

# Taller 1

## Lenguajes de programación

### Entregable #2

#### *Categorías léxicas y expresiones regulares:*

Para el problema planteado se hace uso de varias categorías léxicas distintas, además de expresiones regulares que permitirán su implementación, todas ellas serán explicadas a continuación.

- **DIGITO:** Expresión regular  $[0-9]$ , el cual permite generar cualquier número entero entre éste rango (del 0 al 9).
- **DÍGITOS:** Expresión regular  $\text{DIGITO}^+$ , por medio de ésta podemos acceder a cualquier número entero positivo de al menos un dígito.
- **NUMERO:** Categoría léxica implementada con el fin de poder manejar números decimales, definidos por la expresión regular  $\text{DIGITOS}(\text{DIGITOS})^*$ , en la cual podemos definir cualquier cadena de dígitos, seguidas de un punto el cual es seguido de cualquier cantidad de dígitos que se puedan necesitar.
- **VARARIT:** Ésta categoría léxica será utilizada para almacenar las cuatro posibles variables (p, q, r, t) con las que haremos referencia a las operaciones de suma, resta, multiplicación y división, para los predicados aritméticos. Está descrita por medio de la expresión regular  $[p | q | r | t]$ , en la que se nos permite ingresar cualquiera de dichas letras.
- **VARALGE:** En ésta categoría léxica, el proceso es el mismo que el descrito en la categoría anterior, ahora utilizando las tres posibles variables algebraicas (x, y, z) que serán utilizadas en los predicados algebraicos, de igual manera la expresión regular utilizada es  $[x | y | z]$ , que nos permite realizar dicho proceso.
- **PREDARIT:** Para los predicados aritméticos tenemos 6 posibles escenarios válidos a analizar, todos comienzan con una variable aritmética definida anteriormente, la cual puede estar acompañada de hasta 5 números enteros a su derecha, por lo tanto definimos los posibles casos de la siguiente manera:
  - *Caso 1:* Se define el predicado únicamente con la variable aritmética, sin números a continuación de la misma.
    - Expresión regular: VARARIT.
  - *Caso 2:* Se define el predicado con la variable aritmética, seguida de un dígito.
    - Expresión regular: VARARIT DIGITO
  - *Caso 3:* Se define el predicado con la variable aritmética, seguida de dos dígitos.
    - Expresión regular: VARARIT DIGITO DIGITO
  - *Caso 4:* Se define el predicado con la variable aritmética, seguida de tres dígitos.

- Expresión regular: VARARIT DIGITO DIGITO DIGITO
  - *Caso 5:* Se define el predicado con la variable aritmética, seguida de cuatro dígitos.
    - Expresión regular: VARARIT DIGITO DIGITO DIGITO DIGITO
  - *Caso 6:* Se define el predicado con la variable aritmética, seguida de cinco dígitos.
    - Expresión regular: VARARIT DIGITO DIGITO DIGITO DIGITO DIGITO
- **PREDALGE:** Para los predicados algebraicos seguimos la misma lógica implementada previamente con los predicados aritméticos, poseemos de igual manera los 6 posibles escenarios pero en éste caso la variable a utilizar es la categoría léxica VARALGE, la cual contiene las variables que necesitamos. En ese orden de ideas los escenarios serían los siguientes:
  - *Caso 1:* Se define el predicado únicamente con la variable algebraica, sin números a continuación de la misma.
    - Expresión regular: VARALGE.
  - *Caso 2:* Se define el predicado con la variable algebraica, seguida de un dígito.
    - Expresión regular: VARALGE DIGITO
  - *Caso 3:* Se define el predicado con la variable algebraica, seguida de dos dígitos.
    - Expresión regular: VARALGE DIGITO DIGITO
  - *Caso 4:* Se define el predicado con la variable algebraica, seguida de tres dígitos.
    - Expresión regular: VARALGE DIGITO DIGITO DIGITO
  - *Caso 5:* Se define el predicado con la variable algebraica, seguida de cuatro dígitos.
    - Expresión regular: VARALGE DIGITO DIGITO DIGITO DIGITO
  - *Caso 6:* Se define el predicado con la variable algebraica, seguida de cinco dígitos.
    - Expresión regular: VARALGE DIGITO DIGITO DIGITO DIGITO DIGITO
- **PUNCTUATION:** Categoría léxica que incluye los símbolos de puntuación (‘,’, . , : , ; ), para su implementación se utiliza un solo token llamado TKN\_PUNCTUATION el cual devuelve un valor numérico, el cual será leído y posteriormente utilizado por el analizador sintáctico.
- **OPERARIT:** Categoría léxica que incluye los símbolos que permiten realizar operaciones aritméticas ( +, -, \*, / ), para su implementación se utiliza un solo token llamado TKN\_OPERARIT el cual devuelve un valor numérico, el cual será leído y posteriormente utilizado por el analizador sintáctico.

- **LOGICVAL:** Categoría léxica que incluye los posibles resultados al realizar una operación lógica ( T, F ), para su implementación se utiliza un solo token llamado TKN\_LOGICVAL, el cual devuelve un valor numérico, el cual será leído y posteriormente utilizado por el analizador sintáctico.
- **OPEREL:** Categoría léxica que incluye los posibles operadores de comparación aritmética ( <, >, = ), para su implementación se utiliza un solo token llamado TKN\_OPEREL, el cual devuelve un valor numérico, el cual de igual manera será leído y posteriormente utilizado por el analizador sintáctico.
- **LOGICOPER:** Categoría léxica que incluye los símbolos de equivalencia lógica (E, V,  $\neg$ ,  $\wedge$ ,  $\vee$  ), para su implementación se utiliza un solo token llamado TKN\_LOGICOPER el cual devuelve un valor numérico, el cual será leído y posteriormente utilizado por el analizador sintáctico.