

# Manual técnico Proyecto final

Omar Eduardo Roa Quintero  
Fernando Vargas Montero

Universidad Nacional de Colombia  
Facultad de Ingeniería, Departamento de Ingeniería de Sistemas e Industrial  
Bogotá, Colombia  
2017

# Índice

<b>1. Definición de la gramática de Diunisio</b>	<b>3</b>
<b>2. ANTLR</b>	<b>3</b>
2.1. Tokens del lenguaje . . . . .	4
2.2. Gramática (en notación E-BNF, Extended Backus–Naur Form) . . . . .	4
2.3. Clases de Python . . . . .	5
2.4. Diagramas de sintaxis . . . . .	6
<b>3. Recomendaciones y Sugerencias</b>	<b>9</b>

## 1. Definición de la gramática de Diunisio

$$\begin{aligned}
\langle \text{expresion} \rangle & \models \text{identificador } \langle = \rangle \langle \text{expMAT} \rangle \mid \langle \text{expMAT} \rangle \\
& \mid \langle \text{IMPRIMIR} \rangle ( \langle \text{expresion} \rangle ) \\
& \mid \langle \text{SALTOLINEA} \rangle \\
& \mid \langle \text{problema} \rangle \\
& \mid \langle \text{COMENTARIO} \rangle \\
\langle \text{problema} \rangle & \models \langle \text{funcionTrans} \rangle \langle \text{restriccion} \rangle \langle \text{restriccionVariable} \rangle \langle \text{RESOLVER} \rangle \langle \text{SALTOLINEA} \rangle \\
\langle \text{restriccionVariable} \rangle & \models \langle \text{CON} \rangle \langle : \rangle \langle \text{SALTOLINEA} \rangle ( \langle \text{desigualdadVariable} \rangle \langle \text{SALTOLINEA} \rangle ) \\
\langle \text{desigualdadVariable} \rangle & \models \text{variable } \langle \text{op} \rangle \langle \text{expMAT} \rangle \\
& \mid \langle \text{variable} \rangle \langle \text{mayorIgual} \rangle \langle \text{expMAT} \rangle \\
& \mid \langle \text{variable} \rangle \langle \text{menorIgual} \rangle \langle \text{expMAT} \rangle \\
\langle \text{restriccion} \rangle & \models \langle \text{RESTRICCIONES} \rangle \langle : \rangle \langle \text{SALTOLINEA} \rangle ( \langle \text{desigualdad} \rangle \langle \text{SALTOLINEA} \rangle ) \\
\langle \text{desigualdad} \rangle & \models \text{polinomio } \langle \text{op} \rangle \langle \text{expMAT} \rangle \\
& \mid \langle \text{polinomio} \rangle \langle \text{mayorIgual} \rangle \langle \text{expMAT} \rangle \\
& \mid \langle \text{polinomio} \rangle \langle \text{menorIgual} \rangle \langle \text{expMAT} \rangle \\
\langle \text{op} \rangle & \models < \mid > \mid = \\
\langle \text{funcionTrans} \rangle & \models \langle \text{op2} \rangle \langle : \rangle \langle \text{SALTOLINEA} \rangle \langle \text{funcion} \rangle \langle \text{SALTOLINEA} \rangle \\
\langle \text{op2} \rangle & \models \text{max} \mid \text{min} \\
\langle \text{funcion} \rangle & \models \langle \text{funcionDef} \rangle \langle = \rangle \langle \text{polinomio} \rangle \\
\langle \text{funcionDef} \rangle & \models \langle \text{FUNCION} \rangle ( \langle \text{variable} \rangle ( \langle , \text{variable} \rangle ) ) \\
\langle \text{polinomio} \rangle & \models - \langle \text{monomio} \rangle \langle \text{monomioAdd} \rangle \\
& \mid - \langle \text{monomio} \rangle \\
\langle \text{monomioAdd} \rangle & \models + \langle \text{monomio} \rangle \\
& \mid - \langle \text{monomio} \rangle \\
\langle \text{monomio} \rangle & \models \langle \text{numero} \rangle \langle \text{variable} \rangle \\
& \mid \langle \text{expMAT} \rangle \\
\langle \text{expMAT} \rangle & \models \langle \text{expMAT} \rangle \langle \text{op3} \rangle \langle \text{expMAT} \rangle \\
& \mid ( \langle \text{expMAT} \rangle ) \\
& \mid \langle \text{entero} \rangle \\
& \mid \langle \text{decimal} \rangle \\
& \mid \langle \text{identificador} \rangle \\
& \mid - \langle \text{expMAT} \rangle \\
\langle \text{op3} \rangle & \models ** \mid * \mid / \mid + \mid - \\
\langle \text{variable} \rangle & \models \langle \text{LETRA} \rangle \\
\langle \text{identificador} \rangle & \models \langle \text{LETRA} \rangle ( \langle \text{LETRA} \rangle \mid \langle \text{DIGITO} \rangle \mid \_ ) \\
\langle \text{numero} \rangle & \models ( \langle \text{expMAT} \rangle ) \\
& \mid \langle \text{entero} \rangle \\
& \mid \langle \text{decimal} \rangle \\
& \mid \\
\langle \text{trigonometrica} \rangle & \models \text{op4} ( \langle \text{expMAT} \rangle ) \\
\langle \text{op4} \rangle & \models \text{SENO} \mid \text{COSENO} \mid \text{TANGENTE} \mid \text{EULER} \mid \text{LOGARITMO} \\
\langle \text{decimal} \rangle & \models \langle \text{entero} \rangle . \langle \text{entero} \rangle \\
& \mid \langle \text{decimal} \rangle \langle \text{exponencial} \rangle \\
& \mid \langle \text{entero} \rangle \langle \text{exponencial} \rangle \\
\langle \text{exponencial} \rangle & \models e \langle \text{entero} \rangle \\
& \quad e - \langle \text{entero} \rangle \\
& \quad e + \langle \text{entero} \rangle \\
\langle \text{entero} \rangle & \models \langle \text{DIGITO} \rangle \\
\langle \text{LETRA} \rangle & \models \text{a} \dots \text{z} \langle \text{LETRA} \rangle \mid \text{A} \dots \text{Z} \langle \text{letras} \rangle \mid \text{A} \dots \text{Z} \mid \text{a} \dots \text{z} \\
\langle \text{DIGITO} \rangle & \models 0 \dots 9 \langle \text{DIGITO} \rangle \mid 0 \dots 9 \\
\langle \text{exp} \rangle & \models \text{e} \mid \text{E}
\end{aligned}$$

