

Terminales Json

Terminal	Patron
DIGITO	[0-9]
DOSPTO	"."
BRAIZQ	"{"
BRADER	"}"
COMA	","
CADENA	"\" [^\"\\\\]* \"\\\\\"* [^\"\\\\]* [^\"\\\\]* \"\\\\\"* \"\""

Gramática Json

```
<Inicio> ::= BRAIZQ <listainstrucciones> BRADER
;

<listainstrucciones> ::= <listainstrucciones> COMA <instruccion>
                        | <instruccion>
;

<instruccion> ::= <declaracion>
;

<declaracion> ::= CADENA DOSPTO CADENA
                | CADENA DOSPTO DECIMAL
;
```

Terminales StatPy

Terminal	Patrón
FOR	"for"
DO	"do"
WHILE	"while"
BREAK	"break"
CASE	"case"
SWITCH	"switch"
DEFAULT	"default"
TRUE	"True"
FALSE	"False"
IF	"if"
ELSE	"else"
INT	"int"
DOUBLE	"double"
STRING	"string"
PCHAR	"char"
BOOL	"bool"
VOID	"void"
MAIN	"main"
DEFGLOB	"DefinirGlobales"
GRAFBAR	"GraficaBarras"
GRAFPI	"GraficaPie"
NEWVAL	"NewValor"
TITULO	"Titulo"
EJEX	"EjeX"
VALORES	"Valores"
TITULOY	"TituloY"
TITULOX	"TituloX"
CONSOLE	"Console"
WRITE	"Write"
MAYOR	">"
MENOR	"<"
MAYORIGUAL	">="
MENORIGUAL	"<="
EQUALS	"=="
NOTEQUALS	"!="
AND	"&&"
OR	" "
NOT	"!"
MAS	"+"

MENOS	"_"
POR	"*"
DIVIDIDO	"/"
PARIZQ	"("
PARDER	")"
BRAIZQ	"{"
BRADER	"}"
LLAIZQ	"["
LLADER	"]"
PTOCOMA	","
IGUAL	"="
DOLAR	"\$"
COMA	","
PUNTO	","
IDENT	[a-zA-Z_ñÑ][a-zA-Z_ñÑ0-9]*
CADENA	"([^\n]"
CHAR	'([^\t\n]"
ENTERO	[0-9]+
DECIMAL	[0-9]+(".[0-9]+)?

Gramática StatPy

```

<inicio> ::= VOID MAIN PARIZQ PARDER BRAIZQ <listainstrucciones> BRADER
;

<listainstrucciones> ::= <listainstrucciones> <instrucciones>
                        | <instrucciones>
;

<instrucciones> ::= <sentencias>
                  | <globales>
                  | <barras>
                  | <pie>
;

<sentencias> ::= <declaracion>
               | <print>
               | <if>
               | <while>
               | <for>
               | <dowhile>
               | <switch>
;

```

```

<if> ::= IF PARIZQ <expresion> PARDER BRAIZQ <listasentencias> BRADER
      | ELSE IF PARIZQ <expresion> PARDER BRAIZQ <listasentencias> BRADER
      | ELSE BRAIZQ <listasentencias> BRADER
;

<while> ::= WHILE PARIZQ <expresion> PARDER BRAIZQ <listasentencias> BRADER
;

<for> ::= FOR PARIZQ INT IDENT <asignacion> PTOCOMA IDENT <condicion>
expresion <paso> PARDER BRAIZQ <listasentencias> BRADER
      | FOR PARIZQ INT IDENT <asignacion> PTOCOMA IDENT <condicion>
expresion <paso> PARDER BRAIZQ <listasentencias> BRADER
      | FOR PARIZQ IDENT <asignacion> PTOCOMA IDENT <condicion> expresion
<paso> PARDER BRAIZQ <listasentencias> BRADER
      | FOR PARIZQ IDENT <asignacion> PTOCOMA IDENT <condicion> expresion
<paso> PARDER BRAIZQ <listasentencias> BRADER
;

<dowhile> ::= DO BRAIZQ <listasentencias> BRADER WHILE PARIZQ <expresion>
PARDER PTOCOMA
;

<switch> ::= SWITCH PARIZQ IDENT PARDER BRAIZQ <listacasos> BRADER
;

<listacasos> ::= <listacasos> <caso>
              | <caso>
;

<caso> ::= CASE <expresion> DOSPTO <listasentencias>
          | DEFAULT DOSPTO <listasentencias>
;

<listasentencias> ::= <listasentencias> <sentencias>
                    | <sentencias>
;

<declaracion> ::= <tipodato> IDENT IGUAL <expresion> PTOCOMA
                 | IDENT IGUAL expresion PTOCOMA
                 | IDENT MAS MAS PTOCOMA
                 | BREAK PTOCOMA
;

<print> ::= CONSOLE PUNTO WRITE PARIZQ expresion PARDER PTOCOMA

