

## Evaluación de Aprendizaje N° 2

### Unidades Temáticas: 9 y 10

**Objetivo:** Evaluar los aprendizajes sobre los contenidos de las unidades 9 y 10 y su relación con las unidades anteriores.

**Alcance:** Analizador Lexicográfico, Analizador Sintáctico, Generación de Código Intermedio: Polaca Inversa, Tercetos y Árbol Sintáctico.

**Evaluación:** Deberá tener 4 ejercicios bien. De lo contrario se pedirá re entregar el trabajo

**Presentación:** Las respuestas deben presentarse en un archivo formato PDF con el nombre *EA\_2\_Rtas\_ApellidoNombre*, en la Plataforma Miel hasta el día viernes 19/6/2020. Cada alumno deberá presentarlo a su profesor asignado en el archivo Profesores.

---

La siguiente gramática detalla la sintaxis de una sentencia de asignaciones múltiples. La sentencia asigna una expresión de sumas y multiplicaciones, a múltiples id's separados por un token de asignación llamado **asigna** cuyo lexema es el carácter =

La expresión consta de los tokens mas y mul cuyos lexemas son + y \* y sus operandos están representados por los tokens cte y id.

La funcionalidad de esta sentencia es asignar el resultado de expresión a cada uno de los id's.

Por ejemplo: f1=g1= 0

$$a=b=c=31*d*25+g$$

La gramática ya se encuentra aumentada.

Gramática < {**S, MULT, LISTA, EXP, TERM, FACTOR**}, {cte,id,asigna,mas,mul } , **S , Reglas** }

**Reglas:**

0.  $S \rightarrow MULT$
1.  $MULT \rightarrow LISTA EXP$
2.  $LISTA \rightarrow LISTA id \text{ asigna}$
3.  $LISTA \rightarrow id \text{ asigna}$
4.  $EXP \rightarrow EXP \text{ mas } TERM$
5.  $EXP \rightarrow TERM$
6.  $TERM \rightarrow TERM \text{ mul } FACTOR$
7.  $TERM \rightarrow FACTOR$
8.  $FACTOR \rightarrow id$
9.  $FACTOR \rightarrow cte$

APELLIDO: ..... NOMBRE: .....DNI..... Lenguajes y Compiladores – UNLAM		12/06/2020  SEGUNDA EVALUACION DE APRENDIZAJE
--	--	---

Se pide:

### Ejercicio Nro. 1

Escribir las acciones semánticas para generar árbol sintáctico.

### Ejercicio Nro. 2

Dibujar el árbol sintáctico para  $a=b=c=31*d*25+g$

### Ejercicio Nro. 3

Sea la sentencia TAKE que se asigna a un identificador, presente en la EA1.

$id = TAKE (Operador ; cte ; [lista\ de\ identificadores])$ .

Esta función toma como entrada un *operador* de suma o multiplicación, una *constante* entera y una *lista de identificadores*. Esta función devuelve el valor que resulta de aplicar el *operador* a los primeros “n” elementos de la lista. El valor de n quedará establecido en la componente *cte*.

Los elementos de la lista están separados por blancos.

El resultado de la función puede ser utilizado en otras expresiones dentro del lenguaje.

Ej : TAKE (+;1;[id]) Si id tomase el valor 3 devuelve 3

TAKE (+;3; [a b c d e]) Si a tomase el valor 10, b el valor 20, c el valor 30, d el 40 y e el valor 50) devuelve 60 (resultado de sumar los 3 primeros ids de la lista)

Se pide:

Escribir el conjunto de tercetos necesarios para representar la sentencia

TAKE (+;3; [a1 b1 c1 d1 e1])

### Ejercicio 4

Si el lenguaje controlara que la cantidad de elementos de la lista sea mayor o iguales al valor de la *cte*. en la función del ejercicio anterior (TAKE). ¿En qué regla/s realizaría dicha validación? ¿Cómo se haría?

### Ejercicio 5

“El resultado de la función puede ser utilizado en otras expresiones dentro del lenguaje.”.

Analice esta afirmación e indique donde puede utilizarse el resultado y como se implementaría.