

Curso Laravel 5.4 (Aulas 1 a 50)

SYS4SOFT (João Ribeiro)

https://www.youtube.com/watch?v=OT5gM1WRNsY&list=PLXik_5Br-zO893qVjjP7a4gg4NYrl33w1&index=1

Resumo do curso feito por Roberto Pinheiro

No Google Chrome instale a extensão: **json formatter**.

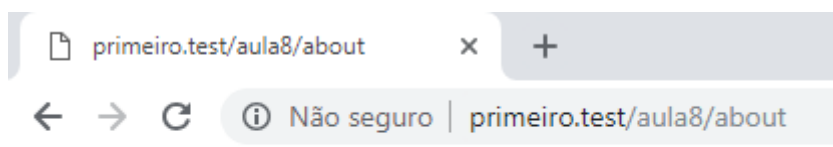
Aula 008 – Laravel Routing

routes/web.php

```
Route::get('aula8/about', function () {  
    return view('aula8/acerca_de');  
});
```

views/aula8/acerca_de.php

```
<h1>Aqui estou eu!</h1>
```

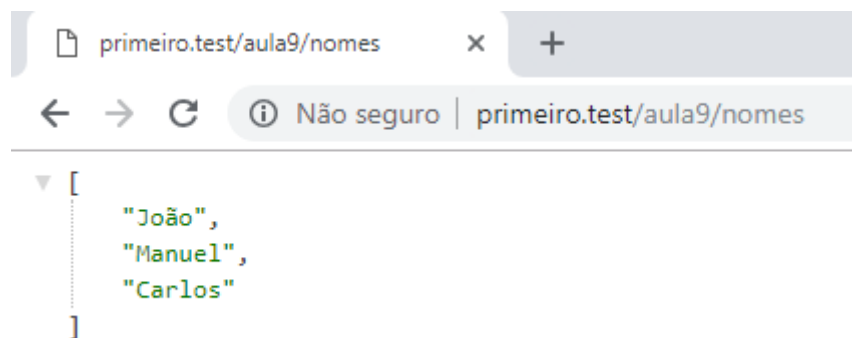


Aqui estou eu!

Aula 009 – Pequenos testes com Routes

routes/web.php

```
Route::get('aula9/about', function () {  
  
    $nomes = ['João','Manuel','Carlos'];  
  
    return $nomes;  
});
```



routes/web.php

```
Route::get('aula9/primeira', function () {  
    return view('aula9/primeira');  
});
```

```
Route::get('aula9/segunda', function () {  
    return view('aula9/segunda');  
});
```

```
Route::get('aula9/terceira', function () {  
    return view('aula9/terceira');  
});
```

views/aula9/primeira.php

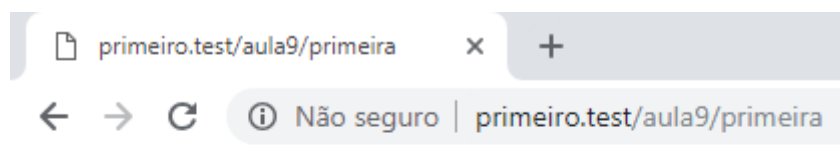
```
<h1>Primeira</h1>  
<a href="segunda">Segunda</a>
```

views/aula9/segunda.php

```
<h1>Segunda</h1>  
<a href="terceira">Terceira</a>
```

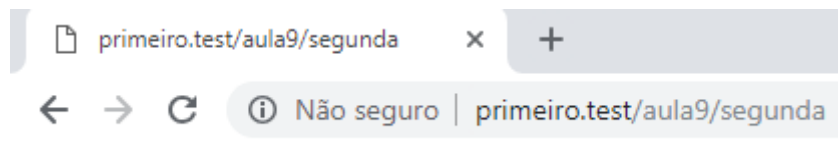
views/aula9/terceira.php

```
<h1>Terceira</h1>  
<a href="primeira">Primeira</a>
```



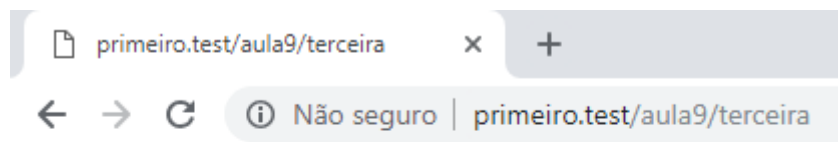
Primeira

[Segunda](#)



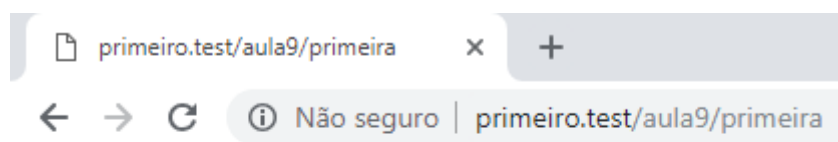
Segunda

[Terceira](#)



Terceira

[Primeira](#)



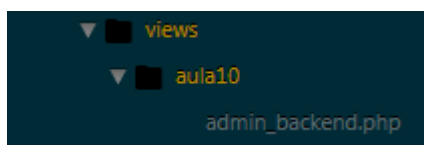
Primeira

[Segunda](#)

Aula 010 – Continuando a falar de routes simples

routes/web.php

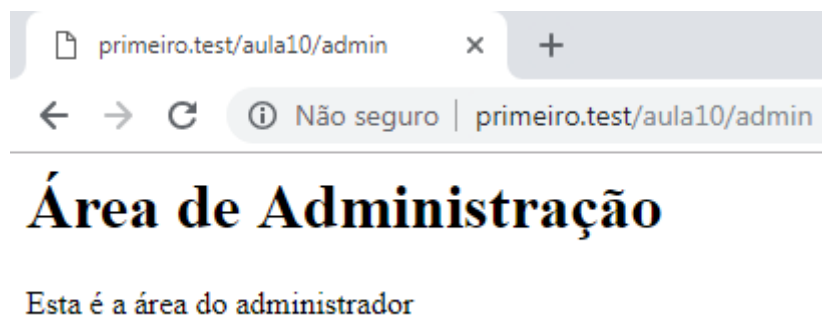
```
/* Routes de administração */  
  
Route::get('aula10/admin', function () {  
    return view('aula10/admin_backend');  
});
```



views/aula10/admin.backend.php

```
<h1>Área de Administração</h1>
```

```
<p>Esta é a área do administrador</p>
```



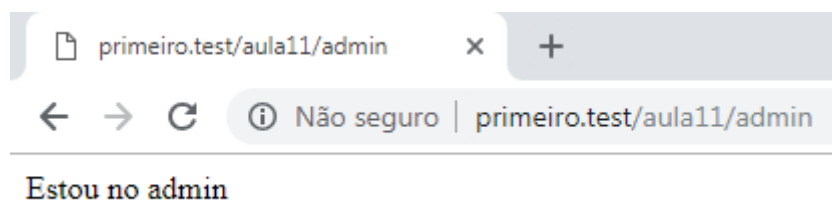
Aula 011 – Passar valores através das Views

routes/web.php

```
Route::get('aula11/admin', function () {  
    return view('aula11/admin/admin');  
});
```

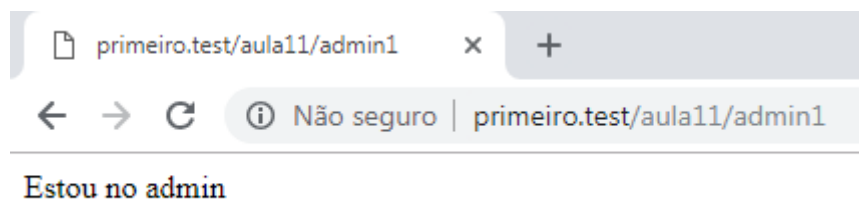
views/aula11/admin/admin.php

```
<?php  
    echo 'Estou no admin';  
?>
```



routes/web.php

```
Route::get('aula11/admin1', function () {  
    return view('aula11.admin.admin');  
});
```

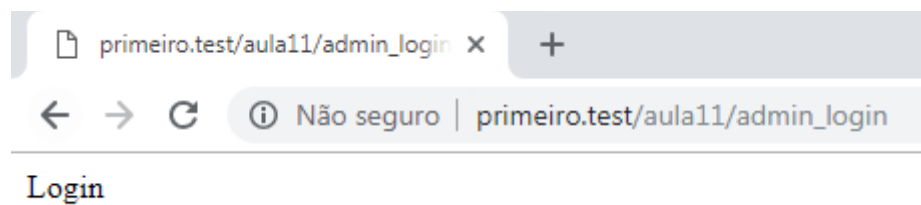


routes/web.php

```
Route::get('aula11/admin_login', function () {  
    return view('aula11.admin.login');  
});
```

views/aula11/admin/admin_login.php

```
<?php  
    echo 'Login';  
?>
```

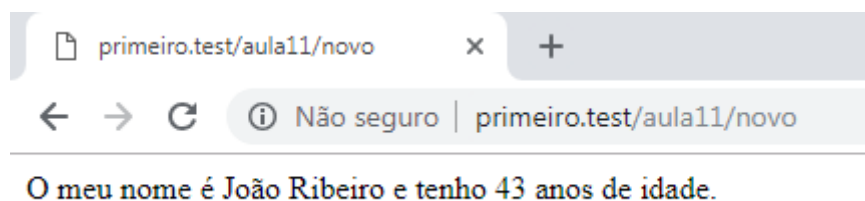


routes/web.php

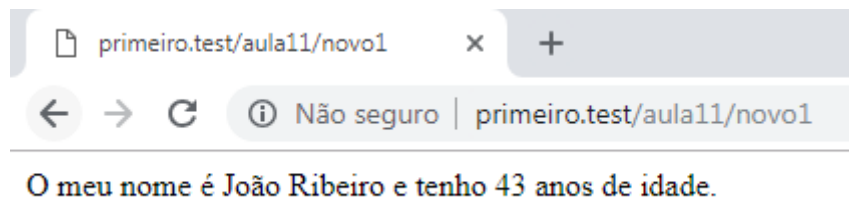
```
Route::get('aula11/novo', function () {  
  
    $i = ['nome' => 'João Ribeiro', 'idade' => 43];  
  
    return view('aula11/novo',$i);  
});
```

views/aula11/novo.php

```
<?php  
  
    echo "O meu nome é " . $nome . " e tenho " . $idade . " anos de idade."  
  
?>
```



```
Route::get('aula11/novo1', function () {  
    return view('aula11/novo',['nome'=>'João Ribeiro'],['idade'=>43]);  
});
```

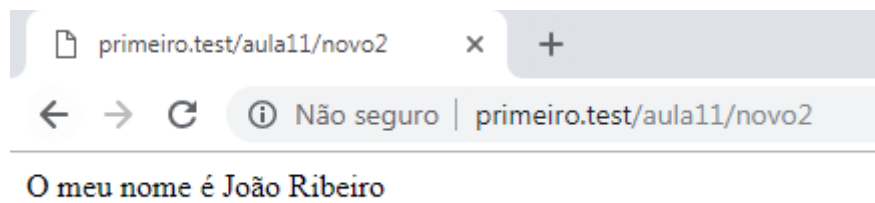


routes/web.php

```
Route::get('aula11/novo2', function () {  
  
    return view('aula11/novo2')->with('nome','João Ribeiro');  
});
```

views/aula11/novo2.php

```
<?php  
  
    echo "O meu nome é " . $nome;  
  
?>
```



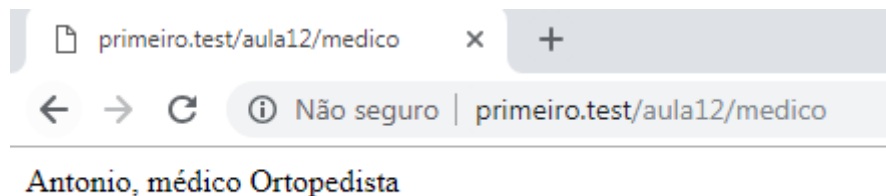
Aula 012 – Passar array multidimensional através das Views

routes/web.php

```
Route::get('aula12/medico', function () {  
  
    $medicos = [  
        'nome_medico'=>'Antonio',  
        'especialidade'=>'Ortopedista'  
    ];  
  
    return view('aula12/medico',$medicos);  
});
```

views/aula12/medico.php

```
<?php  
  
    echo "<p>$nome_medico, médico $especialidade</p>";  
  
?>
```



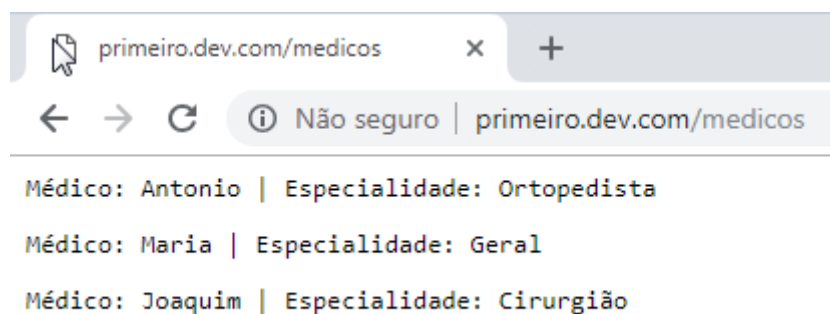
routes/web.php

```
Route::get('aula12/medicos', function () {
    $medicos = [
        0 => [
            'nome_medico'=>'Antonio',
            'especialidade'=>'Ortopedista'
        ],
        1 => [
            'nome_medico'=>'Maria',
            'especialidade'=>'Geral'
        ],
        2 => [
            'nome_medico'=>'Joaquim',
            'especialidade'=>'Cirurgião'
        ]
    ];
    return view('aula12/medicos') -> with('dados',$medicos);
});
```

views/aula12/medicos.php

```
<?php
// var_dump($dados);

foreach ($dados as $medico) {
    echo "<p>Médico: " . $medico['nome_medico'] . " | Especialidade: " .
    $medico['especialidade'] . "</p>";
}
?>
```



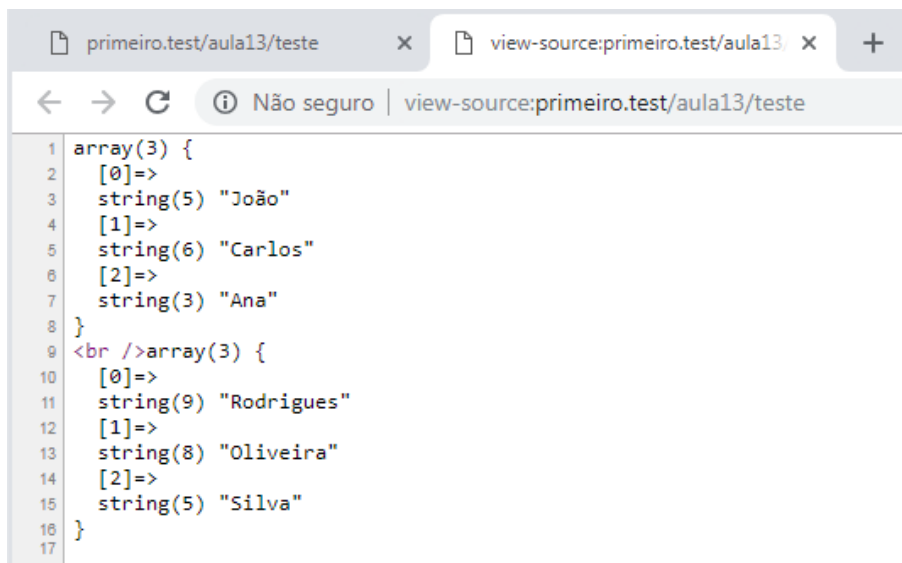
Aula 013 – Routes com parâmetros

routes/web.php

```
Route::get('aula13/teste', function () {
    $nomes = [
        'João',
        'Carlos',
        'Ana'
    ];
    $apelidos = [
        'Rodrigues',
        'Oliveira',
        'Silva'
    ];
    return view('aula13/teste',compact('nomes','apelidos'));
});
```

views/aula13/teste.php

```
<?php
    var_dump($nomes);
    echo "<br />";
    var_dump($apelidos);
?>
```



routes/web.php

```
Route::get('aula13/medicos_da_clinica/{id_medico}', function ($id_medico) {  
  
    return view('aula13/medicos') -> with('id', $id_medico);  
});
```

views/aula13/medicos.php

```
<?php
```

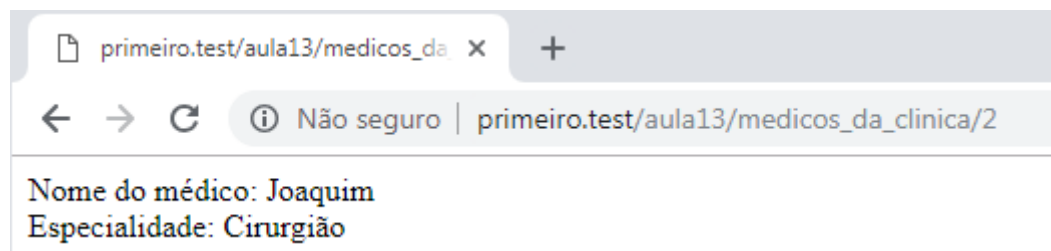
```
$medicos = [
```

```
    0 => [  
        'nome_medico'=>'Antonio',  
        'especialidade'=>'Ortopedista'  
    ],  
    1 => [  
        'nome_medico'=>'Maria',  
        'especialidade'=>'Geral'  
    ],  
    2 => [  
        'nome_medico'=>'Joaquim',  
        'especialidade'=>'Cirurgião'  
    ]  
];
```

```
// echo $id;
```

```
echo "Nome do médico: " . $medicos[$id]['nome_medico'] . "<br />Especialidade: " .  
$medicos[$id]['especialidade'];
```

```
?>
```



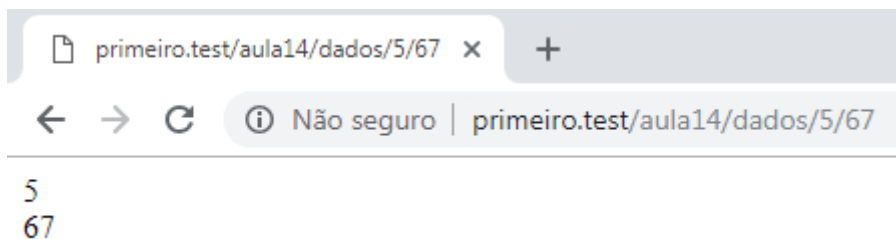
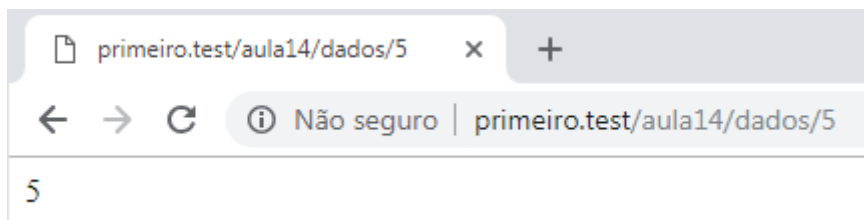
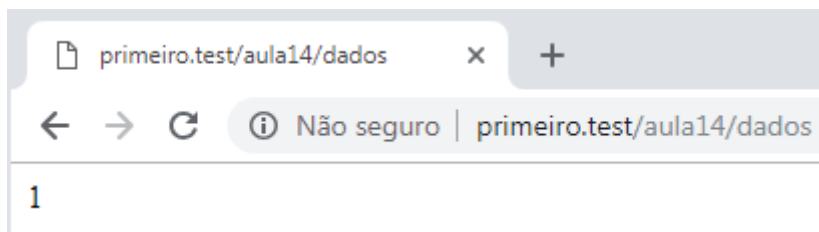
Aula 014 – Routes com múltiplos parâmetros e parâmetro padrão

routes/web.php

```
Route::get('aula14/dados/{id_medico?}/{id_consulta?}', function ($id_medico = 1, $id_consulta = null) {  
    return view('aula14/dados') -> with('id_medico', $id_medico) -> with('id_consulta', $id_consulta);  
});
```

views/aula14/dados.php

```
<?php  
    echo $id_medico . '<br />' . $id_consulta;  
?>
```



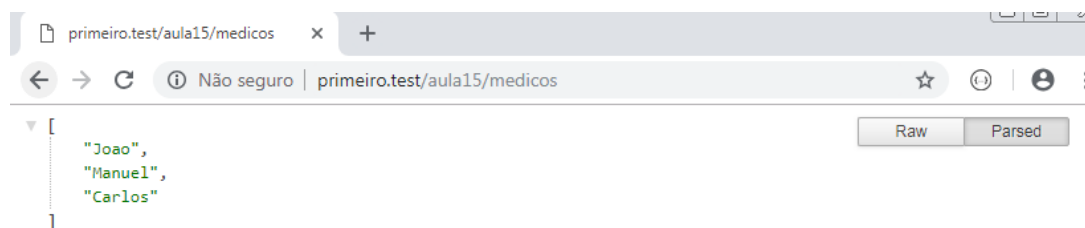
Aula 015 – Controllers

routes/web.php

```
Route::get('aula15/medicos', function () {

    $medicos = [
        'Joao',
        'Manuel',
        'Carlos'
    ];

    return $medicos;
});
```



Criando um controller

No terminal do Laragon, na pasta do projeto (C:\\laragon\\www\\primeiro), crie o controller chamado `medicosController`, na pasta `aula15`, entrando com o comando:

```
php artisan make:controller aula15\\medicosController
```

app/Http/Controllers/aula15/medicosController.php

```
<?php

namespace App\\Http\\Controllers\\aula15;

use Illuminate\\Http\\Request;
use App\\Http\\Controllers\\Controller;

class medicosController extends Controller
{
    //
}
```

routes/web.php

```
Route::get('aula15/medicosListar', ['uses' => 'aula15\medicosController@GetMedicos']);
Route::get('aula15/medicosApagar', ['uses' => 'aula15\medicosController@EliminarMedicos']);
```

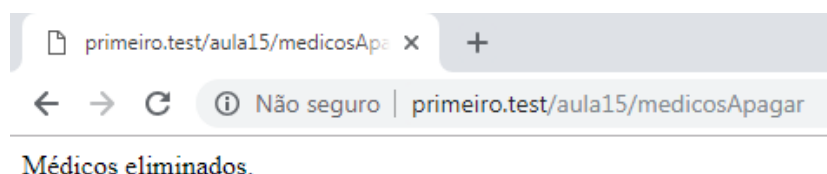
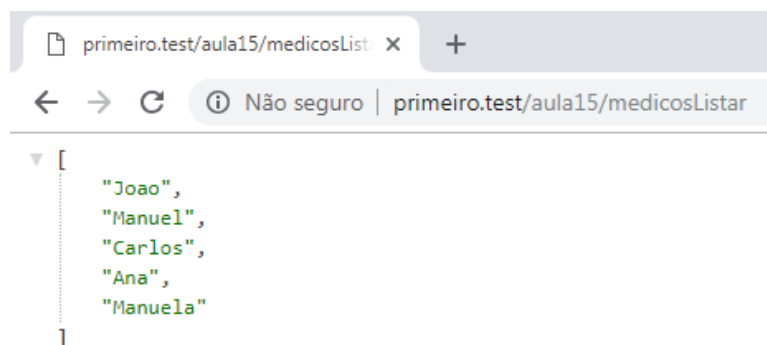
app/Http/Controllers/aula15/medicosController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

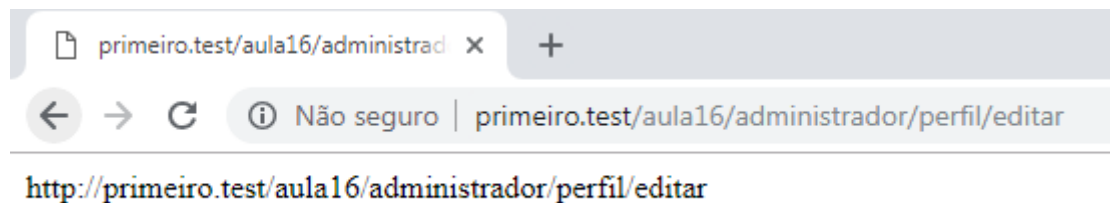
class medicosController extends Controller
{
    public function GetMedicos(){
        $medicos = [
            'Joao',
            'Manuel',
            'Carlos',
            'Ana',
            'Manuela'
        ];
        return $medicos;
    }
    public function EliminarMedicos(){
        return "Médicos eliminados.";
    }
}
```



Aula 016 – Nomeando Routes

routes/web.php

