

# Manual De Tecnico

## Proyecto 1 Organizacion de Lenguajes i Compiladores 1

### Objetivos e Información del Sistema

Este Programa con su propia sintaxis, es capaz de interpretar el lenguaje OLCScript que está basado en el lenguaje de alto nivel Typescript. Este lenguaje incorpora varias funciones del lenguaje de alto nivel Typescript , además incorpora otras funcionalidades de alto nivel para hacerlo un lenguaje versátil, moderno y eficiente.

### Requisitos del sistema

---

- Procesador AMD Ryzen 5 3400G o superior
- 16 Gb Ram
- Gráficos Radeon RX 6600 o superior
- Conexión a internet
- Navegador Web
- Java

### Lenguaje y herramientas utilizadas

#### Python

Se utilizó python como lenguaje de programación.

#### Paradigma de programación

El paradigma usado fue programación orientada a objetos.

#### PLY

PLY fue la herramienta para el análisis léxico del programa.

#### PLY

PLY fue la herramienta para el análisis sintáctico del programa.

# Gramatica utilizadas

init : instrucciones

instrucciones : instrucciones instruccion

instrucciones : instruccion

instruccion : imprimir\_instr  
| declaracion  
| asignacion PUNTOCOMA  
| slf  
| sWhile  
| sFor  
| sContinue  
| sBreak  
| sReturn  
| sSwitch  
| declaracion\_Funcion  
| callFuncion PUNTOCOMA

callFuncion : ID PARIZQ venir\_Expresiones

venir\_Expresiones : lista\_Expresiones PARDER

venir\_Expresiones : PARDER

lista\_Expresiones : lista\_Expresiones COMA op\_Ternario

lista\_Expresiones : op\_Ternario

declaracion\_Funcion : FUNCTION ID PARIZQ parametros tipoFuncion instrucciones  
LLAVDER

parametros : PARDER

parametros : lista\_Parametros PARDER

lista\_Parametros : lista\_Parametros COMA ID DOSPUNTOS tipo  
| ID DOSPUNTOS tipo

tipoFuncion : DOSPUNTOS tipo LLAVIZQ  
| LLAVIZQ

sSwitch : SWITCH PARIZQ op\_Ternario PARDER LLAVIZQ listCases LLAVDER

listCases : listCases CASE op\_Ternario DOSPUNTOS instrucciones  
| listCases DEFAULT DOSPUNTOS instrucciones  
| CASE op\_Ternario DOSPUNTOS instrucciones  
| DEFAULT DOSPUNTOS instrucciones

sContinue : CONTINUE PUNTOCOMA

sBreak : BREAK PUNTOCOMA

sReturn : RETURN PUNTOCOMA

sReturn : RETURN op\_Ternario PUNTOCOMA

sFor : FOR PARIZQ declaracion op\_Ternario PUNTOCOMA asignacion PARDER LLAVIZQ  
instrucciones LLAVDER

sWhile : WHILE PARIZQ op\_Ternario PARDER LLAVIZQ instrucciones LLAVDER

slf : IF PARIZQ op\_Ternario PARDER LLAVIZQ instrucciones slf2

slf2 : LLAVDER ELSE sElse

slf2 : LLAVDER

sElse : LLAVIZQ instrucciones LLAVDER

sElse : slf

asignacion : ID IGUAL op\_Ternario

| ID MAS IGUAL op\_Ternario

| ID MENOS IGUAL op\_Ternario

asignacion : ID MAS MAS

| ID MENOS MENOS

declaracion : tipoVar ID DOSPUNTOS tipo declaracion\_explicita

declaracion : tipoVar ID declaracion\_implicita

declaracion\_explicita : PUNTOCOMA

declaracion\_explicita : IGUAL op\_Ternario PUNTOCOMA

declaracion\_implicita : IGUAL op\_Ternario PUNTOCOMA

tipoVar : VAR

| CONST

tipo : NUMBER

tipo : FLOAT

tipo : STRING

tipo : BOOLEAN

tipo : CHAR

imprimir\_instr : CONSOLE PUNTO LOG PARIZQ lista\_exp PARDER PUNTOCOMA

lista\_exp : lista\_exp COMA op\_Ternario

lista\_exp : op\_Ternario

op\_Ternario : exp\_Or INTERROGACION exp\_Or DOSPUNTOS exp\_Or  
op\_Ternario : exp\_Or

exp\_Or : exp\_Or OR exp\_And  
exp\_Or : exp\_And

exp\_And : exp\_And AND exp\_Not  
exp\_And : exp\_Not

exp\_Not : INot exp\_Comp  
exp\_Not : exp\_Comp

INot : INot NOT  
INot : NOT

exp\_Comp : exp\_Sum\_Menos EQUIVALENTE exp\_Sum\_Menos  
| exp\_Sum\_Menos DISTINTO exp\_Sum\_Menos  
exp\_Comp : exp\_rel

exp\_rel : exp\_Sum\_Menos MAYOR exp\_Sum\_Menos  
| exp\_Sum\_Menos MENOR exp\_Sum\_Menos  
| exp\_Sum\_Menos MAYORIGUAL exp\_Sum\_Menos  
| exp\_Sum\_Menos MENORIGUAL exp\_Sum\_Menos  
exp\_rel : exp\_Sum\_Menos

exp\_Sum\_Menos : exp\_Sum\_Menos MAS exp\_Mul\_Div  
| exp\_Sum\_Menos MENOS exp\_Mul\_Div  
exp\_Sum\_Menos : exp\_Mul\_Div

exp\_Mul\_Div : exp\_Mul\_Div POR exp\_Parenthesis  
| exp\_Mul\_Div DIVIDIDO exp\_Parenthesis  
| exp\_Mul\_Div MODULO exp\_Parenthesis  
exp\_Mul\_Div : exp\_Parenthesis

exp\_Parenthesis : PARIZQ exp\_Or PARDER  
exp\_Parenthesis : exp\_Negativo

exp\_Negativo : MENOS valor

exp\_Negativo : valor

valor : ENTERO  
valor : CADENA  
valor : DECIMAL  
valor : CARACTER  
valor : TRUE  
| FALSE  
valor : NULL

valor : ID  
valor : callFuncion