



UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE
UNIDAD DE EDUCACIÓN A DISTANCIA



Asignatura:

Programación Integrativa de Componentes Web

Ing.

Vilmer David Criollo Chanchicocha

Tema:

Calculadora Básica con JavaScript y Bootstrap

Nombre:

Andres Alejandro Toledo Rojas

Carlos Ramiro Yáñez Yazán

Luis David Flores Pillajo

NRC:

23407

Quito...5...junio de 2025

Calculadora Básica con JavaScript y Bootstrap

Objetivo General

Desarrollar una calculadora básica funcional aplicando JavaScript puro para la lógica y Bootstrap 5 para el diseño visual y responsivo, con el fin de reforzar el manejo del DOM, funciones, estructuras condicionales y el sistema de grillas de Bootstrap.

Objetivos Específicos

- Implementar la lógica de operaciones básicas (suma, resta, multiplicación y división) utilizando JavaScript puro, aplicando validaciones como evitar la división por cero.
- Aplicar el sistema de grillas y utilidades de Bootstrap para organizar visualmente los botones y elementos de la interfaz, asegurando un diseño limpio y ordenado.
- Diseñar una interfaz responsive que se adapte correctamente a distintos tamaños de pantalla (celular, tableta y escritorio), manteniendo la usabilidad y claridad.

Desarrollo

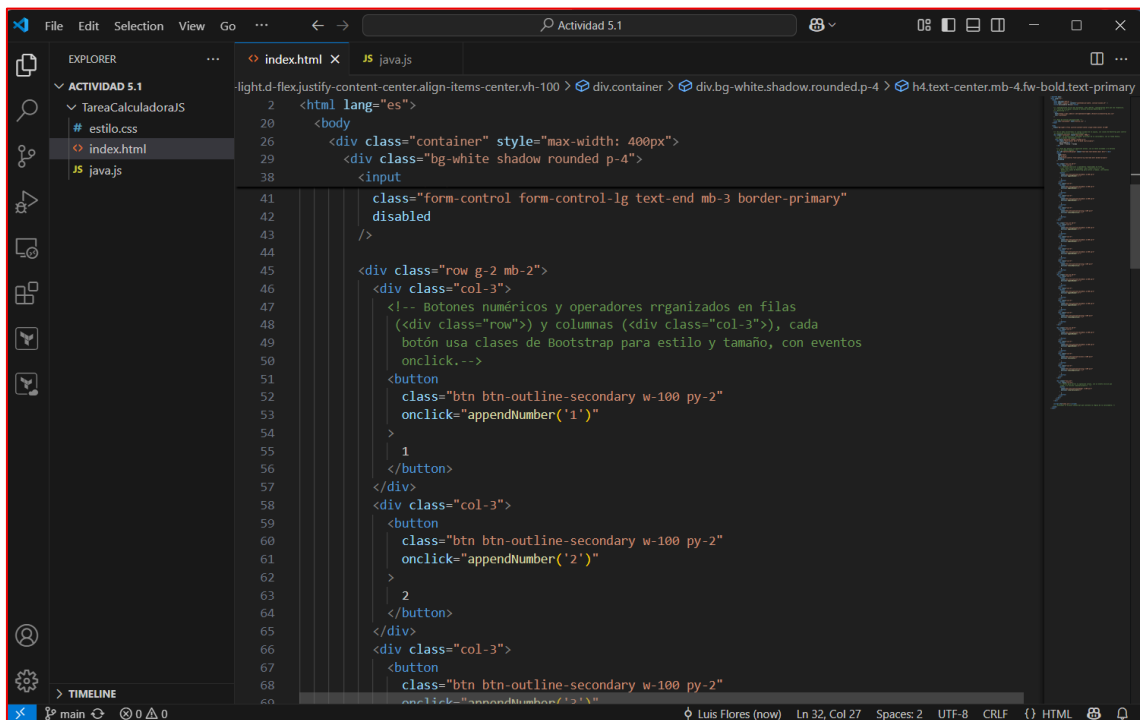
1. Enlace a GitHub

<https://github.com/ldflores8/TareaCalculadoraJS#>

2. Código Fuente

Para este trabajo vamos a desarrollar una calculadora que nos permita realizar las operaciones básicas y un botón que nos permita limpiar los datos, para realizar todas las operaciones vamos a hacer uso de un archivo de JavaScript.

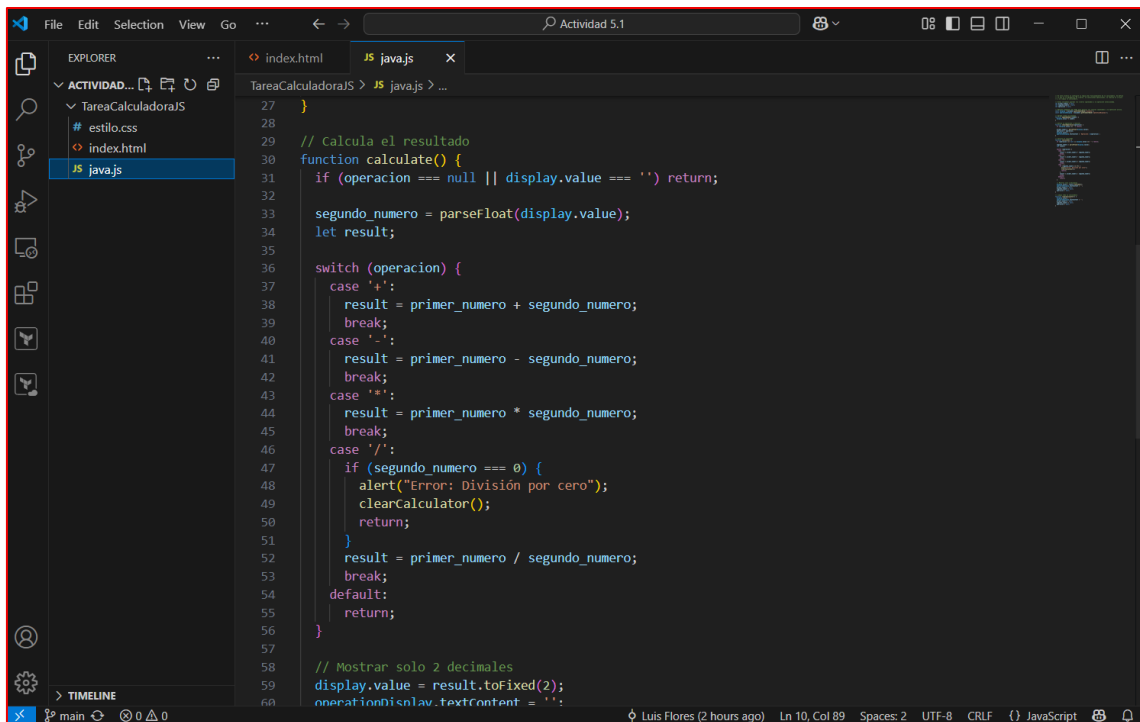
Index.html



```
2 <html lang="es">
20 <body>
26 <div class="container" style="max-width: 400px">
29 <div class="bg-white shadow rounded p-4">
38 <input
41 class="form-control form-control-lg text-end mb-3 border-primary"
42 disabled
43 />
44
45 <div class="row g-2 mb-2">
46 <div class="col-3">
47 <!-- Botones numéricos y operadores rganizados en filas
48 (<div class="row">) y columnas (<div class="col-3">), cada
49 botón usa clases de Bootstrap para estilo y tamaño, con eventos
50 onclick.-->
51 <button
52 class="btn btn-outline-secondary w-100 py-2"
53 onclick="appendNumber('1')"
54 >
55 1
56 </button>
57 </div>
58 <div class="col-3">
59 <button
60 class="btn btn-outline-secondary w-100 py-2"
61 onclick="appendNumber('2')"
62 >
63 2
64 </button>
65 </div>
66 <div class="col-3">
67 <button
68 class="btn btn-outline-secondary w-100 py-2"
69 onclick="appendNumber('3')"
```