```
Route::get('aula16/administrador/perfil/editar', array('as' => 'adm_per_edit', function(){  
    // return "Esta route foi renomeada";  
    return route('adm_per_edit');  
}));
```



Listando as routes criadas

No terminal do Laragon, na pasta do projeto (C:\laragon\www\primeiro), entre com o comando:

```
php artisan route:list
```

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	GET HEAD	about		Closure	web
	GET HEAD	administrador/configuracoes/editar	adm_cfg_edit	Closure	web
	GET HEAD	api/user		Closure	api,auth:api
	GET HEAD	aula10/admin		Closure	web
	GET HEAD	aula11/admin		Closure	web
	GET HEAD	aula11/admin1		Closure	web
	GET HEAD	aula11/admin_login		Closure	web
	GET HEAD	aula11/novo		Closure	web
	GET HEAD	aula11/novo1		Closure	web
	GET HEAD	aula11/novo2		Closure	web
	GET HEAD	aula12/medico		Closure	web
	GET HEAD	aula12/medicos		Closure	web
	GET HEAD	aula13/medicos_da_clinica/{id_medico}		Closure	web
	GET HEAD	aula13/teste		Closure	web
	GET HEAD	aula14/dados/{id_medico}/{id_consulta}		Closure	web
	GET HEAD	aula15/medicos		Closure	web
	GET HEAD	aula15/medicosApagar		App\Http\Controllers\medicosController@EliminarMedicos	web
	GET HEAD	aula15/medicoslistar		App\Http\Controllers\medicosController@GetMedicos	web
	GET HEAD	aula16/administrador/perfil/editar	adm_per_edit	Closure	web
	GET HEAD	aula8/about		Closure	web
	GET HEAD	aula9/nomes		Closure	web
	GET HEAD	aula9/primeira		Closure	web
	GET HEAD	aula9/segunda		Closure	web
	GET HEAD	aula9/terceira		Closure	web

Aula 017 – Utilização de Routes nomeadas

routes/web.php

```
Route::get('aula17/administrador/configuracoes/editar', array('as' => 'adm_cfg_edit',  
function(){  
    return view('aula17/admin/config/editar');  
}));
```



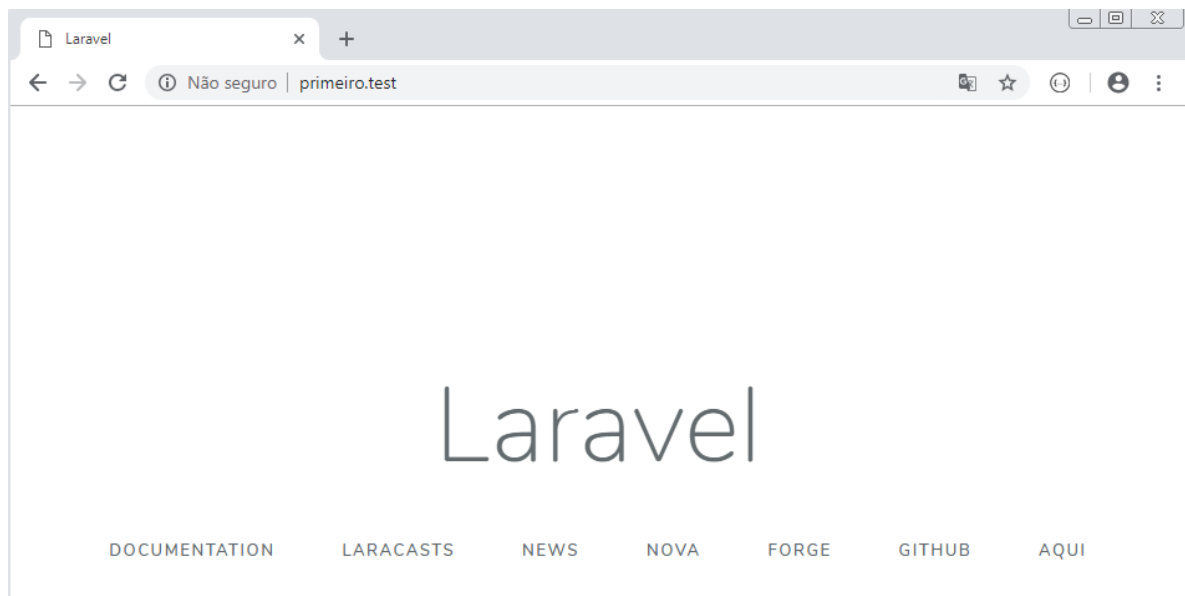
views/aula17/admin/config/editar.php

```
<?php  
    echo '<h1>Estou aqui!</h1>';  
?>
```

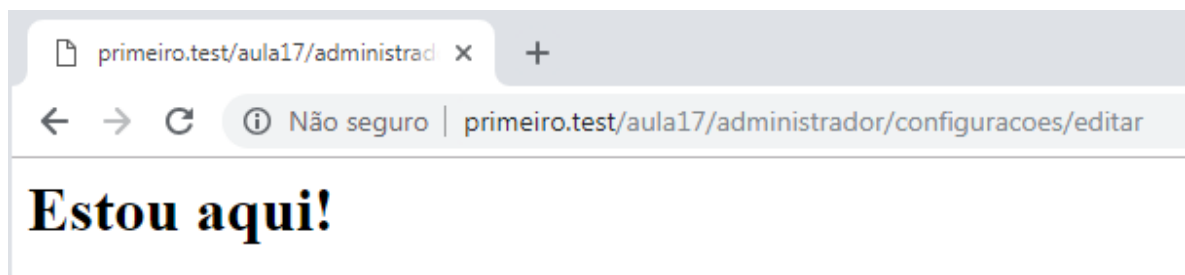
No arquivo **welcome.blade.php**, inserir a linha em negrito:

welcome.blade.php

```
<div class="links">  
    <a href="https://laravel.com/docs">Documentation</a>  
    <a href="https://laracasts.com">Laracasts</a>  
    <a href="https://laravel-news.com">News</a>  
    <a href="https://nova.laravel.com">Nova</a>  
    <a href="https://forge.laravel.com">Forge</a>  
    <a href="https://github.com/laravel/laravel">GitHub</a>  
    <a href="{{ route('adm_cfg_edit') }}">AQUI</a>  
</div>
```



Clicando no link AQUI:

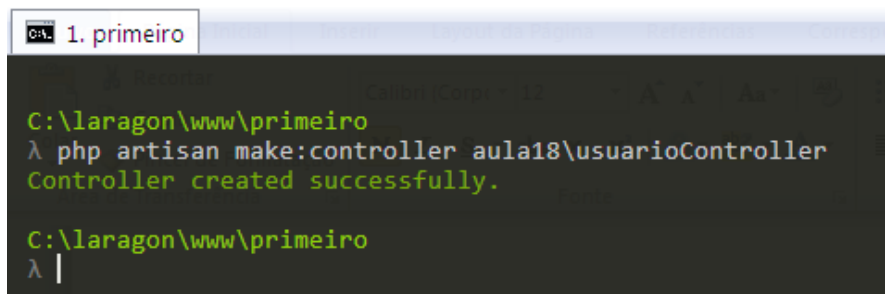


Aula 018 – Passando dados entre Routes e Controllers

Criando o controller

No terminal do Laragon, na pasta do projeto (C:\laragon\www\primeiro), crie o controller chamado **usuarioController**, na pasta **aula18**, entrando com o comando:

```
php artisan make:controller aula18\usuarioController
```



```
1. primeiro
C:\laragon\www\primeiro
λ php artisan make:controller aula18\usuarioController
Controller created successfully.
C:\laragon\www\primeiro
λ |
```

routes/web.php

```
Route::get('aula18/ver_usuario/{nome}/{senha}', 'aula18\usuarioController@ApresentarUsuario');
```

app/Http/Controllers/aula18/usuarioController.php

```
<?php
```

```
namespace App\Http\Controllers\aula18;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

```
class usuarioController extends Controller
```

```
{
```

```
    function ApresentarUsuario($nome, $senha){
```

```
        return "<p>Usuario: " . $nome . "</p><p>Senha: " . $senha . "</p>";
```

```
    }
```

```
    function Tratar($usuario, $senha){
```

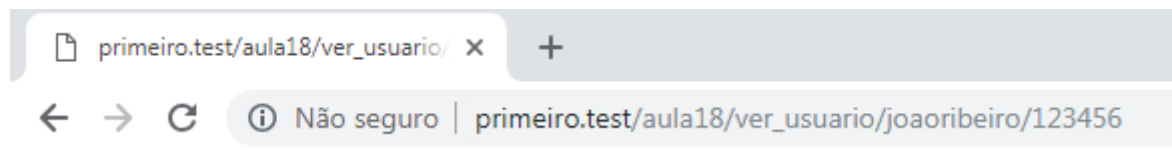
```
        $usuario_tratado = "Bem vindo " . $usuario;
```

```
        $senha_tratada = md5($senha);
```

```
        return view('mostrar_usuario', compact('usuario_tratado', 'senha_tratada'));
```

```
    }
```

```
}
```



Usuario: joaoribeiro

Senha: 123456

Aula 019 – De Route para Controller e de Controller para View

routes/web.php

```
Route::get('aula18/tratar_usuario/{usuario}/{senha}', 'aula18\usuarioController@Tratar');
```

app/Http/Controllers/aula18/usuarioController.php

```
<?php

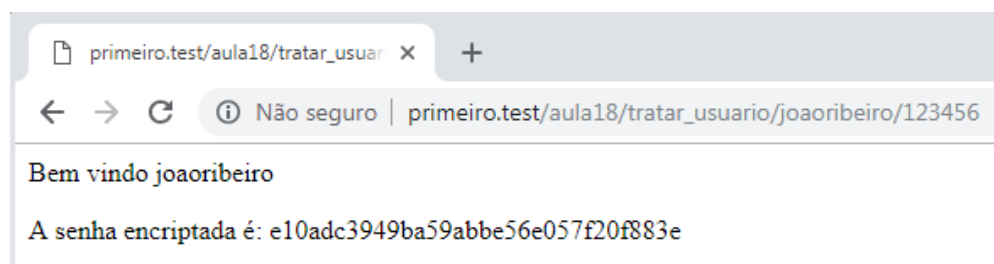
namespace App\Http\Controllers\aula18;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class usuarioController extends Controller
{
    function ApresentarUsuario($nome, $senha){
        return "<p>Usuario: " . $nome . "</p><p>Senha: " . $senha . "</p>";
    }
    function Tratar($usuario, $senha){
        $usuario_tratado = "Bem vindo " . $usuario;
        $senha_tratada = md5($senha);

        return view('aula18/mostrar_usuario', compact('usuario_tratado', 'senha_tratada'));
    }
}
```

mostrar_usuario.php

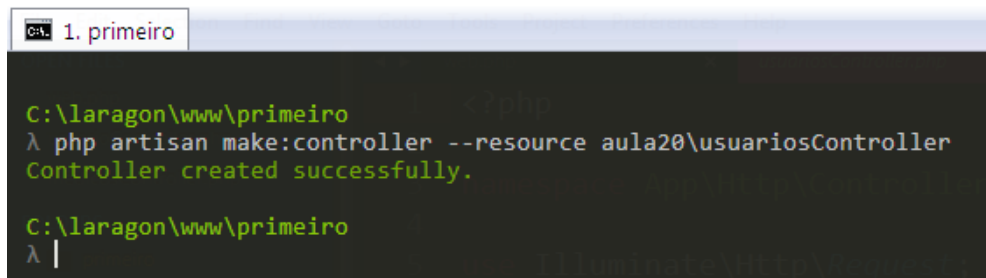
```
<?php
    echo "<p>" . $usuario_tratado . "</p>";
    echo "<p>A senha encriptada é: " . $senha_tratada;
?>
```



Aula 020 – Resource Controller

No terminal do Laragon, na pasta do projeto (C:\laragon\www\primeiro), crie o controller chamado usuariosController, na pasta aula20, entrando com o comando:

```
php artisan make:controller --resource aula20\usuariosController
```



```
C:\laragon\www\primeiro
λ php artisan make:controller --resource aula20\usuariosController
Controller created successfully.

C:\laragon\www\primeiro
λ |
```

Será criado um controller com métodos CRUD:

routes/web.php

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

```
Route::resource('usuarios', 'usuariosController');
```

No terminal do Laragon, na pasta do projeto (C:\laragon\www\primeiro), entre com o comando:

php artisan route:list

```
C:\laragon\www\primeiro
^ php artisan route:list
```

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	GET HEAD	about		Closure	web
	GET HEAD	admin		Closure	web
	GET HEAD	admin1		Closure	web
	GET HEAD	admin_login		Closure	web
	GET HEAD	administrador/configuracoes/editar	adm_cfg_edit	Closure	web
	GET HEAD	administrador/perfil/editar	adm_per_edit	Closure	web
	GET HEAD	api/user		Closure	api,auth:api
	GET HEAD	dados/{id_medico}/{id_consulta?}		Closure	web
	GET HEAD	medico		Closure	web
	GET HEAD	medicos		Closure	web
	GET HEAD	medicosApagar		App\Http\Controllers\medicosController@EliminarMedicos	web
	GET HEAD	medicosListar		App\Http\Controllers\medicosController@GetMedicos	web
	GET HEAD	medicos_da_clinica/{id_medico}		Closure	web
	GET HEAD	medicos_plantao		Closure	web
	GET HEAD	nomes		Closure	web
	GET HEAD	novo		Closure	web
	GET HEAD	novo1		Closure	web
	GET HEAD	novo2		Closure	web
	GET HEAD	primeira		Closure	web
	GET HEAD	segunda		Closure	web
	GET HEAD	tancaira		Closure	web
	GET HEAD	teste		Closure	web
	GET HEAD	tratar_usuario/{usuario}/{senha}		App\Http\Controllers\usuarioController@Tratar	web
	GET HEAD	usuarios	usuarios.index	App\Http\Controllers\usuariosController@index	web
	POST	usuarios	usuarios.store	App\Http\Controllers\usuariosController@store	web
	GET HEAD	usuarios/create	usuarios.create	App\Http\Controllers\usuariosController@create	web
	PUT PATCH	usuarios/{usuario}	usuarios.update	App\Http\Controllers\usuariosController@update	web
	DELETE	usuarios/{usuario}	usuarios.destroy	App\Http\Controllers\usuariosController@destroy	web
	GET HEAD	usuarios/{usuario}	usuarios.show	App\Http\Controllers\usuariosController@show	web
	GET HEAD	usuarios/{usuario}/edit	usuarios.edit	App\Http\Controllers\usuariosController@edit	web
	GET HEAD	ver_usuario/{nome}/{senha}		App\Http\Controllers\usuarioController@ApresentarUsuario	web

```
C:\laragon\www\primeiro
^ |
```

views/aula20/indice.php

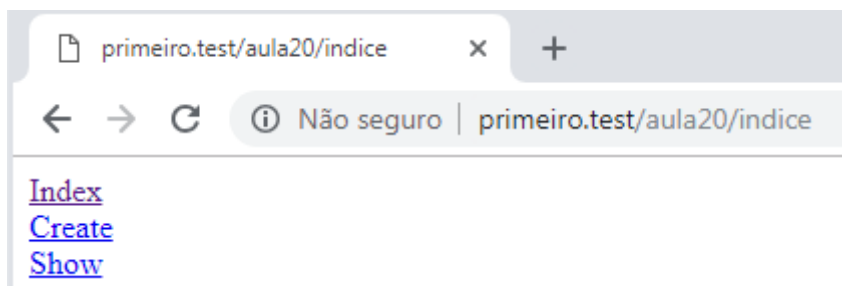
```
<?php
    $id_usuario = 10;
    echo "<a href=" . route('usuarios.index') . ">Index</a><br />";
    echo "<a href=" . route('usuarios.create') . ">Create</a><br />";
    echo "<a href=" . route('usuarios.show', $id_usuario) . ">Show</a>";
?>
```


app/Http/Controllers/aula20/usuariosController.php

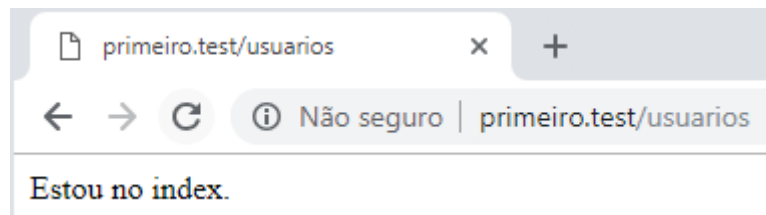
```
<?php

namespace App\Http\Controllers;
use Illuminate\Http\Request;

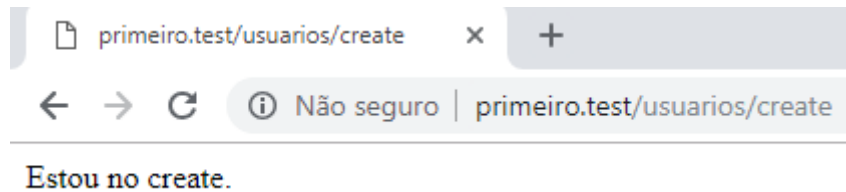
class usuariosController extends Controller
{
    public function index()
    {
        return "Estou no index.";
    }
    public function create()
    {
        return "Estou no create.";
    }
    public function store(Request $request)
    {
        //
    }
    public function show($id)
    {
        return "O usuário a apresentar tem o ID: " . $id;
    }
    public function edit($id)
    {
        //
    }
    public function update(Request $request, $id)
    {
        //
    }
    public function destroy($id)
    {
        //
    }
}
```



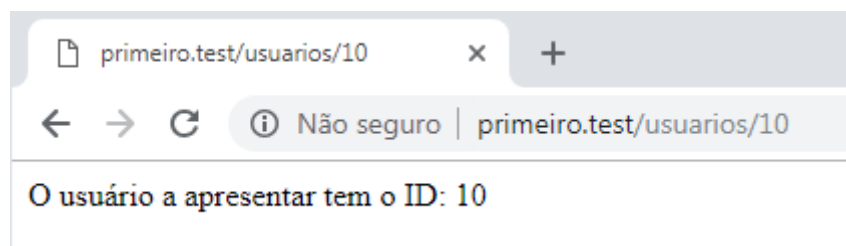
Clicando em “Index”:



Clicando em “Create”:



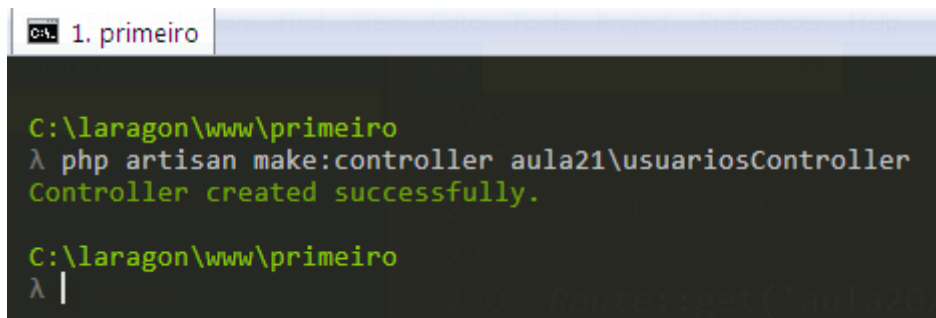
Clicando em “Show”:



Aula 021 – Blade Template Engine

No terminal do Laragon, na pasta do projeto (C:\laragon\www\primeiro), crie o controller chamado usuariosController, na pasta aula21, entrando com o comando:

```
php artisan make:controller aula21\usuariosController
```



```
C:\laragon\www\primeiro
λ php artisan make:controller aula21\usuariosController
Controller created successfully.

C:\laragon\www\primeiro
λ |
```

routes/web.php

```
Route::get('aula21/usuarios', 'aula21\usuariosController@MostrarUsuarios');
```

app/Http/Controllers/aula21/usuariosController.php

```
<?php
```

```
namespace App\Http\Controllers\aula21;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
```

```
class usuariosController extends Controller
{
    public function MostrarUsuarios(){
        $usuarios = [
            'Joao',
            'Maria',
            'Antonio'
        ];
        // $usuarios = [];
        // $usuarios = ['João'];
        // apresentar a view com os usuários
        return view('aula21/show_usuarios', compact('usuarios'));
    }
}
```

views/aula21/show_usuarios.blade.php

<div>

<h1>Blade Template Engine</h1>

<p>Usuários registrados na plataforma (PHP): <?php echo count(\$usuarios); ?></p>

<p>Usuários registrados na plataforma (BLADE): {{count(\$usuarios)}}</p>

</div>



Aula 022 – Blade – Sintaxe básica

No terminal do Laragon, na pasta do projeto (C:\laragon\www\primeiro), crie o controller chamado **usuariosController**, na pasta aula21, entrando com o comando:

```
php artisan make:controller aula22\usuariosController
```

app/Http/Controllers/aula22/usuariosController.php

```
<?php

namespace App\Http\Controllers\aula22;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class usuariosController extends Controller
{
    public function MostrarUsuarios(){

        $usuarios = [
            'Joao',
            'Maria',
            'Antonio'
        ];

        // $usuarios = [];

        // $usuarios = ['João'];

        // apresentar a view com os usuários

        return view('aula22/show_usuarios', compact('usuarios'));

    }
}
```

routes/web.php

```
Route::get('aula22/usuarios', 'aula21\usuariosController@MostrarUsuarios');
```

views/aula22/show_usuarios.blade.php

```
<div>

    <h1>Blade Template Engine</h1>

    {{-- Bloco de código em PHP puro --}}

    @php

        // Esse é um comentário de PHP puro
        echo '<p>Usuários inscritos na plataforma: ' . count($usuarios) . '</p>';

    @endphp

    {{-- Bloco de código condicional - if statement --}}

    @if(count($usuarios) === 0)
        <p>Não existem usuários inscritos na plataforma</p>
    @elseif(count($usuarios) === 1)
        <p>Há apenas um usuário inscrito na plataforma</p>
    @else
        <p>Há {{count($usuarios)}} usuários inscritos</p>
    @endif

    @unless(count($usuarios) === 0)
        <p>Ativou o código de unless</p>
    @endunless

</div>
```

unless significa “a não ser que”

unless → se não for



Aula 023 – Blade Snippets

No Sublime Text Instale o plugin “blade snippets”

routes/web.php

```
Route::get('aula23/usuarios', 'aula23\usuariosController@MostrarUsuarios');
```

views/aula23/show_usuarios.blade.php

```
<div>
```

```
    <h1>Blade Template Engine</h1>
```

```
    {{-- Bloco de código em PHP puro --}}
```

```
    @php
```

```
        // Esse é um comentário de PHP puro
```

```
        echo '<p>Usuários inscritos na plataforma: ' . count($usuarios) . '</p>';
```

```
    @endphp
```

```
    {{-- Bloco de código condicional - if statement --}}
```

```
    @if(count($usuarios) === 0)
```

```
        <p>Não existem usuários inscritos na plataforma</p>
```

```
    @elseif(count($usuarios) === 1)
```

```
        <p>Há apenas um usuário inscrito na plataforma</p>
```

```
    @else
```

```
        <p>Há {{count($usuarios)}} usuários inscritos</p>
```

```
    @endif
```

```
    @unless(count($usuarios) === 0)
```

```
        <p>Ativou o código de unless</p>
```

```
    @endunless
```

```
    @php
```

```
        $nome = "João";
```

```
    @endphp
```

```
    <!-- <p>{{ $nome }}</p> -->
```

```
    @isset($nome)
```

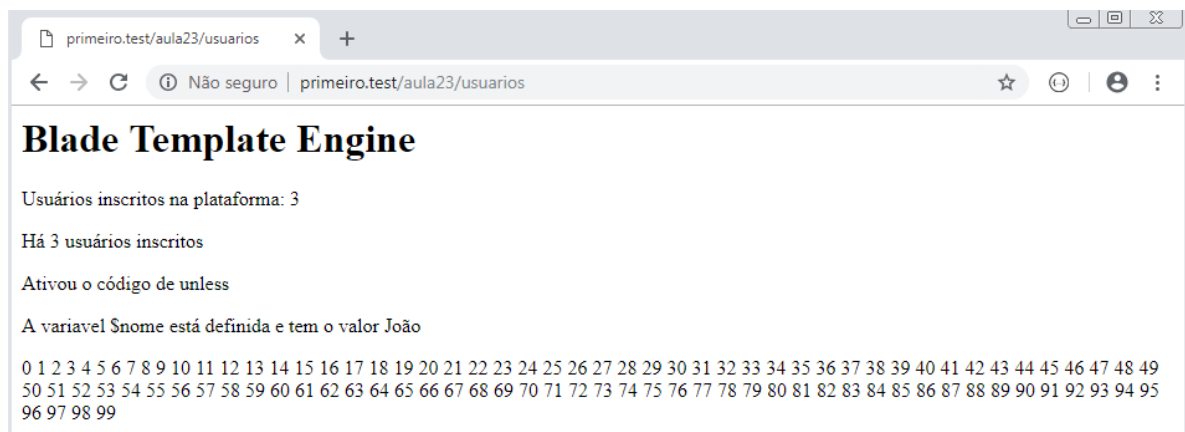
```
        <p>A variável $nome está definida e tem o valor {{ $nome }}</p>
```

```
    @endisset
```

```
@empty($nome)
    <p>A variavel $nome está definida e tem o valor {{$nome}}</p>
@endempty
```

```
@for($i = 0; $i < 100; $i++)
    {{$i}}
@endfor
```

