**FACULDADE DE TECNOLOGIA DE TAUBATÉ**

**Fabiano Romeu**

**Elton Gouvea**

##### Conceitos MVC no Laravel 5.4 em diante

**TAUBATÉ**

**2018**

**Fabiano Romeu**

**Elton Gouvea**

##### Conceitos MVC no Laravel 5.4 em diante

Relatório Técnico apresentado ao professor da disciplina de Programação para WEB do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemasdo Centro Estadual de Educação Tecnológica Paula Souza para a obtenção da avaliação do 2º bimestre do 2º semestre de 2018.

**Professor  
Prof. Me Fernando Salles Claro**

**TAUBATÉ**

**2018**

**RESUMO**

Laravel é um framework poderoso, livre e de código aberto, desenvolvido em 2011 por Taylor B. Otwell para o desenvolvimento de aplicações em PHP com a arquitetura MVC (model, view, controller). Lançado sob a licença MIT e tendo seu código-fonte hospedado no GitHub, rapidamente conquistou a preferência da comunidade de desenvolvedores, chegando a ser listado como o framework mais popular entre os desenvolvedores, de acordo com uma pesquisa feita em 2015.Atualmente o Laravel encontra-se em sua versão 5.7, sendo tão completo que passou a ser considerado um verdadeiro ecossistema para o desenvolvimento de aplicações e web services em PHP. Sua popularidade se dá pela ampla gama de recursos oferecidos ao desenvolvedor, a redução de tempo no desenvolvimento, colaborando também em redução no custo financeiro dos projetos.Uma outra qualidade é a ampla documentação com linguagem simples e direta que inclui tutoriais facilitando a interação de iniciantes com a aplicação

**Palavras-Chave**: Laravel; MVC; Conceitos; Aplicação; Framework; Web Programming; Models, Controllers; Views.

**SUMÁRIO**

[**Desenvolvimento** 5](#_Toc531158244)

[O que é MVC ? 5](#_Toc531158245)

[Como o MVC é implementado em aplicativos do Laravel? 6](#_Toc531158246)

[Models: Criando o Model do produto 7](#_Toc531158247)

[Migrate - Migrações no Laravel 9](#_Toc531158248)

[Controllers: Criando o Controller 10](#_Toc531158249)

[Views: Criando as Views do projeto 12](#_Toc531158250)

[**Conclusão** 14](#_Toc531158251)

[**Referências** 15](#_Toc531158252)

# **Desenvolvimento**

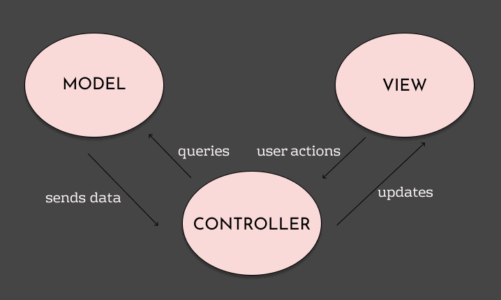
## O que é MVC ?

MVC é um acrônimo para "Model View Controller". Ele representa a arquitetura de desenvolvimento adotada ao criar aplicativos. Com a arquitetura MVC, olhamos para a estrutura da aplicação em relação à forma como o fluxo de dados da nossa aplicação funciona.

O Model gerencia comportamentos e dados fundamentais do aplicativo. Ele pode responder a solicitações de informações, responder a instruções para alterar o estado de suas informações e até mesmo notificar observadores em sistemas controlados por eventos quando as informações forem alteradas. Pode ser um banco de dados ou qualquer quantidade de estruturas de dados ou sistemas de armazenamento. Em resumo, são os dados e o gerenciamento de dados do aplicativo.

A View fornece efetivamente o elemento de interface do usuário do aplicativo. Ele renderizará os dados do Modelo em um formato adequado para a interface do usuário.

O Controller recebe a entrada do usuário e faz chamadas para modelar objetos e a visualização para executar ações apropriadas.



