

UD10.8 – Laravel

Sistema de validación de usuarios JetStream

2º CFGS
Desarrollo de Aplicaciones Web

2022-23

1.- Sistema de validación de usuarios

Laravel incluye varios sistemas de registro y validación de usuarios completos que se puede añadir a los proyectos.

El sistema de validación se puede crear:

- En el momento de creación del proyecto.
- Una vez creado y avanzado el proyecto.

La mejor opción es la primera ya que si se añade el sistema de validación posteriormente pueden producirse comportamientos inadecuados y errores

1.- Sistema de validación de usuarios

Los sistemas de validación disponibles son:

- **Laravel Breeze:** sistema minimalista y simple que incluye registro, login, recuperación de password, verificación de email, confirmación de password y página de perfil donde modificar los datos de la cuenta.

Tiene varias implementaciones:

Breeze + Blade

Breeze + React / Vue

Breeze + Next.js / API

- **Laravel JetStream:** sistema robusto con funcionalidades adicionales como son autenticación de dos factores, manejo de sesiones, soporte por API y gestión de equipos.

Usa Tailwind CSS y tiene dos implementaciones posibles:

Livewire + Blade

Inertia + Vue

2.- Laravel JetStream

Laravel JetStream es el sistema más robusto de validación de usuarios que ofrece Laravel de manera automática.

Laravel Jetstream

Laravel Jetstream is a beautifully designed application starter kit for Laravel and provides the perfect starting point for your next Laravel application. Jetstream provides the implementation for your application's login, registration, email verification, two-factor authentication, session management, API via [Laravel Sanctum](#), and optional team management features.

Jetstream is designed using [Tailwind CSS](#) and offers your choice of [Livewire](#) or [Inertia](#) scaffolding.

2.- Laravel JetStream

Para crear un proyecto Laravel que incorpore la autenticación de usuarios primero se debe crear un proyecto laravel con composer:

```
# composer create-project --prefer-dist laravel/laravel nombreproyecto
```

A continuación, desde la carpeta del proyecto se debe ejecutar el siguiente comando para tener disponible JetStream, antes de ejecutar este comando deberás copiar el archivo composer.phar a la carpeta del proyecto:

```
# composer require laravel/jetstream
```

Recuerda en clase el comando debe comenzar con: **php composer.phar**.

2.- Laravel JetStream

Una vez disponible JetStream se debe instalar en el proyecto indicando la implementación elegida: livewire o inertia

```
# php artisan jetstream:install livewire
```

Si se quiere habilitar el soporte para equipos de trabajo se debe indicar al final del comando con **--teams**:

```
# php artisan jetstream:install livewire --teams
```

2.- Laravel JetStream

Como el sistema de autenticación necesita una base de datos para trabajar antes de poder probar el proyecto y empezar a desarrollarlo es necesario:

- Crear la base de datos.
- Modificar la migración de la tabla de usuarios o añadir una migración para modificar dicha tabla añadiendo los campos necesarios para la tabla Users según las necesidades del proyecto.
- Ejecutar las migraciones.

2.- Laravel JetStream

Si se observa la migración que se ha creado para la tabla Users se pueden ver todos los campos que añade JetStream para la gestión del registro, acceso, autenticación...

```
public function up(): void
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password');
        $table->rememberToken();
        $table->foreignId('current_team_id')->nullable();
        $table->string('profile_photo_path', 2048)->nullable();
        $table->timestamps();
    });
}
```

Además, se pueden añadir todos los campos que se necesitan dependiendo de las necesidades del proyecto.