Como se puede observar se implementó el archivo principal que va a contener los botones, div y los estilos Bootstrap. Ahora procedemos a definir la estructura del archivo JavaScript.

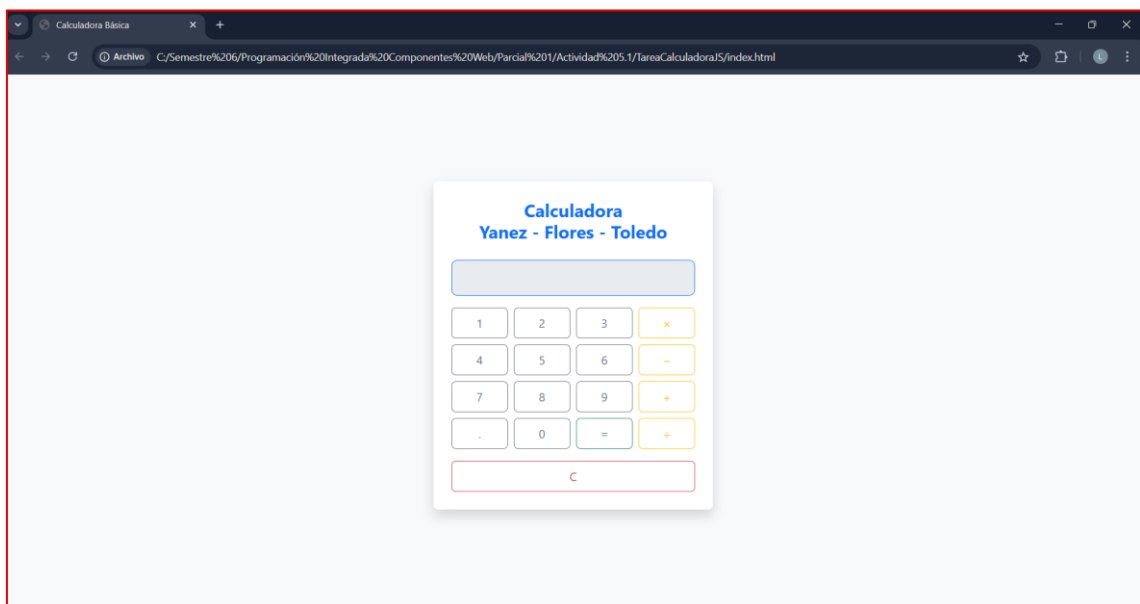
Java.js



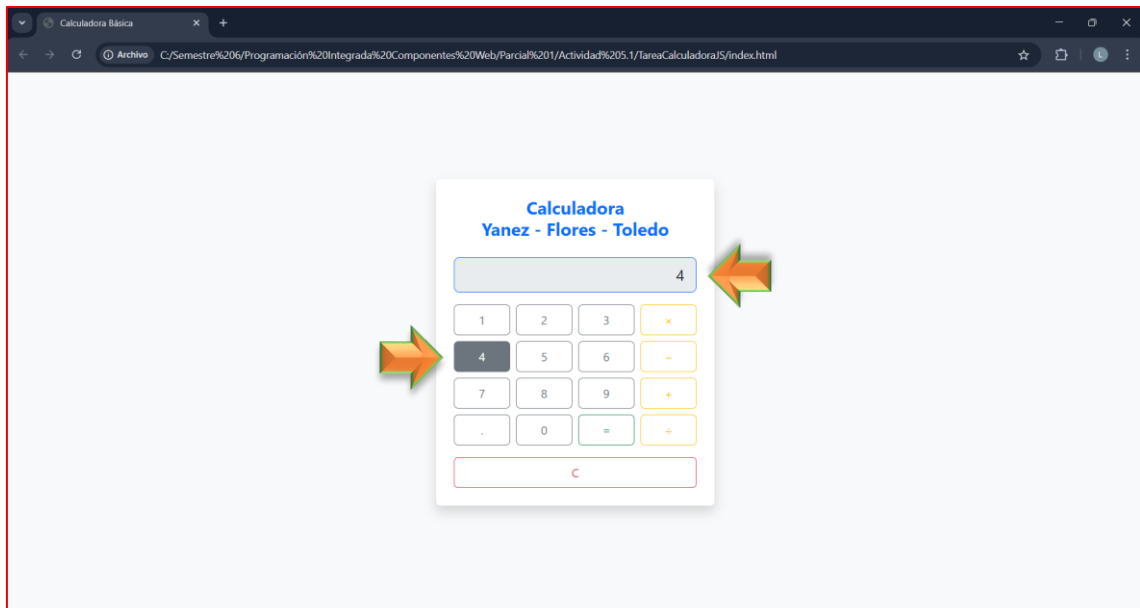
```
27 }
28
29 // Calcula el resultado
30 function calculate() {
31     if (operacion === null || display.value === '') return;
32
33     segundo_numero = parseFloat(display.value);
34     let result;
35
36     switch (operacion) {
37         case '+':
38             result = primer_numero + segundo_numero;
39             break;
40         case '-':
41             result = primer_numero - segundo_numero;
42             break;
43         case '*':
44             result = primer_numero * segundo_numero;
45             break;
46         case '/':
47             if (segundo_numero === 0) {
48                 alert("Error: División por cero");
49                 clearCalculator();
50                 return;
51             }
52             result = primer_numero / segundo_numero;
53             break;
54         default:
55             return;
56     }
57
58     // Mostrar solo 2 decimales
59     display.value = result.toFixed(2);
60     operationDisplay.textContent = '';
```

