

Gramática Tipo 2 (Notación BNF)

Analizador Sintáctico - JavaBridge

Símbolos Terminales

```
TERMINALES = {  
    'public', 'class', 'static', 'void', 'main', 'String', 'args',  
    'int', 'double', 'char', 'boolean', 'true', 'false',  
    'if', 'else', 'for', 'while', 'System', 'out', 'println',  
    '{', '}', '(', ')', '[', ']', ';', ',', '.',  
    '==', '!=', '>=', '<=', '>', '<',  
    '++', '--', '&&', '||',  
    '=', '+', '-', '*', '/',  
    IDENTIFICADOR, INT, DOUBLE, STRING, CHAR, TRUE, FALSE  
}
```

Símbolos No Terminales

```
NO_TERMINALES = {  
    <PROGRAMA>, <MAIN>, <SENTENCIAS>, <SENTENCIA>,  
    <DECLARACION>, <ASIGNACION_O_INCREMENTO>, <PRINT>,  
    <IF_ELSE>, <FOR>, <WHILE>,  
    <DECLARACION_FOR>, <EXPRESION_FOR>,  
    <EXPRESION>, <EXPRESION_LOGICA>, <EXPRESION_RELACIONAL>,  
    <EXPRESION_ARITMETICA>, <TERMINO>, <FACTOR>,  
    <TIPO>, <LISTA_VARIABLES>, <VARIABLE>  
}
```

Símbolo Inicial

<PROGRAMA>

Reglas de Producción

1. Estructura Principal del Programa

<PROGRAMA> ::= 'public' 'class' IDENTIFICADOR '{' <MAIN> '}'

<MAIN> ::= 'public' 'static' 'void' 'main' '(' 'String' '[' ']' 'args'
')' '{' <SENTENCIAS> '}'

```

<EXPRESSION_FOR> ::= IDENTIFICADOR '++'
                    | IDENTIFICADOR '--'
                    | IDENTIFICADOR '=' <EXPRESSION>

```

8. Estructura WHILE

<WHILE> ::= 'while' '(' <EXPRESION> ')' '{' <SENTENCIAS> '}'

9. Expresiones

<EXPRESION> ::= <EXPRESION_LOGICA>

<EXPRESION_LOGICA> ::= <EXPRESION_RELACIONAL> '&&' <EXPRESION_LOGICA>
| <EXPRESION_RELACIONAL> '||' <EXPRESION_LOGICA>
| <EXPRESION_RELACIONAL>

<EXPRESION_RELACIONAL> ::= <EXPRESION_ARITMETICA> '=='
<EXPRESION_RELACIONAL>

| <EXPRESION_ARITMETICA> '!='

<EXPRESION_RELACIONAL> | <EXPRESION_ARITMETICA> '>'
<EXPRESION_RELACIONAL>

| <EXPRESION_ARITMETICA> '<'

<EXPRESION_RELACIONAL> | <EXPRESION_ARITMETICA> '>='
<EXPRESION_RELACIONAL>

| <EXPRESION_ARITMETICA> '<='

<EXPRESION_RELACIONAL> | <EXPRESION_ARITMETICA>

<EXPRESION_ARITMETICA> ::= <TERMINO> '+' <EXPRESION_ARITMETICA>
| <TERMINO> '-' <EXPRESION_ARITMETICA>
| <TERMINO>

<TERMINO> ::= <FACTOR> '*' <TERMINO>
| <FACTOR> '/' <TERMINO>
| <FACTOR>

<FACTOR> ::= IDENTIFICADOR
| INT
| DOUBLE
| STRING
| CHAR
| 'true'
| 'false'
| '(' <EXPRESION> ')'

Precedencia de Operadores

Nivel	Operadores	Asociatividad
1 (Mayor)	*, /	Izquierda
2	+, -	Izquierda
3	>, <, >=, <=	Izquierda
4	==, !=	Izquierda
5 (Menor)	&&, ^	

Ejemplo de Derivación

Dado el programa:

```
public class Test {  
    public static void main(String[] args) {  
        int x = 5;  
        System.out.println(x);  
    }  
}
```

Derivación:

```
<PROGRAMA>  
⇒ 'public' 'class' IDENTIFICADOR '{' <MAIN> '}'  
⇒ 'public' 'class' Test '{' <MAIN> '}'  
⇒ 'public' 'class' Test '{' 'public' 'static' 'void' 'main' '(' 'String'  
'[' ']' 'args' ')' '{' <SENTENCIAS> '}' '}'  
⇒ ... 'int' IDENTIFICADOR '=' <EXPRESION> ';' <SENTENCIAS> ...  
⇒ ... 'int' x '=' <EXPRESION> ';' <SENTENCIAS> ...  
⇒ ... 'int' x '=' INT ';' <SENTENCIAS> ...  
⇒ ... 'int' x '=' 5 ';' <SENTENCIAS> ...  
⇒ ... <PRINT> <SENTENCIAS> ...  
⇒ ... 'System' '.' 'out' '.' 'println' '(' <EXPRESION> ')' ';' ε
```