

# 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, multiplicacion, divicion, potenciacion, raiz y tambien funciones trigonometricas como el seno y el coseno, tambien poder realizar logaritmos naturales y e elevado a la x . Los operandores 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\_ASIGN);}  ";" {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\_ASIGN 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(8-2³/4\*5+7)+cos(3)+ln(1); | El valor de a es: 2.55  FIN del Analisis. Entrada CORRECTA.  Numero de lineas analizadas: 2 |