

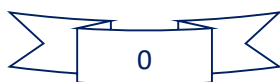


HITO 1 DEL 3º TRIMESTRE DE LM

LENGUAJE DE MARCAS



30 DE ABRIL DE 2024
IKER CONTRERAS ESQUER
1º de ASIR



ÍNDICE

CALCULADORA.....	2
HTML.....	2
CSS	3
JAVA SCRIPT 1	4
JAVA SCRIPT 2	5
JAVA SCRIPT 3	7
COMPROBACIÓN.....	8
CLIMA	9
HTML.....	9
CSS	10
JAVA SCRIPT	11
COMPROBACIÓN.....	13
WEBGRAFÍA	13

CALCULADORA

HTML

```
1 <!DOCTYPE html>
2 <html lang="en">
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 href="https://fonts.googleapis.com/css2?family=Fraunces&display=swap" rel="stylesheet">
8   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMcGjq3xI6Pa3UoOR9736V733F4P7D747W7063359c9b7941DB5A" crossorigin="anonymous">
9   <link href="index.css" rel="stylesheet" />
10  <script src="display.js" type="text/javascript"></script>
11  <script src="calculadora.js" type="text/javascript"></script>
12  <script src="index.js" type="text/javascript" defer></script>
13 </head>
```

En primer lugar, vamos a comentar el encabezado del HTML el cual tiene las referencias a las distintas páginas externas las cuales nos ayudan a obtener nuestra calculadora, a continuación, tendríamos las referencias de las paginas en las que hemos trabajado (en las que hemos podido realizar modificaciones), ya sea el CSS o los tres java scripts que nos ayudan a que nuestra calculadora sea funcional.

```
14 <body>
15   <div class="container">
16     <div class="calculadora">
17       <div class="display">
18         <div id="valor-anterior"></div>
19         <div id="valor-actual"></div>
20       </div>
21       <button class="btn btn-dark" onclick="display.borrarTodo()">C</button>
22       <button class="btn btn-dark" onclick="display.borrar()">⌫</button>
23       <button type="button" class="btn btn-dark operador" value="porcentaje"%</button>
24       <button class="numero"></button>
25       <button type="button" class="btn btn-dark numero">7</button>
26       <button type="button" class="btn btn-dark numero">8</button>
27       <button type="button" class="btn btn-dark numero">9</button>
28       <button type="button" class="btn btn-dark operador float-right" value="dividir"/></button>
29       <button type="button" class="btn btn-dark numero">4</button>
30       <button type="button" class="btn btn-dark numero">5</button>
31       <button type="button" class="btn btn-dark numero">6</button>
32       <button type="button" class="btn btn-dark operador float-right" value="multiplicar"/></button>
33       <button type="button" class="btn btn-dark numero">1</button>
34       <button type="button" class="btn btn-dark numero">2</button>
35       <button type="button" class="btn btn-dark numero">3</button>
36       <button type="button" class="btn btn-dark operador float-right" value="sumar"/></button>
37       <button type="button" class="btn btn-dark numero">0</button>
38       <button type="button" class="btn btn-dark operador float-right" value="igual"/></button>
39       <button type="button" class="btn btn-dark numero"></button>
40       <button type="button" class="btn btn-dark operador float-right" value="restar"/></button>
41       <button type="button" class="btn btn-dark operador" value="raiz"/></button>
42       <button type="button" class="btn btn-dark operador" value="cuadrado"x²</button>
43     </div>
44   </div>
45 </body>
46 </html>
47
```

Y en segundo y último lugar para terminar el HTML de la calculadora, tendríamos el esqueleto en el cual tenemos las clases en las que he trabajado para poder darle estilos y después tenemos los botones, los cuales están extraídos de la página de Bootstrap, los botones los diferencio por 3 grupos, el primer grupo serían los números que van del 1 al 9, el segundo grupo iría compuesto de los operadores, es decir los botones encargados de realizar las operaciones y por último tenemos los botones que borran, los cuales tenemos la C que lo borra todo y la flecha que solo borra un dígito,

