



UJED

Universidad Juárez
del Estado de Durango



Universidad Juárez del Estado de Durango

Facultad de Ingeniería Ciencias y Arquitectura

Ingeniería en Tecnologías Computacionales

Programación Web

Reporte de Práctica 22

Diego Rea Morales

Ing. Fabian Gallegos Gutiérrez

7 de noviembre de 2025

Índice

Contenido	3
• Explicación del código JavaScript	4
Conclusiones	6

Contenido

En la práctica de esta ocasión, vimos a realizar una aplicación web, la cual, esta tome dos valores y las sume, reste, multiplique y/o divida dos números ingresados por el usuario. Para ello, lo haremos en un proyecto de React JS.

```
1 import React, { useState } from "react";
2 function App() {
3   const [num1, setNum1] = useState("");
4   const [num2, setNum2] = useState("");
5   const [result, setResult] = useState("");
6   const [tipoOp, setTipoOp] = useState("");
7   const handleNum1Change = (event) => {
8     setNum1(event.target.value);
9   };
10  const handleNum2Change = (event) => {
11    setNum2(event.target.value);
12  };
13  const handleSubmit = (event) => {
14    event.preventDefault();
15    const sum = parseInt(num1) + parseInt(num2);
16    setResult(sum);
17  };
18  const handleRes = (event) => {
19    event.preventDefault();
20    const sum = parseInt(num1) - parseInt(num2);
21    setResult(sum);
22  };
23  const handleMult = (event) => {
24    event.preventDefault();
25    const sum = parseInt(num1) * parseInt(num2);
26    setResult(sum);
27  };
28  const handleDiv = (event) => {
29    event.preventDefault();
30    const sum = parseInt(num1) / parseInt(num2);
31    setResult(sum);
32  };
33
34  return (
35    <div>
36      <h1>Sumar dos variables</h1>
37      <form onSubmit={handleSubmit}>
38        <input
39          type="number"
40          placeholder="Ingrese el primer número"
41          value={num1}
42          onChange={handleNum1Change}
43          required
44        />
45        <input
46          type="number"
47          placeholder="Ingrese el segundo número"
48          value={num2}
49          onChange={handleNum2Change}
50          required
51        />
52        <button className="btn-op" type="submit">Sumar</button>
53        <button className="btn-op" onClick={handleRes}>Restar</button>
54        <button className="btn-op" onClick={handleMult}>Multiplicar</button>
55        <button className="btn-op" onClick={handleDiv}>Dividir</button>
56      </form>
57      {result && <p>El resultado es: {result}</p>}
58    </div>
59  );
60 }
61 export default App;
```

```
1 .App {
2   text-align: center;
3 }
4 .App-logo {
5   height: 40vmin;
6   pointer-events: none;
7 }
8 @media (prefers-reduced-motion: no-preference) {
9   .App-logo {
10     animation: App-logo-spin infinite 20s linear;
11   }
12 }
13 .App-header {
14   background-color: #282c34;
15   min-height: 100vh;
16   display: flex;
17   flex-direction: column;
18   align-items: center;
19   justify-content: center;
20   font-size: calc(10px + 2vmin);
21   color: white;
22 }
23 .App-link {
24   color: #61dafb;
25 }
26 @keyframes App-Logo-spin {
27   from {
28     transform: rotate(0deg);
29   }
30   to {
31     transform: rotate(360deg);
32   }
33 }
```

Explicación del código JavaScript

```
const [num1, setNum1] = useState("");
const [num2, setNum2] = useState("")
```

Primero para registrar el que es lo que el usuario ingresó, lo tenemos que registrar en una variable, para ello declaramos los *hooks*, `[num1, setNum1]`, `num1` será el valor que vamos a usar, mientras que `setNum1`, es para asignarle el valor.

```
const handleNum1Change = (event) => {
  setNum1(event.target.value);
};

const handleNum2Change = (event) => {
  setNum2(event.target.value);
};
```

Para asignarle el valor, dependemos de eventos, estos se ejecutarán cuando se ingrese algo en el input correspondiente.

```
onChange={handleNum1Change}
```

Esto es controlado por el atributo del input, que cuando detecte un cambio, este cambio, es decir, lo que se ha ingresado, se mandará el valor para que sea “Seteado” en la variable.

Después cuando se presione que botón de la operación que se quiera realizar, este ejecutara el evento correspondiente.

```
<button className="btn-op" type="submit">Sumar</button>
<button className="btn-op" onClick={handleRes}>Restar</button>
<button className="btn-op" onClick={handleMult}>Multiplicar</button>
<button className="btn-op" onClick={handleDiv}>Dividir</button>
```

El evento que se ejecutará será llamado por cada uno de los botones, estos son iguales, pero en lo único que cambia sería en la operación que van a realizar.

```
const handleSubmit = (event) => {
  event.preventDefault();
  const sum = parseInt(num1) + parseInt(num2);
  setResult(sum);
};

const handleRes = (event) => {
  event.preventDefault();
  const sum = parseInt(num1) - parseInt(num2);
  setResult(sum);
};

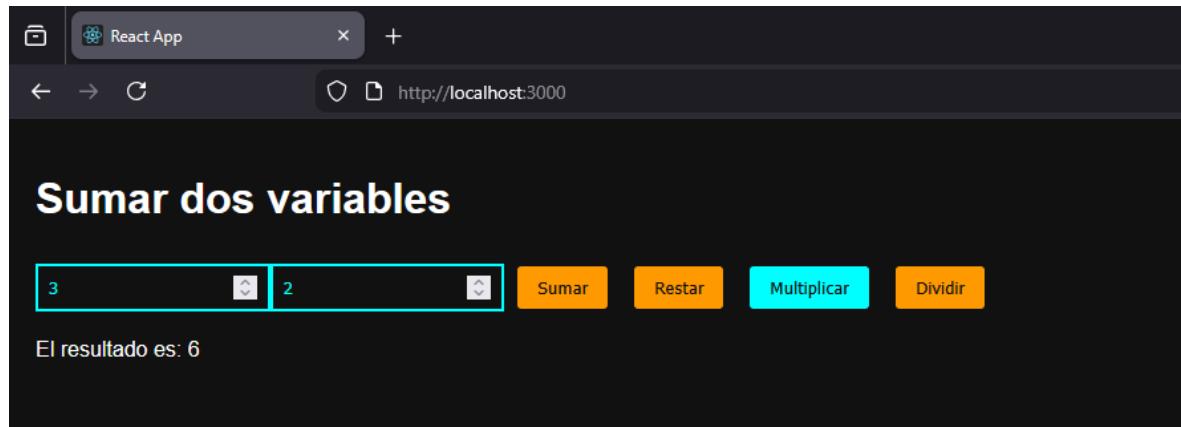
const handleMult = (event) => {
  event.preventDefault();
  const sum = parseInt(num1) * parseInt(num2);
  setResult(sum);
};
```

```
};

const handleDiv = (event) => {
  event.preventDefault();
  const sum = parseInt(num1) / parseInt(num2);
  setResult(sum);
};
```

Cuando la operación se haya realizado, este resultado se “Seteará”, para después mostrarse en pantalla.

```
{result && <p>El resultado es: {result}</p>}
```



Conclusiones

La práctica nos ayudó en la utilización de los eventos y valores en React, usando en este caso una aplicación web la cual devolvía un valor en base a las operaciones aritméticas básicas.