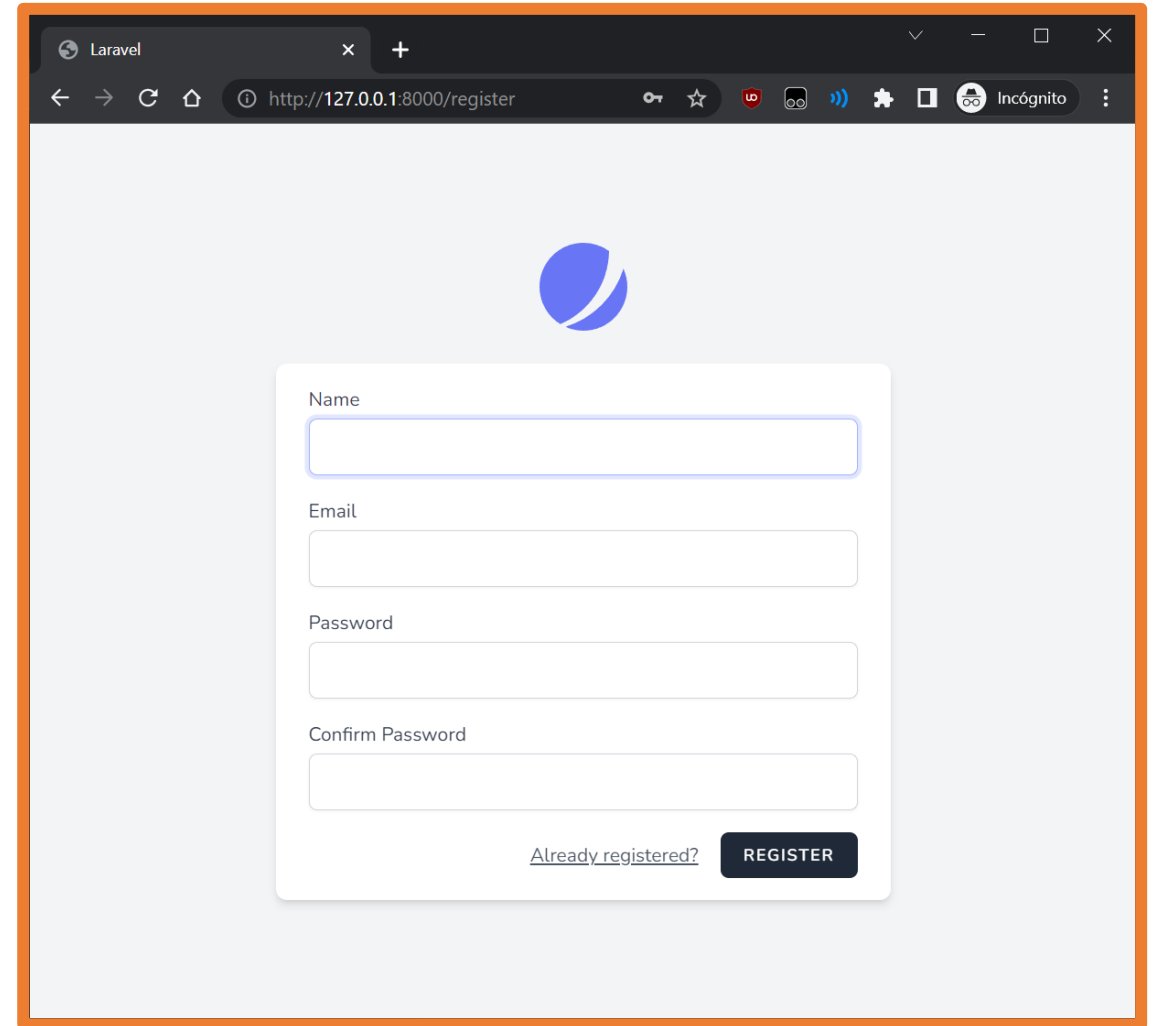