3. Proyecto Ejecutando

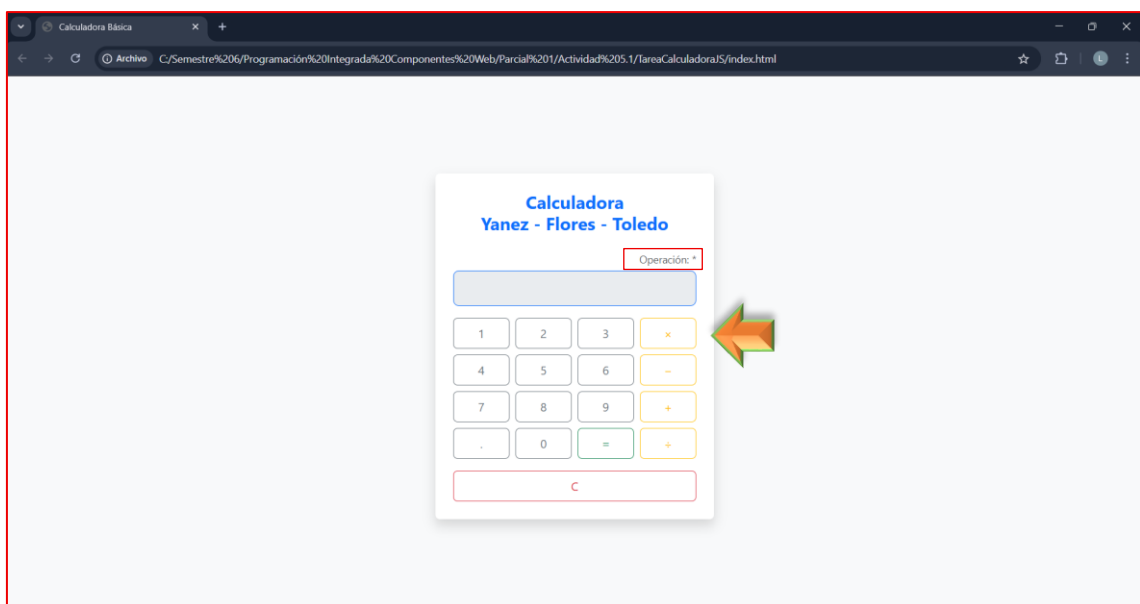
A continuación, vamos a presentar la ejecución del programa.



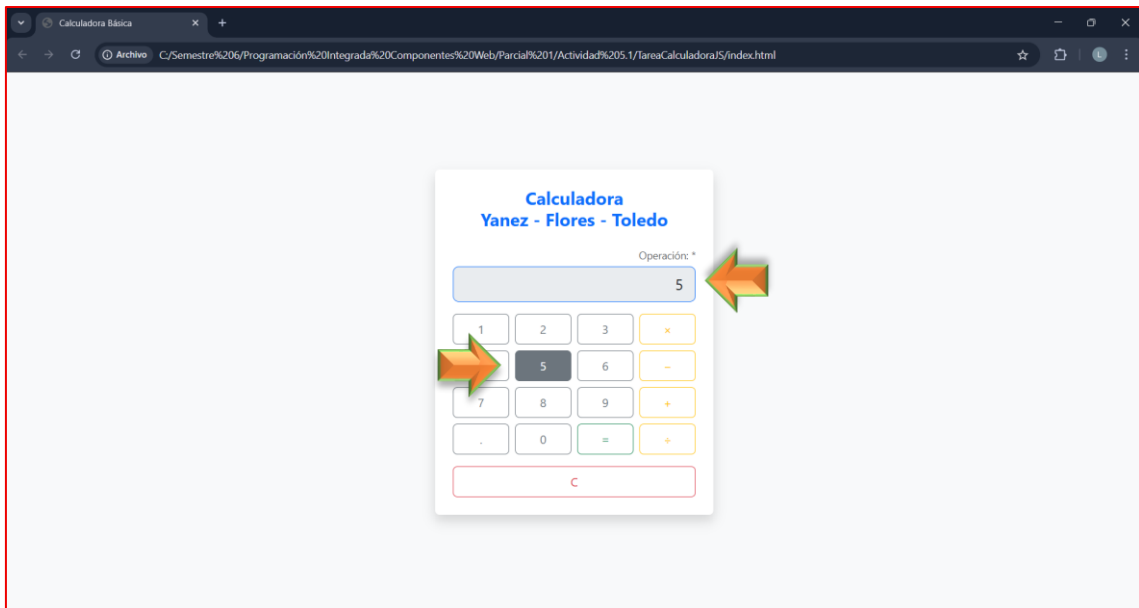
Procedemos a ingresar un primer número en la caja de texto.



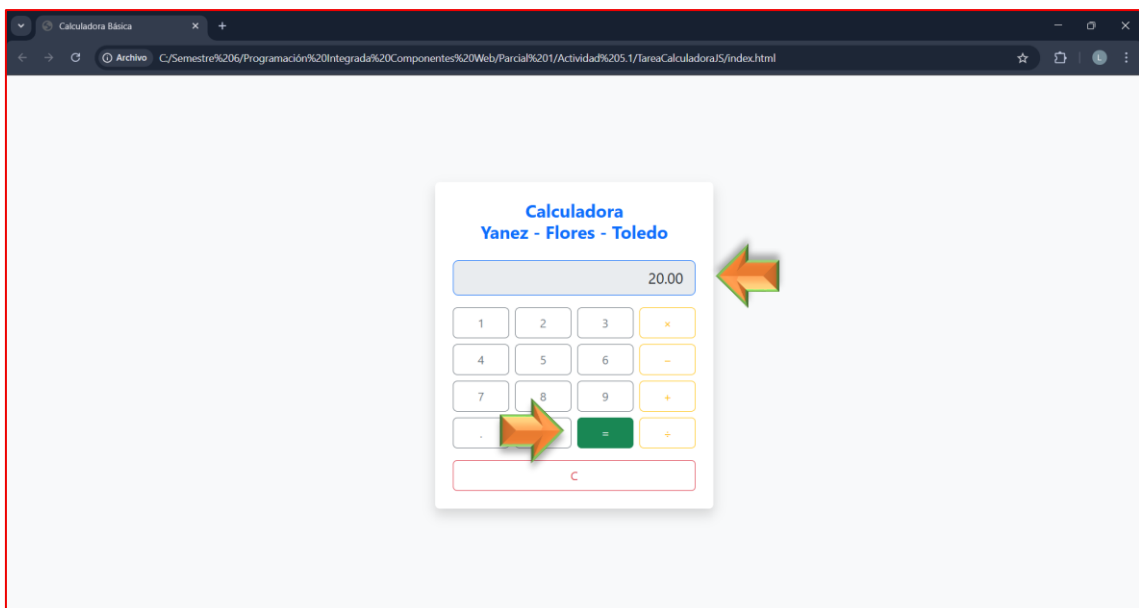
Una vez que ingresamos el número procedemos a seleccionar una operación mediante los botones, y se borra el cuadro de texto y se observa la operación elegida.



