

ENSINO MÉDIO INTEGRADO - INFORMÁTICA

Disciplina de Desenvolvimento Web

Aula 03: Framework Laravel - Rotas

Gil Eduardo de Andrade

Conceitos Preliminares

(<https://laravel.com/docs/8.x>)

Conceito de Framework

O framework é um pacote de códigos prontos utilizado para o desenvolvimento de aplicações. Sua utilização visa a aplicabilidade de padrões, estruturas e funcionalidades prontas que garantem qualidade e produtividade durante o desenvolvimento. Tem como principal objetivo resolver problemas recorrentes com uma abordagem mais genérica, permitindo ao desenvolvedor estar focado no “problema”, não na arquitetura e configuração.

Framework Laravel

O Laravel é um framework voltado para linguagem PHP, podendo ser considerado, atualmente, o framework PHP mais utilizado para o desenvolvimento de aplicações web baseadas nessa linguagem.

Gerenciador de Dependências

Antigamente, quando um desenvolvedor precisava conectar com uma base de dados, ele codificava todos os aspectos desta conexão ou copiava (*copy-paste*) algo previamente implementado. Com a evolução dos procedimentos e ferramentas de desenvolvimento, e visando aumentar a produtividade, rotinas comuns e fundamentais a diversas aplicações começaram a ser disponibilizadas em pacotes. Nesta dinâmica, o desenvolver passou a adicionar os pacotes necessários a sua aplicação, de modo manual.

Contudo, caso os pacotes importados fossem alterados, caberia ao desenvolvedor identificar (saber) tais alterações e repetir todo o procedimento de inclusão, atualizando-os. Como o número de pacotes disponibilizados, e conseqüentemente

utilizados pelos desenvolvedores, cresceu rapidamente, tornou-se quase impraticável o gerenciamento manual. Surgem assim, os gerenciadores de pacotes, ou gerenciadores de dependências, ferramentas (softwares) capazes de realizar, automaticamente, os procedimentos que até então eram efetuados manualmente pelos desenvolvedores. Através dos gerenciadores de dependências torna-se possível focar no desenvolvimento da aplicação.

Composer / Instalação

(<https://getcomposer.org/download/>)

O composer é um gerenciador de dependências do PHP. Para efetuar sua instalação basta executar os seguintes comandos via terminal:

```
php -r "copy('https://getcomposer.org/installer',  
'composer-setup.php');"
```

```
php -r "if (hash_file('sha384', 'composer-setup.php') ===  
'756890a4488ce9024fc62c56153228907f1545c228516cbf63f885e036d37e9a59d27  
d63f46af1d4d07ee0f76181c7d3') { echo 'Installer verified'; } else {  
echo 'Installer corrupt'; unlink('composer-setup.php'); } echo  
PHP_EOL;"
```

```
php composer-setup.php
```

```
php -r "unlink('composer-setup.php');"
```

Ao final da execução dos comandos o arquivo composer.phar será criado no diretório apontado pelo terminal.



Criando Projeto Laravel - Via Composer

(Última Versão Laravel)

```
./composer.phar create-project laravel/laravel nome_projeto
```

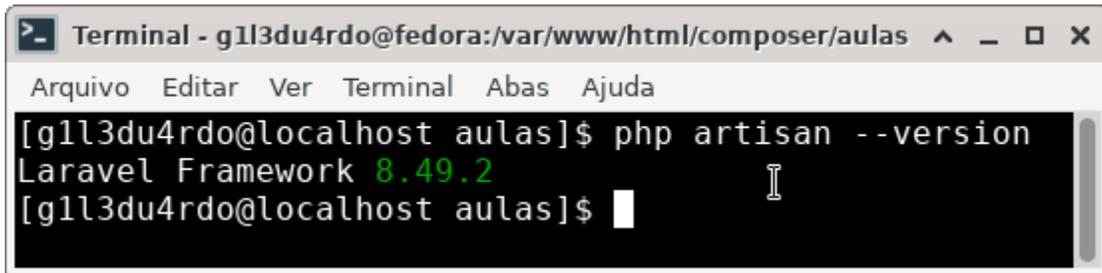
(Versão Laravel Específica)

```
./composer.phar create-project --prefer-dist laravel/laravel  
nome_projeto "8.6.*"
```

Obs.: a execução do comando para criação do projeto poder levar alguns minutos, de acordo com a velocidade de conexão da rede.

