

## 1. Padrão de Desenvolvimento MVC

O Laravel utiliza um padrão moderno de desenvolvimento chamado MVC (model-view-controller). Este padrão divide o seu projeto em 3 camadas:

- MODEL
- VIEW
- CONTROLLER

Dessa forma o seu projeto fica muito mais organizado.

Da forma como estávamos programando tanto o código HTML, quanto a conexão com o banco de dados e o código PHP ficavam no mesmo arquivo.

Mas trabalhando com o padrão MVC cada camada ficará responsável por uma dessas partes.

## 2. MODEL

Esta é a camada responsável pelo banco de dados.

Ela possui classes que representam cada uma das tabelas que você tem no seu banco de dados.

Vamos aprender mais sobre o Model nas próximas aulas.

## 3. VIEW

Esta é a camada responsável basicamente pela parte visual do seu projeto.

Todas as páginas HTML que o usuário visualiza serão produzidas na View.

## 4. CONTROLLER

É aqui que grande parte do código do seu projeto estará.

Todas os cálculos e complicações do seu código deverão ser criados no Controller.

Trata-se da regra de negócio do seu projeto.

## 5. Criando uma "Página"

Toda vez que formos criar uma nova "página" para o seu projeto precisamos lembrar do seguinte:

**ROTA > CONTOLADOR > VISUALIZAÇÃO**

Ou seja, tudo começa na rota.

Ao tentar acessar uma página do seu site o Laravel verifica antes se existe uma rota igual ao endereço que o usuário digitou no navegador.

Se existir, a rota deverá chamar um controlador.

Que por sua vez deverá retornar uma visualização para que o usuário possa visualizar a página HTML que solicitou.

## 6. Criando um Controller

Para criar um controlador vamos utilizar um comando do artisan:

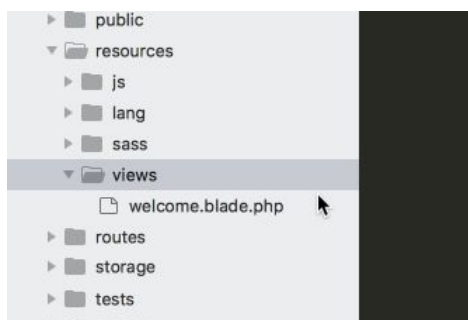
```
php artisan make:controller UsuarioController
```

Para seguir as boas práticas, o controlador deverá ter o nome iniciado com letra maiúscula e o sufixo "Controller".

Todos os controllers ficam armazenados na pasta `app/Http/Controllers` do seu projeto Laravel.

## 7. Criando uma View

Diferente do controlador, para criar uma view basta criar um arquivo em branco dentro da pasta `resources/views`.



O nome do arquivo de uma view deve sempre terminar com o sufixo `".blade.php"`.

## 8. Como fica a Rota?

Para que o Laravel saiba qual controlador chamar quando algum usuário acessar determinada rota precisamos programá-la assim:

```
1 <?php
2
3 Route::get('/', function () {
4     return view('welcome');
5 });
6
7 Route::get('/buscardados', 'HomeController@buscarDados');
8
9
10
```

Neste exemplo, quando o usuário acessar a página "dados-do-usuario" o Laravel vai executar a função "buscarDados" que deve existir dentro do controlador chamado "UsuarioController".

## 9. Como fica o Controller?

O controlador precisa ter uma função chamada "buscarDados" para executar o que a rota precisa.

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class UsuarioController extends Controller
8 {
9     public function buscarDados() {
10         return view('resultado-da-busca');
11     }
12 }
13
14
15
```

Veja que o controller faz pede para o Laravel retornar para o usuário a view chamada "resultado-da-busca", que deve existir dentro de resources/views com o nome "resultado-da-busca.blade.php".

## 10. Como fica a View?

Aqui agora é só desenvolver sua página HTML para que fique mais apresentável possível. Nesse nosso exemplo a página ficou assim:

```
1 <!doctype html>
2 <html>
3     <head>
4         <title>Resultado da Busca</title>
5     </head>
6     <body>
7         <div class="flex-center position-ref full-height">
8             <div class="content">
9                 <div class="title m-b-md">
10                     Aqui deverão aparecer as informações do usuário
11                 </div>
12             </div>
13         </div>
14     </body>
15 </html>
```