

Demo SPA básica con operaciones CRUD

A partir de la entidad “tarea” con las propiedades “tareaid”, “titulo”, “duración” (entero) y “completada” (boolean), vamos a crear una página SPA mínima en la que realizar las operaciones CRUD.

La estructura del proyecto cuenta con los siguientes archivos:

- datos.js: contiene un array con datos de prueba. Array de objetos JSON con el formato indicado.
- gestor.js: contiene el código js de la aplicación.
- index.html: punto de entrada, incluye un contenedor donde mostrar dinámicamente el código HTML.
- estilos.css: estilos mínimos, tabla y formulario.

El aspecto de la aplicación es el siguiente:

Vista listado de tareas

Gestor de tareas

ID	Título	Duración	Completada	Crear
1	Preparar presentación	45	false	<button>Ver</button> <button>Borrar</button>
2	Enviar informe	20	true	<button>Ver</button> <button>Borrar</button>
3	Revisión de código	60	false	<button>Ver</button> <button>Borrar</button>

Vista formulario edición y creación

Gestor de tareas

Titulo

Duración

Completada

Criterios de codificación

Emplea un módulo para javascript.

No hace falta definir clases envoltorio para las tareas, trabajaremos directamente sobre los objetos JSON.

El gestor de tareas en una clase NO una función autoinvocada.

El gestor de tareas genera el HTML dinámicamente y se encarga de gestionar las operaciones CRUD.

Diferenciamos el código de negocio del código de interacción con la interfaz, es decir:

- Tenemos una clase para gestionar las tareas.
- Tenemos métodos auxiliares para gestionar los eventos.
- Usamos el evento “load” para inicializar la aplicación.

Pasos que seguir

Vamos a organizarnos

- Necesito conocer el modelo, por lo que creo datos de prueba en “datos.js”
- Defino el maquetado de las dos vistas que tengo: listado y formulario.
 - Defino los estilos, van a ser los mismos de la versión final.
 - Define como vas a manejar los metadatos, recomiendo dos:
 - data-accion: Indica la operación asociada a los botones: ver, borrar, crear y guardar.
 - data-entidadId: Indica la tarea asociada.

NOTA: usamos “data” pero se puede hacer mediante “class” definiendo nombres de clase equivalentes y posteriormente trabajar con cadenas (método antiguo).

- Creo la estructura del proyecto.
 - Enlazo los ficheros y los pruebo.
- Defino la clase gestor tareas incluyendo sus propiedades y métodos.
 - Comento lo que hace cada elemento.
 - Piensa, necesitas:
 - Guardar las tareas
 - Calcular el tareald para las nuevas tareas
 - El html del listado
 - El html del formulario (tengo creación y edición, ¿cómo lo apaño?)
 - Borrar una tarea
 - Crear una tarea
 - Editar una tarea
- Defino el código auxiliar, aunque no lo implemento. Necesitas al menos:
 - Función encargada de asignar manejadores a eventos.
 - Función encargada de manejar eventos, en este caso sólo el click.

- Defino las variables globales y el evento load arranca la aplicación.

Llegados a este punto tengo la estructura, pero ningún código funcional. ¿Por dónde comienzo? => Por el comienzo, el listado de tareas.

- Codifico el listado de tareas.
- Codifico el borrado.
- Codifico el formulario.

Ten en cuenta que después de cada operación que modifique los datos hay que recargar el listado.