Verificando Versão da Aplicação (dentro da pasta da aplicação)

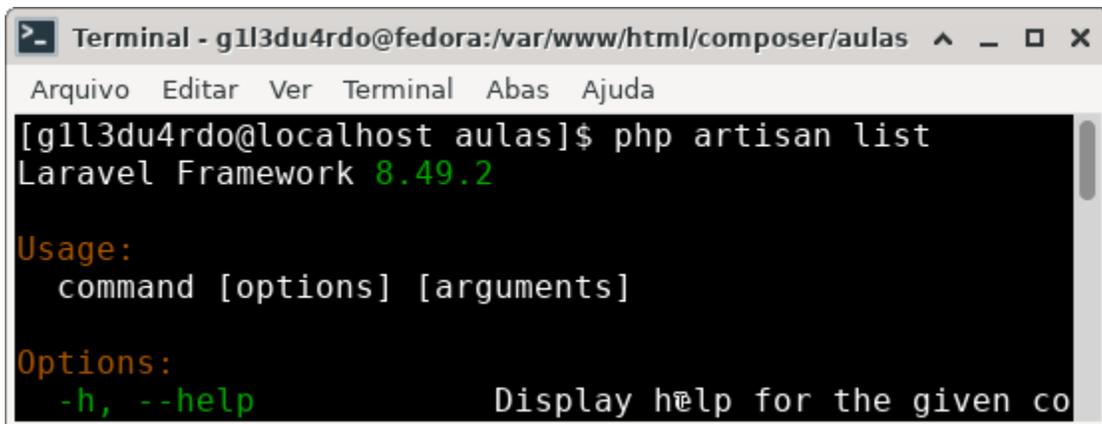
```
php artisan --version
```



```
Terminal - g1l3du4rdo@fedora:/var/www/html/composer/aulas ^ _ □ ×  
Arquivo Editar Ver Terminal Abas Ajuda  
[g1l3du4rdo@localhost aulas]$ php artisan --version  
Laravel Framework 8.49.2  
[g1l3du4rdo@localhost aulas]$
```

