



INSTITUTO FEDERAL
CATARINENSE
Campus Camboíu

Laravel para Noobs

Da Instalação à Autenticação de Usuários

Instrutor: Marcelo Ramos Machado

e-Tic 2018



INSTALAÇÃO

```
Administrador: Prompt de Comando - composer create-project laravel/laravel modelo
C:\xampp\htdocs\bsi17\2017-2\etic>composer create-project laravel/laravel modelo
[32mInstalling laravel/laravel (v5.5.0)[39m
- Installing [32mlaravel/laravel[39m ([33mv5.5.0[39m)
  Loading from cache

[32mCreated project in modelo[39m
[30;43mYou made a reference to a non-existent script @php -r "file_exists('.env') || cop
y('.env.example', '.env');"[39;49m
[32mLoading composer repositories with package information[39m
[32mUpdating dependencies (including require-dev)[39m
-
```

- Por linha de comando
- Copiando os arquivos





INSTALAÇÃO

C:\xampp\htdocs>

composer create-project --prefer-dist laravel/laravel meuprojeto





INSTITUTO FEDERAL
CATARINENSE
Campus Camboiú

ARQUIVOS

<https://github.com/mach3k>





RESUMO DA OFICINA



- Requisitos e instalação
- Apresentação da ferramenta
- Configurações
- Acesso a dados, migrações e seeders
- Controllers, models e rotas
- Views e arquivos Blade
- Autenticação de usuários

- Bônus (se der tempo):
- Instalação de plugin e configuração de alias





INSTITUTO FEDERAL
CATARINENSE
Campus Camboriú

INSTRUTOR



- Tecnólogo em Sistemas para Internet – IFC Camboriú
- Estudante de Bacharelado em Sistemas de Informação – IFC Camboriú
- Técnico em Informática e Administração de Empresas
- +15 anos em tecnologia (automação industrial e comercial)





REQUISITOS

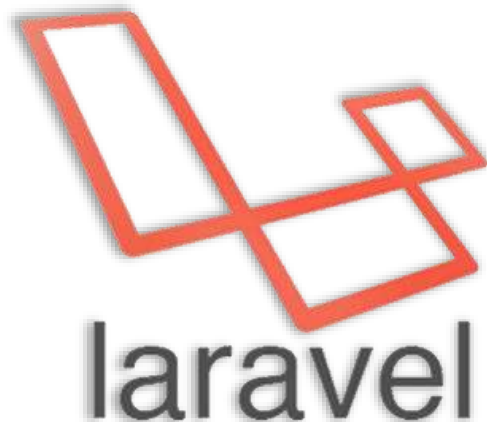


- Composer (<https://getcomposer.org/download/>)
- Servidor Web (Apache, IIS) com PHP
- SGBD (MySQL, MariaDB, SQLite, PostgreSQL, SQL Server)
- IDE (Brackets, VS Code, ~~Bloco de Notas~~, etc)
- Navegador (Chrome, Firefox, Opera, ~~Edge~~, etc)
- XAMPP
(https://www.apachefriends.org/pt_br/download.html)





A FERRAMENTA

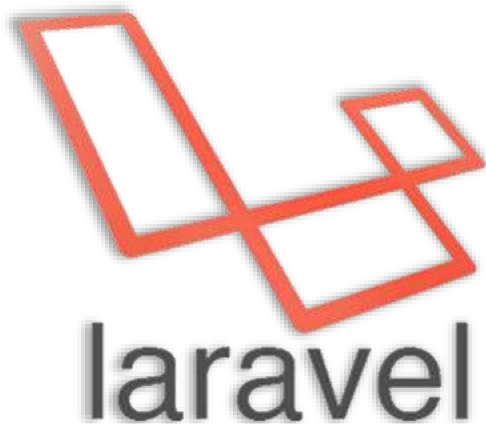


- Framework PHP open-source
- Criado em 2011 por Taylor B. Otwell
- Utilizado para facilitar o desenvolvimento de sistemas web
- Versão atual: 5.6
- Estruturado em MVC
- Composer para administração de dependências
- Artisan para automatizar tarefas





