



### PROGRAMA N°3

Profesora : Consuelo Ramírez  
Fecha : 14 de junio  
Puntaje : 57 puntos  
PREMA : 60%

#### Objetivo

Implementar un analizador sintáctico, utilizando un generador de analizadores sintácticos.

#### Requerimientos

Escriba un programa en Bison para sistema operativo Windows que implemente un analizador sintáctico para la calculadora científica descrita.

Su programa debe:

1. permitir ingresar un archivo de entrada como parámetro en la línea de comandos.
2. indicar por pantalla el número de línea del primer error sintáctico detectado en dicho archivo.

Modifique su Programa N°2 para que funcione en conjunto con este analizador sintáctico.

La Figura 1 representa la forma de ejecución del programa.

```
C:\> sintactico.exe archivo
```

```
Análisis sintáctico exitoso.
```

```
C:\>
```

**Figura 1.** Ejecución.



Las Figuras 2, 3, 4 y 5 representan los tipos de errores que el programa debe detectar.

```
C:\>sintactico.exe
Error: Falta parámetro.
Uso: sintactico.exe archivo

C:\>
```

**Figura 2.** Error 1.

```
C:\> sintactico.exe archivo_1 archivo_2 archivo_3 ...
Error: Demasiados parámetros.
Uso: sintactico.exe archivo

C:\>
```

**Figura 3.** Error 2.

```
C:\> sintactico.exe archivo
Error: El archivo no existe.

C:\>
```

**Figura 4.** Error 3.

```
C:\> sintactico.exe archivo
Error en la línea número: número_línea

C:\>
```

**Figura 5.** Error 4.

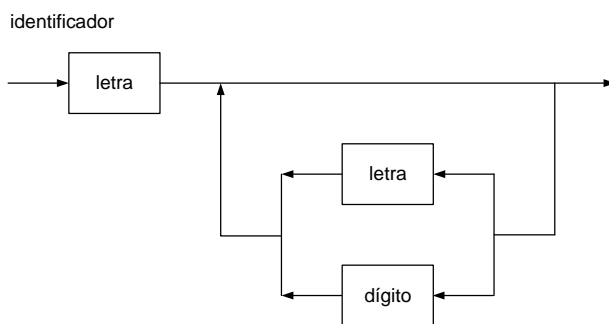
## Entrega

1. Comprima los archivos fuentes en un archivo denominado con su primer nombre seguido de un espacio y de su primer apellido.
2. Suba el archivo resultante a Campus Virtual antes del 28 de junio a las 8:15 hrs.

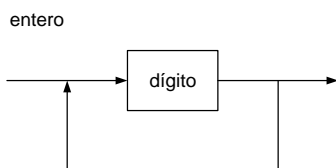


## Componentes Léxicos

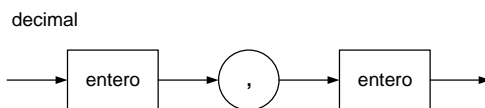
Las Figuras 6, 7, 8 y 9 muestran la estructura de los identificadores, enteros, reales en notación decimal y reales en notación exponencial respectivamente.



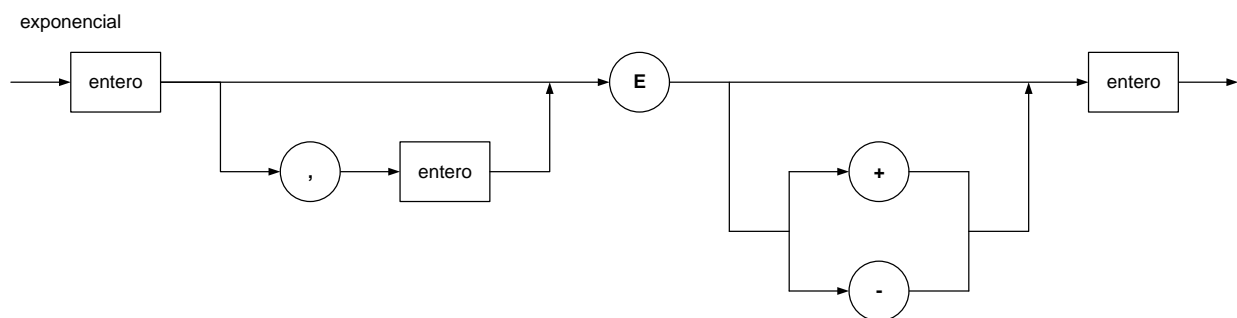
**Figura 6.** Identificador.



**Figura 7.** Entero.



**Figura 8.** Real en notación decimal.



**Figura 9.** Real en notación exponencial.



#### Operaciones:

- + (adición)
- - (sustracción)
- x (multiplicación)
- : (división)
- / (división entera)
- ^ (potencia)
- | (valor absoluto)
- ! (factorial)
- ( , ) (paréntesis)
- = (asignación)
- MOD (resto entero de la división entera)

Para los operadores x y MOD, cuando se usa un identificador como operando, se requiere de un espacio entre el identificador y el operador.

#### Funciones:

- SQR  
Retorna la raíz cuadrada del argumento.  
El argumento debe ser mayor o igual que 0.
- CUR  
Retorna la raíz cúbica del argumento.  
El argumento debe ser mayor o igual que 0.
- EXP  
Retorna e elevado al argumento.
- LN  
Retorna el logaritmo natural del argumento.  
El argumento debe ser mayor que 0.
- LOG  
Retorna el logaritmo decimal del argumento.  
El argumento debe ser mayor que 0.
- SGN  
Retorna un 1 cuando el argumento es mayor que 0, un -1 cuando es menor que 0, y 0 cuando es 0.
- INT  
Retorna el entero mayor que no sobrepasa el valor del argumento.
- FIX  
Retorna la parte entera del argumento.
- FRAC  
Retorna la parte fraccional del argumento.



Constante:

$\pi = 3,14159$

Precedencia:

1. ( , )
2. Funciones
3. ^
4. Signos (+ , -)
5. x , : , / , MOD
6. + , -