Lista de Comandos e Parâmetros - *php artisan*

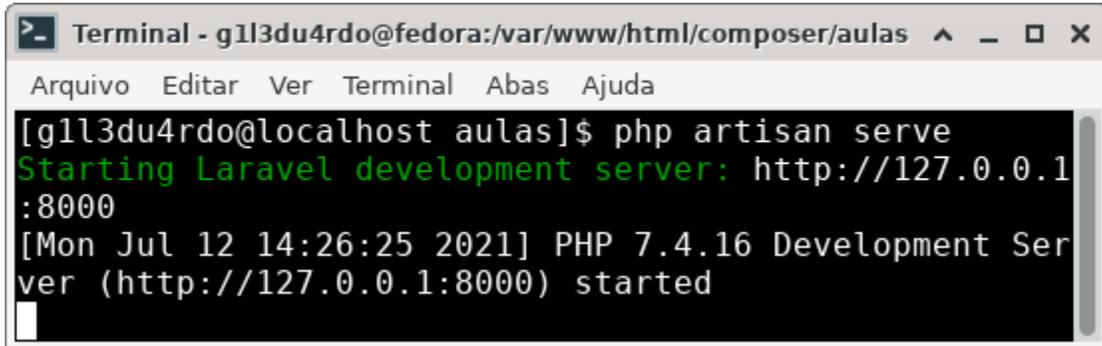
```
php artisan list
```



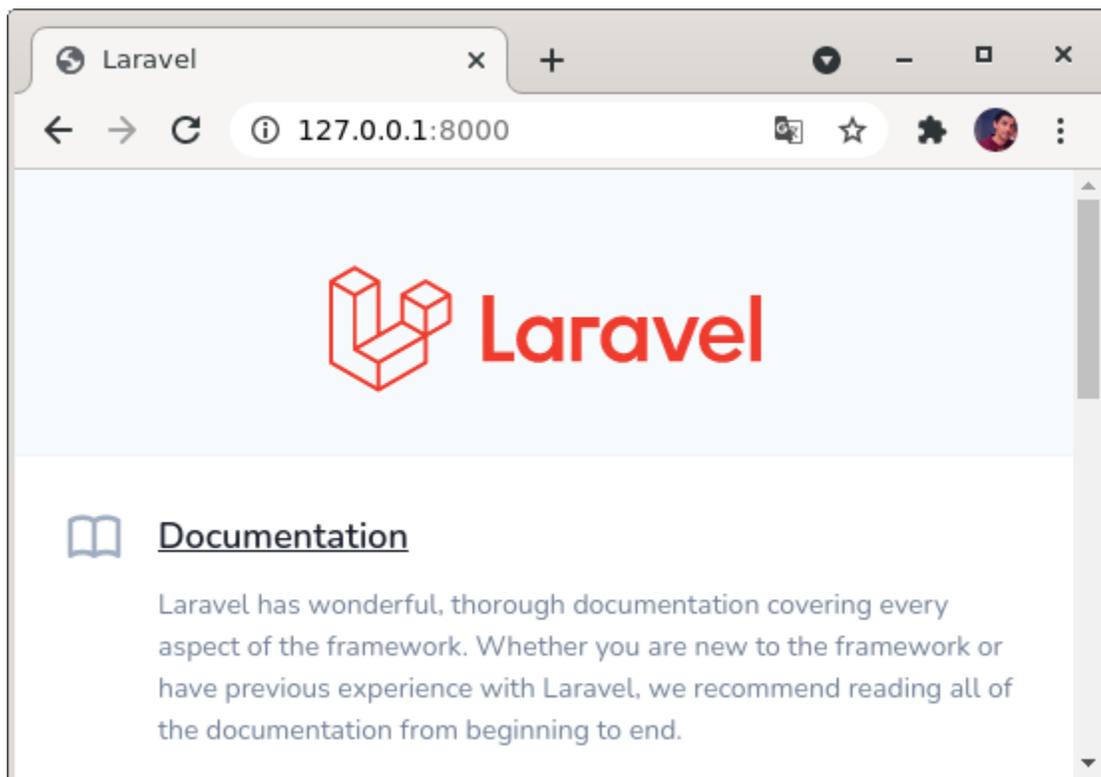
```
Terminal - g1l3du4rdo@fedora:/var/www/html/composer/aulas ^ _ □ ×  
Arquivo Editar Ver Terminal Abas Ajuda  
[g1l3du4rdo@localhost aulas]$ php artisan list  
Laravel Framework 8.49.2  
  
Usage:  
  command [options] [arguments]  
  
Options:  
  -h, --help                Display help for the given co
```

Iniciando a Aplicação / Executando o servidor (*host:porta - 127.0.0.1:8000*)

```
php artisan serve
```



```
Terminal - g1l3du4rdo@fedora:/var/www/html/composer/aulas ^ _ □ ×
Arquivo Editar Ver Terminal Abas Ajuda
[g1l3du4rdo@localhost aulas]$ php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Mon Jul 12 14:26:25 2021] PHP 7.4.16 Development Server (http://127.0.0.1:8000) started
```



Iniciando a Aplicação / Executando o servidor / Porta Específica

```
php artisan serve --port 10000
```



Codificação – Laravel / Rotas

(<https://laravel.com/docs/8.x/routing>)

Trabalhando com Rotas

(Arquivo: “/routes/web.php”)

```
<?php
use Illuminate\Support\Facades\Route;

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

Route::get('/', function () {
    return view('welcome');
});
```

Rotas - Laravel

[Route::get] - define uma rota do tipo GET para URL “/” (principal)
[return view(‘welcome’)] - define a view que será exibida ao acessar à rota “/”

Modificando a Rota “/”

(Arquivo: “/routes/web.php”)

```
<?php
use Illuminate\Support\Facades\Route;
Route::get('/', function () {
    return "<h2>Rota Principal</h2>";
});
```

Modificando a Rota “/” (principal)

[<h2>...</h2>] - Retorna o texto “Rota Principal” ao acessar a rota “/”

(Arquivo-fonte: 03-Rotas/routes/web.php)



← → ↻ ⓘ 127.0.0.1:8000

Rota Principal

(Execução da codificação anterior)

Criando uma Nova Rota (/cliente)

(Arquivo: “/routes/web.php”)

```
Route::get('/clientes', function() {
    $clientes = "<ul>
        <li>Carlos Eduardo</li>
        <li>Maria Cláudia</li>
        <li>João Pedro</li>
    </ul>";
    return $clientes;
});
```

```
})->name('clientes');
```

Criando a Rota “/cliente”

[route::get('/cliente', ...)] - define uma nova rota “/cliente”

