

## Model de Detalhe Agendamento

### Transcrição

No fim do vídeo há um corte que poderia não existir. O Marcelo altera o `string.Format()` para incluir o nome do veículo no alert da mensagem, porém esquece de colocar o novo valor no início dos argumentos, mantendo apenas os antigos. Isso, com certeza, gerou um erro. Acredito eu que vocês tenham cortado pra não deixar o vídeo ainda maior e eu até entendo isso, porém isso é algo que considero importante. Muita gente acaba tendo esse problema e terão se seguirem o vídeo sem olhar o código-fonte, uma vez que no vídeo não mostra esse argumento sendo adicionado.

Enfim, entendo que não vale a pena regravar isso. É mais uma visão sobre o que vale cortar e o que não vale.

Abraços.

[00:00] Agora que a gente está refatorando a code behind da página detalhe view, que a gente está movendo algumas coisas para o modelo veículo. A gente percebe que tem algumas coisas que ainda estão de fora, por exemplo, as propriedades tem o freio ABS, tem ar-condicionado, elas não estão ainda no nosso modelo do veículo.

[00:22] Então, o que a gente vai fazer é copiar para lá essas propriedades, para que o nosso modelo veículo também tenha essa informação de que ele tem esses acessórios incluídos. Então, esse vai modificar aqui o modelo e colocar três propriedades, uma delas vai ser o `TemFreioABS`.

[00:47] E aí, a outra propriedade vai ser também o (booleano), que é o `TemArCondicionado` e a outra propriedade vai ser tem `TemMP3Player`. Agora, o que a gente vai fazer? invés de acessar essas propriedades privadas aqui, tem freio ABS, tem ar-condicionado no code behind, a gente vai remover e vai acessar diretamente a partir do objeto veículo.

[01:34] Então eu vou colocar aqui, invés de return, eu tenho freio ABS, eu coloco `Veiculo.TemFreioABS`. Vou fazer isso aqui para as outras referências desse tem freio ABS. Aqui, eu removo essa referência privada e vou trocar aqui em baixo por tem ar-condicionado, que vem lá do objeto veículo e aqui embaixo também.

[02:05] Vai vir lá do nosso objeto veículo. Removo também o tem MP3 Player aqui e vou trocar por uma referência lá do veículo. Então é isso. Agora, eu vou ter que remover também essa propriedade valor total, que para a gente não faz sentido ter ela aqui no nosso code behind.

[02:33] Então, faz mais sentido ter esse valor total lá no nosso modelo, porque o veículo sabe, deveria saber como tratar, como calcular o preço total, de acordo com as propriedades, de acordo com as chaves, de acordo com as condições, se ele tem um acessório ou não, que ele vai acabar somando com o preço do veículo, com o preço base e vai retornar o valor total.

[03:03] Então aqui, eu vou substituir o valor total no retorno, eu vou substituir por... eu vou retornar, `Veiculo.PreçoTotalFormatado`. Essa propriedade ainda não existe, então a gente vai recortar isso aqui, colar lá no veículo. Então, vamos lá para o veículo, vamos criar uma nova propriedade que vai ter só um get, aqui.

[03:36] Então, vou lá para o veículo e vou colocar aqui: public, string e vou colocar, `PreçoTotalFormatado` e dentro do get eu coloco, vou dar aquele return e vou retornar aqui uma string com o preço total formatado do veículo. Aqui, eu removo a referência para veículo, porque eu não preciso dessa instância.

[04:04] O próprio veículo já tem essas informações, então eu estou removendo aqui. Então, voltando lá para o nosso detalhe view, aqui no code behind. Agora a gente roda a aplicação, vê se não quebrou nada, se continua funcionando, sempre bom a gente garantir que a aplicação funcione a cada mudança pequena que a gente faz na aplicação.

[04:39] Para a gente não ter uma surpresa maior lá na frente, então vamos lá. Rodando a aplicação, agora está rodando o emulador, vai abrir a aplicação. Então, aqui a gente tem os acessórios, conforme eu mudo aqui, altera com sucesso aqui em baixo. Então, continua a aplicação rodando sem problemas.

[05:05] Então, a gente está conseguindo refatorar, sem quebrar a aplicação. Muito bem. Então a gente conseguiu dar uma enxugada no code behind, no detalhe view e a gente vai agora para a terceira tela, que é o agendamento view. A gente vai lá no code behind ver o que a gente pode fazer para mover alguns dados lá para o modelo.

[05:28] Então, olha só, a gente tem aqui a referência para o veículo, nome, fone e Email. A gente também tem: data do agendamento e a hora do agendamento. Então, será que a gente consegue mudar, mover algumas propriedades aqui para algum modelo?