CSS

```
1  * {
2    font-family: 'Fraunces', serif;
3    color: #fff;
4  }
5
6  body {
7    margin: 0;
8    background-color: #224870;
9  }
10
11  .container {
12    display: flex;
13    justify-content: center;
14    align-items: center;
15    height: 100vh;
16  }
17
18  .calculadora {
19    display: grid;
20    grid-template-columns: repeat(4, 75px);
21    grid-template-rows: 160px repeat(5, 75px);
22    background-color: #ffffff;
23    padding: 15px;
24    border-radius: 32px;
25    box-shadow: 15px 10px 0px 5px #00000033;
26  }
27
28  button {
29    cursor: pointer;
30    margin: 5px;
31    padding: 0;
32    border-radius: 32px;
33    font-size: 1.5em;
34    border: none;
35    background-color: #ffffff;
36  }
37
38
39
40  button:focus {
41    outline: none;
42  }
43
44  .col-2 {
45    grid-column: span 2;
46  }
47
48  .display {
49    grid-column: 1 / -1;
50    padding: 16px;
51    display: flex;
52    margin: 10px 10px 20px;
53    flex-direction: column;
54    align-items: flex-end;
55    background-color: #363636;
56    border-radius: 32px;
57    text-align: right;
58    justify-content: space-between;
59    word-break: break-all;
60    box-shadow: 0px 0px 0px 10px #00000033;
61  }
62
63  #valor-actual {
64    font-size: 1.5em;
65  }
66
67  #valor-anterior {
68    font-size: 2em;
69  }
```

En el CSS he realizado la tarea de colocar la calculadora en el centro, darle color, utilizar CSS grid para colocar los botones, dar los márgenes necesarios entre los botones y cambiar el tipo de fuente tanto el de la pantalla como el de los botones.

JAVA SCRIPT 1

```
1  const displayValorAnterior = document.getElementById('valor-anterior');
2  const displayValorActual = document.getElementById('valor-actual');
3  const botonesNumeros = document.querySelectorAll('.numero');
4  const botonesOperadores = document.querySelectorAll('.operador');
5
6  const display = new Display(displayValorAnterior, displayValorActual);
7
8  botonesNumeros.forEach(boton => {
9    |   boton.addEventListener('click', () => display.agregarNumero(boton.innerHTML));
10   | });
11
12  botonesOperadores.forEach(boton => {
13    |   boton.addEventListener('click', () => display.computar(boton.value))
14   | });
```

Los dos primeros elementos seleccionan los elementos de los ID's de valor-anterior y el valor-actual, para después poder mostrarlos en la calculadora. A continuación, los dos siguientes elementos llaman a otros 2 ID's que en este caso son número y operador, estos son para los números y los botones que se mostraran en la calculadora. El siguiente elemento crea una nueva solicitud del objeto, y Display pasa los elementos que representan el valor anterior y el valor actual. Los dos últimos elementos están destinados para los botones, el primero crea un bucle sobre cada botón, para cuando se haga clic llame a la función agregar número, y Display conecta con el HTML para que el botón haga su función; y por último, esta función está destinada para los botones de los operadores y llama a la función computar.

JAVA SCRIPT 2

```
1  class Display {  
2      constructor(displayValorAnterior, displayValorActual) {  
3          this.displayValorActual = displayValorActual;  
4          this.displayValorAnterior = displayValorAnterior;  
5          this.calculadora = new Calculadora();  
6          this.tipoOperacion = undefined;  
7          this.valorActual = '';  
8          this.valorAnterior = '';  
9          this.signos = {  
10             sumar: '+',  
11             dividir: '÷',  
12             multiplicar: 'x',  
13             restar: '-',  
14             raiz: '√',  
15             porcentaje: '%',  
16             cuadrado: '^2',  
17         };  
18     }  
19 }
```