MINHA EXPERIÊNCIA

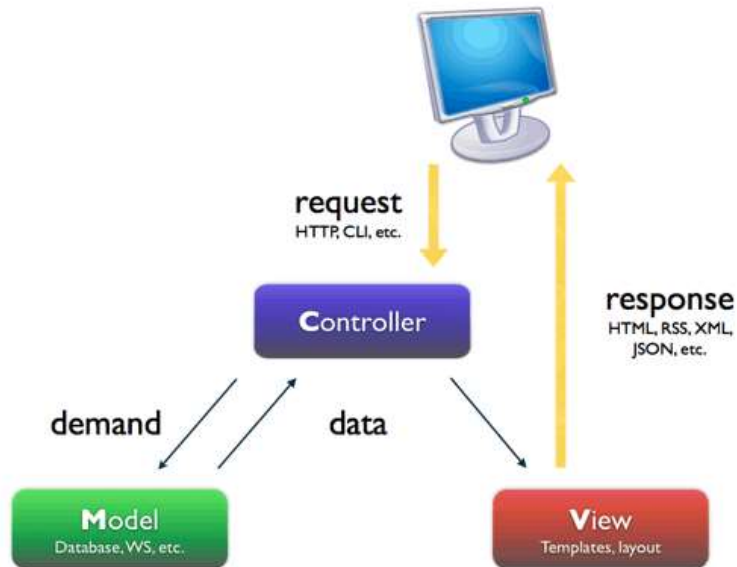


- Projeto Integrador, conclusão TSI
(aprendizado e desenvolvimento em menos de 1 mês)
- Sistema de vendas para construtora
- Sistema de cadastro e divulgação de empresas
- Sistema de controle interno para previdência do município de Camboriú (em desenvolvimento)





O CONCEITO MVC



- Padrão de arquitetura de software
- Divide o programa em camada de dados e regras de negócios (model), camada de apresentação (view) e controladores de fluxo (controllers)





CONFIGURAÇÃO

Para criar o arquivo “.env” execute este comando:

```
C:\...\meuprojeto>
```

```
copy .env.example .env
```





CONFIGURAÇÃO

Para gerar a APP_KEY, no arquivo “.env”, execute este comando:

C:\...\meuprojeto>

php artisan key:generate





CONFIGURAÇÕES DA APLICAÇÃO

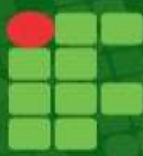
```
1 <?php
2
3 return [
4
5     /*
6      * Application Name
7      *
8      * This value is the name of your application. This value is used when the
9      * framework needs to place the application's name in a notification or
10     * any other location as required by the application or its packages.
11     */
12
13     'name' => env('APP_NAME', 'Meu projeto'),
14
15     /*
16      * Application Environment
17      *
18      * This value determines the "environment" your application is currently
19      * running in. This may determine how you prefer to configure various
20      * services your application utilizes. Set this in your ".env" file.
21      */
22
23     'env' => env('APP_ENV', 'production'),
24
25     /*
26      * Application Debug Mode
27      *
28      * Whether your application is in debug mode. During debug mode, stack
29      * traces are enabled and many other features like "dd()" and "dump()" are
30      * available. In production, these are disabled.
```

Pasta: config/

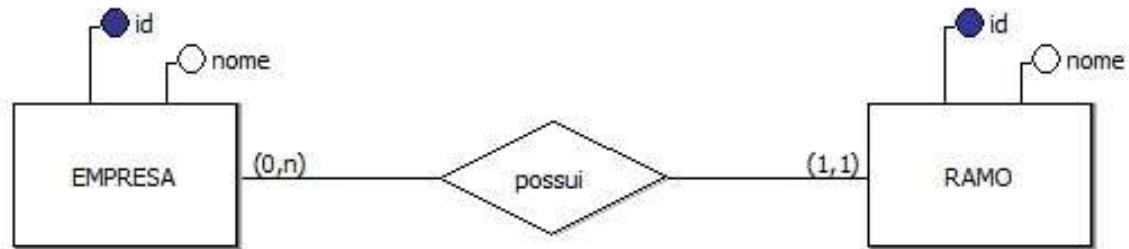
Arquivo: app.config

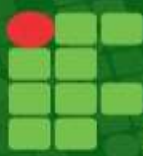
- Nome da aplicação
- Linguagem
- Providers
- Aliases





