

Benemérita Universidad Autónoma de Puebla  
Facultad de Ciencias de la Computación

# DESARROLLO DE APLICACIONES WEB

---

*Examen 3: Calculadora*



Docente:  
Prof. Roberto Elvira Enríquez

Jesús Huerta Aguilar  
202041509

NRC: 12562  
Sección: 003

SEXTO SEMESTRE

## ACTIVIDAD 2:

# CALCULADORA

### ELEMENTOS DETECTADOS A SIMPLE VISTA

El diseño permite ingresar números del 0 al 9, lo que posibilita realizar operaciones matemáticas elementales como suma, resta, multiplicación y división. Una particularidad interesante es la inclusión de la posibilidad de usar decimales, aunque, siguiendo las especificaciones del documento de ejemplo, solo se permite un decimal por número.



El campo de texto ubicado en la parte superior de la calculadora es un componente crucial. Su función principal es mostrar tanto los números que voy ingresando como el resultado final de la operación que selecciono. Esto facilita el seguimiento de los cálculos y permite una revisión rápida de los datos introducidos.



En la parte inferior del diseño, destaca un botón rojo con la leyenda “borrar”. Su función es doble: por un lado, elimina cualquier carácter mostrado en el campo de texto de salida, y por otro lado, borra lógicamente tanto el valor actual como el anterior. Esta característica es esencial para poder comenzar cálculos nuevos desde cero, garantizando que los resultados futuros no se vean afectados por cálculos anteriores.



En cuanto al diseño de la interfaz, los números y los símbolos matemáticos están organizados en lo que parece ser una tabla de 4x4. Esta disposición puede ser estilizada con CSS para definir las columnas y filas de manera clara. Los números se disponen de manera ascendente, en grupos de tres, a lo largo del lado izquierdo, dejando la última fila exclusivamente para los símbolos matemáticos.



Cada botón, excepto el botón de borrar, presenta un color distintivo, lo que ayuda a diferenciar las funciones de manera visual y rápida. El campo de texto superior, con un diseño básico, muestra únicamente los números que están siendo operados en ese momento.

El botón de borrar, ubicado en la parte inferior, es tan ancho como la tabla que contiene los números y los símbolos matemáticos, y se destaca por su color rojo, aportando un contraste visual significativo con el resto de la interfaz.

Para el procesamiento lógico de las operaciones, es imprescindible un documento de JavaScript. En él se definirán tanto las operaciones matemáticas como las restricciones necesarias para el funcionamiento correcto de la calculadora. Este aspecto es fundamental para asegurar que la calculadora no solo sea fácil de usar, sino que también sea precisa y confiable en sus cálculos.

## DESARROLLO DEL CÓDIGO (LO MAS IMPORTANTE)

### *index.html*

**Estructura inicial y metadatos del documento HTML:** Incluye el tipo de documento (HTML5), establece el idioma de la página como español, y el bloque `<head>` contiene metadatos importantes como la codificación de caracteres (UTF-8), la configuración de visualización para dispositivos móviles, el título de la página, y enlaces a hojas de estilo CSS externas, incluyendo Bootstrap y un archivo de estilo personalizado.

```
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</title>
7. <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boot
  strap/4.3.1/css/bootstrap.min.css">
8. <link rel="stylesheet" href="style.css">
9. </head>
```

