

# LARAVEL

AULA 08

# Authentication

The background features abstract, overlapping geometric shapes in various colors including blue, green, yellow, orange, and red. These shapes are arranged in a way that creates a sense of depth and movement, with some shapes appearing to be layered on top of others. The overall composition is modern and dynamic.

**Laravel** oferece um jeito fácil e rápido de gerar toda a estrutura de visualizações e caminhos necessários para conseguir o login e register usando um simples comando.




No arquivo **web.php**, são gerados os seguintes caminhos:

**Auth::routes();**

Na pasta **Controllers/Auth:**


ForgotPasswordController.php 


LoginController.php 

RegisterController.php 

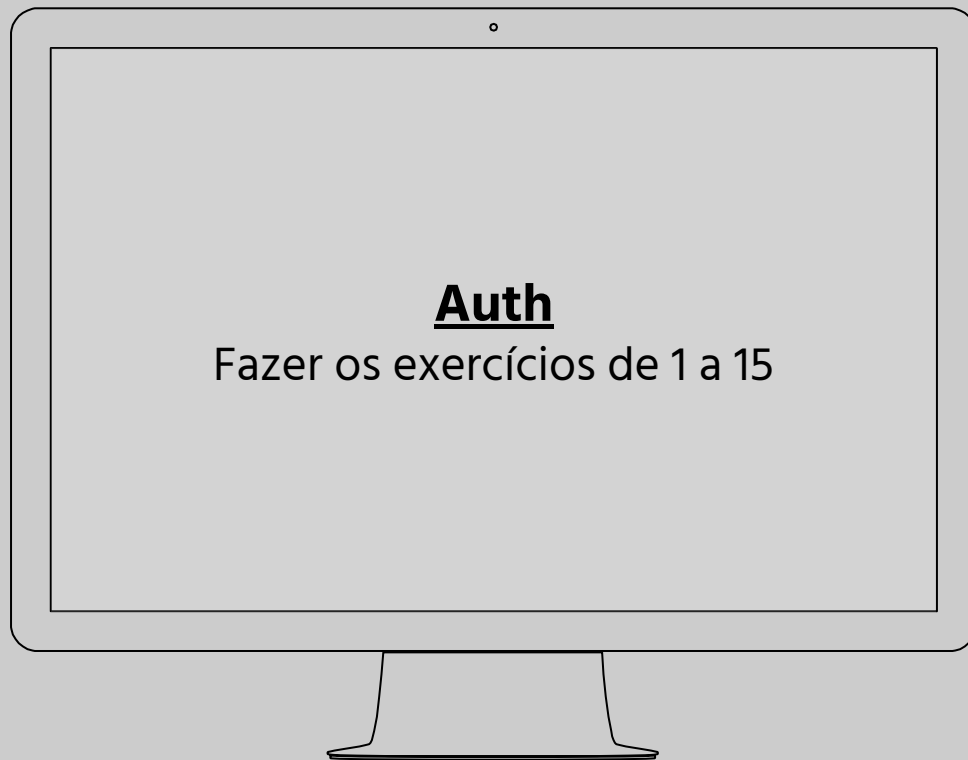
ResetPasswordController.php 

Na pasta **views**, são geradas 2 visualizações:

login.blade.php 

register.blade.php 

**HORA DE PRATICAR!**



# Middleware

The background features abstract, overlapping geometric shapes in various colors including blue, green, yellow, orange, and red. These shapes are arranged in a way that creates a sense of depth and movement, with some shapes appearing to be layered on top of others. The overall composition is modern and dynamic.

O Laravel oferece um mecanismo para filtrar todos os HTTP Request recebidos pelo aplicativo.

O **Middleware**, como o nome indica, fica entre o pedido do usuário (**Request**) e as ações do controlador, que montam e enviam a resposta (**Response**).