Una vez se elige la operación se procede a ingresar el siguiente número para efectuar la operación que se eligió.

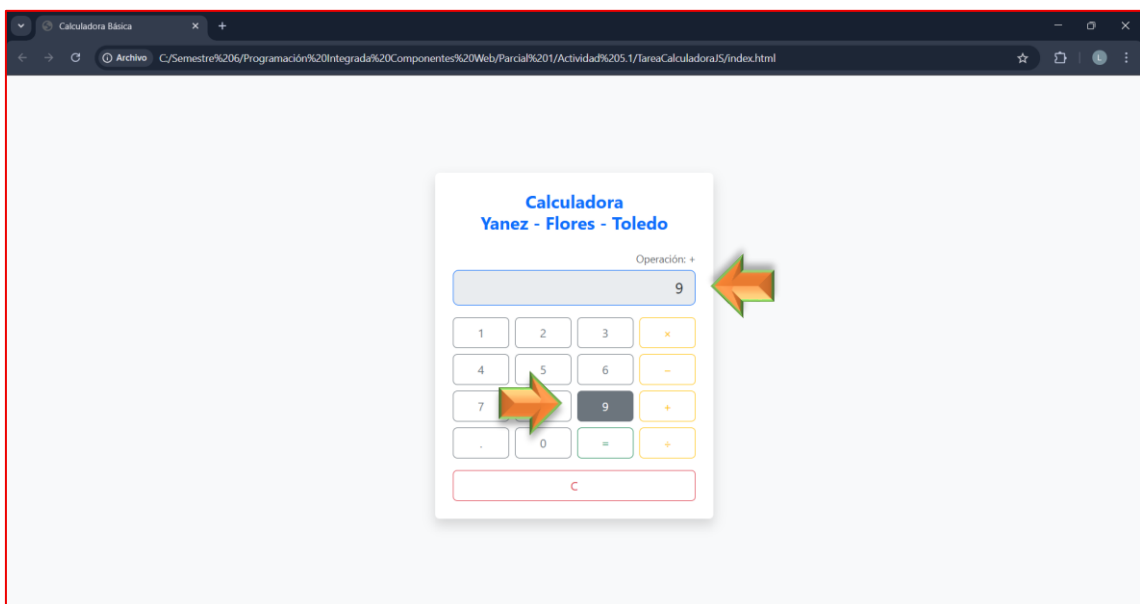
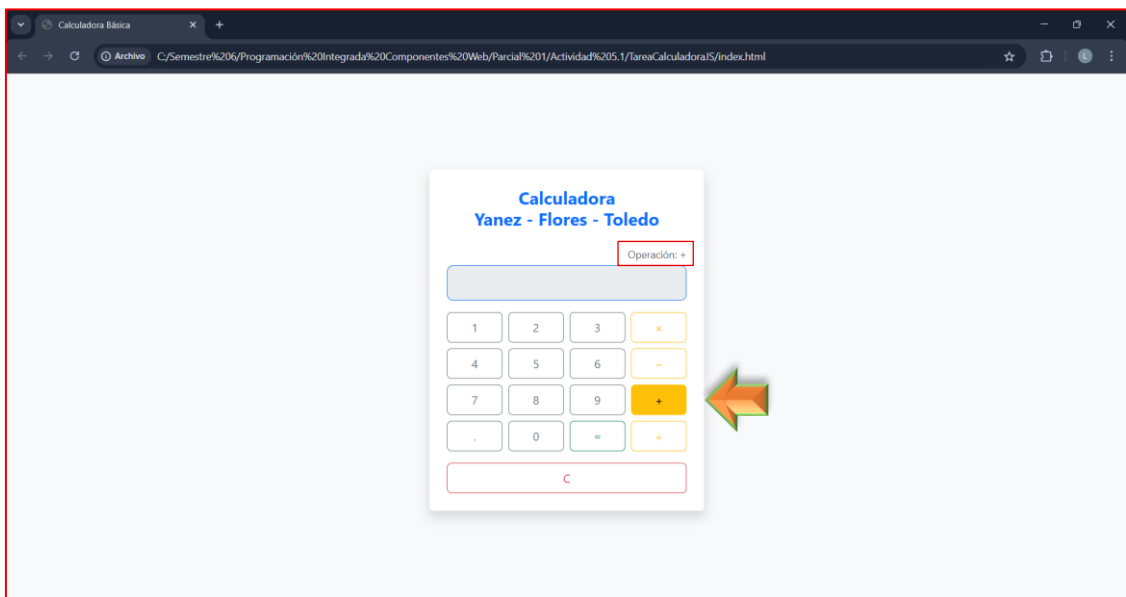
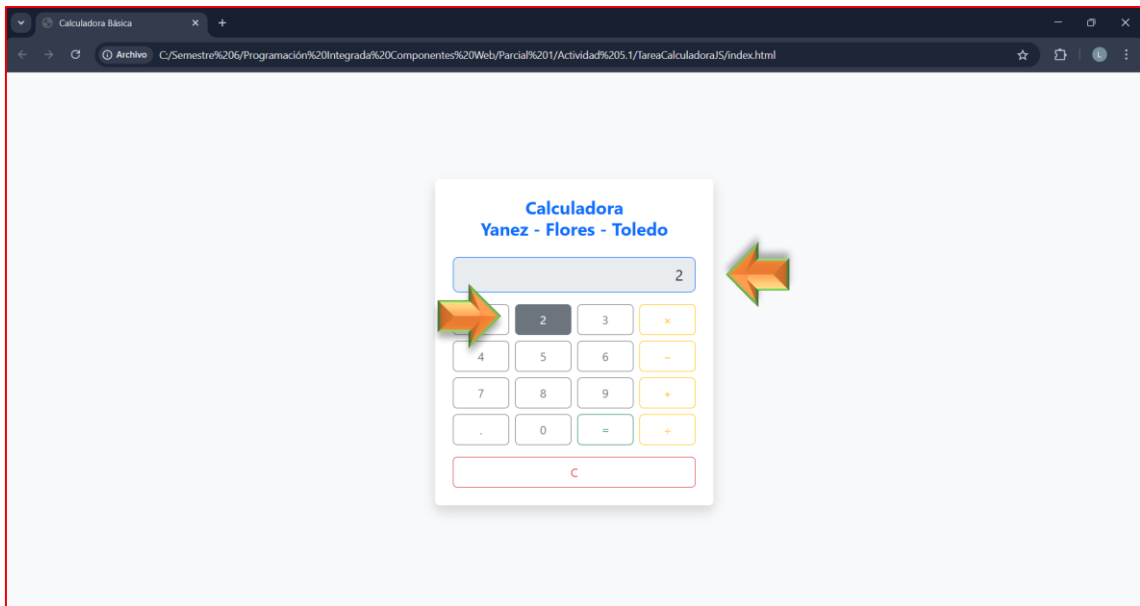