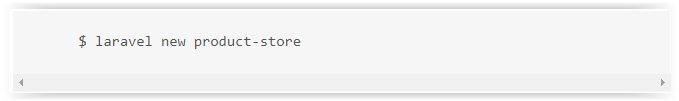
Fonte: <https://blog.pusher.com/LARAVEL-MVC-USE/>

Um Model é uma representação de uma instância ou objeto da vida real em nossa base de código. A View representa a interface através da qual o usuário interage com nosso aplicativo. Quando um usuário executa uma ação, o Controller manipula a ação e atualiza o Model, se necessário.

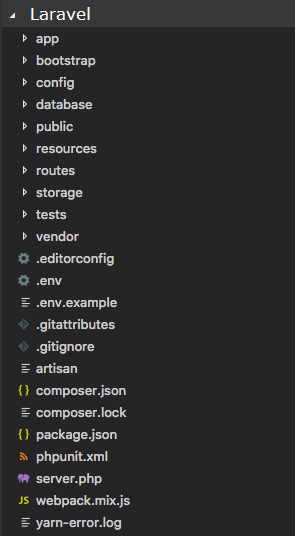
## Como o MVC é implementado em aplicativos do Laravel?

A seguir como o Laravel usa o MVC durante o desenvolvimento. Para fazer isso, criaremos um exemplo de projeto do Laravel que exibe produtos para uma loja.

Para criar um novo projeto, execute o comando abaixo no terminal:



Quando se cria um novo projeto do Laravel, o projeto tem a seguinte estrutura:



## Models: Criando o Model do produto

Criar o Model, 'M' em MVC, para a aplicação. O modelo geralmente faz interface com um armazenamento de dados como um banco de dados SQL. No Laravel, o Model geralmente é uma classe com propriedades que correspondem às colunas no banco de dados.

No banco de dados, um produto terá as seguintes propriedades:

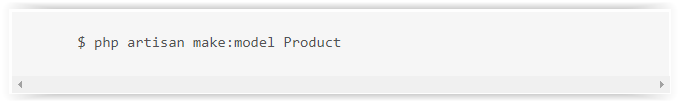
Nome (nome) - Nome do produto.

Descrição resumida (descricao) - Breve descrição de um produto.

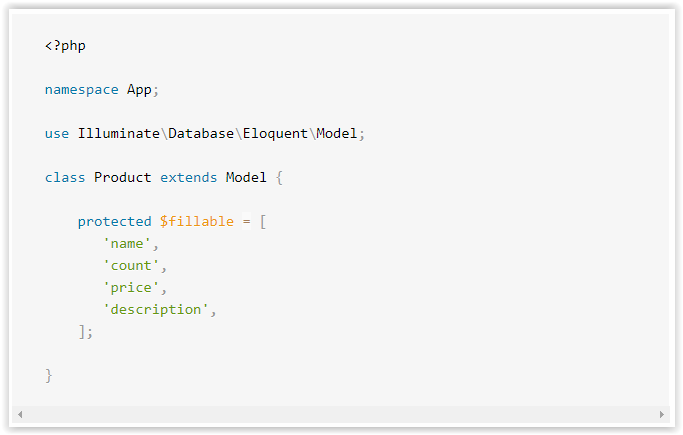
Contagem (contagem) - O número de produtos disponíveis.

Preço unitário (preco) - Quanto custa um único produto.

Para criar um Model no Laravel, execute o comando no terminal:



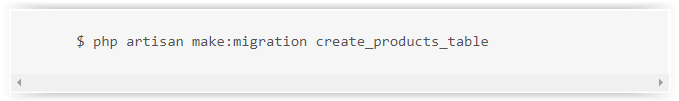
Ao executar este comando, o Laravel criará um arquivo Product.php no diretório do aplicativo. Esta será uma classe PHP com o nome Product e será o modelo para a tabela de produtos no banco de dados.



### Migrate - Migrações no Laravel

A migração permite que os desenvolvedores façam e desfazem alterações no banco de dados de um projeto. As migrações podem ser usadas para tornar o gerenciamento de bancos de dados fácil e previsível.

