UNIT 5 AND 6: CONTROLLERS AND MODELS

CONTROLLERS

On the blog project, from the previous session, we are going to add these changes:

Creates a resource controller ('-r' option) called 'PostController', which will help us manage all the logic of the blogposts.

Automatically assigns with the 'resource' method each route to its corresponding function of the controller, in the file 'routes/web.php'. It limits with 'only' the actions only to the functions of listing ('index'), tab('show'), creation('create') and editing('edit').

Rename the listing and tab views of a post to 'index.blade.php' and 'show.blade.php',within your 'posts' folder, and have the corresponding posthandler methods render these views. For the 'create' and 'edit' methods, it simply returns a plain text indicating 'New post' and 'Post edit', for example.

Make any additional changes that are convenient (for example, in the navigation menu) so that the links continue to work, and test that the four routes (listing, tab, creation and editing) work properly.

MODELS

We are going to add these changes:

Edit the migration of the users table to leave it the same as the library example (only with the fields login and password, in addition to the id and timestamps).

Create a new migration called 'create_table_posts', which will create a table called 'posts' with these fields:

- Id: autonumeric
- title: Post title ('string')
- Content: of the post ('text')
- Timestamps to automatically manage the date of creation or modification of the post

Launches the migrations and verifies that the corresponding tables are created with the associated fields in the database.

Create a new model called 'Post' for our blog posts.

Then, modify the methods of the 'PostController' controller created in previous sessions, like this:

- The 'index' method should get all the posts in the table, and show the 'posts.index' view with that list of posts.
- The 'posts.index' view, will receive the list of posts and will show the titles of each one, and a 'View' button to show all the fields of this post ('posts.show'). You must show the list of posts sorted by title in ascending order, and paginated 5 by 5.
- The 'show' method should get the post passed as a parameter, and display it in the 'posts.show' view.
- The 'posts.show' view will receive the object with the post to be displayed, and we will show the title, content and date of creation of the post, with the format you want.
- The 'destroy' method will delete the post received as a parameter, and will return the

'posts.index' view with the updated list. To test this method, remember that you must define a form in a view (you can do it for each post shown in the 'posts.index' view) that sends to the path 'posts.destroy' using a DELETE method, as we have explained in a previous example (in library).

- The methods 'create', 'edit', 'store' and 'update' at the moment we will leave them undone, until we see how to manage forms.
- To simulate insertion and modification, we will create two additional methods in the controller, which we will use temporarily:
  - A method called 'newPost', which each time we call it will create a post with a random title (for example, "Title X", with X being a random integer), and a random content ("Content X"). You can use PHP's rand function to generate these random numbers for title and content.
  - A method called 'editPost', which will receive as a parameter an id and modify the title and content of the post randomly generated, as in the previous point.
  - These two methods (especially the first) will help us to create a series of test posts that will then serve to test the list and the tab of the posts.
- In the 'routes/web.php' file, remember to add two new temporary 'get' routes to test these insertions and modifications. The first can point to '/posts/newPost', for example, and the second to '/posts/editPost/{id}'. Remember also to remove or edit the 'only' restriction of the controller paths that you set the previous session, so that it not only allows the paths index, show, create and edit,but also allows the destroy (or all possible ones).