

## Lista Linq Order By

### Transcrição

[00:00] Agora que temos essa lista aqui dos agendamentos que foram gravados no banco de dados SQLite local do dispositivo, o nosso cliente, a Aluracar, pediu organizar essa lista, ordenar por ordem de data e hora do agendamento, como fazemos isso?

[00:22] Vamos aqui no `AgendamentoUsuarioViewModel` onde estamos pegando a lista dos agendamentos que estão lá no banco de dados, vamos fazer essa ordenação aqui depois que pegamos a lista do `AgendamentoDAO`.

[00:39] Vamos pegar a “`ListaDB`” e vamos ordenar essa lista, qual é a maneira mais fácil de fazer a ordenação aqui? Na minha opinião, utilizando o link, que são consultas integradas da linguagem, que faz parte do Framework do “.Net”, como eu ordeno uma lista?

[00:58] Eu faço a “`listaDB.OrderBy`” e aqui numa expressão lambda eu passo qual é a propriedade que eu quero ordenar, no caso vai ser o quê? Vai ser “`l`.” a data do agendamento, só que além da data do agendamento, temos que ordenar também pela hora, para desempatar a data e a hora eu preciso ordenar também pela hora do agendamento, “`orderBy`” data do agendamento e então eu quero, “`thenBy`”, eu também quero ordenar em seguida pelo hora do agendamento, “`l.HoraAgendamento`”.

[01:43] Eu vou armazenar o resultado em uma Query, em uma variável local, que eu vou chamar de “`query`”, “`var query = listaDB`” e a ordenação pela data e pela hora, agora, o que eu fazer?

[01:59] Eu vou pegar no laço “`foreach`”, em vez de pegar o item da `listaDB`, eu vou pegar o item da “`query`”, agora vamos rodar a nossa aplicação, vou entrar com usuário o “`joao@alura.com.br`”, senha “`alura123`”, vamos entrar, no menu eu vou entrar em meus agendamentos, ordenou pela data e pela hora, está certo agora.

[02:34] O que vamos fazer em seguida é que cada vez que entrarmos nessa lista e clicar num dos itens que estão em vermelho, ou seja, os itens que falharam o agendamento, queremos reenviar esses dados para o servidor da Aluracar.

[02:51] Por exemplo, se eu clicar nesse item Hilux 4x4, eu quero que ele exiba para mim uma mensagem perguntando se eu quero reenviar os dados e eu confirmando ele vai enviar, se tiver sucesso ele vai salvar no banco de dados da Aluracar, banco de dados remoto que está no servidor deles e na nossa lista o Hilux 4x4 vai ficar com a cor preta, quer dizer, ele vai indicar que teve sucesso no envio dessa vez, dos dados para agendamento.

[03:23] Para isso, o que eu vou fazer é entrar na página `AgendamentosUsuarioView` e aqui nesse `ListView`, que é onde tem a listagem dos agendamentos que foram gravados no banco de dados SQLite, eu quero fazer com que cada vez que um item seja selecionado, aquele item seja enviado.

[03:47] Para eu pegar o item que foi selecionado, o evento de seleção do item, o que eu tenho que fazer é definir uma propriedade do `ListView` chamado “`SelectedItem`” e aqui eu passo um binding, porque eu vou amarrar o item que está selecionado lá com nosso `ViewModel`, onde eu vou ter uma propriedade nova, indicando qual é o agendamento que foi selecionado.

[04:09] Eu passo aqui “`binding`” e em seguida eu coloco aqui o nome da nova propriedade que eu vou criar que é “`AgendamentoSelecionado`”, vou pegar esse nome e vou copiar e agora vamos lá para o `ViewModel` e vamos criar essa propriedade, eu crio a propriedade, vou colocar aqui “`propfull`”, a propriedade vai ser do tipo agendamento e vou chamar de “`AgendamentoSelecionado`”.

