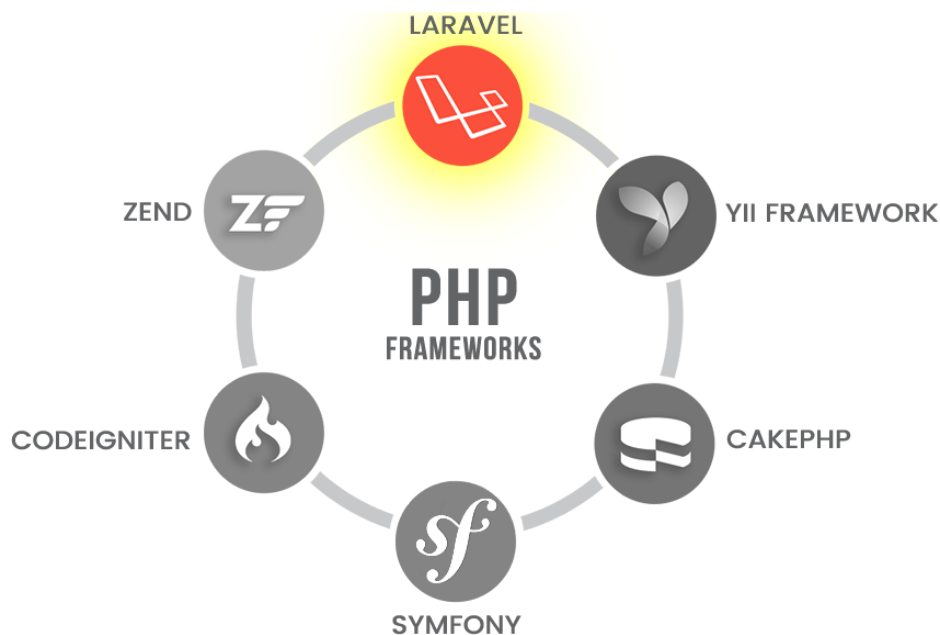


## unidad didáctica 8

# Laravel - formularios y validación (actividades)



## 1. ejercicios

- 1. 1. ejercicio 1
- 1. 2. ejercicio 2 (opcional)
- 1. 3. ejercicio 3

## 2. bibliografía

El uso de estos materiales está sujeto a una licencia Creative Commons [CC BY-NC](#).

Material extraído de <https://nachoiborraies.github.io/laravel/>

# 1. ejercicios

## 1.1. ejercicio 1

Sobre el proyecto **blog** de la sesión anterior, vamos a añadir estos cambios:

- Crea un formulario para dar de alta nuevos posts, en la vista `resources/views/posts/create.blade.php`. Añade un par de campos (un texto corto y un texto largo) para rellenar el título y el contenido, y como autor o usuario del post de momento deja uno predefinido; por ejemplo, el autor con `id = 1`, o el primer usuario que encuentres en la base de datos (`Usuario::get()->first()`). Más adelante ya lo haremos dependiente del usuario que se haya autenticado. Recuerda definir el método `store` en el controlador de posts para dar de alta el post, y redirigir después al listado principal de posts. Para cargar el formulario, añade una nueva opción en el menú principal de navegación.
- En la ficha de un post, añade un botón con un formulario para borrar el post. Deberás definir el código del método `destroy` para eliminar el post y redirigir de nuevo al listado. En el caso de que hayas hecho el ejercicio opcional de la sesión anterior para añadir comentarios a los posts, deberás previamente eliminar todos los comentarios asociados a ese post, y después borrar el post. Para filtrar los comentarios de un post y borrarlos, utiliza la cláusula `where` que se explicó en la sesión 4:

```
1 Comentario::where('post_id', $id)->delete();
2 // aquí ya borramos el post
```

## 1.2. ejercicio 2 (opcional)

Continuamos con el proyecto **blog** anterior. Ahora añadiremos el formulario de edición de un post, también desde la vista de la ficha del post. El formulario deberá mostrar los datos ya rellenos del post. Dicho formulario se carga a partir del método `edit` (que deberá renderizar la vista con el formulario de edición, `resources/views/posts/edit.blade.php`), y el formulario se enviará al método `update` del controlador, pasándole como parámetro el `id` del post a modificar.

## 1.3. ejercicio 3

Continuamos con el proyecto **blog** anterior. Crea un *form request* llamado `PostRequest`, que valide los datos del post. En concreto, deben cumplirse estos requisitos:

- El título del post debe ser obligatorio, y de al menos 5 caracteres de longitud.
- El contenido del post debe ser obligatorio, y de al menos 50 caracteres de longitud.

Define mensajes de error personalizados para cada posible error de validación, y muéstralos junto a cada campo afectado, como en el ejemplo de la biblioteca. Además, utiliza la función `old` para recordar el valor antiguo correcto, en el caso de que un campo pase la validación pero otro(s) no.

### ¿Qué entregar?

Como entrega de esta sesión deberás comprimir el proyecto **blog** con todos los cambios incorporados, y eliminando las carpetas `vendor` y `node_modules` como se explicó en las sesiones anteriores. Renombra el archivo comprimido a `blog_08a.zip`.

## 2. bibliografia

---

- [Nacho Iborra Baeza](#).