

DWC\examenFInal\asincronia\Garruta_Gonzalez_Emilio_prediccionDeCartas.html

```
1 <!DOCTYPE html>
2 <html lang="es">
3
4 <head>
5   <meta charset="UTF-8" />
6   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8   <script src="https://www.gstatic.com/charts/loader.js"></script>
9   <style>
10
11     header{
12       display: flex;
13       flex-direction: row;
14       justify-content: center;
15       background: black;
16       color: white;
17       width: 100%;
18
19     }
20     p{
21       text-decoration-style: double;
22     }
23     main {
24       color: black;
25       display: flex;
26       flex-direction: column;
27       gap: 2rem;
28       align-items: center;
29       justify-content: center;
30       background: white;
31     }
32     #presentacion {
33       display: flex;
34       flex-direction: column;
35       align-items: center;
36       justify-content: center;
37     }
38     #container {
39       display: flex;
40       flex-direction: column;
41       align-items: center;
42       justify-content: center;
43     }
44     #botonesJuego{
45       display: flex;
46       flex-direction: row;
47       align-items: center;
48       gap: 2rem;
49     }
50     button {
51       padding: 0.75rem 1rem;
52       font-size: 1rem;
53       background-color: #cf0000;
54       color: white;
55       border: 2px solid transparent;
56       border-color: black;
57       border-radius: 0.375rem;
```

```
58     font-weight: 600;
59     cursor: pointer;
60     transition: background-color 0.3s, border 0.3s;
61 }
62 button:hover {
63     background-color: rgb(230, 230, 7);
64     border: 2px solid #fbfbfb;
65     color: black;
66     border-color: black;
67 }
68 }
69 button:focus {
70     outline: none;
71     box-shadow: 0 0 0 2px #000000;
72 }
73 button::placeholder{
74     color: rgb(203, 224, 255);
75 }
76 #estadisticas {
77     display: flex;
78     justify-content: center;
79     align-items: center;
80     width: 80%;
81     height: 400px;
82     margin: auto;
83 }
84
85 footer{
86     display: flex;
87     flex-direction: row;
88     justify-content: center;
89     background: black;
90     color: white;
91     width: 100%;
92
93 }
94 </style>
95 </head>
96
97 <body>
98     <header>
99         <div>
100             <h1>Juego de cartas: pulsa si quieres jugar.</h1>
101         </div>
102
103     </header>
104     <main>
105         <div id="presentacion">
106             <h1>Inicio del juego de cartas</h1>
107             <button id="iniciarJuego">Iniciar Juego</button>
108         </div>
109         <div id='container'></div>
110         <div id="botonesJuego"></div>
111         <div id="finJuego"></div>
112         <div id="estadisticas"></div>
113     </main>
114     <footer>
115         <div>
116             <h1>¿Cuántas podrás acertar?</h1>
117         </div>
```

```

118     </footer>
119     <script>
120         let contenedor = document.querySelector('#container');
121         let controles = document.querySelector('#botonesJuego');
122         let finalizar = document.querySelector('#finJuego');
123         let estadisticas = document.querySelector('#estadisticas');
124         document.querySelector('#iniciarJuego').addEventListener('click', comenzarJuego);
125
126         let tamanoMazo = 20;
127         let puntuacion = 0;
128         let indiceCartaActual = 0;
129         let mazo = [];
130         let apuestas = 0;
131         let aciertos = 0;
132         let fallos = 0;
133
134
135         function comenzarJuego() {
136             let botonInicio = document.querySelector('#iniciarJuego');
137             botonInicio.disabled = true;
138             tamanoMazo = 20;
139             puntuacion = 0;
140             indiceCartaActual = 0;
141             mazo = [];
142             apuestas = 0;
143             aciertos = 0;
144             fallos = 0;
145             contenedor.innerHTML = "";
146             controles.innerHTML = "";
147             finalizar.innerHTML = "";
148             estadisticas.innerHTML = "";
149
150             fetch('https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1') //se genera un nuevo
id, no es estrictamente necesario hacerlo
151                                                     // cada vez que se
inicie el juego,
152                                                     // ya que cada id se
puede usar durante 20 días sin que expire, pero
153                                                     //me garantiza que cada
vez que se habra la aplicación funcione ya que
154                                                     //el id generado será
válido
155                 .then(response => response.json())
156                 .then(data => fetch(`https://deckofcardsapi.com/api/deck/${data.deck_id}/draw/?
count=${tamanoMazo}`))//genero el mazo de 20 cartas
157                 .then(response => response.json())
158                 .then(data => {
159                     mazo = data.cards.map(carta => { //procesamos el array para dar valores comparables
y nombres en castellano
160                         carta.suit = { "SPADES": "PICAS", "HEARTS": "CORAZONES", "DIAMONDS":
"DIAMANTES", "CLUBS": "TRÉBOLES" }[carta.suit] || carta.suit;
161                         carta.value = { "KING": "13", "QUEEN": "12", "JACK": "11", "ACE": "14" }
[carta.value] || carta.value;
162                         return carta;
163                     });
164                     //se llama a las funciones necesarias para el juego
165                     mostrarCartaActual();
166                     agregarBotones();
167                     //Se vuelve a habilitar el boton
168                     botonInicio.disabled = false;
169                 })

