

DWC\examenFInal\asincronia\UD06_paises_garruta_emilio.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>Búsqueda de Países</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      margin: 20px;
11      padding: 0;
12      box-sizing: border-box;
13      align-items: stretch;
14      display: flex;
15      flex-direction: column;
16      gap: 20px;
17    }
18    form {
19      margin-bottom: 20px;
20      display: flex;
21      flex-direction: column;
22      gap: 15px;
23    }
24    .container {
25      margin-top: 20px;
26      border: 1px solid #ddd;
27      padding: 20px;
28      border-radius: 5px;
29    }
30    .error {
31      color: red;
32    }
33    img {
34      max-width: 100px;
35      display: block;
36      margin-top: 10px;
37    }
38    p{
39      text-align: justify;
40    }
41  </style>
42 </head>
43 <body>
44   <h1>Búsqueda de Información de Países</h1>
45   <form id="countryForm">
46     <div>
47       <label for="countrySelect">Selecciona un país:</label>
48       <select id="countrySelect">
49         <option value="">--Selecciona un país--</option>
50       </select>
51     </div>
52
53     <div>
54       <label for="countryInput">O introduce el nombre del país:</label>
55       <input type="text" id="countryInput" placeholder="Ej. Spain">
56     </div>
57
```

```

58     <div>
59         <button type="submit">Enviar petición</button>
60     </div>
61
62 </form>
63 <div class="container" id="resultContainer">
64     <p id="error" class="error"></p>
65     <p id="fullName"></p>
66     <p id="capital"></p>
67     <p id="population"></p>
68     <img id="flag" src=""> <!--No incluyo el alt para que no aparezca el texto precargado al no
haber imagen antes de enviar la consulta-->
69 </div>
70     <h3>Cuestiones más relevantes</h3>
71
72     <p> Lo primero ha sido ver cómo están dispuestos los eobjetos que envía la API de Rest
Countries,
73         que devuelve un array de objetos, donde cada objeto representa un país. La estructura de
74         estos
75         objetos contiene una amplia gama de propiedades, aunque las más relevantes
76         para la mayoría de los casos son las relacionadas con el nombre, la capital, la población,
77         la bandera,...</p>
78         <p>Cuando haces una llamada a https://restcountries.com/v3.1/all, obtendrás un array con
79         objetos similares al ejemplo
80         anterior para todos los países. En dfunción de esto se monta todo el código.</p>
81     <div>
82         <ol>
83             <li><strong>name:</strong>
84                 <ul>
85                     <li><strong>common:</strong> Nombre común del país.</li>
86                     <li><strong>official:</strong> Nombre oficial del país.</li>
87                     <li><strong>nativeName:</strong> Nombres en los idiomas nativos del país.</li>
88                 </ul>
89             </li>
90             <li><strong>capital:</strong>
91                 <ul>
92                     <li>Una lista de las capitales del país (normalmente un solo elemento).</li>
93                 </ul>
94             </li>
95             <li><strong>region y subregion:</strong>
96                 <ul>
97                     <li><strong>region:</strong> Región general a la que pertenece el país (por
ejemplo, "Americas").</li>
98                     <li><strong>subregion:</strong> Subregión específica (por ejemplo, "South
America").</li>
99                 </ul>
100             </li>
101             <li><strong>population:</strong>
102                 <ul>
103                     <li>Número de habitantes del país.</li>
104                 </ul>
105             </li>
106             <li><strong>flags:</strong>
107                 <ul>
108                     <li><strong>png:</strong> URL a la imagen de la bandera en formato PNG.</li>
109                     <li><strong>svg:</strong> URL a la imagen de la bandera en formato SVG.</li>
110                 </ul>
111             </li>
112             <li><strong>languages:</strong>
113                 <ul>

```

```

111         <li>Un objeto que mapea los códigos de idioma a sus nombres (por ejemplo, "spa":
"Spanish").</li>
112     </ul>
113 </li>
114 <li><strong>borders:</strong>
115     <ul>
116         <li>Lista de los códigos de los países vecinos (según el estándar ISO 3166-1 alpha-
3).</li>
117     </ul>
118 </li>
119 <li><strong>area:</strong>
120     <ul>
121         <li>Área total del país en kilómetros cuadrados.</li>
122     </ul>
123 </li>
124 </ol>
125 <div>
126
127 </div>
128
129 </div>
130 <script>
131     // Cuando el contenido del DOM esté completamente cargado, se ejecuta la función.
132     document.addEventListener('DOMContentLoaded', () => {
133         // Recogemos las referencias a los elementos del DOM que se utilizarán más adelante.
134         const countryForm = document.getElementById('countryForm'); // Formulario donde el usuario
ingresa o selecciona un país.
135         const countryInput = document.getElementById('countryInput'); // Input donde el usuario
puede escribir el nombre del país.
136         const countrySelect = document.getElementById('countrySelect'); // Select donde el usuario
puede elegir un país de una lista.
137         const resultContainer = document.getElementById('resultContainer'); // Contenedor donde se
mostrarán los resultados del país seleccionado.
138
139         // Elementos que mostrarán la información del país o un mensaje de error.
140         const errorElement = document.getElementById('error'); // Elemento donde se muestra el
mensaje de error.
141         const fullNameElement = document.getElementById('fullName'); // Elemento donde se muestra
el nombre completo del país.
142         const capitalElement = document.getElementById('capital'); // Elemento donde se muestra la
capital del país.
143         const populationElement = document.getElementById('population'); // Elemento donde se
muestra la población del país.
144         const flagElement = document.getElementById('flag'); // Elemento donde se muestra la
bandera del país.
145
146         // Realizamos una solicitud HTTP para cargar los nombres de los países desde la API de Rest
Countries.
147         fetch('https://restcountries.com/v3.1/all') // Solicitud GET a la API para obtener todos
los países.
148         .then(response => { // Cuando recibimos la respuesta de la API.
149             if (!response.ok) { // Si la respuesta no es exitosa (no tiene código 200).
150                 throw new Error('Error al cargar la lista de países'); // Lanzamos un error.
151             }
152             return response.json(); // Si la respuesta es exitosa, la convertimos a JSON.
153         })
154         .then(data => { // Cuando los datos se han recibido y convertido a JSON.
155             console.log(data); // Mostramos los datos recibidos en la consola (para
depuración).
156
157             // Iteramos sobre cada país recibido en el array de países.
158             data.forEach(country => {

```

