

Iniciando com Laravel

DAYANE LIMA DE CASTRO CARMONIO

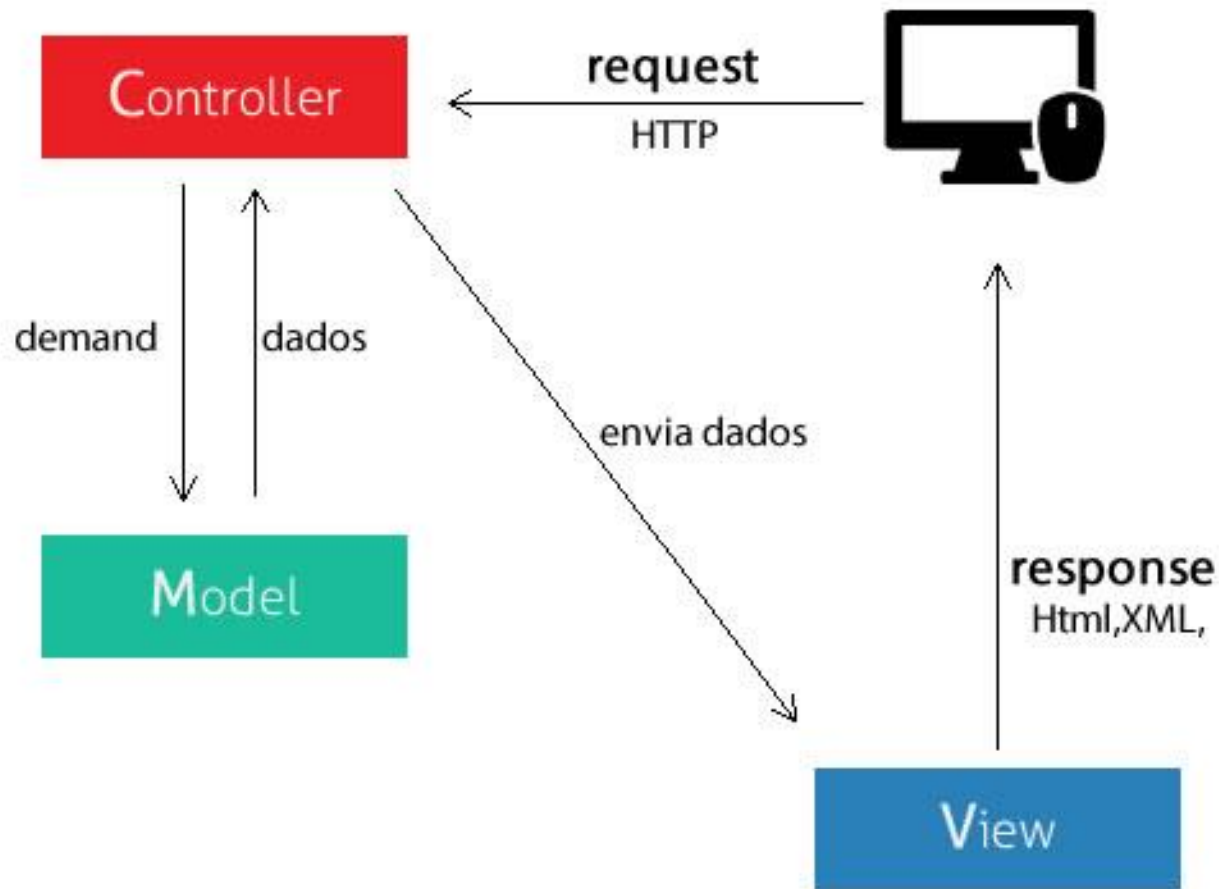
(dayane.lima.castro@gmail.com | dayane@symcode.com.br)

O que é o Laravel?

- Framework em PHP
- Padrão MVC
- Open Source
- Curva de aprendizagem baixa
- Atualmente está na versão 5.3



O que é o padrão MVC?



Model

- Entidade (representação dos dados)
- Manipulação dos dados
- Validação dos dados

View

- Interação com o usuário
- Exibição dos dados

Controller

- Recebe as requisições do usuário
- Funções chamadas de actions (responsável por páginas)
- Responsável por saber qual model usar e qual view chamar
- Intermediária entre a model e a view

Gerenciador de dependência



Gerenciados de dependencia

- Gerenciar bibliotecas e frameworks externos ao projeto
- Dependência da dependência (uma biblioteca que você baixou, depende de outra, ele já baixa!!!)

```
"require-dev": {  
    "phpunit/phpunit": "~5.4"  
}
```


