

Começando com o Core Data

Transcrição

[00:00] De volta ao projeto, agora nós precisamos dar continuidade no nosso formulário. Vou clicar aqui no “+”. Se eu criar um cadastro, escolhendo a foto do usuário, preenchendo os dados e clicar no botão salvar, até agora não vai acontecer nada, porque nós não fizemos nenhuma implementação.

[00:19] Então se eu escolher uma foto, digitar “João Silva”, “MASP” e colocar um telefone para ele, um site, vamos pôr o da Alura, e uma nota, o que precisa acontecer? Quando eu clicar em salvar, eu quero gravar esses dados no meu app, qual seria uma forma de fazermos isso? E conseguir exibir ainda na primeira tela?

[00:47] Se eu criar, por exemplo, uma classe aluno com todas essas propriedades, colocar essas informações dessa classe, gravar dentro de uma lista, eu consigo exibir na primeira tela. Mas qual é o problema dessa abordagem? Eu não vou conseguir persistir esse objeto no meu aplicativo, ou seja, quando eu rodar o aplicativo de novo, não vai estar mais lá, porque ele só vai existir durante o tempo de execução. E como nós fazemos pra salvar esses dados no nosso app?

[01:17] No iOS, nós temos uma forma nativa de se trabalhar com persistência, que é utilizando o Core Data. O Core Data nada mais é do que um framework que faz essa parte para nós de persistência de dados. A vantagem dele é que nós não precisamos saber nada de SQL, porque nós não precisamos digitar os comandos SQL para criar tabela, fazer o insert, ler os dados e tudo mais, ele mesmo gerencia pra nós. Vamos entender como funciona o Core Data.

[01:45] Eu estou com o link da própria documentação da Apple e tem o “Core Data stack”, que é importante nós entendermos como que funciona. Aqui nós temos esse “Object Model”, que faz parte do Core Data, que é onde nós configuramos o esquema da entidade que nós queremos gravar no banco, ou seja, o que vai ter essa tabela? Vai ter nome, endereço, notas, o que mais? Então é o esquema, qual é o tipo desses dados? É string, double, floatting? Então nós configuramos esse cara.

[02:16] Nós temos também o contexto. O contexto é esse “Object Context”, é o que faz para nós as operações de crud, então read, update, delete, é através desse “Object Context” que nós fazemos.

[02:30] E nós temos esse “Store Coordinator”, ele que, na verdade, dá acesso à persistência dos dados no device. Então esse é o Core Data stack, tem esses nomes que nós vamos ver na prática, que é mais fácil.

[02:45] Como faz pra nós utilizarmos o Core Data? O primeiro passo é ativarmos o Core Data quando vamos criar o projeto. Vou fechar aqui. Vamos supor que eu vá começar um novo projeto, o que eu preciso fazer? Eu preciso vir em criar novo projeto, single view application, aqui nós temos essa opção pra utilizarmos o Core Data, que é esse “Use Core Data”, se você quer trabalhar com persistência no seu app, é importante você deixar ativada essa opção.

[03:15] Como nós começamos o projeto já com a parte visual pronta, não vai ter necessidade de configurar isso agora, porque quando eu criei o projeto, eu já marquei a opção do Core Data.

[03:26] Quando nós marcamos essa opção, pra utilizarmos o Core Data no nosso app, se nós viermos no “AppDelegate”, ele já vem com algumas configurações implementadas, inclusive esse “import Core Data”. Aqui nós temos já alguns métodos, esse “persistentContainer”, tem o nome do container do nosso app e tudo mais. Tem o contexto, que é o cara que nós vamos precisar utilizar pra salvar esse cara realmente no nosso device.

[03:58] E aqui ele tem a propriedade contexto, que é do tipo “persistentContainer”, o que nós vimos na documentação, que é o que dá acesso pra conseguirmos salvar os nossos objetos no banco. Então vamos começar a implementar?

