

 09

Rodando no iOS

Transcrição

Agora, queremos rodar o projeto do *Garconapp* igualmente como fizemos antes, mas desta vez no iOS. Ponto importante: precisaremos utilizar um Mac, tanto para rodar o emulador do IOS, como para rodar no próprio dispositivo IOS. Ter um Windows e um iPhone não irá funcionar. Você precisará ter um Mac e um IPhone, por exemplo.

Quando estiver no seu Mac, você deverá ir para a *App Store* e procurar pelo [Xcode](#) (https://developer.apple.com/xcode/download/*Xcode*). Ele é a ferramenta de desenvolvimento da Apple, o Download é gratuito, você terá apenas que fazer a instalação. Eu já tenho instalado, vou abri-lo no meu Mac.

O próximo passo é ir no terminal e instalar dois pacotes no `npm`: o `ios-sim` e o `ios-deploy`.

```
garconapp $ sudo npm install -g ios-sim ios-deploy
```

O primeiro deles trabalha com o emulador e o outro com o dispositivo real. O Cordova precisará deles para conseguir executar. Damos um *Enter*, depois irei digitar a minha senha e ele irá fazer a instalação.

```
garconapp $ sudo npm install -g ios-sim ios-deploy  
Password:  
garconapp $
```

Temos instalado o Cordova, o Xcode e os dois pacotes. Em seguida, irei adicionar a minha plataforma iOS ao meu projeto Cordova. Vamos adicionar ao código:

```
garconapp $ sudo npm install -g ios-sim ios-deploy  
Password:  
garconapp $ cordova platform add ios
```

Quando adicionarmos a plataforma, ele irá baixar a versão mais recente e compatível com o projeto, depois a instalação.

```
garconapp $ sudo npm install -g ios-sim ios-deploy  
Password:  
garconapp $ cordova platform add ios  
Adding ios project...  
Running command: /Users/alura/.cordova/lib/npm_cache/cordova-ios/3.9.2/package/bin/create /User:  
iOS project created with cordova-ios@3.9.2  
Installing "corodova-plugin-whitelist" for ios
```

This plugin is only applicable for versions of cordova-android greater than 4.0. If you have a |

```
garconapp $
```

Próximo passo, queremos rodar o projeto, primeiramente no emulador do sistema. Para isto, vamos no *Xcode*, clicamos no menu e depois em: *Open Developer Tool* e *Simulator*. Você pode baixar emuladores. Em *Preferências* e depois em *Downloads*, veremos que eles tem vários simuladores e versões diferentes - geralmente, ele traz as mais recente, por padrão. O Download é bastante simples.

Com emulador instalado, vamos para o Terminal:

```
garconapp $ cordova run ios
```

Depois disso, ele irá fazer o build do projeto nativo para IOS. O processo levará alguns instantes e ele executará um emulador local. Eu já tenho um emulador dentro da minha máquina. Não estou utilizando um aparelho móvel.

O emulador irá subir o app, que será aberto automaticamente e veremos a mensagem "OI CORDOVA". Então, o app Cordova multiplataforma roda no emulador dentro do Mac.

Iremos em seguida testar em um dispositivo iOS. Eu tenho um iPod Touch, iOS 9, e quero executar o app nele. Primeiro, iremos conectá-lo com o cabo USB ao Mac, para mostrar como é a tela do meu celular. Como faremos para rodar o app? Como o ambiente configurado apenas utilizar o comando `run` não seria suficiente. Por questões de segurança, precisaremos também da chave de assinatura digital do aplicativo. Você não conseguirá fazer isto facilmente ela linha de comando. Será mais fácil fazer pelo Xcode. Iremos abri o projeto Cordova dentro do Xcode. Quando o Cordova adiciona a plataforma iOS, ele gera um Xcode válido. Na pasta de `garconapp` clicaremos em `ios` e depois no `Garçonete Só de Cenoura.xcodeproject`.

Após abrirmos o arquivo no *Xcode*, veremos que tem códigos e configurações de iOS. Em seguida, podemos escolher o emulador. Temos várias opções de iPad, iPhone. Na parte de *Device*, irá aparecer o meu aparelho *iPod touch de Sergio*. Vamos selecioná-lo. Precisamos ainda da assinatura digital. Para isto, vamos selecionar o projeto no Menu lateral e ele irá pedir abrir uma tela com configurações. Na parte **Identidade ("Identity")*, ele irá pedir que o desenvolvedor adicione um time ("Team"). Precisaremos de um usuário da Apple e uma senha, e depois, logar. Temos a opção de adicionar um usuário. Após preenchermos os dados, podemos adicionar o nosso usuário do campo "Team". Ele irá perguntar se queremos utilizar o "provision profile" associado ao time. Para aceitar, vamos clicar em "Fix Issue". Vale ressaltar, que cada vez que um usuário for configurado, o campo "Bundle Identifier" deverá ser preenchido com um `id` único. Se você usar o meu (`org.sergiolopes.garconapp`), não irá funcionar no seu celular, porque ele está vinculado a minha conta. No nosso exemplo, iremos preencher com um usuário qualquer e clicar em "Fix Issue". O Xcode irá registrar o `id` único com o usuário e agora, ficaremos capacitados para rodar no dispositivo iOS.

Em seguida, iremos clicar no ícone do "Play", localizado na parte esquerda da barra superior, ele irá fazer o build e instalar o app no aparelho. Após a instalação, ele irá começar a rodar no meu aparelho.

Revisando: para executar em um emulador local, você não terá grandes dificuldades e não precisará de assinatura. Para executar em um aparelho, você precisará assiná-la primeiro. O modo mais fácil de fazer isto é através do Xcode, que em sua última versão, você irá logar com seu usuário da Apple, configura a opção "Team" com a sua conta, clica em "Fix Issue" e ele irá gerar o certificado e já estará habilitado para rodar no aparelho.

Na época do iOS 8, Xcode 6, era preciso uma conta paga da Apple para executarmos os aplicativos, inclusive no aparelho de teste. A partir do Xcode 7, deixou de ser necessário o uso de uma conta paga. Com uma conta gratuita é possível executar o app no seu dispositivo. Veremos adiante o que é necessário para a publicação na loja (neste caso, será preciso ter uma conta paga). Por enquanto, é suficiente a conta gratuita.

Em seguida, iremos desenvolver o nosso aplicativo.

