

Gramática

```
terminal String CONJUNTO;
terminal String PUNTO;
terminal String DISY;
terminal String MULT;
terminal String PLUS;
terminal String ITR;
terminal String TILDE;
terminal String COMA;
terminal String COM_SIMPLE;
terminal String COM_DOBLE;
terminal String DOS_PUNTOS;
terminal String PUNTO_COMA;
terminal String LLAV_ABIERTA;
terminal String LLAV_CERRADA;
terminal String ENTERO;
terminal String ASCII;
/* terminal String DECIMAL; */
terminal String LETRA;
terminal String IDENTIFICADOR;
terminal String OPERADOR;
terminal String SEPARADOR;

non terminal ini;
non terminal instrucciones;
non terminal instruccion;
non terminal String conjunto;
non terminal String identificador;
non terminal String ascii;
non terminal String Symbol;
non terminal String expresion;
non terminal String exp;
non terminal String conjCall;
non terminal String separadores;
non terminal String entradas;
non terminal String lectura;

start with ini;
```

```

ini ::= LLAV_ABIERTA instrucciones LLAV_CERRADA
;

instrucciones ::= instruccion instrucciones
| instruccion
;

instruccion ::=
    CONJUNTO DOS_PUNTOS identificador:a OPERADOR conjunto:b PUNTO_COMA {:
addConj(a,b); :}
    | expresion
    | separadores
    | error PUNTO_COMA
;

identificador ::=
    IDENTIFICADOR:a      {: RESULT=a; :}
    | LETRA:a             {: RESULT=a; :}
    | ENTERO:a            {: RESULT=a; :}
;

ascii ::=
    ASCII:a {: RESULT=a; :}
;

conjunto ::=
    ENTERO:a TILDE ENTERO:b      {: RESULT=a+"~"+b; :}
    | LETRA:a TILDE LETRA:b      {: RESULT=a+"~"+b; :}
    | LETRA:a COMA LETRA:b       {: RESULT=a+", "+b; :}
    | LETRA:a COMA LETRA:b conjunto:c  {: RESULT=a+", "+b+c; :}
    | ENTERO:a COMA ENTERO:b      {: RESULT=a+", "+b; :}
    | ENTERO:a COMA ENTERO:b conjunto:c  {: RESULT=a+", "+b+c; :}
    | COMA ENTERO:a conjunto:b     {: RESULT=", "+a+b; :}
    | COMA LETRA:a conjunto:b     {: RESULT=", "+a+b; :}
    | ascii:a TILDE ascii:b      {: RESULT=a+"~"+b; :}
    | ascii:a TILDE Symbol:b     {: RESULT=a+"~"+b; :}
    | Symbol:a TILDE ascii:b     {: RESULT=a+"~"+b; :}
    | ascii:a COMA ascii:b       {: RESULT=a+", "+b; :}
    | ascii:a COMA ascii:b conjunto:c  {: RESULT=a+", "+b+c; :}
    | Symbol:a COMA ascii:b      {: RESULT=a+", "+b; :}
    | Symbol:a COMA ascii:b conjunto:c  {: RESULT=a+", "+b+c; :}
    | ascii:a COMA Symbol:b      {: RESULT=a+", "+b; :}
    | ascii:a COMA Symbol:b conjunto:c  {: RESULT=a+", "+b+c; :}
    | ascii:a TILDE COM_DOBLE:b    {: RESULT=a+"~"+b; :}

```

```

| COM_DOBLE:a TILDE ascii:b          {: RESULT=a+"~"+b; :}
| ascii:a TILDE COM_SIMPLE:b          {: RESULT=a+"~"+b; :}
| COM_SIMPLE:a TILDE ascii:b          {: RESULT=a+"~"+b; :}
| ascii:a TILDE COMA:b                {: RESULT=a+"~"+b; :}
| COMA:a TILDE ascii:b                {: RESULT=a+"~"+b; :}
| ascii:a TILDE DOS_PUNTOS:b          {: RESULT=a+"~"+b; :}
| DOS_PUNTOS:a TILDE ascii:b          {: RESULT=a+"~"+b; :}
| ascii:a TILDE LLAV_ABIERTA:b        {: RESULT=a+"~"+b; :}
| LLAV_ABIERTA:a TILDE ascii:b        {: RESULT=a+"~"+b; :}
| ascii:a TILDE LLAV_CERRADA:b        {: RESULT=a+"~"+b; :}
| LLAV_CERRADA:a TILDE ascii:b        {: RESULT=a+"~"+b; :}
| COMA COM_DOBLE:a conjunto:b         {: RESULT=", "+a+b; :}
| COMA COM_SIMPLE:a conjunto:b        {: RESULT=", "+a+b; :}
| COMA DOS_PUNTOS:a conjunto:b        {: RESULT=", "+a+b; :}
| COMA PUNTO_COMA:a conjunto:b        {: RESULT=", "+a+b; :}
| COMA LLAV_ABIERTA:a conjunto:b      {: RESULT=", "+a+b; :}
| COMA LLAV_CERRADA:a conjunto:b      {: RESULT=", "+a+b; :}
| COMA ascii:a                       {: RESULT=", "+a; :}
| COMA Symbol:a                      {: RESULT=", "+a; :}
| COMA LETRA:a                       {: RESULT=", "+a; :}
| COMA ENTERO:a                      {: RESULT=", "+a; :}
| COMA COM_DOBLE:a                   {: RESULT=", "+a; :}
| COMA COM_SIMPLE:a                  {: RESULT=", "+a; :}
| COMA DOS_PUNTOS:a                  {: RESULT=", "+a; :}
| COMA PUNTO_COMA:a                  {: RESULT=", "+a; :}
| COMA LLAV_ABIERTA:a                {: RESULT=", "+a; :}
| COMA LLAV_CERRADA:a                {: RESULT=", "+a; :}
| ascii:a COMA COM_SIMPLE:b          {: RESULT=a + ", "+b; :}
| ascii:a COMA COM_DOBLE:b           {: RESULT=a + ", "+b; :}
| ascii:a COMA DOS_PUNTOS:b          {: RESULT=a + ", "+b; :}
| ascii:a COMA PUNTO_COMA:b          {: RESULT=a + ", "+b; :}
| ascii:a COMA LLAV_ABIERTA:b        {: RESULT=a + ", "+b; :}
| ascii:a COMA LLAV_CERRADA:b        {: RESULT=a + ", "+b; :}
;

expression ::=
    identificador:a OPERADOR exp:b PUNTO_COMA {: br.analizarER(b,a); :}
;

exp ::=
    Symbol:a conjCall:b              {: RESULT= a+b; :}
    | Symbol:a conjCall:b exp:c      {: RESULT=a+b+c; :}
    | Symbol:a exp:b                  {: RESULT=a+b; :}
;

