Entrega 4 - MVC Películas AJAX

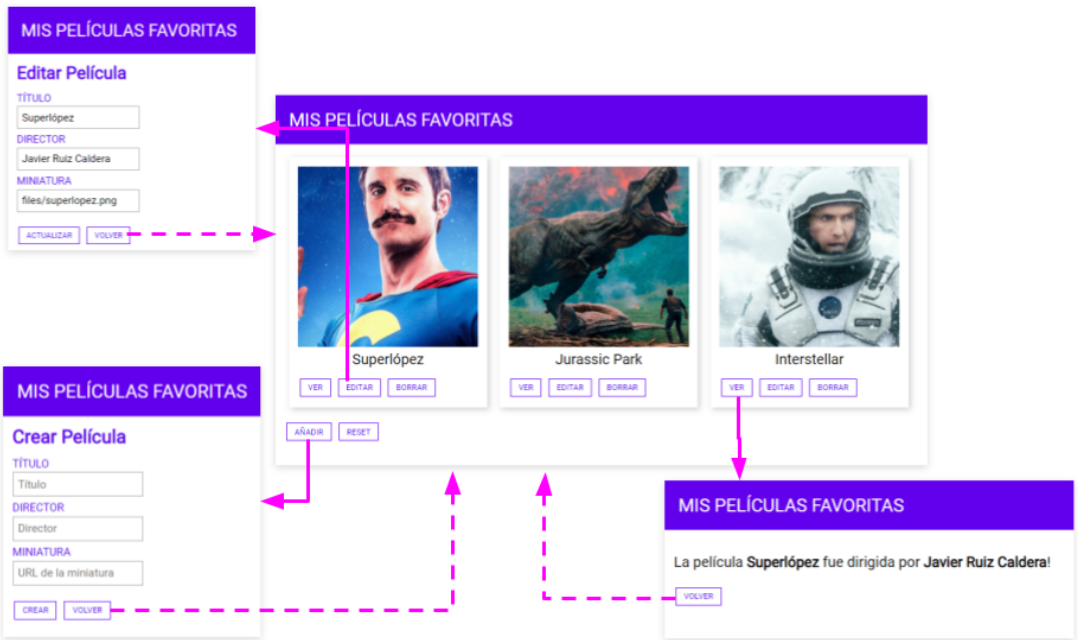
Versión: 20 de Febrero de 2020

Objetivos

- Practicar HTML, CSS y JS y afianzar el concepto de MVC (Modelo-Vista-Controlador).
- Utilizar APIs REST para leer y escribir información en servidores web

Descripción de la práctica

Esta entrega es una modificación de la entrega 2 (MVC películas) en la que se hace uso de la API de myjson (http://myjson.com/) para persistir la base de datos de películas. El resultado obtenido será el mismo que en la entrega 2, sólo que en lugar de almacenar la información de las películas utilizando la API de localStorage del navegador, se almacenará remotamente en el servidor de myjson y estará accesible a través de la API REST que proporciona este servicio.



En el código proporcionado sólo está implementada la funcionalidad de listar las películas existentes y editar película. El alumno debe implementar las funcionalidades restantes (crear, mostrar, eliminar y reiniciar), así como las funciones que permiten leer y actualizar la información de myjson (`getAPI` y `updateAPI`).

Acerca de myjson

El servicio myjson permite almacenar información en formato JSON para poder acceder a ella desde cualquier cliente HTTP. Las acciones que pueden realizarse sobre los endpoints disponibles son las siguientes:

Método HTTP	URL	Descripción
POST	https://api.myjson.com/bins	Crea un almacén nuevo para nuestra aplicación. Devuelve la URL que debemos utilizar como endpoint para guardar la información de nuestra aplicación. Por ejemplo: https://api.myjson.com/bins/xxxxx
GET	https://api.myjson.com/bins/xxxxx (la creada con el método POST)	Lee la información guardada en el endpoint creado
PUT	https://api.myjson.com/bins/xxxxx (la creada con el método POST)	Sobrescribe la información guardada en el endpoint creado con el JSON que se le pasa en el campo "body" de la petición HTTP

Descargar el código del proyecto

El proyecto debe clonarse en el ordenador desde el que se está trabajando:

```
$ git clone https://github.com/CORE-2020/Entrega4_ajax
```

A continuación se debe acceder al directorio de trabajo y abrir el fichero `index.html` con el editor de la elección del alumno.

```
$ cd Entrega4_ajax
```