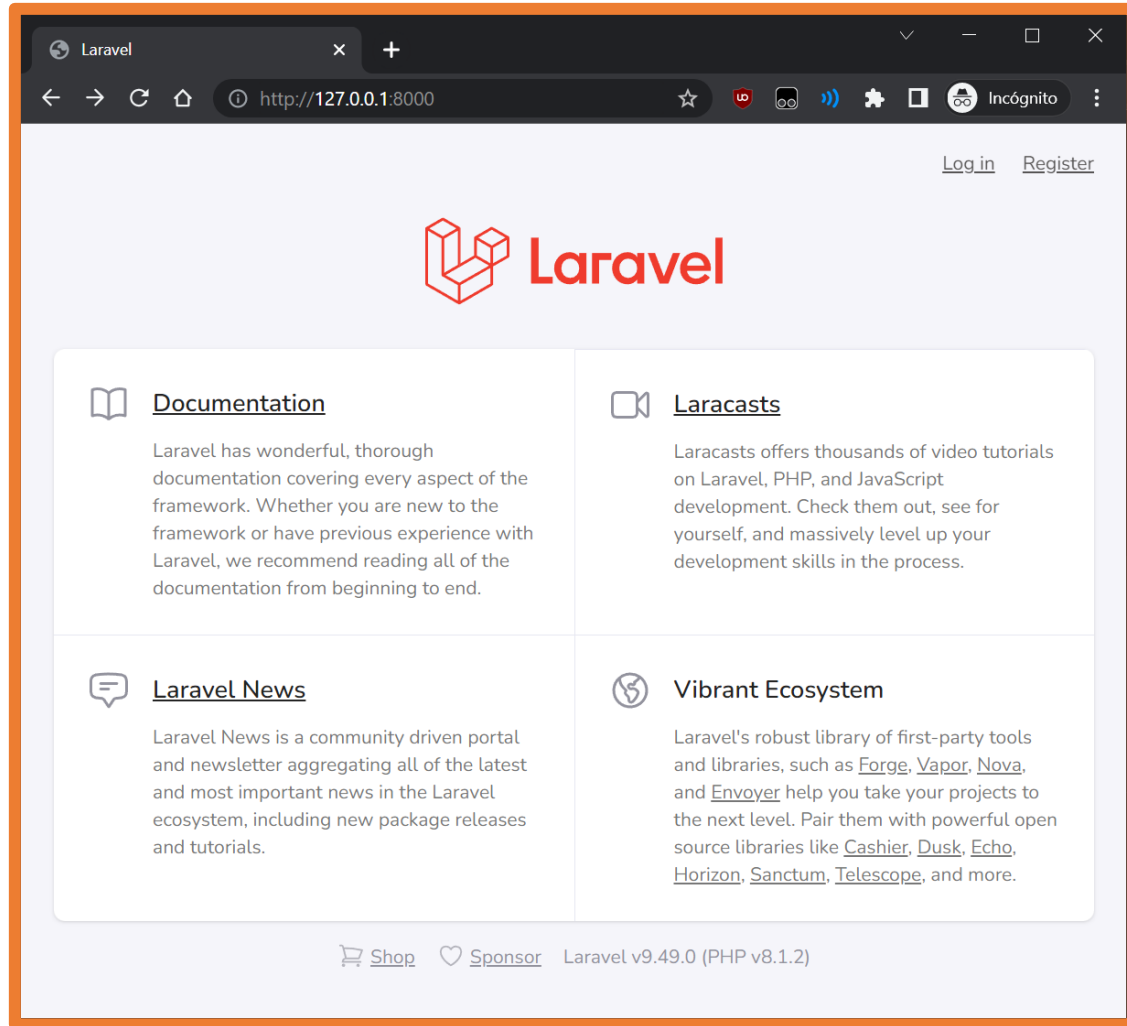
2.- Laravel JetStream

