

Este código en JavaScript implementa una clase llamada `Matriz` que realiza diversas operaciones matriciales. Aquí tienes una explicación paso a paso:

### 1. Definición de la Clase `Matriz`:

```
class Matriz {  
  constructor(numeroMatriz) {  
    this.numeroMatriz = numeroMatriz;  
    this.matriz = this.crearMatriz();  
  }  
}
```

La clase `Matriz` tiene un constructor que toma un parámetro `numeroMatriz` y crea una matriz asociada a ese número utilizando el método `crearMatriz`.

### 2. Método `crearMatriz`:

```
crearMatriz() {  
  console.log(`Ingresa los valores de la matriz ${this.numeroMatriz}`);  
  const matriz = [];  
  for(let i = 0; i < 2; i++) {  
    matriz[i] = [];  
    for(let x = 0; x < 2; x++) {  
      const valor = parseFloat(prompt(`Ingresa el valor de la posición ${i + 1}, ${x + 1}`));  
      matriz[i][x] = valor;  
    }  
  }  
  return matriz;  
}
```

Este método solicita al usuario ingresar los valores para cada posición de la matriz (asumiendo matrices de 2x2). Los valores se almacenan en la matriz y se retorna.

### 3. Métodos para Operaciones Matriciales (`sumarMatriz`, `restarMatrices`, `multiplicarMatrices`, `dividirMatrices`):

```

    sumarMatriz(otraMatriz) {
        const resultado = [];
        for(let i = 0; i < 2; i++) {
            resultado[i] = [];
            for(let x = 0; x < 2; x++) {
                resultado[i][x] = this.matriz[i][x] + otraMatriz.matriz[i][x];
            }
        }
        return resultado;
    }

    restarMatrices(otraMatriz) {
        const resultado = [];
        for(let i = 0; i < 2; i++) {
            resultado[i] = [];
            for(let x = 0; x < 2; x++) {
                resultado[i][x] = this.matriz[i][x] - otraMatriz.matriz[i][x];
            }
        }
        return resultado;
    }

    multiplicarMatrices(otraMatriz) {
        const resultado = [];
        for( let i = 0; i < 2; i++) {
            resultado[i] = [];
            for(let x = 0; x < 2; x++) {
                resultado[i][x] = this.matriz[i][x] * otraMatriz.matriz[i][x];
            }
        }
        return resultado;
    }

    dividirMatrices(otraMatriz) {
        const resultado = [];

```

Cada uno de estos métodos toma otra matriz como parámetro y realiza la operación correspondiente entre la matriz actual y la matriz proporcionada. El resultado se almacena en una nueva matriz y se devuelve.

#### 4. Instanciación de Objetos `Matriz`:

```

const matriz1 = new Matriz(1);
const matriz2 = new Matriz(2);

```

Se crean dos objetos `Matriz` llamados `matriz1` y `matriz2`, cada uno solicitando valores al usuario para crear su matriz.

## 5. Operaciones entre Matrices y Resultados:

```
console.log(matriz1);  
console.log(matriz2);  
  
document.getElementById("la suma es:");  
console.table(suma);  
console.log("La resta es:");  
console.table(resta);  
console.log("La multiplicacion es:");  
console.table(multiplicar);  
console.log("la division es:");  
console.table(dividir);
```

Se realizan operaciones entre las dos matrices creadas y se almacenan los resultados.

## 6. Impresión de Resultados en la Consola y en una Página HTML:

```
console.log(matriz1);  
console.log(matriz2);  
  
document.getElementById("la suma es:");  
console.table(suma);  
console.log("La resta es:");  
console.table(resta);  
console.log("La multiplicacion es:");  
console.table(multiplicar);  
console.log("la division es:");  
console.table(dividir);
```

Se imprimen las matrices originales (`matriz1` y `matriz2`) y los resultados de las operaciones en la consola.