Gramática BNF

Analizador JSON

```
ini::=instrucciones;

instrucciones ::= instruccion instrucciones
 | instruccion
;

instruccion ::=
    LLI declaraciones:a
LLC                                                {:cf.escribirJSON((S
tring) a);:}
;

declaraciones ::= declaracion:a
declaraciones:b                                {:RESULT=String.valueOf(a)
+"\n"+ String.valueOf(b);:}
 |
declaracion:a
 {:RESULT=String.valueOf(a);:}
;

declaracion ::=
  COMD CADENA:a COMD DOSPT
DECIMAL:b                                            {:RESULT=a+":"+b;:}
  | COMD CADENA:a COMD DOSPT COMD CADENA:b
COMD                                {:RESULT=a+":"+b;:}
  | COMD CADENA:a COMD DOSPT DECIMAL:b
COMA                                    {:RESULT=a+":"+b;:}
  | COMD CADENA:a COMD DOSPT COMD CADENA:b COMD
COMA                            {:RESULT=a+":"+b;:}
;
```

```
CADENA ::=
    identificador:a
    {:RESULT=a;:}
    |
DECIMAL:a                                                    {:
RESULT=a;:}
    | DECIMAL:a
CADENA:b                                                {:RESULT=a+"
"+b;:}
    | identificador:a
CADENA:b                                              {:RESULT=a+"
"+b;:}
;
```

Analizador StatPy

```
ini::=instrucciones
;

instrucciones ::= instruccion:a
instrucciones:b                                    {:cf.Escribir(String.valueOf
(a)+"\n"+String.valueOf(b));htmlTokens(toks);:}
 |
instruccion:a
 {:cf.Escribir(String.valueOf(a));htmlTokens(toks);:}
;

instruccion ::=
    main:a
   {:RESULT="def main():\n\t"+String.valueOf(a)+"\nif __name__ ==
\"__main__\":\n\tmain()\n\n";:}
;

main ::=
    wVoid:t1 wMain:t2 PAR_IZQ:t3 PAR_DER:t4 LLI:t5 declaraciones:a
LLC:t6
    {:
    RESULT=String.valueOf(a);
    listaTokens(t1,"wVoid",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"wMain",String.valueOf(t2left),String.valueOf(t2right));
    listaTokens(t3,"PAR_IZQ",String.valueOf(t3left),String.valueOf(t3right))
;
    listaTokens(t4,"PAR_DER",String.valueOf(t4left),String.valueOf(t4right))
;
    listaTokens(t5,"LLI",String.valueOf(t5left),String.valueOf(t5right));
    listaTokens(t6,"LLC",String.valueOf(t6left),String.valueOf(t6right));
    :}
;

declaraciones ::= declaracion:a
declaraciones:b                                    {:RESULT="\n\t"+String.value
Of(a) +"\n\t"+ String.valueOf(b);:}
 |
declaracion:a
 {:RESULT=String.valueOf(a);:}
;
```

```
declaracion ::=
    declaracionINT:a
   {:RESULT=a;:}
    |
declaracionDOUBLE:a                                                {:
RESULT=a;:}
    |
declaracionCHAR:a                                                  {:
RESULT=a;:}
    |
declaracionBOOL:a                                                  {:
RESULT=a;:}
    |
declaracionSTRING:a                                                {:
RESULT=a;:}
    |
declaracionPRINT:a                                                 {:
RESULT=a;:}
    |
sentenciaIF:a                                                      {:
RESULT=a;:}
    |
sentenciaELSE:a                                                    {:
RESULT=a;:}
    |
sentenciaSWITCH:a                                                  {:
RESULT=a;:}
    |
sentenciaFOR:a                                                     {:
RESULT=a;:}
    |
sentenciaWHILE:a                                                   {:
RESULT=a;:}
    |
