



# **INSTITUTO TECNOLÓGICO DE NUEVO LEÓN**



## **“PROYECTO UNIDAD 1”**

**ALUMNO: JESUS ALEXIS JIMENEZ REYNA**

**MATERIA: LENGUAJE Y AUTOMATAS II**

**CATEDRATICO: JUAN PABLO ROSAS B.**

## PROYECTO 1: ÁRBOL DE EXPRESIONES

### INTRODUCCION

Un árbol binario es una estructura de datos en la cual cada nodo puede tener un hijo izquierdo y un hijo derecho. No pueden tener más de dos hijos (de ahí el nombre "binario"). Si algún hijo tiene como referencia a null, es decir que no almacena ningún dato, entonces este es llamado un nodo externo. En el caso contrario el hijo es llamado un nodo interno. Usos comunes de los árboles binarios son los árboles binarios de búsqueda, los montículos binarios y Codificación de Huffman.

La Notación Como su nombre lo indica se refiere a que el operador ocupa la posición después de los operando sus características principales son:

- El orden de los operados se conserva igual que la expresión infija equivalente no utiliza paréntesis ya que no es una operación ambigua.
- La operación posfija no es exactamente lo inverso a la operación prefija equivalente.
- El orden es primer operando, segundo operando, operando.

$$(A+B)*C \quad AB+C*$$

La notación infija es la forma más común que utilizamos para escribir expresiones matemáticas, estas notaciones se refiere a que el operador esta entre los operadores. La notación infija puede estar completamente patentizada o puede basarse en un esquema de precedencia de operadores así como el uso de paréntesis para invalidar los arreglos al expresar el orden de evaluación de una expresión:

$$3*4 = 12$$

$$3*4+ = 14$$

$$3*(4+2) = 18$$

### QUE SE VA A HACER

Realizar un árbol binario de búsqueda de expresiones el cual automáticamente se encarga de reconocer los operadores y los números de manera que genera el árbol binario, el objetivo es que el usuario introduzca una expresión y el programa arroje la expresión en infija y postfija.

Se realizó en el lenguaje de programación java utilizando el NetBeans.

## DESARROLLO

Se realizaron 10 expresiones diferentes, en el cual fue tomado el tiempo de ejecución de cada uno para así observar cuál de las operaciones afectaba más el tiempo en que se está ejecutando en el programa.

Las pruebas de las expresiones se realizaron con la siguiente computadora:

**EQUIPO:** LAPTOP ACER ASPIRE E15 15.6 FULL HD I3 6GB

**RAM:** 1TB HDD

**PROCESADOR:** AMD A9 7th GEN with Turbo CORE Technology up to 3.5

**TIPO DE SISTEMA:** WINDOWS 64 BITS

Expresión	Tiempo de ejecución(s)
$3^3+(5-2)*6/3$	11 seconds
$(4+4)/(2-3)+2$	2 seconds
$6-9*8/7+3$	1 second
$1+2/3*4-5$	3 seconds
$(12^2)^2 / 5 + 6 / 2 - 4 * 1(14^2)^2$	14 seconds
$1+4(6*6)$	1 second
$8-2+4/5^2$	5 seconds
$3+9-9/6-5+7*4$	4 second
$(9+(4+5))^2$	22 seconds
$(12^2)^2 / 5 + 6 / 2 - 4 * 1$	56 seconds

## RESULTADO

En esta grafica mostramos el tiempo de ejecución que se tardó cada una de las expresiones



## CONCLUSIONES

En este proyecto se realizó un programa que fuera capaz de realizar una función completa y de mismo modo que se tomara el tiempo que se tardaba en ejecutar. La función tenía que ser con números y operadores aritméticos, ya que después tenía que mostrar el árbol de expresiones, que es llamado la notación.

Bibliografía

<http://interactivepython.org/runestone/static/pythoned/Trees/RepresentacionDeListaDeListas.html>