## 2. ANTLR

El lenguaje fue desarrollado en ANTLR[2], definiendo el léxico y la gramática en un archivo propio de ANTLR (.g4). Luego se generó el reconocedor, que creaba todas las clases bases en Python. Se

sobrescribió el simplexExpand.py donde se definía la semántica del lenguaje. Finalmente se creó una clase simplex.py que es la que permite ejecutar algoritmos dentro del lenguaje.

## 2.1. Tokens del lenguaje

```
COMENTARIO : ( '#' ~[\r\n]* | '##' .*? '##') -> skip;
mayorIgual: MAYOR IGUAL;
menorIgual: MENOR IGUAL;
MAS : '+';
MENOS : '-';
MULT : '*';
DIV : '/';
POTENCIA : '**';
PUNTO : '.';
IGUAL : '=';
LETRA : [a-zA-Z];
COMA : ',';
SALTOLINEA : '\r'? '\n';
BLANCO : [ \t]+ -> skip ;
DIGITO : [0-9];
PARABIERTO : '(';
PARCERRADO : ')';
MAYOR: '>';
MENOR: '<';
DOSPUNTOS: ':';
EXPONENCIAL: [Ee];
IMPRIMIR : 'imprimir' | 'imp' ;
MAXI : 'maximizar' | 'max';
MINI : 'minimizar' | 'min';
FUNCION: 'funcion'|'fun';
RESTRICCIONES: 'restringir' | 'restr';
RESOLVER: 'resolver' | 'resol' | 'res' ;
CON: 'con';
SENO: 'sen';
COSENO: 'cos';
TANGENTE: 'tan';
EULER: 'exp';
LOGARITMO : 'ln';
```

