

# GRAMÁTICA LNZ

terminal COMA, LIENZOS, LLAVE\_A, LLAVE\_B, DOS\_PUNTOS, COMILLAS, nombre, tipo, Fondo, size, Red, Green, Blue, cuadro, dimension\_x, dimension\_y, png, gif, HEX, NUMERO, ID, codigoHex, PALABRA;

non terminal a, b, COMA2, c, COMA1, f, e, COMA3, g, COMA4;

start with a;

a ::= LLAVE\_A LIENZOS DOS\_PUNTOS LLAVE\_A b LLAVE\_B LLAVE\_B  
;

b ::= ID DOS\_PUNTOS LLAVE\_A c LLAVE\_B COMA2  
;

c ::= nombre DOS\_PUNTOS COMILLAS PALABRA COMILLAS COMA1  
| tipo DOS\_PUNTOS COMILLAS png COMILLAS COMA1  
| tipo DOS\_PUNTOS COMILLAS gif COMILLAS COMA1  
| Fondo DOS\_PUNTOS LLAVE\_A e LLAVE\_B COMA1  
| Fondo DOS\_PUNTOS LLAVE\_A f LLAVE\_B COMA1  
| size DOS\_PUNTOS LLAVE\_A g LLAVE\_B COMA1  
| error COMA1  
;

e ::= Red DOS\_PUNTOS NUMERO COMA3  
| Blue DOS\_PUNTOS NUMERO COMA3  
| Green DOS\_PUNTOS NUMERO COMA3  
;

f ::= HEX DOS\_PUNTOS codigoHex  
;

COMA1 ::= COMA c  
|  
;

COMA2 ::= COMA b  
|  
;

COMA3 ::= COMA e  
|  
;

COMA4 ::= COMA g  
|  
;

# GRAMÁTICA CLRS

terminal COLORES, Green, Red, Blue, HEX, LLAVE\_A, LLAVE\_B, DOS\_PUNTOS, NUMERO, ID  
codigoHex, COMA;

non terminal a, d, COMA3, e, b, COMA1, COMA2, c;

start with a;

a ::= LLAVE\_A COLORES DOS\_PUNTOS LLAVE\_B b LLAVE\_B LLAVE\_B  
;

b ::= ID DOS\_PUNTOS LLAVE\_A c LLAVE\_B COMA1  
| error  
;

c ::= ID DOS\_PUNTOS LLAVE\_A d LLAVE\_B COMA2  
| ID DOS\_PUNTOS LLAVE\_A e LLAVE\_B COMA2  
;

d ::= Red DOS\_PUNTOS NUMERO COMA3  
| Blue DOS\_PUNTOS NUMERO COMA3  
| Green DOS\_PUNTOS NUMERO COMA3  
;

e ::= HEX DOS\_PUNTOS codigoHex  
;

COMA1 ::= COMA b  
|  
;

COMA2 ::= COMA c  
|  
;

COMA3 ::= COMA d  
|  
;

# GRAMÁTICA TMP

terminal COMA, DOS\_PUNTOS, COMILLAS, AGRUPACION\_A, AGRUPACION\_B, ID, NUMERO, TIEMPOS, inicio, fin, imagenes, id, duracion, LLAVE\_A, LLAVE\_B;

non terminal a, COMA1, b, c, COMA2, COMA3, d, COMA4, e;

start with a;

a ::= LLAVE\_A TIEMPOS DOS\_PUNTOS LLAVE\_A b LLAVE\_B LLAVE\_B  
;

b ::= ID DOS\_PUNTOS LLAVE\_A c LLAVE\_B COMA1  
| error COMA1  
;

c ::= inicio DOS\_PUNTOS COMILLAS ID COMILLAS COMA2  
| fin DOS\_PUNTOS COMILLAS ID COMILLAS COMA2  
| imagenes DOS\_PUNTOS AGRUPACION\_A d AGRUPACION\_B COMA2  
;

d ::= LLAVE\_A e LLAVE\_B COMA3  
;

e ::= id DOS\_PUNTOS COMILLAS ID COMILLAS COMA4  
| duracion DOS\_PUNTOS NUMERO COMA4  
| error COMA4  
;

COMA1 ::= COMA b  
|  
;

COMA2 ::= COMA c  
|  
;

COMA3 ::= COMA d  
|  
;

COMA4 ::= COMA e  
|  
;

