

ENSINO MÉDIO INTEGRADO - INFORMÁTICA

Disciplina de Desenvolvimento Web

Aula 04: Framework Laravel - Controllers

Gil Eduardo de Andrade

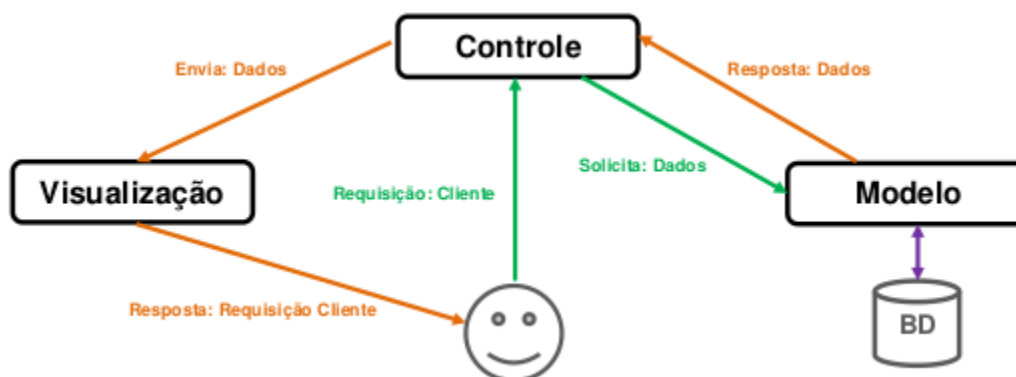
Conceitos Preliminares

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

Modelo MVC

O Framework Laravel utiliza, em sua estrutura de desenvolvimento, o padrão MVC (Model-View-Controller). O padrão MVC é adotado no desenvolvimento de uma grande gama de projetos, visto que sua arquitetura permite dividir o projeto em camadas. Cada uma dessas camadas, Model, Controller e View, é responsável apenas pela execução do que lhe foi definido.

Sendo assim, como benefício do uso do padrão MVC, temos a separação entre as regras do negócio e a interface com o usuário. Tal característica permite a existência de várias interfaces, que podem ser alteradas sem que seja necessário modificar as regras de negócio, provendo maior flexibilidade e possibilidade de reuso de código.





Codificação – Laravel / Controladoras

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

Criando Classes de Controle

(No terminal - dentro da pasta do projeto)

```
php artisan make:controller --help
```



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/04-Controllers
Arquivo Editar Ver Terminal Abas Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/04-Controllers$ php artisan make:controller --help
Description:
  Create a new controller class

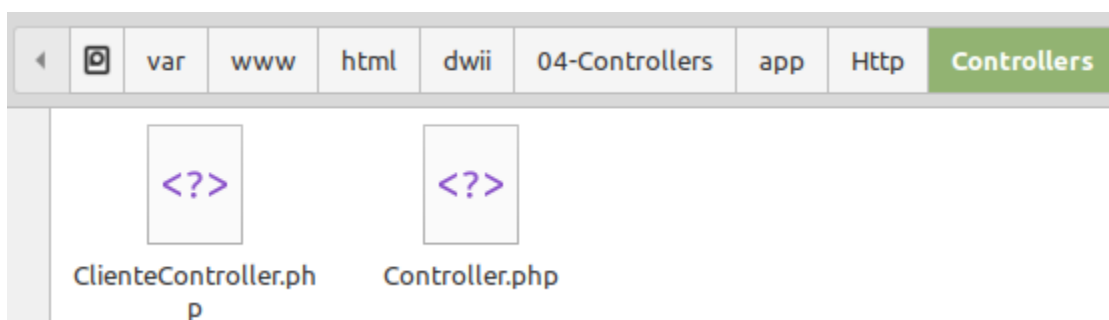
Usage:
  make:controller [options] [--] <name>

