

# ut4\_pd0

## Ejercicio 1

Desarrolla un algoritmo para, dado un árbol binario que representa expresiones aritméticas (operadores, constantes y variables):

- Dados valores concretos de variables, sustituirlas en el árbol
- Evaluar la expresión representada en el árbol y emitir el resultado

NOTAS: se asume que las operaciones son binarias, y son suma, resta, multiplicación y división.

## Solución

Debo procesar el árbol en inorden para poder evaluar primero el subárbol izquierdo, luego el nodo actual y finalmente el subárbol derecho. De esta manera, puedo procesar correctamente las etiquetas de los nodos del árbol de expresión.

precondicion → árbol lleno

postcondicion → no altera estado

```
Procesar_ArbolDeExpresion de TArbolBB
  if EsVacio
    return "Error"
  expresion = raiz.StringProcesarAde
  calculoDeExpresion = raiz.ProcesarAde
  return print(la expresion representada es + expresion, el
```

```
ProcesarAE de TNodeBB
  if nodo no es hoja
    exp 1 = hijo izq.procesarAE
    exp 2 = hijo der.procesarAE
    return procesarOperacion(a, b, c)
```

```
if es hoja
    return this.etiqueta
```

```
procesarOperacion de TNodeBB(a, b ,c)
    switch (b)
        case *
            return a * b
        case +
            return a + b
        case -
            return a - b
        case /
            return a/b
```

## Ejercicio 2

1. Dada una expresión aritmética, representada mediante notación prefija, diseña un algoritmo que cree el árbol binario de expresión correspondiente.
2. Considera las posibles situaciones de error
3. Escribe varias expresiones aritméticas (con paréntesis), exprésalas en notación prefija y luego ejecuta manualmente el algoritmo diseñado para insertar la expresión en el árbol.  
Por último, para cada árbol generado, ejecuta manualmente el algoritmo diseñado en el Ejercicio 1.

```
TArbolBB construirAE(string expresion)
    nuevoElemento = expresion[0]
    raiz = nuevoElemento
    raiz.insertarAE(expresion, contador)
```

```
TElemento1BB String insertarAE(string expresion)
    if etiqueta es operando
        // inserto hijo derecho
        expresion = [1..]
```

```

        new elemento(expresion[0])
        hijo izq = elemento
        expresion = hijoizq.insertar(expresion)
        // inserto hijo derecho
        expresion = [1..]
        new elemento(expresion)
        hijo Der = elemento
        expresion = hijoder.insertar(expresion)

    return expresion;

if etiqueta es letra
    return expresion

```

## Ejercicio 3

1. Diseña clases Java para representar un árbol binario y el nodo correspondiente, que sea útil para representar el modelo de árbol de expresión aritmética.
2. Desarrolla en estas clases los métodos correspondientes a los algoritmos desarrollados en los ejercicios 1 y 2.
3. Desarrolla un programa para probar estos métodos.