

Registo en Laravel

Vamos a desarrollar un sistema de login y un formulario de registro en Laravel 10. Empezamos creando nuestro proyecto:

composer create-project laravel/laravel loginmoderno

Siempre que arranquemos el proyecto, podemos hacer

php artisan serve

para arrancar el servidor y ver la página inicial de cualquier proyecto de Laravel. Ahora, en el archivo .env vamos a configurar la conexión con la base de datos. Creamos la base de datos en nuestro phpmyadmin (en este caso la llamamos loginmoderno) y actualizamos el archivo .env:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=8889
DB_DATABASE=loginmoderno
DB_USERNAME=root
DB_PASSWORD=root
```

Ahora vamos a migrar las tablas que vienen por defecto en Laravel para la autenticación. Para ello correremos el comando

php artisan migrate

Tras esto, podemos comprobar como se han creado unas tablas en nuestra nueva base de datos. Ahora vamos a integrar el sistema de registro y autenticación. Para ello, insertamos el comando

composer require laravel/ui

De esta manera instalaremos los archivos de Laravel para la autenticación de usuario. Ahora

php artisan ui bootstrap --auth

Cuando acabe la instalación, si recargamos la web veremos cómo han aparecido en la parte superior dos opciones de registro y de login. Si le diéramos a esos ítems del menú, veremos un error en el que nos pide la instalación de nuevos componentes. Pero en este caso, vamos a prescindir de ellos puesto que pesan mucho. Así que nosotros crearemos nuestro propio login y formulario de registro.

Vamos a buscar en google un login con bootstrap. Nosotros vamos a usar el que viene en esta dirección: (podemos ver el elegido en la ilustración de la siguiente página)

<https://mdbootstrap.com/docs/standard/extended/login/>

Le damos a Show Code para ver el código correspondiente. Como podremos ver, tenemos tanto el código HTML como el CSS necesario. Copiamos el código HTML y en **resources/views/auth** abrimos el archivo **login.blade.php** que viene por defecto en Laravel, y borramos su contenido. A continuación, pegamos el código HTML que hemos copiado anteriormente.

Si volvemos a pulsar el boton de Login, vemos que ya no nos sale el error de antes, pero la página no tiene el aspecto deseado. Podemos insertar directamente los CDNs de Bootstrap 5, o podemos usar una extensión de Visual Code Studio (Bootstrap 5 Quick Snnipets). Una vez instalada, borramos el contenido del archivo **login.blade.php**, escribimos **bs5-\$**, de manera que nos crea una estructura html preparada con bootstrap 5. Dentro de la capa main podemos ahora pegar el código HTML obtenido anteriormente, de manera que ya el aspecto del login es el que

Login Template

Putting your brand logo on the login page is a great practice, in addition to a logo you can also add a mission statement or other text like in the example below. If you want to create a beautiful background for your text that will match your brand colors use our [gradient generator](#).

deseamos.

De la misma manera, copiamos el código CSS, nos vamos a la carpeta **public**, creamos una carpeta **assets**, y dentro de esa carpeta, un archivo **estilos.css**, donde vamos a pegar nuestro estilo copiado anteriormente. Y lo incluimos en nuestro **login.blade.php**

```
<link rel="stylesheet" href="{{ asset('assets/estilos.css')}}">
```

De esta manera, estamos indicando que nuestro css está en **assets/estilos.css**. Refresca.y verás los cambios, gracias al nuevo estilo inyectado.

Es un buen momento para hacer pequeñas modificaciones en el estilo y las imágenes de nuestra página de login. Podemos pegar una nueva imagen para el logo, ya sea en una subcarpeta de la carpeta assets, o en ella misma. Yo voy a pegarlo dentro de assets. Y lo cambiamos dentro del archivo de login. Y de paso, los textos del login.

Además de cambios estéticos y demás, vamos a agregar el atributo name a los inputs. Además, ahora vamos a agregar el método a nuestro formulario, en este caso será un método POST, y en el action, la ruta hacia el login:

```
<form method="POST" action="{{route('login')}}">  
@csrf
```

Ya sabemos que ese token nos servirá para evitar ataques de terceros que quieran violentar nuestro formulario.

Ahora vamos a crearnos un formulario de registro, usando el mismo código que la plantilla de login, y lo pegamos en **register.blade.php**. Necesitamos que el botón de Crear Cuenta nos lleve a ese formulario de registro, por lo que en **login.blade.php**, cambiamos el botón de Crear Cuenta por:

```
<a href="{{route('register')}}" class="btn btn-outline-danger">Crear cuenta</a>
```

Si lo deseamos, podemos comprobar que el enlace funciona perfectamente. Ahora vamos a modificar la pantalla de registro, cambiando textos y agregando algunos campos. Uno para el nombre de usuario y otro para confirmar la contraseña:

```
<div class="form-outline mb-4">
    <label class="form-label" for="form2Example11">Nombre</label>
    <input type="text" name="name" id="form2Example33" class="form-control"
        placeholder="tu nombre de usuario" />
</div>
```

Y

```
<div class="form-outline mb-4">
    <label class="form-label" for="form2Example22">Confirmar Contraseña</label>
    <input type="password" name="password_confirmation" id="form2Example44"
class="form-control" />
</div>
```

Podemos refrescar la página para ver cómo nos está quedando. Cambiamos también el action del formulario:

```
<form method="POST" action="{{route('register')}}">
```

Para acabar, cambiamos el tipo del botón por un submit y probamos a ingresar a un usuario. Veremos que nos devuelve un error, pero si miramos en la base de datos, podremos comprobar que se ha insertado el usuario en la tabla **users**.

Para solventar el error que nos aparece al registrar el usuario, nos vamos a **resources/views/layouts** y abrimos el archivo **app.blade.php** y copiamos el div que aparece en la línea 57:

```
<div class="dropdown-menu dropdown-menu-end" aria-labelledby="navbarDropdown">
    <a class="dropdown-item" href="{{ route('logout') }}"
        onclick="event.preventDefault();
        document.getElementById('logout-form').submit();">
        {{ __('Logout') }}
    </a>
    <form id="logout-form" action="{{ route('logout') }}" method="POST" class="d-
none">
        @csrf
    </form>
</div>
```

Nos vamos a home, borramos todos y pegamos lo que acabamos de copiar. No se nos debe olvidar que en la página de login también debemos cambiar el type del botón a submit. Y ya tenemos a nuestro sistema funcionando. Esa página Home, que de momento sólo muestra el Logout, puede ser nuestra página de administración o lo que deseemos que se vea una vez logados.