Para criar uma migração, execute o seguinte no terminal:



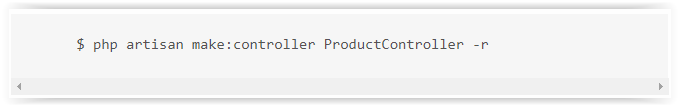
Quando o comando é executado, deve-se ver um novo arquivo *create\_products\_table.php* no diretório *databases/migrations* e pode-se editá-lo para ter o esquema de tabela de produtos como este:



## Controllers: Criando o Controller

Os controladores são responsáveis por concluir as ações do usuário e gerenciar a lógica de negócios de nossos aplicativos. Para o nosso projeto de faz de conta, vamos usar Controladores de Recursos.

Para criar um controlador de recursos no Laravel, execute o seguinte comando:



O sinalizador -r faz dele um controlador de recursos e, portanto, cria todos os métodos necessários para a operação CRUD.

Quando o comando é executado, o Laravel criará um novo arquivo no diretório *app/Http/Controllers* chamado *ProductController.php.*

Para o arquivo *routes/web.php* e adicione a seguinte rota:



Isso diz ao Laravel para criar todas as rotas necessárias para um controlador de recursos e mapeá-las para a classe *ProductController.*

No arquivo do controlador atualiza-se os métodos neles com a seguinte lógica:

O método *create:*



*Create (C em CRUD)*. O controlador carrega uma *View (V em MVC)* chamada *createproduct* e serve como resposta para qualquer hora que alguém visita a rota */products/create* com um método GET HTTP.

O método *store:*



O método *store* é chamado quando um usuário envia uma solicitação *POST HTTP* para o *ENDPOINT* */products*. Essa lógica obtém os dados da solicitação e os armazena no banco de dados usando o modelo *Product.*

O método *index:*

**

O método *index* é chamado quando a rota */products* é carregada com um método *HTTP GET*. Neste método, busca-se todos os produtos disponíveis na tabela de produtos usando o modelo *Product* e passa-se para a *View* como uma variável. Isso significa que na exibição, a variável *$allProducts* estará disponível.

