

Laboratorio 3

Ejercicio 1 – Mostrando comentarios

- 1 - Crear un elemento `li` para cada review que tenga el producto.
- 2 - Mostrar el valor de la propiedad `stars` de cada review dentro de las etiquetas `` de nuestro elemento `<blockquote>`.
- 3 - Mostrar el valor de la propiedad `stars` entre las etiquetas de `` y `<cite>`.
- 4 - Mostrar el valor de la propiedad `author` entre las etiquetas `<cite>` a la izquierda del -.

Ejercicio 2 – Creando el formulario de comentario

- 1 - Asignar la propiedad `stars` de nuestro review utilizando la Directiva `ng-model`.
- 2 - Asignar la propiedad `body` al elemento `<textarea>` utilizando la Directiva `ng-model`.
- 3 - Asignar la propiedad `author` al elemento `<input>` de tipo email utilizando la Directiva `ng-model`.

Ejercicio 3 – Visualización de comentario

- 1 - Mostrar la propiedad `stars` del elemento review actual dentro de las etiquetas ``.
- 2 - Mostrar la propiedad `body` del elemento review actual en el espacio en blanco.
- 3 - Mostrar la propiedad `author` del elemento review actual, dentro de las etiquetas `<cite>` a la derecha del -.

Ejercicio 4 – Creando nuestro ReviewController

- 1 - Crear un controlador llamado `ReviewController`.
- 2 - Crear una propiedad llamada `review` e inicializarla con un objeto vacío.
- 3 - Crear una función vacía llamada `addReview`.
- 4 - Queremos enviar como parámetro el producto al cual le asignaremos nuestro comentario, modificar la función `addReview` acorde.
- 5 - Cuando la función `addReview` sea invocada con un producto como parámetro, debemos agregarle un review a su array de reviews, modificar la función para tal fin.
- 6 - Resetear nuestra propiedad `review` con un objeto vacío luego de ser agregado.

Ejercicio 5 - Utilizando nuestro controlador ReviewController

- 1 - Asignar nuestro controlador `ReviewController` al elemento `<form>` utilizando el alias de `reviewCtrl`.

2 - Agregar el atributo ng-submit y hacerlo llamar a nuestra función addReview recordando pasar nuestro producto como parámetro.

3 - Agregar como prefijo de los tres atributos ng-model el alias de nuestro controlador.

4 - Para volver a mostrar los valores, agregar el alias como prefijo de todas nuestras expresiones.

Ejercicio 6 - Validando nuestro formulario

1 - Desactivar la validación HTML5 por defecto.

2 - Marcar los campos stars y author como requeridos utilizando el atributo required.

3 - Prevenir el submit de formulario si el mismo no es válido.

Ejercicio 7 - Estilizando nuestro formulario

1 - Para los elementos con las clases de ng-invalid y ng-dirty asignar la propiedad de border-color con el valor red.

Ejercicio 8 - Mostrando fecha de creación






1 - Antes de guardar nuestro review en el array de reviews de nuestro objeto, crear una propiedad en el review a guardar llamada createdOn y asignarle el valor Date.now().

2 - Mostrar la propiedad createdOn recientemente creada de forma tal que el resultado final sea -author on createdOn.






3 - Utilizar el filtro de fecha para nuestra propiedad createdOn.

[- Top](#)






Comentario sobre el Laboratorio de la Clase








☐








☐



☐



☐



☐

Guardar Comentario