El fichero index.html contiene el código de la aplicación. Incluye tanto el HTML de la página, como el CSS y el código JavaScript que implementa la lógica de la aplicación siguiendo el patrón MVC. La explicación del modelo y de cada vista y controlador puede encontrarse en el enunciado de la entrega 2. Para elaborar la solución de esta entrega puede reutilizar todo el código que considere conveniente de su solución de la entrega 2. Las funcionalidades a desarrollar son las mismas, sustituyendo el acceso a localStorage por la función correspondiente de comunicación con la API (`postAPI` , `getAPI` o `updateAPI`). Es decir, cambia la parte correspondiente al modelo (M) de la implementación de MVC realizada en la entrega 2.

Cabe mencionar que, respecto al código de la entrega 2, se ha añadido un controlador nuevo (`initContr`), que se encarga de comprobar si el usuario ha creado ya un endpoint en myjson. Si no lo ha hecho, llama a la función `postAPI` que se encarga de crear dicho endpoint con las películas iniciales y lo guarda en la clave "URL" de localStorage. De esa manera, la próxima vez que el usuario acceda a la página, se podrá comprobar que el endpoint ya está creado y la información de las películas será accesible a través de él.

Tareas

En primer lugar, para no repetir el código de lectura/escritura en myjson a lo largo de la aplicación, debe implementar este código en las funciones `getAPI` y `updateAPI`. Debe completar estas funciones para que hagan lo siguiente (se recomienda hacer uso de la función `fetch` de JavaScript):

- **getAPI**: Realiza una petición asíncrona a la URL almacenada en `localStorage.URL` y devuelve la información recibida.
- **updateAPI**: Realiza une petición asíncrona a la URL almacenada en `localStorage.URL` y escribe la información que se le pasa en el argumento "peliculas"

Una vez implementadas estas funciones, se pide modificar el código proporcionado para completar las cinco funcionalidades que faltan. Haga uso de las funciones `getAPI` y `updateAPI`. Como se ha dicho, puede reutilizar todo el código que considere conveniente de su solución de la entrega 2:

- **Show**: Mostrar información sobre la película.
- **New**: Mostrar el formulario de añadir una nueva película.
- **Create**: Añade una nueva película al modelo.
- **Delete**: Elimina una película del modelo. Debe pedir la confirmación del usuario.
- **Reset**: Restaura el modelo al estado inicial, guardando las tres películas iniciales en localStorage.

Prueba de la práctica

Para ayudar al desarrollo, se provee una herramienta de autocorrección que prueba las distintas funcionalidades que se piden en el enunciado. Para utilizar esta herramienta debes tener node.js (y npm) (https://nodejs.org/es/) y Git instalados.

Para instalar y hacer uso de la herramienta de autocorrección en el ordenador local, ejecuta los siguientes comandos en el directorio del proyecto:

```
$ npm install -g autocorector     ## Instala el programa de test
$ autocorector                    ## Pasa los tests al fichero a entregar
.........................         ## en el directorio de trabajo
... (resultado de los tests)
```

También se puede instalar como paquete local, en el caso de que no se dispongas de permisos en el ordenador desde el que estás trabajando:

```
$ npm install autocorector     ## Instala el programa de test
$ npx autocorector             ## Pasa los tests al fichero a entregar
.........................      ## en el directorio de trabajo
... (resultado de los tests)
```

Se puede pasar la herramienta de autocorrección tantas veces como se desee sin ninguna repercusión en la calificación.

Instrucciones para la Entrega y Evaluación.

Una vez satisfecho con su calificación, el alumno puede subir su entrega a Moodle con el siguiente comando:

```
$ autocorector --upload
```

o, si se ha instalado como paquete local:

```
$ npx autocorector --upload
```

La herramienta de autocorrección preguntará por el correo del alumno y el token de Moodle. En el enlace https://www.npmjs.com/package/autocorector se proveen instrucciones para encontrar dicho token.

RÚBRICA: Se puntuará el ejercicio a corregir sumando el % indicado a la nota total si la parte indicada es correcta:

- **25%:** Se muestran correctamente las películas recibidas a través de la API.
- **25%:** Se crean películas nuevas a través de la API.
- **25%:** Se borran películas existentes a través de la API.
- **25%:** Se resetea la API para que devuelva las películas iniciales.

Si pasa todos los tests se dará la máxima puntuación.