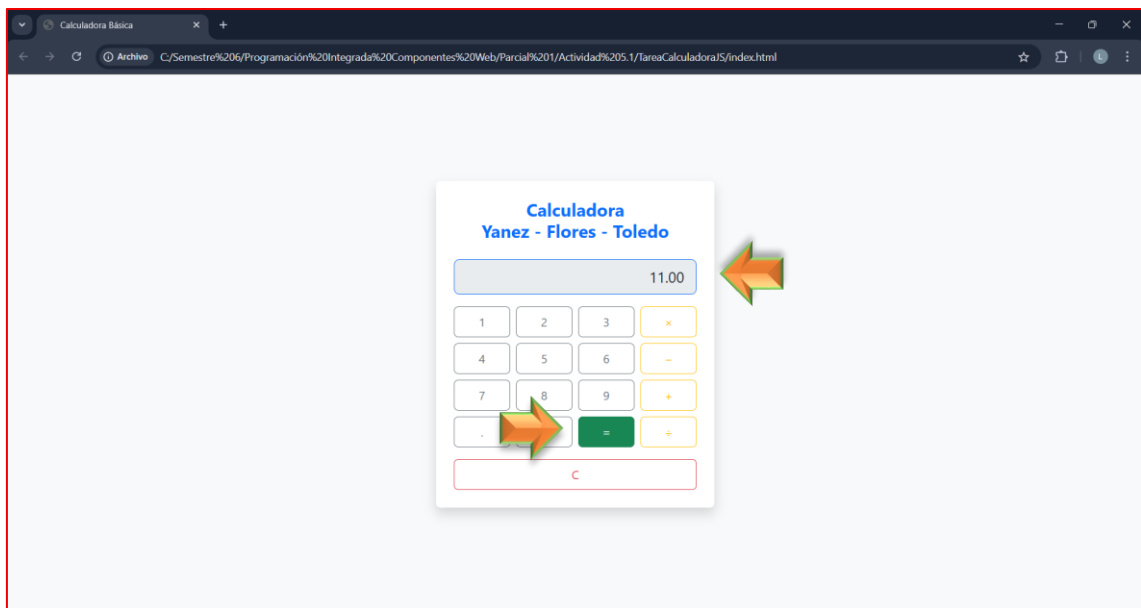
Para finalizar con la practica procedemos a seleccionar el botón igual para que muestre el resultado de la operación.



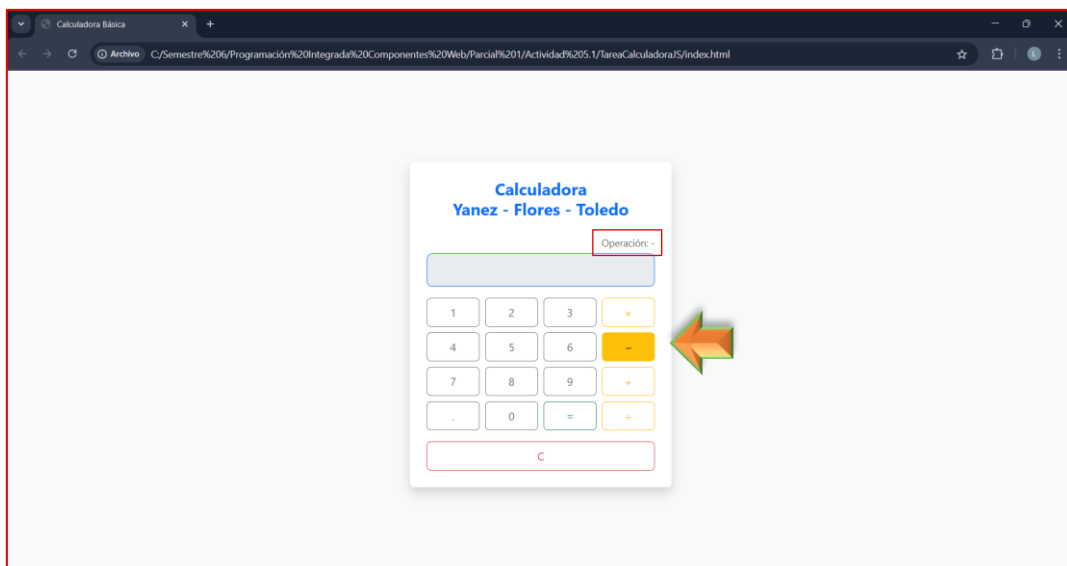
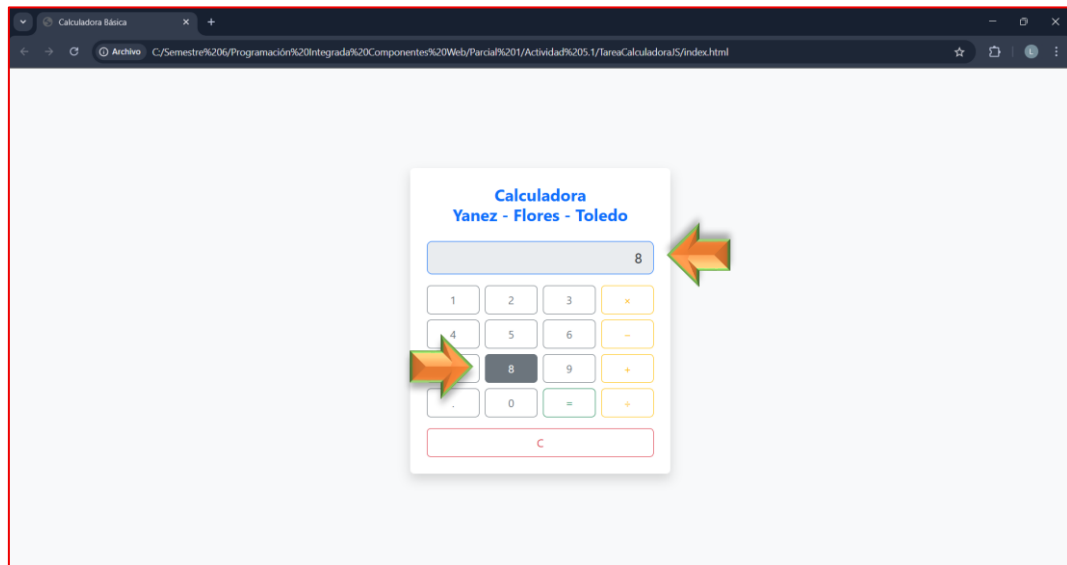
Ahora procedemos a realizar para las diferentes operaciones y visualizar su funcionamiento. En este caso lo hacemos:

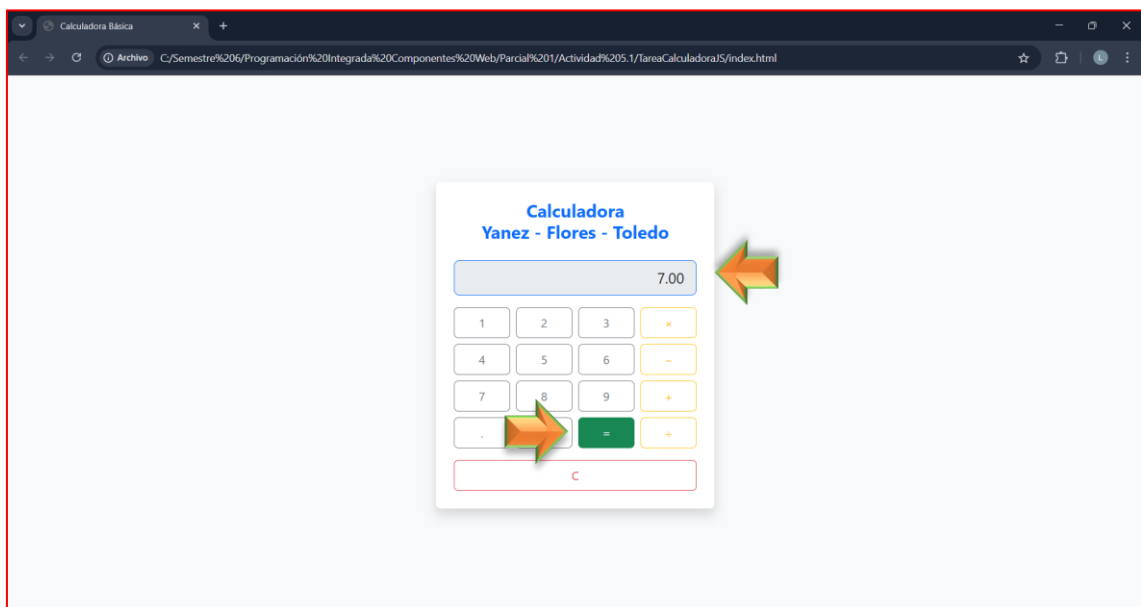
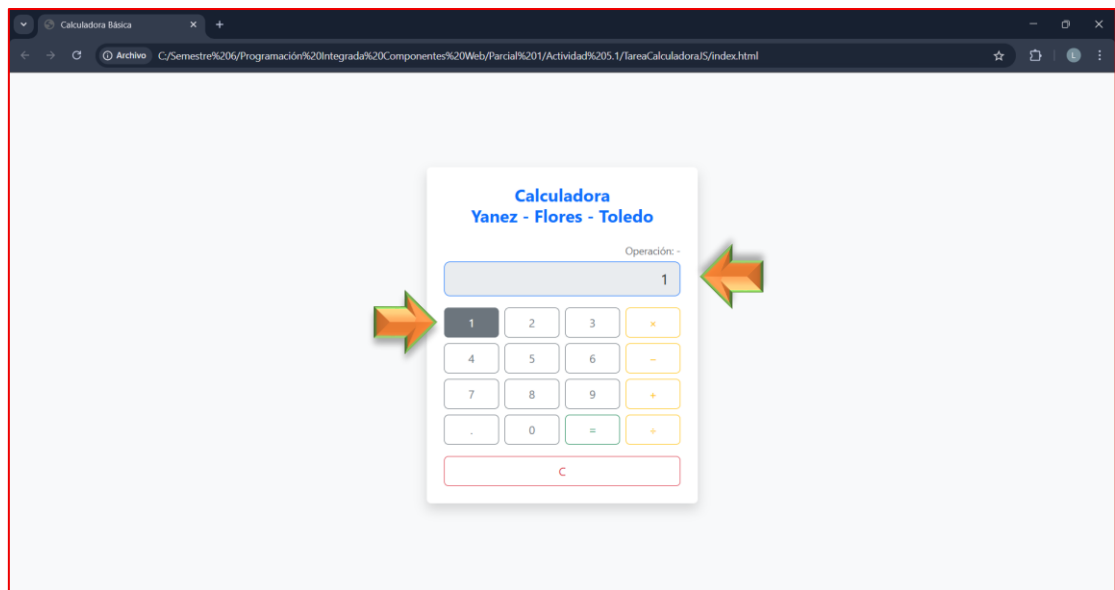
Para la suma.