[04:45] Agora temos que modificar o “setter” dessa propriedade, eu vou modificar aqui, para quê? Temos que fazer com que o sistema envie uma mensagem, no caso de um agendamento selecionado, para podermos capturar em outro lugar para poder fazer o reenvio desses dados, eu preciso lançar uma mensagem nesse momento, passando dentro dessa mensagem qual é o agendamento que eu quero reenviar para o servidor da Aluracar.

[05:13] Para enviar uma mensagem eu tenho que utilizar o serviço de mensagem, `MessagingCenter.Send`, para enviar e aqui eu tenho que passar tipo do objeto que vai na mensagem, que é `Agendamento` e aqui dentro vai a instância do `AgendamentoSelecioneado`, aqui vai um novo tipo dessa mensagem, eu vou chamar esse tipo de `AgendamentoSelecioneado` e para capturarmos essa mensagem, para assinarmos essa mensagem, vamos onde?

[05:45] Vamos no Code Behind do `AgendamentoUsuarioView`, aqui nesse Code Behind vamos fazer a assinatura da mensagem do `AgendamentoSelecioneado`, para agendar, para assinar essa mensagem eu tenho que criar aqui um “override” do método “`OnAppearing`” e aqui fazer a assinatura da mensagem, “`MessagingCenter`.”, para assinar é `Subscribe` e eu passo o tipo do objeto que recebe na mensagem, que é o `Agendamento`.

[06:20] Aqui eu passo a instância que está assinando, que no caso é a instância “`this`”, a referência “`this`” e eu passo aqui o tipo da mensagem, que é “`AgendamentoSelecioneado`” e o método anônimo, que é uma função lambda, uma expressão lambda, que vai receber o objeto do agendamento selecionado e vai fazer o tratamento adequado.

[06:48] Eu passo aqui `Agendamento` e aqui dentro eu vou fazer o que eu preciso para poder reenviar os dados para o servidor, como eu faço para exibir uma mensagem pedindo confirmação para o usuário?

[07:04] Eu tenho que utilizar o `DisplayAlert` e eu vou passar aqui qual é o título da mensagem, vou passar aqui “Reenviar”, que vai ser o título do Messenger Box, do nosso Alerta, reenviar e aqui o texto, o corpo da mensagem que vai ser o quê?

[07:25] “Deseja reenviar o agendamento?” e eu tenho que passar também qual é o texto do botão “Ok” e “Cancel”, eu tenho o Ok e o Cancel, eu coloco que “sim” e outro botão “não”, o botão cancela eu coloco não.

[07:58] Agora eu também tenho que pegar o resultado dessa janela, dessa dialog de reenvio do agendamento, se o usuário confirmar ou se ele cancelar eu tenho que saber e armazenar isso em uma variável, eu vou fazer isso agora.

[08:13] “`var reenviar = DisplayAlert`”, só que esse `DisplayAlert` vai ser executado e o processo segue independente do usuário ter respondido já ou não, o que eu preciso fazer é aguardar, preciso esperar porque o `DisplayAlert` é um “awaitable”, ele é uma tarefa, ele é uma task, ele não vai, por padrão, esperar o resultado do usuário, a resposta do usuário, preciso forçar ele a esperar uma resposta do usuário para que eu possa pegar o resultado e tratar na ordem correta.

[08:59] Eu coloco aqui para aguardar a resposta, “`await`” e eu vou poder tratar o resultado que é a variável `reenviar`, só que quando eu coloquei “`await`” aqui ele está reclamando porque o meu método ele não é um “`async`”, eu preciso o meu método anônimo como “`async`” e agora sumiu o erro aqui embaixo.

[09:24] Agora eu vou tratar o resultado, que é o `reenviar`, a variável `reenviar`, “`if`” `reenviar`, se for verdadeiro, que eu quero reenviar, eu quero criar uma nova requisição post que vai fazer o envio dos dados lá para o servidor da Aluracar.