```

158         if (country.name && country.name.common) { // Verificamos que cada país tenga
un nombre común.
159             const option = document.createElement('option'); // Creamos una nueva
opción para el <select>.
160             option.value = country.name.common; // Establecemos el valor de la opción
(nombre común del país).
161             option.textContent = country.name.common; // Establecemos el texto visible
de la opción.
162             countrySelect.appendChild(option); // Añadimos la opción al <select>.
163         }
164     });
165 })
166     .catch(err => { // Si ocurre un error al realizar la solicitud.
167         errorElement.textContent = 'No se pudo cargar la lista de países. Por favor,
intenta nuevamente más tarde.'; // Mostramos un mensaje de error.
168         console.error('Error al cargar países:', err); // Mostramos el error en la
consola.
169     });
170
171     // Función para obtener los datos de un país específico a través de su nombre.
172     const fetchCountryData = (name) => {
173         fetch(`https://restcountries.com/v3.1/name/${name}`) // Realizamos una solicitud GET
para obtener los datos del país por nombre.
174         .then(response => { // Cuando recibimos la respuesta de la API.
175             if (!response.ok) { // Si la respuesta no es exitosa.
176                 if (response.status === 404) { // Si el país no existe (código de estado
404).
177                     throw new Error('El país no existe. Por favor, verifica el nombre
ingresado.');
```

// Lanzamos un error con un mensaje personalizado.

```

178                 } else {
179                     throw new Error('Ocurrió un problema al buscar los datos. Intenta
nuevamente más tarde.');
```

// Si ocurrió otro tipo de error.

```

180                 }
181             }
182             return response.json(); // Si la respuesta es exitosa, la convertimos a JSON.
183         })
184         .then(data => { // Cuando los datos se han recibido y convertido a JSON.
185             const country = data[0]; // Tomamos el primer país del array (ya que la
respuesta es un array).
186
187             // Actualizamos el DOM con la información del país.
188             errorElement.textContent = ''; // Limpiamos cualquier mensaje de error previo.
189             fullNameElement.textContent = `Nombre completo: ${country.name.official}`; //
Mostramos el nombre completo del país.
190             capitalElement.textContent = `Capital: ${country.capital ? country.capital[0] :
'N/A'}`; // Mostramos la capital (si existe, si no, 'N/A').
191             populationElement.textContent = `Población:
${country.population.toLocaleString()}`; // Mostramos la población formateada con comas.
192             flagElement.src = country.flags.png; // Establecemos la URL de la imagen de la
bandera en formato PNG.
193             flagElement.alt = `Bandera de ${country.name.common}`; // Establecemos el
texto alternativo de la imagen de la bandera.
194         })
195         .catch(error => { // Si ocurre un error al obtener los datos del país.
196             // Limpiamos los elementos de información y mostramos el mensaje de error.
197             errorElement.textContent = error.message;
198             fullNameElement.textContent = '';
199             capitalElement.textContent = '';
200             populationElement.textContent = '';
201             flagElement.src = '';
202             flagElement.alt = '';

```

```
203         });
204     };
205
206     // Event listener para manejar el formulario cuando el usuario lo envíe.
207     countryForm.addEventListener('submit', (e) => {
208         e.preventDefault(); // Evita que el formulario se envíe de manera convencional.
209
210         // Verificamos si el usuario ha completado ambos campos (Input y Select).
211         const isInputFilled = countryInput.value.trim() !== ''; // Verificamos si el input de
texto tiene algún valor.
212         const isSelectFilled = countrySelect.value !== ''; // Verificamos si el select tiene
una opción seleccionada.
213
214         if (isInputFilled && isSelectFilled) { // Si ambos campos están llenos, mostramos un
mensaje de error.
215             errorElement.textContent = 'Por favor, usa solo un medio para buscar: selecciona un
país o escribe su nombre, no ambos.';
216             return; // Detenemos la ejecución del código para evitar la búsqueda.
217         }
218
219         const countryName = countryInput.value.trim() || countrySelect.value; // Si el input
tiene valor, lo usamos, si no, usamos el valor del select.
220
221         if (!countryName) { // Si no se ha proporcionado ningún nombre de país (ni en el input
ni en el select).
222             errorElement.textContent = 'Por favor, selecciona o introduce un nombre de país.';
// Mostramos un mensaje de error.
223             return; // Detenemos la ejecución del código.
224         }
225
226         fetchCountryData(countryName); // Si todo está bien, buscamos los datos del país.
227     });
228 });
229 </script>
230
231
232 </body>
233 </html>
234
235
236
```