Si no se añade ningún campo extra a la migración Users, una vez realizados los pasos anteriores en los que se ha instalado Jetstream, en Laravel ya estará operativa toda la funcionalidad del sistema de autenticación de usuarios para:

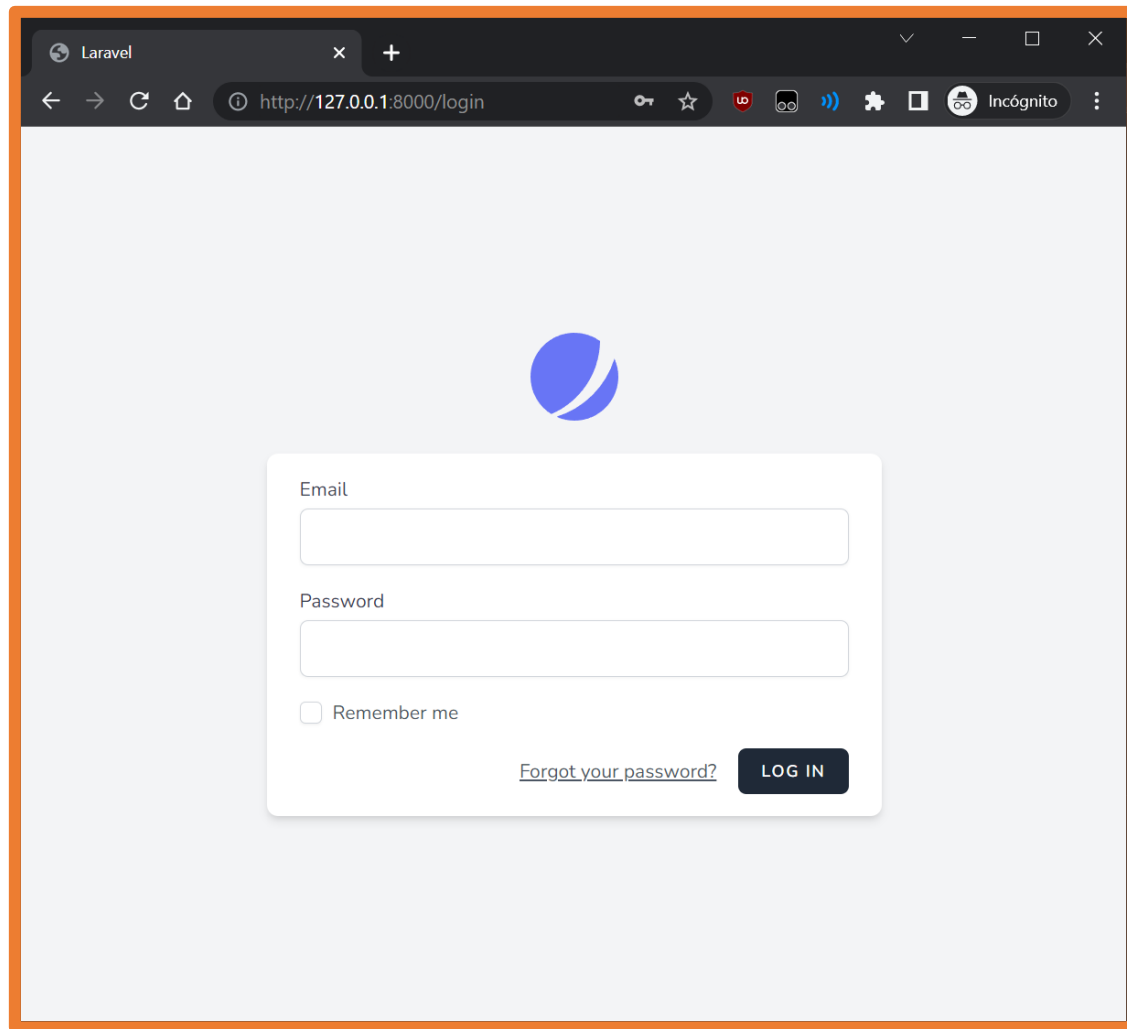
- Registrarse.
- Verificar el email.
- Loguearse.
- Recuperar contraseña.
- Cambiar foto de perfil.
- Guardar información de login.

En la instalación se han incluido todas las clases necesarias, todas las vistas necesarias y todas las rutas necesarias para el sistema de validación.