sentenciaDOWHILE:a                                                 {:
RESULT=a;:}
    |
declaracionFUNCIONES                                               {:
RESULT="";:}
;

sentenciaIF ::=
    wIF:t1 PAR_IZQ:t2 CONDICIONES:a PAR_DER:t3 LLI:t4 declaraciones:b LLC:t5
    {:
```

```
    RESULT = "if " + a +":\n" + String.valueOf(b);
    listaTokens(t1,"wIF",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"PAR_IZQ",String.valueOf(t2left),String.valueOf(t2right))
;
    listaTokens(t3,"PAR_DER",String.valueOf(t3left),String.valueOf(t3right))
;
    listaTokens(t4,"LLI",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"LLC",String.valueOf(t5left),String.valueOf(t5right));
    :}
;

sentenciaELSE ::=
    wELSE:t1 sentenciaIF:a
    {:
    RESULT = "el"+a;
    listaTokens(t1,"wELSE",String.valueOf(t1left),String.valueOf(t1right));
    :}
;

sentenciaSWITCH ::=
    wSWITCH:t1 PAR_IZQ:t2 identificador:a PAR_DER:t3 LLI:t4 sentenciaCASE:b
LLC:t5
    {:
    RESULT ="def switcher(case,"+a+"):\n\tswitcher={"+b+"\n\t}" ;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"wSWITCH",String.valueOf(t1left),String.valueOf(t1right))
;
    listaTokens(t2,"PAR_IZQ",String.valueOf(t2left),String.valueOf(t2right))
;
    listaTokens(t3,"PAR_DER",String.valueOf(t3left),String.valueOf(t3right))
;
    listaTokens(t4,"LLI",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"LLC",String.valueOf(t5left),String.valueOf(t5right));
    :}
;

sentenciaCASE::=
    sentenciaCASE:a
CASE:b                                                  {: RESULT = "\t"
+a + "\n\t" + String.valueOf(b); :}
    |
CASE:a                                                          {:
RESULT = "\n\t" +a; :}
;
```

```
CASE ::=
    wCASE:t1 CADENA:a DOSPT:t2 declaraciones:b wBREAK:t3
PTCOMA:t4
    {:
    RESULT=a+":"+b;
    listaTokens(t1,"wCASE",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"DOSPT",String.valueOf(t2left),String.valueOf(t2right));
    listaTokens(t3,"wBREAK",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(t4,"PTCOMA",String.valueOf(t4left),String.valueOf(t4right));
    :}
;

sentenciaFOR::=
    wFOR:t1 PAR_IZQ:t2 declaracion:a identificador:b RELACIONES:c CADENA:d
PTCOMA:t3 identificador:f MAS:t4 MAS:t5 PAR_DER:t6 LLI:t7 declaraciones:g
LLC:t8
    {:
    String[] dec =
String.valueOf(a).split("=");

    RESULT="for "+dec[0]+" in range("+dec[1]+","+d+"):"+g;
    listaTokens(t1,"wFOR",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"PAR_IZQ",String.valueOf(t2left),String.valueOf(t2right))
;
    listaTokens(t3,"PAR_DER",String.valueOf(t3left),String.valueOf(t3right))
;
    listaTokens(b,"identificador",String.valueOf(bleft),String.valueOf(brigh
t));
    listaTokens(t3,"PTCOMA",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(f,"identificador",String.valueOf(fleft),String.valueOf(frigh
t));
    listaTokens(t4,"MAS",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"MAS",String.valueOf(t5left),String.valueOf(t5right));
    listaTokens(t6,"PAR_DER",String.valueOf(t6left),String.valueOf(t6right))
;
    listaTokens(t7,"LLI",String.valueOf(t7left),String.valueOf(t7right));
    listaTokens(t8,"LLC",String.valueOf(t8left),String.valueOf(t8right));
    :}
;

sentenciaWHILE::=
    wWHILE:t1 PAR_IZQ:t2 identificador:a RELACIONES:b CADENA:c PAR_DER:t3
LLI:t4 declaraciones:d LLC:t5
    {:
```

```