*Então, por exemplo, quando quer criar um novo Produto ou Post, você envia um pedido. No entanto, antes que esse pedido seja processado, o middleware pode verificar se você realmente tem permissão para executar essa ação e, caso não tenha, enviar você à página de login ou exibir um erro 401 (dependendo do Middleware usado ou definido).*



Agora que entendemos como funcionam os middlewares, vamos criar o nosso:

***php artisan make:middleware***

Depois que o middleware for criado, ele deve ser registrado

Existem dois tipos de Middleware: os **globais** e os **específicos por caminho**. Os dois são registrados no arquivo **Kernel.php**, localizado em `app/Http`.

Os Middlewares globais são executados em cada pedido feito para o aplicativo, sem importar o caminho acessado. Eles são declarados no array `$middleware` que fica dentro da classe **Kernel.php**.

Os Middlewares específicos por caminho são aqueles que, como indica o nome, são executados apenas nos caminhos especificados. Eles são declarados no array `$routeMiddleware` dentro do arquivo **Kernel.php**.



## Exemplo de middleware aplicado a um caminho:

### routes.php:

```
Route::get('admin', 'adminController@controlPanel')->middleware('role:admin');
```

### No nosso middleware:

```
public function handle($request, Closure $next, $role)
{
    if (! $request->user()->hasRole($role)) {
        // Redirect...

    }

    return $next($request);
}
```

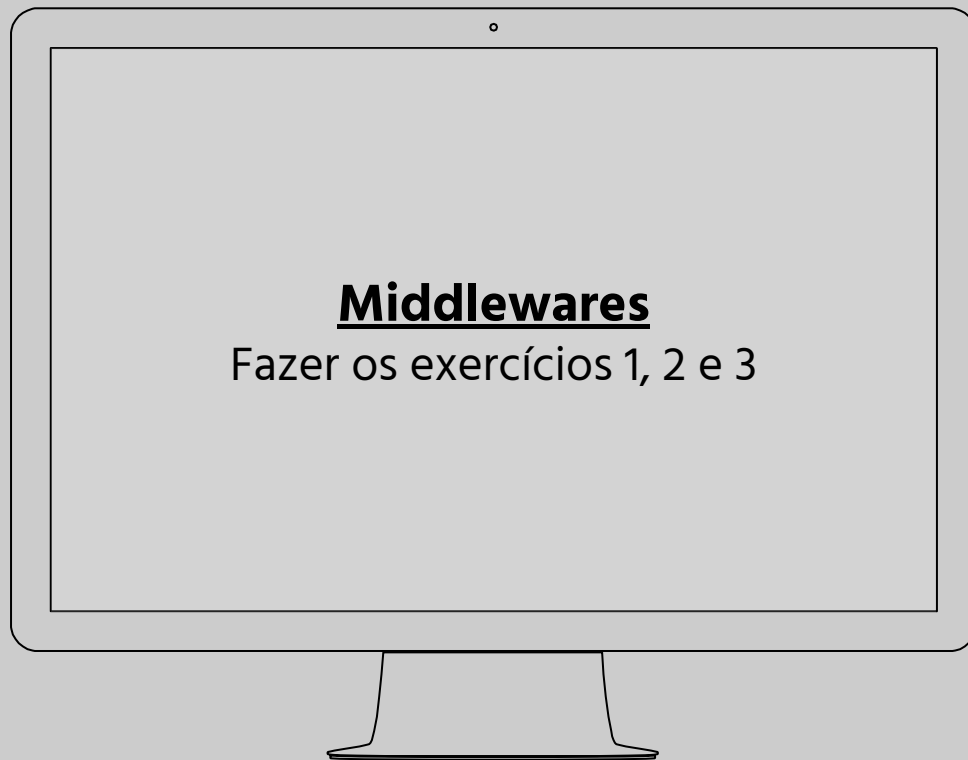


Se não for “admin”, redirecionar



Se for “admin”, o request continua

**HORA DE PRATICAR!**



The background features abstract, overlapping geometric shapes in various colors including light blue, green, cyan, magenta, orange, and red. These shapes are arranged in a way that creates a sense of depth and movement, with some shapes appearing to be layered on top of others. The overall composition is modern and dynamic.

**ATÉ A  
PRÓXIMA!**