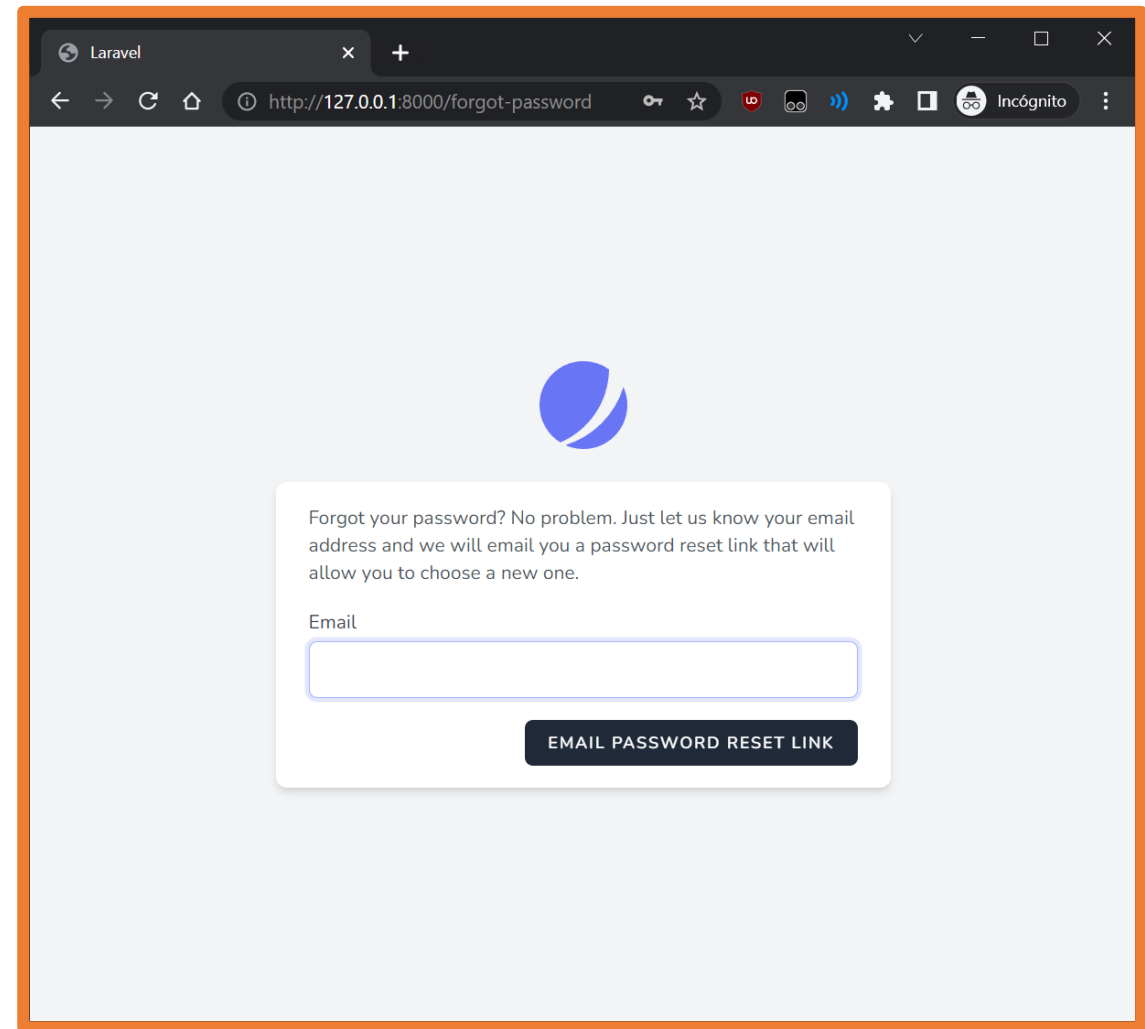
2.- Laravel JetStream



2.- Laravel JetStream



A screenshot of a web browser displaying the Laravel JetStream login page. The browser's address bar shows the URL `http://127.0.0.1:8000/login`. The page features a light gray background with a blue Laravel logo at the top center. Below the logo is a white login form with two input fields labeled "Email" and "Password". There is a checkbox labeled "Remember me" and a dark blue button labeled "LOG IN". A link labeled "Forgot your password?" is located at the bottom right of the form.



A screenshot of a web browser displaying the Laravel JetStream forgot password page. The browser's address bar shows the URL `http://127.0.0.1:8000/forgot-password`. The page features a light gray background with a blue Laravel logo at the top center. Below the logo is a white form with a text block that reads: "Forgot your password? No problem. Just let us know your email address and we will email you a password reset link that will allow you to choose a new one." Below this text is a single input field labeled "Email". At the bottom right of the form is a dark blue button labeled "EMAIL PASSWORD RESET LINK".

3.- Personalizando Laravel JetStream

Como se ha observado en las páginas anteriores al instalar Jetstream se han incorporado a Laravel una serie de vistas para poder mostrar todas las funcionalidades anteriores.

Estas vistas se encuentran en la carpeta **resources/views/auth** y en la carpeta **resources/views/profile**.

Las vistas creadas automáticamente se pueden modificar para ajustarse al layout de la aplicación web.

Se debe tener especial cuidado de mantener la funcionalidad que otorgan.

3.- Personalizando Laravel JetStream

En muchos casos la migración creada automáticamente para la tabla Users no satisface las necesidades del proyecto que se va a realizar y se añaden campos a dicha tabla mediante las migraciones.

Se debe tener en cuenta que Jetstream instala el paquete Fortify para ayudarse con el sistema de autenticación, por ello para realizar cambios en el registro hay que hacer las cosas de manera diferente a lo visto anteriormente.

3.- Personalizando Laravel JetStream

Los pasos a seguir para modificar el registro son los siguientes:

- Si no se han ejecutado aún las migraciones:
Añadir los campos nuevos a la migración de la tabla Users ya existente.
- Si se han ejecutado las migraciones con anterioridad
Crear una migración (add_fields_to_users_table) y añadir los campos nuevos
- Ejecutar las migraciones: `php artisan migrate`

3.- Personalizando Laravel JetStream

Los pasos a seguir para modificar el registro son los siguientes:


- Modificar la vista de registro para añadir al formulario los campos nuevos:
resources/views/auth/register.blade.php
- Modificar la vista de actualización de perfil para añadir al formulario los campos nuevos:
resources/views/profile/update-profile-information-form.blade.php

3.- Personalizando Laravel JetStream

Los pasos a seguir para modificar el registro son los siguientes:

- En el modelo User se debn añadir los campos rellenables:
app/models/User.php

```
protected $fillable = [  
    'name',  
    'email',  
    'password',  
];
```



En el modelo también se definen los campos que no se deben recuperar cuando se obtiene un objeto de la clase User, si se quiere añadir alguno se debe hacer en el array **\$hidden**.

