

Hoja06_MVC_10

Seguimos con la web para la gestión de un zoológico realizando la autenticación de usuarios

1.- Sistema de autenticación

En primer lugar, instalaremos Laravel Jetstream siguiendo los apuntes de clase

```
composer require laravel/jetstream
php artisan jetstream:install livewire
npm install
npm run dev
php artisan migrate
```

2.- Seeders de usuarios

Vamos a rellenar la tabla con datos iniciales.

Crearemos un nuevo seeder llamado UserSeeder, donde creamos un usuario que sea tú.

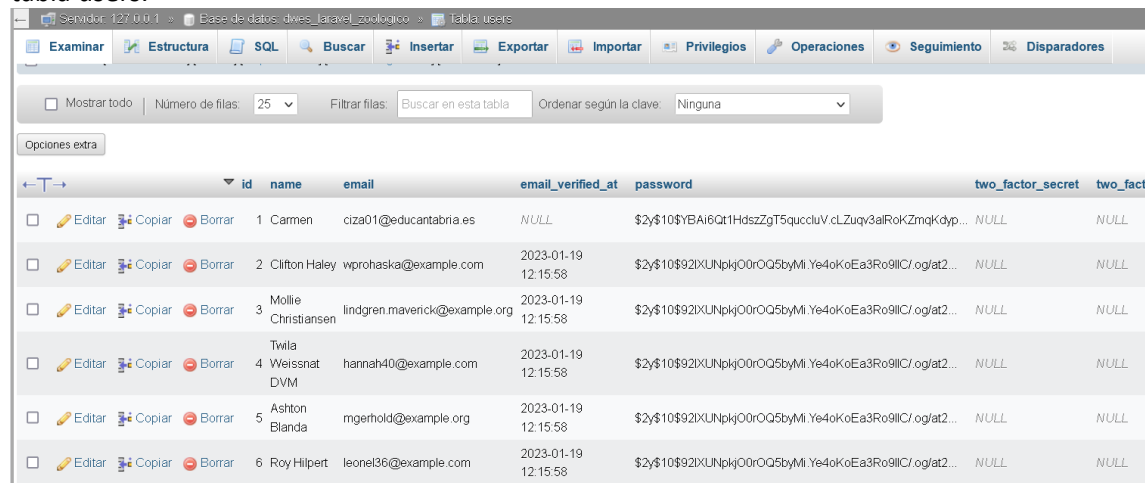
Recuerda que para guardar el password es necesario encriptarlo manualmente usando el método bcrypt

Para esto editamos el fichero de semillas situado en database/seeds/DatabaseSeeder.php y añadiremos el código de borrar la tabla de usuario, llamar al seeder de usuario que crearemos y en este caso utilizaremos factory para crear cinco usuarios más

Mirar el UserFactory

Por último, tendremos que ejecutar el comando de Artisan que procesa las semillas.

Una vez realizado esto comprobamos en PHPMyAdmin que se han añadido los usuarios a la tabla users.



| | id | name | email | email_verified_at | password | two_factor_secret | two_fact |
|--------------------------|----|---------------------|-------------------------------|---------------------|---|-------------------|----------|
| <input type="checkbox"/> | 1 | Carmen | ciza01@educantabria.es | NULL | \$2y\$10\$YBAi6Qt1HdsZgT5quclluVcLZuqv3alRokZmqKdyp... | NULL | NULL |
| <input type="checkbox"/> | 2 | Clifton Haley | wprohaska@example.com | 2023-01-19 12:15:58 | \$2y\$10\$92XUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | NULL | NULL |
| <input type="checkbox"/> | 3 | Mollie Christiansen | lindgren.maverick@example.org | 2023-01-19 12:15:58 | \$2y\$10\$92XUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | NULL | NULL |
| <input type="checkbox"/> | 4 | Twila Weissnat DVM | hannah40@example.com | 2023-01-19 12:15:58 | \$2y\$10\$92XUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | NULL | NULL |
| <input type="checkbox"/> | 5 | Ashton Blanda | mgerthold@example.org | 2023-01-19 12:15:58 | \$2y\$10\$92XUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | NULL | NULL |
| <input type="checkbox"/> | 6 | Roy Hilpert | leonel36@example.com | 2023-01-19 12:15:58 | \$2y\$10\$92XUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2... | NULL | NULL |

3.- Controles de autorización, traducción a castellano

A continuación, edita el fichero routes/web.php y realiza las siguientes acciones:

- Añade un *middleware* que aplique el filtro auth para proteger las rutas de crear y editar animales.

```
Route::get('animales/crear', 'create')->name('animales.create')->middleware('auth');
```

```
Route::get('animales/{animal}', 'show')->name('animales.show');
```

```
Route::get('animales/{animal}/editar', 'edit')->name('animales.edit')->middleware('auth');
```

- Revisa mediante el comando de Artisan `php artisan route:list` las nuevas rutas y que

Hoja06_MVC_10

el filtro auth se aplique correctamente.

Realizar además las siguientes operaciones:

- Modificar en **App\Providers\RouteServiceProvider** para que cuando se realice el login te redirija a la ruta /animales.
- “Traducir” los datos de la aplicación para que se muestre en castellano.
- Esto se puede hacer “a mano”, pero también se puede instalar las traducciones en español que trae por defecto Laravel. Para ello hacemos lo siguiente:

basado en el artículo <https://desarrolloweb.com/articulos/localizacion-laravel>

- Instalación del paquete de idiomas Laravel-lang
`composer require laravel-lang/lang -dev`
- Instalación del publisher, el translation manager
`composer require laravel-lang/publisher laravel-lang/lang -dev`
- Instalación de los archivos del idioma español
`php artisan lang:add es`
- Ahora los archivos de idioma se encuentran en el directorio `\vendor\laravel-lang\lang\locales\es`
- Copia los archivos que se encuentran ahí a la carpeta `\lang\es`
- Cambia el parámetro locale de la configuración de Laravel en `config/app.php` a `'es'`
- Ejecutar
`php artisan config:cache`
- Modificar el fichero `nav.blade.php`, para que llame a las rutas de usuario (login, logout y register). Si el usuario ha sido registrado que nos muestre el nombre de la persona conectada y el botón de cerrar sesión si el usuario no está conectado solo mostrará como está ahora el botón de iniciar sesión.

Comprueba en este punto que el sistema de autenticación funciona correctamente: no te permite entrar a las rutas protegidas si no estás autenticado, puedes acceder con los usuarios definidos en el fichero de semillas y funciona el botón de cerrar sesión.

Información de Jetstream <https://jetstream.laravel.com/2.x/introduction.html>

Para modificar la apariencia de las páginas de jetstream, es muy probable que hagan referencia a los componentes definidos en:

vendor\laravel\jetstream\resources\view\components pero no debemos modificar en ese punto puesto que perdemos las modificaciones si actualizamos versión, lo que tenemos que realizar es publicarlas para que realice una copia en nuestro proyecto.

```
PS C:\xampp\htdocs\laravel_zoologico> php artisan vendor:publish --tag=jetstream-views
```

INFO Publishing [jetstream-views] assets.

```
Copying directory [C:\xampp\htdocs\laravel_zoologico\vendor\laravel\jetstream\resources\views] to [C:\xampp\htdocs\laravel_zoologico\resources\views\vendor\jetstream] DONE
```