Trabajo voluntario José López Arribas

Grado Superior en Desarrollo de Aplicaciones Web

Desarrollo Web en Entorno Cliente

Índice

1.	Contenido	3
2.	Ejecución	.15
3.	Problemas	.18
4.	Bibliografía	19





1. Contenido

El ejercicio ha sido resuelto en dos documentos HTML. De modo externo, han sido incluidos la funcionalidad (archivo JavaScript) y el estilo (dos archivos CSS, uno para cada HTML). Decidí mostrar una primera página de inicio que enlazara a la página de formulario. En esta última se carga la tabla de las citas.

El HTML de la <u>página de inicio</u> es simple. Cuenta con una cabecera con el nombre de la empresa a modo de título, una sección que contiene un enlace a la página de formulario y un pie de página con los datos de contacto.

```
18
19
       <h1>MedacDent</h1>
20
21
22
23
        <section class="container">
           <div class="row">
24
              <a id="nueva" href="/TrabajoClienteFormulario.html">Nueva cita</a>
25
26
27
28
29
        <footer class="container">
30
           <div class="row">
               <div class="col-xs-5 col-sm-6 col-md-7 col-lg-8">
32
              MedacDent
33
34
               <div class="col-xs-4 col-sm-3col-md-3 col-lg-2">
35
              Email: medacdent@gmail.com
36
37
38
              <div class="col-xs-3 col-sm-3 col-md-2 col-lg-2">
40
               Teléfono: 640319934
              </div>
41
42
43
44
        </footer>
45
```

La <u>página del formulario</u> tiene una pequeña cabecera que nos sitúa. La parte principal es una gran sección que embute al formulario. Con dos partes: 'Día de la cita' e 'Información del paciente'. En total, siete campos a rellenar por el usuario. Un pie de página con el contacto termina por definir la estructura.



```
cbody onload="cargarDatos()">
17
18 🗸
         <header class="container">
19 🗸
             <div class="row";</pre>
                 <div class="col-xs-6 col-sm-6 col-md-6 col-lg-6">
20 ~
21
                 <h1>MedacDent</h1>
22
23
24 ~
                 <div class="col-xs-6 col-sm-6 col-md-6 col-lg-6">
25
                 <h4>Nueva cita</h4>
26
27
28
         </header>
```

```
100
101
         <footer class="container">
102
            <div class="row">
103
                <div class="col-xs-5 col-sm-6 col-md-7 col-lg-8">
               MedacDent
104
105
106
107
                <div class="col-xs-4 col-sm-3 col-md-3 col-lg-2">
108
               Email: medacdent@gmail.com
109
110
               <div class="col-xs-3 col-sm-3 col-md-2 col-lg-2">
111
                 Teléfono: 640319934
112
113
114
115
116
         <script src="/JS/TrabajoCliente.js"></script>
117
118
119
     </body>
120
```



Se puede observar en las imágenes anteriores el uso de las clases **Bootstrap** para estilizar el proyecto y el responsive. He usado este framework en su versión última, la 5.3, junto a **CSS**. Los dos archivos HTML comparten el siguiente 'head' con los enlaces al framework, a las fuentes y al fichero de estilo, respectivamente.

El **JS** es largo. Contabilizo 16 funciones, un número considerable de líneas de código y un espacio que penaliza y excede el recomendado en las instrucciones de entrega.

La función 'inicio' nos da acceso desde la página de inicio al formulario.

