

# Framework Laravel

Desenvolvimento rápido e simples



# O que é PHP?

- PHP Significa HyperText Preprocessor
- Linguagem server-side para desenvolvimento web
- Começou a ser desenvolvida em 1995 por Rasmus Lerdorf
- Atualmente a última versão lançada é 7.3



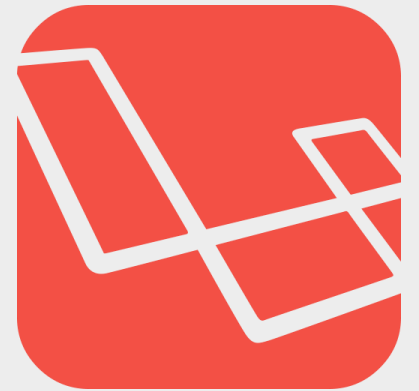
# O que é um Framework?

**Framework** é um "esqueleto" que une códigos comuns entre vários software, provendo uma funcionalidade genérica. O **Framework** é diferente de uma biblioteca, tendo em vista que o **Framework** dita um modelo, e este será responsável por nos proporcionar por meio de um conjunto de bibliotecas essenciais que nos permite desenvolver alguma solução específica.



# O que é Laravel?

- Laravel é um Framework em PHP baseado no conceito RAD (Rapid Application Development).
- Desenvolvido por Taylor Otwell.
- Possui código aberto e uma extensa comunidade ativa.
- Utiliza o Composer, fornecendo todo o poder das demais bibliotecas escritas para PHP.




# O que é Composer?

- Gerenciador de dependências para aplicações em PHP.
- Não precisamos ter nossas dependências juntas com o projeto, podemos instalar utilizando o Composer.
- Qualquer pessoa pode escrever uma biblioteca para PHP e publicá-la para que todos possam utilizá-la.



# Criando nossa aplicação Laravel



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

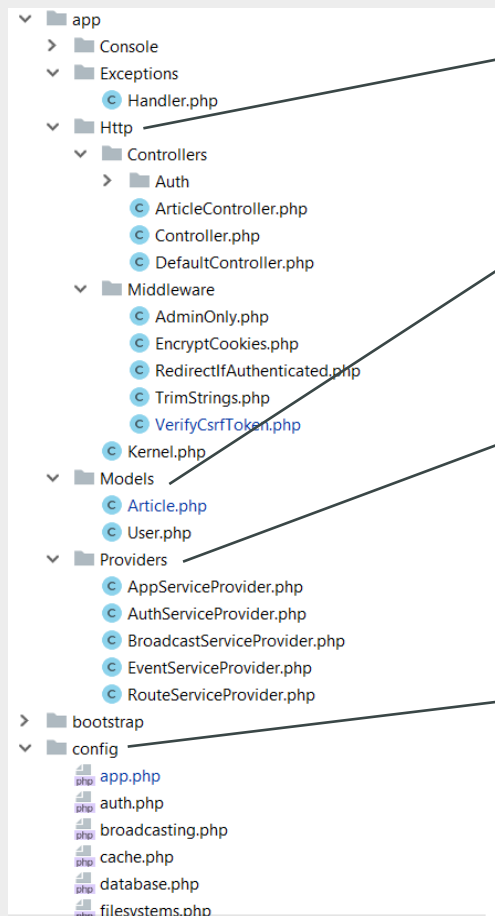
**Composer** é o comando geral instalado no sistema.

**Create-project** indica a criação de um novo projeto.

**Meuprojeto** é o nome do projeto a ser criado.

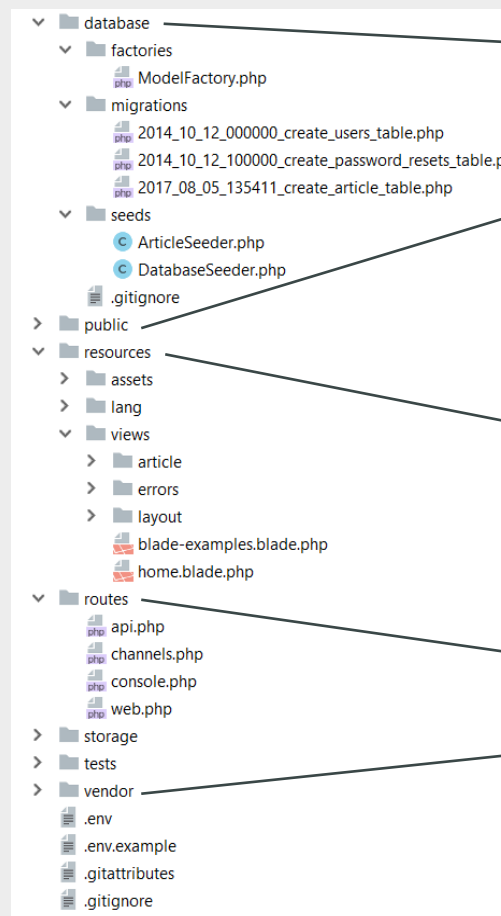
**--prefer-dist** indica que é para baixar a versão do Laravel mais atualizada.

