# UD8 - Almacenamiento de datos en el lado cliente

Relación de Ejercicios 1

short line

### **EJERCICIO 1:**

Crea un fichero que se llame “cookies.js” y que permita trabajar con las cookies de manera que puedas crear, borrar y consultar una cookie. Consulta los métodos tratados en clase y en los apuntes y recuerda que el método borrar(removeCookie) no ha sido creado aún así que deberás crearlo tú mismo. Las operaciones son:

* setCookie(cname, cvalue, exdays) --- Nelson
* getCookie(cname) --- Pablo
* removeCookie(cname) --- Juan Jose

Implemente una página web sencilla donde podamos probarlos, para ello, debemos tener varios botones para cada funcionalidad, y un campo de texto donde vayamos viendo el estado de las cookies (cuando se añaden o cuando se eliminan).

### **EJERCICIO 2:**

Realice una web sencilla en la que haya al menos 1 campo que debe de validar una determinada condición. La lógica del programa debe de implementar: cada vez que el usuario trate de enviar el formulario y haya algún error, una variable contador (almacenada en una cookie o webstorage) se incrementará y se mostrará.

* El resultado del número de intentos se reflejará en un campo de texto que se encontrará al final del formulario.
* Si el usuario sale de la web y vuelve a entrar, el campo de texto mostrará el número almacenado de reintentos en las sesiones anteriores.
* Junto al campo de texto habrá un botón que, al pulsarlo, permitirá reiniciar el valor de la cookie/webStorage a 0 (reiniciar el numero de reintentos).

Seleccione qué tecnología va a usar: ¿cookies o webstorage?

### 

### 

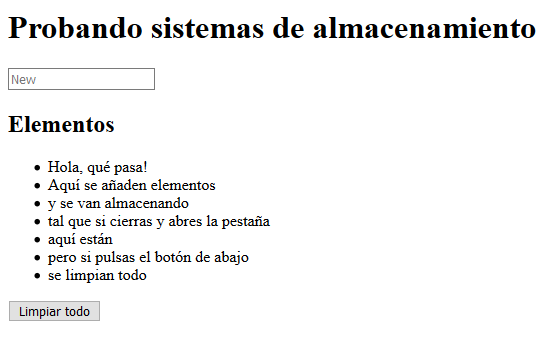
### 

### **EJERCICIO 3:**

Partiendo del siguiente código HTML:

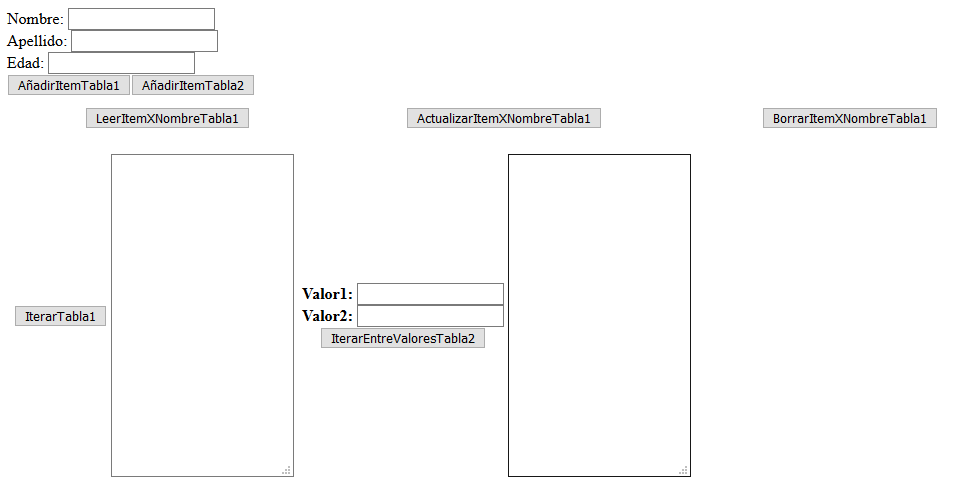
|  |
| --- |
| <div class="small-container">  <h1>Probando sistemas de almacenamiento</h1>    <form>  <input id="item" type="text" placeholder="New">  </form>    <h2>Elementos</h2>  <ul></ul>    <button>Limpiar todo</button>  </div> |

Y debemos de implementar la siguiente lógica haciendo uso de WebStorage (uno de ellos, elija cual es el que más encaja para esta funcionalidad). Fíjese en el siguiente ejemplo:



## **Ejercicio 4 NO SE HACE**

En el siguiente ejercicio debemos de implementar haciendo uso de **indexedDB** una base de datos con 2 object stores el cual debe de manejar objetos con 3 atributos **{nombre, apellido, edad}** y se deben implementar las siguientes operaciones:



El código del esqueleto HTML es:

|  |
| --- |
| <html lang="en"> <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <meta http-equiv="X-UA-Compatible" content="ie=edge">  <title>Document</title> </head> <body>   Nombre: <input type="text" id="nombre"> </br>  Apellido: <input type="text" id="apellido"> </br>  Edad: <input type="text" id="edad"> </br>   <table style="width:100%">  <td>  <tr><button id="btnAnadirItemBD">AñadirItemTabla1</button></tr>  <tr><button id="btnAnadirItemBD2">AñadirItemTabla2</button></tr>  </td>  <td>  <th><button id="btnLeerBD">LeerItemXNombreTabla1</button></th>  <th><button id="btnActualizarValorBD1">ActualizarItemXNombreTabla1</button></th>  <th><button id="btnBorrarValorBD1">BorrarItemXNombreTabla1</button></th>  </td>    </table>   </br>   <table>  <td>  <th><button id="iterar1">IterarTabla1</button></th>  <th><textarea id="textArea1" rows="20" cols="20"></textarea></th>  </td>  <td>  <th>  Valor1: <input type="text" id="valor1"> </br>  Valor2: <input type="text" id="valor2"> </br>  <button id="iterar2">IterarEntreValoresTabla2</button>  </th>  <th><textarea id="textArea2" rows="20" cols="20"></textarea></th>  </td>  </table> <script> |

* **Descripción de las operaciones:**

1. **AñadirItemTabla1:** Sobre la ObjectStore 1 añadir objetos que contengan los campos nombre, apellido y edad.
2. **AñadirItemTabla2:** Sobre la ObjectStore 2 añadir objetos que contengan los campos nombre, apellido y edad.
3. **LeerItemXNombreTabla1:** Mostrar por un “dialog” un elemento que esté contenido en el Object Store 1 y que coincida con el campo nombre introducido.
4. **ActualizarItemXNombreTabla1:** Ateniendonos al campo nombre, este botón debe de ser capaz de actualizar el valor de un registro de el ObjectStore 1.
5. **BorrarItemXNombreTabla1:** Igual que el Leer y el Actualiza, ahora debemos de borrar el registro por el nombre descrito del ObjectStore 1.
6. **IterarTabla1:** Mediante un cursor, se debe de mostrar todos las filas/registros del objectStore 1 en el TextArea1:
7. **IterarEntreValoresTabla2:** Igual que el anterior y esta vez sobre el ObjectStore2, debemos de aplicar un cursos y mostrar todos sus resultados pero estableciendo como condición que los nombres de los items devueltos cumplan la condición de estar comprendidos entre los 2 valores/letras contenidas.