'CargarDatos' es llamada desde el 'body' de la página formulario. Muestra la tabla con las citas o una tabla vacía, en función de si tenemos o no datos almacenados.

```
28
     // Llamada a la función cargarDatos()
29
     window.onload = cargarDatos;
38
     function cargarDatos() {
         let citas = JSON.parse(localStorage.getItem("citas")) || [];
39
40
41
         if(citas.length == 0) {
42
              mostrarTablaVacia();
43
           else {
44
             mostrarTabla(citas);
45
46
```



```
function mostrarTablaVacia() {
48
         let tabla = document.getElementById("tabla");
49
50
         //thead
51
         crearThead();
52
53
         //tbody
54
         let tbody = document.createElement("tbody");
55
         let tr = document.createElement("tr");
56
         let td = document.createElement("td");
57
58
         let tdTextoVacio = document.createTextNode("No hay citas.");
59
         td.appendChild(tdTextoVacio);
60
         tr.appendChild(td);
         tbody.appendChild(tr);
61
62
63
         tabla.appendChild(tbody);
64
```

```
function crearThead() {
 66
          let thead = document.createElement("thead");
 67
 68
          let thOrden = document.createElement("th");
 69
          let thTextoOrden = document.createTextNode("Orden de la cita");
 70
          thOrden.appendChild(thTextoOrden);
 71
          thead.appendChild(thOrden);
 72
 73
          let thDia = document.createElement("th");
 74
          let thTextoDia = document.createTextNode("Día de la cita");
 75
          thDia.appendChild(thTextoDia);
 76
          thead.appendChild(thDia);
 77
 78
          let thNombre = document.createElement("th");
 79
          let thTextoNombre = document.createTextNode("Nombre");
 80
          thNombre.appendChild(thTextoNombre);
 81
          thead.appendChild(thNombre);
 82
          let thApellido = document.createElement("th");
 83
 84
          let thTextoApellido = document.createTextNode("Apellidos");
 85
          thApellido.appendChild(thTextoApellido);
 86
          thead.appendChild(thApellido);
 87
 88
          let thDNI = document.createElement("th");
 29
          let thTextoDNIn = document.createTextNode("DNI");
 90
          thDNI.appendChild(thTextoDNIn);
 91
          thead.appendChild(thDNI);
92
 93
          let thTelefono = document.createElement("th");
 94
          let thTextoTelefono = document.createTextNode("Teléfono");
          thTelefono.appendChild(thTextoTelefono);
 95
          thead.appendChild(thTelefono);
 96
 97
 98
          let thNacimiento = document.createElement("th");
 99
          let thTextoNacimiento = document.createTextNode("Fecha de nacimiento");
100
          thNacimiento.appendChild(thTextoNacimiento);
101
          thead.appendChild(thNacimiento);
```



```
let thObservaciones = document.createElement("th");
let thTextoObservaciones = document.createTextNode("Observaciones");
thObservaciones.appendChild(thTextoObservaciones);
thead.appendChild(thObservaciones);

tabla.appendChild(thead);

tabla.appendChild(thead);
}
```

'CrearTabla' es una de las funciones principales. Es llamada desde un evento 'onsubmit' en el formulario. Crea y renderiza la tabla con las citas almacenadas. Incorpora otras funciones para almacenar la cita ('pecirCita'¹) y mostrarlas en una tabla ('mostrarTabla').

```
121
       function crearTabla() {
122
123
          let diaCita = document.getElementById("dia").value;
124
          let nombre = document.getElementById("nombre").value;
125
          let apellidos = document.getElementById("apellidos").value;
126
          let dni = document.getElementById("dni").value;
          let telefono = document.getElementById("telefono").value;
127
          let nacimiento = document.getElementById("nacimiento").value;
128
129
          let observaciones = document.getElementById("observaciones").value;
130
131
          let id = diaCita.concat(nombre);
132
133
134
          let citas = JSON.parse(localStorage.getItem("citas")) || [];
135
136
137
          pedirCita(citas, id, diaCita, nombre, apellidos, dni, telefono, nacimiento, observaciones);
138
139
140
141
          localStorage.setItem("citas", JSON.stringify(citas));
142
143
144
          mostrarTabla(citas);
145
          return false;
146
```

 $^{^{\}rm 1}$ Ver apartado Problemas, página . La función pedir Cita interfiere al modificar una cita.



```
tion pedirCita(citas, id, diaCita, nombre, apellidos, dni, telefono, nacimiento, observaciones) {
169
170
             let citaPedida = false:
             for(let i = 0; i < citas.length; i++) {</pre>
172
173
                   for(let atributo in citas[i]) {
   if(atributo == "id") {
173
174
175
                            if(id == citas[i][atributo]) {
                                 //alert("No puede pedir más de una cita.");
citaPedida = true;
176
177
178
179
180
181
                       if(atributo == "diaCita") {
                            if(!validarFecha(diaCita)) {
182
183
184
185
186
187
                       if(atributo == "nacimiento") {
                            if(!validarNacimiento(nacimiento)) {
188
189
190
191
193
194
             if(!citaPedida) {
                  let nuevaCita = {
196
197
198
199
                       diaCita: diaCita,
                       nombre: nombre,
                       apellidos: apellidos,
                       dni: dni,
telefono: telefono,
200
201
                       nacimiento: nacimiento,
                       observaciones: observaciones
```

