Autenticación

Todo lo que implica autenticar: formulario, modelo, validación, errores, recuperac password,etc, LARAVEL lo hace muy fácil.

Escribe: **php artisan make: auth**

En .env: en la parte de mail, hay que escribir **log** en el primer línea que dice mail driver.

trait

middleware

es una capa intermedia, ya que accede a la petición del usuario antes de que llegue al controlador

es una clase.

para ser ejecutadas antes de que llegue a las capas del controlador

luego de creado con php artisan make:middleware. Se crea en **http**

**el código debe estar escrito en la clase antes de return $next($request);**

ir a **KERNEL** para registrar los Middleware de rutas GLOBALES

**DENTRO del array** escribió:

* *En protected $middlewareGroups*: al escrbirlo aquí, en todas las vistas se ve lo que haya escrito en el middleware.

**\app \http \Middleware AdminMiddleware::class**

* ***En protected $routeMiddleware* = [**

**le creó un nombre:**

**‘admin’ => \app \http \Middleware AdminMiddleware::class**

Luego fue a web a las rutas y escribió: en la misma del home

* **->middleware (‘admin.chek’); aquí va con comillas el nombre que le he creado en *En protected $routeMiddleware* = [**

Luego en public function en handle

If(!auth ()->user ()->Admin()) {

}

If(!auth ()->user ()->Admin()) {

**\App::abort (403);**

}

En la carpeta ENV, está **APP DEBUG**: suele estar true, pero si le escribo falso veo otra vista.

Esta función se puede usar en la vista, en blade.php, cuando quiero que algunos usuarios no vean cierta parte de la página web.