(Arquivo-fonte: 03-Rotas/routes/web.php)

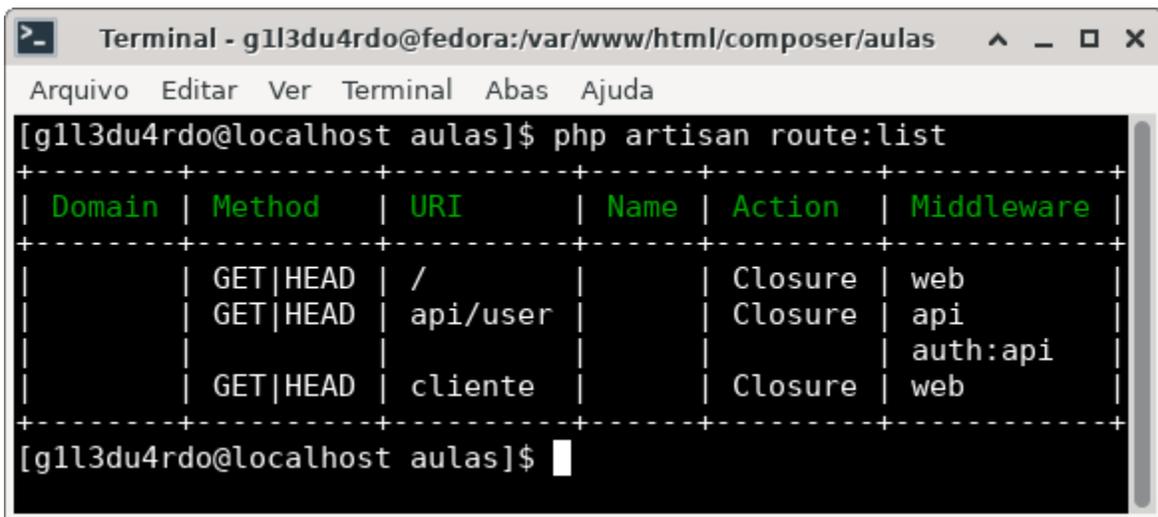
← → ↻ ⓘ 127.0.0.1:8000/cliente

- Carlos Eduardo
- Maria Claudia
- João Pedro

(Execução da codificação anterior)

Listando Rotas via *artisan*

`php artisan route:list`



```
Terminal - g1l3du4rdo@fedora:/var/www/html/composer/aulas
Arquivo Editar Ver Terminal Abas Ajuda
[g1l3du4rdo@localhost aulas]$ php artisan route:list
+-----+-----+-----+-----+-----+-----+
| Domain | Method | URI      | Name | Action | Middleware |
+-----+-----+-----+-----+-----+-----+
|         | GET|HEAD | /        |      | Closure | web        |
|         | GET|HEAD | api/user |      | Closure | api        |
|         | GET|HEAD | cliente  |      | Closure | auth:api   |
|         | GET|HEAD |         |      | Closure | web        |
+-----+-----+-----+-----+-----+-----+
[g1l3du4rdo@localhost aulas]$
```

Parâmetros - Rota (/cliente/{valor})

(Arquivo: “/routes/web.php”)

```
Route::get('/clientes/{total}', function($total) {
```

```
$dados = array(
    "Carlos Eduardo",
    "Maria Cláudia",
    "João Pedro"
);

$clientes = "<ul>";

if($total <= count($dados)) {
    $cont = 0;

    foreach($dados as $nome) {
        $clientes .= "<li>$nome</li>";
        $cont++;
        if($cont >= $total) break;
    }
}

else {
    $clientes = $clientes."<li>Tamanho Máximo = ".count($dados)."</li>";
}

$clientes .= "</ul>";

return $clientes;
});
```

Passando Parâmetro - Rota

[route::get(, function(\$valor))] - parâmetro passado (\$valor) para rota “/cliente”

(Arquivo-fonte: 03-Rotas/routes/web.php)

← → ↻ ⓘ 127.0.0.1:8000/cliente/2

- Carlos Eduardo
- Maria Claudia

(Execução da codificação anterior)

Parâmetros - Rota (/cliente/{total}/{nome})

(Arquivo: "/routes/web.php")

```
Route::get('/clientes/{total}/{nome}', function($total, $nome) {  
  
    $clientes = "<ul>";  
  
    for($cont=0; $cont<$total; $cont++) {  
  
        $clientes .= "<li>$nome</li>";  
  
    }  
  
    $clientes .= "</ul>";  
  
    return $clientes;  
  
});
```