Arguments:
```

```
php artisan make:controller ClienteController -r
```



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/04-Controllers
Arquivo Editar Ver Terminal Abas Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/04-Controllers$ php artisan make:controller Cliente
Controller -r
Controller created successfully.
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/04-Controllers$
```



(Arquivo: “/app/Http/Controllers/CienteController.php”)

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class CienteController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }
    ...
}
```

Classe de Controle Criada - CienteController.php

Redirecionando as Requisições para um Método da Classe de Controle

(Arquivo: “/app/Providers/RouteServiceProvider.php”)

```
// Descomentar a linha
```

```
protected $namespace = 'App\Http\Controllers';
```

Configura o Espaço de Nomes para as Classes de Controle

(Arquivo-fonte: 04-Controllers/app/Providers/RouteServiceProvider.php)

(Arquivo: “/routes/web.php”)

```
use App\Http\Controllers\ClienteController;
```

```
...
```

```
// Route::get('/clientes', [ClienteController::class, 'index']);
```

```
Route::get('/clientes', 'ClienteController@index');
```

Redirecionado a Rota “/cliente” para Classe de Controle “Cliente”

[‘ClienteController@index’] - indica a classe de controle e o método a ser invocado

(Arquivo-fonte: 04-Controllers/routes/web.php)

(Arquivo: “/Controllers/ClienteController.php”)

```
public function index() {  
  
    return "<h2>Lista de Clientes</h2>";  
  
}
```

Classe de Controle “ClienteController” - Método “index()”

(Arquivo-fonte: 04-Controllers/app/Http/Controllers/ClienteController.php)



Lista de Clientes

(Execução da codificação anterior)

Listando Rotas via *artisan*

`php artisan route:list`

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/04-Controllers
g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/04-Controllers$ php artisan route:list
```

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	GET HEAD	api/user		Closure	api
	GET HEAD	clientes		App\Http\Controllers\ClienteController@index	App\Http\Middleware\Authenticate:sanctum
	GET HEAD	sanctum/csrf-cookie		Laravel\Sanctum\Http\Controllers\CsrfCookieController@show	web
					web

```
g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/04-Controllers$
```

Passando Parâmetro para Classe de Controle (/cliente/{nome}/{idade})

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

```
Route::get('/clientes/{nome}/{idade}', 'ClienteController@index');
```

Passando Parâmetro - Classe de Controle "ClienteController"

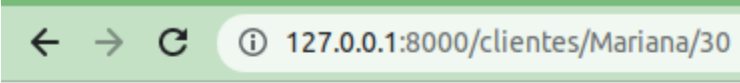
(Arquivo-fonte: 04-Controllers/routes/web.php)

(Arquivo: "/Controllers/Cliente.php")

```
public function index($nome, $idade) {
    $dados = "<h2>Lista de Clientes</h2>".
        "<b>Nome:</b> $nome<br>".
        "<b>Idade:</b> $idade";
    return $dados;
}
```

Classe de Controle "ClienteController" - Método "index()"

(Arquivo-fonte: 04-Controllers/app/Http/Controllers/ClienteController.php)



← → ↻ ⓘ 127.0.0.1:8000/clientes/Mariana/30

Lista de Clientes

Nome: Mariana

Idade: 30

(Execução da codificação anterior)

Otimizando o Processo / Rotas e Classes de Controle

Rotas Comentada: “/clientes”

Rotas Comentada: “/clientes/{nome}/{idade}”

(Arquivo: “/routes/web.php”)

```
Route::resource('clientes', 'ClienteController');
```

Utilizando o Conjunto de Rotas “Resource”

(Arquivo-fonte: 04-Controllers/routes/web.php)