    RESULT = "while " + a+" "+b+" "+c+":\n" + String.valueOf(d);
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"wWHILE",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"PAR_IZQ",String.valueOf(t2left),String.valueOf(t2right))
;
    listaTokens(t3,"PAR_DER",String.valueOf(t3left),String.valueOf(t3right))
;
    listaTokens(t4,"LLI",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"LLC",String.valueOf(t5left),String.valueOf(t5right));
    :}
;

sentenciaDOWHILE::=
    wDO:t1 LLI:t2 declaraciones:a LLC:t3 wWHILE:t4 PAR_IZQ:t5
identificador:b RELACIONES:c expresion:d PAR_DER:t6 PTCOMA:t7
    {:
    RESULT="while True:"+a+"\n\t"+b+"="+b+"+1\n\tif("+b+c+d+"):\nbreak";
    listaTokens(b,"identificador",String.valueOf(bleft),String.valueOf(brigh
t));
    listaTokens(t1,"wDO",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"LLI",String.valueOf(t2left),String.valueOf(t2right));
    listaTokens(t3,"LLC",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(t4,"wWHILE",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"PAR_IZQ",String.valueOf(t5left),String.valueOf(t5right))
;
    listaTokens(t6,"PAR_DER",String.valueOf(t6left),String.valueOf(t6right))
;
    listaTokens(t7,"PTCOMA",String.valueOf(t7left),String.valueOf(t7right));
    :}
;

declaracionINT ::=
    varINT:t1 identificador:a IGUAL:t2 expresion:b
PTCOMA:t3
    {:
    RESULT="\n\t"+a+"="+b;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"varINT",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"IGUAL",String.valueOf(t2left),String.valueOf(t2right));
    listaTokens(t3,"PTCOMA",String.valueOf(t3left),String.valueOf(t3right));
    :}
    | varINT:t1 identificador:a
PTCOMA:t2
```

```
    {:
    RESULT="\n\t"+a;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"varINT",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"PTCOMA",String.valueOf(t2left),String.valueOf(t2right));
    :}
;

declaracionDOUBLE ::=
    varDOUBLE:t1 identificador:a IGUAL:t2 expresion:b
PTCOMA:t3
    {:
    RESULT="\n\t"+a+"="+b;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"varDOUBLE",String.valueOf(t1left),String.valueOf(t1right
));
    listaTokens(t2,"IGUAL",String.valueOf(t2left),String.valueOf(t2right));
    listaTokens(t3,"PTCOMA",String.valueOf(t3left),String.valueOf(t3right));
    :}
    | varDOUBLE:t1 identificador:a
PTCOMA:t2
    {:
    RESULT="\n\t"+a;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"varDOUBLE",String.valueOf(t1left),String.valueOf(t1right
));
    listaTokens(t2,"PTCOMA",String.valueOf(t2left),String.valueOf(t2right));
    :}
;

declaracionCHAR ::=
    varCHAR:t1 identificador:a IGUAL:t2 COMS:t3 identificador:b COMS:t4
PTCOMA:t5
    {:
    RESULT="\n\t"+a+"="+"\'"+b+"\'";
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(b,"identificador",String.valueOf(bleft),String.valueOf(brigh
t));
    listaTokens(t1,"varCHAR",String.valueOf(t1left),String.valueOf(t1right))
;
    listaTokens(t2,"IGUAL",String.valueOf(t2left),String.valueOf(t2right));
```

```
    listaTokens(t3,"COMS",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(t4,"COMS",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"PTCOMA",String.valueOf(t5left),String.valueOf(t5right));
    :}
    | varCHAR:t1 identificador:a IGUAL:t2 COMD:t3 identificador:b COMD:t4
PTCOMA:t5
    {:
    RESULT="\n\t"+a+"="+"\""+b+"\"";
