



Instituto Tecnológico de Estudios Superiores de Monterrey

Segundo Proyecto: Software Design Document
Matemáticas Computacionales

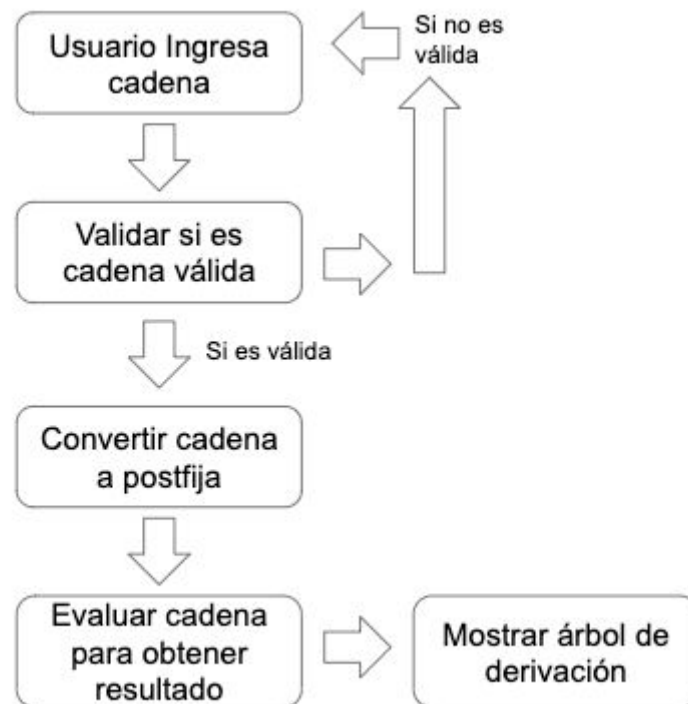
Luis Ricardo Peña Llamas

Santiago Yeomans

A01251000

Miércoles 28 de Octubre de 2020

Vista general del programa:



Algoritmos:

1. Validación de cadena:

Método: esValida(String cadena)

1. Separar la cadena en un arreglo y recorrer el arreglo desde la posición 0 a la n.
2. Checar para cada elemento de la cadena el elemento izquierdo y el derecho:
3. Si el elemento es un operador:
 - a. Si el elemento a la izquierda es un número o un paréntesis cerrado.
 - b. Si el elemento a la derecha es un número o un paréntesis abierto.
 - c. Si no la cadena no es válida y termina el algoritmo.
4. Si el elemento es un paréntesis abierto:
 - a. Si el elemento a la izquierda es un operador o un paréntesis abierto.
 - b. Si el elemento a la izquierda es un número o un paréntesis abierto.
 - c. Si no, la cadena no es válida y termina el algoritmo.
 - d. Si la pila está vacía o el tope de la pila es un paréntesis abierto, añadir a la pila el elemento.
 - e. Si no, la cadena no es válida y termina el algoritmo.
5. Si el elemento es un paréntesis cerrado:
 - a. Si el elemento a la izquierda es un número o un paréntesis abierto o un operador.

- b. Si el elemento a la derecha es un operador o un paréntesis cerrado.
 - c. Si no, la cadena no es válida y termina el algoritmo.
 - d. Si la pila está vacía, la cadena no es válida y termina el algoritmo.
 - e. Si el tope de la pila es igual a un paréntesis abierto, sacar el último elemento de la pila y continuar.
- 6. Si el elemento es un número:
 - a. Si el elemento a la izquierda es un número o un paréntesis abierto o un operador.
 - b. Si el elemento a la derecha es un número o un paréntesis cerrado o un operador.
 - c. Si no, la cadena no es válida y el algoritmo termina.
- 7. Si la pila no está vacía, la cadena no se acepta y termina el algoritmo.
- 8. La cadena se acepta.
- 9. Fin del algoritmo.

2. Convertir cadena a postfija:

Método: convertiraPostfija(String infija)

- 1. Obtener caracteres de la expresión y repetir los pasos 2 y 4 para cada carácter.
- 2. Si es operando, pasarlo a la expresión postfija.
- 3. Si es operador:
 - 3.1. Si la pila está vacía, meterlo en la pila. Repetir a partir del 1.
 - 3.2 Si la pila no está vacía:
 - Si la prioridad del operador leído es mayor que la prioridad del operador cima de la pila, meterlo en la pila y repetir a partir de 1.
 - cima de la pila, meterlo en la pila y repetir a partir del 1.
 - Si la prioridad del operador es menor o igual que la prioridad del operador de la cima, sacar cima de la pila y pasarlo a la expresión postfija, volver a 3.
- 4. Si es paréntesis derecho:
 - 4.1. Sacar cima de la pila y pasarlo a postfija
 - 4.2. Si nueva cima es paréntesis izquierdo , suprimir elemento cima.
 - 4.3 Si cima no es paréntesis izquierdo, volver a 4.1
 - 4.4 Volver a partir de 1.
- 5. Si quedan elementos en la pila pasarlos a la expresión postfija.
- 6. Fin del algoritmo

3. Evaluar cadena postfija:

Método: evaluarPostfija(String postfija)

- 1. Examinar postfija desde el elemento 1 hasta el n. Repetir los pasos 2 y 3 para cada elemento de cadena en postfijo.
- 2. Si el elemento es un operando meterlo en la pila.

3. si el elemento es un operador, lo designamos con "&", entonces:
 - Sacar los dos elementos superiores de la pila, los llamamos X e Y, respectivamente.
 - Evaluar $Y \& X$, el resultado es $Z = Y \& X$.
 - El resultado z, meterlo en la pila.
4. Repartir a partir del paso 1.
5. El resultado de la evaluación de la expresión está en el elemento cima de la pila.
6. Fin del algoritmo.

4. Obtener árbol de derivación:

Método: getArbol(String cadena)

1. Recorrer cadena desde el elemento 1 hasta el n. Repetir pasos 1 y 2 para cada elemento de la cadena
2. Si el elemento es un número:
 - a. Crear Nodo de tipo 'Node' con el elemento actual de la cadena
 - b. Hacer push a la pila de nodos con el nodo recién creado.
3. Si el elemento es un operador:
 - a. Crear un nuevo nodo con el valor actual de la cadena
 - b. Sacar la cima de la pila y asignarlo como el nodo hijo izquierdo del nodo recién creado.
 - c. Sacar la cima de la pila y asignarlo como el nodo hijo derecho del nodo recién creado.
4. Sacar la cima de la pila y asignarlo como la raíz del árbol y devolver el valor.
5. Fin del algoritmo.