Listando Rotas via *artisan*

```
php artisan route:list
```

```
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/04-Controllers$ php artisan route:list
+-----+-----+-----+-----+-----+
| Domain | Method | URI | Name | Action |
+-----+-----+-----+-----+-----+
| | GET|HEAD | clientes | clientes.index | App\Http\Controllers\ClienteController@index |
| | POST | clientes | clientes.store | App\Http\Controllers\ClienteController@store |
| | GET|HEAD | clientes/create | clientes.create | App\Http\Controllers\ClienteController@create |
| | GET|HEAD | clientes/{cliente} | clientes.show | App\Http\Controllers\ClienteController@show |
| | PUT|PATCH | clientes/{cliente} | clientes.update | App\Http\Controllers\ClienteController@update |
| | DELETE | clientes/{cliente} | clientes.destroy | App\Http\Controllers\ClienteController@destroy |
| | GET|HEAD | clientes/{cliente}/edit | clientes.edit | App\Http\Controllers\ClienteController@edit |
+-----+-----+-----+-----+-----+
```

Construindo um CRUD - Cliente

(Arquivo: “/routes/web.php”)

```
Route::redirect('/', '/clientes', 301);  
Route::resource('cliente', 'Cliente');
```

Redirecionando a rota “/” (principal) para “/cliente”

(Arquivo-fonte: 04-Controllers/routes/web.php)

(Arquivo: “/Controllers/ClienteController.php”)

```
...  
  
// define um atributo com array inicial e dados  
  
public $clientes = [  
  
    'id' => 1,  
  
    'nome' => 'Gil Eduardo',  
  
    'email' => 'gil@gil.com'  
  
];  
  
public function __construct() {  
  
    // obtém o conteúdo da variável de sessão "clientes"  
  
    $aux = session('clientes');  
  
    // verifica se a sessão já estava setada  
  
    if(!isset($aux)) {  
  
        // seta a sessão "clientes" com o array  
  
        session(['clientes' => $this->clientes]);  
  
    }  
  
}  
  
public function index() {  
  
    // obtém o conteúdo da variável de sessão "clientes"  
  
    $clientes = session('clientes');
```

```
// envia "$clientes" para view "index"
return view('clientes.index', compact('clientes'));
}
...
```

Construindo um CRUD - Métodos `__construct()` e `index()`

(Arquivo-fonte: `04-Controllers/app/Http/Controllers/ClienteController.php`)

(Arquivo: `"/Controllers/ClienteController.php"`)

```
...
public function create() {
    return view('clientes.create');
}

public function store(Request $request) {
    // obtém os dados da sessão "clientes"
    $aux = session('clientes');

    // retorna um array contendo apenas os dados da coluna "id"
    $ids = array_column($aux, 'id');

    // verifica o total de elementos do array "id"
    if(count($ids) > 0) {
        // obtém o valor máximo do array "id" + 1
        $new_id = max($ids) + 1;
    }
    else {
        // configura novo id com 1
        $new_id = 1;
    }
}
```



```
// Array com os dados do novo cadastro

$novo = [

    'id' => $new_id,

    'nome' => $request->nome,

    'email' => $request->email

];

// Insere novo cadastro no array

array_push($aux, $novo);

// Atualiza a sessão com o novo cadastro

session(['clientes' => $aux]);

// redireciona para lista de clientes

return redirect()->route('clientes.index');

}

...
```

Construindo um CRUD - Métodos create() e store()

(Arquivo-fonte: 04-Controllers/app/Http/Controllers/ClienteController.php)

Novos Arquivos

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

```
<h2>Cadastrar Cliente</h2>

<form action="{ route('cliente.store') }" method="POST">

    <!-- Token de segurança salvo na sessão, verificar o formulário submetido -->

    @csrf

    <a href="{ route('cliente.index') }" ><h4>voltar</h4></a>

    <label>Nome: </label> <input type='text' name='nome'>

    <label>E-mail: </label> <input type='text' name='email'>

    <input type="submit" value="Salvar">
```

```
</form>
```

View “Create” - Utilizada para cadastrar os clientes

(Arquivo-fonte: 04-Controllers/resources/views/clientes/create.blade.php)