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(b,"identificador",String.valueOf(bleft),String.valueOf(brigh
t));
    listaTokens(t1,"varCHAR",String.valueOf(t1left),String.valueOf(t1right))
;
    listaTokens(t2,"IGUAL",String.valueOf(t2left),String.valueOf(t2right));
    listaTokens(t3,"COMD",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(t4,"COMD",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"PTCOMA",String.valueOf(t5left),String.valueOf(t5right));
    :}
    | varCHAR:t1 identificador:a PTCOMA:t2
    {:
    RESULT="\n\t"+a;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"varCHAR",String.valueOf(t1left),String.valueOf(t1right))
;
    listaTokens(t2,"PTCOMA",String.valueOf(t2left),String.valueOf(t2right));
    :}
;

declaracionBOOL ::=
    varBOOL:t1 identificador:a IGUAL:t2 identificador:b PTCOMA:t3
    {:
    if(b.equals("true")){
    RESULT="\n\t"+a+"= True";
    }else if(b.equals("false")){
    RESULT="\n\t"+a+"= False";
    }
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(b,"identificador",String.valueOf(bleft),String.valueOf(brigh
t));
    listaTokens(t1,"varBOOL",String.valueOf(t1left),String.valueOf(t1right))
;
    listaTokens(t2,"IGUAL",String.valueOf(t2left),String.valueOf(t2right));
```

```
    listaTokens(t3,"PTCOMA",String.valueOf(t3left),String.valueOf(t3right));
    :}
;

declaracionSTRING ::=
    varSTRING:t1 identificador:a IGUAL:t2 COMS:t3 CADENA:b COMS:t4
PTCOMA:t5
    {:
    RESULT="\n\t"+a+"="+"\'"+b+"\'";
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"varSTRING",String.valueOf(t1left),String.valueOf(t1right
));
    listaTokens(t2,"IGUAL",String.valueOf(t2left),String.valueOf(t2right));
    listaTokens(t3,"COMS",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(t4,"COMS",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"PTCOMA",String.valueOf(t5left),String.valueOf(t5right));
    :}
    | varSTRING:t1 identificador:a IGUAL:t2 COMD:t3 CADENA:b COMD:t4
PTCOMA:t5
    {:
    RESULT="\n\t"+a+"="+"\""+b+"\"";
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"varSTRING",String.valueOf(t1left),String.valueOf(t1right
));
    listaTokens(t2,"IGUAL",String.valueOf(t2left),String.valueOf(t2right));
    listaTokens(t3,"COMD",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(t4,"COMD",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"PTCOMA",String.valueOf(t5left),String.valueOf(t5right));
    :}
    | varSTRING:t1 identificador:a
PTCOMA:t2
    {:
    RESULT=a;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"varSTRING",String.valueOf(t1left),String.valueOf(t1right
));
    listaTokens(t2,"PTCOMA",String.valueOf(t2left),String.valueOf(t2right));
    :}
;

declaracionPRINT ::=
```

```
    wPrint:t1 PAR_IZQ:t2 expresion:a PAR_DER:t3
PTCOMA:t4
    {:
    RESULT="\n\t"+"print("+a+")";
    listaTokens(t1,"wPRINT",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"PAR_IZQ",String.valueOf(t2left),String.valueOf(t2right))
;
    listaTokens(t3,"PAR_DER",String.valueOf(t3left),String.valueOf(t3right))
;
    listaTokens(t4,"PTCOMA",String.valueOf(t4left),String.valueOf(t4right));
    :}
    | wPrint:t1 PAR_IZQ:t2 COMS:t3 CADENA:a COMS:t4 MAS:t5 identificador:b
PAR_DER:t6 PTCOMA:t7
    {:
    RESULT="\n\t"+"print("+"\'"+a+"\'"+","+b+")";
    listaTokens(b,"identificador",String.valueOf(bleft),String.valueOf(brigh
