

# Trabajo práctico N° 4

## *SSL Lenguajes*

Nombre y Apellido	Legajo
Stefano Alejandro Gassmann	208.380-2
Santiago Rodriguez	209.171-9

### ***Índice:***

- **Resolución**
  - *Planteo inicial*
  - *Funcionamiento del programa*
  - *Ejemplo 1*
  - *Ejemplo 2*
  - *Ejemplo 3*

# RESOLUCIÓN DEL PROBLEMA

## **Planteo inicial:**

Comenzamos este proyecto a partir del código del compilador proporcionado en la cursada, utilizando en este caso el editor de código Visual Studio Code.

Leer y comprender el código resultó ser la parte más robusta debido al ida y vuelta entre las diversas funciones y estructuras utilizadas en el análisis por las diferentes partes del compilador. Hizo más óptimo y eficiente empezar estudiando el analizador léxico y su uso del scanner, siguiendo luego por las funciones de análisis sintáctico, rutinas semánticas, etc.

Una vez hecho esto, procedimos a probar y verificar el correcto funcionamiento del compilador.

## **Funcionamiento del programa:**

Al ejecutar el programa, introducimos el archivo con el que queremos trabajar, el mismo contiene instrucciones pertenecientes al lenguaje de programación Micro.

```
1  inicio
2  leer(a,b);
3  fin
```

Continuamos, y el programa responde de manera correcta.

```
PS C:\Users\Santi\SSL> ./tp4 archi.m
Declara a,Entera,
Read a,Entera,
Declara b,Entera,
Read b,Entera,
Detiene ,,
PS C:\Users\Santi\SSL> █
```

Finalmente, el programa se detiene y queda a la espera del siguiente archivo.

### **Ejemplo 1 (sin errores):**

Como primer ejemplo tenemos a las siguientes líneas de código:

```
1  inicio
2  leer(a,b);
3  d:=a+(b+2);
4  escribir(a,d);
5  fin
```

Nuevamente, le pasamos el archivo al programa y este analiza:

```
PS C:\Users\Santi\SSL> ./tp4 archi.m
Declara a,Entera,
Read a,Entera,
Declara b,Entera,
Read b,Entera,
Declara d,Entera,
Declara Temp&1,Entera,
Sumar b,2,Temp&1
Declara Temp&2,Entera,
Sumar a,Temp&1,Temp&2
Almacena Temp&2,d,
Write a,Entera,
Write d,Entera,
Detiene ,,
PS C:\Users\Santi\SSL> █
```

El programa ejecuta y analiza correctamente.

### **Ejemplo 2 (error léxico):**

```
1  inicio#
2  leer(a,b);
3  cc:=a+(b+2);
4  escribir(cc,a+b);
5  fin
```

En este caso damos instrucciones similares a la anterior, solo que esta vez, introducimos un caracter que el compilador no reconoce. Al pasar por el scanner, el algoritmo del mismo lo coloca en el token 'ERRORLEXICO' provocando lo que vemos en pantalla.

```
PS C:\Users\Santi\SSL> ./tp4 archi.m
Error Lexico
Error Sintactico
Error Sintactico
Detiene ,,
PS C:\Users\Santi\SSL> █
```

Como respuesta, obtenemos en primer lugar el error léxico debido a lo antes mencionado, y dado a este caso particular, aparece a su vez, un error sintáctico ya que no se respeta la estructura esperada. El programa espera comenzar con el

token 'inicio' (lo cual no es respetado), a partir de ahí recibir una serie de sentencias y finalizar con el token 'fin'.

**Ejemplo 3 (error sintáctico y semántico):**

```
1  inicio
2  leer(a,b);
3  d:= 2 + texto;
4  leer();
5  escribir(a,d);
6  fin
```

Instrucciones que le pasamos.

```
PS C:\Users\Santi\SSL> ./tp4 archi.m
Declara a,Entera,
Read a,Entera,
Declara b,Entera,
Read b,Entera,
Declara d,Entera,
Declara texto,Entera,
Declara Temp&1,Entera,
Sumar 2,texto,Temp&1
Almacena Temp&1,d,
Error Sintactico
Declara ),Entera,
Read ),Entera,
Error Sintactico
Error Sintactico
Error Sintactico
Error Sintactico
Detiene ,,
PS C:\Users\Santi\SSL> █
```

Básicamente, luego de detectar el token 'inicio', este continúa llamando a 'listaDeSentencias()', lee correctamente las siguientes instrucciones hasta la 4ta línea de código, donde recibe la sentencia 'leer()' vacía. El programa espera de esta sentencia recibir una expresión/caracteres que entienda. Finalmente, dado que se produce el error sintáctico, se detiene el análisis y se deja de ejecutar.