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

```
<h2>Lista de Clientes</h2>

<a href="{{ route('cliente.create') }}"><h4>Novo Cliente</h4></a>

<table>

  <thead>

    <tr>

      <th>ID</th>

      <th>NOME</th>

      <th>E-MAIL</th>

      <th>INFO</th>

      <th>EDITAR</th>

      <th>REMOVER</th>

    </tr>

  </thead>

  <tbody>

    <!-- Funcionalidade Blade / Laço Repetição -->

    <!-- Percorre o array clientes enviado pela Controller -->

    @foreach ($clientes as $item)

      <tr>

        <!-- Acessa os campos de cada item do array -->

        <td>{{ $item['id'] }}</td>

        <td>{{ $item['nome'] }}</td>
```

```
<td>{{ $item['email'] }}</td>

<td><a href="{{ route('cliente.show', $item['id']) }}">info</a></td>

<td><a href="{{ route('cliente.edit', $item['id']) }}">editar</a></td>

<td>

    <form action="{{ route('cliente.destroy', $item['id']) }}"

method="POST">

        <!-- Token de Segurança -->

        <!-- Define o método de submissão como delete -->

        @csrf

        @method('DELETE')

        <input type='submit' value='remove'>

    </form>

</td>

</tr>

@endforeach

</tbody>

</table>
```

View “index” - Utilizada para listar os dados dos clientes

(Arquivo-fonte: 04-Controllers/resources/views/clientes/index.blade.php)

(View "index")

← → ↻ ⓘ 127.0.0.1:8000/clientes

Lista de Clientes

[Novo Cliente](#)

ID	NOME	E-MAIL	INFO	EDITAR	REMOVER
1	Gil Eduardo	gil@gil.com	info	editar	<input type="button" value="remove"/>

(View "create")

← → ↻ ⓘ 127.0.0.1:8000/clientes/create

Cadastrar Cliente

[voltar](#)

Nome: E-mail:

(View "index")

← → ↻ ⓘ 127.0.0.1:8000/clientes

Lista de Clientes

[Novo Cliente](#)

ID	NOME	E-MAIL	INFO	EDITAR	REMOVER
1	Gil Eduardo	gil@gil.com	info	editar	<input type="button" value="remove"/>
2	Mariana Santos	mariana@gmail.com	info	editar	<input type="button" value="remove"/>

(Execução da codificação anterior)

Novo Arquivo / Método “show()”

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

```
<h2>Informações do Cliente</h2>

<a href="{{route('clientes.index')}}"><h4>voltar</h4></a>

<h4>ID: {{ $dados['id'] }}</h4>

<h4>Nome: {{ $dados['nome'] }}</h4>

<h4>E-mail: {{ $dados['email'] }}</h4>
```

View “show” - Utilizada para exibir as informações de um cliente específico

(Arquivo-fonte: 04-Controllers/resources/views/clientes/show.blade.php)

Trecho de Código Adicionado / Método “show()”

(Arquivo: “/Controllers/Cliente.php”)

```
public function show($id) {

    // Obtém os dados da variável de sessão "clientes"

    $aux = session('clientes');

    // Obtém o índice do array "$aux" onde está o "$id" buscado

    $indice = array_search($id, array_column($aux, 'id'));

    // Armazena os dados do cliente para o índice obtido

    $dados = $aux[$indice];

    // retorna a "view" e passa os dados do cliente

    return view('clientes.show', compact('dados'));

}
```

Classe de Controle “ClienteController” - Método “show()”

(Arquivo-fonte: 04-Controllers/app/Http/Controllers/ClienteController.php)

← → ↻ ⓘ 127.0.0.1:8000/clientes/2

Informações do Cliente

[voltar](#)

ID: 2

Nome: Mariana Santos

E-mail: mariana@gmail.com

(Execução da codificação anterior)

Novo Arquivo / Métodos “edit()” e “update()”