Passando Parâmetro - Rota

(Arquivo-fonte: 03-Rotas/routes/web.php)

(Execução da codificação anterior)

Listando Rotas via *artisan*

```
php artisan route:list
```

Parâmetros Opcionais - Rota (/animal/{total}/{raça?})

(Arquivo: "/routes/web.php")

```
Route::get('/racas/{total}/{raca?}', function($total, $raca=null) {  
  
    $dados = array(  

```

```
"Bulldog Inglês",  
  
"Labrador",  
  
"Pastor Alemão",  
  
"Akita"  
  
);  
  
$pets = "<ul>";  
  
if($raca == null) {  
  
    if($total <= count($dados)) {  
  
        $cont = 0;  
  
        foreach($dados as $item) {  
  
            $pets .= "<li>$item</li>";  
  
            $cont++;  
  
            if($cont >= $total) break;  
  
        }  
  
    }  
  
    else {  
  
        $pets .= "<li>Tamanho Máximo = ".count($dados)."</li>";  
  
    }  
  
}  
  
else {  
  
    for($cont=0; $cont<$total; $cont++) {  
  
        $pets .= "<li>$raca</li>";  
  
    }  
  
}  
  
$pets .= "</ul>";  
  
return $pets;  
  
});
```

Parâmetro Opcional - Rota

`[/total/{raca?}]` - indica que o parâmetro “raca” é opcional
`[function(..., raca=null)]` - define o parâmetro “raca” como “null”, por padrão

(Arquivo-fonte: 03-Rotas/routes/web.php)

 127.0.0.1:8000/animal/3/

- Bulldog Inglês
- Street Dog
- Labrador

 127.0.0.1:8000/animal/5/Pastor

- Pastor
- Pastor
- Pastor
- Pastor
- Pastor

(Execução da codificação anterior)

Parâmetros com Regras - Rota (`/veterinario/{total}/{nome}`)

(Arquivo: “/routes/web.php”)

```
Route::get('/veterinarios/{total}/{nome}', function ($total, $nome) {  
  
    $vets = "<ul>";  
  
    for($a=0; $a<$total; $a++) {  
  
        $vets = $vets."<li>$nome</li>";  
  
    }  
  
    $vets = $vets."</ul>";  
  
    return $vets;  
  
})->where('total', '[0-9]+')->where('nome', '[A-Za-z]+');
```

Parâmetro com Regra - Rota

[->where('total', '[0-9]')] - indica que o parâmetro "total" deve ser um número
[('nome', '[A-Za-z]+')] - indica que o parâmetro "nome" deve ser uma string

(Arquivo-fonte: 03-Rotas/routes/web.php)

← → ↻ ⓘ 127.0.0.1:8000/veterinario/2/Rodolfo

- Rodolfo
- Rodolfo

← → ↻ ⓘ 127.0.0.1:8000/veterinario/2/3

404 | NOT FOUND

← → ↻ ⓘ 127.0.0.1:8000/veterinario/Gil/3

404 | NOT FOUND

(Execução da codificação anterior)

Agrupamento de Rotas (/consulta)

(Arquivo: "/routes/web.php")

```
Route::prefix('/consulta')->group(function() {  
  
    Route::get('/', function() {  
  
        return "<h2>Lista de Consultas</h2>";  
  
    });  
  
    Route::get('/agendar', function() {  
  
        return "<h2>Agendar Consulta</h2>";  
  
    });  
  
});
```

```
Route::get('/cancelar', function() {  
    return "<h2>Cancelar Consulta</h2>";  
});  
});
```

Agrupando as rotas “agendar” e “cancelar”

[route::prefix('consulta')->group] - configura o prefixo “consulta” para agrupar rotas
[Route::get] - define uma nova rota GET dentro do grupo “consulta”

(Arquivo-fonte: 03-Rotas/routes/web.php)

← → ↻ ⓘ 127.0.0.1:8000/consulta

Lista de Consultas

← → ↻ ⓘ 127.0.0.1:8000/consulta/agendar

Agendar Consulta

← → ↻ ⓘ 127.0.0.1:8000/consulta/cancelar

Cancelar Consulta

(Execução da codificação anterior)

Criando/Retornando uma View (/consulta)

