# JS en la Web: Almacenando datos en el navegador

## Conclusión 1: Trabajando con fechas

* Utilizar el componente de calendario
* Manipular fechas
* Destructurar objetos en JavaScript

## Conclusión 2: LocalStorage

* Guardar las tareas en el navegador utilizando SessionStorage
* Almacenar las tareas con LocalStorage
* Transformar objetos en string utilizando JSON.stringify()

## Conclusión 3: Renderizando la lista

* Recorrer el arreglo con forEach() de las tareas guardadas en LocalStorage
* Utilizar pipe
* Convertir las tareas almacenadas en formato string a objetos con JSON.parse()

## Conclusión 4: Ordenando tareas

## Conclusión 5: Finalizando el proyecto

## Anexos:

* <https://momentjs.com/>
* <https://cdnjs.com/libraries/moment.js>
* <https://developer.mozilla.org/es/docs/Web/API/Web_Storage_API>
* <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON>

## Código:

* <input type="datetime-local">
* <script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.29.4/moment.min.js"></script>
* <input type="datetime-local" data-form-date placeholder="YYYY/MM/DD"/>
* moment(date).format("DD/MM/YYYY");
* import { addTask } from "./components/addTask.js";
* document.querySelector('[data-form-btn]')
* .addEventListener('click', function, let o const);
* export const addTask = (evento) => {}
* .appendChild(let o const);
* evento.preventDefault();
* const taskList = JSON.parse(localStorage.getItem('tasks')) || [];
* localStorage.setItem('tasks', JSON.stringify(taskList));
* input.value;
* .value;
* document.createElement(‘TagHTML’);
* .classList.add(‘TagHTML’);
* const taskObj = {

value,

dateFormat

}

* .innerText
* .innerHTML
* Return
* .toggle(‘TagHTML’);
* event.target;
* export default
* event.target.parentElement;
* .remove();
* .forEach((elemento) => {})

## Vocabulario:

* API de navegador
* Local storage
* Sesión storage
* desestructuración de su objeto

## Capturas:

|  |
| --- |
|  |