista_expresiones enton:vari2 CODIGO:lista_codigo { : ...
> OSI ::= OSI:lista_osi CONDELOSI:lista_conosi{ : ...
> CONDELOSI ::= o_si:vari EXPRESION:lista_expresionesI enton:vari2 CODIGO:lista_codigo{ : ...
> INST_SEGUN ::= segun:vari1 var_identificador:vari2 hacer:vari3 CASOS:lista_casos de_lo_contrario:vari4 enton:vari5 CODIGO:lista_codigo{ : ...
> CASOS ::= CASOS:lista_casos1 CASO:lista_casos2{ : ...
> CASO ::= interrogacionA:vari1 EXPRESIONVAL:vari2 interrogacionC:vari3 enton:vari4 CODIGO:lista_codigo{ : ...
> CICLOPARA ::= para:vari1 var_identificador:vari2 flecha:vari3 digito:vari4 hasta:vari5 EXPRESION:vari6 hacer:vari7 CODIGO:lista_codigo fin_para:vari8{ : ...
> CICLOMIENTRAS ::= mientras:vari1 EXPRESION:lista_expresiones hacer:vari2 CODIGO:lista_codigo fin_mientras:vari3{ : ...
> CICLOREPETIR ::= repetir:vari1 CODIGO:lista_codigo hasta_que:vari3 EXPRESION:lista_expresiones{ : ...
> METODO ::= metodo:vari1 var_identificador:vari2 CODIGO:lista_codigo fin_metodo:vari3{ : ...
> FUNCION ::= funcion:vari1 var_identificador:vari2 TIPODATO:lista_para CODIGO:lista_codigo fin_funcion:vari3{ : ...
> LLAMADA ::= ejecutar:vari1 var_identificador:vari2 aparentecis:vari3 cparentecis:vari4 { : ...
> LPARAMETROS ::= LPARAMETROS:lista_lpar coma:vari1 PARAMETROS:lista_parametros{ : ...
> PARAMETROS ::= var_identificador:vari1 TIPODATO:lista_parametros{ : ...
> LVALORES ::= LVALORES:lista_valores coma:vari1 EXPRESION:lista_expresiones{ : ...
/* para los puntos colocar en parser
```

## Metodos y Funciones Principales

Código utilizado para traducir a Python

```
public String Python(ArrayList<NodoAST> nodo) {
    String cadena="";
    nombres=0;
    boolean metodo = false;
    int indice=0;
    String tabulacion="";
    for (int i = 0; i < nodo.size(); i++) {
        if (ntabulaciones!=0) {

        }
        cadena+=enumPython(nodo.get(i), nodo, i, metodo);
    }
    String encabezado=agregarImportaciones();

    return cadena;
}
```

Código utilizado para traducir a Golang

```

public String Golang(ArrayList<NodoAST> nodo) {
    for (int i = 0; i < nodo.size(); i++) {
        nodo.get(i).setToken(nodo.get(i).getToken().toLowerCase());
    }
    String cadena="";
    nombres=0;
    boolean metodo = false;
    int indice=0;
    String tabulacion="";
    for (int i = 0; i < nodo.size(); i++) {
        if (ntabulaciones!=0) {

        }
        cadena+=enumGolang(nodo.get(i), nodo, i, metodo);
    }
    String encabezado=agregarImportaciones();

    return encabezado+cadena;
}

```

Código Para Generar Archivo de Errores

```

public void generarArchivoErrores(String nombre, String contenido){
    FileWriter archivo = null;
    PrintWriter pw = null;
    String dir = "./src/Files/" + nombre + ".txt";
    try {
        archivo = new FileWriter(dir);
        pw = new PrintWriter(archivo);
        pw.println(contenido);

    } catch (Exception e) {
        System.out.println("error, no se realizo el archivo");
    } finally {
        try {
            if (null != archivo) {
                archivo.close();
            }
        } catch (Exception e2) {
            e2.printStackTrace();
        }
    }

    try {
        File path = new File (dir);
        Desktop.getDesktop().open(path);
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}

```

Código Para Generar Grafo de Árbol AST

```
public String recorrido(NodoAST raiz){
    String contenido="";
    for (NodoAST hijo : raiz.Nodos) {
        String cadena = hijo.getToken().replace("<", "").replace("\\", "").replace(">", "").replace("'", "");

        if (hijo.Nodos!=null) {
            contenido += "nodo"+hijo.getId()+"[label =\""+cadena+"\"]"; nodo"+raiz.getId()+"->nodo"+hijo.getId()+";\n";
            contenido += recorrido(hijo);
        }else{
            contenido += "nodo"+hijo.getId()+"[label =\""+cadena+"\"]"; nodo"+raiz.getId()+"->nodo"+hijo.getId()+";\n";
        }
    }
    return contenido;
}
```

## Repositorio GitHub

<https://github.com/Pdante1897/OLC1-201700945.git>