t));
    listaTokens(t1,"wPRINT",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"PAR_IZQ",String.valueOf(t2left),String.valueOf(t2right))
;
    listaTokens(t3,"COMS",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(t4,"COMS",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"MAS",String.valueOf(t5left),String.valueOf(t5right));
    listaTokens(t6,"PAR_DER",String.valueOf(t6left),String.valueOf(t6right))
;
    listaTokens(t7,"PTCOMA",String.valueOf(t7left),String.valueOf(t7right));
    :}
    | wPrint:t1 PAR_IZQ:t2 COMS:t3 CADENA:a DOSPT:t4 COMS:t5 MAS:t6
identificador:b PAR_DER:t7 PTCOMA:t8
    {:
    RESULT="\n\t"+"print("+"\'"+a+"\'"+","+b+")";
    listaTokens(b,"identificador",String.valueOf(bleft),String.valueOf(brigh
t));
    listaTokens(t1,"wPRINT",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"PAR_IZQ",String.valueOf(t2left),String.valueOf(t2right))
;
    listaTokens(t3,"COMS",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(t4,"DOSPT",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"COMS",String.valueOf(t5left),String.valueOf(t5right));
    listaTokens(t6,"MAS",String.valueOf(t6left),String.valueOf(t6right));
    listaTokens(t7,"PAR_DER",String.valueOf(t7left),String.valueOf(t7right))
;
    listaTokens(t8,"PTCOMA",String.valueOf(t8left),String.valueOf(t8right));
    :}
```

```
    | wPrint:t1 PAR_IZQ:t2 COMS:t3 CADENA:a COMS:t4 PAR_DER:t5
PTCOMA:t6
    {:
    RESULT="\n\t"+"print("+"\'"+a+"\')";
    listaTokens(t1,"wPRINT",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"PAR_IZQ",String.valueOf(t2left),String.valueOf(t2right))
;
    listaTokens(t3,"COMS",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(t4,"COMS",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"PAR_DER",String.valueOf(t5left),String.valueOf(t5right))
;
    listaTokens(t6,"PTCOMA",String.valueOf(t6left),String.valueOf(t6right));
    :}
    | wPrint:t1 PAR_IZQ:t2 COMD:t3 CADENA:a COMD:t4 MAS:t5 identificador:b
PAR_DER:t6 PTCOMA:t7
    {:
    RESULT="\n\t"+"print("+"\""+a+"\""+","+b+")";
    listaTokens(b,"identificador",String.valueOf(bleft),String.valueOf(brigh
t));
    listaTokens(t1,"wPRINT",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"PAR_IZQ",String.valueOf(t2left),String.valueOf(t2right))
;
    listaTokens(t3,"COMD",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(t4,"COMD",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"MAS",String.valueOf(t5left),String.valueOf(t5right));
    listaTokens(t6,"PAR_DER",String.valueOf(t6left),String.valueOf(t6right))
;
    listaTokens(t7,"PTCOMA",String.valueOf(t7left),String.valueOf(t7right));
    :}
    | wPrint:t1 PAR_IZQ:t2 COMD:t3 CADENA:a DOSPT:t4 COMD:t5 MAS:t6
identificador:b PAR_DER:t7 PTCOMA:t8
    {:
    RESULT="\n\t"+"print("+"\""+a+"\""+","+b+")";
    listaTokens(b,"identificador",String.valueOf(bleft),String.valueOf(brigh
t));
    listaTokens(t1,"wPRINT",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"PAR_IZQ",String.valueOf(t2left),String.valueOf(t2right))
;
    listaTokens(t3,"COMD",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(t4,"DOSPT",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"COMD",String.valueOf(t5left),String.valueOf(t5right));
    listaTokens(t6,"MAS",String.valueOf(t6left),String.valueOf(t6right));
    listaTokens(t7,"PAR_DER",String.valueOf(t7left),String.valueOf(t7right))