En esta página de empezamos con el elemento constructor que nos va a ayudar a inicializar nuestra calculadora con elementos de visualización, con diferentes variables necesarias, como el tipo de operación actual y los valores anteriores y también los actuales, además de definir los símbolos de las operaciones.

```
20     borrar() {  
21         this.valorActual = this.valorActual.toString().slice(0, -1);  
22         this.imprimirValores();  
23     }  
24 }
```

Este elemento borra el ultimo dígito del valor actual.

```
25     borrarTodo() {  
26         this.valorActual = '';  
27         this.valorAnterior = '';  
28         this.tipoOperacion = undefined;  
29         this.imprimirValores();  
30     }  
31 }
```

Este elemento restablece todos los valores y la operación actual.

```
32     computar(tipo) {  
33         this.tipoOperacion && this.calcular();  
34         this.tipoOperacion = tipo;  
35         this.valorAnterior = this.valorActual || this.valorAnterior;  
36         this.valorActual = '';  
37         this.imprimirValores();  
38     }  
39 }
```

Este elemento está destinado a una operación si ya hay una operación pendiente, por lo que establece el nuevo tipo de operación y actualiza los valores anteriores y actuales.

```

40     agregarNumero(numero) {
41         if (numero === '.' && this.valorActual.includes('.')) return;
42         this.valorActual = this.valorActual.toString() + numero.toString();
43         this.imprimirValores();
44     }
45

```

Lo que realiza este elemento es agregar un numero al valor actual, evitando duplicar el punto decimal.

```

46     imprimirValores() {
47         this.displayValorActual.textContent = this.valorActual;
48         this.displayValorAnterior.textContent = `${this.valorAnterior} ${this.signos[this.tipoOperacion] || ''}`;
49     }

```

El elemento actualiza la pantalla con los valores actuales y los anteriores, junto con el símbolo de la operación.

```

50
51     calcular() {
52         const valorAnterior = parseFloat(this.valorAnterior);
53
54         const valorActual = parseFloat(this.valorActual);
55
56         if (this.tipoOperacion == 'cuadrado' || this.tipoOperacion == 'raiz' )
57         {
58             this.valorAnterior = this.calculador[this.tipoOperacion](valorAnterior);
59         } else {
60             if( isNaN(valorActual) || isNaN(valorAnterior) ) return
61             this.valorActual = this.calculador[this.tipoOperacion](valorAnterior, valorActual);
62         }
63     }
64

