

# Documentación calculadora en JS vanilla

## Proceso de solución

### Primer intento

Para poder solucionar el ejercicio primero comencé explorando con cadenas de texto, dada una cadena de texto

1. Se abre un bucle que itere la cadena
2. Cuando encontrara un operador intentaría realizar la operación
3. El resultado de esta se guardaba en una variable que después se utilizaría para hacer la siguiente operación (si es que hubiese)
4. Así hasta que terminara el bucle y se mandara mostrar por pantalla

El problema con este tipo de solución es que solo me sumaba, restaba, dividía y multiplicaba los datos de forma individual, es decir si metíamos  $3+3+3+33$  el resultado era el correcto (42), sin embargo, con operaciones más complejas, daba problemas por ejemplo  $3+3/3$  y aun faltaba validar la precedencia de operaciones

### Segundo intento

En este nuevo intento investigue aun mas en internet sobre alguna solución posible o algún tipo de algoritmo que me pudiera ayudar a solucionarlo, así fue como encontré los tipos de notaciones (posfija, infija y prefija), comencé a explorar más acerca de esto y así halle una posible solución, yo decidí ocupara la notación posfija, ya que me pareció la más apropiada

1. Recibir la cadena de la pantalla
2. Crear un bucle que recorriera esta cadena
3. Si encontraba un número guardaba ese número en una pila (array)
4. Si encontraba un operador
  - a. Primero si no había nada en la pila de los operadores, este pasaba directo
  - b. Si había algo en la pila debemos verificar si:
    - i. Si el operador que viene es de igual precedencia entonces sacamos el que esta en la pila de operadores para meterlo en la pila de números y metemos el nuevo a la pila de operadores
    - ii. Si el operador que viene es de mayor precedencia entonces pasa directo a la pila de operadores
    - iii. Si el operador que viene es de menor precedencia sacamos todos los operadores que están en la pila operadores y metemos el que viene
  - c. Si estamos en la última posición del arreglo y pila de operadores tiene operadores entonces sacamos esos operadores y los metemos en pila de números hasta vaciar la pila
5. El resultado de esta notación se metía en un arreglo que después ocuparemos
6. Ahora que ya tenemos nuestra notación posfija en un arreglo pasamos a analizarla con un bucle
7. Creamos el bucle
8. Lo primero es comprobar si lo que viene es un numero o un operador

9. Si es un numero lo guardamos en una pila (salvarDatos)
10. Si es un operador, primero debemos averiguar que operador es (+, -, /, \*)
11. Una vez comprobado que tipo de operador es, se debe hacer la operación del penúltimo numero que esta en la pila SalvarDatos (+, -, \* /) con el último número y el resultado se guarda al final de la pila Salvar datos. (en esta operación se validó la división entre 0 = error)
12. Esta última instrucción se repite hasta que ya no quedan más datos de nuestra pila en notación posfija
13. Finalmente el resultado de esto se manda a la pantalla de la calculadora

Lo interesante de este método es que ya validamos la precedencia de los operadores, y que se puede llegar a hacer hasta con potencia, raíz cuadrada y paréntesis.

Yo solo lo realice hasta la multiplicación y división y también solo se pueden hacer operaciones unitarias  $3-4-5+5/9*2$

## Algoritmo

La parte de documentación técnica se encuentra dentro del código JS, allí explico para que sirven cada una de las variables y los métodos que se ocuparon, pero a grandes rasgos,

