

#### Rodrigo Merino de la Parra A00836396

# **Expresiones Regulares**

```
ID = ^[A-Za-z][A-Za-z0-9_]*$

CTE_INT = ^[+\-]?\d+$

CTE_FLOAT = ^[+\-]?\d+\.\d+$

CTE_STRING = ^"(?:\\.|[^"\\])*"$
```

## **Patito**

```
<Programa>
<Body>
<ASSIGN>
<CTE>
<FUNCS>
<EXPRESSION>
<STATEMENT>
<EXP>
<TERMINO>
<VARS>
<PRINT>
<CYCLE>
<CONDITION>
```

```
<FACTOR>
<F_CALL>
<TYPE>
```

## Lista de tokens

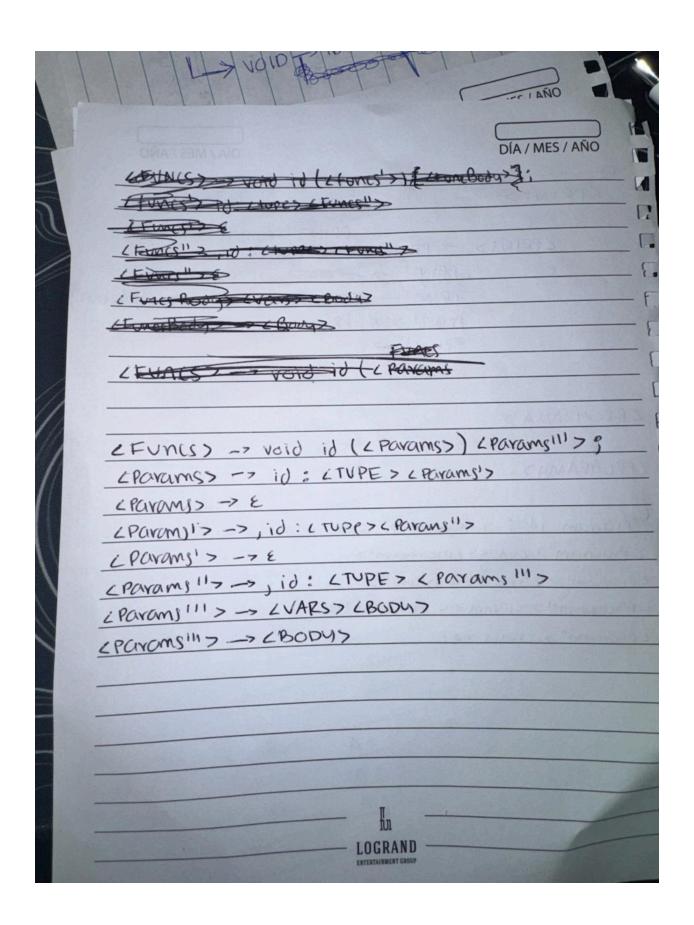
```
PROGRAM → "program"
VAR → "var"
INT → "int"
FLOAT → "float"
VOID → "void"
MAIN → "main"
IF → "if"
ELSE → "else"
WHILE → "while"
DO \rightarrow "do"
PRINT → "print"
END → "end"
PLUS → "+"
MINUS → "-"
MULT → "*"
DIV → "/"
GT → ">"
LT \rightarrow "<"
NEQ \rightarrow "!="
EQ → "="
LPAREN → "("
RPAREN → ")"
LBRACE → "{"
RBRACE → "}"
LBRACKET → "["
RBRACKET → "]"
```

```
COMMA → ","
SEMICOLON → ";"
COLON → ":"
COMMENT_LINE \rightarrow //[^{n}]*
COMMENT_BLOCK \rightarrow / ([^*] | + [^*/]) * + /
WHITESPACE → [\t\r\n]+
A \rightarrow "A" a \rightarrow "a"
B \rightarrow "B" b \rightarrow "b"
C \rightarrow "C" c \rightarrow "c"
D \rightarrow "D" \quad d \rightarrow "d"
E \rightarrow "E" \quad e \rightarrow "e"
F \rightarrow "F" \quad f \rightarrow "f"
G \rightarrow "G" \quad g \rightarrow "g"
H \rightarrow "H" \quad h \rightarrow "h"
| \rightarrow "|" \quad | \rightarrow "|"
\mathsf{J} \to \mathsf{"J"} \quad \mathsf{j} \to \mathsf{"j"}
K \rightarrow "K" \quad k \rightarrow "k"
L \rightarrow "L" \quad | \rightarrow "|"
M \rightarrow "M" \quad m \rightarrow "m"
N \rightarrow "N" \quad n \rightarrow "n"
O \rightarrow "O" o \rightarrow "o"
P → "P"
                    p \rightarrow p"
Q \rightarrow "Q" \quad q \rightarrow "q"
R → "R"
                    r \rightarrow "r"
S \rightarrow "S" \quad s \rightarrow "s"
T \rightarrow T'' \quad t \rightarrow t''
U \rightarrow "U" \quad u \rightarrow "u"
V \rightarrow V'' \quad V \rightarrow V''
W \rightarrow W'' \quad w \rightarrow W''
                    x \rightarrow "x"
X → "X"
Y \rightarrow "Y" \quad y \rightarrow "y"
Z \rightarrow "Z" \quad z \rightarrow "z"
```