Configuração do ambiente

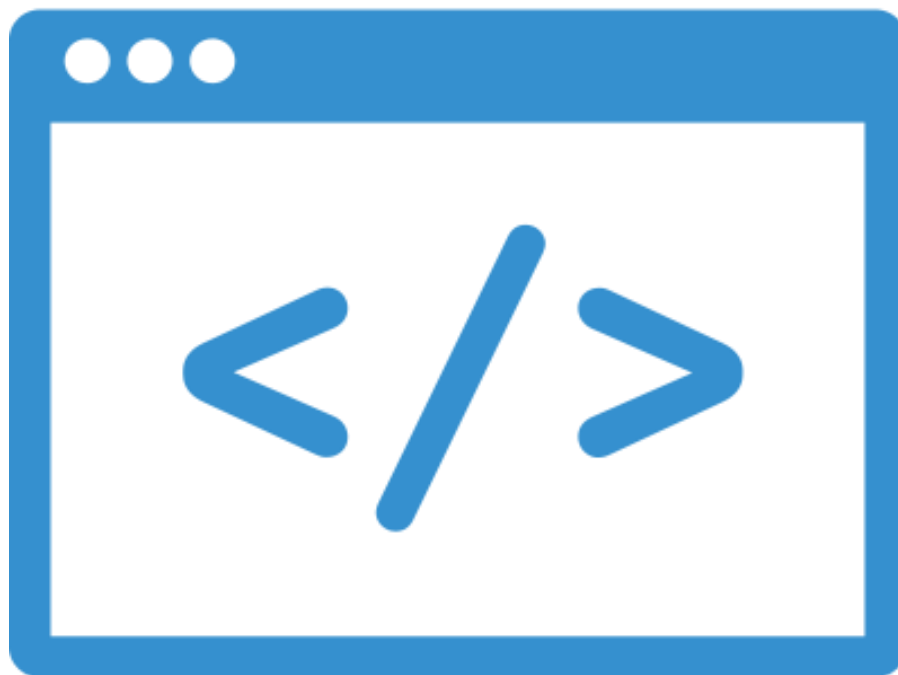
- Instalar o PHP ($\geq 5.6.4$) , Apache e MySql
- Xampp (https://www.apachefriends.org/pt_br/download.html)
- Instalar o composer (<https://getcomposer.org/download/>)
- Na instalação do composer: escolher onde está o php.exe
- IDE: Visual Studio Code (<https://code.visualstudio.com/download>)



Instalando o Laravel


- Via composer
 - `composer create-project --prefer-dist laravel/laravel blog`
- Via Laravel Installer:
 - `composer global require "laravel/installer"`
 - `laravel new blog`

Código!!!



Tutorial - Passo a Passo Instalação

- No console vá até a pasta que deseja criar o projeto
- Digite o seguinte comando: `laravel new minicursolaravel`

 Prompt de Comando

```
D:\programas\xampp\htdocs>laravel new minicursolaravel1
```

Tutorial - Passo a Passo Abrindo Projeto

Abra o Visual Studio Code

Clique em file -> open folder

Escolha a pasta do projeto e clique em abrir



EXPLORER

OPEN EDITORS

MINICURSOLARAVEL1

- app
- bootstrap
- config
- database
- public
- resources
- routes
- storage
- tests
- vendor
- .env
- .env.example
- .gitattributes
- .gitignore
- artisan
- composer.json
- composer.lock
- gulpfile.js
- package.json
- phpunit.xml
- readme.md
- server.php
- yarn.lock



Tutorial - Passo a Passo

Estruturação das Pastas

- app -> Http -> Controllers: Controllers do projeto
- app -> Http -> Middleware: Middlewares do projeto
- app -> Http -> kernel.php: Configuração das middlewares
- app: Models ficam na raiz do app, mas é bom criar uma pasta model dentro de app

Tutorial - Passo a Passo

Estruturação das Pastas

- config: Arquivos de configuração (ex.: email, banco de dados, sessão)
- public: O que pode ser acessado no seu sistema (js e css)
- resources -> view: Telas

Tutorial - Passo a Passo

Estruturação das Pastas

- routes: Configuração das rotas
- vendor: Framework e outras bibliotecas baixadas pelo composer
- .env: Arquivo de configuração

Tutorial - Passo a Passo Abrindo no navegador

Digite no navegador: localhost/nomedoprojeto/public/

Laravel

[DOCUMENTATION](#)

[LARACASTS](#)


[NEWS](#)


[FORGE](#)


[GITHUB](#)


Tutorial - Passo a Passo Banco de Dados


Abrir o phpmyadmin: <http://localhost/phpmyadmin/>


 Servidor: 127.0.0.1

 Base de Dados



 SQL

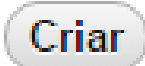
 Estado


 User accounts

 Exportar

Base de Dados

 Criar base de dados 






 Criar


Base de Dados 	Agrupamento (Collation)
---	-------------------------


Tutorial - Passo a Passo

Criar Tabela Aluno

Servidor: 127.0.0.1 » Base de Dados: minicursolaravel1

 Estrutura  SQL  Pesquisar  Pesquisa por formulário  Exportar

 Nenhuma tabela encontrada na base de dados.