</div>



Aula 024 – Blade Layouts

Dentro da pasta **views** → **aula24**, criar a pasta layouts. E dentro da pasta layouts criar o arquivo **base_layout.blade.php**

views/aula24/layouts/base_layout.blade.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Título da página</title>
</head>
<body>

    {{-- header --}}
    <div>
        @yield('cabecalho')
    </div>

    {{-- content --}}
    <div>
        @yield('conteudo')
    </div>

    {{-- footer --}}
    <div>
        @yield('rodape')
    </div>

</body>
</html>
```

routes/web.php

```
Route::get('aula24/contato', function(){
    return view('aula24/contato');
});

Route::get('aula24/about', function(){
    return view('aula24/about');
});
```

Dentro da pasta **views** → **aula24** criar dois arquivos: **contato.blade.php** e **about.blade.php**.

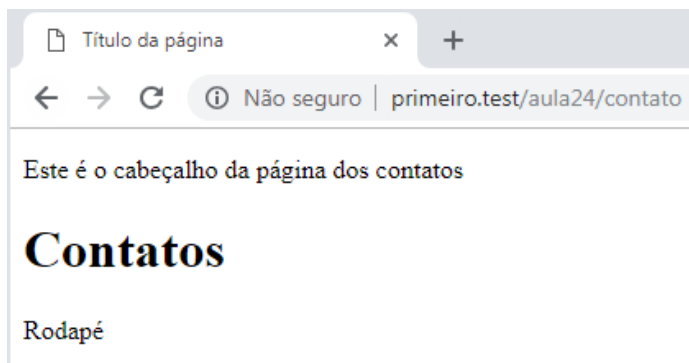
views/aula24/contato.blade.php

```
@extends('layouts.base_layout')

@section('cabecalho')
    <p>Este é o cabeçalho da página dos contatos</p>
@stop

@section('conteudo')
    <h1>Contatos</h1>
@stop

@section('rodape')
    <p>Rodapé</p>
@stop
```



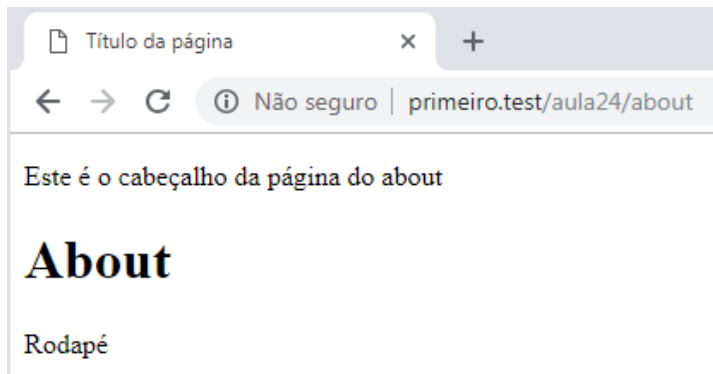
views/aula24/about.blade.php

```
@extends('layouts.base_layout')
```

```
@section('cabecalho')  
    <p>Este é o cabeçalho da página do about</p>  
@stop
```

```
@section('conteudo')  
    <h1>About</h1>  
@stop
```

```
@section('rodape')  
    <p>Rodapé</p>  
@stop
```



Aula 025 – Blade Layouts – Parte 2

- Dentro da pasta **views** → **aula25** → **layouts** criar o arquivo **app.blade.php**;

views/aula25/layouts/app.blade.php

```
<!DOCTYPE html>
<html>
<head>
    <title>@yield('titulo', 'Título default')</title>
</head>
<body>

{{-- navegacao --}}

@section('navigacao')
    <p>Conteúdo transversal</p>
@show

</body>
</html>
```

views/aula25/usuarios.blade.php

```
@extends('aula25.layouts.app')

{{-- titulo --}}

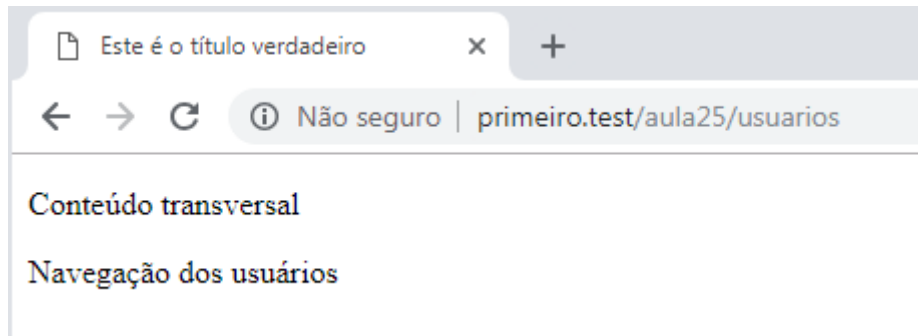
@section('titulo')
    {{ 'Este é o título verdadeiro' }}
@stop

{{-- navegação --}}

@section('navigacao')
    @parent
    <p>Navegação dos usuários</p>
@stop
```

routes/web.php

```
Route::get('aula25/usuarios', function(){  
    return view('aula25/usuarios');  
});
```



Aula 026 – Blade Layouts – Parte 3

Usando o blade para exibir mensagens de erro

titulo

texto da mensagem

titulo – fundo cinza

mensagem – texto cinza com fundo branco

routes/web.php

```
Route::get('aula26/teste', function(){  
    return view('aula26/teste');  
});
```

- Na pasta **view** → **aula26** criar dois arquivos: **teste.blade.php** e **comp_mensagem.blade.php**:

views/aula26/teste.blade.php

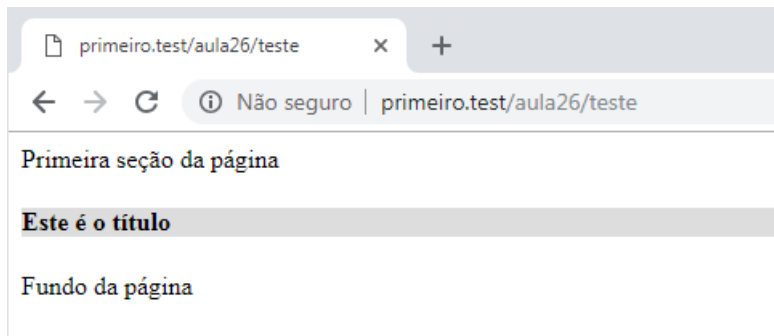
```
<div>  
    <p>Primeira seção da página</p>  
</div>
```

```
@component('aula26.comp_mensagem')  
    {{-- titulo --}}  
    Este é o título  
@endcomponent
```

```
<div>  
    <p>Fundo da página</p>  
</div>
```

views/aula26/comp_mensagem.blade.php

```
{{-- titulo --}}  
  
<div style="background-color: #ddd">  
    <h4>{{ $slot }}</h4>  
</div>
```



views/aula26/comp2_mensagem.blade.php

```
<div style="background-color: #ddd">

    <h4>{{$titulo}}</h4>

</div>

<div style="color: #ddd">

    <p>{{$mensagem}}</p>

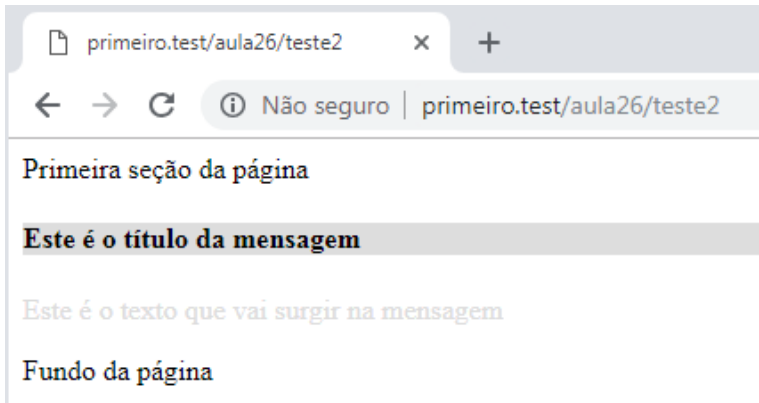
</div>
```

views/aula26/teste2.blade.php

```
<div>
    <p>Primeira seção da página</p>
</div>

@component('comp_mensagem')
    @slot('titulo')
        Este é o título da mensagem
    @endslot
    @slot('mensagem')
        Este é o texto que vai surgir na mensagem
    @endslot
@endcomponent

<div>
    <p>Fundo da página</p>
</div>
```



Aula 027 – Blade include sub-views

routes/web.php

```
Route::get('show', function(){  
    return view('apresentacao');  
});
```

- Dentro da pasta **views** → **aula27** criar dois arquivos:

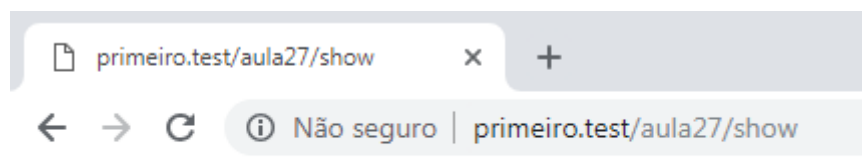
apresentacao.blade.php
inc/dados.php

views/aula27/apresentacao.blade.php

```
<h1>Include</h1>  
  
<div>Este é o texto da primeira seção</div>  
  
{{-- include --}}  
@include('inc.dados')  
  
<div>Este é o texto da última seção</div>
```

views/aula27/dados.php

```
<p>Texto dentro de dados.</p>
```



Include

Este é o texto da primeira seção

Texto dentro de dados.

Este é o texto da última seção

routes/web.php

```
Route::get('aula27/show2', function(){  
    return view('aula27/apresentacao2');  
});
```

views/aula27/apresentacao2.blade.php

```
<h1>Include</h1>
```

```
<div>Este é o texto da primeira seção</div>
```

```
@php
```

```
    $nomes = [  
        'Joao',  
        'Manuel',  
        'Maria'  
    ];
```

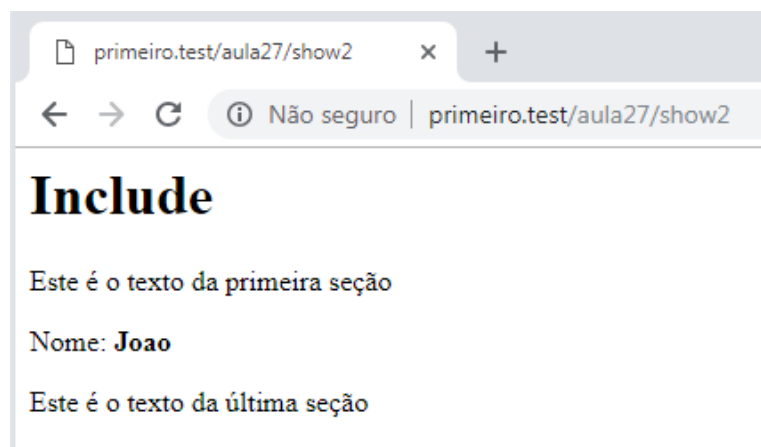
```
@endphp
```

```
@include('aula27.inc.linha_dados2')
```

```
<div>Este é o texto da última seção</div>
```

views/aula27/inc/linha_dados2.blade.php

```
<p>Nome: <strong> {{ $nomes[0] }} </strong></p>
```



routes/web.php

```
Route::get('aula27/show3', function(){
    return view('aula27/apresentacao3');
});
```

views/aula27/apresentacao3.blade.php

<h1>Include</h1>

<div>Este é o texto da primeira seção</div>

```
@php
    $nomes = [
        'Joao',
        'Manuel',
        'Maria'
    ];
@endphp
```

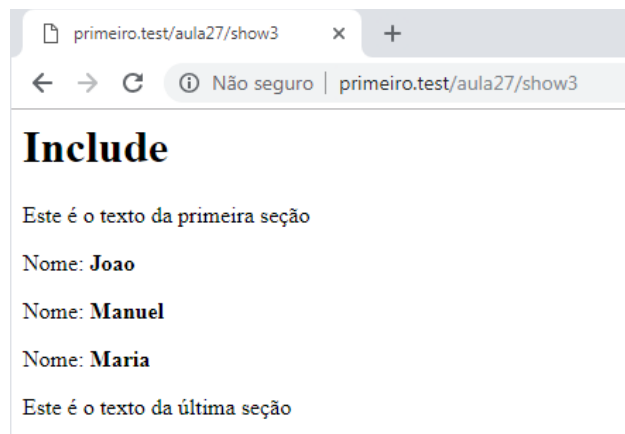
{{-- ciclo de apresentação dos dados --}}

```
@foreach ($nomes as $nome)
    @includeIf('aula27.inc.linha_dados3')
@endforeach
```

<div>Este é o texto da última seção</div>

views/aula27/inc/linha_dados3.blade.php

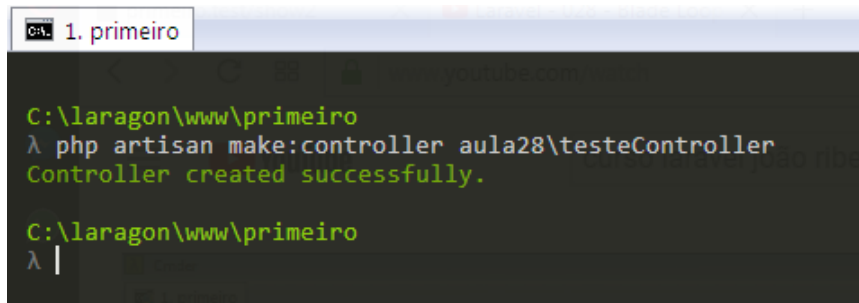
<p>Nome: {{ \$nome }} </p>



Aula 028 – Blade Loops

- No terminal, entre com o seguinte comando:

```
php artisan make:controller aula28\testeController
```



```
C:\laragon\www\primeiro
λ php artisan make:controller aula28\testeController
Controller created successfully.

C:\laragon\www\primeiro
λ |
```

routes/web.php

```
Route::get('aula28/teste', 'aula28\testeController@Executar');
```

Loop For

app/http/Controllers/aula28/testeController.php

```
<?php
```

```
namespace App\Http\Controllers\aula28;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

```
class testeController extends Controller
```

```
{
```

```
    public function executar(){
```

```
        // variáveis
```

```
        // view
```

```
        return view('aula28/teste');
```

```
    }
```

```
}
```

views/aula28/teste.blade.php

```
<h1>TESTES DE LOOPS</h1>
```

```
{{-- for loop --}}
```

```
@for ($m = 0; $m <= 10; $m++)
```

```
<p>O valor de $m é {{$m}}</p>
```

```
@endfor
```



Foreach

routes/web.php

```
Route::get('aula28/teste2', 'aula28\testeController@Executar2');
```

app/http/Controllers/aula28/testeController.php

```
<?php
```

```
namespace App\Http\Controllers\aula28;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

```
class testeController extends Controller  
{
```

```
    public function executar(){  
        // variáveis
```

```
        // view  
        return view('aula28/teste');  
    }
```

```
    public function executar2(){
```

```
        // variáveis  
        $max = 10;
```

```
        // nomes  
        $nomes = [  
            'Joao',  
            'Ana',  
            'Manuel',  
            'Francisco',  
            'Maria'  
        ];
```

```
        // view  
        return view('aula28/teste2', compact('max', 'nomes'));  
    }
```

views/aula28/teste2.blade.php

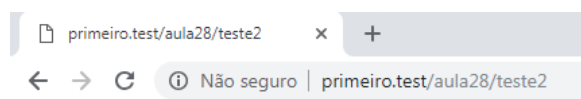
```
<h1>TESTES DE LOOPS</h1>
```

```
{{-- for loop --}}
```

```
@for ($m = 0; $m <= $max; $m++)  
    <p>O valor de $m é {{ $m }}</p>  
@endfor
```

```
{{-- foreach --}}
```

```
@foreach ($nomes as $nome)  
    @if ($loop->first)  
        <p style="color: red;">O nome é {{ $nome }}</p>  
    @elseif($loop->last)  
        <p style="color: blue;">O nome é {{ $nome }}</p>  
    @else  
        <p>O nome é {{ $nome }}</p>  
    @endif  
@endforeach
```



TESTES DE LOOPS

O valor de \$m é 0

O valor de \$m é 1

O valor de \$m é 2

O valor de \$m é 3

O valor de \$m é 4

O valor de \$m é 5

O valor de \$m é 6

O valor de \$m é 7

O valor de \$m é 8

O valor de \$m é 9

O valor de \$m é 10

O nome é Joao

O nome é Ana

O nome é Manuel

O nome é Francisco

O nome é Maria

Aula 029 – Blade Views para coleções

routes/web.php

```
Route::get('aula29/teste', 'aula29\testeController@Executar');
```

app/Http/Controllers/aula29/testeController.php

```
<?php

namespace App\Http\Controllers\aula29;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class testeController extends Controller
{

    public function executar(){

        // nomes
        $nomes = [
            'Joao',
            'Ana',
            'Manuel',
            'Francisco',
            'Maria'
        ];

        // view
        return view('aula29/teste', compact('nomes'));
    }

}
```

views/aula29/teste.blade.php

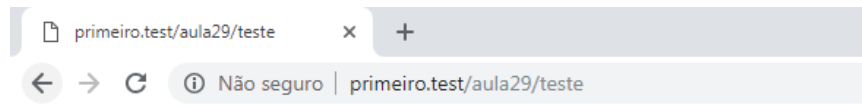
```
<h1>TESTES DE VIEWS PARA COLEÇÕES</h1>
```

```
@each ('aula29.linha', $nomes, 'nome', 'aula29.vazio')
```


- Na pasta **Views** criar dois arquivos: **linha.blade.php** e **vazio.blade.php**

views/aula29/linha.blade.php

```
<li>Nome: {{$nome}}</li>
```



TESTES DE VIEWS PARA COLEÇÕES

- Nome: Joao
 - Nome: Ana
 - Nome: Manuel
 - Nome: Francisco
 - Nome: Maria
-

app/Http/Controllers/aula29/testeController.php

```
<?php
```

```
namespace App\Http\Controllers\aula29;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

```
class testeController extends Controller  
{
```

```
    public function executar(){
```

```
        // nomes
```

```
        // $nomes = [
```

```
        // 'Joao',
```

```
        // 'Ana',
```

```
        // 'Manuel',
```

```
        // 'Francisco',
```

```
        // 'Maria'
```

```
        // ];
```

```
        $nomes = [];
```

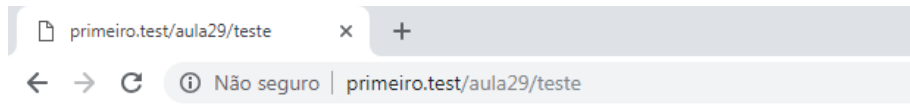
```
        return view('teste3', compact( 'nomes'));
```

```
    }
```

```
}
```

views/aula29/vazio.blade.php

<p>Ops! Não foram encontrados registros.</p>



TESTES DE VIEWS PARA COLEÇÕES

Ops! Não foram encontrados registros.

Aula 030 – Passando dados de formulário para Controller

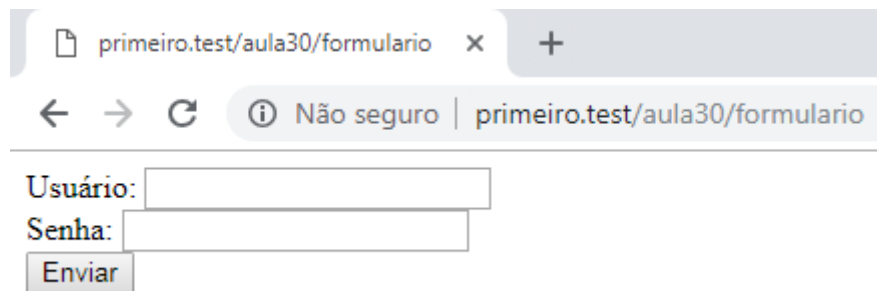
routes/web.php

```
Route::get('aula30/formulario', function(){  
    return view('aula30/formulario');  
});
```

- Na pasta Views → aula30, crie um arquivo chamado formulario.blade.php

views/aula30/formulario.blade.php

```
<form method="POST" action="" ">  
    {{csrf_field()}}  
  
    Usuário:  
    <input type="text" name="text_usuario">  
    <br />  
    Senha:  
    <input type="password" name="text_senha">  
    <br />  
    <input type="submit" value="Enviar">  
</form>
```



primeiro.test/aula30/formulario

Usuário:

Senha:

Enviar

No terminal criar, na pasta aula30, o controller “**usuariosController**”

```
C:\laragon\www\primeiro
λ php artisan make:controller aula30\usuariosController
Controller created successfully.

C:\laragon\www\primeiro
λ |
```

app/http/Controllers/aula30/usuariosController.php

```
<?php
```

```
namespace App\Http\Controllers\aula30;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

```
class usuariosController extends Controller
```

```
{
```

```
    // Apresentar o formulário
```

```
    public function show(){
```

```
        return view('aula30/formulario');
```

```
    }
```

```
    // Tratar os dados do formulário
```

```
    public function tratar(Request $dados){
```

```
        return 'Cheguei!';
```

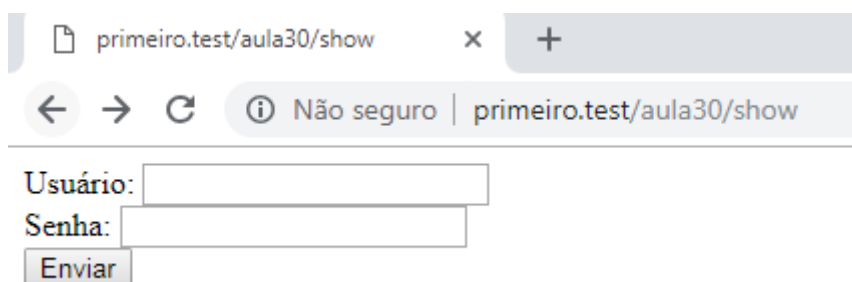
```
    }
```

```
}
```

routes/web.php

```
Route::get('aula30/show', 'aula30\usuariosController@show');
```

```
Route::post('aula30/tratar', 'aula30\usuariosController@tratar');
```



primeiro.test/aula30/show

Não seguro | primeiro.test/aula30/show

Usuário:

Senha:

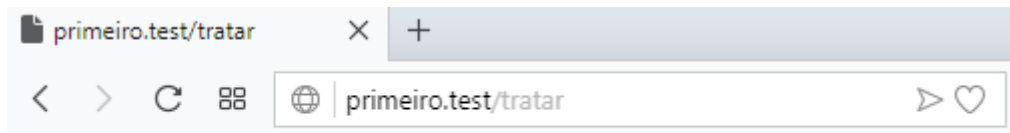
Enviar

views/aula30/formulario.blade.php

```
<form method="POST" action="tratar">
    {{csrf_field()}}

    Usuário:
    <input type="text" name="text_usuario">
    <br />
    Senha:
    <input type="password" name="text_senha">
    <br />
    <input type="submit" value="Enviar">
</form>
```

- Clique no botão “Enviar”

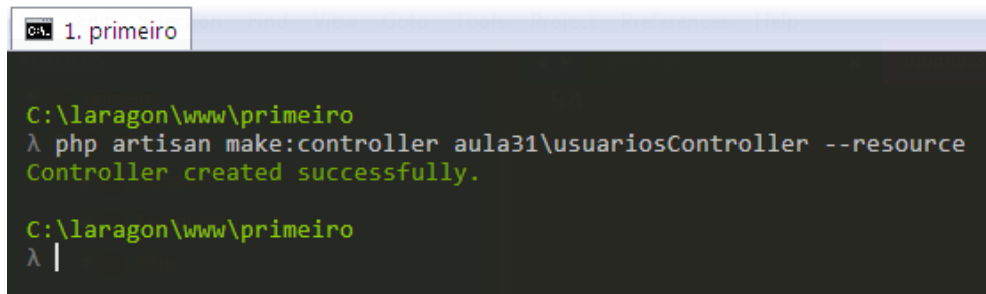


Cheguei!

Aula 031 – Resource Controller e dados de formulário

No terminal criar, na pasta **aula31**, o controller “**usuariosController**” do tipo resource:

```
php artisan make:controller aula31\usuariosController --resource
```



The screenshot shows a terminal window with a title bar that says "1. primeiro". The terminal content is as follows:

```
C:\laragon\www\primeiro
λ php artisan make:controller aula31\usuariosController --resource
Controller created successfully.