```

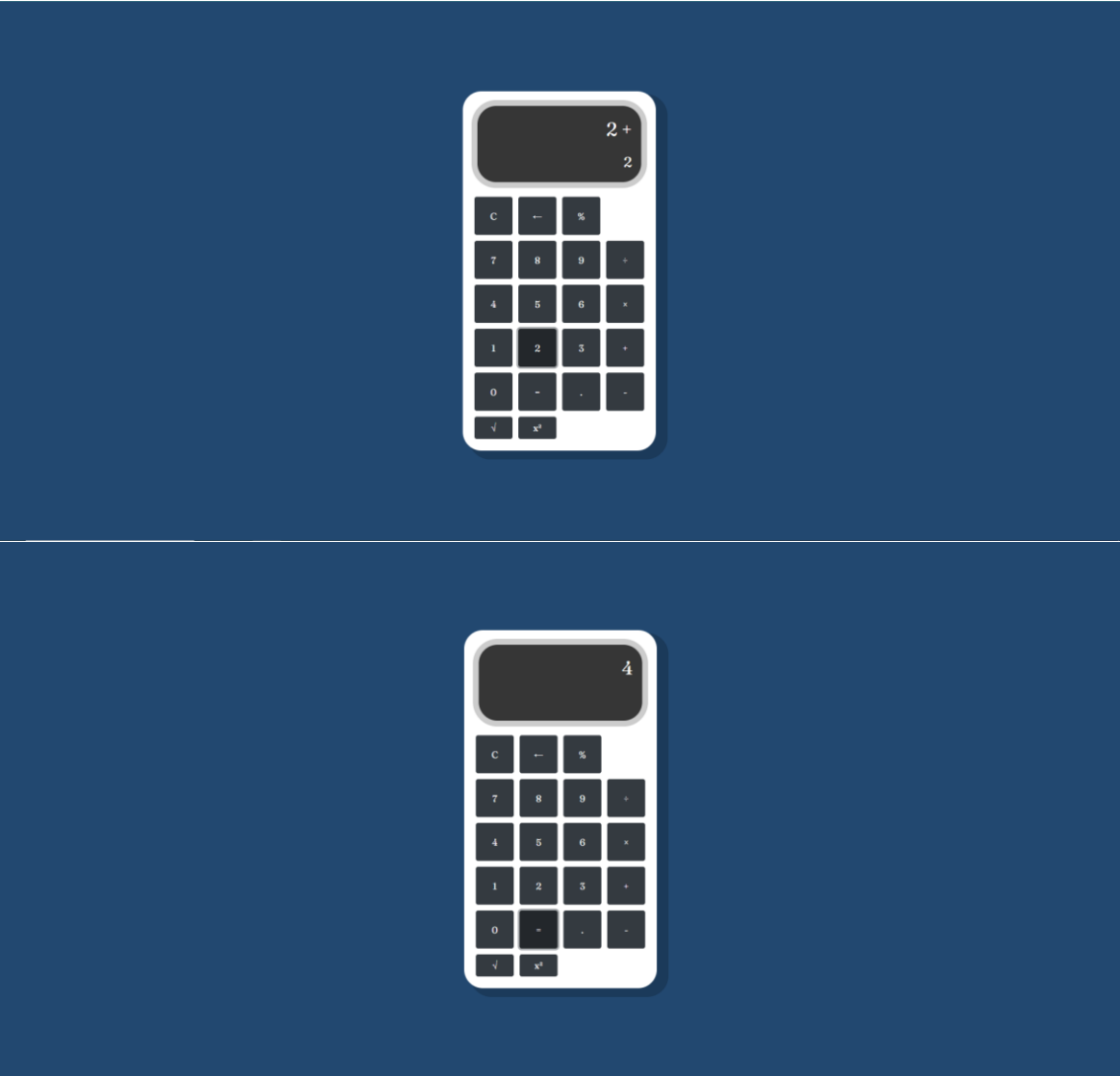
Y por último este elemento realiza el cálculo dependiendo el tipo de operación, utilizando el objeto calculador.

JAVA SCRIPT 3

```
1  class Calculadora {  
2      sumar(num1, num2) {  
3          return num1 + num2;  
4      }  
5  
6      restar(num1, num2) {  
7          return num1 - num2;  
8      }  
9  
10     dividir(num1, num2) {  
11         return num1 / num2;  
12     }  
13  
14     multiplicar(num1, num2) {  
15         return num1 * num2;  
16     }  
17  
18     cuadrado(num1) {  
19         return num1 * num1;  
20     }  
21  
22     porcentaje(num1, num2) {  
23         return num1 * (num2 / 100);  
24     }  
25  
26     raiz(num1) {  
27         return Math.sqrt(num1);  
28     }  
29 }  
30
```

La clase calculadora define los métodos para poder realizar las operaciones matemáticas.