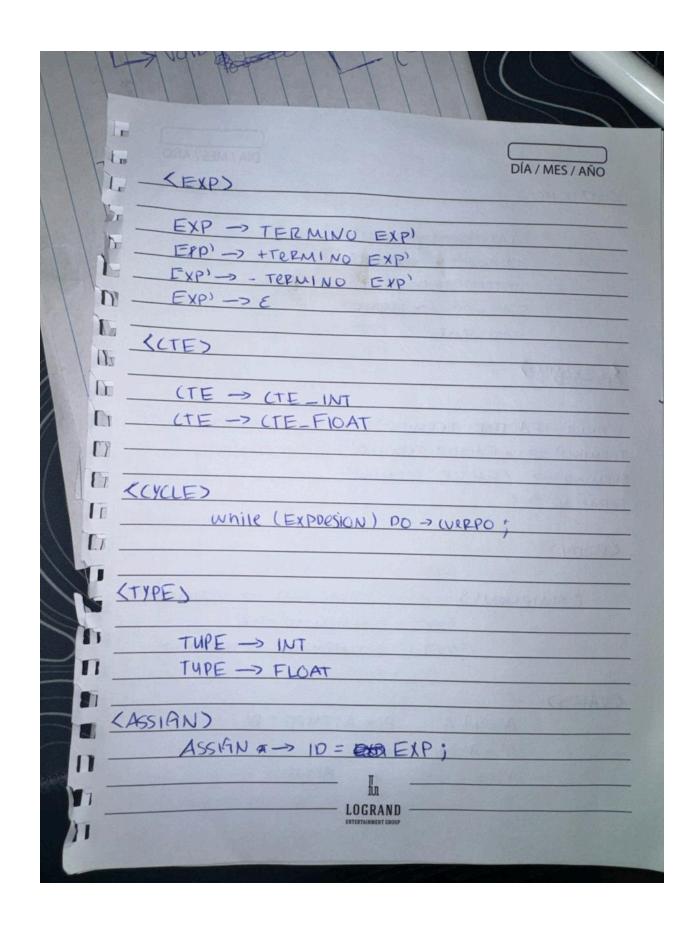
```
DIGIT_0 → "0"
DIGIT_1 → "1"
DIGIT_2 → "2"
DIGIT_3 → "3"
DIGIT_4 → "4"
DIGIT_5 → "5"
DIGIT_6 → "6"
DIGIT_7 → "7"
DIGIT_8 → "8"
DIGIT_9 → "9"
UNDERSCORE → "_"
QUOTE → "'"
DQUOTE → "\""
BACKSLASH → "\\"
AMPERSAND → "&"
PIPE → "|"
EXCLAM → "!"
QUESTION → "?"
PERCENT → "%"
CARET → "^"
TILDE → "~"
AT → "@"
HASHTAG → "#"
DOLLAR → "$"
SPACE → " "
TAB \rightarrow "\t"
NEWLINE \rightarrow "\n"
RETURN \rightarrow "\r"
EOF \rightarrow fin de archivo
ERROR → cualquier carácter no reconocido
```