C:\laragon\www\primeiro
λ |
```

app/Http/Controllers/aula31/usuariosController.php

```
<?php
```

```
namespace App\Http\Controllers\aula31;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

```
class usuariosController extends Controller
```

```
{
```

```
    public function index()
```

```
    {
```

```
        // Apresenta o formulário
```

```
        return view('formulario');
```

```
    }
```

```
    public function create()
```

```
    {
```

```
        //
```

```
    }
```

```
    public function store(Request $request)
```

```
    {
```

```
        $usuario = $request->text_usuario;
```

```
        $senha = $request->text_senha;
```

```
        $token = $request->_token;
```

```
        $senha_enc = md5($senha);
```

```
        echo "Usuário: $usuario <br />Senha: $senha <br />Senha encriptada: $senha_enc";
```

```
        // return 'Cheguei';
```

```
    }
```

```
    public function show($id)
```

```
    {
```

```
        //
```

```
    }
```

```

public function edit($id)
{
    //
}
public function update(Request $request, $id)
{
    //
}
public function destroy($id)
{
    //
}
}

```

POST	usuarios	usuarios.store	App\Http\Controllers\aula31\usuariosController@store
GET HEAD	usuarios	usuarios.index	App\Http\Controllers\aula31\usuariosController@index
GET HEAD	usuarios/create	usuarios.create	App\Http\Controllers\aula31\usuariosController@create
DELETE	usuarios/{usuario}	usuarios.destroy	App\Http\Controllers\aula31\usuariosController@destroy
PUT PATCH	usuarios/{usuario}	usuarios.update	App\Http\Controllers\aula31\usuariosController@update
GET HEAD	usuarios/{usuario}	usuarios.show	App\Http\Controllers\aula31\usuariosController@show
GET HEAD	usuarios/{usuario}/edit	usuarios.edit	App\Http\Controllers\aula31\usuariosController@edit

views/aula31/formulario.blade.php

```

<form method="POST" action="usuarios">
    {{csrf_field()}}
    Usuário:
    <input type="text" name="text_usuario">
    <br />
    Senha:
    <input type="password" name="text_senha">
    <br />
    <input type="submit" value="Enviar">
</form>

```

routes/web.php

```
Route::resource('aula31/usuarios', 'aula31\usuariosController');
```

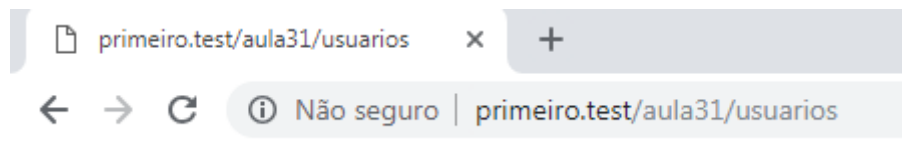
primeiro.test/aula31/usuarios

Não seguro | primeiro.test/aula31/usuarios

Usuário: João

Senha:

Enviar



Usuário: João

Senha: 123456

Senha encriptada: e10adc3949ba59abbe56e057f20f883e

Aula 032 – Bases de dados - Configurações

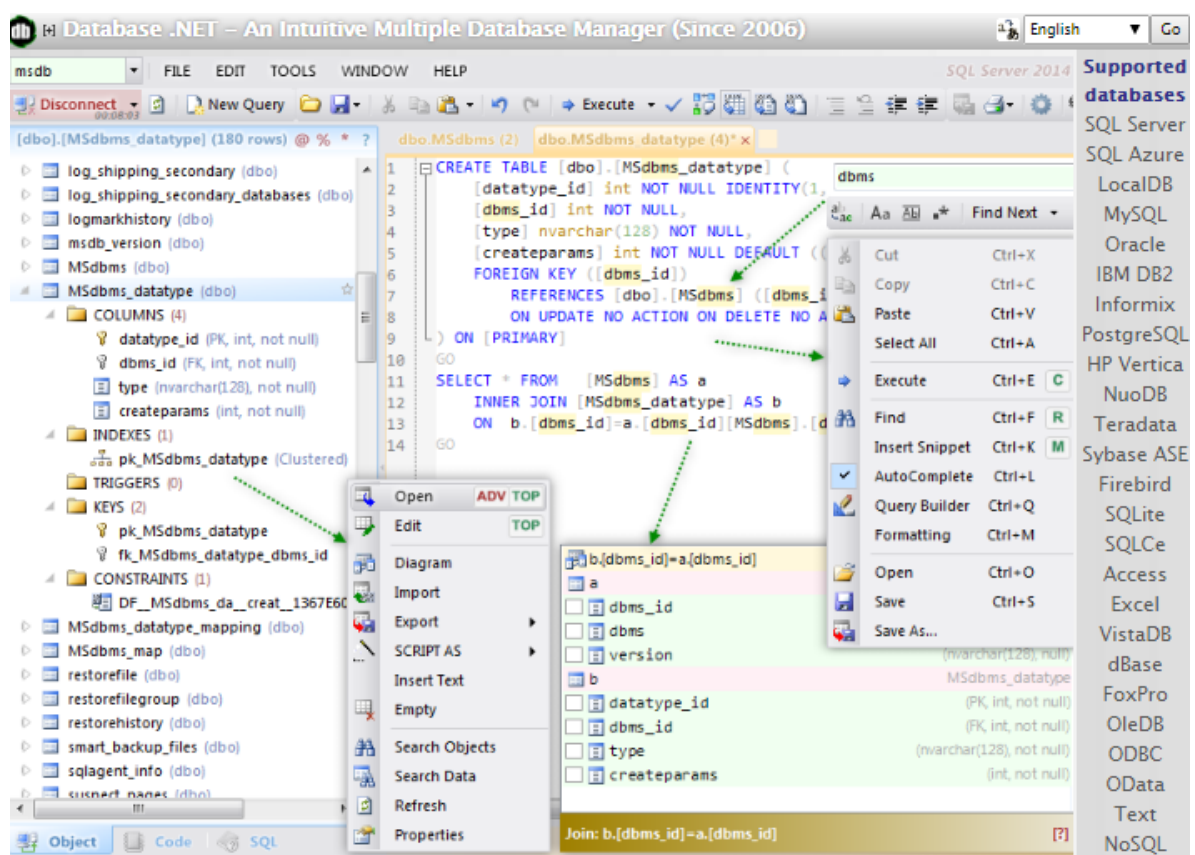
Acessar:

<https://fishcode.lib.com/database.htm>

E baixar database .NET

É um arquivo único. Inserir-lo em uma pasta chamada: C:\database.net e criar um atalho na área de trabalho do Windows.

Para iniciar o aplicativo basta clicar sobre o arquivo.



Database .NET is an innovative, powerful and intuitive multiple database management tool

You can browse/grant objects, design tables, edit rows, run queries, generate scripts, analyze/monitor SQL and import/export/migrate/sync data with a consistent interface.

(32bit) Free and Plus Edition 26.8⁶⁹⁸¹ (2019/02/11), 17.2MB

(64bit) Standard and Professional Edition 26.8⁶⁹⁸¹ (2019/02/11), 18.5MB

Comparison / History / Features / Support / Feedback / License

- Criar um banco de dados chamado “primeiro”.

- Abrir o arquivo .env e configurar a conexão com banco de dados.

```


1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:Ej8Ji0YQ3a4b4oqTD44CiMBIjDmbp0cbouIGLHeLfJI=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=localhost
11 DB_PORT=3306
12 DB_DATABASE=primeiro
13 DB_USERNAME=root
14 DB_PASSWORD=
15
16 BROADCAST_DRIVER=log
17 CACHE_DRIVER=file
18 QUEUE_CONNECTION=sync
19 SESSION_DRIVER=file
20 SESSION_LIFETIME=120
21
22 REDIS_HOST=127.0.0.1
23 REDIS_PASSWORD=null
24 REDIS_PORT=6379
25
26 MAIL_DRIVER=smtp
27 MAIL_HOST=smtp.mailtrap.io

```

Instalando o phpmyadmin dentro do Laragon

Baixar o aplicativo phpmyadmin, na versão zipada (phpMyAdmin-4.8.5-english.zip), em: <https://www.phpmyadmin.net/downloads/>

[Home](#)
[News](#)
[Security](#)
[Support](#)
[Docs](#)
[Try](#)
[Contribute](#)
[Sponsors](#)
[Themes](#)
[Download](#)
[GitHub](#)



Bringing MySQL to the web

[Download 4.8.5](#)
[Try demo](#)
[Donate](#)

Download

Many operating systems already include a phpMyAdmin package and will automatically keep it updated, however these versions are sometimes slightly outdated and therefore may be missing the latest features. Additionally, the configuration process varies widely by package and may not adhere to the official phpMyAdmin documentation. That being said, it is usually the quickest and easiest way of keeping an updated installation. Please contact your OS vendor for more information. Some additional information is also available in our [documentation](#). If you just want to try phpMyAdmin in a virtual machine, you might want to check the [available software appliances which provide phpMyAdmin](#).

If you do not find a suitable package or wish to install your own phpMyAdmin, you can download one of the following kits. Please note that each version has requirements for the minimum PHP and MySQL versions.

Since July 2015 all phpMyAdmin releases are cryptographically signed by the releasing developer. You should verify that the signature matches the archive you have downloaded. Verification instructions are placed in our documentation in the [Verifying phpMyAdmin releases](#) chapter.

phpMyAdmin 4.8.5

Released 2019-01-26, see [release notes](#) for details.

Current version compatible with PHP 5.5 to 7.2 and MySQL 5.5 and newer.

File	Size	Verification
phpMyAdmin-4.8.5-all-languages.zip	10.3 MB	[PGP] [SHA256]
phpMyAdmin-4.8.5-all-languages.tar.gz	9.3 MB	[PGP] [SHA256]
phpMyAdmin-4.8.5-all-languages.tar.xz	5.7 MB	[PGP] [SHA256]
phpMyAdmin-4.8.5-english.tar.gz	5.0 MB	[PGP] [SHA256]
phpMyAdmin-4.8.5-english.tar.xz	3.9 MB	[PGP] [SHA256]
phpMyAdmin-4.8.5-english.zip	6.1 MB	[PGP] [SHA256]

- Descompactar dentro da pasta C:\laragon\etc\apps

Nome	Data de modificaç...	Tipo	Tamanho
adminer	08/02/2019 01:14	Pasta de arquivos	
laragon	08/02/2019 01:14	Pasta de arquivos	
memcached	08/02/2019 01:14	Pasta de arquivos	
phpMyAdmin-4.8.5-english	26/01/2019 01:05	Pasta de arquivos	
phpRedisAdmin	08/02/2019 01:14	Pasta de arquivos	

- Alterar o nome da pasta para: phpmyadmin:

Nome	Data de modificaç...	Tipo	Tamanho
adminer	08/02/2019 01:14	Pasta de arquivos	
laragon	08/02/2019 01:14	Pasta de arquivos	
memcached	08/02/2019 01:14	Pasta de arquivos	
phpmyadmin	26/01/2019 01:05	Pasta de arquivos	
phpRedisAdmin	08/02/2019 01:14	Pasta de arquivos	

No Laragon, no botão “Banco de dados” clicar em <http://localhost/phpmyadmin/>:



Welcome to phpMyAdmin

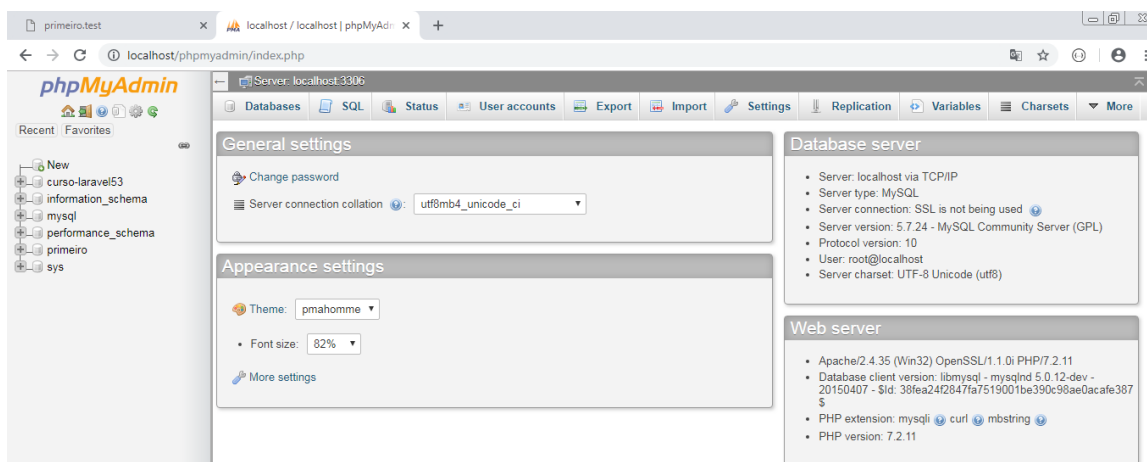
Log in

Username:

Password:

Go

Entrar com usuário: root e senha vazia.



The screenshot shows the phpMyAdmin web interface in a browser. The left sidebar displays a tree view of databases: 'curso-laravel53', 'information_schema', 'mysql', 'performance_schema', 'primeiro', and 'sys'. The main content area is divided into several panels. The 'General settings' panel includes a 'Change password' link and a 'Server connection collation' dropdown set to 'utf8mb4_unicode_ci'. The 'Appearance settings' panel shows a 'Theme' dropdown set to 'pmahomme' and a 'Font size' dropdown set to '82%'. The 'Database server' panel lists server details: 'Server: localhost via TCP/IP', 'Server type: MySQL', 'Server connection: SSL is not being used', 'Server version: 5.7.24 - MySQL Community Server (GPL)', 'Protocol version: 10', 'User: root@localhost', and 'Server charset: UTF-8 Unicode (utf8)'. The 'Web server' panel lists: 'Apache/2.4.35 (Win32) OpenSSL/1.1.0i PHP/7.2.11', 'Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 38fea24f2847fa7519001be390c98ae0acafe387\$', 'PHP extension: mysqli, curl, mbstring', and 'PHP version: 7.2.11'.

- Clicar na aba “user accounts” e criar um novo usuário com todos os privilégios.

Aula 033 – Introdução a Migrations

- Na pasta **Database** → **Migrations** o Laravel cria automaticamente dois arquivos para registro de usuários:

- 2014_10_12_000000_create_users_table.php
- 2014_10_12_100000_create_password_resets_table.php

Database/Migrations/2014_10_12_000000_create_users_table.php

<?php

```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateUsersTable extends Migration
{
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```

A função up é usada para criar a estrutura da tabela e função down é usada para apagá-la (faz um rollback).

A função timestamps() cria os campos created_at e updated_at.

Atualizando uma tabela

Para que atualizações feitas em uma tabela sejam refletidas no banco de dados, entre, no terminal, com o seguinte comando:

```
php artisan migrate:refresh
```

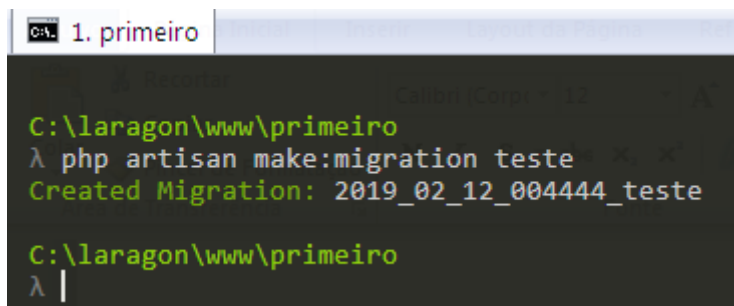
Aula 034 – Criação de Migration

Para criar uma migration, no terminal entre com o seguinte comando:

```
php artisan make:migration <nome_migration>
```

Exemplo:

```
php artisan make:migration teste
```



```
1. primeiro  
C:\laragon\www\primeiro  
λ php artisan make:migration teste  
Created Migration: 2019_02_12_004444_teste  
C:\laragon\www\primeiro  
λ |
```

database/migrations/2019_02_12_004444_teste.php

```
<?php
```

```
use Illuminate\Support\Facades\Schema;  
use Illuminate\Database\Schema\Blueprint;  
use Illuminate\Database\Migrations\Migration;
```

```
class Teste extends Migration
```

```
{  
    public function up()  
    {  
        //  
    }
```

```
    public function down()  
    {  
        //  
    }  
}
```

Criando uma migration e dando o nome para a tabela

Apague a migration criada anteriormente e crie uma nova migration entrando com o comando:

```
php artisan make:migration usuarios --create=usuarios
```

database/migrations/ 2019_02_12_005501_usuarios.php

```
<?php
```

```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class Usuarios extends Migration
{
    public function up()
    {
        Schema::create('usuarios', function (Blueprint $table) {
            $table->increments('id');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('usuarios');
    }
}
```

- Insira os seguintes campos na tabela:

database/migrations/2019_02_12_005501_usuarios.php

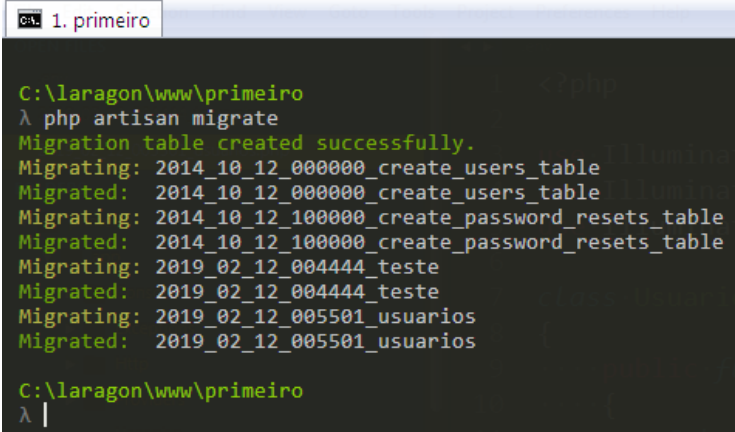
```
<?php
```

```
use Illuminate\Support\Facades\Schema;  
use Illuminate\Database\Schema\Blueprint;  
use Illuminate\Database\Migrations\Migration;
```

```
class Usuarios extends Migration  
{  
    public function up()  
    {  
        Schema::create('usuarios', function (Blueprint $table) {  
            $table->increments('id');  
            $table->string('usuario');  
            $table->string('senha');  
            $table->string('email');  
            $table->timestamps();  
        });  
    }  
  
    public function down()  
    {  
        Schema::dropIfExists('usuarios');  
    }  
}
```

Para criar a tabela, entre no terminal com o seguinte comando:

php artisan migrate

A screenshot of a terminal window with a dark background. The title bar shows '1. primeiro'. The terminal text shows the command 'php artisan migrate' being executed in the directory 'C:\laragon\www\primeiro'. The output shows several migrations being applied successfully, including '2014_10_12_000000_create_users_table', '2014_10_12_100000_create_password_resets_table', '2019_02_12_004444_teste', and finally '2019_02_12_005501_usuarios'.

```
C:\laragon\www\primeiro  
λ php artisan migrate  
Migration table created successfully.  
Migrating: 2014_10_12_000000_create_users_table  
Migrated: 2014_10_12_000000_create_users_table  
Migrating: 2014_10_12_100000_create_password_resets_table  
Migrated: 2014_10_12_100000_create_password_resets_table  
Migrating: 2019_02_12_004444_teste  
Migrated: 2019_02_12_004444_teste  
Migrating: 2019_02_12_005501_usuarios  
Migrated: 2019_02_12_005501_usuarios  
  
C:\laragon\www\primeiro  
λ |
```


No Laravel, por convenção, os nomes das migrations devem ser criados como:

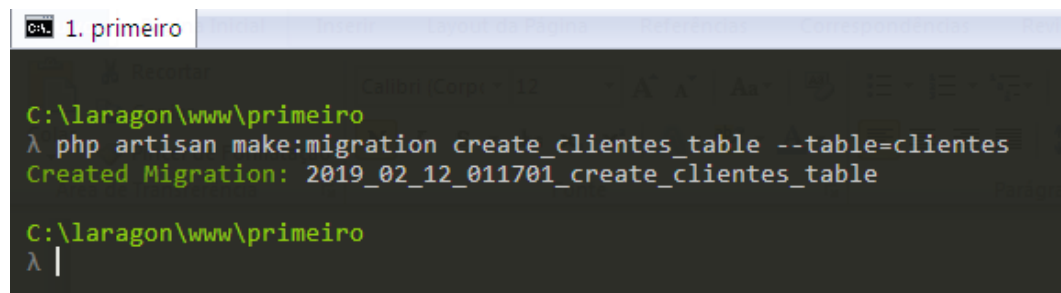
`create_<nome_migration>_table`

Exemplo:

`create_clientes_table`

- Entre no terminal e crie a migration com o seguinte comando:

`php artisan make:migration create_clientes_table --table=clientes`

A screenshot of a terminal window with a dark background. The title bar shows 'C:\> 1. primeiro'. The terminal content shows the command 'php artisan make:migration create_clientes_table --table=clientes' being executed in the directory 'C:\laragon\www\primeiro'. The output is 'Created Migration: 2019_02_12_011701_create_clientes_table'. The prompt 'λ' is visible at the bottom.