# Estrutura de pastas



- **app/Http** pasta onde ficam os **Controllers**, **Middlewares** e o **Kernel**.
- Todos os **Models** ficam localizados na pasta **app/Models**.
- Os provedores de serviços que são funções de bootstrapping em nosso aplicativo estão localizados na pasta **app/Providers**.
- Todos os arquivos de configurações ficam na pasta **app/config**.

# Estrutura de pastas



- Pasta do banco de dados que contém as migrações e Seeds.
- A pasta **/public** é a pasta real que você está abrindo no servidor da web. Todos os arquivos CSS/JS estão localizados nela.
- A pasta **/resources** contém todas as traduções, views e assets (SASS, LESS, JS) que são compilados e transferidos para pasta **/public**.
- A pasta **/routes** contém todas as rotas do projeto.
- A pasta do **/vendor** contém todos os pacotes do composer (bibliotecas).



# Arquitetura do banco de dados

Tenha cuidado com a arquitetura do banco de dados, nunca esqueça os índices de colunas pesquisáveis.

```
/**
 * Run the migrations.
 *
 * @return void
 */
public function up()
{
    Schema::create('veiculos', function (Blueprint $table) {
        $table->increments('id');
        $table->string('nome_proprietario');
        $table->string('marca');
        $table->string('placa');
        $table->string('cor');
        $table->string('ano_fab');
        $table->string('ano_modelo');
        $table->string('chassi');
        $table->string('renavam');
        $table->string('descricao');
        $table->index(['nome_proprietario', 'placa', 'chassi', 'renavam']);
        $table->integer('modelo_id')->unsigned();
        $table->foreign('modelo_id')->references('id')->on('modelos')->onDelete('cascade');
        $table->integer('user_id')->unsigned();
        $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
        $table->timestamps();
    });
}
```

# Route

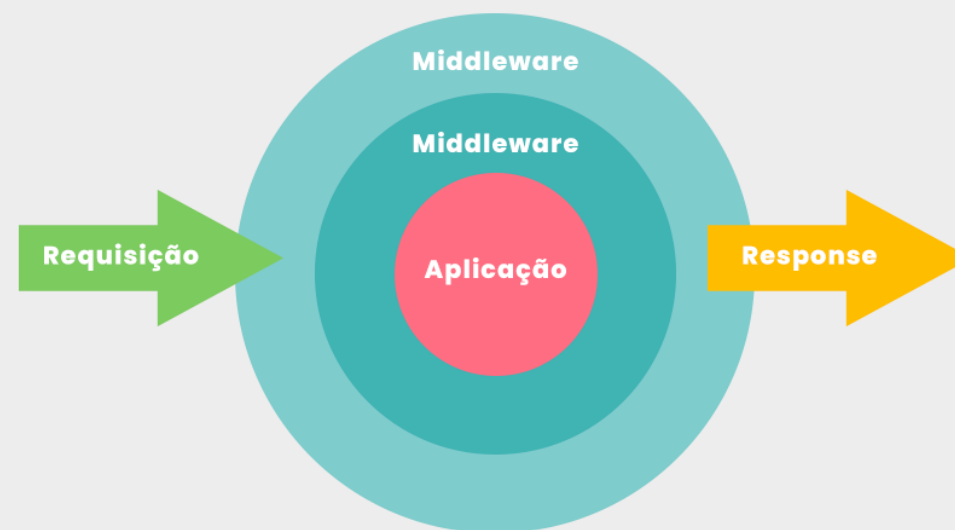
- É melhor e o mais prático sistema de rotas.
- Roteamento pode ser feito por middleware, prefixo ou namespace.
- Roteamento por método de requisição (GET, POST, DELETE, etc.).

## Cuidados

- Sempre nomeie as suas rotas ou utilize o método resource.
- Tenha cuidado com a ordem das suas rotas.