;
    listaTokens(t8,"PTCOMA",String.valueOf(t8left),String.valueOf(t8right));
```

```
    :}
    | wPrint:t1 PAR_IZQ:t2 COMD:t3 CADENA:a COMD:t4 PAR_DER:t5
PTCOMA:t6
    {:
    RESULT="\n\t"+"print("+"\""+a+"\")";
    listaTokens(t1,"wPRINT",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"PAR_IZQ",String.valueOf(t2left),String.valueOf(t2right))
;
    listaTokens(t3,"COMD",String.valueOf(t3left),String.valueOf(t3right));
    listaTokens(t4,"COMD",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"PAR_DER",String.valueOf(t5left),String.valueOf(t5right))
;
    listaTokens(t6,"PTCOMA",String.valueOf(t6left),String.valueOf(t6right));
    :}
;

expresion  ::=
    MENOS:t1
expresion:a
    {:
    RESULT="-"+a;
    listaTokens(t1,"MENOS",String.valueOf(t1left),String.valueOf(t1right));
    :}%prec UMENOS
    |  expresion:a
MAS:t1        expresion:b
    {:
    RESULT=a+"+"+b;
    listaTokens(t1,"MAS",String.valueOf(t1left),String.valueOf(t1right));
    :}
    |  expresion:a
MENOS:t1      expresion:b
    {:
    RESULT=a+"-"+b;
    listaTokens(t1,"MENOS",String.valueOf(t1left),String.valueOf(t1right));
    :}
    |  expresion:a
POR:t1        expresion:b
    {:
    RESULT=a+"*"+b;
    listaTokens(t1,"POR",String.valueOf(t1left),String.valueOf(t1right));
    :}
    |  expresion:a
DIV:t1        expresion:b
    {:
    RESULT=a+"/"+b;
```

```
    listaTokens(t1,"DIV",String.valueOf(t1left),String.valueOf(t1right));
    :}
    |  ENTERO:a

    {:
    RESULT=a;
    listaTokens(a,"ENTERO",String.valueOf(aleft),String.valueOf(aright));
    :}
    |  DECIMAL:a

    {:
    RESULT=a;
    listaTokens(a,"DECIMAL",String.valueOf(aleft),String.valueOf(aright));
    :}
    |  PAR_IZQ:t1 expresion:a
PAR_DER:t2
    {:
    RESULT=a;
    listaTokens(t1,"PAR_IZQ",String.valueOf(t1left),String.valueOf(t1right))
;
    listaTokens(t2,"PAR_DER",String.valueOf(t2left),String.valueOf(t2right))
;
    :}
    |  DOLLAR:t1 LLI:t2 wNewValor:t3 COMA:t4 COMD:t5 CADENA:a COMD:t6
COMA:t7 COMD:t8 identificador:b COMD:t9 LLC:t10
    {:
    RESULT=cf.buscarJSON(String.valueOf(a),String.valueOf(b));
    listaTokens(t1,"DOLLAR",String.valueOf(t1left),String.valueOf(t1right));
    listaTokens(t2,"LLI",String.valueOf(t2left),String.valueOf(t2right));
    listaTokens(t3,"wNewValor",String.valueOf(t3left),String.valueOf(t3right
));
    listaTokens(t4,"COMA",String.valueOf(t4left),String.valueOf(t4right));
    listaTokens(t5,"COMD",String.valueOf(t5left),String.valueOf(t5right));
    listaTokens(t6,"COMD",String.valueOf(t6left),String.valueOf(t6right));
    listaTokens(t7,"COMA",String.valueOf(t7left),String.valueOf(t7right));
    listaTokens(t8,"COMD",String.valueOf(t8left),String.valueOf(t8right));
    listaTokens(t9,"COMD",String.valueOf(t9left),String.valueOf(t9right));
    listaTokens(t10,"LLC",String.valueOf(t10left),String.valueOf(t10right));
    listaTokens(b,"identificador",String.valueOf(bleft),String.valueOf(brigh
t));
    :}
;

CADENA ::=
```

```
    identificador:a

    {:
    RESULT=a;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    :}
    |
ENTERO:a
    {:
    RESULT=a;
    listaTokens(a,"ENTERO",String.valueOf(aleft),String.valueOf(aright));
    :}
    | ENTERO:a
CADENA:b
    {:
    RESULT=a+" "+b;
    listaTokens(a,"ENTERO",String.valueOf(aleft),String.valueOf(aright));