COMPROBACIÓN



CLIMA

HTML

```
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>TU CLIMA DE CONFIDENCIA</title>
7   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyY4C3T6n2aLXB4lbeZw3Wq9aszk8qL8nV6s3fVpVdw3Hj48qyU6W43cnk" crossorigin="anonymous">
8   <link href="css/index.css" rel="stylesheet" />
9 </head>
```

En primer lugar, está el head el cual tiene el título de la página tiene dos links de CSS uno para el Bootstrap y el otro para el CSS propio.

```

11 </body>
12 <nav navbar-expand-lg navbar-light bg-light>
13   <a class="navbar-brand d-flex align-items-center href="#">
14     
15     <span class="h4">TU CLIMA DE CONFIANZA </span>
16   </a>
17   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
18     <span class="navbar-toggler-icon"></span>
19   </button>
20   <div class="collapse navbar-collapse justify-content-end" id="navbarSupportedContent">
21     <ul class="navbar-nav ml-auto">
22       <li class="nav-item dropdown">
23         <label for="provincias" class="m-2">Selecciona una provincia:</label>
24         <select id="provincias" class="custom-select custom-select-lg" onchange="cargarMunicipios()">
25           <option value="" selected disabled>Selecciona una provincia...</option>
26         </select>
27       </li>
28       <li class="nav-item dropdown">
29         <label for="municipios" class="m-2">Selecciona un municipio:</label>
30         <select id="municipios" class="custom-select custom-select-lg" onchange="mostrarClima()">
31           <option value="" selected disabled>Selecciona una provincia primero...</option>
32         </select>
33       </li>
34     </ul>
35   </div>
36 </nav>
37
38 <div id="clima" class="container-fluid">
39   <div class="table-responsive">
40     <table id="tabla-clima" class="table">
41       <thead class="thead-dark">
42         <tr>
43           <th>Fecha:</th>
44           <th>Probabilidad de precipitación (%)</th>
45           <th>Estado del cielo:</th>
46           <th>Temperatura Máxima (°C)</th>
47           <th>Temperatura Mínima (°C)</th>
48           <th>Velocidad del viento (km/h)</th>
49           <th>Dirección del viento:</th>
50         </tr>
51       </thead>
52       <tbody>
53         <tr>
54           <td> </td>
55         </tr>
56       </tbody>
57     </table>
58   </div>
59
60 <script src="tempo.js"></script>
61 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JstQIAqVgZp9w9o9kx4p4VvFFv6+P6Bx4jtr6b2w7" crossorigin="anonymous"></script>
62 <script src="https://code.jquery.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" integrity="sha384-Uo2eKJKSjAfbPHz6IMVz63GBuBMl97k5wDZC4o9wTE8KqE17W8VqZ2/V7MlS901" crossorigin="anonymous"></script>
63 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-gdbO4jgIaQB3ddX26zS3XXQ4aWGXIjQ1V47F770P6hD5Lp7X9kS99pxL97zuLGM" crossorigin="anonymous"></script>
64 </html>

```

Por lo que respecta al cuerpo del HTML tenemos las clases para barra de navegación en la cual tenemos la clase del Bootstrap, junto a la foto del logo y el título. A continuación, tenemos la clase del Bootstrap para que sea responsive tanto el botón de las barras y para los desplegables, en cuanto a la selección de provincias y de los municipios con un “onchange” para que lo saque del Java que he realizado. En cuanto a la tabla de contenido le hemos puesto Bootstrap para que se haga una tabla con la información que he puesto después y que en caso de que no se pueda ver toda tabla pueda moverse la pantalla y poder ver toda la información, y por último ponemos los enlaces al Java script propio y externos.

CSS

```
1  .navbar-brand {
2      font-size: 24px;
3      margin-left: 100px;
4      margin-right: auto;
5      display: block;
6      width: fit-content;
7  }
8
9  .navbar-nav {
10     margin-left: auto;
11 }
12
13 .navbar-nav li {
14     margin-right: 20px;
15 }
16
17 .navbar-nav label {
18     margin-right: 10px;
19 }
20
21 #clima table {
22     width: 100%;
23     border-collapse: collapse;
24     margin-top: 20px;
25 }
26
27 #clima table tbody tr:nth-child(even) {
28     background-color: #f2f2f2;
29 }
30
31 #clima table th {
32     padding-top: 12px;
33     padding-bottom: 12px;
34     text-align: left;
35     background-color: rgb(44, 113, 218);
36     color: white;
37 }
38
39 #logo {
40     height: 80px;
41     margin-left: -110px;
42     position: relative;
43 }
44
```

En el CSS he realizado la tarea de colocar los dos desplegables, darle color a la tabla y a la barra de navegación, dar los márgenes necesarios para poder colocar el logo, el título, los desplegables y la tabla.