# Middleware

- O middleware é mecanismo para filtrar as requisições HTTP.
- Laravel inclui vários middlewares - Autenticação, Proteção CSRF e etc.
- O middleware de autenticação verifica se o usuário que está visitando a página está autenticado por meio do cookie de sessão.
- O middleware de proteção CSRF protege seu aplicativo contra ataques de falsificação de solicitações entre sites, adicionando uma token para cada formulário gerado.



# Eloquent & Database

```
/**
 * Store a newly created resource in storage.
 *
 * @param  \App\Http\Requests\Modelos\CreateModeloRequest $request
 * @return \Illuminate\Http\Response
 */
public function store(CreateModeloRequest $request)
{
    Modelo::create([
        'modelo' => 'Meu modelo'
    ]);

    return redirect()->route('admin.modelo.index')->with('message', 'Modelo de veículo adicionado com sucesso');
}
```

- O Eloquent ORM (mapeamento objeto-relacional) fornece implementação simples do Active Record para trabalhar com o banco de dados.
- Relacionamentos do Eloquent são eles: **um-para-um**, **um-para-vários**, **vários-para-vários**, **polimórficos**.

# Prática

- Vamos brincar com o Laravel e criar CRUD para gerenciar Modelos de veículos (Criar, Visualizar, Atualizar, Apagar).

O modelo terá as seguintes colunas / campos:

- **id** - chave primária - não nula.
- **modelo** - varchar - 255 de tamanho - não nulo.
- **status** – enum [ativo/inativo] - não nulo - padrão é ativo.
- **created\_at** - datetime - não nulo
- **updated\_at** - datetime - não nulo

# Boas práticas no Laravel

```
/**
 * Display a listing of the resource.
 *
 * @return \Illuminate\Http\Response
 */
public function index()
{
    Modelo::listarUltimosModelos(6);

    Modelo::orderBy('created_at', 'DESC')->paginate(6);
}
```

- **NUNCA** escreva consultas ou lógica que consistem em utilizar o seu banco de dados dentro do controlador. O trabalho do controlador é se comunicar com o modelo e passar dados para a **view**.

# Blade Template

- Blade é o poderoso mecanismo de template fornecido pelo Laravel.
- Todo o código dentro do arquivo Blade é compilado para um arquivo HTML estático e gerado um Cache.
- Suporta PHP simples.
- Poupa tempo
- Melhor mobilidade de componentes, ampliar e incluir parciais.



# View – Blade Template

```
@extends('admin.template.default')
@section('content')
    <div class="m-content">
        <div class="m-alert">
            <div class="m-alert__icon">
                <i class="flaticon-exclamation m--font-brand"></i>
            </div>
            <div class="m-alert__text">
                Aqui vai a sua mensagem de texto
            </div>
        </div>
    </div>
@endsection
```

Estenda e inclua parciais. Por exemplo, crie uma página estendendo o template padrão de sua aplicação.



# Seguranças de formulário

```
<form method="POST" action="{{ route('admin.modelos.store') }}">
    @csrf

    <div class="form-group">
        <input type="text" class="form-control" name="modelo" placeholder="Nome do modelo">
    </div>

    <button type="submit" class="btn btn-success">
        Salvar
    </button>
</form>
```

```
<form method="POST" action="http://localhost:8000/admin/modelos">
    <input type="hidden" name="_token" value="HB6XTrpmbtv4ZhrPCx60dx0c8wNYLLTVGeOwEzGd">

    <div class="form-group">
        <input type="text" class="form-control" name="modelo" placeholder="Nome do modelo">
    </div>

    <button type="submit" class="btn btn-success">
        Salvar
    </button>
</form>
```

- Sempre use a proteção de token CSRF que o Laravel fornece nos formulários criados, os hackers não poderão enviar spam para seus formulários e banco de dados.
- O retorno do **@CSRF** é um campo oculto com um token gerado, este valor é enviado e verificado na requisição.

# PHPDoc - Documentação



```
/**
 * Display a listing of the resource.
 *
 * @return \Illuminate\Http\Response
 */
public function index()
{
    return view('admin.mapas.index');
}
```

Não se esqueça de escrever comentários para cada método ou lógica desenvolvida. Os comentários do PHPDoc estão ajudando a IDE a auto sugerir mais facilmente, funções, helpers e também, auxiliar aos desenvolvedores a entender o que está escrito no código.

# Obrigado! Dúvidas?



- CEO, Alumiar.
- Breno Martins, 22.
- Github: [www.github.com/brenopereira](https://www.github.com/brenopereira).
- E-mail: [breno.pereira@alumiar.me](mailto:breno.pereira@alumiar.me)
- Website: [www.alumiar.me](http://www.alumiar.me)