

Practica semana 08

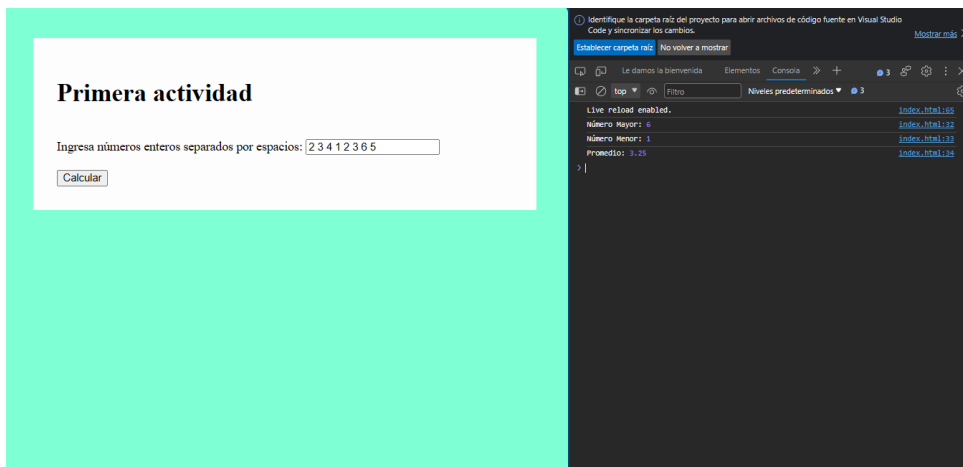
Instrucciones:

Escriba el código correspondiente a cada uno de los siguientes escenarios, esto se deben visualizar en la consola.

1.- Escriba una aplicación que tenga la siguiente funcionalidad:

- Defina un arreglo de números enteros (cantidad de elementos y valores de elección libre)
- Recorrer el arreglo y obtener el número mayor, el número menor y el promedio de los valores de todos los elementos.
- Describe de manera detallada el proceso seguido para la creación de la aplicación. Debe quedar claro el proceso de análisis y determinación de requerimientos, así como la manera como se utilizan los arreglos para resolver estos requerimientos.
- sube el código fuente de la aplicación a tu repositorio Github, y a tu documento agrega imágenes de las pantallas de la ejecución de la aplicación.

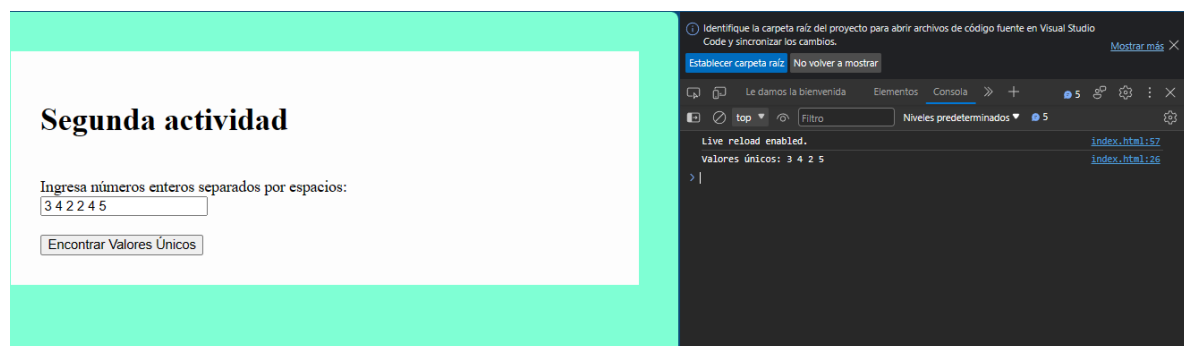
```
index.html x
prac1 > index.html > html > body > script > calcularNumeros
1 <!DOCTYPE html>
2 <html style="background-color: #aquamarine;">
3 <head>
4   <title>Numeros Mayor y Menor</title>
5 </head>
6 <body style="padding: 30px; margin: 40px; background-color: #rgb(253, 253, 253);">
7   <h1>Primera actividad</h1><br>
8   <label for="numeros">Ingresa números enteros separados por espacios:</label>
9   <input type="text" id="numeros"><br>
10  <button onclick="calcularNumeros()">Calcular</button>
11
12  <script>
13    function calcularNumeros() {
14      // Obtén el valor del input y divídelo en un arreglo de números
15      const inputElement = document.getElementById('numeros');
16      const numerosStr = inputElement.value;
17      const numeros = numerosStr.split(' ').map(Number);
18
19      // Verifica si todos los elementos ingresados son números
20      if (numeros.some(isNaN)) {
21        console.log('Por favor, ingresa números enteros válidos.');
```