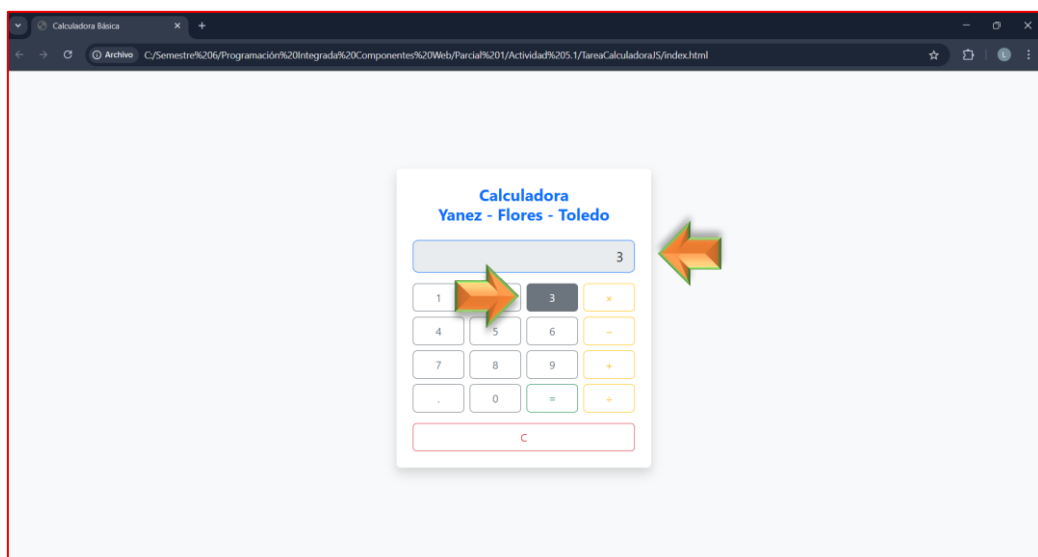


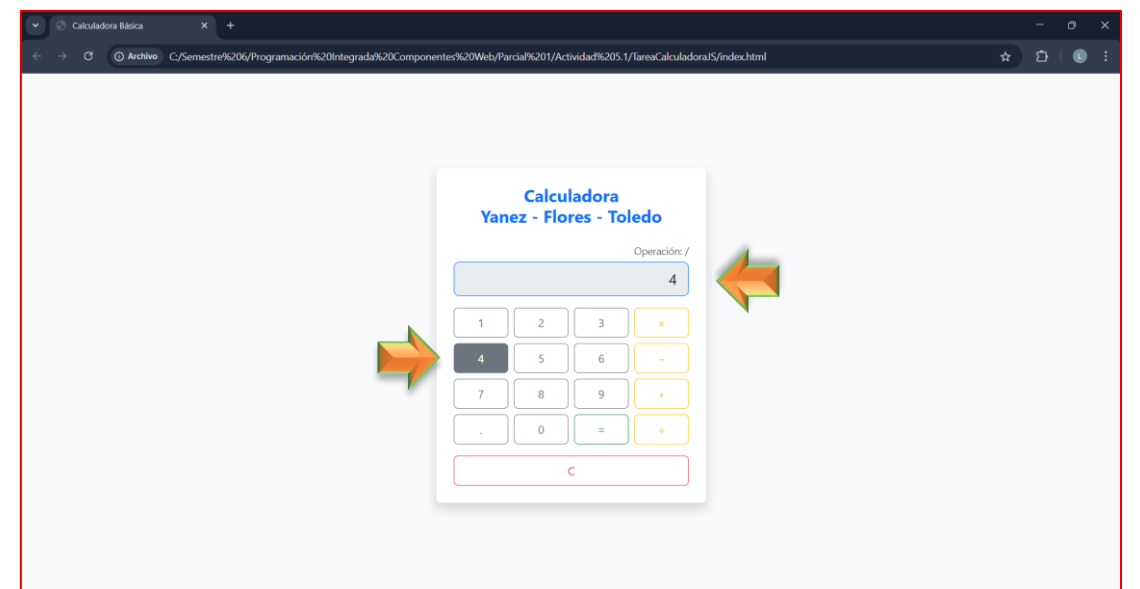
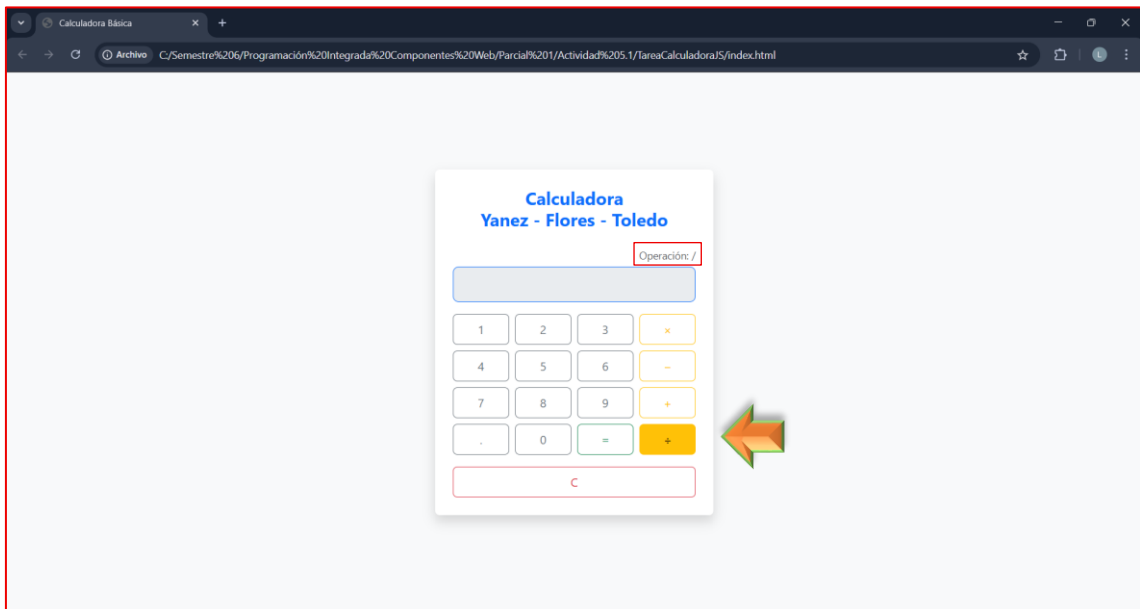
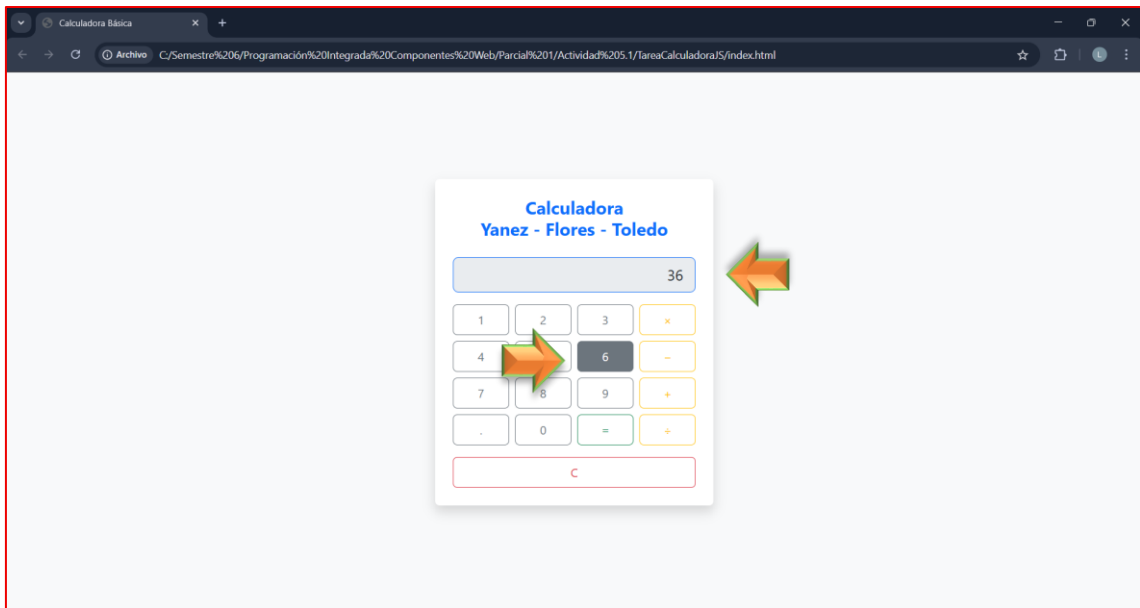
Para la resta.