    :}
    | identificador:a
CADENA:b
    {:
    RESULT=a+" "+b;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    :}
    | identificador:a PT:t1
CADENA:b
    {:
    RESULT=a+"."+b;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"PT",String.valueOf(t1left),String.valueOf(t1right));
    :}
;




CONDICIONES ::=
    identificador:a RELACIONES:b
CADENA:c
    {:
    RESULT=a+" "+b+" "+c;
```

```
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    :}
    | identificador:a RELACIONES:b CADENA:c AND:t1
CONDICIONES:d
    {:
    RESULT=a+" "+b+" "+c+" and "+d;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"AND",String.valueOf(t1left),String.valueOf(t1right));
    :}
    | identificador:a RELACIONES:b CADENA:c OR:t1
CONDICIONES:d
    {:
    RESULT=a+" "+b+" "+c+" or "+d;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"OR",String.valueOf(t1left),String.valueOf(t1right));
    :}
    | NOT:t1
identificador:a
    {:
    RESULT="not "+a;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"NOT",String.valueOf(t1left),String.valueOf(t1right));
    :}
    | NOT:t1 identificador:a RELACIONES:b
CADENA:c
    {:
    RESULT="not "+a+" "+b+" "+c;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(t1,"NOT",String.valueOf(t1left),String.valueOf(t1right));
    :}
    | NOT:t1 identificador:a RELACIONES:b identificador:c
CONDICIONES:d
    {:
    RESULT="not "+a+" "+b+" "+c+" "+d;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(c,"identificador",String.valueOf(cleft),String.valueOf(crigh
t));
    listaTokens(t1,"NOT",String.valueOf(t1left),String.valueOf(t1right));
    :}
```

```
    | NOT:t1 identificador:a RELACIONES:b ENTERO:c
CONDICIONES:d
    {:
    RESULT="not "+a+" "+b+" "+c+" "+d;
    listaTokens(a,"identificador",String.valueOf(aleft),String.valueOf(arigh
t));
    listaTokens(c,"NOT",String.valueOf(cleft),String.valueOf(cright));
    listaTokens(t1,"NOT",String.valueOf(t1left),String.valueOf(t1right));
    :}
;


//FUNCIONES_ESTADISTICAS-----------------------------------------------------
-------------------------------------------------------------------


declaracionFUNCIONES::=
    wVoid wDefGlob PAR_IZQ PAR_DER LLI declaracionesG LLC
    | wVoid wGraphB PAR_IZQ PAR_DER LLI declaracionesGB LLC
    {:
        cf.graficaBarras(tituloGB,tituloGBX,tituloGBY,ejeXGB,valoresGB);
        tituloGB="";tituloGBX="";tituloGBY="";ejeXGB=null;valoresGB=null;
    :}
    | wVoid wGraphP PAR_IZQ PAR_DER LLI declaracionesGP LLC
    {:
        cf.graficoPie(tituloGP,valoresGP,ejeXGP);
        tituloGP="";valoresGP=null;ejeXGP=null;
    :}
    | wVoid identificador:a PAR_IZQ PAR_DER LLI declaraciones:b LLC
;

declaracionesG::=
    declaracionG:a declaracionesG:b
    | declaracionG:a
;

declaracionG::=
    declaracionINTG
    |declaracionDOUBLEG
    |declaracionCHARG
    |declaracionBOOLG
    |declaracionSTRINGG
;

declaracionINTG::=
```

```
    varINT identificador:a IGUAL expresion:b
PTCOMA                              {:guardarGlobales("int",String.valueOf(a)
,String.valueOf(b));:}
    | varINT identificador:a PTCOMA
;

declaracionDOUBLEG::=
    varDOUBLE identificador:a IGUAL expresion:b
PTCOMA                            {:guardarGlobales("double",String.valueOf(a)
,String.valueOf(b));:}
    | varDOUBLE identificador:a
PTCOMA
;

declaracionCHARG::=