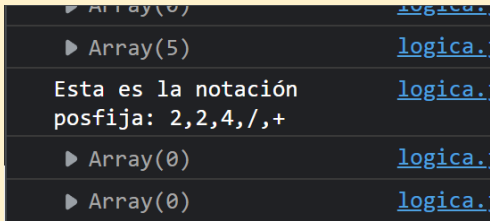
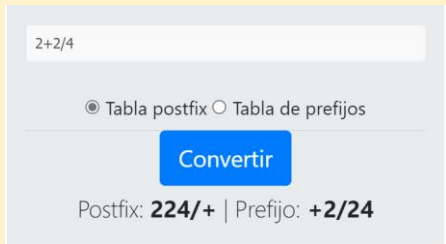
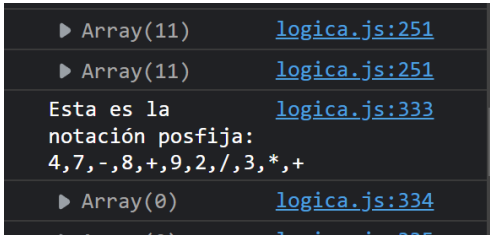
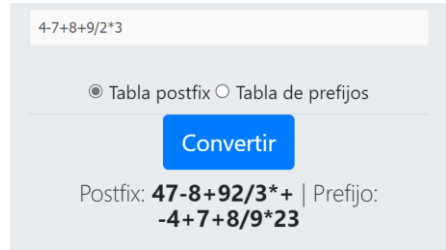
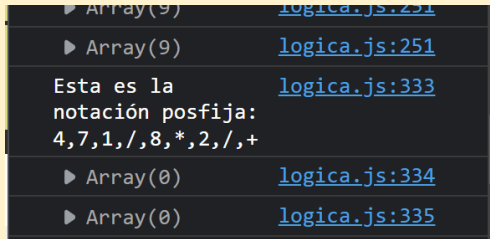

1. Se introduce una cadena a través de la calculadora
2. Cuando se presiona igual se manda esa cadena a transformarla en notación posfija
3. Este resultado se guarda en otro arreglo
4. Y ahora se realiza la solución de la notación posfija bajo la premisa de:
  - a. Guardo números hasta que encuentro un operador,
  - b. es entonces que realizo la operación de los dos últimos números y
  - c. guardo el resultado al final de este arreglo para seguir buscando operadores
  - d. si ya terminé.
5. Imprimo el resultado en la pantalla de la calculadora.

## Pruebas unitarias

Para la comprobar que la notación posfija esta correcta ocupamos una calculadora en línea:  
[Conversor de infijo a postfijo/prefijo en línea \(sas.com.ru\)](https://sas.com.ru/)

1.  $2+2/4 = 2.5$
2.  $4-7+8+9/2*3 = 18.5$
3.  $4+7/1*8/2 = 32$

## Notación posfija:

Programa	Calculadora en línea
 <p>► Array(0) <a href="#">logica.js:251</a> ► Array(5) <a href="#">logica.js:251</a> Esta es la notación posfija: 2,2,4,/,+ ► Array(0) <a href="#">logica.js:333</a> ► Array(0) <a href="#">logica.js:334</a></p>	 <p>2+2/4 <input checked="" type="radio"/> Tabla postfix <input type="radio"/> Tabla de prefijos <b>Convertir</b> Postfix: <b>224/+</b>   Prefijo: <b>+2/24</b></p>
 <p>► Array(11) <a href="#">logica.js:251</a> ► Array(11) <a href="#">logica.js:251</a> Esta es la notación posfija: 4,7,-,8,+,9,2,/,3,*,+ ► Array(0) <a href="#">logica.js:334</a> ► Array(0) <a href="#">logica.js:335</a></p>	 <p>4-7+8+9/2*3 <input checked="" type="radio"/> Tabla postfix <input type="radio"/> Tabla de prefijos <b>Convertir</b> Postfix: <b>47-8+92/3*+</b>   Prefijo: <b>-4+7+8/9*23</b></p>
 <p>► Array(9) <a href="#">logica.js:251</a> ► Array(9) <a href="#">logica.js:251</a> Esta es la notación posfija: 4,7,1,/,8,*,2,/,+ ► Array(0) <a href="#">logica.js:334</a> ► Array(0) <a href="#">logica.js:335</a></p>	 <p>4+7/1*8/2 <input checked="" type="radio"/> Tabla postfix <input type="radio"/> Tabla de prefijos <b>Convertir</b> Postfix: <b>471/8*2/+</b>   Prefijo: <b>+4/7*1/82</b></p>

Resultados:

Operación	Programa	Calculadora científica online
<b><math>2+2/4</math></b>		
<b><math>4-7+8+9/2*3</math></b>		
<b><math>4+7/1*8/2</math></b>		