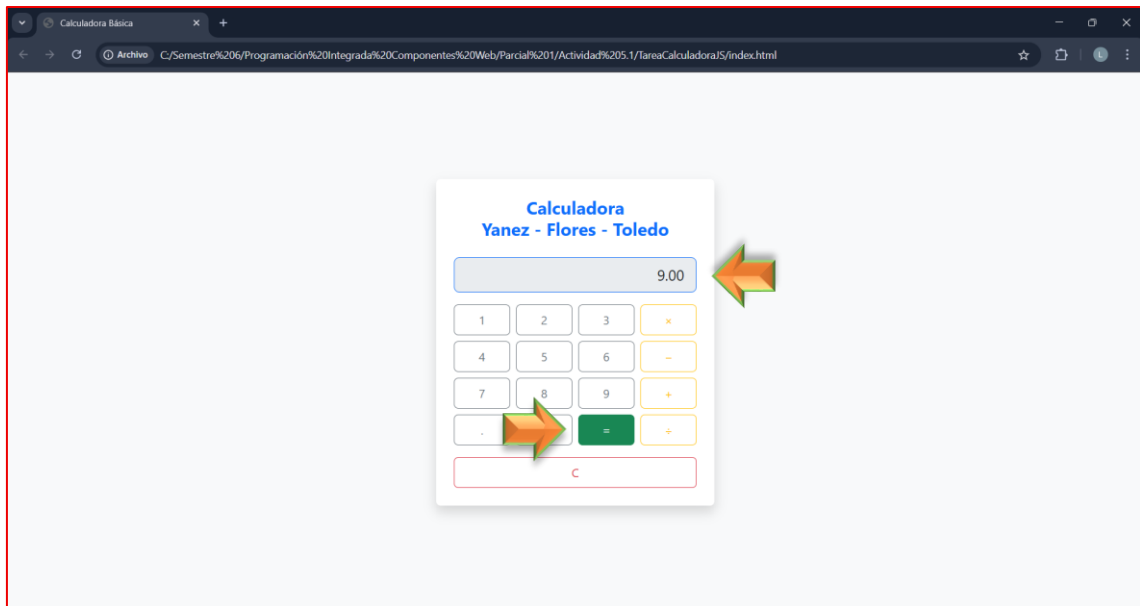




Para la división.







Como se puede visualizar una vez que se realizaron las operaciones se evidencia la funcionalidad de las operaciones y el cumplimiento con el código.

4. Explicación breve del funcionamiento del archivo index.html y el archivo js.

Estructura principal del archivo donde se define el idioma, título, los estilos externos de Bootstrap.

```
<> index.html X JS java.js
TareaCalculadoraJS > <> index.html > html > body.bg-light.d-flex.justify-content-center.align-items-center.vh-100 > div.container >
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="UTF-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6 <title>Calculadora Básica</title>
7
8 <!-- Configuración básica del documento, como (UTF-8), configuración para que sea responsivo,
9 el título de la página y enlaces a estilos externos Bootstrap 5.-->
10 <!-- Bootstrap 5 -->
11 <link
12 href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
13 rel="stylesheet"
14 />
15
16 <!-- Hoja de estilos personalizada -->
17 <link rel="stylesheet" href="estilo.css" />
18 </head>
19
```

Se definen contenedores para cada función y los elementos de los botones que ejecutan las diferentes funciones de JavaScript.

Adicional a esto, se validan los enlaces a archivos de JavaScript externos para la lógica de la actividad.

```
index.html X JS java.js
TareaCalculadoraJS > index.html > html > body.bg-light.d-flex.justify-content-center.align-items-center.vh-100 > div.container >
2 <html lang="es">
20 <body>
26 <div class="container" style="max-width: 400px">
29 <div class="bg-white shadow rounded p-4">
153 <div class="row g-2 mb-3">
185 </div>
186 </div>
187
188 <div class="row g-2">
189 <div class="col-12">
190 <!-- Botón para borrar la operación actual, con un evento onclick que
191 llama a la función clearCalculator(). -->
192 <button
193 class="btn btn-outline-danger w-100 py-2"
194 onclick="clearCalculator()"
195 >
196 C
197 </button>
198 </div>
199 </div>
200 </div>
201 </div>
202
203 <script src="java.js"></script>
204 <!-- Enlazamos el archivo JavaScript que contiene la lógica de la calculadora.-->
205 </body>
206 </html>
207
```

Ahora vamos a ver el archivo de JavaScript asociado al index para el correcto funcionamiento de la actividad. En el archivo encontramos lo que son las variables de los números ingresados y la operación seleccionada, además de mostrar la obtención de los elementos para mostrar los valores que se ingresan y la operación activa.

```
index.html X JS java.js X
TareaCalculadoraJS > JS java.js > ...
1 // En este archivo se configuró la lógica del funcionamiento de la calculadora. Se define
2 // lo que es la agregación de números, se seleccionan operaciones, se realiza el cálculo
3 // y se limpia la calculadora.
4
5 // Estas variables guardan los números ingresados y la operación seleccionada.
6 let primer_numero = null;
7 let segundo_numero = null;
8 let operacion = null;
9
10 // Obtiene elementos del HTML para mostrar los valores ingresados y la operación activa.
11 const display = document.getElementById('display');
12 const operationDisplay = document.getElementById('operationDisplay');
13
14 // Agrega número al display
15 function appendNumber(number) {
16   display.value += number;
17 }
18
19 // Define la operación a realizar
20 function chooseOperation(operation) {
21   if (display.value === '') return;
22
23   primer_numero = parseFloat(display.value);
24   operacion = operation;
25   display.value = '';
26   operationDisplay.textContent = `Operación: ${operation}`;
27 }
28
29 // Calcula el resultado
30 function calculate() {
31   if (operacion === null || display.value === '') return;
32
33   segundo_numero = parseFloat(display.value);
34   let result:
```

Conclusiones

- La integración de JavaScript puro con Bootstrap permite crear aplicaciones web funcionales y visualmente atractivas sin necesidad de frameworks complejos, facilitando el aprendizaje del manejo del DOM y los eventos.
- El uso del sistema de grillas de Bootstrap (row, col, gutters) contribuyó a una distribución ordenada y responsiva de los elementos, mejorando la experiencia de usuario en distintos dispositivos.
- La práctica de separar la lógica (JS), el estilo (CSS) y la estructura (HTML) refuerza una buena organización del código, permitiendo un mantenimiento más sencillo y fomentando buenas prácticas de desarrollo web.

Recomendaciones

- Agregar validaciones adicionales para mejorar la robustez de la calculadora, como evitar múltiples puntos decimales, bloquear letras o símbolos inválidos, y controlar errores inesperados.
- Optimizar el código JavaScript modularizando las funciones, lo cual mejora la legibilidad, facilita el mantenimiento y permite reutilización de código en proyectos futuros.
- Ampliar la funcionalidad de la calculadora incluyendo operaciones adicionales, como potencias o raíces, y agregar historial de operaciones, para seguir desarrollando habilidades con JavaScript y el DOM.