```
C:\laragon\www\primeiro
λ php artisan make:migration create_clientes_table --table=clientes
Created Migration: 2019_02_12_011701_create_clientes_table

C:\laragon\www\primeiro
λ |
```

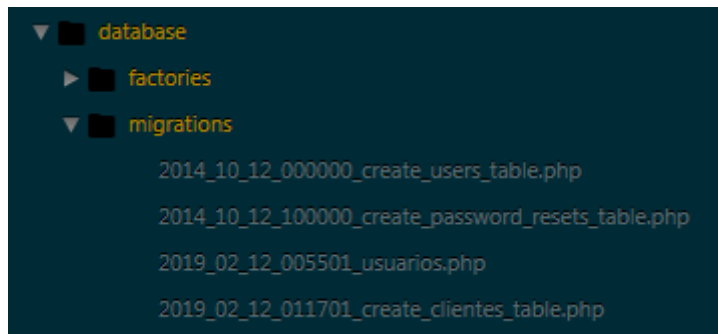
database/migrations/2019_02_12_011701_create_clientes_table.php

`<?php`

```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class CreateClientesTable extends Migration
{
    public function up()
    {
        Schema::table('clientes', function (Blueprint $table) {
            //
        });
    }

    public function down()
    {
        Schema::table('clientes', function (Blueprint $table) {
            //
        });
    }
}
```

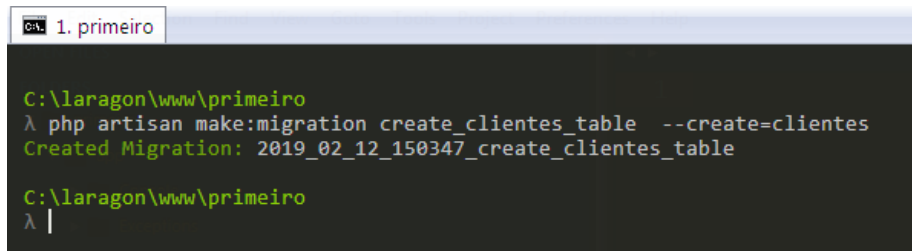


- Apague as duas tabelas criadas: de usuários e de clientes

Aula 036 - Migrations – Criações de tabelas com Blueprint

- No terminal crie uma migration com o comando:

```
php artisan make:migration create_clientes_table --create=clientes
```



```
C:\laragon\www\primeiro
λ php artisan make:migration create_clientes_table --create=clientes
Created Migration: 2019_02_12_150347_create_clientes_table

C:\laragon\www\primeiro
λ |
```

database/migrations/2019_02_12_144606_create_testes_table.php

```
<?php
```

```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class CreateClientesTable extends Migration
{
    public function up()
    {
        Schema::create('clientes', function (Blueprint $table) {
            $table->increments('id_cliente');
            $table->string('nome_cliente');
            $table->string('endereco',50);
            $table->string('telefone',30);
            $table->boolean('ativo');
            $table->dateTime('data_nascimento');
            $table->text('observacoes');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('clientes');
    }
}
```

No terminal, crie a tabela com o seguinte comando:

php artisan migrate

```
1. primeiro

C:\laragon\www\primeiro
λ php artisan migrate
Migrating: 2019_02_12_150347_create_clientes_table
Migrated: 2019_02_12_150347_create_clientes_table

C:\laragon\www\primeiro
λ
```

| | id_cliente
<small>INT UNSIGNED</small> | nome_cliente
<small>VARCHAR(255)</small> | endereco
<small>VARCHAR(50)</small> | telefone
<small>VARCHAR(30)</small> | ativo
<small>TINYINT</small> | data_nascimento
<small>DATETIME</small> | observacoes
<small>TEXT</small> | created_at
<small>TIMESTAMP</small> | updated_at
<small>TIMESTAMP</small> |
|---|-------------------------------------------|---------------------------------------------|----------------------------------------|----------------------------------------|---------------------------------|--------------------------------------------|------------------------------------|----------------------------------------|----------------------------------------|
| 1 | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

TABLES (3)

clientes

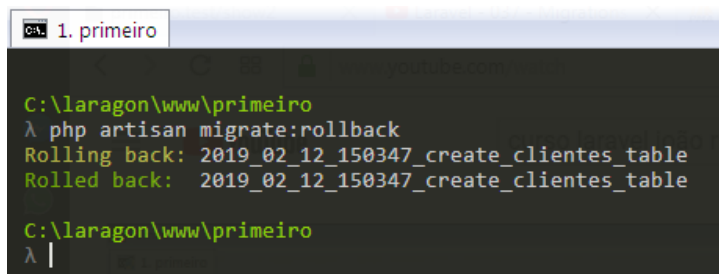
COLUMNS (9)

- id_cliente (PK, INT UNSIGNED, not null)
- nome_cliente (VARCHAR(255), not null)
- endereco (VARCHAR(50), not null)
- telefone (VARCHAR(30), not null)
- ativo (TINYINT, not null)
- data_nascimento (DATETIME, not null)
- observacoes (TEXT, not null)
- created_at (TIMESTAMP, null)
- updated_at (TIMESTAMP, null)

Aula 037 – Migrations – Criação de relação entre tabelas

No terminal, faça um rollback da tabela criada anteriormente (tabela clientes), com o seguinte comando:

```
php artisan migrate:rollback
```

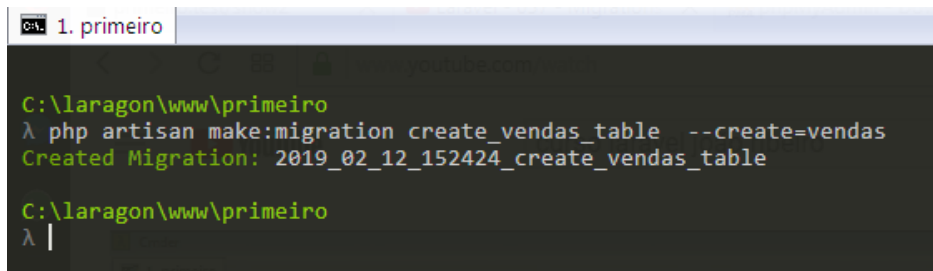


```
C:\laragon\www\primeiro
λ php artisan migrate:rollback
Rolling back: 2019_02_12_150347_create_clientes_table
Rolled back: 2019_02_12_150347_create_clientes_table

C:\laragon\www\primeiro
λ |
```

No terminal, crie uma migration chamada “**create_vendas_table**” com o seguinte comando:

```
php artisan make:migration create_vendas_table --create=vendas
```



```
C:\laragon\www\primeiro
λ php artisan make:migration create_vendas_table --create=vendas
Created Migration: 2019_02_12_152424_create_vendas_table

C:\laragon\www\primeiro
λ |
```

database/migrations/2019_02_12_152424_create_vendas_table.php

```
<?php
```

```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateVendasTable extends Migration
{
    public function up()
    {
        Schema::create('vendas', function (Blueprint $table) {
            $table->bigIncrements('id_venda');

            // relação
            $table->integer('id_cliente')->unsigned();

            $table->string('produto');
            $table->integer('quantia');
            $table->timestamps();

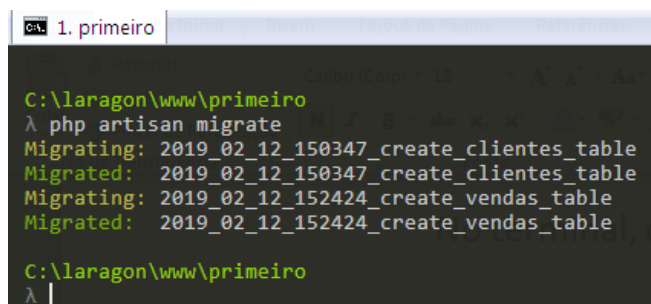
            // criar a relação
            $table->foreign('id_cliente')->references('id_cliente')->on('clientes');

        });
    }

    public function down()
    {
        Schema::dropIfExists('vendas');
    }
}
```

No terminal, crie as tabelas clientes e vendas com o seguinte comando:

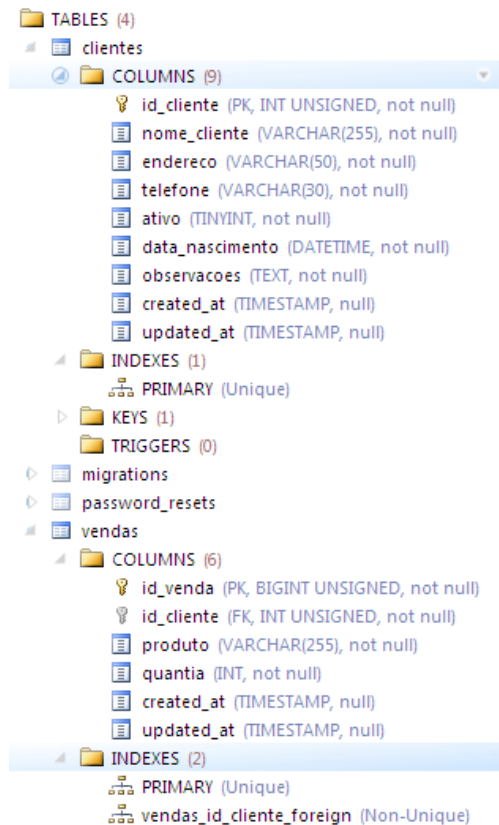
php artisan migrate



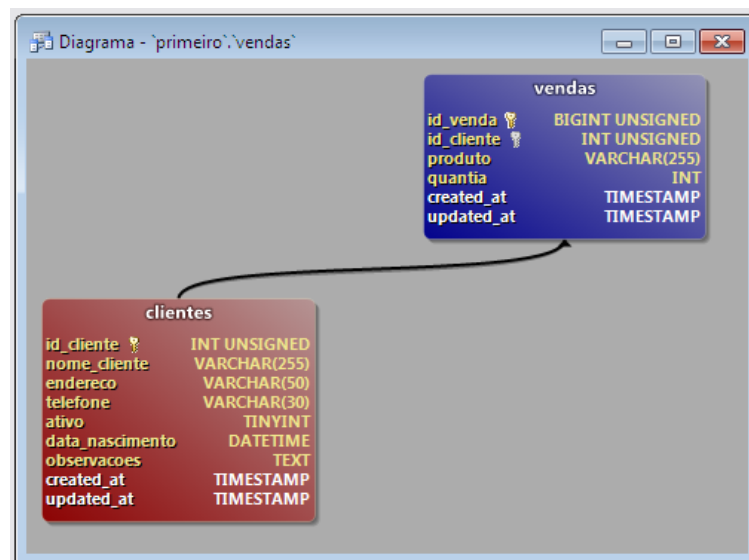
```
1. primeiro

C:\laragon\www\primeiro
λ php artisan migrate
Migrating: 2019_02_12_150347_create_clientes_table
Migrated: 2019_02_12_150347_create_clientes_table
Migrating: 2019_02_12_152424_create_vendas_table
Migrated: 2019_02_12_152424_create_vendas_table

C:\laragon\www\primeiro
λ |
```



Clique com o botão direito sobre a tabela “vendas” e no menu de contexto que irá se abrir, selecione a opção “diagrama”:



É uma relação de “um para muitos” (um cliente pode fazer várias vendas).

Na migration da tabela de vendas faça a seguinte alteração:

```
// criar a relação
$table->foreign('id_cliente')->references('id_cliente')->on('clientes')->onDelete('cascade');
```

Dessa forma, se um cliente for eliminado, todas as suas vendas também serão.

E para atualizar a tabela, entre no terminal com o seguinte comando:

php artisan migrate:refresh

```
1. primeiro

C:\laragon\www\primeiro
λ php artisan migrate:refresh
Rolling back: 2019_02_12_152424_create_vendas_table
Rolled back: 2019_02_12_152424_create_vendas_table
Rolling back: 2019_02_12_150347_create_clientes_table
Rolled back: 2019_02_12_150347_create_clientes_table
Migration not found: 2019_02_12_005501_usuarios
Migration not found: 2019_02_12_004444_teste
Rolling back: 2014_10_12_100000_create_password_resets_table
Rolled back: 2014_10_12_100000_create_password_resets_table
Rolling back: 2014_10_12_000000_create_users_table
Rolled back: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2019_02_12_150347_create_clientes_table
Migrated: 2019_02_12_150347_create_clientes_table
Migrating: 2019_02_12_152424_create_vendas_table
Migrated: 2019_02_12_152424_create_vendas_table

C:\laragon\www\primeiro
λ |
```

Foreign key constraints

| Actions | Constraint properties | Column | Foreign key constraint (INNODB) | | |
|-----------------------------|--------------------------------------|-------------------------|---------------------------------|---------------------|-----------------------|
| | | | Database | Table | Column |
| <div>Drop</div> | <div>vendas_id_cliente_foreign</div> | <div>id_cliente</div> | <div>primeiro</div> | <div>clientes</div> | <div>id_cliente</div> |
| | ON DELETE <div>CASCADE</div> | <div>+ Add column</div> | | | |
| | ON UPDATE <div>RESTRICT</div> | | | | |
| | <div>Constraint name</div> | <div></div> | <div></div> | <div></div> | <div></div> |
| | ON DELETE <div>RESTRICT</div> | <div>+ Add column</div> | | | |
| | ON UPDATE <div>RESTRICT</div> | | | | |
| <div>+ Add constraint</div> | | | | | |

Preview SQL

Save

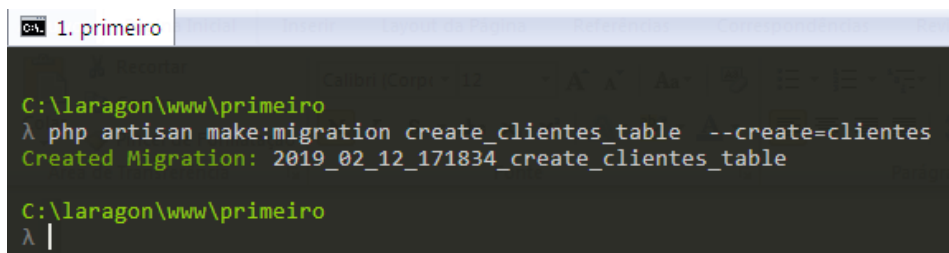
Aula 038 – Query Builder

Apague as duas migrations criadas anteriormente (clientes e vendas).

Apague o banco de dados primeiro e o recrie vazio.

No terminal recrie a migration “**create_clientes_table**” com o comando:

php artisan make:migration create_clientes_table --create=clientes

A screenshot of a terminal window with a dark background. The title bar at the top shows 'C:\laragon\www\primeiro'. The terminal content shows the command 'php artisan make:migration create_clientes_table --create=clientes' being executed, followed by the output 'Created Migration: 2019_02_12_171834_create_clientes_table'. The prompt 'C:\laragon\www\primeiro' and a cursor are visible at the bottom.

```
C:\laragon\www\primeiro
λ php artisan make:migration create_clientes_table --create=clientes
Created Migration: 2019_02_12_171834_create_clientes_table
C:\laragon\www\primeiro
λ |
```

database/migrations/2019_02_12_171834_create_clientes_table.php

<?php

```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class CreateClientesTable extends Migration
{
    public function up()
    {
        Schema::create('clientes', function (Blueprint $table) {
            $table->increments('id_cliente');
            $table->string('nome_cliente');
            $table->string('cidade');
        });
    }

    public function down()
    {
        Schema::dropIfExists('clientes');
    }
}
```

No terminal crie a tabela com o comando:

php artisan migrate

```
C:\laragon\www\primeiro
λ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2019_02_12_171834_create_clientes_table
Migrated: 2019_02_12_171834_create_clientes_table

C:\laragon\www\primeiro
λ |
```

Na tabela, no banco de dados, insira manualmente os seguintes registros:

| | <i>id_cliente</i>
INT UNSIGNED | <i>nome_cliente</i>
VARCHAR(255) | <i>cidade</i>
VARCHAR(255) |
|-----|-----------------------------------|-------------------------------------|-------------------------------|
| ▶ 1 | 2 | carlos | lisboa |
| 2 | 3 | cristina | rio de janeiro |
| 3 | 1 | joao | viseu |
| * 4 | NULL | NULL | NULL |

Crie a route a seguir, para testar a comunicação com a base de dados:

routes/web.php

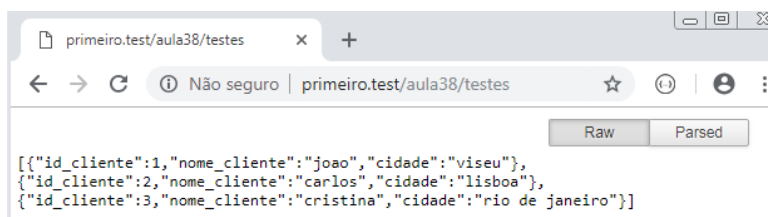
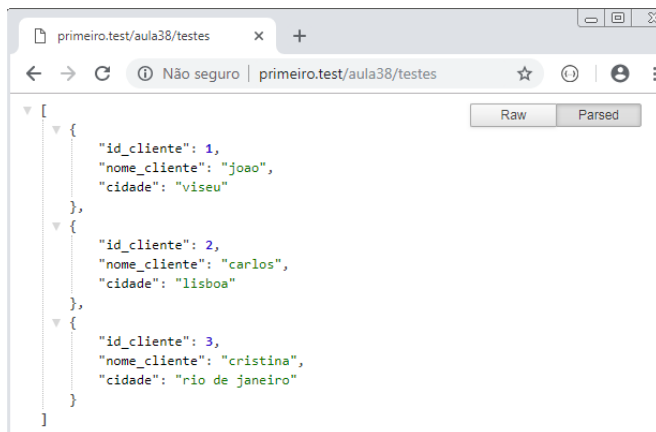
```
Route::get('aula38/testes', function(){
    // listar todos os registros da tabela clientes
    $dados = DB::table('clientes')->get();
    DD($dados);
});
```

```
primeiro.test/aula38/testes
Não seguro | primeiro.test/aula38/testes

Collection {#342}
  #items: array:3 [▼]
    0 => {#344}
      +id_cliente: 1
      +nome_cliente: "joao"
      +cidade: "viseu"
    }
    1 => {#346}
      +id_cliente: 2
      +nome_cliente: "carlos"
      +cidade: "lisboa"
    }
    2 => {#347}
      +id_cliente: 3
      +nome_cliente: "cristina"
      +cidade: "rio de janeiro"
    }
  ]
}
```

routes/web.php

```
Route::get('aula38/testes', function(){  
    // listar todos os registros da tabela clientes  
    $dados = DB::table('clientes')->get();  
    return $dados;  
});
```



Listar os dados de um cliente específico

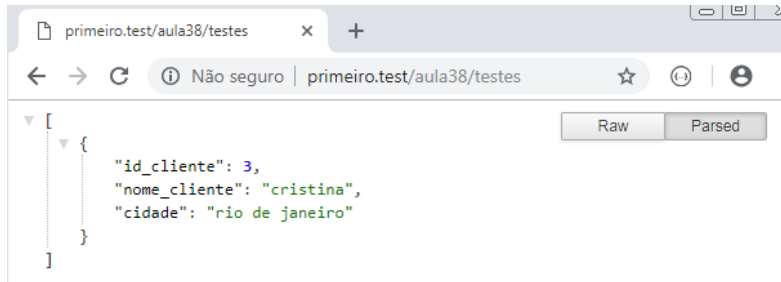
routes/web.php

```
Route::get('aula38/testes', function(){  
    // listar os dados do cliente joao  
    $dados = DB::table('clientes')->where('nome_cliente','joao')->get();  
    return $dados;  
});
```



routes/web.php

```
Route::get('aula38/testes', function(){  
    // listar os dados dos clientes do rio de janeiro  
    $dados = DB::table('clientes')->where('cidade','rio de janeiro')->get();  
    return $dados;  
});
```

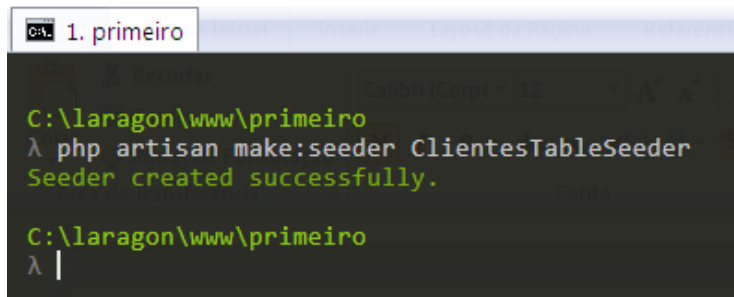


Aula 039 – Seeding

É um mecanismo que permite popular uma base de dados.

No terminal, crie o seeder chamado **ClientesTableSeeder** entrando com o comando:

php artisan make:seeder ClientesTableSeeder



```
1. primeiro  
C:\laragon\www\primeiro  
λ php artisan make:seeder ClientesTableSeeder  
Seeder created successfully.  
C:\laragon\www\primeiro  
λ |
```

database/seeds/ClientesTableSeeder.php

```
<?php
```

```
use Illuminate\Database\Seeder;
```

```
class ClientesTableSeeder extends Seeder
```

```
{
```

```
    public function run()
```

```
    {
```

```
        $dados = [
```

```
            'nome_cliente'=>'novo cliente',
```

```
            'cidade'=>'paris'
```

```
        ];
```

```
        DB::table('clientes')->insert($dados);
```

```
    }
```

```
}
```

database/seeds/CientesTableSeeder.php

```
<?php
```

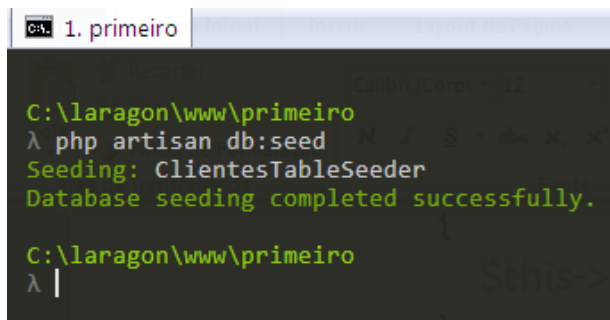
```
use Illuminate\Database\Seeder;
```

```
class DatabaseSeeder extends Seeder
```

```
{  
    public function run()  
    {  
        $this->call(CientesTableSeeder::class);  
    }  
}
```

Para inserir o registro, entre no terminal com o comando:

```
php artisan db:seed
```



```
C:\laragon\www\primeiro  
λ php artisan db:seed  
Seeding: CientesTableSeeder  
Database seeding completed successfully.  
  
C:\laragon\www\primeiro  
λ |
```

✓ Showing rows 0 - 3 (4 total, Query took 0.0030 seconds.)

`SELECT * FROM `clientes``

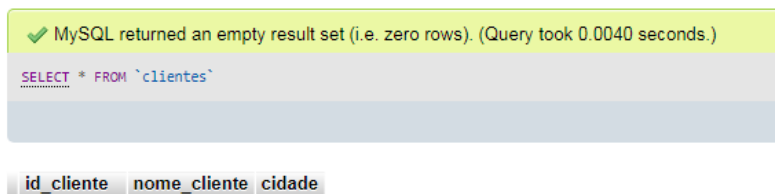
☐ Show all | Number of rows: 25 | Filter rows:

+ Options

| | | | id_cliente | nome_cliente | cidade |
|--------------------------|------|------|------------|--------------|-------------------------|
| <input type="checkbox"/> | Edit | Copy | Delete | 1 | joao viseu |
| <input type="checkbox"/> | Edit | Copy | Delete | 2 | carlos lisboa |
| <input type="checkbox"/> | Edit | Copy | Delete | 3 | cristina rio de janeiro |
| <input type="checkbox"/> | Edit | Copy | Delete | 4 | novo cliente paris |

Crie a route a seguir para apagar todos os dados da tabela clientes:

```
Route::get('aula39/delete_all', function(){
    DB::table('clientes')->delete();
    return 'Ok';
});
```



database/seeds/CientesTableSeeder.php

```
<?php

use Illuminate\Database\Seeder;

class CientesTableSeeder extends Seeder
{
    public function run()
    {
        $nomes = [
            'joao','antonio','carlos','miguel','ana','maria','suzana','cristina'

        ];

        $cidades = [
            'viseu','lisboa','lisboa','rio de janeiro','coimbra','são paulo','são paulo','rio de janeiro'
        ];

        for($indice = 0; $indice < count($nomes) ; $indice++){
            $dados = [
                'nome_cliente' => $nomes[$indice],
                'cidade' => $cidades[$indice]
            ];

            DB::table('clientes')->insert($dados);
        }
    }
}
```

























Para inserir os registros, entre no terminal com o comando:

php artisan db:seed

```
1. primeiro
C:\laragon\www\primeiro
λ php artisan db:seed
Seeding: ClientesTableSeeder
Database seeding completed successfully.

C:\laragon\www\primeiro
λ |
```

+ Options

| | | | id_cliente | nome_cliente | cidade |
|--------------------------|----------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|--------------|-------------------------|
| <input type="checkbox"/> |  Edit |  Copy |  Delete | 6 | joao viseu |
| <input type="checkbox"/> |  Edit |  Copy |  Delete | 7 | antonio lisboa |
| <input type="checkbox"/> |  Edit |  Copy |  Delete | 8 | carlos lisboa |
| <input type="checkbox"/> |  Edit |  Copy |  Delete | 9 | miguel rio de janeiro |
| <input type="checkbox"/> |  Edit |  Copy |  Delete | 10 | ana coimbra |
| <input type="checkbox"/> |  Edit |  Copy |  Delete | 11 | maria são paulo |
| <input type="checkbox"/> |  Edit |  Copy |  Delete | 12 | suzana são paulo |
| <input type="checkbox"/> |  Edit |  Copy |  Delete | 13 | cristina rio de janeiro |

Aula 040 – Tinker

No terminal, entre com o seguinte comando:

```
php artisan tinker
```

Em seguida, no prompt entre com:

```
DB::table('clientes')->get()
```

```
1. primeiro
C:\laragon\www\primeiro
λ php artisan tinker
Psy Shell v0.9.9 (PHP 7.2.11 - cli) by Justin Hileman
>>> DB::table('clientes')->get()
=> Illuminate\Support\Collection {#2999
  all: [
    {#2992
      +"id_cliente": 6,
      +"nome_cliente": "joao",
      +"cidade": "viseu",
    },
    {#3004
      +"id_cliente": 7,
      +"nome_cliente": "antonio",
      +"cidade": "lisboa",
    },
    {#3005
      +"id_cliente": 8,
      +"nome_cliente": "carlos",
      +"cidade": "lisboa",
    },
    {#3006
      +"id_cliente": 9,
      +"nome_cliente": "miguel",
      +"cidade": "rio de janeiro",
    },
    {#3007
      +"id_cliente": 10,
      +"nome_cliente": "ana",
      +"cidade": "coimbra",
    },
    {#3008
      +"id_cliente": 11,
      +"nome_cliente": "maria",
      +"cidade": "são paulo",
    },
  ],
}
```

Entre com o seguinte comando:

```
DB::table('clientes')->where('nome_cliente', 'joao')->get()
```

```
>>> DB::table('clientes')->where('nome_cliente', 'joao')->get()
=> Illuminate\Support\Collection {#3002
  all: [
    {#3018
      +"id_cliente": 6,
      +"nome_cliente": "joao",
      +"cidade": "viseu",
    },
  ],
}
>>> |
```

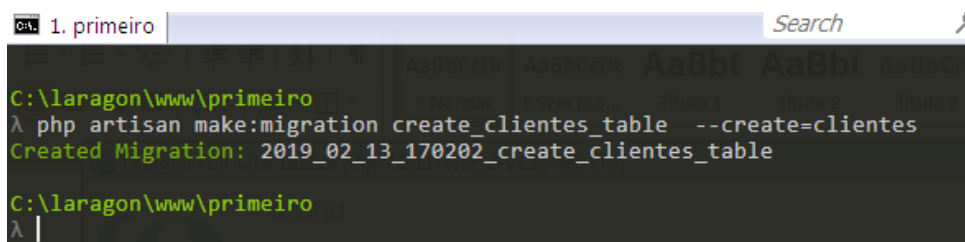
Aula 041 – Exercício de Seeding

Apague a migration `create_clientes_table`.

Apague o banco de dados primeiro e o recrie vazio.

No terminal recrie a migration “`create_clientes_table`” com o comando:

`php artisan make:migration create_clientes_table --create=clientes`

A screenshot of a terminal window with a dark background. The title bar shows '1. primeiro' and a search icon. The terminal text shows the command 'php artisan make:migration create_clientes_table --create=clientes' being executed in the directory 'C:\laragon\www\primeiro'. The output is 'Created Migration: 2019_02_13_170202_create_clientes_table'.