```

```
170         .catch(error => {
171             contenedor.innerHTML = `

Error al obtener los datos del juego: ${error}</p>`;
172             botonInicio.disabled = false;
173         });
174     }
175
176     function mostrarCartaActual () { //depliega la carta que corresponde en cada momento empezando
por la inicial del mazo
177         contenedor.innerHTML = "";
178         let carta = mazo[indiceCartaActual];
179         let verValorCarta = document.createElement("p");
180         verValorCarta.textContent = `La carta es el ${carta.value} de ${carta.suit}`;
181         contenedor.append(verValorCarta);
182
183         let verImgCarta = document.createElement("img");
184         verImgCarta.setAttribute("src", carta.image);
185         verImgCarta.setAttribute("alt", `Carta: ${carta.value} de ${carta.suit}`);
186         contenedor.append(verImgCarta);
187         actualizarPuntuacion();
188     }
189
190     function jugar(eleccion) {
191         if (indiceCartaActual >= mazo.length - 1) return; //Si se ha visto la última carta sale
192         let valorActual = parseInt(mazo[indiceCartaActual].value); //valor numérico
193         let valorSiguiente = parseInt(mazo[++indiceCartaActual].value);
194         //comparamos las elecciones con los valores y actualizamos variables
195         if ((eleccion === "mayor" && valorSiguiente > valorActual) || (eleccion === "menor" &&
valorSiguiente < valorActual)) {
196             aciertos ++;
197             puntuacion++;
198             apuestas ++;
199         } else if (valorSiguiente !== valorActual) {
200             fallos ++
201             puntuacion--;
202             apuestas ++;
203         }
204         mostrarCartaActual (); //actualizamos
205         //si se llega al final sale
206         if (indiceCartaActual === mazo.length - 1) finalizarJuego();
207     }
208
209     function finalizarJuego () { //deja la página con los resultados y el boton para iniciar juego
210         controles.innerHTML = "";
211         finalizar.innerHTML = "";
212         contenedor.innerHTML = "";
213
214         presentarResultados ();
215     }
216
217     function actualizarPuntuacion () {
218         let puntuacionDisplay = document.querySelector("#puntuacion");
219         if (!puntuacionDisplay) { //Si no existe e display lo crea
220             puntuacionDisplay = document.createElement("p");
221             puntuacionDisplay.setAttribute("id", "puntuacion");
222             contenedor.append(puntuacionDisplay);
223         }
224         //actualiza el texto con el valor de la puntuación
225         puntuacionDisplay.textContent = `Puntuación: ${puntuacion}`;
226     }
227


```

```
228     function agregarBotones () { //agrega los botones necesarios
229         let botonMas = document.createElement('button');
230         botonMas.textContent = "La carta será mayor.";
231         botonMas.addEventListener("click", () => jugar("mayor"));
232         controles.append(botonMas);
233
234         let botonMenos = document.createElement('button');
235         botonMenos.textContent = "La carta será menor.";
236         botonMenos.addEventListener("click", () => jugar("menor"));
237         controles.append(botonMenos);
238
239         let botonFin = document.createElement('button');
240         botonFin.textContent = "Finalizar Juego.";
241         botonFin.addEventListener("click", finalizarJuego);
242         finalizar.append(botonFin);
243     }
244
245     function presentarResultados () { //uso de google charts para presentar los datos recogidos del
juego
246
247         google.charts.load('current', { packages: ['corechart'] });
248         google.charts.setOnLoadCallback(() => {
249             let data = google.visualization.arrayToDataTable([
250                 ['Tipo', 'Cantidad'],
251                 ['Aciertos', aciertos],
252                 ['Fallos', fallos],
253                 ['Total Apuestas', apuestas],
254                 ['Puntuación', puntuacion]
255             ]);
256             let options = {
257                 title: 'Estadísticas del Juego',
258                 chartArea: { width: '50%' },
259                 hAxis: {
260                     title: 'Cantidad',
261                     minValue: Math.min(0, puntuacion)
262                 },
263                 vAxis: {
264                     title: 'Categoría'
265                 },
266                 colors : ['#727372', '#727372', '#727372', '#727372']
267             };
268
269             estadisticas.innerHTML = ""; //Limpio
270
271             let chart = new google.visualization.BarChart(estadisticas);
272             chart.draw(data, options);
273         });
274     }
275 </script>
276 </body>
277
278 </html>
279
```