



Interactuando con formularios

Práctica integradora

Objetivo

Validar un formulario del lado del cliente es uno de los desafíos más típicos de JavaScript y se puede hacer de diversas maneras, con alternativas de código distintas, que pueden funcionar perfectamente en todos los casos.

Normalmente el proceso es siempre el mismo: acceder campo a campo del formulario y comprobar que los valores introducidos tienen la forma que estamos esperando. En caso que todos los campos de formulario estén correctos, entonces se realiza el envío hacia el servidor donde se tenga que tratar el formulario. En el caso contrario, se le informa al usuario qué campo no está cumpliendo con los requisitos establecidos en los diferentes controles y no se envía hasta que esté correcto.

En esta ejercitación tendremos la responsabilidad de asegurarnos que se completen todos los controles de formulario requeridos y en el formato correcto antes de enviar los datos al servidor.

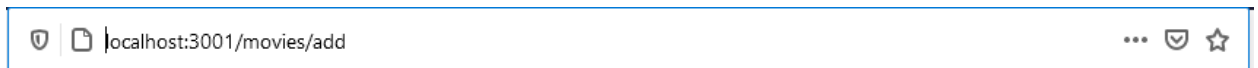
¡Buena suerte! 🤙👍✨

Micro desafío - Paso 1:

Utilizaremos de base el siguiente [proyecto creado con Express](#) (recordemos instalar todas las dependencias del proyecto, ejecutando la instrucción **npm install** 😊). Además, aprovecharemos la base de datos [movies db](#) (no olvidemos activar el servicio de MySQL en nuestro equipo). De esta manera, todo funcionará correctamente.

Para activar el servidor debemos ejecutar el comando **npm test** y finalmente en el navegador ejecutar: <http://localhost:3001/>. El objetivo de esta práctica es validar completamente el formulario. La validación deberá seguir los siguientes requerimientos:

- Deberemos tomar como punto de partida el archivo de JavaScript: **/public/js/moviesAddValidation.js**, y debemos vincularlo con la vista: **/views/moviesAdd.ejs**.
- Comprobar que el archivo quedo bien vinculado, para ello, desde el browser debemos ejecutar:



- Apenas cargue el formulario, debemos posicionarnos o tener el foco en el campo **"Título de la película"** de forma automática.
- Todos los campos son obligatorios, ninguno puede estar vacío.
- Si se trata de enviar el formulario (presionando el botón **"Agregar"**) estando todos o algún campo vacío —o que no cumpla con los requerimientos—, no será posible enviar el formulario; y a aquellos campos **input** o **select** con error se les deberá agregar la clase **"is-invalid"**. Esta ya se encuentra creada en el archivo: **/public/css/style.css**.
 - Las validaciones no solo deberán hacerse al enviar el formulario si no también al momento en el que el visitante interactúa con cada campo (**validación on-time**).
- De igual manera, aquellos campos que poseen error deberán tener un texto que especifique el tipo de error. Dicho texto deberá estar presente en el elemento con clase **"is-invalid"**. Esta ya se encuentra creada en el archivo: **/public/css/style.css**.

- El formulario contará con validaciones especiales para los siguientes campos:
 - **Calificación y Premios:** deberá validar que el valor ingresado esté comprendido entre cero (0) y diez (10).
 - **Duración:** deberá validar que el valor ingresado esté comprendido entre 60 y 360 minutos.
- Si los campos cumplen con los requerimientos exigidos, debemos hacer lo siguiente:
 - A todos los campos **input o select** se les deberá agregar la clase **"is-valid"**, como constancia que ese campo pasó la validación. Esta ya se encuentra creada en el archivo: **/public/css/style.css**.



Micro desafío - Paso 2:

- Si existen errores en el registro de la película, ya que el usuario no cumplió con los requerimientos en los controles del formulario, entonces en la vista **/views/moviesAdd.ejs** se le deben mostrar al usuario una lista con los errores. Para ello se recomienda capturar en el archivo **/public/js/moviesAddValidation.js** elemento `<ul class="errores">`, ubicado en el archivo: **/views/moviesAdd.ejs**. Al mismo se le deberá agregar la clase **"alert-warning"**. Esta ya se encuentra creada en el archivo: **/public/css/style.css**.
- Si todos los datos cumplen con los requerimientos exigidos, se le debe enviar al usuario un mensaje indicando **"La película se guardó satisfactoriamente"**.



Bonus Track:

Si logramos realizar toda la práctica, una buena idea es replicar el proceso para las vistas: **moviesEdit.ejs** y **moviesDetail.ejs**. Podemos ejecutar sobre ellas todos los ajustes que deseemos. ¡A usar la imaginación!

Conclusión

Con esta práctica pudimos comprobar la ventaja de manipular el **DOM**, así como dominar lo referido al manejo de los eventos, por medio de los cuales podemos controlar qué es ejecutado por el usuario en nuestros formularios.

La validación en el lado del cliente es una verificación inicial y una característica importante para garantizar una buena experiencia de usuario. Mediante la detección de datos no válidos en el lado del cliente, el usuario puede corregirlos de inmediato. Si el servidor lo recibe y, a continuación, lo rechaza, se produce un retraso considerable en la comunicación entre el servidor y el cliente que insta al usuario a corregir sus datos.

Sin embargo, la validación en el lado del cliente **no debe considerarse** una medida de seguridad exhaustiva. Nuestras aplicaciones siempre deben realizar comprobaciones de seguridad de los datos enviados por el formulario **en el lado del servidor**. La validación en el lado del cliente es demasiado fácil de evitar, por lo que los usuarios malintencionados pueden enviar fácilmente datos incorrectos a nuestro servidor.

¡Hasta la próxima!