```
C:\laragon\www\primeiro
λ php artisan make:migration create_clientes_table --create=clientes
Created Migration: 2019_02_13_170202_create_clientes_table

C:\laragon\www\primeiro
λ |
```

database/migrations/2019_02_13_170202_create_clientes_table.php

`<?php`

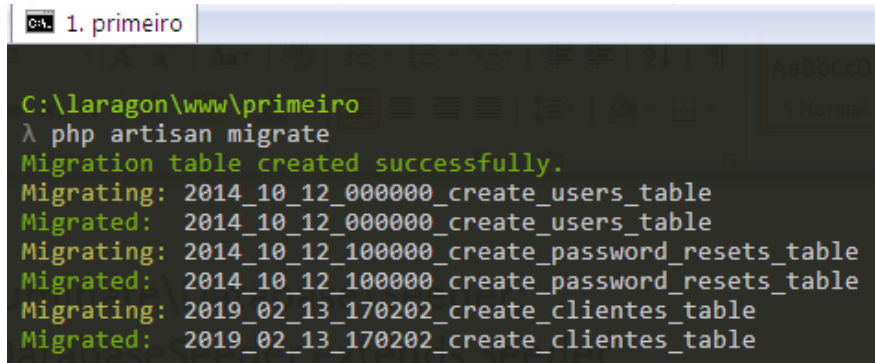
```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class CreateClientesTable extends Migration
{
    public function up()
    {
        Schema::create('clientes', function (Blueprint $table) {
            $table->increments('id_cliente');
            $table->string('nome');
            $table->string('sobrenome');
            $table->string('cidade');
            $table->string('email');
            $table->date('data_nascimento');
        });
    }

    public function down()
    {
        Schema::dropIfExists('clientes');
    }
}
```

Para criar a tabela:

php artisan migrate



```
C:\laragon\www\primeiro
λ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2019_02_13_170202_create_clientes_table
Migrated: 2019_02_13_170202_create_clientes_table
```

database/seeds/CientesTableSeeder.php

<?php

use Illuminate\Database\Seeder;

class CientesTableSeeder extends Seeder

{

public function run()

{

// Número total de clientes que pretende inserir na base de dados

\$numero_total_clientes = 500;

// Define fonte de informações para seeds

\$nomes =

['Joao','Ana','Antonio','Marilia','Gabriela','Carlos','Francisco','Marta','Suzana','Alice','Pedro','Rui','Manuel','Gilberto','Mario','Guilherme','Rubens','Mateus','Luis','Miguel','Elias','Mariana','Diego','Alberto','Alexandre','Rita','Alfredo','Ademir','Augusto','Bernardo','Bruno','Cesar','Rodrigo','Reinaldo','Afonso','Marcia','Simone','Diogenes','Fernando','Isabel','Roberto','Ricardo','Tulio','Marcelo','Nair','Nadia','Katia','Renata','Geraldo','Irene','Adalberto','Denise','Jose','Jair','Eduardo','Heitor','Alex','Sandra','Erica','Karina','Marlene','Josias','Luana','Tereza','Monica','Marina','Moacir','Paulo','Ramon','Sergio'];

\$sobrenomes=['Rodrigues','Santos','Oliveira','Martins','Costa','Teixeira','Almeida','Ribeiro','Pereira','Simoes','Damiao','Souza','Alves','Valentino','Guimaraes','Pacheco','Cardoso','Moreira','Silva','Alencar','Figueiredo','Carvalho','Gutierrez','Batista','Pinheiro','Bueno','Andrade','Cruz','Aguiar','Malta','Albuquerque','Vilela','Domingues','Camargo','Lima','Ferraz','Vaz','Rufino','Dias','Barroso','Seixas','Junqueira','Alcantara','Garcia','Matos','Costa','Vasconcelos','Pinto','Moreira','Ferreira'];

```
$cidades=['Lisboa','Porto','Coimbra','Viseu','Faro','Braga','Aveiro','Fortaleza','Sao  
Paulo','Santos','Belem','Brasilia','Paris','Manaus','Roma','Curitiba','Salvador','Vitoria','Cuiaba','Guarulh  
os','Osasco','Goiania','Londrina','Limeira','Campinas','Taubate','Niteroi','Jundiai','Barretos','Franca','C  
olatina','Linhares','Teresina','Aracaju','Natal','Palmas','Arizona','Texas','Setubal','Madrid'];
```

```
$dominios_de_email=['gmail.com','hotmail.com','uol.com.br','oesp.com.br','redcard.com.br','ves  
per.com.br','kepler.com.br','yahoo.com.br','ig.com.br','bol.com.br','terra.com.br','venturini.com.br','  
globo.com','aol.com','artplan.com.br','click21.com.br','uninet.com.br','barralink.com.br','avipal.com.  
br','edunet.com.br','brturbo.com','ibest.com.br','msn.com','digiplus.com.br','sercomtel.com.br','twist  
er.com.br','engeplus.com.br','zipmail.com.br','fasterair.com.br','maxiplast.com'];
```

```
// ciclo para inserção de clientes no BD
```

```
for($i = 0; $i < $numero_total_clientes; $i++){  
    $nome = $nomes[array_rand($nomes)];  
    $sobrenome = $sobrenomes[array_rand($sobrenomes)];  
    $cidade = $cidades[array_rand($cidades)];
```

```
// data de nascimento
```

```
$ano = rand(1960, 2000);  
$mes = rand(1,12);  
$dia = rand(1,28);
```

```
$data_nascimento = $ano . '-' . $mes . '-' . $dia;
```

```
// cria o endereço de email do cliente
```

```
$email = strtolower($nome) . '.' . strtolower($sobrenome) . '@' .  
$dominios_de_email[array_rand($dominios_de_email)];
```

```
$dados = [  
    'nome' => $nome,  
    'sobrenome' => $sobrenome,  
    'cidade' => $cidade,  
    'email' => $email,  
    'data_nascimento' => $data_nascimento  
];
```

```
DB::table('clientes')->insert($dados);
```

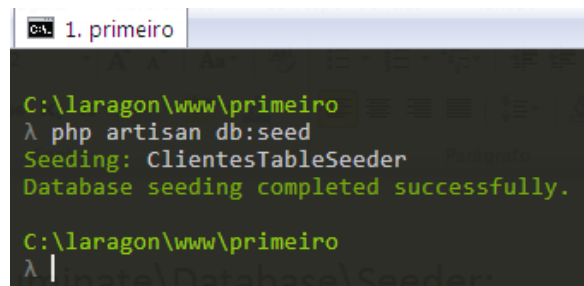
```
}  
}  
}
```

database/seeds/DatabaseSeeder.php

```
<?php
use Illuminate\Database\Seeder;
class DatabaseSeeder extends Seeder
{
    public function run()
    {
        $this->call(ClientesTableSeeder::class);
    }
}
```

Para inserir os dados na tabela, no terminal, entre com o comando:

php artisan db:seed



```
C:\laragon\www\primeiro
λ php artisan db:seed
Seeding: ClientesTableSeeder
Database seeding completed successfully.

C:\laragon\www\primeiro
λ
```

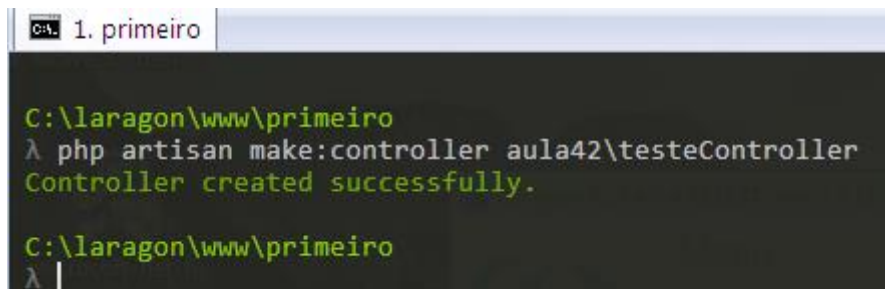
| | <i>id_cliente</i>
UInt32 | <i>nome</i>
VarChar(255) | <i>sobrenome</i>
VarChar(255) | <i>cidade</i>
VarChar(255) | <i>email</i>
VarChar(255) | <i>data_nascimento</i>
Date |
|----|-----------------------------|-----------------------------|----------------------------------|-------------------------------|---------------------------------|--------------------------------|
| 1 | 1 | Bernardo | Domingues | Taubate | bernardo.domingues@venturini... | 1983-10-10 |
| 2 | 2 | Nadia | Simoes | Belem | nadia.simoes@aol.com | 1968-04-16 |
| 3 | 3 | Erica | Bueno | Coimbra | erica.bueno@oesp.com.br | 1983-12-24 |
| 4 | 4 | Alice | Costa | Franca | alice.costa@aol.com | 2000-04-09 |
| 5 | 5 | Mateus | Domingues | Linhares | mateus.domingues@msn.com | 1986-10-16 |
| 6 | 6 | Diego | Pinto | Teresina | diego.pinto@avipal.com.br | 1970-04-21 |
| 7 | 7 | Adalberto | Matos | Roma | adalberto.matos@kepler.com.br | 1980-09-01 |
| 8 | 8 | Denise | Vaz | Barretos | denise.vaz@aol.com | 1996-11-13 |
| 9 | 9 | Ademir | Andrade | Cuiaba | ademir.andrade@oesp.com.br | 1996-01-21 |
| 10 | 10 | Simone | Pinheiro | Brasilia | simone.pinheiro@avipal.com.br | 1994-05-21 |
| 11 | 11 | Marcelo | Vasconcelos | Santos | marcelo.vasconcelos@gmail.com | 1968-12-09 |
| 12 | 12 | Rubens | Cardoso | Santos | rubens.cardoso@brturbo.com | 1996-01-24 |
| 13 | 13 | Carlos | Malta | Texas | carlos.malta@globo.com | 1993-03-02 |
| 14 | 14 | Alex | Malta | Guarulhos | alex.malta@edunet.com.br | 1993-01-25 |
| 15 | 15 | Denise | Vasconcelos | Franca | denise.vasconcelos@hotmail.com | 1985-02-18 |
| 16 | 16 | Rita | Alencar | Brasilia | rita.alencar@ig.com.br | 1966-06-18 |
| 17 | 17 | Paulo | Matos | Osasco | paulo.matos@yahoo.com.br | 1966-10-14 |
| 18 | 18 | Paulo | Albuquerque | Teresina | paulo.albuquerque@terra.com.br | 1983-09-18 |
| 19 | 19 | Miguel | Lima | Arizona | miguel.lima@ig.com.br | 1991-12-21 |
| 20 | 20 | Reinaldo | Vaz | Belem | reinaldo.vaz@msn.com | 1981-09-13 |

| | <i>id_cliente</i>
<i>UInt32</i> | <i>nome</i>
<i>VarChar(255)</i> | <i>sobrenome</i>
<i>VarChar(255)</i> | <i>cidade</i>
<i>VarChar(255)</i> | <i>email</i>
<i>VarChar(255)</i> | <i>data_nascimento</i>
<i>Date</i> |
|-----|------------------------------------|------------------------------------|-----------------------------------------|--------------------------------------|-------------------------------------|---------------------------------------|
| 481 | 481 | Diogenes | Pinto | Arizona | diogenes.pinto@artplan.com.br | 2000-10-01 |
| 482 | 482 | Mario | Simoes | Roma | mario.simoes@edunet.com.br | 1982-09-03 |
| 483 | 483 | Sergio | Rodrigues | Colatina | sergio.rodrigues@maxiplast.com | 1977-03-21 |
| 484 | 484 | Luis | Costa | Manaus | luis.costa@ibest.com.br | 1969-07-11 |
| 485 | 485 | Rui | Almeida | Manaus | rui.almeida@hotmail.com | 1999-04-24 |
| 486 | 486 | Sergio | Camargo | Linhares | sergio.camargo@maxiplast.com | 1984-03-17 |
| 487 | 487 | Bernardo | Martins | Campinas | bernardo.martins@ibest.com.br | 1992-07-04 |
| 488 | 488 | Cesar | Moreira | Arizona | cesar.moreira@digiplus.com.br | 1978-04-27 |
| 489 | 489 | Ramon | Costa | Fortaleza | ramon.costa@ibest.com.br | 1972-09-25 |
| 490 | 490 | Marina | Almeida | Londrina | marina.almeida@barralink.com.br | 1998-09-14 |
| 491 | 491 | Karina | Figueiredo | Franca | karina.figueiredo@vesper.com.br | 1983-05-24 |
| 492 | 492 | Antonio | Silva | Salvador | antonio.silva@avipal.com.br | 1972-04-19 |
| 493 | 493 | Monica | Matos | Osasco | monica.matos@twister.com.br | 1988-12-25 |
| 494 | 494 | Josias | Pinto | Sao Paulo | josias.pinto@gmail.com | 1969-10-26 |
| 495 | 495 | Mario | Pinto | Roma | mario.pinto@engeplus.com.br | 1987-11-08 |
| 496 | 496 | Adalberto | Cardoso | Arizona | adalberto.cardoso@globo.com | 1995-02-08 |
| 497 | 497 | Ramon | Vasconcelos | Niteroi | ramon.vasconcelos@msn.com | 1989-12-04 |
| 498 | 498 | Marlene | Vaz | Cuiaba | marlene.vaz@bol.com.br | 1979-04-24 |
| 499 | 499 | Isabel | Carvalho | Setubal | isabel.carvalho@gmail.com | 1995-11-22 |
| 500 | 500 | Gabriela | Martins | Manaus | gabriela.martins@yahoo.com.br | 1960-12-08 |

Aula 042 – Query Builder - Exercícios

No terminal, crie um controlador chamado "testeController" em uma pasta chamada "aula42", entrando com o seguinte comando:

```
php artisan make:controller aula42\testeController
```



```
1. primeiro  
C:\laragon\www\primeiro  
λ php artisan make:controller aula42\testeController  
Controller created successfully.  
C:\laragon\www\primeiro  
λ
```

O Query Builder usa a classe DB.

app/Http/Controllers/aula42/testeController.php

```
<?php
```

```
namespace App\Http\Controllers\aula42;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

```
use Illuminate\Support\Facades\DB;
```

```
class testeController extends Controller
```

```
{
```

```
    public function teste(){
```

```
        $dados = DB::table('clientes')->get();
```

```
        return $dados;
```

```
    }
```

```
}
```

routes/web

```
Route::get('aula42/teste', 'aula42\testeController@teste');
```

primeiro.test/aula42/teste

Não seguro | primeiro.test/aula42/teste

Raw Parsed

```
[
  {
    "id_cliente": 1,
    "nome": "Bernardo",
    "sobrenome": "Domingues",
    "cidade": "Taubate",
    "email": "bernardo.domingues@venturini.com.br",
    "data_nascimento": "1983-10-10"
  },
  {
    "id_cliente": 2,
    "nome": "Nadia",
    "sobrenome": "Simoes",
    "cidade": "Belem",
    "email": "nadia.simoes@aol.com",
    "data_nascimento": "1968-04-16"
  },
  {
    "id_cliente": 3,
    "nome": "Erica",
    "sobrenome": "Bueno",
    "cidade": "Coimbra",
    "email": "erica.bueno@oesp.com.br",
    "data_nascimento": "1983-12-24"
  },
  {
    "id_cliente": 4,
    "nome": "Alice",
    "sobrenome": "Costa",
    "cidade": "Franca",
    "email": "alice.costa@aol.com",
    "data_nascimento": "2000-04-09"
  }
],
```

primeiro.test/aula42/teste

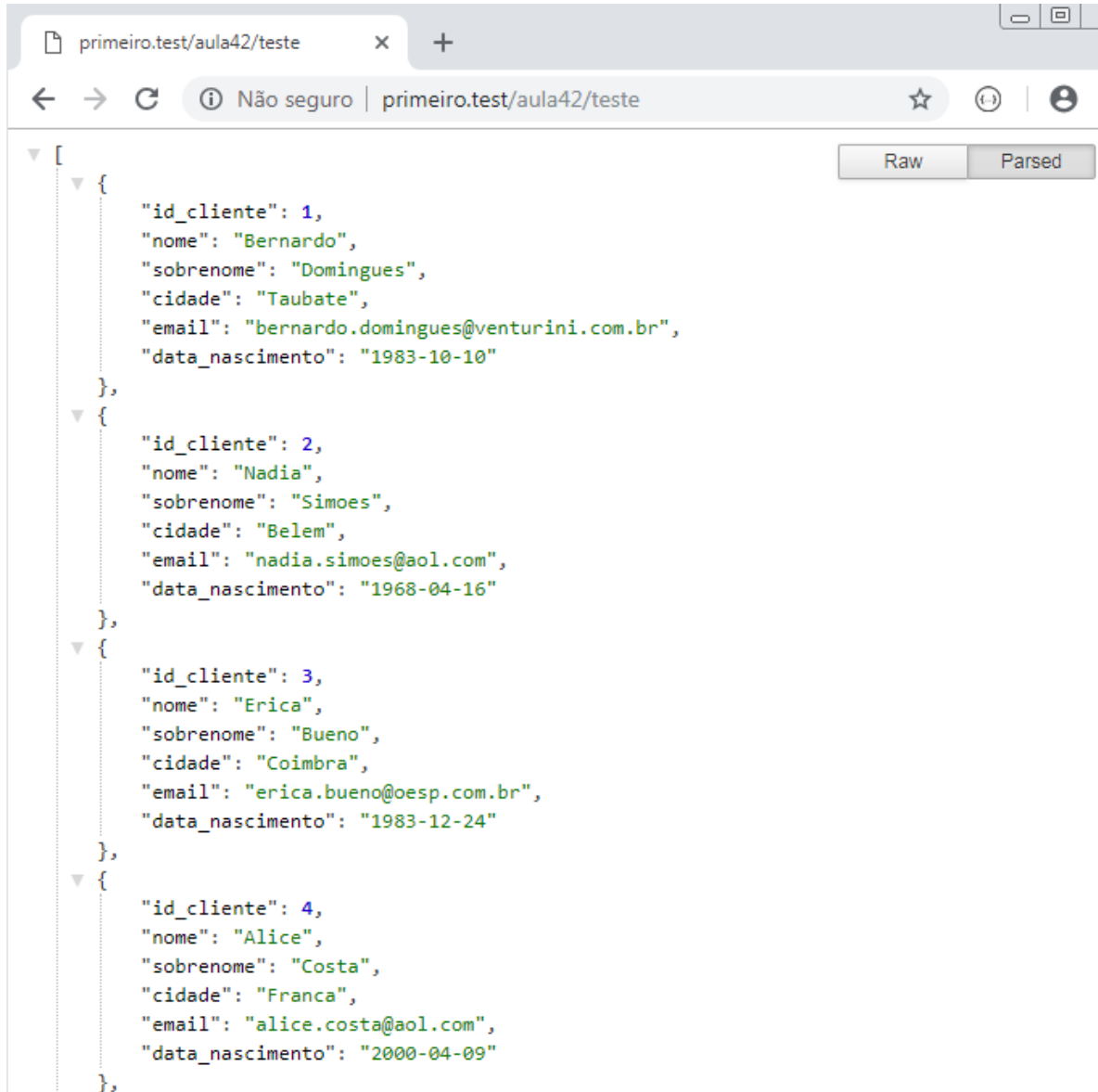
Não seguro | primeiro.test/aula42/teste

Raw Parsed

▶ [...] // 500 items

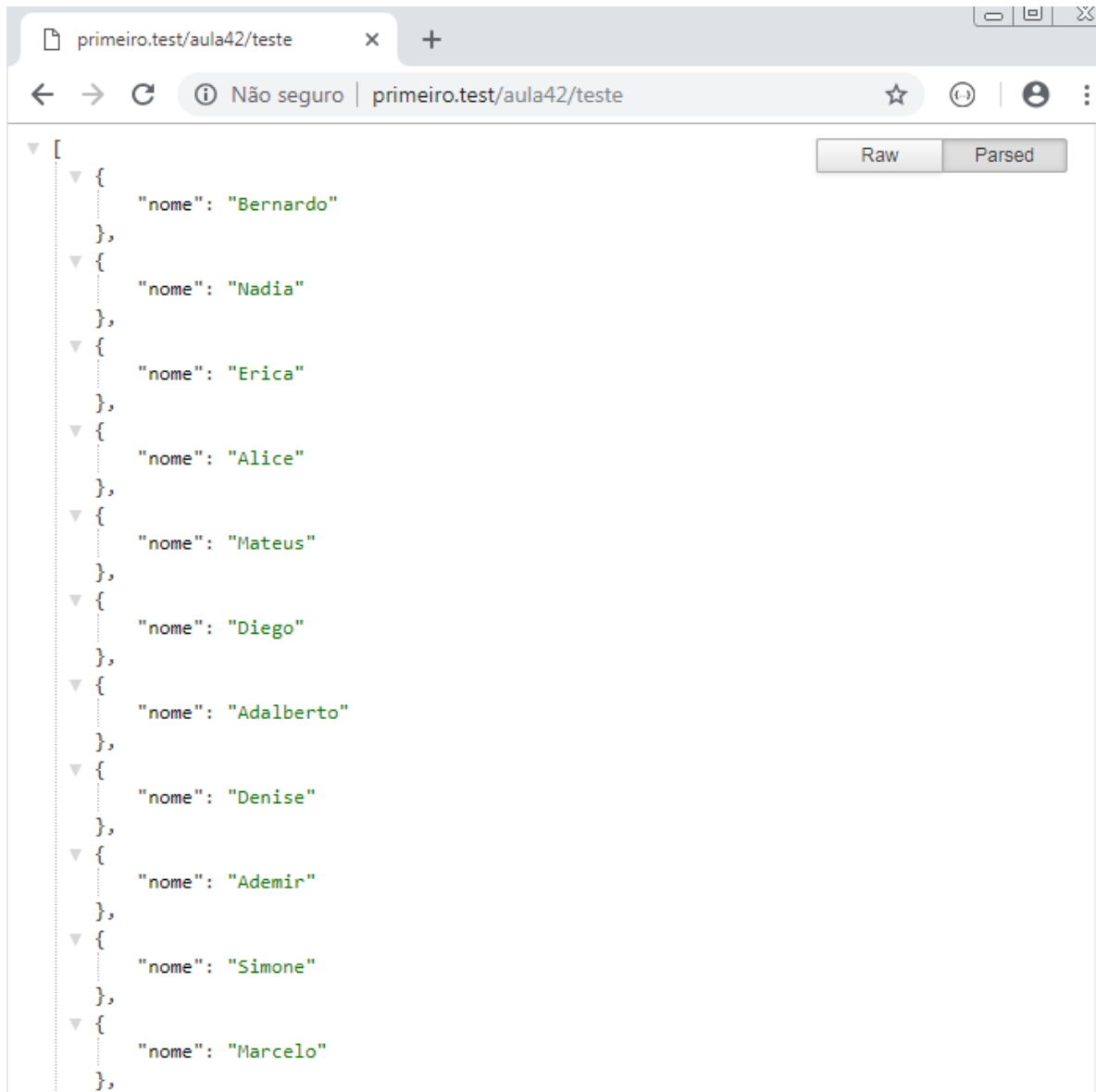
app/Http/Controllers/aula42/testeController.php

```
public function teste(){  
    $dados = DB::select('select * from clientes');  
    return $dados;  
}
```