**Cuerpo del documento y estructura visual:** El `<body>` contiene el contenido visible, incluyendo un encabezado con el título de la calculadora, y un div principal que actúa como contenedor de la calculadora. Dentro de este contenedor, hay un input de texto deshabilitado para mostrar los números y resultados, seguido por varias filas, cada una conteniendo botones. Estos botones representan dígitos, operadores matemáticos y una función de borrado, y están vinculados a funciones JavaScript.

```
10. <body>
11. <header class="text-center">
12.   <h1>Calculadora FCC</h1>
13. </header>
14. <div class="calculator">
15.
16.   <br><br>
17.   <input type="text" class="form-control calc-display" id="calc-
    display" disabled>
18.   <div class="row">
19.     <button class="btn btn-
      secondary col button" onclick="appendNumber('7')">7</button>
20.     <button class="btn btn-
      secondary col button" onclick="appendNumber('8')">8</button>
21.     <button class="btn btn-
      secondary col button" onclick="appendNumber('9')">9</button>
22.     <button class="btn btn-
      primary col button" onclick="setOperation('/')">/</button>
23.   </div>
24.   <div class="row">
25.     <button class="btn btn-
      secondary col button" onclick="appendNumber('4')">4</button>
```

```
26.     <button class="btn btn-  
secondary col button" onclick="appendNumber('5') ">5</button>  
27.     <button class="btn btn-  
secondary col button" onclick="appendNumber('6') ">6</button>  
28.     <button class="btn btn-  
primary col button" onclick="setOperation('*') ">x</button>  
29. </div>  
30. <div class="row">  
31.     <button class="btn btn-  
secondary col button" onclick="appendNumber('1') ">1</button>  
32.     <button class="btn btn-  
secondary col button" onclick="appendNumber('2') ">2</button>  
33.     <button class="btn btn-  
secondary col button" onclick="appendNumber('3') ">3</button>  
34.     <button class="btn btn-primary col button" onclick="setOperation('-') ">-  
</button>  
35. </div>  
36. <div class="row">  
37.     <button class="btn btn-  
secondary col button" onclick="appendNumber('0') ">0</button>  
38.     <button class="btn btn-  
secondary col button" onclick="appendNumber('.') ">.</button>  
39.     <button class="btn btn-  
secondary col button" onclick="calculate() ">=</button>  
40.  
41.     <button class="btn btn-  
primary col button" onclick="setOperation('+') ">+</button>  
42. </div>  
43. <div class="row">  
44.     <button class="btn btn-  
danger col button" onclick="clearDisplay() ">BORRAR</button>  
45. </div>  
46. <br><br>  
47. </div>
```

**Pie de página y referencias:** El footer incluye una imagen representativa de la BUAP (Benemérita Universidad Autónoma de Puebla), así como información de contacto y referencias a la facultad y al desarrollador de la aplicación.

```
48.     <footer class="row">  
49.     <div class="col-md-6">  
50.           
51.     </div>  
52.     <address class="text-right col-md-6">  
53.         <a href="mailto:jesus.huertakg@gmail.com">Jesus Huerta Aguila  
r</a>  
54.         - Desarrollo de Aplicaciones Web<br>  
55.         Benemérita Universidad Autónoma de Puebla<br>  
56.         Facultad de Ciencias de la Computación<br>  
57.     </address>  
58. </footer>
```

**Enlace a la funcionalidad de JavaScript:** Al final del `<body>`, se encuentra un enlace a un archivo JavaScript externo (`script.js`), que se presume contiene la lógica para manejar las interacciones de los botones y las operaciones de la calculadora.

| 59.        `<script src="script.js"></script>`

## *style.css*

**.calculator:** Este bloque de estilo se aplica a cualquier elemento con la clase **calculator**. Establece un ancho de 300 píxeles para el contenedor de la calculadora, lo centra horizontalmente en la página con un margen automático y agrega un padding (relleno) de 10 píxeles en todos los lados. Esto define la dimensión y la posición central del contenedor principal de la calculadora en la página web.

```
1. .calculator {  
2.   width: 300px;  
3.   margin: 50px auto;  
4.   padding: 10px;  
5. }
```

**.button:** Este bloque se aplica a elementos con la clase **button**. Establece un ancho del 1% (lo cual podría ser demasiado pequeño y podría requerir ajustes según el diseño de la página), una altura de 60 píxeles, y un tamaño de fuente de 24 píxeles, lo que hace que los botones sean grandes y fácilmente legibles. Además, se agrega un margen de 2 píxeles para espaciar ligeramente los botones entre sí.

```
7. .button {  
8.   width: 1%;  
9.   height: 60px;  
10.  font-size: 24px;  
11.  margin: 2px;  
12. }
```

**.calc-display:** Este estilo se aplica a elementos con la clase **calc-display**, generalmente utilizada para la pantalla de la calculadora. Se establece una fuente **Consolas**, que es una fuente monoespaciada adecuada para mostrar números y operaciones matemáticas. El ancho se ajusta al 100% para que ocupe todo el ancho del contenedor de la calculadora, con un tamaño de fuente de 24 píxeles para una lectura clara y un margen inferior de 5 píxeles para separarlo de otros elementos.

```
14. .calc-display {  
15.   font-family: "Consolas";  
16.   width: 100%;  
17.   margin-bottom: 5px;  
18.   font-size: 24px;  
19. }
```

