

## GRAMÁTICA

La gramática a utilizar es tipo 2 basándonos en la jerarquía de Chomsky debido a que en dicha jerarquía encontramos que la gramática libre de contexto sigue la siguiente estructura:

$$x \rightarrow y$$

$$X \in NT$$

$$Y \in (NT|T)^*$$

Donde x puede ser reemplazado por y si x pertenece a los símbolos NO TERMINALES e y es un TERMINAL o NO TERMINAL incluyendo a su vez la cadena vacía.

### ❖ ALFABETO

$$\Sigma = [\{a,b,c,\dots,x,y,z\}, \{A,B,C,\dots,X,Y,Z\}, \{0,1,2,3,4,\dots,9\}]$$

### ❖ OPERADORES ARITMETICOS

| OPERADORES |                    |
|------------|--------------------|
| *          | Multiplicación     |
| -          | Resta/Substracción |
| +          | Suma/Adición       |
| /          | División           |

### ❖ OPERADORES LOGICOS

| OPERADORES |     |
|------------|-----|
| &&         | AND |
|            | OR  |

### ❖ OPERADORES RELACIONALES

| OPERADORES |                   |
|------------|-------------------|
| ==         | Igual a           |
| >          | Mayor que         |
| <          | Menor que         |
| >=         | Mayor o igual que |
| <=         | Menor o igual que |
| !=         | Diferente de      |

## ❖ TIPOS DE DATO

| TIPOS DE DATO |                             |
|---------------|-----------------------------|
| int           | entero de 32 bits           |
| char          | carácter Unicode de 16 bits |

## ❖ PALABRAS RESERVADAS

| PALABRAS RESERVADAS |                             |
|---------------------|-----------------------------|
| begin               | Inicio de programa          |
| end                 | Fin de programa             |
| main                | Función principal           |
| if                  | Condición si                |
| else                | sentencia entonces          |
| while               | Ciclo mientras              |
| input               | Entrada de datos            |
| output              | Salida de datos             |
| int                 | entero de 32 bits           |
| char                | carácter Unicode de 16 bits |

## ❖ DELIMITADORES

| DELIMITADORES |                        |
|---------------|------------------------|
| IMPLICITOS    |                        |
| begin         | Inicio del programa    |
| end           | Final del programa     |
| main          | Función principal      |
|               |                        |
| EXPLICITOS    |                        |
| ;             | Fin de línea de código |
| {             | Inicio para bloque     |
| }             | Fin de bloque          |
| ()            | Agrupación             |
|               |                        |

## ❖ SENTENCIAS CONDICIONALES

| if   | Condición si       |
|------|--------------------|
| else | Condición entonces |

## ❖ SENTENCIAS DE ITERACION

|       |                      |
|-------|----------------------|
| while | Ciclo 'mientras que' |
|-------|----------------------|

## ❖ ENTRADAS Y SALIDAS DE DATOS

| input  | Entrada de datos por teclado |
|--------|------------------------------|
| output | Salida de datos por pantalla |

## ❖ COMENTARIOS

|             |                            |
|-------------|----------------------------|
| #comentario | Comentario precedido por # |
|-------------|----------------------------|

## GRAMATICA ESTILO BNF

```
programa ::= 'begin'
           'main' '(' ')' '{'
               variables | operaciones_aritmeticas | sentencias_control |
               sentencias_iterativas | comentarios | captura_datos | despliegue_datos
           '}'
           'end'
variables ::= tipo_variable nombre_variable (valor)? ';'
tipo_variable ::= 'int' | 'char'
nombre_variable ::= letra numeros | palabras
letra ::= 'a-z' | 'A-Z'
numeros ::= '0-9'
palabras ::= { letra } +
valor ::= { numeros } +
operaciones_aritmeticas ::= numeros | variables operador numeros | variables
operador ::= '*' | '+' | '-' | '/'
sentencias_control ::= 'if' '(' condicion ')' '{'
                    acciones
                    '}'
condicion ::= numeros | variables operador comparacion numeros | variables
operador_comparacion ::= '==' | '>' | '<' | '>=' | '<=' | '!='
acciones ::= operaciones_aritmeticas | sentencias_control | sentencias_iterativas |
variables | captura_datos | despliegue_datos
sentencias_iterativas ::= 'while' '(' condicion ')' '{'
                    acciones
                    '}'
captura_datos ::= 'input' '(' ')'
despliegue_datos ::= 'output' '(' ')'
comentarios ::= '#' numeros | letra
```