JAVA SCRIPT

```
1 const provinciasDropdown = document.getElementById('provincias');
2 const municipiosDropdown = document.getElementById('municipios');
3 const climaDiv = document.getElementById('clima');
4
```

Estos elementos seleccionan la información del HTML en este caso seleccionamos una provincia un municipio y la información del clima.

```
5 fetch('https://www.el-tiempo.net/api/json/v2/provincias')
6   .then(response => response.json())
7   .then(data => {
8     if (data.provincias) {
9       data.provincias.forEach(provincia => {
10         const option = document.createElement('option');
11         option.text = provincia.NOMBRE_PROVINCIA;
12         option.value = provincia.CODPROV;
13         provinciasDropdown.add(option);
14       });
15     } else {
16       console.error('Los datos de las provincias no están en el formato esperado:', data);
17     }
18   })
19   .catch(error => console.error('Error al cargar las provincias:', error));
20
```

La función de fetch realiza una solicitud a la página API que proporciona la información del clima y el elemento de “option” procesa la información para crear opciones de selección que representan las provincias, esto se agrega al elemento “select” correspondiente en el HTML.

```
21 function cargarMunicipios() {
22   const selectedProvincia = provinciasDropdown.value;
23   if (!selectedProvincia) return;
24
25   municipiosDropdown.innerHTML = '<option value="" selected disabled>Cargando...</option>';
26
27   fetch(`https://www.el-tiempo.net/api/json/v2/provincias/${selectedProvincia}/municipios`)
28     .then(response => response.json())
29     .then(data => {
30       if (data.municipios) {
31         municipiosDropdown.innerHTML = '';
32         data.municipios.forEach(municipio => {
33           const option = document.createElement('option');
34           option.text = municipio.NOMBRE;
35           option.value = municipio.COD_GEO;
36           municipiosDropdown.add(option);
37         });
38       } else {
39         console.error('Los datos de los municipios no están en el formato esperado:', data);
40       }
41       console.log('Municipios cargados:', data);
42     })
43     .catch(error => console.error('Error al cargar los municipios:', error));
44 }
45
```