TABELAS DO BANCO



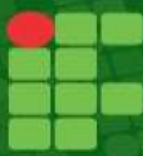


CRIANDO O BANCO DE DADOS



- Abrir no phpMyAdmin
- <http://localhost/phpmyadmin>
- Criar no banco com o nome “meuprojeto”





ACESSANDO O BANCO DE DADOS

```
23  // Uncomment this line to use a database other than SQLite. You must have the
24  // choice installed on your machine before you begin development.
25  //
26  // 'connections' => [
27  //     'sqlite' => [
28  //         'driver' => 'sqlite',
29  //         'database' => env('DB_DATABASE', database_path('database.sqlite')),
30  //         'prefix' => '',
31  //     ],
32  //     'mysql' => [
33  //         'driver' => 'mysql',
34  //         'host' => env('DB_HOST', '127.0.0.1'),
35  //         'port' => env('DB_PORT', '3306'),
36  //         'database' => env('DB_DATABASE', 'root'),
37  //         'username' => env('DB_USERNAME', 'root'),
38  //         'password' => env('DB_PASSWORD', ''),
39  //         'unix_socket' => env('DB_SOCKET', ''),
40  //         'charset' => 'utf8mb4',
41  //         'collation' => 'utf8mb4_unicode_ci',
42  //         'prefix' => '',
43  //         'strict' => true,
44  //         'engine' => null,
45  //     ],
46  //     'pgsql' => [
47  //         'driver' => 'pgsql',
48  //         'host' => env('DB_HOST', '127.0.0.1'),
49  //         'port' => env('DB_PORT', '5432'),
50  //         'database' => env('DB_DATABASE', 'root'),
51  //         'username' => env('DB_USERNAME', 'root'),
52  //         'password' => env('DB_PASSWORD', ''),
53  //         'charset' => 'utf8',
54  //         'collation' => 'utf8_unicode_ci',
55  //         'prefix' => '',
56  //         'strict' => true,
57  //         'engine' => null,
58  //     ],
59  // ],
```

Pasta: config/

Arquivo: database.config

- Informar o banco de dados criado
- Usuário e senha





ACESSANDO O BANCO DE DADOS

```
1 APP_NAME='Meu Projeto'
2 APP_ENV=local
3 APP_KEY=base64:zVZnsBOS/XpVXmfa2z87Sk0IEFQeJ9KcZ1HaIENtEGk=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=127.0.0.1
11 DB_PORT=3306
12 DB_DATABASE=meuprojeto
13 DB_USERNAME=root
14 DB_PASSWORD=
15
```

Pasta: \

Arquivo: .env

- Informar o nome do projeto
- Banco de dados criado
- Usuário e senha





MIGRAÇÕES

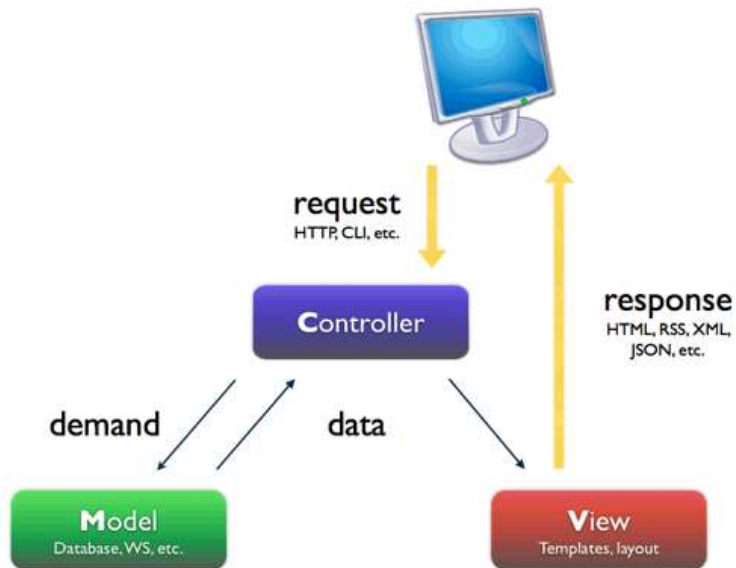
```
1 <?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class CreateRamoTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('ramos', function (Blueprint $table) {
17             $table->increments('id');
18             $table->string('nome');
19             $table->timestamps();
20         });
21     }
22
23     /**
24      * Reverse the migrations.
25      *
26      * @return void
27      */
28     public function down()
29     {
30         Schema::dropIfExists('ramos');
31     }
32 }
```

