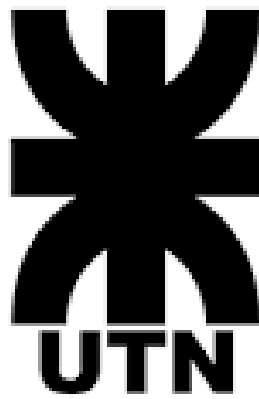


# INFORME TP 3 FLEX Y BISON

Materia: SSL

## Explicación del pensamiento detrás del trabajo



Fecha de entrega: 17/11/2023

Integrantes Grupo 25:

-CACACE, Guillermo Federico

-CALÓ, Ignacio

-GOMEZ PEREYRA, Manuel Francisco

- MAJER, Cecilia Alejandra

-TROSSERO, Agustín Francisco

Primeramente, armamos la estructura del lenguaje MICRO Fisher, reconociendo y denotando todos sus tokens en Flex/Lex. Separamos en DÍGITO, LETRA, IDENTIFICADOR y constEntera, implementandolos como REGEX; además de luego introducir los tokens necesarios de puntuación y operadores, junto al 'inicio' y 'fin' del programa. Aquí agregamos el mensaje de "Error léxico", en el caso de que se introduzca al flujo un lexema no perteneciente al lenguaje.

*Caso de prueba:* Una división (el operador '/' no pertenece al lenguaje)

```
Usuario@DESKTOP-4QE2QT2 MINGW64 ~  
$ /d/Universidad/UTN/2_Nivel/ssl/ssl-repo/tp3/ejecutable/tp3_DUMMY-g25.exe  
inicio id:=4/4; fin  
  
RUTINA SEMANTICA:      La longitud del ID es: 2  
RUTINA SEMANTICA:      Valor de la constante: 4  
Caracter inesperado: /  
  
Motivo del error: Error lexico
```

Posteriormente, en Bison definimos las estructuras válidas para un 'programa', el cual está compuesto por INICIO, sentencias y FIN. Describimos cuales son todas las sentencias válidas, y su estructura debida.

Personalizamos los errores sintácticos, como por ejemplo, un ID de más de 32 caracteres:

```
Usuario@DESKTOP-4QE2QT2 MINGW64 ~  
$ /d/Universidad/UTN/2_Nivel/ssl/ssl-repo/tp3/ejecutable/tp3_DUMMY-g25.exe  
inicio aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa:=4+5+2; fin  
  
RUTINA SEMANTICA:      La longitud del ID es: 53  
Motivo del error: La longitud del ID es mayor a la permitida
```

Además, si la ESTRUCTURA de las declaraciones es errónea, muestra por pantalla un error sintáctico. En este ejemplo, decidimos no terminar la sentencia con un ';' para que sea errónea:

```
Usuario@DESKTOP-4QE2QT2 MINGW64 ~  
$ /d/Universidad/UTN/2_Nivel/ssl/ssl-repo/tp3/ejecutable/tp3_DUMMY-g25.exe  
inicio id:=4 fin  
  
RUTINA SEMANTICA:      La longitud del ID es: 2  
RUTINA SEMANTICA:      Valor de la constante: 4  
  
Motivo del error: syntax error
```

Luego elegimos como rutinas semánticas, la devolución de longitud y valor de los identificadores introducidos en el programa, y además, el resultado de las operaciones (expresiones) que se ingresan. A continuación ejemplos de diferentes casos:

## CASO 1

Ingresamos dos ID, uno llamado 'id', de longitud 2 y valor 444, y otro llamado 'a', de longitud 1 y valor 8.

```
Usuario@DESKTOP-4QE2QT2 MINGW64 ~  
$ /d/Universidad/UTN/2_Nivel/ssl/ssl-repo/tp3/ejecutable/tp3_DUMMY-g25.exe  
  
inicio id:=444; a:=8; fin  
  
RUTINA SEMANTICA:      La longitud del ID es: 2  
RUTINA SEMANTICA:      Valor de la constante: 444  
  
RUTINA SEMANTICA:      La longitud del ID es: 1  
RUTINA SEMANTICA:      Valor de la constante: 8
```

## CASO 2

Ingresamos tres operaciones diferentes, una suma, una resta y una combinación de ambas. Las rutinas semánticas nos devuelven la longitud de los ID a los que están asignadas esas expresiones, los valores de cada número detectado en la operación, y el resultado de la misma. En el caso de una combinada, también nos va arrojando la resolución parcialmente de izquierda a derecha.

```
Usuario@DESKTOP-4QE2QT2 MINGW64 ~
$ /d/Universidad/UTN/2_Nivel/ssl/ssl-repo/tp3/ejecutable/tp3_DUMMY-g25.exe

inicio suma:=4+4; resta:=5-4; opCombinada:=4+4-6; fin

RUTINA SEMANTICA:      La longitud del ID es: 4
RUTINA SEMANTICA:      Valor de la constante: 4
RUTINA SEMANTICA:      Valor de la constante: 4
RUTINA SEMANTICA:      El resultado es 8
RUTINA SEMANTICA:      La longitud del ID es: 5
RUTINA SEMANTICA:      Valor de la constante: 5
RUTINA SEMANTICA:      Valor de la constante: 4
RUTINA SEMANTICA:      El resultado es 1
RUTINA SEMANTICA:      La longitud del ID es: 11
RUTINA SEMANTICA:      Valor de la constante: 4
RUTINA SEMANTICA:      Valor de la constante: 4
RUTINA SEMANTICA:      El resultado es 8
RUTINA SEMANTICA:      Valor de la constante: 6
RUTINA SEMANTICA:      El resultado es 2
```

} Suma

} Resta

} Operación combinada

## CASO 3

No implementamos las rutinas semánticas de verificar la declaración previa , ya que no tenemos una pila donde almacenar estos datos temporalmente, pero sí es válido desde un punto tanto léxico como sintáctico.

```
Usuario@DESKTOP-4QE2QT2 MINGW64 ~  
$ /d/Universidad/UTN/2_Nivel/ssl/ssl-repo/tp3/ejecutable/tp3_DUMMY-g25.exe  
  
inicio leer(4); escribir(4+5); fin  
RUTINA SEMANTICA:      Valor de la constante: 4  
RUTINA SEMANTICA:      Valor de la constante: 4  
RUTINA SEMANTICA:      Valor de la constante: 5  
RUTINA SEMANTICA:      El resultado es 9
```