

01

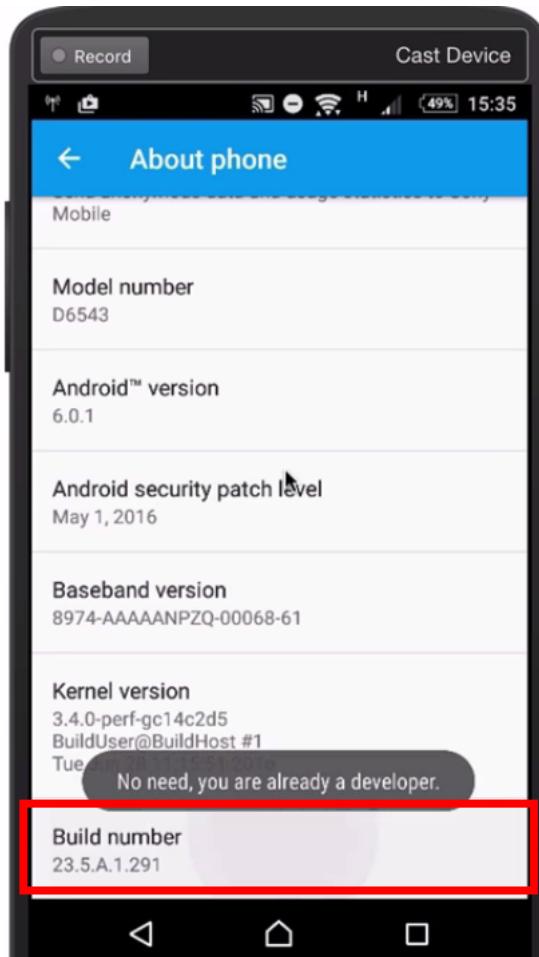
Emulação da aplicação direto no celular Android

Transcrição

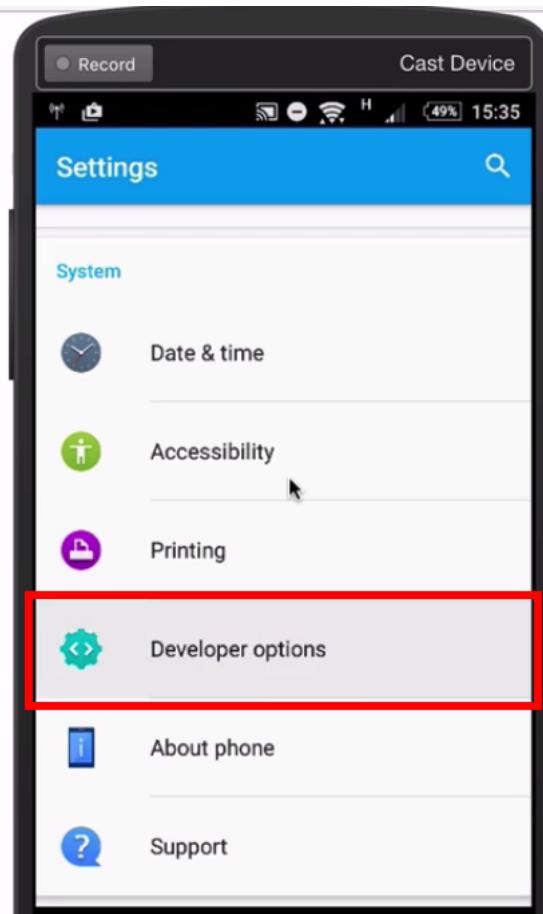
Agora que já construimos a tela em que o usuário poderá editar os dados, adicionaremos uma funcionalidade de câmera para que o usuário possa colocar uma foto no cadastro. Porém, esta funcionalidade não é nativa do Ionic. É necessário instalar um plugin.

Nós não conseguiremos emular esta funcionalidade pelo browser, então, teremos que rodar direto no celular. Vou mostrar como podemos fazer isso. Começaremos com o Android.

O primeiro passo será deixar o celular no modo de desenvolvedor. Acessaremos as "Configurações" do aparelho, depois clicaremos em "About phone". Vamos procurar a opção "Build number" e selecionaremos a opção **sete** vezes e ele te permitirá ser desenvolvedor no celular.



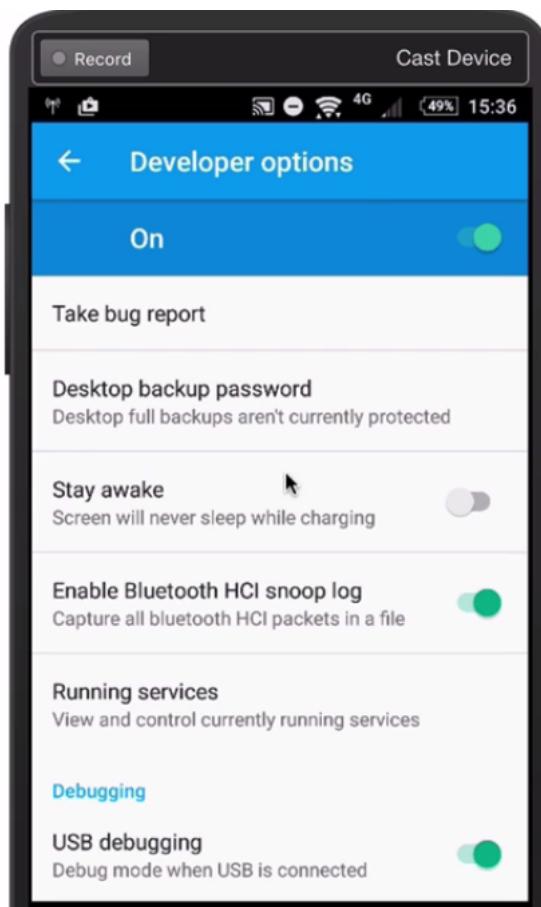
Ele mostrou a mensagem avisando que eu já sou desenvolvedor. Em seguida, será habilitada a opção "Developer options". A partir de agora, conectando o celular com um cabo USB, você conseguirá instalar um aplicativo diretamente no celular fora da loja.