- Scripts que criam, alteram ou excluem tabelas do banco de dados
- Criadas e executadas por linha de comando





O CONCEITO MVC





MIGRAÇÕES

Para criar:

C:\...\meuprojeto>

```
php artisan make:migration create_ramos_table --create=ramos
```





MIGRAÇÕES

Para criar:

C:\...\meuprojeto>

```
php artisan make:migration create_empresas_table --create=empresas
```





MIGRAÇÕES

Você encontra suas migrations na pasta
database\migrations

*duas migrations são “nativas”:
create_users_table e **create_password_resets_table**





MIGRAÇÕES

```
7 class CreateRamosTable extends Migration
8 {
9     /* ... */
14    public function up()
15    {
16        Schema::create('ramos', function (Blueprint $table) {
17            $table->increments('id');
18            $table->string('nome');
19            $table->timestamps();
20        });
21    }
22
23    /* ... */
28    public function down()
29    {
30        Schema::dropIfExists('ramos');
31    }
32 }
33
```

```
7 class CreateEmpresasTable extends Migration
8 {
9     /* ... */
14    public function up()
15    {
16        Schema::create('empresas', function (Blueprint $table) {
17            $table->increments('id');
18            $table->string('nome');
19            $table->integer('ramo_id')->unsigned()->default(0);
20            $table->foreign('ramo_id')->
21                references('id')->on('ramos')->
22                onDelete('cascade');
23            $table->timestamps();
24        });
25    }
26
27    /* ... */
32    public function down()
33    {
34        Schema::dropIfExists('empresas');
35    }
36 }
37
```





MIGRAÇÕES

Para executar:

C:\...\meuprojeto>

php **artisan** **migrate**





MIGRAÇÕES

Caso ocorra o erro

“SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was too long”,

incluir em **App/Providers/AppServiceProvider:**

```
use Illuminate\Support\Facades\Schema;  
public function boot() {  
    Schema::defaultStringLength(191);  
}
```





SEEDERS

```
1 <?php
2
3 use Illuminate\Database\Seeder;
4
5 class UsersTableSeeder extends Seeder
6 {
7     /**
8      * Run the database seeds.
9      *
10     * @return void
11     */
12     public function run()
13     {
14         $user = new \App\User();
15         $user->name = 'Marcelo Ramos Machado';
16         $user->email = 'mr.machado@gmail.com';
17         $user->password = bcrypt('123456');
18         $user->permissao = 1;
19         $user->cargo = 'Programador';
20         $user->foto = 'marcelo.png';
21         $user->save();
22     }
23 }
```

- Preenchem as tabelas do banco de dados com informações de fábrica
- Útil para testar recursos





SEEDERS

Para criar:

C:\...\meuprojeto>

php artisan make:seeder RamosTableSeeder

Obs.: é necessário criar a chamada na classe **DataBaseSeeder**





SEEDERS

Para economizar tempo, copie os arquivos da pasta
database\seeds para seu projeto

Mais tarde vamos utilizar estas classes...