2.- Escriba una aplicación que tenga la siguiente funcionalidad:

- Defina un arreglo de números enteros (cantidad de elementos y valores de elección libre).
- Mostrar un listado con los valores que son únicos en el arreglo.
- Describe de manera detallada el proceso seguido para la creación de la aplicación. Debe quedar claro el proceso de análisis y determinación de requerimientos, así como la manera como se utilizan los arreglos para resolver estos requerimientos.
- sube el código fuente de la aplicación a tu repositorio Github, y a tu documento agrega imágenes de las pantallas de la ejecución de la aplicación.

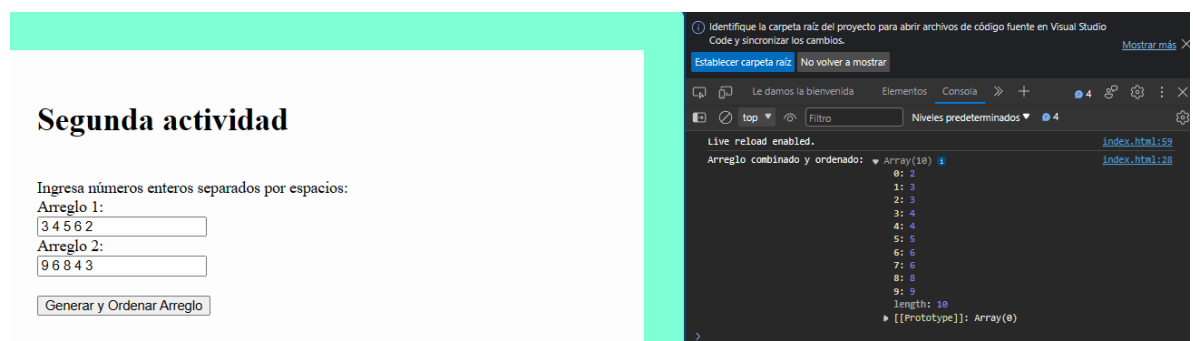
```
index.html x
prac 2 > index.html > html > body > br
1 <!DOCTYPE html>
2 <html style="background-color: aquamarine;">
3 <head>
4   <title>Valores Únicos</title>
5 </head>
6 <body style="padding:30px; margin: 40px;background-color: rgb(253, 253, 253);">
7   <h1>Segunda actividad</h1><br>
8   <label for="numeros">Ingresa números enteros separados por espacios:</label><br>
9   <input type="text" id="inputArray"><br><br>
10  <button onclick="findUniqueValues()">Encontrar Valores Únicos</button>
11
12  <script>
13    function findUniqueValues() {
14      const inputArray = document.getElementById("inputArray").value;
15
16      // Convertir la cadena de entrada en un arreglo de números
17      const numbers = inputArray.split(" ").map(Number);
18
19      // Utilizar un conjunto (Set) para encontrar valores únicos
20      const uniqueValues = new Set(numbers);
21
22      // Convertir el conjunto de valores únicos en un arreglo
23      const uniqueArray = [...uniqueValues];
24
25      // Mostrar los valores únicos en la consola sin comas
26      console.log("Valores únicos:", uniqueArray.join(" "));
27    }
28  </script>
29 </body>
30 </html>
31
```



3.- Escriba una aplicación que tenga la siguiente funcionalidad:

- Defina dos arreglos de números enteros (cantidad de elementos y valores de elección libre).
- Genere un tercer arreglo que contenga todos los valores de los dos primeros arreglos, pero ordenados de manera ascendente.
- Describe de manera detallada el proceso seguido para la creación de la aplicación. Debe quedar claro el proceso de análisis y determinación de requerimientos, así como la manera como se utilizan los arreglos para resolver estos requerimientos.
- sube el código fuente de la aplicación a tu repositorio Github, y a tu documento agrega imágenes de las pantallas de la ejecución de la aplicación.

```
index.html x
pract 3 > index.html > html > body > script > concatAndSortArrays
1 <!DOCTYPE html>
2 <html style="background-color: #aquamarine;">
3 <head>
4   <title>Arreglos en JavaScript</title>
5 </head>
6 <body style="padding:30px; margin: 40px;background-color: #rgb(253, 253, 253);">
7   <h1>Segunda actividad</h1><br>
8   <label for="numeros">Ingresa números enteros separados por espacios:</label><br>
9   <label for="array1">Arreglo 1:</label><br>
10  <input type="text" id="array1" placeholder="Ingrese valores separados por comas">
11  <br>
12  <label for="array2">Arreglo 2:</label><br>
13  <input type="text" id="array2" placeholder="Ingrese valores separados por comas">
14  <br><br>
15  <button onclick="concatAndSortArrays()">Generar y Ordenar Arreglo</button>
16
17  <script>
18    function concatAndSortArrays() {
19      const array1Input = document.getElementById("array1").value;
20      const array2Input = document.getElementById("array2").value;
21
22      const array1 = array1Input.split(" ").map(Number);
23      const array2 = array2Input.split(" ").map(Number);
24
25      const combinedArray = array1.concat(array2);
26      const sortedArray = combinedArray.sort((a, b) => a - b);
27
28      console.log("Arreglo combinado y ordenado:", sortedArray);
29    }
30  </script>
31 </body>
32 </html>
```



4.- Escriba una aplicación que tenga la siguiente funcionalidad:

- Defina un arreglo de números enteros (cantidad de elementos y valores de elección libre).
- Obtenga y muestre el valor de la moda de los valores del arreglo. Recuerda que la moda es el valor del número que más se repite en la lista de elementos.
- Describe de manera detallada el proceso seguido para la creación de la aplicación. Debe quedar claro el proceso de análisis y determinación de requerimientos, así como la manera como se utilizan los arreglos para resolver estos requerimientos.
- sube el código fuente de la aplicación a tu repositorio Github, y a tu documento agrega imágenes de las pantallas de la ejecución de la aplicación.

```
pract 4 > index.html > html
1 <!DOCTYPE html>
2 <html style="background-color: aquamarine;">
3 <head>
4   <title>Moda de un arreglo de números</title>
5 </head>
6 <body style="padding:30px; margin: 40px;background-color: rgb(253, 253, 253);">
7   <h1>Calculadora de Moda</h1>
8   <p>Ingrese los números separados por comas:</p>
9   <input type="text" id="numbersInput"><br>
10  <button onclick="calcularModa()">Calcular Moda</button>
11  <div id="result"></div>
12
13  <script>
14    function calcularModa() {
15      const numbersInput = document.getElementById("numbersInput").value;
16      const numbersArray = numbersInput.split(' ').map(Number);
17
18      // Calcular la moda
19      const frequencyMap = {};
20      let moda = null;
21      let maxFrequency = 0;
22
23      for (const number of numbersArray) {
24        frequencyMap[number] = (frequencyMap[number] || 0) + 1;
25        if (frequencyMap[number] > maxFrequency) {
26          moda = number;
27          maxFrequency = frequencyMap[number];
28        }
29      }
30
31      if (moda !== null) {
32        console.log("numero de modal es: ",moda);
33      }
34    }
35  </script>
36</body>
37</html>
```

Calculadora de Moda

Ingrese los números separados por comas:

Identifique la carpeta raíz del proyecto para abrir archivos de código fuente en Visual Studio Code y sincronizar los cambios. [Mostrar más](#)

Establecer carpeta raíz No volver a mostrar

Le damos la bienvenida Elementos Consola +

top Filtro Niveles predeterminados 4

Live reload enabled. [index.html:66](#)

numero de modal es: 3 [index.html:32](#)

>|

5.- Escriba una aplicación que tenga la siguiente funcionalidad:

- Defina dos arreglos de números enteros (cantidad de elementos y valores de elección libre. Los arreglos son del mismo tamaño).
- Genere un tercer arreglo que contenga en cada posición el valor menor de los valores de los dos primeros arreglos, pero ordenados de manera descendente.
- Describe de manera detallada el proceso seguido para la creación de la aplicación. Debe quedar claro el proceso de análisis y determinación de requerimientos, así como la manera como se utilizan los arreglos para resolver estos requerimientos.
- sube el código fuente de la aplicación a tu repositorio Github, y a tu documento agrega imágenes de las pantallas de la ejecución de la aplicación.

```
index.html x
pract 5 > index.html > html > body > p
1 <!DOCTYPE html>
2 <html style="background-color: aquamarine;">
3 <head>
4   <title>Arreglos y Mínimos</title>
5 </head>
6 <body style="padding:30px; margin: 40px;background-color: rgb(253, 253, 253);">
7   <h1>Arreglos y Mínimos</h1>
8
9   <p>Ingresa los elementos de los dos arreglos separados por comas:</p>
10  <input type="text" id="inputArray1" placeholder="Arreglo 1"><br><br>
11  <input type="text" id="inputArray2" placeholder="Arreglo 2"><br><br>
12  <button onclick="findMinValues()">Encontrar Mínimos</button>
13
14  <p>Arreglo 1: <span id="array1Result"></span></p>
15  <p>Arreglo 2: <span id="array2Result"></span></p>
16  <p>Arreglo de Mínimos Descendentes: <span id="minValuesResult"></span></p>
17
18  <script>
19    function findMinValues() {
20      // Obtén los valores de los dos arreglos desde los campos de entrada
21      const inputArray1 = document.getElementById("inputArray1").value;
22      const inputArray2 = document.getElementById("inputArray2").value;
23
24      // Convierte los valores de cadena en arreglos numéricos
25      const arr1 = inputArray1.split(' ').map(Number);
26      const arr2 = inputArray2.split(' ').map(Number);
27
28      // Encuentra los valores mínimos en los dos arreglos
29      const minValues = arr1.map((value, index) => Math.min(value, arr2[index]));
30
31      // Ordena los valores mínimos de manera descendente
32      minValues.sort((a, b) => b - a);
33    }
34  </script>
35</body>
36</html>
```

Arreglos y Mínimos

Ingresa los elementos de los dos arreglos separados por comas:

Arreglo 1: [2,7,4,0,8]

Arreglo 2: [1,0,7,5,3]

Arreglo de Mínimos Descendentes: [4,3,1,0,0]

6.- Escriba una aplicación que tenga la siguiente funcionalidad:

- Defina un arreglo de dos dimensiones (cantidad de elementos y valores de elección libre. Debe ser una matriz cuadrada).
- Genere un tercer arreglo que contenga los valores de la diagonal principal.
- Describe de manera detallada el proceso seguido para la creación de la aplicación. Debe quedar claro el proceso de análisis y determinación de requerimientos, así como la manera como se utilizan los arreglos para resolver estos requerimientos.
- sube el código fuente de la aplicación a tu repositorio Github, y a tu documento agrega imágenes de las pantallas de la ejecución de la aplicación.

```
index.html X
pract 6 > index.html > html > body > input#array2
1 <!DOCTYPE html>
2 <html style="background-color: #aquamarine;">
3 <head>
4   <title>Arreglos en JavaScript</title>
5 </head>
6 <body style="padding:30px; margin: 40px;background-color: #rgb(253, 253, 253);">
7   <label for="numeros">Ingresa números enteros separados por espacios:</label><br>
8   <h4>Arreglo 1</h4>
9   <input type="number" id="array1" placeholder="Ingresa números separados por comas">
10
11   <h4>Arreglo 2</h4>
12   <input type="number" id="array2" placeholder="Ingresa números separados por comas">
13
14   <button onclick="findMinAndSort()">Calcular</button>
15
16   <h3>Arreglo Resultante (Mínimos Ordenados Descendente)</h3>
17   <div id="result"></div>
18
19   <script>
20     function findMinAndSort() {
21       // Obtener los valores ingresados en los dos arreglos
22       const inputArray1 = document.getElementById('array1').value;
23       const inputArray2 = document.getElementById('array2').value;
24
25       // Convertir los valores en arreglos de números
26       const arr1 = inputArray1.split(' ').map(Number);
27       const arr2 = inputArray2.split(' ').map(Number);
28
29       // Encontrar los valores mínimos de los dos arreglos
30       const minValues = [];
31       for (let i = 0; i < arr1.length; i++) {
32         minValues.push(Math.min(arr1[i], arr2[i]));
```

Ingresa números enteros separados por espacios:

Arreglo 1

Arreglo 2

Arreglo Resultante (Mínimos Ordenados Descendente)

3, 3, 2, 2, 2