 Criar tabela

Nome: Número de colunas:

Tutorial - Passo a Passo

Criar Tabela Aluno

Table name: Add column(s) Executar

Estrutura								
Nome	Tipo	Tamanho/Valores*	Predefinido	Agrupamento (Collation)	Atributos	Nulo	Índice	A_I
<input type="text" value="idAluno"/>	<input type="text" value="INT"/>	<input type="text"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text" value="UNSIGNED"/>	<input type="checkbox"/>	<input type="text" value="PRIMARY"/>	<input checked="" type="checkbox"/>
PRIMARY								
<input type="text" value="nome"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="255"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="---"/>	<input type="checkbox"/>
<input type="text" value="matricula"/>	<input type="text" value="INT"/>	<input type="text"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text" value="UNSIGNED"/>	<input type="checkbox"/>	<input type="text" value="---"/>	<input type="checkbox"/>
<input type="text" value="email"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="45"/>	<input type="text" value="None"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="---"/>	<input type="checkbox"/>

Comentários da tabela:

Collation:

Motor de armazenamento:

PARTITION definition:

Tutorial - Passo a Passo

Inserir Registros

 Procurar

 Estrutura

 SQL

 Pesquisar

 Inserir

 Exportar

 Importar

 Privilegios

Coluna	Tipo	Funções	Nulo	Valor
idAluno	int(10) unsigned	<input type="text"/>		<input type="text"/>
nome	varchar(255)	<input type="text"/>		<input type="text" value="Dayane"/>
matricula	int(10) unsigned	<input type="text"/>		<input type="text" value="123"/>
email	varchar(45)	<input type="text"/>		<input type="text" value="dayane.lima.castro@gmail.com"/>

Executar

☒ Ignora

Tutorial - Passo a Passo

De volta ao Laravel

Criação da Controller alunosController:

- `php artisan make:controller AlunosController`

Criação da rota:

```
Route::get("alunos", "AlunosController@index")  
    ->name("aluno.consulta");
```

Criação da view alunos:

- `alunos.blade.php`

Tutorial - Passo a Passo

De volta ao Laravel

Controller retornar a view alunos

```
public function index(){  
    return view("alunos");  
}
```

Criar array de alunos

```
public function index(){  
    $alunos = ["maria", "ana", "marcela"];  
    return view("alunos", ["alunos" => $alunos]);  
}
```


Tutorial - Passo a Passo

De volta ao Laravel

Percorrer alunos na view:

- @foreach

```
@foreach($alunos as $aluno)
    <p>{{$aluno}}</p>
@endforeach
```

Diferença entre {{ }} e {!! !!}: Escape de tags HTMLs

Tutorial - Passo a Passo database.php

DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=minicursolaravel1

DB_USERNAME=root

DB_PASSWORD=

Tutorial - Passo a Passo Model Aluno

Criar pasta model dentro de app

Comando: `php artisan make:model Model\Aluno`

Tutorial - Passo a Passo

Model Aluno

```
<?php

namespace App\Model;

use Illuminate\Database\Eloquent\Model;

class Aluno extends Model
{
    protected $table = "aluno";
    protected $primaryKey = "idAluno";
    public $timestamps = false;
    protected $guarded = [];
}
```

Tutorial - Passo a Passo Listando Dados

Substituir o array de alunos pelos dados do banco

```
public function index(){  
  
    $arrayAlunos = Aluno::all();  
  
    return view("alunos",["alunos" => $arrayAlunos]);  
}
```

Imprimir o nome do aluno

```
@foreach($alunos as $aluno)  
    <p>{{$aluno->nome}}</p>  
@endforeach
```

Tutorial - Passo a Passo Listando Dados

Criar tabela no html com matrícula, nome e e-mail

```
<table border="1">
  <thead>
    <tr>
      <td>Matrícula</td>
      <td>Nome</td>
      <td>E-mail</td>
    </tr>
  </thead>
  <tbody>
    @foreach($alunos as $aluno)
      <tr>
        <td>{{ $aluno->matricula }}</td>
        <td>{{ $aluno->nome }}</td>
        <td>{{ $aluno->email }}</td>
      </tr>
    @endforeach
  </tbody>
</table>
```