MODELS

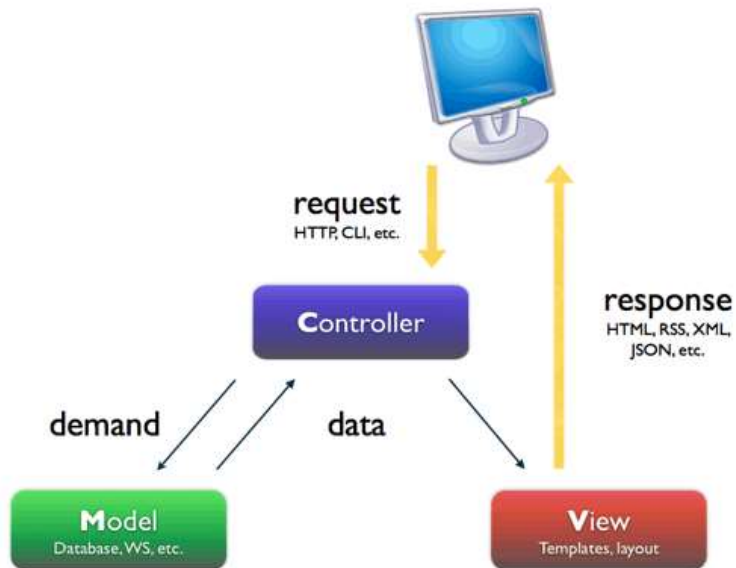
```
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Ramo extends Model
8  {
9      public function empresas()
10     {
11         return $this->hasMany('App\Empresa');
12     }
13 }
14
```

- Eloquent ORM
- Provê métodos de acesso aos dados
- Nome da classe no singular >> refere-se à uma tabela com nome no plural
- Ex: Empresa (model) >> empresas (BD)





O CONCEITO MVC





MODELS

Para criar um model:

```
C:\...\meuprojeto>
```

```
php artisan make:model Empresa
```

ou

```
C:\...\meuprojeto> php artisan make:model Empresa --migration  
(para criar a migration junto)
```





MODELS

Para criar um model:

C:\...\meuprojeto>

php artisan make:model Ramo

ou

C:\...\meuprojeto> php artisan make:model Ramo --migration
(para criar a migration junto)





MODELS

- Criar os métodos de acesso aos relacionamentos

```
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Empresa extends Model
8  {
9      public function ramo()
10     {
11         return $this->belongsTo('App\Ramo');
12     }
13 }
14
```

```
1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Ramo extends Model
8  {
9      public function empresas()
10     {
11         return $this->hasMany('App\Empresa');
12     }
13 }
14
```





EXECUTANDO OS SEEDERS

Para executar:

composer dump-autoload

C:\...\meuprojeto>

php artisan db:seed

ou

C:\...\meuprojeto> php artisan db:seed --class=UsersTableSeeder

ou

C:\...\meuprojeto> php artisan migrate:refresh --seed





CONTROLLERS

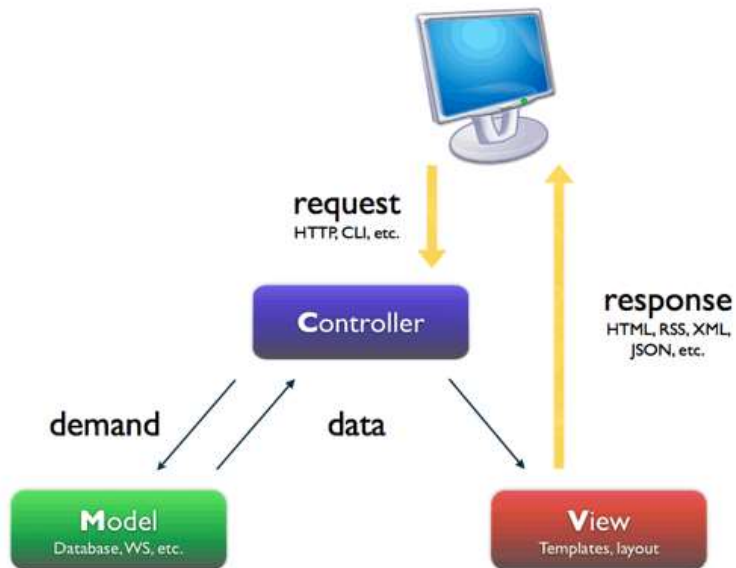
```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Cliente;
7
8 class ClienteController extends Controller
9 {
10     /**
11      *
12      */
13     public function __construct()
14     {
15         $this->middleware('auth');
16     }
17
18     /**
19      *
20      */
21     public function index()
22     {
23         $registros = Cliente::all();
24         $title = 'Clientes';
25         return view('cliente.list', ['registros' => $registros])->withTitle($title);
26     }
27
28     /**
29      *
30      */
31     public function create()
32     {
33         $title = 'Novo cliente';
34         return view('cliente.create')->withTitle($title);
35     }
36
37     /**
38      *
39      */
40     public function store(Request $request)
41     {
42         $this->validate($request, ['nomeFantasia' => 'required',
43                                 'telefone' => 'required',
44                                 ]);
45     }
46 }
```

