



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE
SISTEMAS CIENCIAS DE LA
COMPUTACION
COMPILADORES Y LENGUAJES

Informe - Prueba 4

GRUPO: GR1COM

FECHA DE ENTREGA: 12/03/2021

INTEGRANTES: Alejandro Moya, Leonardo Andrade, Jefferson Cando

Usamos LEX y YACC para implementar una pequeña calculadora que nos permita realizar las operaciones aritméticas básicas como suma, resta, multiplicación, división, potenciación, raíz y también funciones trigonométricas como el seno y el coseno, también poder realizar logaritmos naturales y e elevado a la x . Los operandos pueden ser de tipo entero o real aplicando una Gramática de Contexto Libre Ambigua y reglas de precedencia y asociatividad.

CÓDIGO LEX

```
%{
#include <stdio.h>
#include <stdlib.h>
#include "y.tab.h"
int nlines = 0;
%}
DIGITO      [0-9]
ID          [a-zA-Z][a-zA-Z0-9]*
%%

{DIGITO}+("."{DIGITO})?      {yylval.real=atof(yytext); return (TKN_NUM);}
"="                          {return (TKN_ASSIGN);}
";"                          {return (TKN_PTOCOMA);}
"*"                          {return (TKN_MULT);}
"/"                          {return (TKN_DIV);}
```

"+"	{return (TKN_MAS);}
"_"	{return (TKN_MENOS);}
"^"	{return (TKN_ELEV);}
"sqrt"	{return (TKN_RAIZ);}
"("	{return (TKN_PAA);}
")"	{return (TKN_PAC);}
"cos"	{return (TKN_COS);}
"sen"	{return (TKN_SEN);}
"ln"	{return (TKN_LOGNAT);}
"e"	{return (TKN_E);}
{ID}	{return (TKN_ID);}
"\n"	{nlines++;}
.	
%%	

CÓDIGO YACC

```
%{
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
extern int yylex(void);
extern char *yytext;
extern int nlines;
extern FILE *yyin;
void yyerror(char *s);
%}
%union
{
    float real;
}
%start Calculadora
%token <real> TKN_NUM
%token TKN_ASIGN
%token TKN_PTOCOMA
%token TKN_MULT
%token TKN_DIV
%token TKN_MAS
%token TKN_MENOS
%token TKN_ELEV
%token TKN_RAIZ
%token TKN_LOGNAT
%token TKN_E
%token TKN_PAA
%token TKN_PAC
%token TKN_COS
%token TKN_SEN
%token <real> TKN_ID
%type Calculadora
```

```

%type <real> Expresion
%left TKN_MAS TKN_MENOS
%left TKN_MULT TKN_DIV
%right TKN_ELEV TKN_RAIZ
%%

Calculadora:  TKN_ID {printf("El valor de %s es: ", yytext);}
               TKN_ASSIGN Expresion TKN_PTOCOMA {printf("%5.2f\n", $4);};
Expresion:    TKN_NUM {$$=$1;} |
               Expresion TKN_MAS Expresion    {$$=$1+$3;} |
               Expresion TKN_MENOS Expresion {$$=$1-$3;} |
               Expresion TKN_MULT Expresion   {$$=$1*$3;} |
               Expresion TKN_DIV Expresion     {$$=$1/$3;} |
               Expresion TKN_ELEV TKN_PAA Expresion TKN_PAC  {$$=pow($1,$4);} |
               TKN_RAIZ TKN_PAA Expresion TKN_PAC    {$$=sqrt($3);} |
               TKN_PAA Expresion TKN_PAC    {$$=$2;} |
               TKN_COS TKN_PAA Expresion TKN_PAC      {$$=cos($3);} |
               TKN_SEN TKN_PAA Expresion TKN_PAC      {$$=sin($3);} |
               TKN_LOGNAT TKN_PAA Expresion TKN_PAC {$$=log($3);} |
               TKN_E TKN_ELEV TKN_PAA Expresion TKN_PAC  {$$=exp($4);} ;
%%

void yyerror(char *s){
    printf("Error %s", s);
}

int main (int argc, char **argv){
    if (argc>1){
        yyin=fopen(argv[1], "rt");
    }else{
        yyin=stdin;
    }

    yyparse();
    printf("FIN del Analisis. Entrada CORRECTA.\n");
    printf("Numero de lineas analizadas: %d\n", nlines);
    return 0;
}

```

Entrada	Salida
$a = \sqrt{8 - 2^3/4 * 5 + 7} + \cos(3) + \ln(1)$;	El valor de a es: 2.55 FIN del Analisis. Entrada CORRECTA. Numero de lineas analizadas: 2