## 2.2. Gramática (en notación E-BNF, Extended Backus–Naur Form)

```
iniciar : programa

programa : expresion* EOF;

expresion : identificador IGUAL expMAT
          | expMAT
          | IMPRIMIR PARABIERTO expresion PARCERRADO
          | SALTOLINEA
          | problema
          | COMENTARIO
          ;

problema: funcionTrans restriccion restriccionVariable RESOLVER SALTOLINEA ;

restriccionVariable : CON DOSPUNTOS SALTOLINEA (desigualdadVariable SALTOLINEA)+ ;

desigualdadVariable : variable tipo=(MENOR | MAYOR | IGUAL) expMAT
                    | variable mayorIgual expMAT
                    | variable menorIgual expMAT
                    ;

restriccion : RESTRICCIONES DOSPUNTOS SALTOLINEA (desigualdad SALTOLINEA)+;

desigualdad: polinomio tipo=(MENOR | MAYOR | IGUAL) expMAT
            | polinomio mayorIgual expMAT
            | polinomio menorIgual expMAT
            ;

funcionTrans: operacion = (MAXI | MINI) DOSPUNTOS SALTOLINEA funcion SALTOLINEA ;
```

```

funcion : funcionDef IGUAL polinomio ;

funcionDef: FUNCION PARABIERTO variable (COMA variable)* PARCERRADO
        ;

polinomio : MENOS? monomio monomioAdd+
        | MENOS? monomio
        ;

monomioAdd: MAS monomio
        | MENOS monomio
        ;

monomio : e = numero variable
        | expMAT
        ;

expMAT: expMAT POTENCIA expMAT
        | expMAT operador = (MULT | DIV) expMAT
        | expMAT operador = (MAS | MENOS) expMAT
        | PARABIERTO expMAT PARCERRADO
        | entero
        | decimal
        | identificador
        | MENOS expMAT
        ;

variable : LETRA+ ;

identificador : LETRA (LETRA | DIGITO | '_' )*;

numero : PARABIERTO expMAT PARCERRADO
        | decimal
        | entero
        |
        ;

trigonometrica : SENO PARABIERTO expMAT PARCERRADO
        | COSENO PARABIERTO expMAT PARCERRADO
        | TANGENTE PARABIERTO expMAT PARCERRADO
        | EULER PARABIERTO expMAT PARCERRADO
        | LOGARITMO PARABIERTO expMAT PARCERRADO
        ;

decimal : entero PUNTO entero
        | decimal exponencial
        | entero exponencial
        ;

exponencial : EXPONENCIAL entero
        | EXPONENCIAL MENOS entero
        | EXPONENCIAL MAS entero
        ;

entero : DIGITO+;

```

## 2.3. Clases de Python

- simplexExpand.py en donde se sobrescriben los métodos de simplexVisitor.py previamente generado a partir de la gramática en ANTLR.
- simplex.py donde se ejecuta el código de simplex.