'LimpiarCampos' elimina de la pantalla las letras de los campos de formulario. 'LimpiarBotonG' hace lo mismo con el botón 'Guardar los datos' que aparece el modificar una cita. En las siguientes tomas, muestro las únicas validaciones que hago en JS, la del día de la fecha ('validarFecha()') y la del día de nacimiento ('validarNacimiento()').



```
224
      function validarFecha(diaCita) {
225
          let fechaHoy = new Date();
226
227
          let dia = fechaHoy.getDate();
          let mes = fechaHoy.getMonth() +1;
228
229
          let año = fechaHoy.getFullYear();
230
          dia = dia<10 ? `0${dia}` : dia;</pre>
231
232
          mes = mes<10 ? `0${mes}` : mes;
233
234
           let fechaFormato = `${año}-${mes}-${dia}`;
235
236
          if(new Date(diaCita) < fechaHoy) {</pre>
237
               let info = document.getElementById("info");
238
               info.innerHTML = "Selecciona fecha correcta";
               info.classList.add("final");
239
240
               return false;
241
242
           return true;
243
244
      function validarNacimiento(nacimiento) {
245
           let fechaHoy = new Date();
246
247
           let dia = fechaHoy.getDate();
248
          let mes = fechaHoy.getMonth() +1;
249
          let año = fechaHoy.getFullYear();
250
251
          dia = dia<10 ? `0${dia}` : dia;
          mes = mes<10 ? `0${mes}` : mes;
252
253
           let fechaFormato = `${año}-${mes}-${dia}`;
254
255
256
           if(new Date(nacimiento) > fechaHoy) {
257
               let info = document.getElementById("info");
258
               info.innerHTML = "Selecciona fecha correcta";
               info.classList.add("final");
259
260
               return false;
261
           }
262
          return true;
```



```
268
      function mostrarTabla(citas) {
269
          limpiarTabla();
270
          let idCita;
271
          //thead
272
273
          crearThead();
274
275
          //tbody
276
          let tbody = document.createElement("tbody");
277
278
          for(let i = 0; i < citas.length; i++) {</pre>
279
              let tr = document.createElement("tr");
280
281
              //Columna de Orden de la cita
282
              let tdOrden = document.createElement("td");
283
               let tdOrdenTexto = document.createTextNode(i+1);
284
              tdOrden.appendChild(tdOrdenTexto);
              tr.appendChild(tdOrden)
285
286
              //Las demás columnas
287
              for(let atributo in citas[i]) {
288
                   if(atributo != "id") {
289
290
                       let td = document.createElement("td");
291
                       let tdTexto = document.createTextNode(citas[i][atributo]);
                       td.appendChild(tdTexto);
292
293
                       tr.appendChild(td);
294
                    else {
295
                       idCita = citas[i][atributo];
296
297
298
              //Botones modificar y elmiminar
299
300
              let botonModificar = document.createElement("button");
              botonModificar.innerHTML = "Modificar";
301
              botonModificar.setAttribute("id", "botonM")
302
              botonModificar.setAttribute("onclick", `modificar('${idCita}')`);
303
304
              tr.appendChild(botonModificar);
310
               tr.appendChild(botonEliminar);
311
312
               tbody.appendChild(tr);
313
               tabla.appendChild(tbody);
314
315
316
317
```