[04:11] O primeiro passo é virmos nesse arquivo “.datamodel”. Então quando nós usamos o Core Data, ele já vem com esse arquivo para nós utilizarmos. Como faz pra nós criarmos uma entidade? Aqui é bem intuitivo, não tem como errar. Aqui nós temos a opção de adicionar entidade, eu vou clicar nela e nós vamos criar, na verdade, uma entidade aluno. Criei a entidade, vou dar dois cliques e vou nomeá-la pra “Aluno”. Criei a identidade “Aluno”.

[04:42] O que nós precisamos colocar nela? Os atributos. E quais são os atributos? Nós vamos dar uma olhada aqui no formulário. O primeiro atributo que nós podemos colocar é o nome, que é uma string, então vamos colocar “String”. Qual é o próximo atributo? O próximo atributo é o endereço, então vamos vir em adicionar propriedade, “endereço”, também uma string.

[05:11] A próxima propriedade é telefone, então “telefone”, é uma string também. Nós temos o site do aluno, também é uma string. E uma nota. Nós vamos colocar aqui uma nota, só que a nota nós podemos pôr, por exemplo, “Double”.

[05:32] O que mais falta nós salvarmos? Falta a foto do aluno, que nós selecionamos através da câmera ou da biblioteca, que nós já aprendemos a utilizar. Vou criar uma nova propriedade que vai se chamar “foto”, vai ser a foto do aluno. Pra salvarmos a imagem é legal usarmos esse tipo, esse “Transformable”.

[05:54] Esse tipo, como o próprio nome diz, nós conseguimos transformar algum tipo de dado nessa propriedade e depois fazer o casting novamente pra recuperar. Então nós vamos salvar a UI Image nesse cara e depois nós fazemos o casting de novo pra recuperar.

[06:10] Agora que já criamos a entidade, nós podemos voltar no View Controller para começarmos a utilizar. E quando nós criamos uma entidade no Core Data, ele já cria como se fosse uma classe pra nós. Pra vermos melhor isso, vamos criar aqui uma action do botão salvar, que nós ainda não temos no nosso View Controller.

[06:32] Vou vir aqui, vou clicar no botão “salvar”, nós vamos dividir a tela e criar a action pra esse botão. Vou descer, cliquei, segurei a tecla “Ctrl” e puxei. Eu vou mudar o tipo connection pra “Action”, o type vai ser “UIButton” e aqui eu vou chamar de “button”, que é o tipo do componente visual que eu estou usando, “Salvar”, e vou dar um “Connect”. Já criamos a action do botão salvar.

[07:07] Vamos voltar no arquivo no View Controller, pra começarmos a implementação? O primeiro passo é criarmos como se fosse mesmo a instância de uma classe, como que nós fazemos? “let aluno”, nós chamamos a classe “Aluno”, então o Core Data meio que cria uma classe pra nós utilizarmos. Repara que nós não criamos classe nenhuma, só criamos a entidade lá.

[07:31] Pra inicializar essa classe, repara que ele pede aqui um contexto, que é essa última opção que eu vou utilizar. Esse cara é do tipo “ObjectContext”, nós já vimos esse cara no App Delegate, lembra? Se eu vier no App Delegate nós temos o contexto e é esse cara que nós precisamos utilizar, pra ele criar uma nova entidade pra nós. Então nós precisamos, de alguma forma, acessar o App Delegate e utilizar o contexto para inicializar a nossa classe, pra nós conseguirmos salvar.

[08:09] Vamos criar uma nova variável que vai se chamar contexto, então olha só, “var contexto”, pra acessarmos o contexto, nós vamos precisar importar o Core Data, então vou dar um “import Core Data”. Aí nós conseguimos utilizar esse tipo de variável, que é esse “NS ObjectContext”, é essa cara que nós vamos utilizar. E eu vou criar aqui uma variável computada.

[08:39] Agora nós precisamos acessar o App Delegate, como nós fazemos? Eu vou criar uma constante, “appDelegate”. Pra acessar o App Delegate, nós vamos fazer isso através dessa classe “UIApplication.shared.delegate”, e podemos dar um casting aqui, então “as! AppDelegate”.