2.4. Diagramas de sintaxis

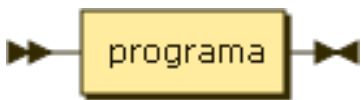


Figura 1: Iniciar

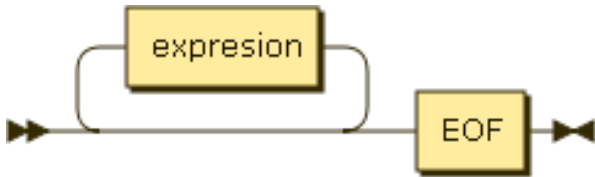


Figura 2: Programa

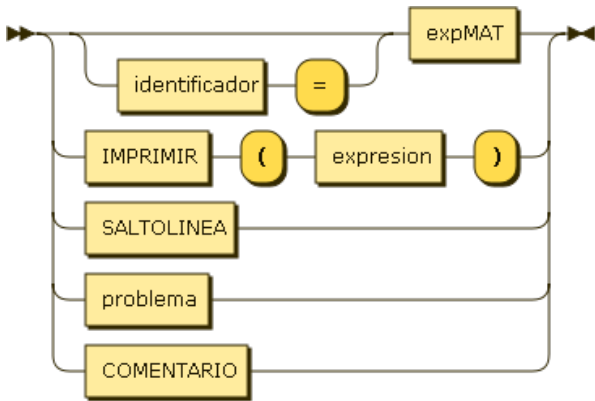


Figura 3: Expresión



Figura 4: Problema

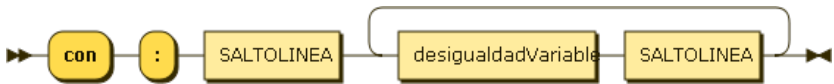


Figura 5: restriccionVariable

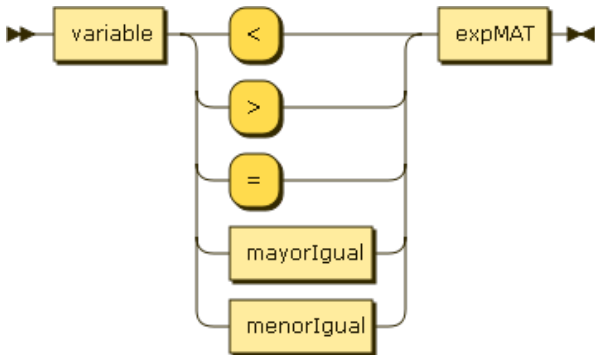


Figura 6: desigualdadVariable

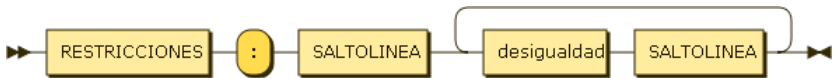


Figura 7: restriccion

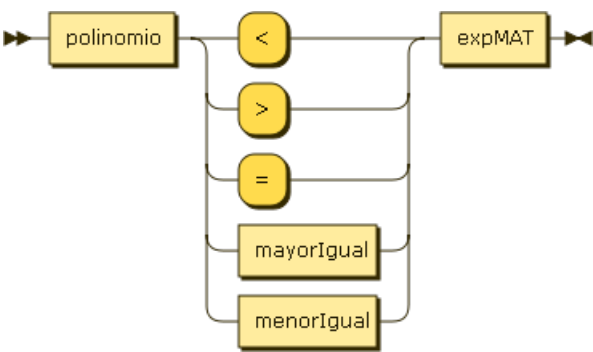


Figura 8: desigualdad

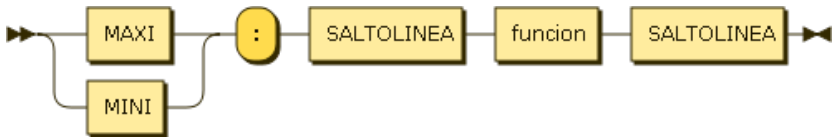


Figura 9: funcionTrans

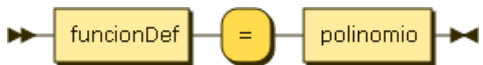


Figura 10: funcion

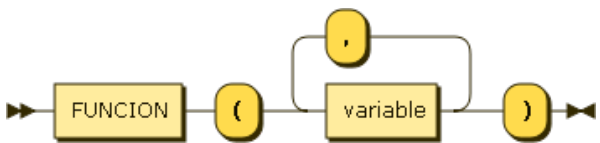


Figura 11: funcionDef

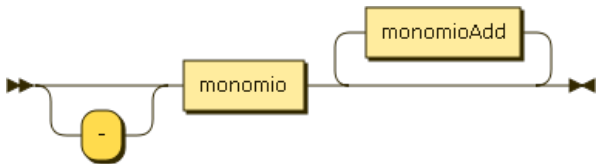


Figura 12: polinomio

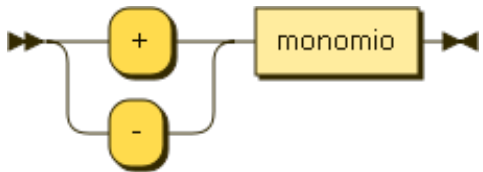


Figura 13: monomioAdd

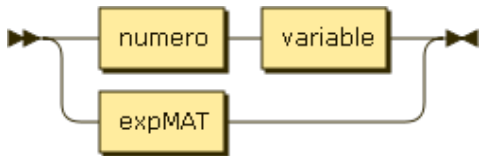


Figura 14: monomio

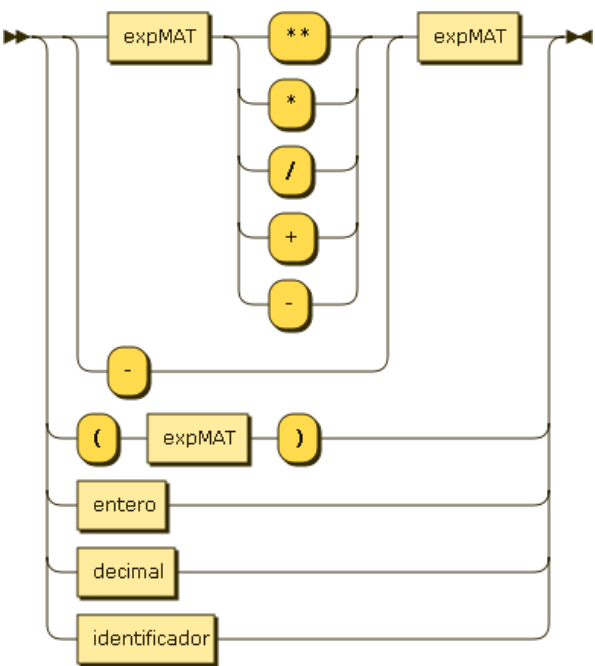


Figura 15: expMAT



Figura 16: variable

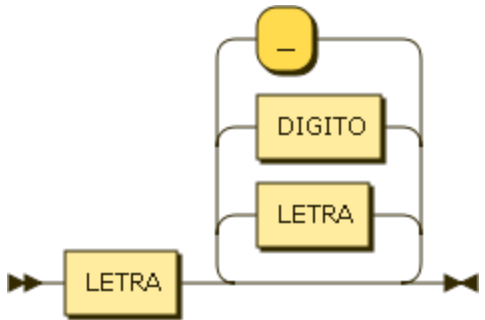