**header:** Define el estilo para los elementos **<header>** de la página. Establece un color de fondo **#003b5c** (un tono oscuro de azul) y un color de texto blanco, con un relleno de 20 píxeles en la parte superior e inferior para dar espacio al contenido del encabezado. Este estilo es probablemente para el encabezado de la página donde se encuentra la calculadora, proporcionando un aspecto distintivo y coherente.

```
21.     header {  
22.   background-color: #003b5c;  
23.   color: white;  
24.   padding: 20px 0;  
25. }  
26.
```

**footer:** Aplica estilos a los elementos **<footer>**. Utiliza el mismo color de fondo que el header (**#003b5c**), pero cambia el color del texto a **#00b5e2**, un tono claro de azul cian. El relleno de 10 píxeles proporciona un espacio alrededor del contenido del pie de página. Este estilo asegura que el pie de página coincida visualmente con el encabezado, manteniendo la coherencia del diseño en toda la página.

```
27.     footer{  
28.   background-color: #003b5c;  
29.   color: #00b5e2;  
30.   padding: 10px;  
31. }
```



## *script.js*

**Variables de Estado:** **operacionActual** almacena el operador de la operación actual (+, -, \*, /). **valorActual** mantiene el número que se está ingresando actualmente. **valorAnterior** guarda el número ingresado anteriormente antes de elegir una operación.

```
1. let operacionActual = null;
2. let valorActual = '';
3. let valorAnterior = '';
```

**Función appendNumber:** Esta función evita que se ingresen múltiples puntos decimales en el mismo número. Concatena el número o punto decimal presionado al **valorActual** y actualiza la pantalla de la calculadora.

```
5. function appendNumber(number) {
6.   // Evitar multiples decimales
7.   if (number === '.' && valorActual.includes('.')) return;
8.   valorActual += number;
9.   updateDisplay();
10. }
```

**Función setOperation:** Previene la asignación de una operación si no hay un número actual. Si ya hay una operación en curso, realiza esa operación primero. Establece la nueva operación, mueve el **valorActual** al **valorAnterior**, y limpia **valorActual**.

```
12. function setOperation(operator) {
13.   if (valorActual === '') return;
14.   if (valorAnterior !== '' && operacionActual !== null) {
15.     calculate();
16.   }
17.   operacionActual = operator;
18.   valorAnterior = valorActual;
19.   valorActual = '';
20. }
```

**Función calculate:** Realiza el cálculo basado en la operación almacenada y los valores actuales y anteriores. Convierte los valores de string a números flotantes y maneja las operaciones matemáticas. También verifica errores como la división por cero y actualiza la pantalla después del cálculo.

```
22. function calculate() {
23.   let calculation;
24.   const anterior = parseFloat(valorAnterior);
25.   const actual = parseFloat(valorActual);
26.   if (isNaN(anterior) || isNaN(actual)) return;
27.   switch (operacionActual) {
```

```
28.     case '+':
29.         calculation = anterior + actual;
30.         break;
31.     case '-':
32.         calculation = anterior - actual;
33.         break;
34.     case '*':
35.         calculation = anterior * actual;
36.         break;
37.     case '/':
38.         if (actual === 0) {
39.             alert("No puedes dividir entre cero");
40.             return;
41.         }
42.         calculation = anterior / actual;
43.         break;
44.     default:
45.         return;
46. }
47. valorActual = calculation.toString();
48. operacionActual = null;
49. valorAnterior = '';
50. updateDisplay();
51. }
```

**Función clearDisplay:** Resetea todas las variables de estado y limpia la pantalla de la calculadora.

```
53.     function clearDisplay() {
54.         valorActual = '';
55.         valorAnterior = '';
56.         operacionActual = null;
57.         updateDisplay();
58.     }
59.
```

**Función updateDisplay:** Actualiza la pantalla de la calculadora con el valor actual, reflejando los cambios y resultados de las operaciones.

```
60.     function updateDisplay() {
61.         document.getElementById('calc-display').value = valorActual;
62.     }
```