

## Configurando mapa

### Transcrição

[00:00] Agora que nós já temos acesso ao aluno dentro desse arquivo, do “MapViewController”, é hora de utilizarmos as informações dele para mostrar o aluno no mapa.

[00:09] O primeiro passo para identificarmos qual vai ser a primeira tarefa que nós temos que fazer, é abrir o mapa, então vamos pensar. Cliquei no mapa. Como nós fazemos no iOS? Como que é visualmente que nós identificamos qualquer coisa no mapa? É através de um pino, não é verdade? Então o primeiro passo é criarmos um método pra criar esses pinos pra nós.

[00:31] Eu quero mostrar, por exemplo, onde está um aluno, eu preciso criar um pino com a localização dele e tudo mais, com um título, pra exibi-lo no mapa. Então vamos fazer isso? Vamos criar o método pra criar o pino?

[00:43] Esse “print” nós podemos tirar, que nós deixamos só pra testar. E agora aqui é onde nós vamos criar um método pra deixar o pino. Só vou criar aqui um marcador para deixar os nossos métodos, “Métodos”, e aqui embaixo eu vou criar um método que vai se chamar “configuraPino”. Ele vai me retornar um objeto do tipo “MKPointAnnotation”.

[01:09] Para nós termos acesso a esse método, nós precisamos importar um framework que chama Map Kit, porque senão eu nem tenho referência dele, MKPoint, está vendo? Não aparece nada. Então eu preciso importar esse framework, “import MapKit”, eu tenho acesso ao pino. Aqui embaixo “MKPointAnnotation”, eu vou devolver um pino para ele.

[01:34] O primeiro passo é criarmos o pino, então “let pino” é do tipo “MKPointAnnotation”. E eu já vou dar um “return” no pino pra ele não ficar apontando nenhum erro pra nós. Vou dar o “return pino”.

[01:50] O pino nós podemos configurar algumas coisas como, por exemplo, o título, a localização em que ele vai ficar e tudo mais, então vamos fazer isso, “pino.title”, eu quero exibir, por exemplo, o título do aluno. O nome do aluno, na verdade. Pra termos acesso a isso, eu vou pedir como parâmetro que me passe um título, que vai ser uma “String”, então aqui eu seto o “titulo”.

[02:14] O outro parâmetro que nós precisamos configurar no pino é o seguinte, “pino.coordinate”, que é a coordenada geográfica desse pino. Então eu preciso pedir também aqui no parâmetro qual vai ser a “localizacao”, que é do tipo “CLPlacemark”. Tendo a localização eu posso fazer o seguinte, eu pego a “localizacao.location.coordinate”, eu tenho acesso a localização onde o pino vai ficar. Com essas duas informações eu já consigo criar o pino.

[02:50] Agora sim, nós precisamos pedir pro mapa abrir em alguma localização específica. Por exemplo, quando eu abrir o nosso mapa, legal seria que mostrasse a Caelum como localização central, e em volta os alunos que nós queremos verificar se moram ao redor ou não, então nós precisamos setar uma localização inicial para o mapa.

[03:11] Só que nós não temos acesso ao mapa ainda, nós não criamos nenhum outlet para o mapa, então nós precisamos criar um outlet, assim nós conseguimos acessar as propriedades do mapa, então vamos criar um novo outlet, “IBOutlet”.

[03:28] Para fazer isso eu vou abrir o storyboard e vou fazer o outlet. Como é que eu faço? Vou segurar a tecla “Ctrl”, vou arrastar para dentro do View Controller e eu vou chamá-lo de “mapa”. Agora que nós já temos acesso, eu vou voltar no View Controller e vou criar um método pra setar essa configuração inicial. Vou vir aqui embaixo, “func localizacaoInicial”. Para não esquecer, eu já vou chamá-la no viewDidLoad, então “localizacaoInicial”.

[04:07] Quando nós trabalhamos com o Waze, nós criamos uma classe em que nós convertíamos um endereço, que era uma string, em coordenadas geográficas, e nós vamos utilizá-la novamente. Eu vou chamar a classe “Localizacao”, vou instanciar e vou chamar o método “converteEnderecoEmCoordenadas”. Para localização inicial eu vou querer que ele focalize o mapa na Caelum, então eu vou colocar “Caelum - São Paulo”, e ele vai me retornar uma localização, então “localizacaoEncontrada”.