(Arquivo: “/resources/views/consulta.blade.php”)

```
<h2 style="color: blue">Lista de Consultas</h2>
```

(Arquivo: “/resources/views/agendar.blade.php”)

```
<h2 style="color: red">Agendar Consulta</h2>
```

(Arquivo: “/resources/views/cancelar.blade.php”)

```
<h2 style="color: green">Cancelar Consulta</h2>
```

(Arquivo: “/routes/web.php”)

```
Route::prefix('/consulta')->group(function() {  
  
    Route::get('/', function() {  
  
        return view('consulta');  
  
    });  
  
    Route::get('/agendar', function() {  
  
        return view('agendar');  
  
    });  
  
    Route::get('/cancelar', function() {  
  
        return view('cancelar');  
  
    });  
  
});
```

Criando e Retornando View - Rota

[return view('agendar')] - retorna a view “agendar”, armazenada em “/resources/views”

(Arquivo-fonte: 03-Rotas/routes/web.php)



← → ↻ ⓘ 127.0.0.1:8000/consulta

Lista de Consultas



← → ↻ ⓘ 127.0.0.1:8000/consulta/agendar

Agendar Consulta

← → ↻ ⓘ 127.0.0.1:8000/consulta/cancelar

Cancelar Consulta

(Execução da codificação anterior)

Navegando entre Rotas (/consulta)

(Arquivo: “/resources/views/consultas.blade.php”)

```
<h2 style="color: blue">Lista de Consultas</h2>
<a href="/consultas/agendar">Agendar</a>
<br>
<a href="/consultas/cancelar">Cancelar</a>
```

← → ↻ ⓘ 127.0.0.1:8000/consulta

Lista de Consultas

[Agendar](#)
[Cancelar](#)

(Execução da codificação anterior)

Nomenado Rotas (/consulta)

(Arquivo: “/routes/web.php”)

```
Route::prefix('/consultas')->group(function() {
    Route::get('/', function() {
        return view('consultas');
    })->name('consultas');
    Route::get('/agendar', function() {
        return view('agendar');
    })->name('consultas.agendar');
    Route::get('/cancelar', function() {
```

```
return view('cancelar');  
})->name('consultas.cancelar');  
});
```

Nomeando Rotas

```
[->name('consulta')] - nomeia a rota "consulta" como "consulta"  
[->name('consulta.agendar')] - nomeia a rota "agendar" como "consulta.agendar"  
[->name('consulta.cancelar')] - nomeia a rota "cancelar" como "consulta.cancelar"
```

(Arquivo-fonte: 03-Rotas/routes/web.php)

(Arquivo: "/resources/views/consulta.blade.php")

```
<h2 style="color: blue">Lista de Consultas</h2>  
<a href="{{route('consulta.agendar')}}">Agendar</a>  
<br>  
<a href="{{route('consulta.cancelar')}}">Cancelar</a>
```

Nomeando Rotas

```
[[{route()}]] - método blade utilizado para navegar para outra rota
```

(Arquivo-fonte: 03-Rotas/resources/views/consultas.php)

Redirecionando Rotas / *Route::redirect* ("/ -> "/cliente")

(Arquivo: "/routes/web.php")

```
Route::get('/cliente', function () {  
  
    $clientes = "<ul>".  
  
        "<li>Carlos Eduardo</li>".  
  
        "<li>Maria Claudia</li>".
```

```
"<li>João Pedro</li>".  
  
"</ul>";  
  
return $clientes;  
})->name('cliente');  
  
...  
Route::redirect('/', 'cliente', 301);
```

Redirecionar Rota - Route::redirect()

[Route::redirect()] - redireciona o acesso a rota "/" para rota "/cliente"

(Arquivo-fonte: 03-Rotas/routes/web.php)

Redirecionando Rotas / *redirect()->route()* ("/veterinario" -> "/cliente")

(Arquivo: "/routes/web.php")

```
Route::get('/veterinario', function () {  
  
    return redirect()->route('cliente');  
  
});
```

Redirecionar Rota - redirect()->route()

[redirect()->route('cliente')] - redireciona o acesso "/veterinario" para "/cliente"

(Arquivo-fonte: 03-Rotas/routes/web.php)

Requisição HTTP POST (/veterinario/add)

(Arquivo: "/routes/web.php")

```
use \Illuminate\Http\Request;  
  
...  
Route::post('/veterinarios/add', function (Request $request) {
```

```
return "<h1>Adicionar Veterinário (POST)</h1>";  
});
```