Precisamos deixar o celular pronto para receber o aplicativo. Teremos que adicionar a plataforma no Alura Car, para isto, usaremos o seguinte comando:

```
AluraCar ionic cordova platform add android
```

Com esse comando, adicionaremos a plataforma que desejamos.



Ele criou uma pasta chamada `platforms` . Podemos ver usando o comando `ls` para que ele faça uma listagem. Em seguida, digitaremos `cd platforms` .

```
[→ AluraCar cd platforms
[→ platforms ls
  android   platforms.json
[→ platforms ]
```

Ele conseguiu instalar a plataforma `android` . Caso você decidir instalar o `iOS`, ele aparecerá também, ao lado no Android.

Queremos fazer o build, o **arquivo final**. Arquivos Android tem a extensão `.apk` e será o formato que subiremos para um loja Google Play. Vamos gerar o build, usando o seguinte comando:

```
ionic build android
```

Ele criará o build com sucesso.

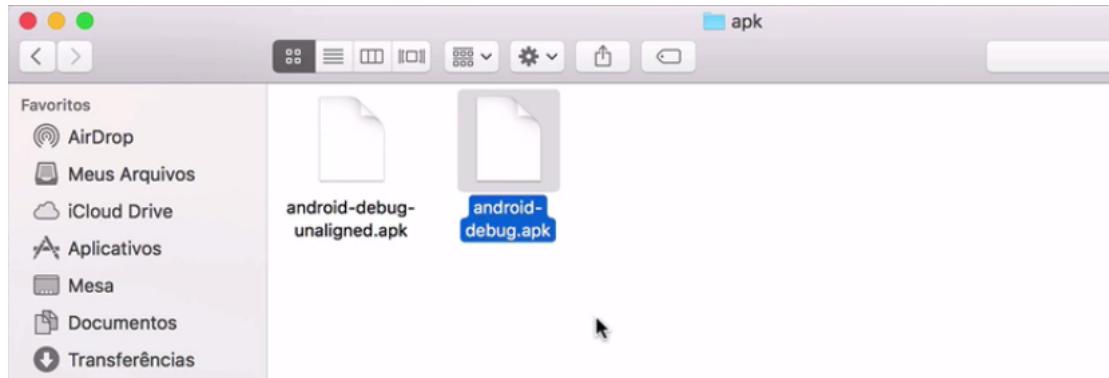
```

:mergeDebugAssets
:generateDebugResValues
:generateDebugResources
:mergeDebugResources
:processDebugManifest
:processDebugResources
:generateDebugSources
:incrementalDebugJavaCompilationSafeguard
:compileDebugJavaWithJavac
:compileDebugJavaWithJavac - is not incremental (e.g. outputs have changed, no previous execution, etc.).
:compileDebugNdk UP-TO-DATE
:compileDebugSources
:prePackageMarkerForDebug
:transformClassesWithDexForDebug
To run dex in process, the Gradle daemon needs a larger heap.
It currently has approximately 910 MB.
For faster builds, increase the maximum heap size for the Gradle daemon to more than 2048 MB.
To do this set org.gradle.jvmargs=-Xmx2048M in the project gradle.properties.
For more information see https://docs.gradle.org/current/userguide/build_environment.html
:mergeDebugJniLibFolders
:transformNative_libsWithMergeJniLibsForDebug
:processDebugJavaRes UP-TO-DATE
:transformResourcesWithMergeJavaResForDebug
:validateDebugSigning
:packageDebug
:zipalignDebug
:assembleDebug
:cdvBuildDebug

BUILD SUCCESSFUL

```

Vamos ver pelo Finder onde ele guardou o arquivo: AluraCar -> platforms -> android -> build -> outputs -> apk. O nosso arquivo será o android-debug.apk :



Esse arquivo podemos subir tanto na loja, como no celular. Nós queremos rodá-lo no nosso dispositivo, que está conectado com um cabo USB ao computador. Usaremos o comando abaixo:

```
ionic run android --device
```

Lembrando que este é o comando para rodar no dispositivo e não em um emulador que você tenha instalado em sua máquina. É preciso estar plugado em um cabo USB para que ele funcione.

O comando irá rodar perfeitamente no computador e aplicação funcionará diretamente no celular Android.



Avançaremos a seguir.