

# Curso de Laravel

## Ejercicios propuestos - Parte 4 (I)

### Ejercicio 1

---

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

- Crea una base de datos llamada `blog` en tu servidor de bases de datos a través de phpMyAdmin. Modifica también el archivo `.env` del proyecto para acceder a dicha base de datos con las credenciales adecuadas, similares a las del ejemplo de la biblioteca (cambiando el nombre de la base de datos).
- Elimina todas las migraciones existentes, salvo la de `create_users_table`. Edita esta migración de la tabla usuarios para dejarla igual que el ejemplo de la biblioteca (únicamente con los campos login y password, además del id y los timestamps).
- Crea una nueva migración llamada `crear_tabla_posts`, que creará una tabla llamada `posts` con estos campos:
  - Id autonumérico
  - Título del post (`string`)
  - Contenido del post (`text`)
  - Timestamps para gestionar automáticamente la fecha de creación o modificación del post
- Lanza las migraciones y comprueba que se crean las tablas correspondientes con los campos asociados en la base de datos.
- Opcionalmente, puedes desactivar las migraciones de Sanctum en el proyecto Laravel, como se ha explicado al final de [este documento](#), si al crear las migraciones aparece alguna otra tabla no deseada, como la de los `personal_access_tokens`. Recuerda ejecutar `php artisan migrate:fresh` de nuevo para borrar el rastro de esas tablas.

### Ejercicio 2

---

Continuamos con el proyecto **blog** anterior. Modifica si no lo has hecho aún el modelo `User` que viene por defecto para que se llame `Usuario`, igual que hemos hecho en el ejemplo de la biblioteca. Crea un nuevo modelo llamado `Post` para los posts de nuestro blog. Asegúrate de que ambos modelos se ubican en la carpeta `App\Models` del proyecto.

Después, modifica los métodos del controlador `PostController` creado en sesiones anteriores, de este modo:

- El método `index` debe obtener todos los posts de la tabla, y mostrar la vista `posts.index` con ese listado de posts.
  - La vista `posts.index`, por su parte, recibirá el listado de posts y mostrará los títulos de cada uno, y un botón `Ver` para mostrar su ficha (`posts.show`).
  - Debes mostrar el listado de posts ordenado por título en orden ascendente, y paginado de 5 en 5.
- El método `show` debe obtener el post cuyo id se pasará como parámetro, y mostrarlo en la vista `posts.show`.

- La vista `posts.show` recibirá el objeto con el post a mostrar, y mostraremos el título, contenido y fecha de creación del post, con el formato que quieras.
- El método `destroy` eliminará el post cuyo id recibirá como parámetro, y devolverá la vista `posts.index` con el listado actualizado. Para probar este método, recuerda que debes definir un formulario en una vista (lo puedes hacer para cada post mostrado en la vista `posts.index`) que envíe a la ruta `posts.destroy` usando un método DELETE, como hemos explicado en un ejemplo anterior.
- Los métodos `create`, `edit`, `store` y `update` de momento los vamos a dejar sin hacer, hasta que veamos cómo gestionar formularios.
- Para simular la inserción y la modificación, vamos a crear dos métodos adicionales en el controlador, que usaremos de forma temporal:

- Un método llamado `nuevoPrueba`, que cada vez que lo llamemos creará un post con un título al azar (por ejemplo, "Título X", siendo X un entero aleatorio), y un contenido al azar ("Contenido X"). Puedes emplear la función `rand` de PHP para generar estos números aleatorios para título y contenido.
- Un método llamado `editarPrueba`, que recibirá como parámetro un id y modificará el título y contenido del post otros generados aleatoriamente, como en el punto anterior.
- Estos dos métodos (especialmente el primero) nos servirán para crear una serie de posts de prueba que luego nos servirán para probar el listado y la ficha de los posts.
- En el archivo `routes/web.php`, recuerda añadir dos nuevas rutas temporales de tipo `get` para probar estas inserciones y modificaciones. La primera puede apuntar a `/posts/nuevoPrueba`, por ejemplo, y la segunda a `/posts/editarPrueba/{id}`. Recuerda también eliminar o editar la restricción `only` de las rutas del controlador que estableciste la sesión anterior, para que no sólo permita las rutas `index`, `show`, `create` y `edit`, y además permita la de `destroy` (o todas las posibles, si quieres, ya que tarde o temprano las utilizaremos).

**IMPORTANTE:** los métodos `nuevoPrueba` y `editarPrueba` que has creado en `PostController` NO son métodos estándar de un controlador de recursos, y de ninguna manera estarán disponibles a través de `Route::resource` en `routes/web.php`. Por eso debes definir a mano una ruta para cada uno de ellos en ese archivo, a través de `Route::get`, y esas rutas deben definirse ANTES de la de recursos (`Route::resource`) porque de lo contrario no se emparejarán correctamente.

### ¿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_04.zip`.