Con 'modificar' el usuario puede alterar sus datos almacenados de cita en base a su 'id' único. Va unida a 'guardarCambios', que es la función que guarda las modificaciones.

```
function modificar(id) {
331
332
          limpiarBotonG();
333
          let citas = JSON.parse(localStorage.getItem("citas")) || [];
334
335
336
          for(let i = 0; i < citas.length; i++) {</pre>
337
              if(id == citas[i]["id"]) {
338
339
                  idCita = citas[i]["id"];
340
341
                   let d = document.getElementById("dia");
                  d.value = citas[i]["diaCita"];
342
343
344
                  let nom = document.getElementById("nombre");
345
                  nom.value = citas[i]["nombre"];
346
347
                  let ape = document.getElementById("apellidos");
348
                  ape.value = citas[i]["apellidos"];
349
350
                  let dn = document.getElementById("dni");
351
                  dn.value = citas[i]["dni"];
352
                  let tel = document.getElementById("telefono");
353
354
                  tel.value = citas[i]["telefono"];
355
356
                  let nac = document.getElementById("nacimiento");
357
                  nac.value = citas[i]["nacimiento"];
358
359
                   let ob = document.getElementById("observaciones");
360
                  ob.value = citas[i]["observaciones"];
361
362
                  break;
363
364
365
```

```
let botonG = document.getElementById("botonG");
367
          let botonGuardar = document.createElement("button");
368
          botonGuardar.innerHTML = "Guardar los datos";
369
          botonGuardar.setAttribute("id", "botonGuardar");
370
          botonGuardar.addEventListener("click", function () {
371
372
              guardarCambios(idCita);
          })
373
          botonG.appendChild(botonGuardar);
374
375
376
```



```
function guardarCambios(id) {
397
398
          limpiarTabla();
399
400
          let citaInfo = false;
401
402
          let diaCitaModificada = document.getElementById("dia").value;
403
          let nombreModificado = document.getElementById("nombre").value;
404
          let apellidosModificado = document.getElementById("apellidos").value;
405
          let dniModificado = document.getElementById("dni").value;
406
          let telefonoModificado = document.getElementById("telefono").value;
407
          let nacimientoModificado = document.getElementById("nacimiento").value;
408
          let observacionesModificado = document.getElementById("observaciones").value;
409
          let idModificado = diaCitaModificada.concat(nombreModificado);
410
411
          let citas = JSON.parse(localStorage.getItem("citas")) || [];
412
413
          for(let i = 0; i < citas.length; i++) {</pre>
414
              if(id == citas[i]["id"]) {
415
                   citas[i]["id"] = idModificado;
416
                   citas[i]["nombre"] = nombreModificado;
417
                   citas[i]["apellidos"] = apellidosModificado;
418
                   citas[i]["dni"] = dniModificado;
419
                   citas[i]["telefono"] = telefonoModificado;
                   citas[i]["observaciones"] = observacionesModificado;
420
421
422
                   if(diaCitaModificada) {
423
                      if(!validarFecha(diaCitaModificada)) {
424
                           return false;
425
                      } else {
426
                          citas[i]["diaCita"] = diaCitaModificada;
427
428
```

```
430
                     (nacimientoModificado) {
431
                     if(!validarNacimiento(nacimientoModificado)) {
432
                     } else {
433
                           citas[i]["nacimiento"] = nacimientoModificado;
434
435
436
437
438
                   localStorage.setItem("citas", JSON.stringify(citas));
439
                   citaInfo = true:
440
                   break;
441
442
443
444
445
           limpiarBotonG();
446
447
          limpiarCampos();
448
449
          mostrarInfo(citaInfo);
450
451
          mostrarTabla(citas);
452
453
          return true;
454
455
```



'Eliminar' borra la cita almacenada.

```
471
      function eliminar(id) {
472
473
          let citaInfo = true;
474
475
          let citas = JSON.parse(localStorage.getItem("citas")) || [];
476
477
          for(let i = 0; i < citas.length; i++) {</pre>
              if(id == citas[i]["id"]) {
478
479
                   citas.splice(i, 1);
                   localStorage.setItem("citas", JSON.stringify(citas));
480
                   citaInfo = false;
481
                   limpiarBotonG();
482
483
                   limpiarTabla();
484
                   limpiarCampos();
485
                   cargarDatos();
486
                   mostrarInfo(citaInfo);
487
488
489
490
```

'MostrarInfo()'² añade una información en pantalla para alertar al usuario que sus datos de cita han sido modificados o eliminados.

```
function mostrarInfo(citaInfo) {
501
502
503
          let info = document.getElementById("info");
504
505
          if(info) {
              if(citaInfo == true) {
506
                   info.innerHTML = "Cita modificada";
507
                   info.classList.add("final");
508
              } else {
509
510
                   info.innerHTML = "Cita eliminada";
                   info.classList.add("final");
511
512
            else {
513
              console.log("elemento no encontrado")
514
515
516
```

 $^{^{2}}$ Ver apartado Problemas, página .. . La función 'mostrarInfo' no actúa todas las veces que es llamada.