```

[illegible]

```

    | COM_SIMPLE COM_SIMPLE:a COM_SIMPLE conjCall:b           {:
RESULT="\ "'+a+"\ "'+b; :}
    | COM_DOBLE COMA:a COM_DOBLE                               {:
RESULT="\ "'+a+"\ "'; :}
    | COM_DOBLE COMA:a COM_DOBLE conjCall:b                   {:
RESULT="\ "'+a+"\ "'+b; :}
    | COM_SIMPLE COMA:a COM_SIMPLE                             {:
RESULT="\ "'+a+"\ "'; :}
    | COM_SIMPLE COMA:a COM_SIMPLE conjCall:b                 {:
RESULT="\ "'+a+"\ "'+b; :}
    | COM_DOBLE DOS_PUNTOS:a COM_DOBLE                        {:
RESULT="\ "'+a+"\ "'; :}
    | COM_DOBLE DOS_PUNTOS:a COM_DOBLE conjCall:b             {:
RESULT="\ "'+a+"\ "'+b; :}
    | COM_SIMPLE DOS_PUNTOS:a COM_SIMPLE                       {:
RESULT="\ "'+a+"\ "'; :}
    | COM_SIMPLE DOS_PUNTOS:a COM_SIMPLE conjCall:b           {:
RESULT="\ "'+a+"\ "'+b; :}
    | COM_DOBLE PUNTO_COMA:a COM_DOBLE                        {:
RESULT="\ "'+a+"\ "'; :}
    | COM_DOBLE PUNTO_COMA:a COM_DOBLE conjCall:b             {:
RESULT="\ "'+a+"\ "'+b; :}
    | COM_SIMPLE PUNTO_COMA:a COM_SIMPLE                       {:
RESULT="\ "'+a+"\ "'; :}
    | COM_SIMPLE PUNTO_COMA:a COM_SIMPLE conjCall:b           {:
RESULT="\ "'+a+"\ "'+b; :}
    | COM_DOBLE LLAV_ABIERTA:a COM_DOBLE                      {:
RESULT="\ "'+a+"\ "'; :}
    | COM_DOBLE LLAV_ABIERTA:a COM_DOBLE conjCall:b           {:
RESULT="\ "'+a+"\ "'+b; :}
    | COM_SIMPLE LLAV_ABIERTA:a COM_SIMPLE                     {:
RESULT="\ "'+a+"\ "'; :}
    | COM_SIMPLE LLAV_ABIERTA:a COM_SIMPLE conjCall:b         {:
RESULT="\ "'+a+"\ "'+b; :}
    | COM_DOBLE LLAV_CERRADA:a COM_DOBLE                      {:
RESULT="\ "'+a+"\ "'; :}
    | COM_DOBLE LLAV_CERRADA:a COM_DOBLE conjCall:b           {:
RESULT="\ "'+a+"\ "'+b; :}
    | COM_SIMPLE LLAV_CERRADA:a COM_SIMPLE                     {:
RESULT="\ "'+a+"\ "'; :}
    | COM_SIMPLE LLAV_CERRADA:a COM_SIMPLE conjCall:b         {:
RESULT="\ "'+a+"\ "'+b; :}
;

```

```

Symbol ::=
    PUNTO:a                {: RESULT=a; :}
    | DISY:a                {: RESULT=a; :}
    | MULT:a                {: RESULT=a; :}
    | PLUS:a                {: RESULT=a; :}
    | ITR:a                 {: RESULT=a; :}
;

lectura ::= '
    identificador:a {: RESULT=a; :}
    | Symbol:a {: RESULT=a; :}
    | ascii:a {: RESULT=a; :}
    | identificador:a lectura:b {: RESULT=a+b; :}
    | Symbol:a lectura:b {: RESULT=a+b; :}
    | ascii:a lectura:b {: RESULT=a+b; :}
;

entradas ::=
    identificador:a DOS_PUNTOS COM_DOBLE lectura:b COM_DOBLE PUNTO_COMA {:
analizarCadena(a,b); :}
;

separadores ::=
    SEPARADOR
    | entradas
;

```