#### Gramatica libre de contexto

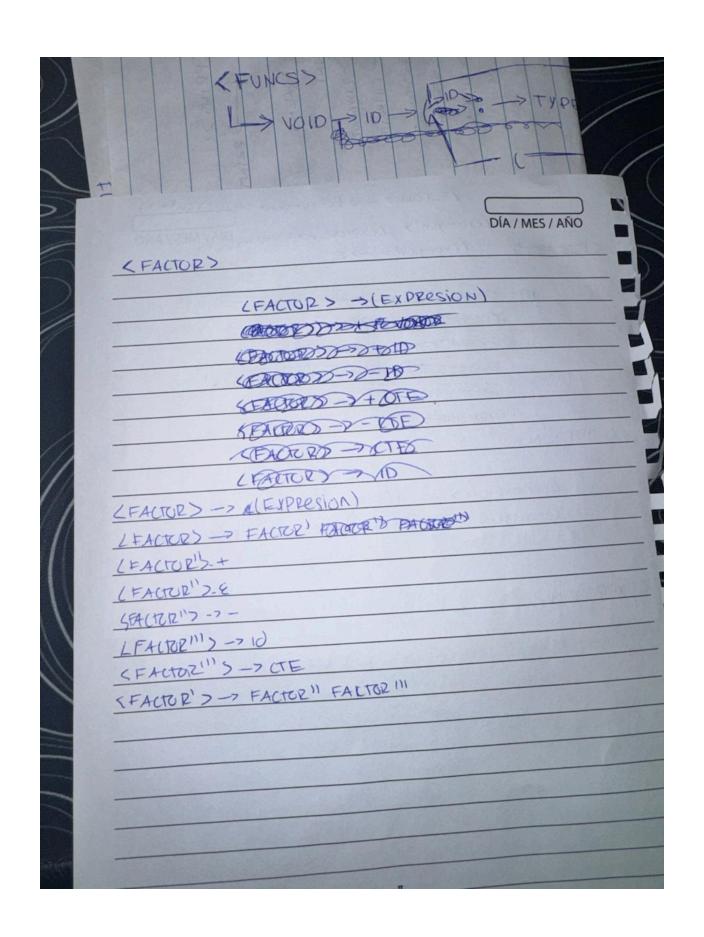
1	KFUNCS > 110 > 100 > TYPE
li	DÍA / MES / AÑO
	CPRINT >
	- CPRINTS - PRINT ( DAPPER TON)
	PRINT -> PREMOTO EXPRESION PRINTIL
	PRINT' -> EATE STRING CTE-STRING PRINT"
	PRINTI-> 6, PRINTI
5	PRINT 11 -> E
	TO THE TOTAL PROPERTY OF THE PARTY OF THE PA
Te le	A STATE OF THE PARTY OF THE PAR
	<u> ZPROARAMA</u> >
M	The state of the s
2	CPROFRAMA) -
	1 3 - CHANGES
1	Gerogram id ; cerogram's main Body end
1	LPVOGRAM'> CVARS> LDYCOGRAM'>
1	LPropen' > (pras)(program")
	cprogram 11 > cfuncs > 6 program 11>
	LONDGROWN > LEVACS 74E
展	
1	
0 -	



	DÍA / MES / AÑO
<b>ZSTATEM</b>	
	STATEMENT -> ASSIAN
	STATEMENT -> CONDITION
	STATEMENT -> CYCIE
	STATEMENT -> F_CALL
	STATUMENT -> PRINT
ZIERMU ZEMONOWA ZEMON	6)
[ Barows	
TERMAN ->	FACTOR TERMINO
TOPMINO' -	-> . FACTOR TERMINO'
TERMINO' -	-> I FACTOR TORMINO'
TERNILO'	3 6
1212	
(BODY)	
2 51/	ATEMENTS STATEMENT -> STATEMENT
	STATEMENT 1 -> E
	STATEMENT) -> STOTEMENT
	AND
LVARS)	
	A = Id A' B = A: TYPE; B'
	$A^2 = A$ $B^3 = E$
	A'=E $B'=B$
	101



A I I I I I I I I I I I I I I I I I I I	
(F- (all > -> od (CExpression) (expression) ))	
Lexpresion > scexpresion > cexpresion > cexpresion > 2	
Lexpresion > -> E	5
M) SCALLS 210 (EXPLESION);	
(DOOR DOP (P)	
Company of the second of	
(Expresion" Expresion	
(EXPrejon) -> & Exprision'	
(Expresion) => E	
LCONDITIONS	
	11
CCONDITION > -> If (EXPRESION) BODY;	
(CONDITION) -> if (EXPRESION) BODY EISE BODY;	
(EXPRESION)	
- CEAPTESION >	
CEXPRESION > -> EXP EXP'I	
EXP' -> > EXP'	
EXPII -> CEXPII	
Exp'1-7 != Exp"	
	3/2
_ h	
LOGRAND —	
ENTERTAINMENT GROUP	
	4. 10



Genera, como parte de la entrega, una documentación que describa brevemente los principales hallazgos del análisis de las diferentes herramientas. Agrega, a la documentación previa, cómo (en qué formato) diste de alta las reglas de construcción de Patito que contenga la definición de las expresiones regulares y reglas gramaticales desarrolladas. Agrega los principales Test-cases desarrollados para validar su funcionamiento. Considera que este documento irá creciendo conforme trabajes en las siguientes entregas.

\*\*Tests del Lexer:\*\*

cd tests → python3|python test\_lexer.py

1 Reconocimiento de palabras reservadas

2 Validación de identificadores con letras, números y guiones bajos

3 Reconocimiento de constantes enteras (positivas, negativas)

4 Reconocimiento de constantes flotantes con signo

5 Manejo de cadenas literales entre comillas

6 Operadores aritméticos y comparación

7 Delimitadores y puntuación (paréntesis, llaves, corchetes, comas, etc.)

8 Ignorar comentarios de línea y de bloque

```
**Tests del Parser:**
cd tests → python3 python test_parser.py
1 Estructura mínima de programa válido
2 Declaraciones de variables múltiples con tipos
3 Asignaciones simples
4 Expresiones aritméticas con operadores múltiples
5 Expresiones de comparación
6 Sentencias if simples
7 Sentencias if-else completas
8 Ciclos while-do
9 Sentencias print con múltiples argumentos
10 Declaración de funciones con parámetros
11 Llamadas a funciones con argumentos
12 Expresiones complejas con precedencia y paréntesis
13 Detección de errores sintácticos (ej. punto y coma faltante)
14 Programa completo integrando todas las características
```