[09:05] Agora nós precisamos retornar o contexto, que é desse tipo aqui, "ObjectContext". Como que nós fazemos isso? Vamos dar um "return", pegar esse "App Delegate" que criamos aqui, ".persistentContainer", que se nós viermos no App Delegate é esse cara aqui, "persistentContainer.viewContext", então vamos vir aqui, vamos dar um "viewContext". Qual é o tipo dela? "ObjectContext", então é isso mesmo que nós precisamos.

[09:38] Então aqui nós já temos acesso ao contexto. Eu vou descer aqui, como ele estava pedindo o contexto, não dava pra prosseguirmos sem criar o contexto, agora nós já conseguimos, então vai passar aqui o contexto.

[09:52] Agora nós precisamos serializar esse objeto que criamos do aluno, ou seja, tudo o que está no text field nós vamos setar aqui no aluno. Então eu vou pegar "aluno.nome" e aqui eu chamo "textFieldNome.text". Qual é o próximo? É o endereço, então "aluno.endereco", vai ser o "textFieldEndereco.text". Depois do endereço temos o telefone, então "aluno.telefone = textFieldTelefone.text". O próximo é o site, então "aluno.site = textFieldSite.text". Depois temos a nota.

[10:55] A nota, lembra que tínhamos colocado no schema da nossa entidade que ia ser um double? Então nós precisamos tratar aqui, porque o text field devolve uma string e não um double. Então vamos fazer o seguinte, "aluno.nota", eu vou transformar essa string, então vou pegar o "textFieldNota.text" e vou transformá-la num "NSString", porque a classe NSString, que vem do object c, tem essa propriedade ".doubleValue", então ele já consegue converter pra nós.

[11:34] E por último nós temos também a imagem. Não podemos esquecer da imagem. Então "aluno.foto" é igual ao outlet que nós temos aqui, "imageAluno.image".

[11:51] Será que só isso já basta pra nós salvarmos? Nós persistirmos esse objeto no nosso app? Ainda não. Sempre que formos fazer alguma operação no Core Data, precisamos chamar o contexto e chamar o método save. Então eu preciso do "contexto.save". Repare que esse save é uma throw function, ou seja, nós precisamos tratar caso dê alguma exception pra nós, então nós não conseguimos fazer isso, "contexto.save". E isso daqui nós precisamos colocar dentro de um "try".

[12:28] Só que pra colocar dentro de um try nós podemos usar um "do", aí nós tentamos fazer isso, vamos tentar salvar. Então vamos fazer o seguinte, vamos tentar salvar, e se conseguirmos salvar, o que eu vou fazer? Eu vou chamar o "navigationController" e vou voltar pra tela anterior, então ".popViewController", animado eu passo verdadeiro.

[12:53] Caso não dê, eu vou chamar o "catch" e eu vou dar um print do erro, então vou dar um "print(error.localizedDescription)", a descrição do erro. Agora sim nós conseguimos salvar. Como ele era uma throw function nós não conseguimos chamar o método save de uma vez.

[13:14] Vamos testar? Vamos rodar o aplicativo e vamos tentar cadastrar um aluno. Vou clicar no "+", nós já podemos colocar uma foto da nossa biblioteca, do próprio simulador. Então vamos abrir a biblioteca de imagens e depois nós conseguimos preencher o formulário e ver se estamos conseguindo realmente salvar.

[13:39] Selecionei uma imagem qualquer, vou salvar o meu aluno, João Silva, ele mora aqui no MASP, São Paulo. O telefone dele eu vou pôr qualquer um, só pra testarmos. Site eu vou pôr o da Alura. A nota dele eu vou pôr 8. Vou clicar em save, vamos ver o que vai acontecer.

[14:03] Cliquei no salvar, ele voltou pra tela anterior, ou seja, ele entrou aqui. Repara que aqui nós colocamos o "navigationController.pop", para ele voltar pra tela anterior, então ele não caiu no catch, ou seja, nós conseguimos salvar o nosso objeto aluno utilizando o Core Data. Daqui a pouco nós vamos aprender a listar esse aluno na primeira tela do nosso app.