- Classes que controlam o fluxo do programa
- Comando muito útil: **dd(\$objeto);**
- ***Dump and Die:** exibe o conteúdo do objeto





O CONCEITO MVC





CONTROLLERS

Para criar um controller:

C:\...\meuprojeto>

php artisan make:controller EmpresaController --resource

ou

C:\...\meuprojeto> php artisan make:controller EmpresaController





CONTROLLERS

Você encontra os controllers que criou na pasta
app\Http\Controllers

Para economizar tempo, de novo, copie o controller **EmpresaController**
dos arquivos de exemplo para seu projeto





CONFIGURAÇÃO DE ROTAS

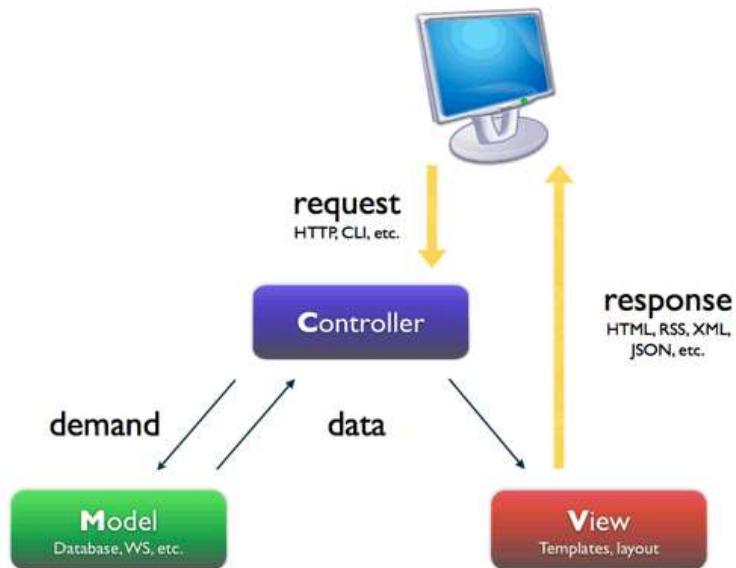
```
1  <?php
2
3  /*
4  |-----
5  | Web Routes
6  |-----
7  |
8  | Here is where you can register web routes for your application. These
9  | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 | */
13
14 Route::get('/', function () {
15     return view('welcome');
16 });
17
```

- Recebe o endereço requisitado e traduz em um método de um controlador
- Web
- API
- Channels
- Console





O CONCEITO MVC





CONFIGURAÇÃO DE ROTAS

```
14 Auth::routes();
15
16 Route::get('/', 'HomeController@index');
17
18 Route::get('/home', ['as' => 'home', 'uses' => 'HomeController@index']);
19
20 Route::get('/perfil/{idUsuario}', ['as' => 'perfil', 'uses' => 'UserController@perfil'])
21     ->where(['idUsuario' => '[0-9]+']);
22
23 Route::get('/projeto/{idProjeto}', ['as' => 'showProjeto', 'uses' => 'ProjetoController@showPro'])
24     ->where(['idProjeto' => '[0-9]+']);
25
26 Route::get('/projeto/{idProjeto}/etapa/{idEtapa}', ['as' => 'showEtapa', 'uses' => 'EtapaControl'])
27     ->where(['idProjeto' => '[0-9]+' , 'idEtapa' => '[0-9]+']);
28
29 Route::get('/projeto/{idProjeto}/etapa/{idEtapa}/tarefa/{idTarefa}', ['as' => 'showTarefa', 'use'])
30     ->where(['idProjeto' => '[0-9]+' , 'idEtapa' => '[0-9]+' , 'idTarefa' => '[0-9]+']);
31
32 Route::resource('comentarios', 'ComentarioController');
33
34 Route::get('/ultimoscomentarios', ['as' => 'ultimoscomentarios', 'uses' =>
35     'ComentarioController@ultimosComentarios']);
36
37 Route::get('/ultimosdocumentos', ['as' => 'ultimosdocumentos', 'uses' =>
38     'DocumentoController@ultimosDocumentos']);
39
40 + Route::group(['middleware' => ['web', 'FiltroTecnico']], function(){
41
42     Route::get('/projetosparticipo', ['as' => 'projetosparticipo', 'uses' =>
43         'ProjetoController@projetosqueParticipo']);
44
45     Route::get('/minhasTarefas', ['as' => 'minhasTarefas', 'uses' => 'TarefaController@minhasTare'])
46
47     Route::get('/tarefa/{idTarefa}/atendimento', ['as' => 'atendimento', 'uses' =>
48         'TarefaController@atendimento'])
49 }
```

