Fontes de Estudos: **Site Buteco Open Sourcer** escrito por *Jonathan Schweder.*

*Link url: https://blog.butecopensource.org/tutorial-laravel-5/*

**Instalação**

Antes de iniciar o primeiro projeto precisa instalar o **composer** no terminal.

**Criando um projeto**

Para criar um novo projeto usando o laravel 5, precisa digitar o seguinte comando no terminal.

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

**Explicações de diretórios**

*app/*  🡨diretório dos arquivos da sua aplicação

*boostrap/* 🡨 Arquivos de inicialização, são chamados em cada requisição ao servidor.

*Database/* 🡨 Aqui existem basicamente duas pastas, “**migrations**” que guardam os arquivos que fazem as alterações na estrutura do banco, como nome das tabelas, colunas, e etc, e “**seeds**” que seriam arquivos e classes que geram registros no banco de dados, seja para testes ou para possuir informações iniciais de utilização.

*public/* 🡨 são os arquivos públicos, normalmente é aqui que são colocados os arquivos de imagens, css e js.

*resources/* 🡨 arquivos de recursos como bibliotecas de JS e CSS, arquivos como arrays de traduções de mensagens, normalmente usado numa aplicação multilinguagem e as próprias views do sistema.

*storage/* 🡨 arquivos de cache, sessões (quando usado armazenamento em arquivo), views compiladas e logs.

*tests/* 🡨 Arquivos de testes do sistema.

*vendor/* 🡨 pasta criada pelo composer para controle e versionamento de bibliotecas.

**Configuração**

**Banco de dados**

O arquivos de configuração de banco de dados está na pasta **config/database.php**, abrindo esse arquivo você perceberá que basicamente retorna array, o laravel permite a utilização de vários bancos, você pode se conectar e utilizar diferentes conexões para diferentes bancos, inclusive mais de uma.

Neste ponto já percebe uma mudança de versões anteriores do laravel, os valores de **host, database, username, passwordN estão vindo de variáveis de ambiente,** isto permite uma maior flexibilidade, pois pode ter diferentes valores para produção e teste, estas variáveis são carregadas a partir do arquivo **.env** que esta localizado na raiz do projeto. Esse arquivo ele é oculto, no em tanto você precisa mostrar os arquivos ocultos no Windows. Altere as informações do arquivo **.env** com as informações do seu banco de dados.

**Artisan**

O laravel utiliza o artisan como programa de linha de comando para a execução de algumas tarefas digitando o comando abaixo na raiz do projeto você verá diversos comando possíveis e uma lista e uma breve explicação do que cada um faz.

*php artisan list*

A maioria dos comandos é quase autoexplicativa.

*php artisan migrate:install*

Este comando criará uma tabelano banco chamado **migrations** que o laravel utiliza para verificar quais as migrações que foram executadas e quais não, apenas deixando mais claro na prática uma migração é um arquivo que fica localizado **database/migrations** e faz para qualquer banco, como criar e alteras tabelas, mudar nomes de bancos e etc, isto vale para qualquer banco que o laravel tem suporte.

Após executar o comando acima aparecera uma mensagem *“Migration Table created successfully”* deve ser exibida, do contrário, possivelmente algum parâmetro foi passado errado no arquivo **.env**.

O Laravel 5 em especial já vem sua parte de autenticação pré-criada, mas para

Comandos em que o alex passo pelo whatsapp

vamos testar criando uma tabela, digita:

*php artisan make:model Carros –m*

você agora precisa dizer para o Laravel que tem migrations para atualizar

basta digitar: *php artisan migrate*

em up é definida as colunas da tabela

por exemplo:

*$table->increments('id');*

*$table->string('nome');*

*$table->string('email');*

*$table->text('mensagem');*

depois do migrate você deve indicar uma outra coisa já que é para atualizar uma tabela existente

php artisan migrate:refresh