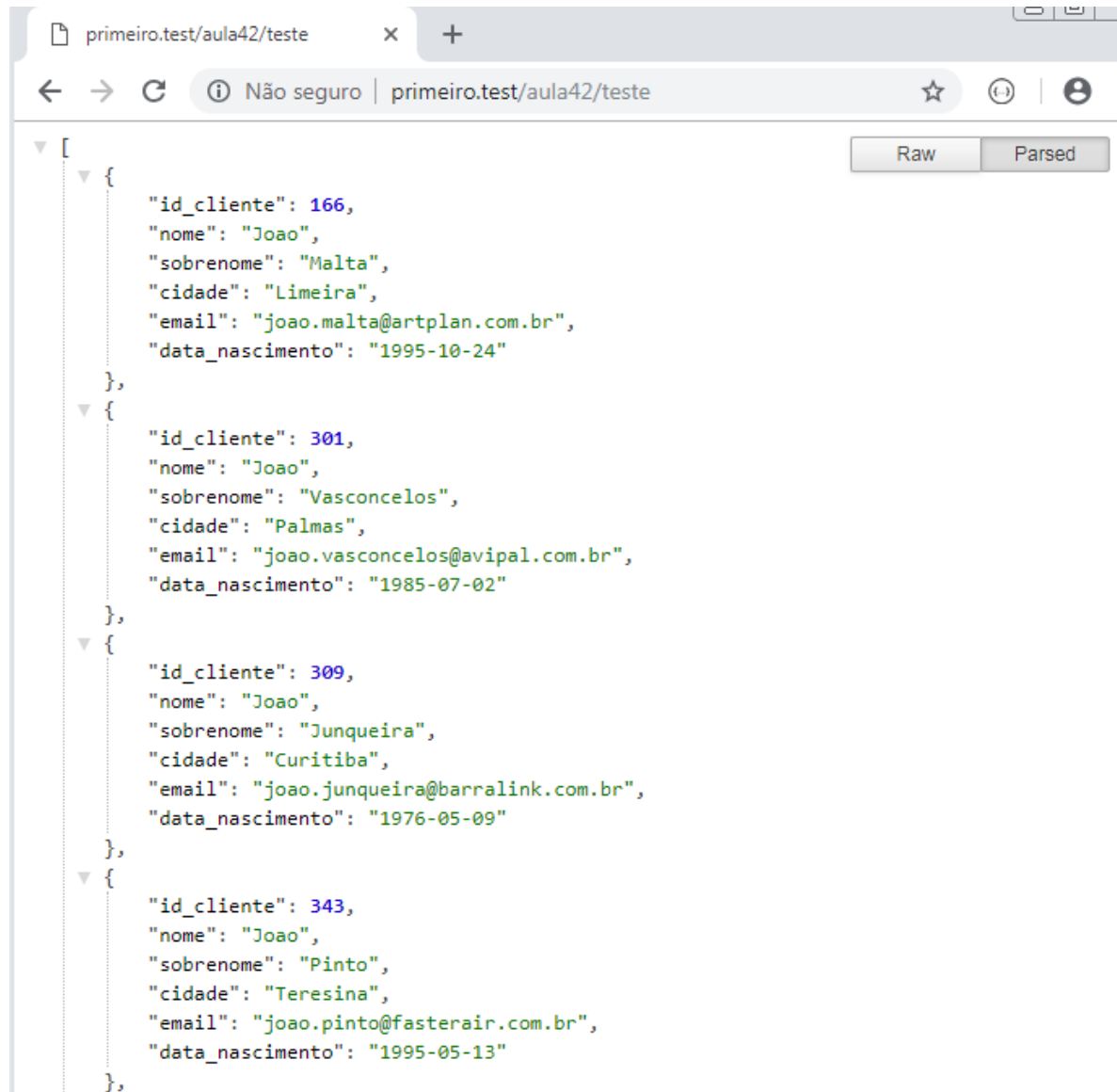
app/Http/Controllers/aula42/testeController.php

```
public function teste(){  
    $dados = DB::table('clientes')->select('nome')->get();  
    return $dados;  
}
```



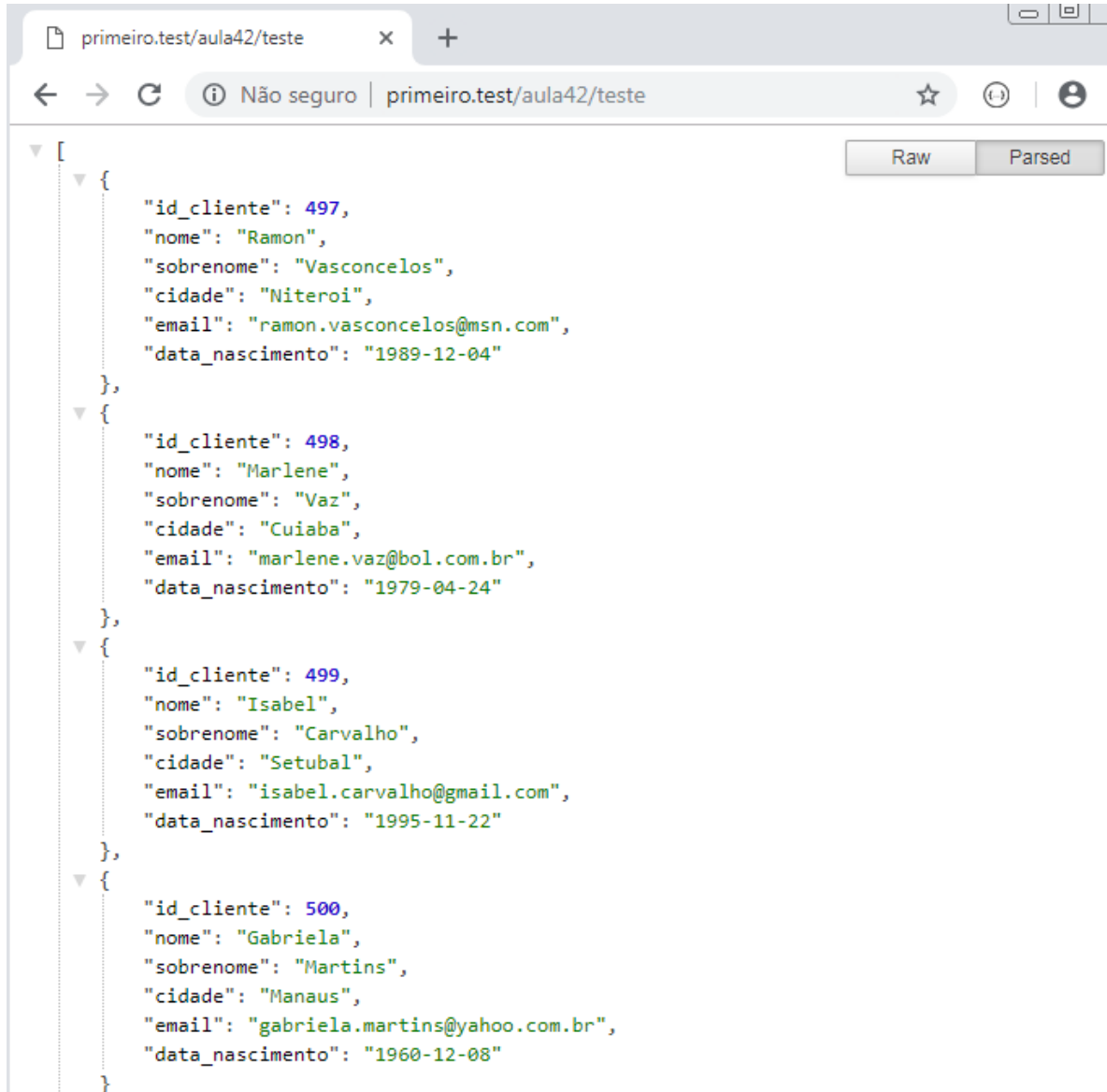
app/Http/Controllers/aula42/testeController.php

```
public function teste(){  
    $dados = DB::table('clientes')->where('nome','Joao')->get();  
    return $dados;  
}
```



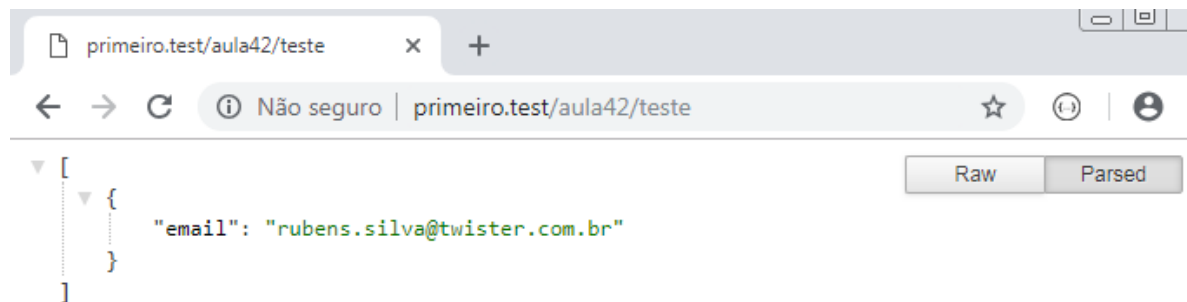
app/Http/Controllers/aula42/testeController.php

```
public function teste(){
    $dados = DB::table('clientes')->where('id_cliente','>',496)->get();
    return $dados;
}
```



app/Http/Controllers/aula42/testeController.php

```
public function teste(){  
    $id = 123;  
    $dados = DB::table('clientes')  
        ->where('id_cliente',$id)  
        ->select('email')  
        ->get();  
  
    return $dados;  
}
```



app/Http/Controllers/aula42/testeController.php

```
public function teste(){
    $dados = DB::table('clientes')
        ->whereYear('data_nascimento','=',1961)
        ->orWhere('sobrenome','Pinheiro')
        ->get();

    return $dados;
}
```

primeiro.test/aula42/teste

Não seguro | primeiro.test/aula42/teste

Raw | Parsed

```
[
  {
    "id_cliente": 10,
    "nome": "Simone",
    "sobrenome": "Pinheiro",
    "cidade": "Brasília",
    "email": "simone.pinheiro@avipal.com.br",
    "data_nascimento": "1994-05-21"
  },
  {
    "id_cliente": 60,
    "nome": "Eduardo",
    "sobrenome": "Pinheiro",
    "cidade": "Fortaleza",
    "email": "eduardo.pinheiro@ibest.com.br",
    "data_nascimento": "1964-07-21"
  },
  {
    "id_cliente": 117,
    "nome": "Elias",
    "sobrenome": "Pinheiro",
    "cidade": "Franca",
    "email": "elias.pinheiro@ig.com.br",
    "data_nascimento": "1984-05-06"
  },
  {
    "id_cliente": 130,
    "nome": "Alfredo",
    "sobrenome": "Moreira",
    "cidade": "Osasco",
    "email": "alfredo.moreira@uol.com.br",
    "data_nascimento": "1961-12-14"
  }
]
```

primeiro.test/aula42/teste

Não seguro | primeiro.test/aula42/teste

Raw | Parsed

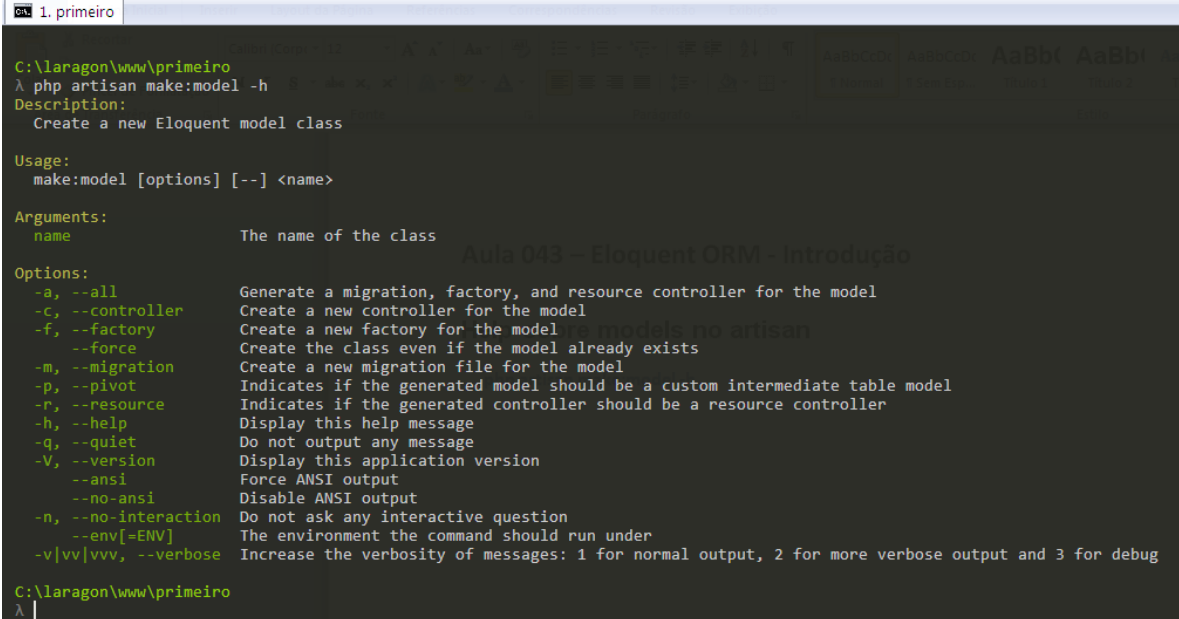
▶ [...] // 19 items

Aula 043 – Eloquent ORM - Introdução

Help sobre models no artisan

No terminal, entre com o comando:

```
php artisan make:model -h
```



```
C:\laragon\www\primeiro
λ php artisan make:model -h
Description:
  Create a new Eloquent model class

Usage:
  make:model [options] [--] <name>

Arguments:
  name                The name of the class

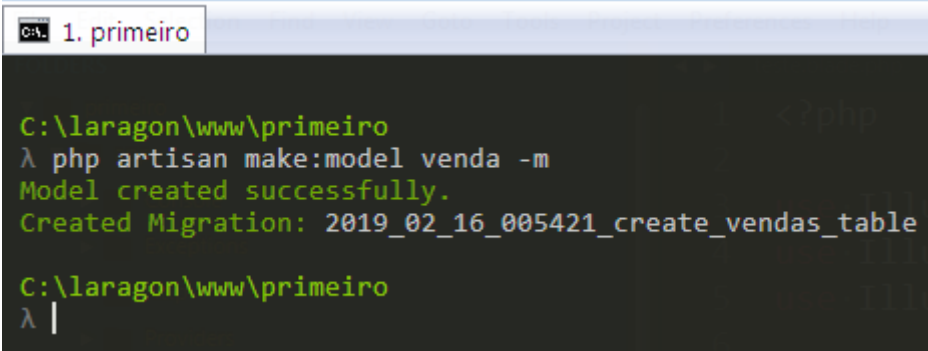
Options:
  -a, --all            Generate a migration, factory, and resource controller for the model
  -c, --controller     Create a new controller for the model
  -f, --factory        Create a new factory for the model
  --force             Create the class even if the model already exists
  -m, --migration      Create a new migration file for the model
  -p, --pivot          Indicates if the generated model should be a custom intermediate table model
  -r, --resource       Indicates if the generated controller should be a resource controller
  -h, --help           Display this help message
  -q, --quiet          Do not output any message
  -V, --version        Display this application version
  --ansi              Force ANSI output
  --no-ansi           Disable ANSI output
  -n, --no-interaction Do not ask any interactive question
  --env[=ENV]         The environment the command should run under
  -v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

C:\laragon\www\primeiro
λ
```

Criando uma model e uma migration simultaneamente

No terminal, entre com o comando:

```
php artisan make:model venda -m
```



```
C:\laragon\www\primeiro
λ php artisan make:model venda -m
Model created successfully.
Created Migration: 2019_02_16_005421_create_vendas_table

C:\laragon\www\primeiro
λ
```

Model

app/venda.php

```
<?php

namespace App;
use Illuminate\Database\Eloquent\Model;

class venda extends Model
{
    //
}
```

Migration

Http/models/migrations/2019_02_16_005421_create_vendas_table.php

```
<?php

use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateVendasTable extends Migration
{
    public function up()
    {
        Schema::create('vendas', function (Blueprint $table) {
            $table->increments('id');
            $table->timestamps();
        });
    }

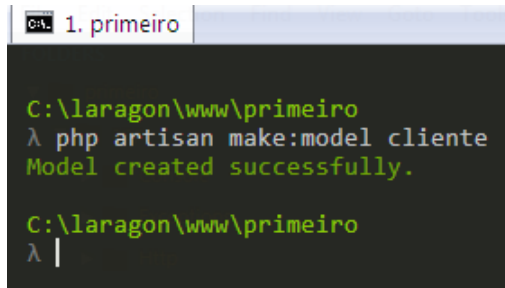
    public function down()
    {
        Schema::dropIfExists('vendas');
    }
}
```

O Model é criado no singular, porém a tabela será criada no plural: vendas (snake case).

Aulas 044 – Eloquent - Relação entre Model e tabela

No terminal, entre com o comando:

`php artisan make:model cliente`



```
C:\laragon\www\primeiro
λ php artisan make:model cliente
Model created successfully.

C:\laragon\www\primeiro
λ |
```

routes/web.php

```
Route::get('aula44/clientes', function(){
    // Query Builder
    $dados = DB::table('clientes')->get();
    return $dados;
});
```



```
primeiro.test/aula44/clientes x +
← → ↻ ⓘ Não seguro | primeiro.test/aula44/clientes

[
  {
    "id_cliente": 1,
    "nome": "Bernardo",
    "sobrenome": "Domingues",
    "cidade": "Taubate",
    "email": "bernardo.domingues@venturini.com.br",
    "data_nascimento": "1983-10-10"
  },
  {
    "id_cliente": 2,
    "nome": "Nadia",
    "sobrenome": "Simoes",
    "cidade": "Belem",
    "email": "nadia.simoes@aol.com",
    "data_nascimento": "1968-04-16"
  },
  {
    "id_cliente": 3,
    "nome": "Erica",
    "sobrenome": "Bueno",
    "cidade": "Coimbra",
    "email": "erica.bueno@oesp.com.br",
    "data_nascimento": "1983-12-24"
  },
  {
    "id_cliente": 4,
    "nome": "Alice",
    "sobrenome": "Costa",
    "cidade": "Franca",
    "email": "alice.costa@aol.com",
    "data_nascimento": "2000-04-09"
  }
]
```

routes/web.php

```
use App\cliente;
```

```
Route::get('aula44/clientes', function(){  
    // Eloquent ORM  
    $dados = cliente::get();  
  
    return $dados;  
});
```



routes/web.php

```
Route::get('aula44/clientes', function(){  
  
    // Eloquent ORM  
    $dados = cliente::where('nome','Joao')->get();  
  
    return $dados;  
});
```



app/teste.php

```
<?php
```

```
namespace App;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class teste extends Model
```

```
{
    protected $table = 'clientes';
}
```

routes/web.php

use App\teste;

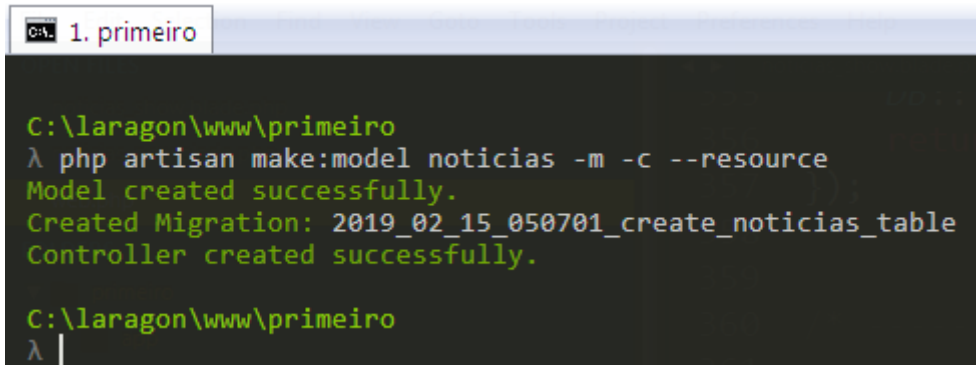
```
Route::get('aula44/clientes', function(){  
    $dados = teste::where('nome','Joao')->get();  
    return $dados;  
});
```



Aulas 045, 046 e 047 – Colocando em prática o MVC

No terminal, crie simultaneamente um model chamado "**noticias**", uma migration (chamada **create_noticias_table**) e um controller chamado "**noticias**", entrando com o seguinte comando:

```
php artisan make:model noticias -m -c --resource
```



```
1. primeiro

C:\laragon\www\primeiro
λ php artisan make:model noticias -m -c --resource
Model created successfully.
Created Migration: 2019_02_15_050701_create_noticias_table
Controller created successfully.

C:\laragon\www\primeiro
λ |
```

database/migrations/2019_02_15_050701_create_noticias_table.php

```
<?php
```

```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class CreateNoticiasTable extends Migration
{
    public function up()
    {
        Schema::create('noticias', function (Blueprint $table) {
            $table->increments('id_noticia');
            $table->string('titulo');
            $table->text('texto');
            $table->string('autor');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('noticias');
    }
}
```

Rotas criadas

php route:list

| | | | |
|-----------|-------------------------|------------------|-------------------------------------------------|
| GET HEAD | noticias | noticias.index | App\Http\Controllers\noticiasController@index |
| POST | noticias | noticias.store | App\Http\Controllers\noticiasController@store |
| GET HEAD | noticias/create | noticias.create | App\Http\Controllers\noticiasController@create |
| GET HEAD | noticias/{noticia} | noticias.show | App\Http\Controllers\noticiasController@show |
| PUT PATCH | noticias/{noticia} | noticias.update | App\Http\Controllers\noticiasController@update |
| DELETE | noticias/{noticia} | noticias.destroy | App\Http\Controllers\noticiasController@destroy |
| GET HEAD | noticias/{noticia}/edit | noticias.edit | App\Http\Controllers\noticiasController@edit |

routes/web.php

```
Route::resource('noticias', 'noticiasController');
```

Criando a tabela

Para criar, entre no terminal com o seguinte comando:

php artisan migrate

```
1. primeiro
C:\laragon\www\primeiro
λ php artisan migrate
Migrating: 2019_02_15_050701_create_noticias_table
Migrated: 2019_02_15_050701_create_noticias_table

C:\laragon\www\primeiro
λ
```

Inserindo dados na tabela

Insira manualmente na tabela três registros:

| | id_noticia | titulo | texto | autor | created_at | updated_at |
|---------------------------------------------------------------------------------------------|------------|------------------------------|-------------------------------------------------------|--------|---------------------|---------------------|
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 1 | Greve geral | Aconteceu um greve geral no meu país. | João | 2019-02-15 00:00:00 | 2019-02-15 00:00:00 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 2 | Atentado à bomba na Síria. | Aconteceu um atentado à bomba na Síria que resulto... | Ana | 2019-02-15 00:01:00 | 2019-02-15 00:01:00 |
| <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete | 3 | Encontrado a cura do câncer. | Descoberta a vacina para prevenção de qualquer tip... | Filipe | 2019-02-15 00:02:00 | 2019-02-15 00:02:00 |

app/Http/NoticiasController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\noticias;
```

