

# Calendario mediante HTML

Isaac Rosales García

12 de octubre de 2020

## Índice

### 1. Introducción

En este trabajo se muestran los diagramas de flujos para ver el comportamiento de las funciones usadas para el uso del calendario

### 2. HTML

La base de HTML para crear el calendario se hizo mediante elementos `div`, el `div` que general, el cual contiene todos los elementos del calendario tiene como `id=container`. A partir de ahí se generan un `div` el cual contendrá a su vez `div` de los días de la semana. Para los demás elementos del mes se hizo mediante JavaScript por lo se menciona más adelante.

Además de los `div` mencionados, se agrego un modal con boton de cierre y un formulario en su interior el cual recibe tres `inputs` y un `submit`

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" href="assets/index.css" />
5
6     <title>Calendario</title>
7   </head>
8
9   <body>
10    <select id="month" onchange="changeMonth()"></select>
11    <div id="miModal" class="modal">
```

```

12     <div class="modalContent">
13         <div>
14             <span class="close" onclick="closeModal()"> &
                times; </span>
15             <h2>Agregar Evento</h3>
16         </div>
17         <p class="modalForm modalDate">Fecha</p>
18         <form id="calendarForm" class="modalForm">
19             <label>Titulo</label><br />
20             <input type="text" name="title" required /><br />
21             <label>Descripcion</label><br />
22             <input type="text" name="description" required/>
                <br />
23             <label>Participantes</label><br />
24             <input type="text" name="people"/><br /><br />
25             <div>
26                 <input type="submit" value="Guardar" class="
                    modalForm modalButton" />
27             </div>
28         </form>
29     </div>
30 </div>
31 <div id="container">
32     <div>
33         <h1 id="monthDate">Junio 2020</h1>
34     </div>
35     <div class="calendarContainerHeader">
36         <div>Lunes</div>
37         <div>Martes</div>
38         <div>Miercoles</div>
39         <div>Jueves</div>
40         <div>Viernes</div>
41         <div>Sabado</div>
42         <div>Domingo</div>
43     </div>
44 </div>
45 <hr>
46 <div id="dynamicHTML">
47 </div>
48 <script type="application/javascript" src="assets/
    index.js"></script>
49 </body>
50 </html>

```

### 3. CSS

A continuación se muestran las clases.

#### 3.1. #container

dado que en el HTML solo existe un elemento container, se decidió crear su clase de manea particular, para acomodar sus elementos internos se hizo mediante `display:flex` el cual permitirá tener un mejor control de sus elementos. Para que los elementos se muestre uno debajo de otro se hizo uso de `flex-direction:column` y `align-items`. Finalmente para que ocupe toda la pantalla se asigno una altura total de toda la vista `height:100vh`

```
1  #container ,
2  #dynamicHTML {
3      display: flex;
4      flex-direction: column;
5      align-items: center;
6      height: 100vh;
7  }
```

```
1  #container ,
2  #dynamicHTML {
3      display: flex;
4      flex-direction: column;
5      align-items: center;
6      height: 100vh;
7  }
8
9  .calendarContainer {
10     display: flex;
11     flex-direction: row;
12     justify-content: space-evenly;
13     text-align: center;
14 }
15
16 .calendarContainerHeader {
17     display: flex;
18     flex-direction: row;
19     justify-content: space-evenly;
```

```

20     text-align: center;
21 }
22
23 .calendarContainer > div {
24     height: 5rem;
25     width: 5rem;
26     background-color: white;
27     border: 1px solid #cdcdcd;
28     display: flex;
29     flex-direction: column-reverse;
30     justify-content: flex-end;
31     align-items: flex-end;
32     margin: 0px;
33     padding: 5px;
34     font-weight: bold;
35 }
36
37 .calendarContainerHeader > div {
38     height: 1rem;
39     width: 5rem;
40     background-color: #d1d1d1;
41     border: 1px solid #bbbbbb;
42     display: flex;
43     justify-content: space-evenly;
44     align-items: center;
45     margin: 0px;
46     padding: 5px;
47     font-weight: bold;
48 }
49
50 .dayClass {
51     color: #474b4e;
52 }
53
54 .lastMonth {
55     color: #bbbbbb;
56 }
57
58 .modal {
59     display: none;

```

```

60     position: fixed;
61     z-index: 99999;
62     width: 100vw;
63     height: 100vh;
64     background-color: rgba(0, 0, 0, 0.3);
65     left: 0;
66     top: 0;
67     padding-top: 100px;
68     overflow: hidden;
69 }
70
71 .modalContent {
72     display: flex;
73     flex-direction: column;
74     justify-content: space-evenly;
75     text-align: center;
76     background-color: #ebebeb;
77     margin: auto;
78     padding: 20px;
79     border: 1px solid #888;
80     width: 25em;
81     height: 28em;
82 }
83
84 .modalForm {
85     display: flex;
86     flex-direction: column;
87     justify-content: space-evenly;
88     text-align: left;
89     margin-left: 60px;
90     margin-right: 60px;
91     font-weight: bold;
92     background-color: #ebebeb;
93 }
94
95 .modalDate {
96     display: flex;
97     flex-direction: column;
98     justify-content: space-evenly;
99     text-align: right;

```

```

100     margin-right: 60px;
101     background-color: #ebebeb;
102 }
103
104 .modalButton {
105     flex-direction: row;
106     background-color: #595959;
107     color: white;
108     width: 8em;
109     height: 2em;
110     margin-left: 85px;
111     border-radius: 15px;
112 }
113
114 .close {
115     color: #aaa;
116     float: right;
117     font-size: 28px;
118     font-weight: bold;
119 }
120
121 .close:hover,
122 .close:focus {
123     cursor: pointer;
124     color: #000;
125     text-decoration: none;
126 }
127
128 .calendarContainer > div:hover {
129     background-color: #e3e3e3;
130     transition-delay: 50ms;
131     cursor: pointer;
132 }
133
134 .descripcion {
135     background-color: #24f0b6;
136     margin: auto;
137     padding: 3px;
138     border: 1px solid #117055;
139     border-radius: 10px;

```

```

140     box-shadow: -2px -2px 5px #32705f;
141     width: 4em;
142     height: 1em;
143 }
144
145 .descripcion:hover {
146     background-color: #21d9a5;
147     transition-delay: 50ms;
148     cursor: pointer;
149 }

```

## 4. Creación de días

El objetivo de esta función consiste en crear un elemento div

```

1  let createDays = () => {
2
3      let container = document.getElementById("container");
4
5      for (const weekEl in dayObjectBox) {
6          let div = document.createElement("div");
7          div.className = "calendarContainer";
8          container.appendChild(div);
9
10         dayObjectBox[weekEl].forEach((dayEl) => {
11             let divDay = document.createElement("div");
12             divDay.dataset.position = dayEl;
13             div.appendChild(divDay);
14             divDay.className = "dayClass";
15         });
16     }
17 };

```