## Views: Criando as Views do projeto

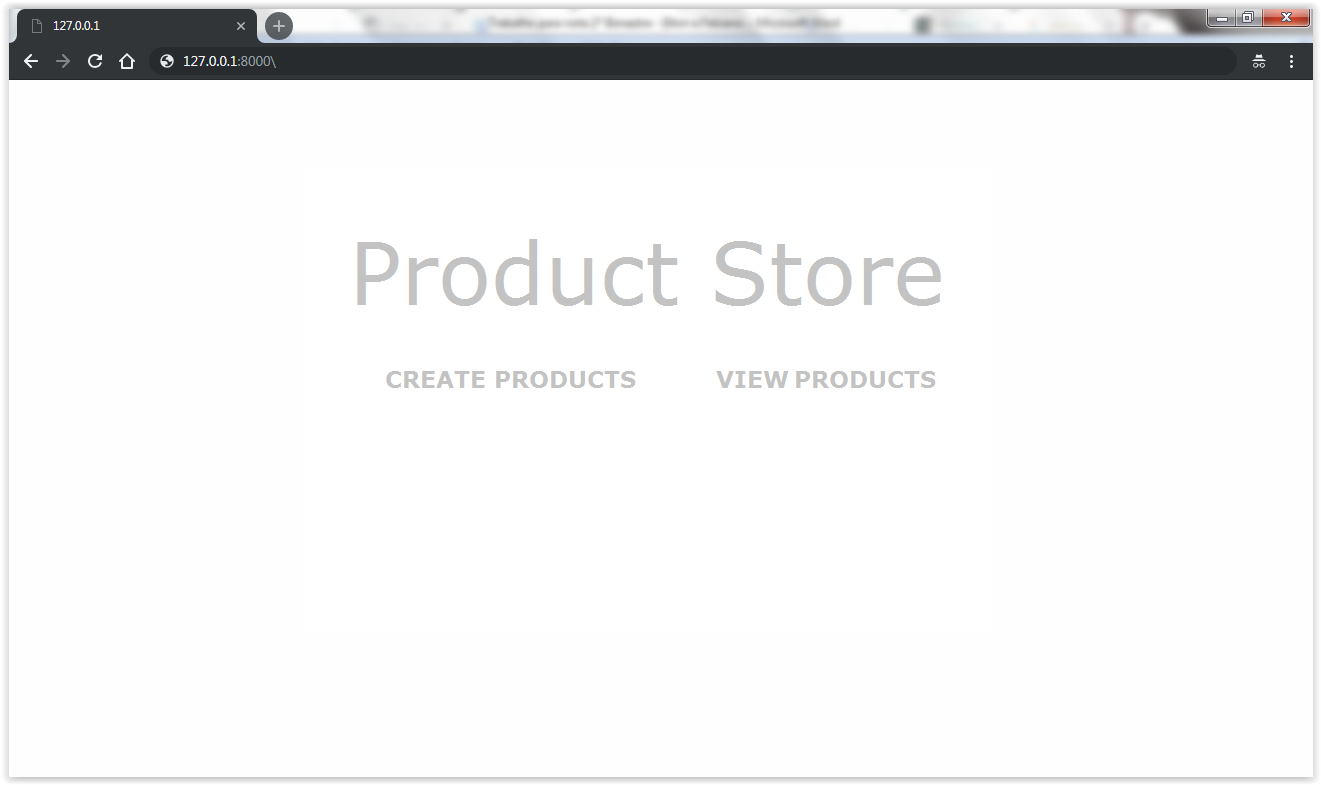
No Laravel, todas as V*iews* são armazenadas no diretório *resources/views*. Suas visualizações geralmente armazenam o *HTML* de sua página e são a camada de apresentação da arquitetura MVC.

O Laravel usa o *Blade* como um mecanismo de *templates*. O *Blade* é praticamente *HTML*, mas com alguma sintaxe semelhante a *PHP injetável*.

Para criar a visualização da página inicial. Atualize o arquivo *welcome.blade.php* no diretório *resources/views* para incluir o seguinte código dentro da *tag <body>* do *HTML* existente:



Verificando o arquivo *routes/web.php*, verá que a visualização *welcome.blade.php* deve ser renderizada para o usuário quando o (/) é visitado. Se você visitar o URL da página da Web http://127.0.0.1:8000/, verá esta página:



# **Conclusão**

Neste relatório, consideramos como o MVC funciona e como o Laravel o implementa. Nós consideramos por que você deve usar o MVC e como implementá-lo em um aplicativo Laravel do mundo real.

Utilizou-se de exemplos simples e claros para que o entendimento sobre os conceitos MVC no Laravel não saiam do foco.

Conclui-se que o uso do framework Laravel utilizando a arquitetura MVC traz inúmeros benefícios ao programador.

Dividir Funções no seu projeto é mais fácil.O MVC pode forçá-lo a dividir seus arquivos em diretórios lógicos, o que facilita a localização de arquivos ao trabalhar em grandes projetos.

.

# **Referências**

*Disponível em: <*<https://blog.pusher.com/laravel-mvc-use/>>*Acessado em <25/11/2018>*

*Disponível em: <*<https://laravel.com/docs/5.7>>*Acessado em <25/11/2018>*

*Disponível em: <*<https://www.devmedia.com.br/laravel-tutorial/33173>>*Acessado em <25/11/2018>*

*Disponível em: <*https://configr.com/blog/por-que-laravel-se-tornou-uma-das-melhores-opcoes-para-quem-trabalha-com-aplicacoes-em-php/>*Acessado em <25/11/2018>*

O código-fonte do artigo está disponível no GitHub:

<https://github.com/Fabianomsk/trabalho-web-fsclaro.git>