```
use Illuminate\Http\Request;
```

```
class NoticiasController extends Controller
```

```
{
```

```
    public function index()
```

```
    {
```

```
        $datos = noticias::all();
```

```
        return $datos;
```

```
    }
```

```
    public function create()
```

```
    {
```

```
        //
```

```
    }
```

```
    public function store(Request $request)
```

```
    {
```

```
        //
```

```
    }
```

```
    public function show($id)
```

```
    {
```

```
        $datos = noticias::where('id_noticia', $id)->get();
```

```
        return $datos;
```

```
    }
```

```
    public function edit(noticias $noticias)
```

```
    {
```

```
        //
```

```
    }
```

```
    public function update(Request $request, noticias $noticias)
```

```
    {
```

```
        //
```

```
    }
```

```
    public function destroy(noticias $noticias)
```

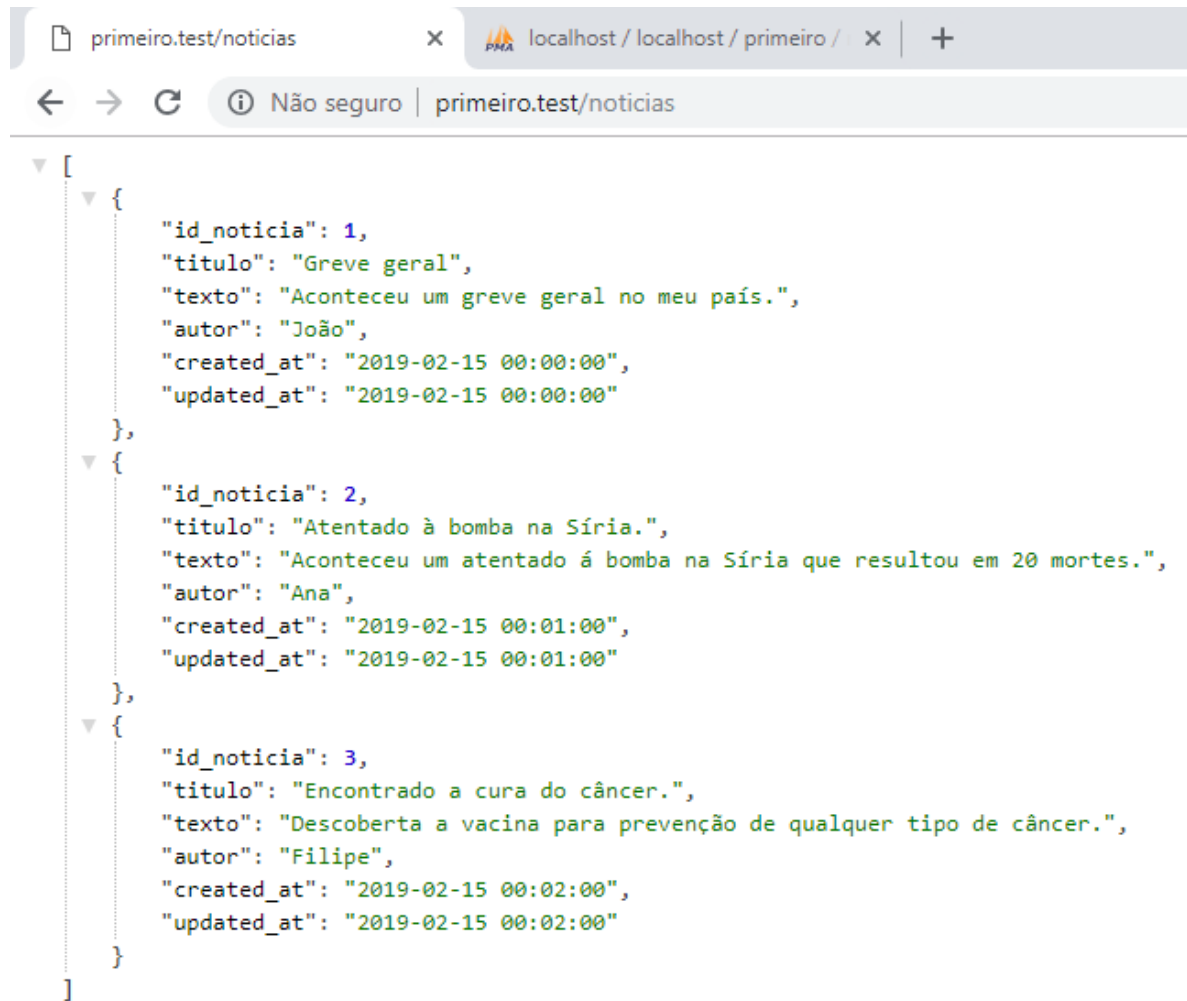
```
    {
```

```
        //
```

```
    }
```

```
}
```

Visualizando as notícias



The screenshot shows a web browser window with the address bar displaying "primeiro.test/noticias". The page content is a JSON array of three news items, each with fields for id, title, text, author, creation time, and update time. The first item is about a general strike, the second is about a bombing in Syria, and the third is about a cancer cure discovery.

```
[
  {
    "id_noticia": 1,
    "titulo": "Greve geral",
    "texto": "Aconteceu um greve geral no meu país.",
    "autor": "João",
    "created_at": "2019-02-15 00:00:00",
    "updated_at": "2019-02-15 00:00:00"
  },
  {
    "id_noticia": 2,
    "titulo": "Atentado à bomba na Síria.",
    "texto": "Aconteceu um atentado á bomba na Síria que resultou em 20 mortes.",
    "autor": "Ana",
    "created_at": "2019-02-15 00:01:00",
    "updated_at": "2019-02-15 00:01:00"
  },
  {
    "id_noticia": 3,
    "titulo": "Encontrado a cura do câncer.",
    "texto": "Descoberta a vacina para prevenção de qualquer tipo de câncer.",
    "autor": "Filipe",
    "created_at": "2019-02-15 00:02:00",
    "updated_at": "2019-02-15 00:02:00"
  }
]
```



```
primeiro.test/noticias/1 x localhost / localhost / primeiro / x +
← → ↻ ⓘ Não seguro | primeiro.test/noticias/1
[
  {
    "id_noticia": 1,
    "titulo": "Greve geral",
    "texto": "Aconteceu um greve geral no meu país.",
    "autor": "João",
    "created_at": "2019-02-15 00:00:00",
    "updated_at": "2019-02-15 00:00:00"
  }
]
```

```
primeiro.test/noticias/2 x localhost / localhost / primeiro / x +
← → ↻ ⓘ Não seguro | primeiro.test/noticias/2
[
  {
    "id_noticia": 2,
    "titulo": "Atentado à bomba na Síria.",
    "texto": "Aconteceu um atentado á bomba na Síria que resultou em 20 mortes.",
    "autor": "Ana",
    "created_at": "2019-02-15 00:01:00",
    "updated_at": "2019-02-15 00:01:00"
  }
]
```

```
primeiro.test/noticias/3 x localhost / localhost / primeiro / x +
← → ↻ ⓘ Não seguro | primeiro.test/noticias/3
[
  {
    "id_noticia": 3,
    "titulo": "Encontrado a cura do câncer.",
    "texto": "Descoberta a vacina para prevenção de qualquer tipo de câncer.",
    "autor": "Filipe",
    "created_at": "2019-02-15 00:02:00",
    "updated_at": "2019-02-15 00:02:00"
  }
]
```

Criar novo registro a partir de formulário

app/Http/NoticiasController.php

```
<?php

namespace App\Http\Controllers\aula45;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\noticias;

class noticiasController extends Controller
{
    public function index()
    {
        $dados = noticias::all();
        return view('aula47/noticias_show', compact('dados'));
    }
    public function create()
    {
        return view('aula47/noticias_new');
    }
    public function store(Request $request)
    {
        $dados = [
            $request->text_titulo,
            $request->text_texto,
            $request->text_autor
        ];
        return $dados;
    }
    public function show($id)
    {
        $dados = noticias::where('id_noticia', $id)->get();
        return $dados;
    }
    public function edit($id)
    {
        //
    }
    public function update(Request $request, $id)
    {
        //
    }
}
```

```

    public function destroy($id)
    {
        //
    }
}

```

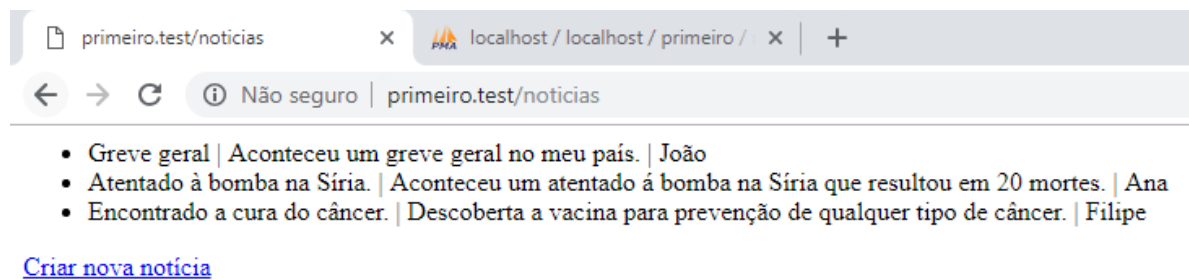
resources/views/aula47/noticias_show.blade.php

```

<ul>
    @foreach ($dados as $noticia)
        <li>{{ $noticia->titulo }} | {{ $noticia->texto }} | {{ $noticia->autor }}</li>
    @endforeach
</ul>

<div>
    <a href="noticias/create">Criar nova notícia</a>
</div>

```



resources/views/aula47/noticias_new.blade.php

```

<form method="POST" action="/noticias">

    {{csrf_field()}}

    <label>Título: </label>
    <input type="text" name="text_titulo">
    <br>
    <label>Texto: </label>
    <input type="text" name="text_texto">
    <br>
    <label>Autor: </label>
    <input type="text" name="text_autor">
    <br>
    <input type="submit" value="Gravar">
</form>

```

The first screenshot shows a web browser at the URL `primeiro.test/noticias/create`. It contains a form with three input fields: 'Título:' with the value 'Ataque na Coreia.', 'Texto:' with the value 'Aconteceu um ataque na Co', and 'Autor:' with the value 'João'. Below the fields is a 'Gravar' button.

The second screenshot shows the browser at the URL `primeiro.test/noticias`. It displays a JSON array representing the saved news item:

```
[
  [
    "Ataque na Coreia.",
    "Aconteceu um ataque na Coreia.",
    "João"
  ]
]
```

Gravando o registro no banco de dados

app/Http/NoticiasController.php

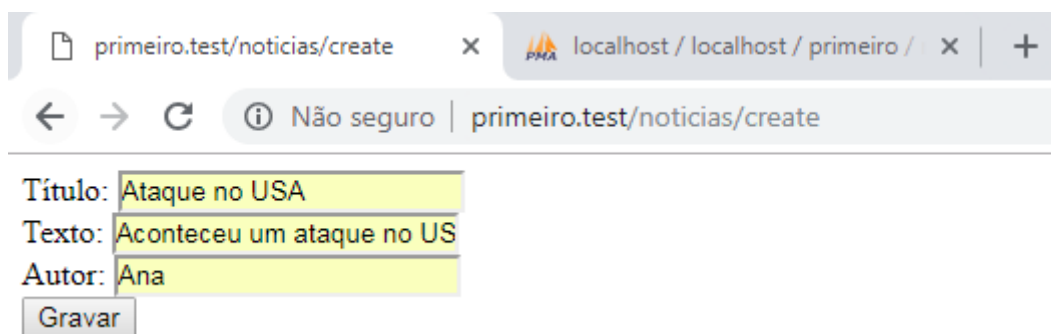
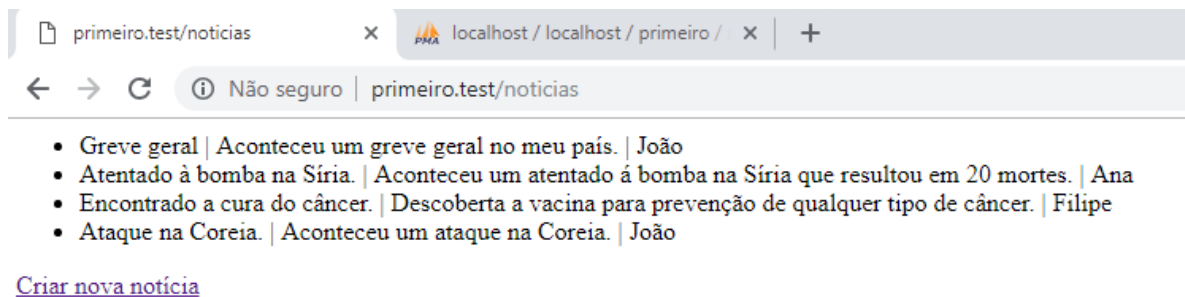
```
public function store(Request $request)
{
    // $dados = [
    //     $request->text_titulo,
    //     $request->text_texto,
    //     $request->text_autor
    // ];

    // return $dados;

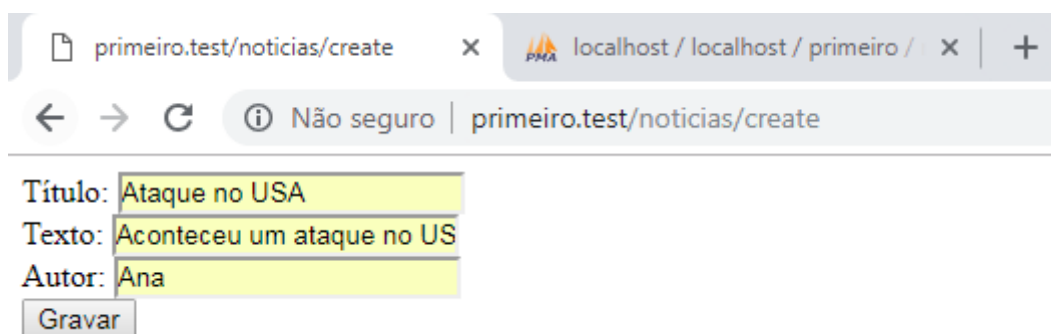
    $noticia = new noticias;

    $noticia->titulo = $request->text_titulo;
    $noticia->texto = $request->text_texto;
    $noticia->autor = $request->text_autor;
    $noticia->save();

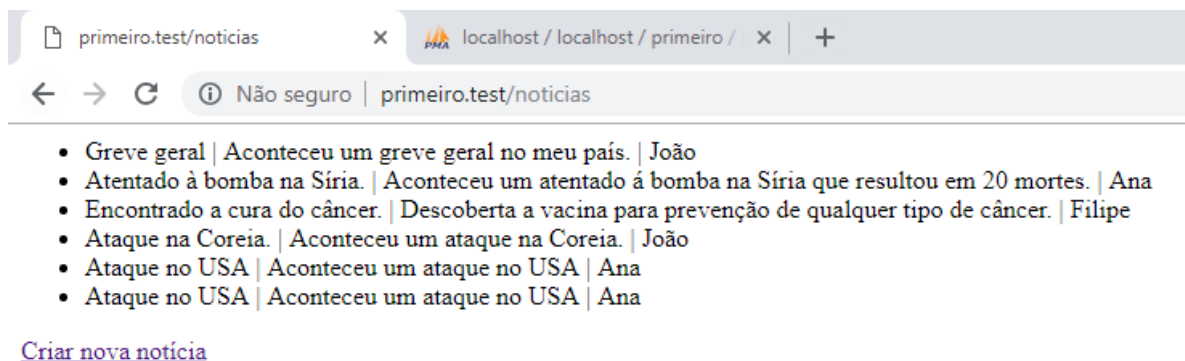
    return redirect('/noticias');
}
```



Clique no botão "Gravar"



Clique no botão "Gravar"



Aula 048 – Adaptar o Model para apresentar registros

app/Http/NoticiasController.php

```
public function show($id)
{
    // $dados = noticias::where('id_noticia', $id)->get();
    $dados = noticias::find($id);

    return $dados;
}
```

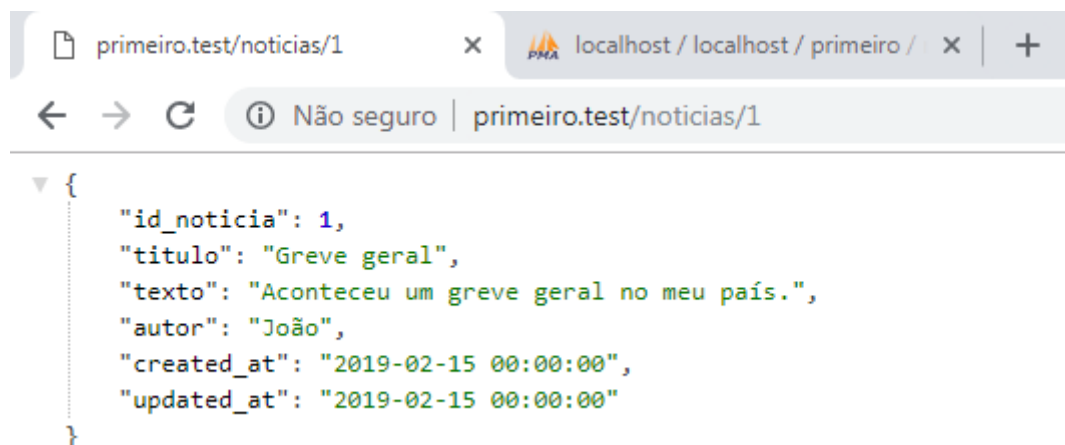
app/noticias.php (model)

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class noticias extends Model
{
    protected $primaryKey = 'id_noticia';
}
```



Aula 049 – Eloquent - Editar e eliminar registros

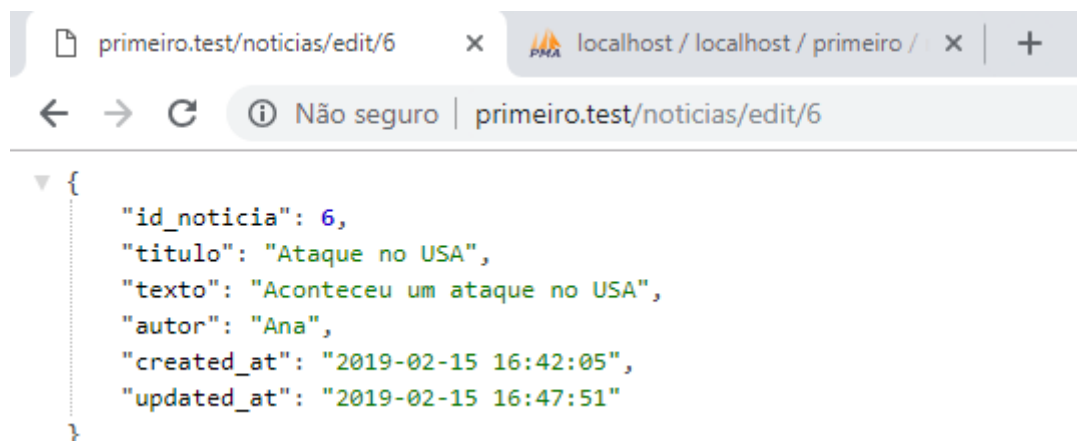
routes/web

```
Route::get('noticias/edit/{id}', 'NoticiasController@edit');
Route::get('noticias/update/{id}', 'NoticiasController@update');
Route::get('noticias/delete/{id}', 'NoticiasController@destroy');
```

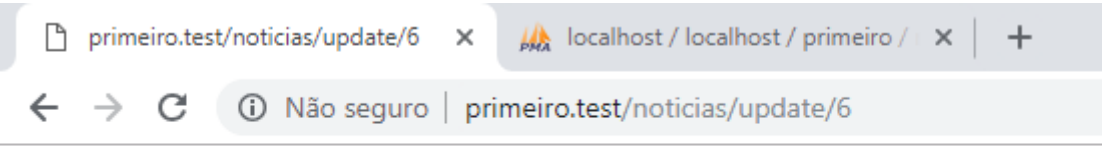
app/Http/NoticiasController.php

```
public function edit($id)
{
    return noticias::find($id);
    // return "Editar {$id}";
}
public function update(Request $request, $id)
{
    $noticia = noticias::find($id);
    $noticia->titulo = "Não aconteceu um ataque";
    $noticia->save();
    // return "Update {$id}";
}
public function destroy($id)
{
    $noticia = noticias::find($id);
    $noticia->delete();
    // return "Apagar {$id}";
}
```

Editando registro

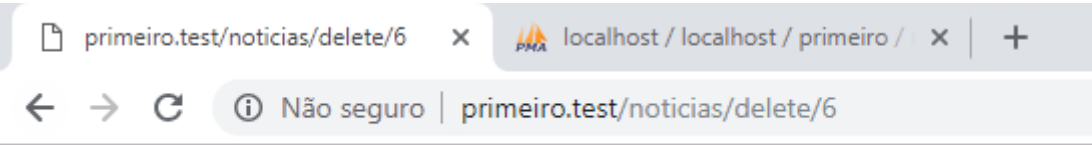


Atualizando registro



| + Options | | | | | | | |
|--------------------------|------------------|------------|------------------------------|---------------------------------------------------------|--------|---------------------|---------------------|
| ← T → | | | | | | | |
| | | id_noticia | titulo | texto | autor | created_at | updated_at |
| <input type="checkbox"/> | Edit Copy Delete | 1 | Greve geral | Aconteceu um greve geral no meu país. | João | 2019-02-15 00:00:00 | 2019-02-15 00:00:00 |
| <input type="checkbox"/> | Edit Copy Delete | 2 | Atentado à bomba na Síria. | Aconteceu um atentado à bomba na Síria que resultado... | Ana | 2019-02-15 00:01:00 | 2019-02-15 00:01:00 |
| <input type="checkbox"/> | Edit Copy Delete | 3 | Encontrado a cura do câncer. | Descoberta a vacina para prevenção de qualquer tip... | Filipe | 2019-02-15 00:02:00 | 2019-02-15 00:02:00 |
| <input type="checkbox"/> | Edit Copy Delete | 4 | Ataque na Coreia. | Aconteceu um ataque na Coreia. | João | 2019-02-15 06:13:54 | 2019-02-15 06:13:54 |
| <input type="checkbox"/> | Edit Copy Delete | 5 | Ataque no USA | Aconteceu um ataque no USA | Ana | 2019-02-15 16:41:36 | 2019-02-15 16:41:36 |
| <input type="checkbox"/> | Edit Copy Delete | 6 | Não aconteceu um ataque | Aconteceu um ataque no USA | Ana | 2019-02-15 16:42:05 | 2019-02-15 16:56:12 |

Deletando registro



✓ Showing rows 0 - 4 (5 total, Query took 0.0010 seconds.)

SELECT * FROM `noticias`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Ref

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

| + Options | | | | | | | |
|--------------------------|------------------|------------|------------------------------|---------------------------------------------------------|--------|---------------------|---------------------|
| ← T → | | | | | | | |
| | | id_noticia | titulo | texto | autor | created_at | updated_at |
| <input type="checkbox"/> | Edit Copy Delete | 1 | Greve geral | Aconteceu um greve geral no meu país. | João | 2019-02-15 00:00:00 | 2019-02-15 00:00:00 |
| <input type="checkbox"/> | Edit Copy Delete | 2 | Atentado à bomba na Síria. | Aconteceu um atentado à bomba na Síria que resultado... | Ana | 2019-02-15 00:01:00 | 2019-02-15 00:01:00 |
| <input type="checkbox"/> | Edit Copy Delete | 3 | Encontrado a cura do câncer. | Descoberta a vacina para prevenção de qualquer tip... | Filipe | 2019-02-15 00:02:00 | 2019-02-15 00:02:00 |
| <input type="checkbox"/> | Edit Copy Delete | 4 | Ataque na Coreia. | Aconteceu um ataque na Coreia. | João | 2019-02-15 06:13:54 | 2019-02-15 06:13:54 |
| <input type="checkbox"/> | Edit Copy Delete | 5 | Ataque no USA | Aconteceu um ataque no USA | Ana | 2019-02-15 16:41:36 | 2019-02-15 16:41:36 |

Aula 050 – Utilização do bootstrap no Laravel

routes/web.php

```
Route::get('aula50/teste', function(){  
    return view('aula50/teste');  
});
```

resources/views/aula50/teste.blade.php

```
@extends('layouts.app');
```

```
@section('conteudo')
```

```
<h1>Título</h1>
```

```
<p>Este é o texto de um parágrafo</p>
```

```
<button class="btn btn-primary" type="submit">Button</button>
```

```
@endsection
```

```
@section('conteudo')
```

```
@endsection
```

```
@section('conteudo')
```

```
@endsection
```

Baixar bootstrap e:

- copiar para pasta public/css o arquivo bootstrap.min.css
- copiar para pasta public/js o arquivo bootstrap.min.js
- copiar para pasta public a pasta fonts

resources/views/layouts/app.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- As 3 meta tags acima *devem* vir em primeiro lugar dentro do `head`; qualquer outro
conteúdo deve vir *após* essas tags -->
    <title>Teste</title>

    <!-- Bootstrap -->
    <link href="/css/bootstrap.min.css" rel="stylesheet">
    <!-- <link href="/css/app.css" rel="stylesheet"> -->

    <!-- HTML5 shim e Respond.js para suporte no IE8 de elementos HTML5 e media queries --
>
    <!-- ALERTA: Respond.js não funciona se você visualizar uma página file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>

    <div class="container">
      @yield('navegacao')

      @yield('conteudo')

      @yield('rodape')
    </div>

    <!-- jQuery (obrigatório para plugins JavaScript do Bootstrap) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
    <!-- Inclui todos os plugins compilados (abaixo), ou inclua arquivos separados se
necessário -->
    <script src="js/bootstrap.min.js"></script>
    <!-- <script src="/js/app.js"></script> -->
  </body>
</html>
```

