

## ACTIVIDAD 2

### CENTRAL DE MEDIDAS (Segunda Parte)

Partiremos del archivo creado en la primera parte de este ejercicio durante la unidad anterior:

- **centralMedidas.js**. Definición de la clase **CentralMedidas**.

Vamos a añadir un método a la clase **CentralMedidas** con las siguientes características:

Clase <b>CentralMedidas</b>																																																																																																			
Métodos																																																																																																			
Nombre	Parámetros	Devuelve	Descripción																																																																																																
<b>toHTML()</b>	-	cadena	Devuelve una cadena que representa una tabla HTML con la misma estructura que la propiedad <b>medidas</b> , añadiendo una columna adicional que representa la media de las medidas del mes para esa fila. Algo así: <table border="1" style="margin: 5px 0;"> <thead> <tr> <th></th> <th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>Media</th> </tr> </thead> <tbody> <tr> <td>SANTANDER</td> <td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td><td>10</td> </tr> <tr> <td>OVIEDO</td> <td>40</td><td>11</td><td>28</td><td>27</td><td>17</td><td>20</td><td>40</td><td>8</td><td>4</td><td>31</td><td>14</td><td>34</td><td>5</td><td>25</td><td>31</td><td>25</td><td>-5</td><td>20</td><td>27</td><td>-1</td><td>35</td><td>22</td><td>38</td><td>-8</td><td>18</td><td>-1</td><td>-2</td><td>21</td><td>7</td><td>16</td><td>18.2</td> </tr> </tbody> </table>		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Media	SANTANDER	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	OVIEDO	40	11	28	27	17	20	40	8	4	31	14	34	5	25	31	25	-5	20	27	-1	35	22	38	-8	18	-1	-2	21	7	16	18.2
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Media																																																																				
SANTANDER	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10																																																																				
OVIEDO	40	11	28	27	17	20	40	8	4	31	14	34	5	25	31	25	-5	20	27	-1	35	22	38	-8	18	-1	-2	21	7	16	18.2																																																																				

Ahora ya estamos listos para incorporar la parte visual. Para ello contamos con el siguiente formulario, definido en **temperaturas.html**, con la hoja de estilos correspondiente.

#### Control de temperaturas

Ciudad

Medidas

Manual  Aleatorio

La funcionalidad relacionada con los eventos se implementará en el archivo **medidasFormulario.js**.

El formulario nos va a permitir añadir nuevas ciudades (campo **Ciudad**) y sus medidas (campo **Medidas**).

- El texto que escribamos en el campo de Ciudad se convertirá a mayúsculas de manera automática, al soltar cada tecla.
- En función de los botones de radio se podrán introducir las medidas de dos maneras:
  - **Manual**. Se introducirán 30 valores, separados entre comas.
  - **Aleatorio**. Los 30 valores se generarán de manera automática (con valores en el rango permitido), pasando el campo a ser de sólo lectura:

#### Control de temperaturas

Ciudad

Medidas

Manual  Aleatorio

#### Control de temperaturas

Ciudad

Medidas

Manual  Aleatorio

- Al pulsar en el botón **Guardar**, se deberán realizar las siguientes validaciones:
  - El campo ciudad tiene contenido y no está repetido (no existe en la tabla).
  - El campo de medidas contiene 30 valores separados por comas, y todos los valores son enteros en el rango de temperaturas permitido.
    - Busca en internet como parsear todo el array a enteros de forma fácil.

En caso de haber algún error, se mostrará en la capa con identificador **errores** (que debe ocultarse una vez que se han subsanado los errores):

La ciudad ya existe

Si las validaciones son correctas, se mostrará en la capa con id **tabla-medidas** una tabla con las ciudades que se han insertado hasta el momento, incluyendo la ciudad actual. Por ejemplo, en la siguiente captura se han insertado dos ciudades:

### Control de temperaturas

Ciudad

Medidas

Manual  Aleatorio

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	Media
SANTANDER	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
OVIEDO	40	11	28	27	17	20	40	8	4	31	14	34	5	25	31	25	-5	20	27	-1	35	22	38	-8	18	-1	-2	21	7	16	18.2

Además, la capa con id **temperatura-media** mostrará la temperatura media de todas las ciudades:

Temperatura media: 14.1

**NOTA:** Podemos establecer el estilo de un elemento del DOM accediendo a su propiedad `style` y a continuación especificando el nombre de la propiedad CSS (en notación `camelCase`) a la que queremos dar valor. Por ejemplo, si queremos ocultar un elemento podemos hacerlo mediante el código siguiente:

```
document.getElementById("elemento").style.display="none"
```

Si queremos mostrar el mismo elemento podemos usar la sentencia

```
document.getElementById("elemento").style.display="block" | "inline"
```

- Al pulsar el botón **Borrar** se consultará el campo Ciudad. En caso de no estar vacío y contener un nombre de ciudad existente, se borrará la fila correspondiente, actualizando la tabla y la temperatura media. Si las validaciones no son correctas, se mostrará el error en el campo de error.