Figura 17: identificador

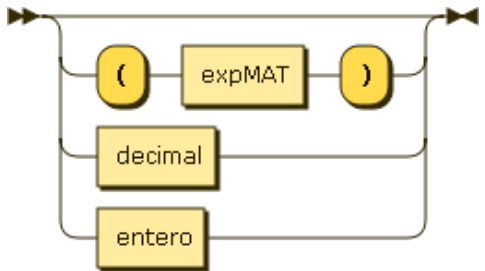


Figura 18: numero

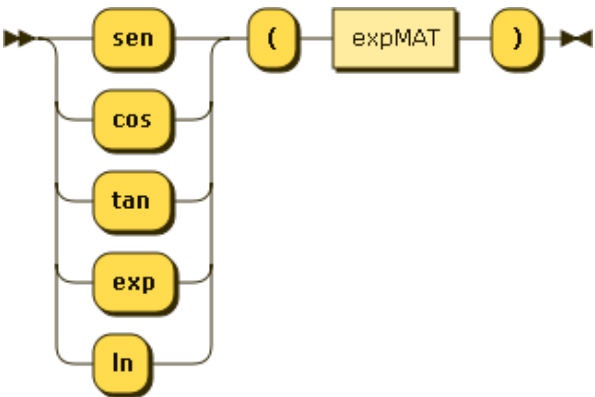


Figura 19: trigonometrica



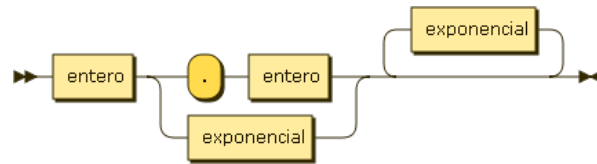


Figura 20: decimal

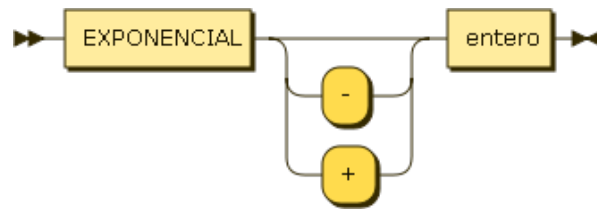


Figura 21: exponencial

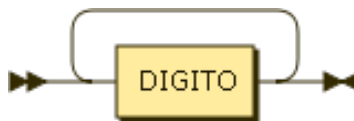


Figura 22: entero

### 3. Recomendaciones y Sugerencias

Para generar un lenguaje con ANTLR es necesario diseñar desde el comienzo bien el lenguaje antes de implementarlo y desde un comienzo se debe definir una gramática correctamente para que en su implementación no se generen modificaciones ni alteraciones. ANTLR cuenta con muchas herramientas que nos ayudara a generar grandes lenguajes complejos, por ello se debe estudiar. Es importante hacer pruebas con la gramática inicial para reconocer falencias tempranamente y corregirlas.

### Referencias

- [1] Generación de los diagramas E-BNF: <http://www.bottlecaps.de/rr/ui>
- [2] ANTLR (Versión 4.7) {software}. (2017). Obtenido de: <http://www.antlr.org/>