Por último, las tres funciones de limpieza que eliminan de la pantalla algunos elementos.

```
523
      function limpiarCampos() {
          let diaLimpiar = document.getElementById("dia");
524
525
          diaLimpiar.value = "";
526
527
          let nombreLimpiar = document.getElementById("nombre");
528
          nombreLimpiar.value = "";
529
          let apellidosLimpiar = document.getElementById("apellidos");
530
          apellidosLimpiar.value = "";
531
532
533
          let dniLimpiar = document.getElementById("dni");
534
          dniLimpiar.value = "";
535
536
          let telefonoLimpiar = document.getElementById("telefono");
537
          telefonoLimpiar.value = "";
538
539
          let nacimientoLimpiar = document.getElementById("nacimiento");
540
          nacimientoLimpiar.value = "";
541
542
          let obLimpiar = document.getElementById("observaciones");
          obLimpiar.value = "";
543
544
552
        function limpiarTabla() {
553
             let tabla = document.getElementById("tabla");
554
             tabla.innerHTML = "";
```

```
555
```

```
562
      function limpiarBotonG() {
563
          let botonG = document.getElementById("botonG");
          botonG.innerHTML = "";
564
565
```



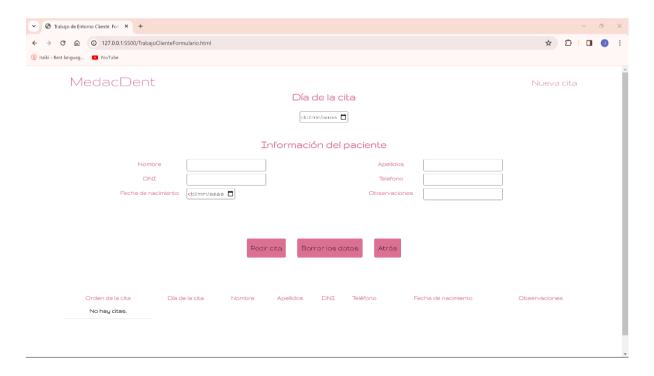


2. Ejecución

El usuario entra en la web y ve la siguiente pantalla:



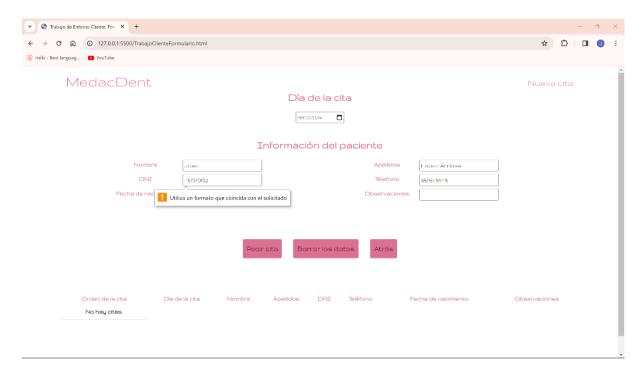
Entra en 'Nueva cita'.



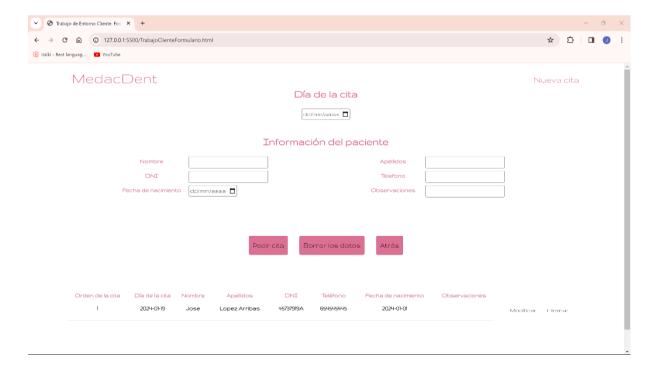




Observa el **formulario** con los datos para pedir la cita, **tres botones** (uno para confirmar, otro para borrar los datos y otro para volver a inicio), y la **tabla** de citas, donde aún no hay citas almacenadas. El usuario introduce sus <u>datos</u>.