[05:43] Então, olha só. Eu tenho um veículo, até aqui tudo bem, mas eu também tenho nome, fone e Email do usuário, data do agendamento e hora do agendamento. Então, essas informações todas, elas poderiam estar numa outra entidade, numa outra classe, lá no nosso modelo.

[06:03] Essa classe representaria o quê? Ela representaria o agendamento que o usuário faz no seu smartphone. Então o que a gente vai fazer é criar uma classe chamada agendamento. Então, lá no nosso modelo, eu venho aqui e adiciono classe, coloco agendamento, ter uma classe

[06:25] Que é o nome da classe e aí, eu vou ter uma classe vazia, vou colocar como pública e aí, eu vou preencher essa classe, eu vou acrescentar algumas propriedades que vão conter, que vão armazenar o agendamento. Essas propriedades vão ser essas quadro: veículo, nome, fone e Email.

[06:48] Eu coloco aqui: veículo, nome, fone e Email e vou ter também a data do agendamento e a hora do agendamento. Então, eu copio também a data do agendamento e a hora do agendamento. Agora, o que eu vou fazer? Eu vou criar aqui uma nova propriedade para o agendamento view, que vai ser agendamento.

[07:21] Agendamento, agendamento. Agora, dentro de veículo, eu vou trocar aqui o get, aqui dentro eu vou colocar retornando uma referência para o agendamento, Agendamento.Veiculo e no set, eu retorno. Aliás, eu vou setar o veículo que está lá no agendamento. Então, eu coloco Agendamento.Veiculo = value.

[07:55] Formato aqui o nosso código, eu faço isso para as outras propriedades também. Aqui para o nome, eu vou colocar nome return, Agendamento.Nome. Aqui no set, vou fazer Agendamento.Nome = value. Aqui em baixo no fone, dou um return, Agendamento.Fone e aí, no set, eu coloco Agendamento.Fone = value.

[08:45] Aí, lá no Email, coloco get, Agendamento.Email e aí, no set, eu coloco aqui Agendamento.Email = value. Lá em baixo na data de agendamento, eu vou expandir aqui, vou remover essa linha, porque já tem o return agendamento, coloco Agendamento.DataAgendamento.

[09:27] E aqui na linha de baixo, no set, eu coloco Agendamento.DataAgendamento = value. E agora, na hora do agendamento, eu coloco no get: return Agendamento.HoraAgendamento e no set, eu modifico também para (setar) o valor lá no Agendamento.HoraAgendamento = value.

[10:00] Agora eu formato aqui o documento para ficar bonitinho com o Ctrl + K + D, formatou. É isso. Então, a gente conseguiu limpar aqui esse code behind. O que a gente vai fazer é rodar a aplicação, ver se não quebrou nada. Então está rodando o emulador, vai rodar a aplicação.

[10:20] Agora a gente vai lá para a terceira página. Vou digitar aqui Fulano de Tal no nome, vou colocar um telefone aqui: 2345678, ok. No Email: fulano@gmail.com. Agora, na data do agendamento, eu coloco semana que vem. Horário, eu vou colocar cinco horas da tarde e vou clicar em agendar.

[10:57] Apareceu agendamento, nome: Fulano de Tal, fone, E-mail, hora e data do agendamento. Correto. Bom, então agora, a gente quer colocar também na mensagem que vai ser exibida para o usuário, qual foi o veículo que ele selecionou. Então, eu vou colocar aqui na string, vou deixar veiculo: {0}, a posição zero.

[11:22] Aí, eu vou renumerar. Aqui no nome: {01}, no fone: {02}, no Email: {03}, na data do agendamento: {04} e na hora do agendamento: {05}. Só que, olha só, eu ainda não inicializei o nosso objeto agendamento. Eu tenho aqui a inicialização do veículo, só que esse veículo, na verdade, ele vai pegar lá do agendamento.

[11:42] Então, eu vou remover essa linha, vou colocar this.Agendamento, eu vou instanciar um novo agendamento, que é a nossa classe do modelo. E aí, eu vou fazer this.Agendamento.Veiculo = veiculo e vou rodar a aplicação. Vamos lá, vou preencher de novo o nome, o fone e o Email. Aqui, data e hora, não vou mexer.

[12:31] E vou agendar. Apareceu lá: agendamento, veículo, o nome, o telefone, o Email, a data e a hora do agendamento. Então, com isso, a gente conseguiu mover algumas coisas que estavam nos nossos views. Lá para o modelo, porque aí, a gente conseguiu separar.

[12:53] Aquilo que era regra de negócio, a gente moveu do nosso view. A gente conseguiu ter um desacoplamento de algumas funcionalidades da aplicação.