**Seção 5: Introdução as Rotas, Controllers e Views**

**Aula 23**

Para inicar a execução do Laravel, deve-se executar php -S localhost:\*porta disponível\* no console e deixa-lo executando para ser possível a sua utilização.

**Aula 24**

Mostrado o artisan, script responsavel por garantir execução de comandos uteis de maneira mais fácil, como por exemplo, basta executar o comando php artisan serve para rodar a aplicação http do laravel instalado com a porta padrão 8000, podendo ser alterada com o parâmetro --port=\*porta disponível\*.

**Aula 25**

Introução as rotas, que no ambiente laravel são segmentadas em 4 partes: API, channels, console e web. Cada uma possui sua própria função dentro do sistema a ser desenvolvido e serão utilizadas de forma mais prática futuramente.

**Aula 26**

Utilizada a rota Web para criar duas novas rotas, a de /sobre-nos e a /contato. É utlizado o método Route::\*método http\*($uri, $callback). O método http pode ser get, post, patch, put, delete, etc. Já a $uri é o local que deseja ser mapeado como no caso /sobre-nos e /contato. O $callback se trata do que será realizado quando a local for acessado, geralemente sendo uma função que retorna uma view.

**Aula 27**

Explicação breve sobre controladores. Se tratam da parte do código responsável pela lógica do negócio, após ser tatados pelas rotas que levarão até determinada página do site.

**Aula 28**

Adicionado os controllers da aplicação. Com o artisan, é possível criar um novo controller com o comando php artisan make:controller, que irá criar um novo arquivo no diretório /app/http/controllers/. Após a criação dos controllers, no primeiro momento foram criadas uma função por controller somente exibindo a mesma mensagem que já estava sendo trasmitida em forma de callback nas rotas. Agora nas rotas deve ser passada uma string, que será interpretada de forma a esperar o nome da controller e uma ação a ser realizada na chamada desse controller, isso no laraval 7 para trás. A nova utilização deve-se ser utilizado como [\*diretório\*::class, ‘\*funcão\*’].

**Aula 29**

Explicação sobre view. Se trata da visão produzida no lado do servidor responsável por exibir os elementos para o usuário. Também é dito que este modelo se trata do modelo tradicional de criação de sites, existindo um modelo mais moderno, tratando back e front-end com softwares diferentes, unidos por uma api.

**Aula 30**

Feita a criação das views do projeto. Estão presentes dentro da pasta \app\resources\views, em que é criada a pasta site que irá conter as 3 páginas criadas até o momento. Essas views irão conter o que será de fato exibido para o usuário.

**Seção 6: Avançando com Rotas (Routes)**

**Aula 31**

Mostrado como receber parâmetros em rotas. Deve-se criar uma nova rota que pode ser o mesmo endereço, porém com mais uma separação com / e entre {} é passado o parâmetro que com uma função de callcabk poderá ser utilizado como uma variável.

**Aula 32**

Exibido como fazer com que os parâmetros sejam opcionais e não obrigatórios. Para realizar isso, basta definir um safe null operator no final do parâmetro presente na rota e na função de callback, junto da declaração da variável deve ser definido um valor, que será tratado com o valor padrão, caso nenhum seja passado no parâmetro. O laravel possui a limitação de que o parâmetro opcional só pode estar em falta da direita para a esquerda, pois caso alguma parâmetro não seja passado, porém e sequente seja passado, a página não será encontrada.

**Aula 33**

Feita a aplicação de expressões regulares para tratar os parâmetros passados. Para fazer o tratamento dos parâmetros, ao final do método get do route, deve-se por a expressão ->where(), onde serão passados o nome do parâmetro a ser tratado e depois separado por vírgula as expressões regulares desejadas, ambos sendo contidos em strings.

**Aula 34**

Criada uma lista com os links das rotas criadas até o momento.

**Aula 35**

Criadas 4 novas rotas para login, clientes, produtos e fornecedores.

**Aula 36**

Realizado o agrupamento das rotas presentes na aula passada utilizando o método prefix do Route. Route::prefix(‘/app’)->group, esse método recebe uma função que deverá possuir as rotas que serão agrupadas por esse prefixo.

**Aula 37**

Atribuídos nomes as rotas definidas até o momento, que facilitam e muito em suas chamadas em links por exemplo, fazendo com que a chamada seja a mesma independente do diretório da aplicação, ou caso a rota em si seja alterada. Para definir um nome, basta ultilizar o método name->(‘\*nome do site\*’) no final da linha da rota que se deseja nomear.

**Aula 38**

Mostrado como realizar redirecionamento de rotas. Para isso, outras duas rotas foram criadas somente para teste. Existem duas maneiras de realizar o redirecionamento. Uma delas é por meio da função redirect()->route(‘\*rota destino\*’), que deve ser passado dentro da função de callback da rota original. Outra maneira de redirecionar uma rota é utilizando o método redirect de Route (Route::redirect(‘/rota2’, ‘/rota1’);).

**Aula 39**

Criada uma rota de fallback, que é acessada toda vez que a rota não é encontrada. Sua sintaxe é Route::fallback(function() {\*ação da rota fallback\*});