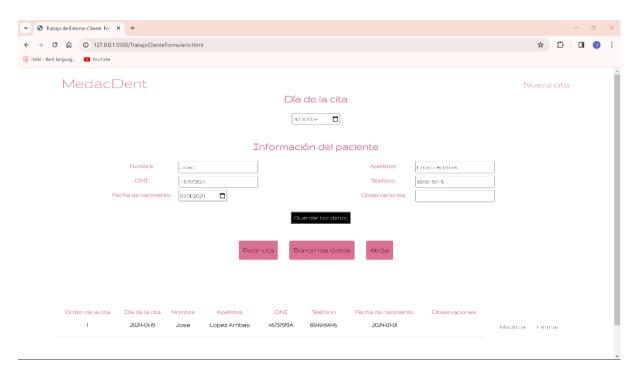
Parece que está teniendo problemas. Debe introducir un DNI con el formato correcto.



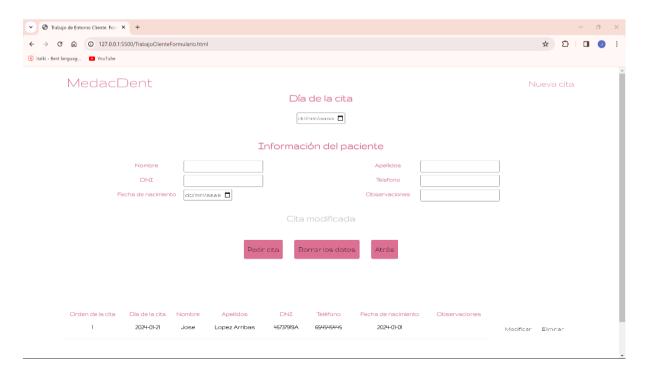




Lo hace y, automáticamente, su cita aparece en la tabla como almacenada. Resulta que quiere modificar el día de la cita. Para ello clica en '**Modificar'**. Los datos de la cita vuelven a los <u>campos del formulario</u> listos para ser cambiados. Tan solo debe seleccionar el día y **guardar los datos** en el <u>nuevo botón</u> que aparece.



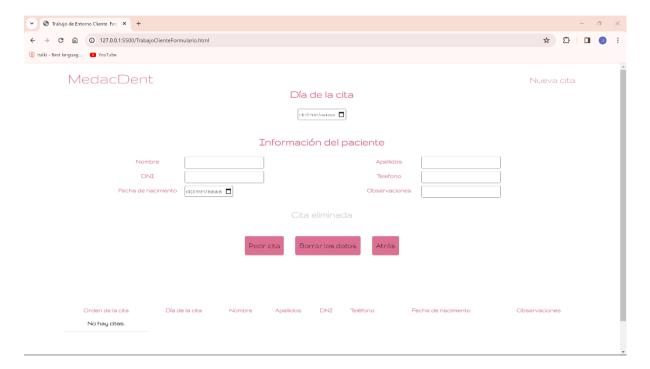
La cita del paciente ha sido modificada.







¿Y si se arrepiente y la quiere eliminar?



Cita eliminada. No hay citas.

3. Problemas

A pesar de que el programa funciona: almacena, valida, modifica y elimina; hay un tipo de interferencia en la función 'pedirCita' a la hora de modificar los datos. Creo que no debiera porqué haberlo. No obstante, no he acertado a resolver. Mi solución ha sido de pega, pues el problema sigue estando; solo eliminé un 'alert' que informaba de que un mismo id no podía pedir más de una cita para el mismo día. Hasta esa línea profunda de función distinta penetra la aplicación a la hora de modificar.

¿Por qué la <u>información</u> que quiero mostrar al modificar y al eliminar una cita solo aparece la primera vez que se hacen tales acciones? Tampoco lo he podido detectar.

Ambos defectos permanecen.



4. Bibliografía

- Temario asignatura 'Desarrollo Web en Entorno Cliente', temas 1-9.
- W3School. https://www.w3schools.com/
- MDN web docs. https://developer.mozilla.org/es/
- OpenAI. (2023). ChatGPT.