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

```
<h2>Alterar Cliente</h2>

<form action="{{ route('clientes.update', $dados['id']) }}" method="POST">

    <!-- Token de Segurança -->

    <!-- Define o método de submissão como PUT -->

    @csrf

    @method('PUT')

    <a href="{{ route('clientes.index') }}"><h4>voltar</h4></a>

    <label>Nome: </label> <input type="text" name="nome" value="{{ $dados['nome'] }}">

    <label>E-mail: </label> <input type="text" name="email" value="{{ $dados['email'] }}">

    <input type="submit" value="Salvar">

</form>
```

View “edit” - Utilizada para alterar as informações de um cliente

(Arquivo-fonte: 04-Controllers/resources/views/clientes/edit.blade.php)

Trecho de Código Adicionado / Métodos “edit()” e “update()”

(Arquivo: “/Controllers/ClienteController.php”)

```
public function edit($id) {  
  
    // Obtém os dados da variável de sessão "clientes"  
  
    $aux = session('clientes');  
  
    // Obtém o índice do array "$aux" onde está o "$id" buscado  
  
    $indice = array_search($id, array_column($aux, 'id'));  
  
    // Armazena os dados do cliente para o índice obtido  
  
    $dados = $aux[$indice];  
  
    // retorna a "view" e passa os dados do cliente  
  
    return view('clientes.edit', compact('dados'));  
  
}  
  
public function update(Request $request, $id) {  
  
    // Cria o array com os novos dados do cliente  
  
    $alterado = [  
  
        'id' => $id,  
  
        'nome' => $request->nome,  
  
        'email' => $request->email  
  
    ];  
  
    // Obtém os dados da variável de sessão "clientes"  
  
    $aux = session('clientes');  
  
    // Obtém o índice do array "$aux" onde está o "$id" buscado  
  
    $indice = array_search($id, array_column($aux, 'id'));  
  
    // Substitui os dados do cliente com as novas informações  
  
    $aux[$indice] = $alterado;  
  
    // Atualiza a sessão com a nova alteração  
  
    session(['clientes' => $aux]);  
  
    // redireciona para lista de clientes
```

```
return redirect()->route('clientes.index');  
}
```

Classe de Controle “ClienteController” - Métodos “edit()” e “update()”

(Arquivo-fonte: 04-Controllers/app/Http/Controllers/Cliente.php)

← → ↻ ⓘ 127.0.0.1:8000/clientes/1/edit

Alterar Cliente

[voltar](#)

Nome: E-mail:

← → ↻ ⓘ 127.0.0.1:8000/clientes

Lista de Clientes

[Novo Cliente](#)

ID	NOME	E-MAIL	INFO	EDITAR	REMOVER
1	Gil Eduardo de Andrade	gil@gmail.com	info	editar	<input type="button" value="remove"/>
2	Mariana Santos	mariana@gmail.com	info	editar	<input type="button" value="remove"/>

(Execução da codificação anterior)

Trecho de Código Adicionado / Método “destroy()”

(Arquivo: “/Controllers/ClienteController.php”)

```
public function destroy($id) {  
  
    // Obtém os dados da variável de sessão "clientes"
```



```
$aux = session('clientes');  
  
// Obtém o índice do array "$aux" onde está o "$id" buscado  
  
$indice = array_search($id, array_column($aux, 'id'));  
  
// Remove o elemento em "indice" do array "$aux"  
  
unset($aux[$indice]);  
  
// Atualiza a sessão com a nova alteração  
  
session(['clientes' => $aux]);  
  
// redireciona para lista de clientes  
  
return redirect()->route('clientes.index');  
  
}
```

Classe de Controle “Cliente” - Método “destroy()”

(Arquivo-fonte: 04-Controllers/app/Http/Controllers/ClienteController.php)

← → ↻ ⓘ 127.0.0.1:8000/clientes

Lista de Clientes

[Novo Cliente](#)

ID	NOME	E-MAIL	INFO	EDITAR	REMOVER
1	Gil Eduardo de Andrade	gil@gmail.com	info	editar	<input type="button" value="remover"/>

(Execução da codificação anterior)