Esta función selecciona na provincia, que se ejecuta la función “cargarMunicipios”, esta función obtiene el valor de la provincia seleccionada y hace una solicitud a la API para obtener el valor de dicha provincia y los municipios de la misma, asimismo, se ejecuta de manera similar a la de las provincias y se agrega el elemento “select”.

```

46 function mostrarClima() {
47   const selectedProvincia = provinciasDropdown.value;
48   const selectedMunicipio = municipiosDropdown.value;
49   console.log('selectedProvincia:', selectedProvincia);
50   console.log('selectedMunicipio:', selectedMunicipio);
51   if (!selectedProvincia || !selectedMunicipio) return;
52
53   fetch('https://corsproxy.io/?https://www.el-tiempo.net/api/json/v1/provincias/${selectedProvincia}/municipios/${selectedMunicipio}/weather')
54     .then(response => response.json())
55     .then(data => {
56       climaDiv.innerHTML = '';
57
58       data.prediccion.dia.forEach(dia => {
59         climaDiv.innerHTML += `<h3>Fecha: ${dia['@attributes'].fecha}</h3>`;
60         climaDiv.innerHTML += `<p>Probabilidad de precipitación: ${dia.prob_precipitacion[0]}%</p>`;
61         climaDiv.innerHTML += `<p>Estado del cielo: ${dia.estado_cielo[0]}</p>`;
62         climaDiv.innerHTML += `<p>Temperatura Máxima: ${dia.temperatura.maxima} °C</p>`;
63         climaDiv.innerHTML += `<p>Temperatura Mínima: ${dia.temperatura.minima} °C</p>`;
64         climaDiv.innerHTML += `<p>Velocidad del viento: ${dia.viento[0].velocidad} km/h</p>`;
65         climaDiv.innerHTML += `<p>Dirección del viento: ${dia.viento[0].direccion}</p>`;
66         climaDiv.innerHTML += `<br>`;
67       });
68       console.log('Clima cargado:', data);
69     })
70     .catch(error => console.error('Error al cargar el clima:', error));
71 }
72
73 function mostrarClima() {
74   const selectedProvincia = provinciasDropdown.value;
75   const selectedMunicipio = municipiosDropdown.value;
76   console.log('selectedProvincia:', selectedProvincia);
77   console.log('selectedMunicipio:', selectedMunicipio);
78   if (!selectedProvincia || !selectedMunicipio) return;
79
80   fetch('https://corsproxy.io/?https://www.el-tiempo.net/api/json/v1/provincias/${selectedProvincia}/municipios/${selectedMunicipio}/weather')
81     .then(response => response.json())
82     .then(data => {
83       const tablaClima = document.getElementById('tabla-clima');
84       tablaClima.getElementsByTagName('tbody')[0].innerHTML = '';
85
86       data.prediccion.dia.forEach(dia => {
87         const fila = `
88           <tr>
89             <td>${dia['@attributes'].fecha}</td>
90             <td>${dia.prob_precipitacion[0]}%</td>
91             <td>${dia.estado_cielo[0]}</td>
92             <td>${dia.temperatura.maxima} °C</td>
93             <td>${dia.temperatura.minima} °C</td>
94             <td>${dia.viento[0].velocidad} km/h</td>
95             <td>${dia.viento[0].direccion}</td>
96           </tr>
97         `;
98         tablaClima.getElementsByTagName('tbody')[0].insertAdjacentHTML('beforeend', fila);
99       });
100       console.log('Clima cargado:', data);
101     })
102     .catch(error => console.error('Error al cargar el clima:', error));
103   }
104 }
105

```

Los elementos de “mostarclima” obtiene la información tanto de la provincia como de los municipios seleccionados, después realiza una petición a la API para obtener los datos del clima específicos de ese municipio seleccionado. Una vez recibidos los datos se muestran en un HTML, en este caso en una tabla, los datos se integran como las filas “tr”.

```

103   }
104 }
105

```

Este elemento se usa para el manejo de errores en caso de que ocurran durante la consulta con la API y se incluyen bloques con “catch”.

COMPROBACIÓN

 TU CLIMA DE CONFIANZA

Selecciona una provincia:


Illes Balears

Selecciona un municipio:

Algaida

Fecha	Probabilidad de precipitación (%)	Estado del cielo	Temperatura Máxima (°C)	Temperatura Mínima (°C)	Velocidad del viento (km/h)	Dirección del viento
2024-04-26	95%	43	21 °C	10 °C	15 km/h	SO
2024-04-27	95%	24	22 °C	13 °C	20 km/h	S
2024-04-28	100%	26	19 °C	13 °C	15 km/h	S
2024-04-29	100%	25	19 °C	12 °C	10 km/h	O

Dimensions: iPhone XR 414 x 896 92% No throttling

 TU CLIMA DE CONFIANZA

Selecciona una provincia:

Illes Balears

Selecciona un municipio:

Algaida

Fecha	Probabilidad de precipitación (%)	Estado del cielo	Temperatura Máxima (°C)
2024-04-26	95%	43	21 °C
2024-04-27	95%	24	22 °C
2024-04-28	100%	26	19 °C
2024-04-29	100%	25	19 °C

WEBGRAFÍA

<https://www.w3schools.com/>

<https://www.youtube.com/>

<https://getbootstrap.com/>

<https://github.com/>