[04:42] Com a localização, nós podemos criar um pino pra Caelum. Vamos fazer isso? “let pino = self.configuraPino”. Qual vai ser o título? O título vai ser “Caelum”, e qual vai ser a localização? A “localizacaoEncontrada”.

[05:00] Agora que eu já tenho o pino, eu preciso criar uma região onde vai focalizar ali o mapa, qual vai ser a distância que eu quero de zoom e tudo mais. Então nós vamos criar uma região, “let regio”. Para criar a região, eu vou utilizar a classe “MKCoordinate” com distância, então essa primeira classe aqui mesmo.

[05:21] O que eu preciso passar? Eu preciso passar primeiro as informações do pino, então “pino.coordinate”, que é a localização dele e aqui eu passo uma unidade de medida pra dar um zoom no mapa. Eu vou pôr, por exemplo, “5000” e a distância daqui também, “5000”.

[05:41] Com a região, o próximo passo é utilizarmos o outlet do mapa pra setarmos essas informações que nós criamos. Então vamos fazer isso. Vou chamar aqui o mapa, como eu estou dentro de uma closure eu preciso do “self”, então “self.mapa.set”, aqui eu vou setar a região, é a região que eu criei, então eu vou dar um “true”, porque eu quero que quando ele abra o mapa, ele focalize de forma animada.

[06:06] Agora eu preciso adicionar o pino do mapa, o pino que nós criamos no mapa. Como eu vou fazer isso? “self.mapa.addAnnotation”. Como o mapa é do tipo MKAnnotation, nós o passamos aqui, vou passar aqui o pino. Então nós já temos uma região e já temos um pino. Vamos testar isso no simulador? Pra ver se o mapa já está aparecendo na localização que nós setamos, que é na Caelum. Vou gerar um build pra nós testarmos.

[06:35] Com o simulador aberto eu vou fazer um long press em cima do João e vou escolher a opção “localizar no mapa”. Ele vai vir pra cá e trouxe a localização central aqui, que é a Caelum. Para eu dar zoom no mapa eu posso segurar a tecla “Alt” do teclado, o optional, e eu posso clicar e arrastar, então eu consigo dar um zoom aqui. Ele está exibindo exatamente a localização da Caelum.

[07:03] Só que tem um problema, eu cliquei em cima do João e ele não mostrou onde está o João, então nós precisamos agora criar outro pino para localizar esse aluno. Vamos fazer isso?

[07:14] Aqui embaixo eu vou fazer o seguinte, eu vou criar um método que vai se chamar “func localizarAluno”. Vai ser parecido com a implementação que nós fizemos da configuração inicial, eu vou chamar a classe “Localizacao”, vou chamar o método “converteEnderecoEmCoordenadas”. Qual vai ser o endereço? Vai ser o endereço do aluno. Como o aluno está optional, eu vou fazer uma verificação aqui, “if let aluno”, é igual ao “aluno,” que ele está optional, aí sim eu vou entrar e vou fazer essas configurações.

[07:56] Aqui eu vou passar o “aluno” que eu criei no “if let”, “endereço” e ele vai me dar uma localização, então “localizacaoEncontrada”. Agora que eu tenho já a localização do aluno, eu preciso criar um pino pra ele também, então vou fazer o seguinte, “let pino = self.configuraPino”, o título desse pino vai ser o nome do aluno, então aqui eu vou passar “aluno.nome” e a localização vai ser “localizacaoEncontrada”.

[08:39] Agora que eu tenho o pino, o que eu preciso fazer? Eu preciso adicionar esse pino no mapa, então eu vou chamar o mapa, “self.mapa.addAnnotation” e aqui eu passo o “pino”. Agora eu preciso chamar essa função, então eu vou vir aqui, “localizarAluno”.

[09:00] Vamos rodar o app de novo pra nós testarmos? Agora nós vamos testar. Eu vou fazer um long press em cima do João e vou dar um “localizar no mapa”. Então ele exibiu aqui a Caelum e aqui tem outro pino. Aparentemente está sem o título, mas aqui é um bug, se eu der um zoom aparece o título, que é o nome que nós colocamos do aluno.

[09:26] Então aqui nós conseguimos localizar a Caelum, que é onde nós definimos a região central, e aqui está o aluno João, que é onde nós criamos o pino dele. Essa é a implementação que nós fizemos inicial do mapa.