

### Planteamiento:

Desarrollar un analizador de lenguaje que tome como entrada un archivo de texto, el cual será analizado en su léxico y en su sintaxis. El texto de entrada contará con las instrucciones cuyos formatos son los siguientes:

**programa** nombre;  
**iniciar**  
listado-de-instrucciones;  
**terminar.**

1. El nombre de programa estará compuesto sólo con una letra del abecedario, seguida de 0 o más letras y/o dígitos del 0 al 9.
2. Cada línea debe estar alineada a la izquierda y debe tener como máximo un espacio entre cada palabra.
3. Cada línea debe estar separada una de la otra por un salto.
4. El listado de instrucciones puede constar de una sola o de varias separadas por un símbolo de “;” de la siguiente manera:

Instrucción1;	Donde la	<b>identificador:=</b> expresión-aritmética;
Instrucción2;	instrucción	<b>leer</b> identificador;
Instrucción3;	debe ser,	<b>imprimir</b> identificador;
...	cualquiera de	
InstrucciónN;	las 3	
	siguientes:	

5. El identificador deberá iniciar con una letra seguido de (0 o más) letras y/o dígitos.
6. Las expresiones aritméticas serán las siguientes: suma, multiplicación, resta, división y potencia. Serán válidos los paréntesis.

De la siguiente forma:

**expresión-aritmética**  $\rightarrow$  expresión-aritmética + expresión-aritmética | expresión-aritmética \* expresión-aritmética | expresión-aritmética – expresión-aritmética | expresión-aritmética / expresión-aritmética | expresión-aritmética ^expresión-aritmética | (expresión) | constante

*La constante es un número entero, compuesto por dígitos del 0 al 9, con o sin signo.*

7. Las letras sólo serán minúsculas.
8. El resultado del analizador deberá decir si hubo error en el léxico o en la sintaxis de acuerdo a lo analizado. En caso de no tener error, deberá decir que no hubo errores.

Un ejemplo válido es:

```
programa final12;  
iniciar  
leer i;  
leer j;  
leer k;  
p := i+j*2;  
q := (p*k)/50;  
imprimir q;  
terminar.
```