

# Instituto Tecnológico de Nuevo León



## INGENIERÍA EN SISTEMAS COMPUTACIONALES

Lenguajes y Autómatas II

Proyecto 1

Salvador Eliud Carranza de la Rosa

15480101

Prof. Juan Pablo Rosas Baldazo

Guadalupe N.L.

01 de Junio del 2018

## Introducción:

En el siguiente reporte se describe lo realizado para la realización de la aplicación que se pidió como proyecto de la unidad 1 de la materia Lenguajes y autómatas II.

El presente proyecto consiste en realizar el código de un programa en algún lenguaje de programación el cual consiste en insertar una expresión aritmética y ordenarla de las diferentes formas vistas en clase, el programa a realizar debería de tener una interfaz en la cual se pudiera insertar cualquier expresión aritmética posteriormente ordenar los datos insertados, crear un árbol binario con dichos datos y mostrarlo gráficamente en pantalla.

Pseudocodigo:

```
imprimir ("Digite expresión que desea evaluar")
String infija, infija1 = la expresión
imprimir("Resultado y ordenes de la expresión")
String prefija= convertir expresión a prefija
String posfija = convertir expresión a posfija
imprimir ("La expresión posfija es: " + posfija)
imprimir("La expresión infija es: " + infija)
imprimir("La expresión prefija es: " + prefija)
regresa (Resultado de la expresión)
```

```
pila = caracteres de la expresión
for (recorrido de la pila)
separar carácter por carácter
si (si el carácter no es operador)
si (si la pila está vacía)
se apila carácter en la pila
si no
int pe = prioridad en la expresión del operando
int pp = prioridad en pila del siguiente operando
si (pe > pp)
apilar operando en pila
si no
prefija += se desapila el operando de la pila
se apila el operando en la pila e ingresa a prefija
si no
prefija += el operador se ingresa a prefija
mientras (la pila no esté vacía)
prefija += se desapilan los operandos y se ingresan a prefija
regresa el orden de prefija
para convertirpos (String infija)
pila = número de caracteres de la expresión
for (recorrido de la pila)
char letra = separar carácter por carácter
si (si el carácter es operador)
si (si la pila está vacía)
se apila carácter en la pila
si no
int pe = prioridad en la expresión del operador
int pp = prioridad en pila del operador
si (pe > pp)
apilar operador en pila
si no
posfija = se desopila el operador de la pila
se apila el operador en la pila e ingresa a posfija
sino
posfija = el operando se ingresa a posfija
mientras (la pila no esté vacía)
```

posfija = se desapilan los operadores y se ingresan a posfija  
devuelve el orden de posfija  
private static int prioridadEnExpresion (char operador)  
si (operador == '^') devuelve ID# 4  
si (operador == '\*' || operador == '/') devuelve ID# 2  
si (operador == '+' || operador == '-') devuelve ID# 1  
si (operador == '(') devuelve ID# 5  
regresa 0

esta n es = tamaño de la pila  
tope = 0  
pila = tiene un tamaño = n  
si esta vacía regresa un 0  
si está llena regresa el número de elementos en la pila  
recibe un dato  
si (la pila esta llena)  
regresa un falso  
si no  
posicion de la pila actual = dato  
posicion de la pila avanza 1  
regresa un verdadero  
si (la pila esta vacía)  
regresa un valor nulo  
posición de la pila retrocede 1  
regresa posición de la pila  
regresa posición anterior de la pila

## Descripción:

El proyecto consiste en Introducir por teclado una expresión matemática de números enteros.

1. Los operadores a utilizar son:  $\wedge$ ,  $*$ ,  $/$ ,  $+$  y  $-$ .
2. Deberá Imprimir en pantalla la expresión en notación prefija.
3. E Imprimir en pantalla la expresión en notación postfija.

## Resultados:

En este proyecto realizamos un código en Java que nos permite insertar una operación aritmética y esta operación la ordena en notación pre-fija y notación post-fija.

Como resultado tenemos que al darle de entrada una operación nos da como salida dichas notaciones en el orden prefija y postfija como se muestra a continuación:

Entrada:

$a + c/d * e$

Salida:

La expresion posfija es:  $acd/e*+$

La expresion prefija es:  $+a/c*de$

## Conclusiones:

Llegamos a la conclusión de los arboles binarios sirven para organizar datos para facilitar su manipulación, ya sea el ingreso, borrado o búsqueda de datos, y precisamente una de las principales ventajas de los árboles binarios es la búsqueda, ya que como en muchos algoritmos de búsqueda necesitamos tener la información ordenada y en los árboles binarios precisamente los datos se van ingresando de forma ordenada.

En este proyecto se logro realizar un programa que al introducirle una expresión aritmética este imprime en pantalla dicha expresión en orden pre-fijo y post-fijo

## Referencias:

areadelprogramador. (2017). CONVERTIR EXPRESIÓN INFIJA A POSTFIJA Y PREFIJA EN JAVA. 01/06/18, de Area del programador Sitio web:  
<http://www.areadelprogramador.com/java/convertir-expresion-infija-a-postfija-y-prefija-en-java/>

Ivan Said. (2008). Código: Conversión de Infijo a Prefijo por medio de pilas en Java. 01/06/18, de wordpress.com Sitio web:  
<https://srpoder.wordpress.com/2008/10/05/codigo-conversion-de-infijo-a-prefijo-por-medio-de-pilas/>