# GRAMÁTICA PNT

terminal COMA, COMENTARIO\_A, PUNTO\_COMA, COMILLAS, RANGO, DOBLE\_IGUAL,  
IGUAL, MENOR, MAYOR, MENOR\_IGUAL,  
MAYOR\_IGUAL, DIFERENCIA, LLAVE\_A, LLAVE\_B, PARENTESIS\_A,  
PARENTESIS\_B, AGRUPACION\_A, AGRUPACION\_B,  
MAS, MENOS, POR, COMENTARIO\_B\_A, COMENTARIO\_B\_B, VARS, INT, STRING,  
BOOLEAN, TRUE, FALSE,  
INSTRUCCIONES, PINTAR, IF, ELSE, WHILE, NUMERO, ID, AND, OR, DIV;

non terminal a, b, b3, b4, b6, b8, c, c1, c4, c5, COMA1, COMA2, COMA3, COMA4, else1, b1, b5, b2  
b2prima, b2prima1, b2prima2, relacional, b2prima3, b2prima4, c3, ext, c2, bools, idColor,  
idImagen, posX, posY;

precedence left MAS, MENOS;

precedence left POR, DIV;

start with a;

a ::= b c  
| error  
;

b ::= VARS AGRUPACION\_A b1 AGRUPACION\_B  
| error  
;

b1 ::= INT ID IGUAL b2 b3 PUNTO\_COMA b1  
| INT ID b3 PUNTO\_COMA b1  
| STRING ID IGUAL COMILLAS ID COMILLAS b4 PUNTO\_COMA b1  
| STRING ID b4 PUNTO\_COMA b1  
| BOOLEAN ID b5 PUNTO\_COMA b1  
| BOOLEAN ID IGUAL TRUE b5 PUNTO\_COMA b1  
| BOOLEAN ID IGUAL FALSE b5 PUNTO\_COMA b1  
|  
;

b2prima3 ::= b2prima3 MAS b2prima3  
| b2prima3 MENOS b2prima3  
| b2prima3 POR b2prima3  
| b2prima3 DIV b2prima3  
| ID  
| NUMERO  
;

```

b2prima4 ::= b2prima4 MAS b2prima4
|           b2prima4 MENOS b2prima4
|           b2prima4 POR b2prima4
|           b2prima4 DIV b2prima4
|           ID
|           NUMERO
;

```

```

b3 ::= COMA ID IGUAL b2prima3
|
;

```

```

b4 ::= COMA ID IGUAL COMILLAS ID COMILLAS b4
|     COMA ID b4
;

```

```

b5 ::= COMA ID b5
|     COMA ID IGUAL TRUE b5
|     COMA ID IGUAL FALSE b5
;

```

```

c ::= INSTRUCCIONES PARENTESIS_A ID PARENTESIS_B AGRUPACION_A c1
AGRUPACION_B COMA4
;

```

```

    c1 ::= PINTAR PARENTESIS_A idColor COMA idImagen COMA posX COMA posY
PARENTESIS_B PUNTO_COMA c1
| IF PARENTESIS_A c2 PARENTESIS_B LLAVE_A c1 LLAVE_B else1
| WHILE PARENTESIS_A c2 PARENTESIS_B LLAVE_A c1 LLAVE_B c1
| b8 c1
|
| error c1
;

```

```

else1 ::= ELSE LLAVE_A c1 LLAVE_B c1
|       c1
;

```

```

b8 ::= ID IGUAL b2prima3 PUNTO_COMA
|     ID IGUAL COMILLAS ID COMILLAS PUNTO_COMA
|     ID IGUAL TRUE PUNTO_COMA
|     ID IGUAL FALSE PUNTO_COMA
;

```

```

c2 ::= bools ext
;

```

```

c3 ::= ID
|     NUMERO

```

```
ext ::= AND bools
|      OR bools
|      error
;
```

```
bools ::= c3 relacional c3
|      ID
|      TRUE
|      FALSE
|      error bools
;
```

```
idColor ::= ID
|      COMILLAS ID COMILLAS
;
```

```
idImagen ::= ID
|      COMILLAS ID COMILLAS
;
```

```
posX ::= ID RANGO ID
|      NUMERO RANGO NUMERO
|      ID RANGO NUMERO
|      NUMERO RANGO ID
|      b2prima4
;
```

```
posY ::= ID RANGO ID
|      NUMERO RANGO NUMERO
|      ID RANGO NUMERO
|      NUMERO RANGO ID
|      b2prima4
;
```

```
COMA4 ::= c
|
;
```

```
relacional ::= DOBLE IGUAL
|      MENOR
|      MAYOR
|      MENOR_IGUAL
|      MAYOR_IGUAL
|      DIFERENCIA
```