3.- Personalizando Laravel JetStream

Los pasos a seguir para modificar el registro son los siguientes:

- Modificar el archivo de creación de usuario de Fortify para añadir las condiciones sobre los campos nuevos:

app/Actions/Fortify/CreateNewUser.php



```
public function create(array $input): User
{
    Validator::make($input, [
        'name' => ['required', 'string', 'max:255'],
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'password' => $this->passwordRules(),
        'terms' => Jetstream::hasTermsAndPrivacyPolicyFeature() ? ['accepted', 'required'] : '',
    ]->validate();

    return User::create([
        'name' => $input['name'],
        'email' => $input['email'],
        'password' => Hash::make($input['password']),
    ]);
}
```

- Modificar el archivo de modificación de usuario de Fortify por la misma razón:
app/Actions/Fortify/UpdateUserProfileInformation.php

6.- Protegiendo rutas

Una vez incluido el sistema de validación de usuarios ya podrán visitar la aplicación web tanto usuarios invitados como registrados.

Con el sistema de validación activo es posible **proteger páginas completas** protegiendo sus rutas, para esto se utiliza un **middleware** incluido en Laravel.

O también se puede utilizar para **mostrar/ocultar ciertas partes de las páginas web** según si el usuario está logueado o si cumple un rol determinado.

7.- Protegiendo rutas completas

Usando el **middleware auth** que ofrece Laravel para proteger rutas completas:

```
Route::get('/sales/empresa/{nombre}', [SaleController::class, 'empresa'])
    ->name('ofertasempresa')
    ->middleware('auth');
```

```
Route::resource('/products', ProductController::class)
    ->middleware('auth');
```

```
Route::resource('/discounts', DiscountController::class)
    ->only(['create', 'edit'])
    ->middleware('auth');
```

```
Route::resource ('/discounts', DiscountController::class
    ->except(['create', 'edit']));
```

8.- Mostrando información a usuarios logueados

Mediante la clase **Auth** es posible saber si el usuario está logueado o no en el sistema.

Se puede usar tanto en controladores como en vistas.

En el caso de estar logueado con la clase **Auth** se podrá obtener información del usuario.

Gracias a **Auth** también se puede mostrar en una vista información específica para el usuario logueado.

8.- Mostrando información a usuarios logueados

Dentro de las vistas existen dos maneras de comprobar si el usuario está logueado o no.

- Usando la estructura de control **@auth**:

```
@auth
  Bienvenido {{Auth::user()->name}} -
  <a href="{{route('products.index')}}">Llista de productos</a> -
  <a href="{{route('logout')}}">Salir</a>
@else
  <a href="{{route('login')}}">loguéate</a> -
  <a href="{{route('register')}}">regístrate</a>
@endauth
```

8.- Mostrando información a usuarios logueados

Dentro de las vistas existen dos maneras de comprobar si el usuario está logueado o no.

- Usando la estructura de control **@if** y la clase **Auth**:

```
@if(Auth::check())
    Bienvenido {{Auth::user()->name}} -
    <a href="{{route('products.index')}}">Llista de productos</a> -
    <a href="{{route('logout')}}">Salir</a>
@else
    <a href="{{route('login')}}">loguéate</a> -
    <a href="{{route('register')}}">regístrate</a>
@endif
```

8.- Mostrando información a usuarios logueados

Con la clase **Auth** se permite usar el operador ternario:

```
Bienvenido {{Auth::check() ? Auth::user()->name : 'Invitado'}}
```

Cuando el usuario está logueado a través de **Auth::user()** se tiene acceso a los datos del usuario:

Auth::user()->name

Auth::user()->email

Auth::user()->rol

Auth::user()->posts

9.- Añadir sistema de autenticación a posteriori

Si fuera necesario se podría añadir el sistema de validación de usuarios una vez que ya se ha avanzado en el proyecto.

Esta acción es desaconsejable tal y como se indica en la documentación de JetStream.

New Applications Only

Jetstream should only be installed into new Laravel applications. Attempting to install Jetstream into an existing Laravel application will result in unexpected behavior and issues.

En el caso de querer añadir el sistema de validación a un proyecto ya avanzado los pasos son los mismos que los indicados para proyectos nuevos.

Práctica

Actividad 14:

Crea un proyecto nuevo de Laravel y añade el sistema de validación.

Ejecuta el servidor y navega por todas las opciones del sistema de validación: registro, login, cuenta...