Tutorial - Passo a Passo

Edição de dados

Criar html com três inputs: nome, e-mail e idAluno

- alunoEdicao.blade.php

```
<form method="post" action="{{route('aluno.gravar')}}">
    {{csrf_field()}}
    <label for="">Nome</label>
    <input type="text" name="nome" value="{{ $aluno->nome }}">
    <br>
    <label for="">E-mail</label>
    <input type="text" name="email" value="{{ $aluno->email }}">
    <br>
    <input type="hidden" name="idAluno" value="{{ $aluno->idAluno }}">
    <br>
    <button type="submit" >Gravar</button>
</form>
```

Tutorial - Passo a Passo

Edição de dados

Criar rota de edição:

```
Route::get("aluno/editar/{idAluno}", "AlunosController@editar")  
    ->name("aluno.editar");
```

Criar botão de editar (Tag <a>) na tela de alunos (alunos.blade.php)

```
<td>  
    <a href="{{route('aluno.editar',$aluno->idAluno)}}">Editar</a>  
</td>
```


Tutorial - Passo a Passo

Edição de dados

Na controller: chamar a view criada, passando o objeto do Aluno

```
public function editar($idAluno){  
    $aluno = Aluno::find($idAluno);  
    return view("alunoEdicao",["aluno" => $aluno]);  
}
```

Na view: Setar os dados do aluno no value dos campos

Tutorial - Passo a Passo

Edição de dados

Criar rota de gravar:

```
Route::post("aluno", "AlunosController@gravar")  
    ->name("aluno.gravar");
```

Token: {{csrf_field()}}

Função dd(): dump and die - para imprimir os dados e sair da aplicação

Tutorial - Passo a Passo

Edição de dados

Implementar função de gravar na Controller

- Importar a classe Illuminate\Http\Request;

```
public function gravar(Request $request){
    $aluno = Aluno::findOrCreate($request->get("idAluno"));
    $aluno->nome = $request->get("nome");
    $aluno->email = $request->get("email");

    if($aluno->idAluno == ""){
        $aluno->matricula = $request->get("matricula");
    }

    $aluno->save();

    return redirect()->route("aluno.consulta");
}
```

Tutorial - Passo a Passo

Cadastro de Aluno

Criar tela de cadastrar aluno (matrícula, nome, e-mail e idAluno)

- alunoForm.blade.php

```
<form method="post" action="{{route('aluno.gravar')}}">
    {{csrf_field()}}
    <label for="">Nome</label>
    <input type="text" name="nome" value="">
    <br>
    <label for="">E-mail</label>
    <input type="text" name="email" value="">
    <br>
    <label for="">Matrícula</label>
    <input type="text" name="matricula" value="">
    <br>
    <input type="hidden" name="idAluno" value="">
    <br>
    <button type="submit" >Gravar</button>
</form>
```

Tutorial - Passo a Passo

Cadastro de Aluno

Criar rota para obter a tela

```
Route::get("aluno/novo", "AlunosController@novo")  
    ->name("aluno.novo");
```

Na Controller, Criar a função para retornar a tela

```
public function novo(){  
    return view("alunoForm");  
}
```

Tutorial - Passo a Passo

Cadastro de Aluno

Rota de gravar: A mesma da edição!

Na alunos.blade.php: Colocar botão para ir à tela de cadastro

```
<a href="{{route('aluno.novo')}}" target="_blank">Novo Aluno</a>
```

Tutorial - Passo a Passo

Exclusão de Aluno

Criar função de exclusão de aluno na Controller

```
public function excluir($idAluno){  
    $aluno = Aluno::find($idAluno);  
    $aluno->delete();  
  
    return redirect()->route("aluno.consulta");  
}
```

Criar Rota de exclusão de aluno

```
Route::get("aluno/excluir/{idAluno}", "AlunosController@excluir")  
    ->name("aluno.excluir");
```

Tutorial - Passo a Passo

Exclusão de Aluno

Na tela de listagem do alunos (alunos.blade.php), criar uma coluna para colocar a opção de exclusão

```
<td>
    <a href="{{route('aluno.excluir',$aluno->idAluno)}}">
        Excluir
    </a>
</td>
```


Tutorial - Passo a Passo Internacionalização

- Função trans: Obtém o texto de acordo com a linguagem configurada.
- Arquivo configuração da linguagem: config -> app.php
 - 'locale' => 'en'
- Arquivo com os textos: resources -> lang -> [en,pt-BR,es]
- Sintaxe: nome_arquivo.chave_associativa

```
{{ trans('aplicacao.aluno') }}  
@lang('aplicacao.aluno')
```

Obrigada pela atenção!

=]

Dúvidas?



Código Fonte e Slide:

- <https://github.com/dayaneLima/minicursolaravel.git>



Contatos:

- <https://www.facebook.com/dayane.lima.714>
- dayane.lima.castro@gmail.com
- dayane@symcode.com.br