- Rotas Get
- Resource
- Agrupadas
- Autenticadas





CONFIGURAÇÃO DE ROTAS

Incluir a seguinte rota, abaixo da entrada existente:

```
Route::resource('empresa', 'EmpresaController');
```





VIEWS E ARQUIVOS BLADE

```
resources/views/home.blade.php (project_tracking) - Brackets

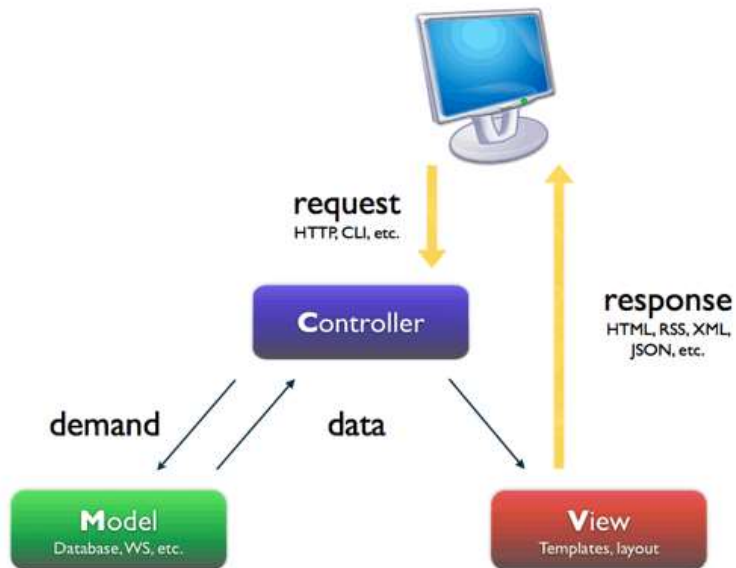
1  @extends('app')
2
3  @section('title')
4  {{$title}}
5  @endsection
6
7  @section('content')
8  <section class="content-header">
9    <h1>
10      Sistema de Acompanhamento
11      <small>de Projetos</small>
12    </h1>
13  </section>
14  <section class="content">
15    <div class="row">
16      <div class="col-lg-3 col-xs-6">
17        <!-- small box -->
18        <div class="small-box bg-aqua">
19          <div class="inner">
20            <h3>{{ $projetosAtivos }}</h3>
21
```

- Arquivos que montam a interface gráfica
- Modulares e com linguagem própria, mas de simples aprendizado
- Permitem a criação de componentes
- Não acessa diretamente o banco de dados





O CONCEITO MVC





VIEWS E ARQUIVOS BLADE

Você encontra as views na pasta
resources\views

Para economizar tempo, pela última vez, copie a pasta
resources\views\Empresas
dos arquivos de exemplo para seu projeto





AUTENTICAÇÃO



- O Artisan já oferece um comando para ajudar na criação do módulo de autenticação
- *Na verdade ele faz tudo sozinho...