```

```

;

<expresion> ::= <symbols>
            | <uno>
            | <aritmetica>
            | <logica>
            | PARIZQ <expresion> PARDER
;

<symbols> ::= IDENT
          | CADENA
          | CHAR
          | TRUE
          | FALSE
;

<aritmetica> ::= expresion MAS expresion
             | expresion MENOS expresion
             | expresion POR expresion
             | expresion DIVIDIDO expresion
             | DECIMAL
             | ENTERO
;

<logica> ::= expresion MAYOR expresion
          | expresion MENOR expresion
          | expresion MAYORIGUAL expresion
          | expresion MENORIGUAL expresion
          | expresion EQUALS expresion
          | expresion NOTEQUALS expresion
          | expresion AND expresion
          | expresion OR expresion
;

<uno> ::= MENOS expresion %prec Umenos
      | NOT expresion %prec NOT
;

<globales> ::= VOID DEFGLOB PARIZQ PARDER BRAIZQ <listaglobales> BRADER
;

<listaglobales> ::= <listaglobales> <variablesglobales>
                  | <variablesglobales>
;

```

```

<variablesglobales> ::= <declaracion>
                        | IDENT IGUAL <expresion> PTOCOMA
;

<barras> ::= VOID GRAFBAR PARIZQ PARDER BRAIZQ <listabarras> BRADER
;

<listabarras> ::= <listabarras> <instruccionesbarras>
                | <instruccionesbarras>
;

<instruccionesbarras> ::= STRING TITULO IGUAL <expresion> PTOCOMA
                        | STRING TITULOX IGUAL <expresion> PTOCOMA
                        | STRING TITULOY IGUAL <expresion> PTOCOMA
                        | STRING LLAIZQ LLADER EJEX IGUAL BRAIZQ <listastring>
BRADER PTOCOMA
                        | DOUBLE LLAIZQ LLADER VALORES IGUAL BRAIZQ
<listadouble> BRADER PTOCOMA
;

<listastring> ::= <listastring> COMA <arreglostring>
                | <arreglostring>
;

<arreglostring> ::= CADENA
                  | IDENT
                  | variablesjson
;

<listadouble> ::= <listadouble> COMA <arreglodouble>
                | <arreglodouble>
;

<arreglodouble> ::= DECIMAL
                  | IDENT
                  | variablesjson
                  | ENTERO
;

<pie> ::= VOID GRAFPI PARIZQ PARDER BRAIZQ <listapie> BRADER
;

<listapie> ::= <listapie> <instruccionespie>
              | <instruccionespie>
;

```

```

<instruccionespie> ::= STRING TITULO IGUAL <expresion> PTOCOMA
                        | STRING LLAIZQ LLADER EJEX IGUAL BRAIZQ
<listastringpie> BRADER PTOCOMA
                        | DOUBLE LLAIZQ LLADER VALORES IGUAL BRAIZQ
<listadoublepie> BRADER PTOCOMA
;

<listastringpie> ::= <listastringpie> COMA <arreglostringpie>
                        | <arreglostringpie>
;

<listadoublepie> ::= <listadoublepie> COMA <arreglodoublepie>
                        | <arreglodoublepie>
;

<arreglostringpie> ::= CADENA
                        | IDENT
                        | variablesjson
;

<arreglodoublepie> ::= DECIMAL
                        | IDENT
                        | variablesjson
                        | ENTERO
;

```