Requisição HTTP / Método POST

[Route::post()] - define uma nova rota do tipo POST

[Request \$request] - objeto que possui os parâmetros enviados na requisição POST

(Arquivo-fonte: 03-Rotas/routes/web.php)

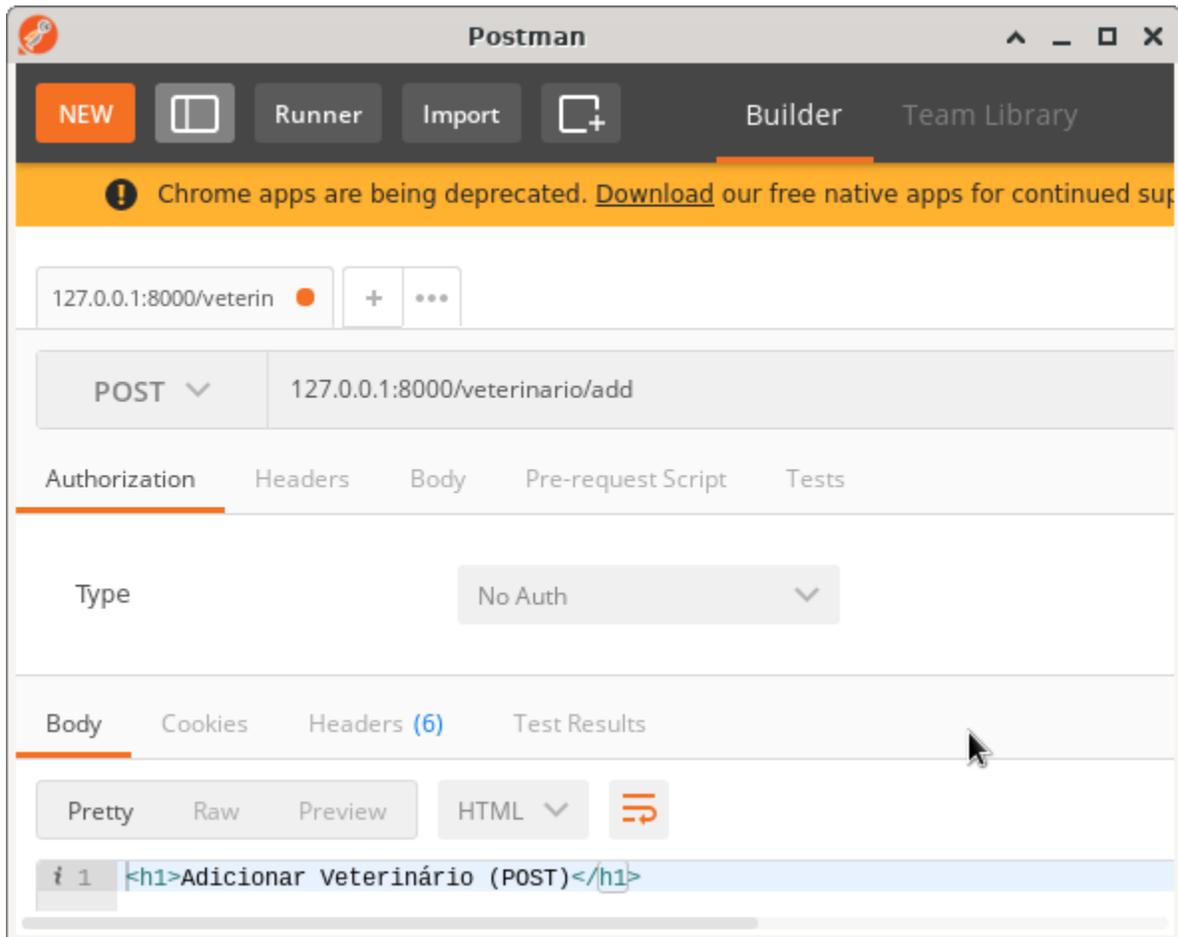
(Arquivo: "/app/Http/Middleware/VerifyCsrfToken.php")

```
protected $except = [  
  
    //  
  
    'veterinarios/add'  
  
];
```

Requisição HTTP / Método POST

['veterinario/add'] - adiciona a rota como exceção - Verificação CSRF Token

(Arquivo-fonte: 03-Rotas/app/Http/Middleware/VerifyToken.php)



(Execução da codificação anterior)