AUTENTICAÇÃO

Para gerar magicamente todo o sistema de autenticação:

```
C:\...\meuprojeto>
```

```
php artisan make:auth
```





CONFIGURAÇÃO DE ROTAS

```
1  <?php
2
3  ▶ /* ... */
13
14  ▼ Route::get('/', function () {
15      return view('welcome');
16  });
17
18  Auth::routes();
19
20  Route::get('/home', 'HomeController@index')->name('home');
21  Route::resource('empresa', 'EmpresaController');
22  |
```





TESTE

Acesse <http://localhost/.../public>

Informe usuário e senha

Acesse <http://localhost/.../public/empresa>





INSTITUTO FEDERAL
CATARINENSE
Campus Camboiá

INSTALANDO UM PLUGIN



Para instalar:

```
C:\...\meuprojeto> composer require nesbot/carbon
```





CONFIGURANDO O ALIAS

```
193 * 'aliases' => [  
194     'App' => Illuminate\Support\Facades\App::class,  
195     'Artisan' => Illuminate\Support\Facades\Artisan::class,  
196     'Auth' => Illuminate\Support\Facades\Auth::class,  
197     'Blade' => Illuminate\Support\Facades\Blade::class,  
198     'Broadcast' => Illuminate\Support\Facades\Broadcast::class,  
199     'Bus' => Illuminate\Support\Facades\Bus::class,  
200     'Cache' => Illuminate\Support\Facades\Cache::class,  
201     'Config' => Illuminate\Support\Facades\Config::class,  
202     'Cookie' => Illuminate\Support\Facades\Cookie::class,  
203     'Crypt' => Illuminate\Support\Facades\Crypt::class,  
204     'DB' => Illuminate\Support\Facades\DB::class,  
205     'Eloquent' => Illuminate\Database\Eloquent\Model::class,  
206     'Event' => Illuminate\Support\Facades\Event::class,  
207     'File' => Illuminate\Support\Facades\File::class,  
208     'Gate' => Illuminate\Support\Facades\Gate::class,  
209     'Hash' => Illuminate\Support\Facades\Hash::class,  
210     'Lang' => Illuminate\Support\Facades\Lang::class,  
211     'Log' => Illuminate\Support\Facades\Log::class,  
212     'Mail' => Illuminate\Support\Facades\Mail::class,  
213     'Notification' => Illuminate\Support\Facades\Notification::class,  
214     'Password' => Illuminate\Support\Facades>Password::class,  
215     'Queue' => Illuminate\Support\Facades\Queue::class,  
216     'Redirect' => Illuminate\Support\Facades\Redirect::class,  
217     'Redis' => Illuminate\Support\Facades\Redis::class,  
218     'Request' => Illuminate\Support\Facades\Request::class,  
219     'Response' => Illuminate\Support\Facades\Response::class,  
220     'Route' => Illuminate\Support\Facades\Route::class,  
221     'Schema' => Illuminate\Support\Facades\Schema::class,  
222     'Session' => Illuminate\Support\Facades\Session::class,  
223     'Storage' => Illuminate\Support\Facades\Storage::class,  
224     'URL' => Illuminate\Support\Facades\URL::class,  
225     'Validator' => Illuminate\Support\Facades\Validator::class,  
226     'View' => Illuminate\Support\Facades\View::class,  
227     'Carbon' => 'Carbon\Carbon',  
228 ]
```

- Em “Config/App”
- Adicionar: ‘Carbon’ => ‘Carbon/Carbon’,





LINKS

- Composer: <https://getcomposer.org/download/>
- XAMPP: https://www.apachefriends.org/pt_br/download.html
- Laravel: <https://laravel.com/>
- Canal do Fábio Vedovelli no YouTube:
<https://www.youtube.com/user/vedovelli/search?query=laravel>
- Documentação oficial do Laravel: <https://laravel.com/docs/5.5>
- Comandos Artisan: <http://laravel-docs-pt-br.readthedocs.io/en/5.0/artisan/>
- Customização Artisan: <http://laravel-docs-pt-br.readthedocs.io/en/5.0/commands/>
- Carbon: <https://github.com/briannesbitt/Carbon>





INSTITUTO FEDERAL
CATARINENSE
Campus Camboíu

Laravel para Noobs

Da Instalação à Autenticação de Usuários

Muito Obrigado!

Instrutor: Marcelo Ramos Machado

e-Tic 2018