Laravel

# Instalando

Para instalar o laravel é necessário antes instalar o composer: <https://getcomposer.org/download/>

Após instalar o composer, instale o laravel com o seguinte comando: composer global require laravel/installer.

# Iniciando Um Projeto Laravel

Para criar um projeto laravel pelo terminal, digite o comando laravel new nome\_do\_projeto dentro da pasta onde deseja que o projeto fique.

Após criar o projeto, acesse pelo terminal a pasta recém criada. O laravel possui muitos comandos que são dados pelo terminal, portanto ficaremos com o terminal aberto dentro da pasta do projeto.

# Criando A Logica de Autenticação.

O laravel possui um comando que cria automaticamente toda a logica necessária para realizar login e resgistro de usuários. Para criar a logica de autenticação, de o seguinte comando php artisan make:auth. Para a autenticação funcionar ainda é necessário configurar o banco de dados do projeto onde serão armazenadas as informações dos usuários.

# Configurando o Banco de Dados

O arquivo .env contem varias variáveis de configuração do projeto, dentre elas, as do banco de dados.

Altere as seguintes variáveis do arquivo .env para conectar ao banco mysql:

* DB\_DATABASE=nome\_do\_banco
* DB\_USERNAME=nome\_do\_usuario
* DB\_PASSWORD=senha\_do\_usuario

Depois de configurado o acesso ao banco dê o comando php artisan migrate para criar algumas tabelas predefinidas do laravel no banco de dados.

# Inicializando o Servidor Laravel

Para inicializar o servidor laravel dê o seguinte comando: php artisan serve. Abra o navegador com o link gerado pelo comando anterior.

# Arquivo de Rotas

Todos os links navegáveis de um site laravel devem estar cadastrados no arquivo de rotas web.php.

A classe Route contem vários métodos que representam as requisições http feitas ao servidor. esses métodos possuem dois parâmetros principais. O primeiro parâmetro representa a url da requisição feita. O segundo parâmetro é a função a ser executada para atender a requisição.

# O Padrão de Projeto MVC

O laravel organiza seu projeto utilizando o padrão MVC que separa a aplicação em três camadas: Model View e Controller. Resumidamente, a view é o que o usuário vê, isto é, as paginas html. O controller gerencia a comunicação entre o usuário, a aplicação e o model, isto é, atende às requisições e gera as respostas ao usuário. O model é quem faz o acesso ao banco de dados.

# View

As views são as páginas html do laravel. Essas páginas ficam dentro da pasta resources/views e possuem a extensão .blade.php.

A função view() retorna uma página com o mesmo nome que foi passado como parâmetro.

# Controller

Os controllers recebem as requisições do usuário e geram as respostas correspondentes. Os arquivos de rotas web.php é um tipo de controller. Porem para não sobrecarregar o arquivo de rotas e deixa-lo muito grande criamos outros controllers para delegar tarefas especificas a cada um deles. Os controllers ficam na pasta app/http/controllers. Para criar um controller pelo terminal, execute o camando php artisan make:controller nome\_do\_controller.

# Variáveis e valores

O usuário consegue enviar variáveis através da url. Para isso, ao cadastrar uma rota no arquivo web.php, as variáveis devem ser identificadas por nomes entre chaves. Essas variáveis serão enviadas como parâmetros à função que será executada.

A função view() pode receber um segundo parâmetro, que é um vetor de variáveis que serão enviadas para a página. Geralmente, usamos a função compact(), que cria um vetor com as variáveis passadas como parâmetros.

# O Blade

Como dito anteriormente, todas as views possuem a extensão .blade.php. O blade é um recurso do laravel que permite inserir códigos de programação nas views de modo simples. O blade é chamado de templating engine.

Os códigos em blade começam com um @ e são chamados de diretivas.

# Model

O Model representa uma entidade(tabela do banco de dados). Para criar um model execute o comando: php artisan make:model Produto --migration.

O Comando make:model cria a classe Produto detro do diretório App essa classe é a que acessa as informações da tabela produtos no banco de dados.

A opção --migration cria o arquivo create\_produtos\_table dentro do diretório database/migrations, que contem o schema da tabela produtos no banco de dados.

# Semeadores

Um semeador popula(preenche com registros) uma tabela no banco de dados. Para criar um semeador, execute o comando php artisan make:seeder ProdutoSeeder. Por convenção, o nome do semeador é nome do modelo seguido pela palavra Seeder. Por exemplo, se o modelo se chama Produto, o semeador sera chamado ProdutoSeeder. Esse comando cria a classe ProdutoSedder dentroo do diretório database/seeds.

O Semeador so sera executado se estiver declarado dentro da classe DatabaseSeeder. Para isso acrescente a linha $this->call([ProdutoSeeder::class]); dentro da função run() da classe DatabaseSeeder.

# Criando e Populando Tabelas no Banco de Dados

Primeiramente, para conectar o laravel ao banco de dados, preencha as configurações de banco de banco de dados no arquivo .env.

Para criar as tabelas(ou atualiza-las), execute o comando php artisan migrate:fresh --seed. O camando migrate:fresh apaga(drop) as tabelas, caso elas existam, e cria as tabelas definidas pelos arquivos de migração.

A opção --seed executa os semeadores(popula as tabelas recém criadas).