    varCHAR identificador:a IGUAL COMS identificador:b COMS
PTCOMA              {:guardarGlobales("char",String.valueOf(a),String.valueO
f(b));:}
    | varCHAR identificador:a IGUAL COMD identificador:b COMD
PTCOMA            {:guardarGlobales("char",String.valueOf(a),String.valueOf(
b));:}
    | varCHAR identificador:a PTCOMA
;

declaracionBOOLG::=
    varBOOL identificador:a IGUAL identificador:b PTCOMA {:
    if(b.equals("true")){

    guardarGlobales("boolean",String.valueOf(a),"True");
    }else if(b.equals("false")){

    guardarGlobales("boolean",String.valueOf(a),"False");
    }
    :}
;

declaracionSTRINGG::=
    varSTRING identificador:a IGUAL COMS CADENA:b COMS
PTCOMA                    {:guardarGlobales("string",String.valueOf(a),String
.valueOf(b));:}
    | varSTRING identificador:a IGUAL COMD CADENA:b COMD
PTCOMA                  {:guardarGlobales("string",String.valueOf(a),String.v
alueOf(b));:}
    | varSTRING identificador:a PTCOMA
;
```

```
declaracionesGB::=
    declaracionGB:a declaracionesGB:b
    | declaracionGB:a
;

declaracionGB::=
    decTituloGB
    | decTituloGBX
    | decTituloGBY
    | decEjeX
    | decValores
;

declaracionesGP::=
    declaracionGP:a declaracionesGP:b
    | declaracionGP:a
;

declaracionGP::=
    decTituloGP
    | decEjeXP
    | decValoresP
;


decTituloGB::=
    varSTRING wTITULO IGUAL identificador:a
PTCOMA                              {:tituloGB=a;:}
    | varSTRING wTITULO IGUAL COMD identificador:a COMD
PTCOMA                  {:tituloGB=a;:}
;

decTituloGBX::=
    varSTRING wTITULOX IGUAL identificador:a
PTCOMA                              {:tituloGBX=a;:}
    | varSTRING wTITULOX IGUAL COMD identificador:a COMD
PTCOMA                 {:tituloGBX=a;:}
;

decTituloGBY::=
    varSTRING wTITULOY IGUAL identificador:a
PTCOMA                              {:tituloGBY=a;:}
```

```
    | varSTRING wTITULOY IGUAL COMD identificador:a COMD
PTCOMA                {:tituloGBY=a;:}
;

decEjeX::=
    varSTRING CORI CORD wEJEX IGUAL LLI chainX:a LLC
PTCOMA                    {:ejeXGB=String.valueOf(a).split(",");:}
;

chainX::=
    COMD CADENA:a COMD COMA
chainX:b                                          {:RESULT=a+","+b;:}
    | CADENA:a COMA
chainX:b                                                {:RESULT=buscarG
lobales((String) a)+","+b;:}
    |
CADENA:a                                                              {:
RESULT=buscarGlobales((String) a);:}
    | DOLLAR LLI wNewValor COMA COMD CADENA:a COMD COMA COMD identificador:b
COMD LLC

    {:RESULT=cf.buscarJSON(String.valueOf(a),String.valueOf(b));:}
    | COMA
chainX:a                                                          {:RESUL
T=a;:}
;

decValores::=
    varDOUBLE CORI CORD wVALORES IGUAL LLI chainX:a LLC
PTCOMA                {:valoresGB=String.valueOf(a).split(",");:}
;

decTituloGP::=
    varSTRING wTITULO IGUAL identificador:a
PTCOMA                            {:tituloGP=a;:}
    | varSTRING wTITULO IGUAL COMD identificador:a COMD
PTCOMA                {:tituloGP=a;:}
;

decEjeXP::=
    varSTRING CORI CORD wEJEX IGUAL LLI chainX:a LLC
PTCOMA                    {:ejeXGP=String.valueOf(a).split(",");:}
;

decValoresP::=
```

```
    varDOUBLE CORI CORD wVALORES IGUAL LLI chainX:a LLC
PTCOMA                  {:valoresGP=String.valueOf(a).split(",");:}
;
```