Use a Cabeça!
Desenvolvendo para Android

Dawn Griffiths
David Griffiths

ALTA BOOKS
EDITORA
Rio de Janeiro, 2016

Autores do Use a Cabeça! Desenvolvendo para Android

Dawn Griffiths

David Griffiths

Dawn Griffiths começou sua vida profissional como matemática em uma importante universidade do Reino Unido, onde obteve grau de excelência em matemática. Passou a se concentrar na carreira de desenvolvimento de software e tem 20 anos de experiência prática na área de TI.

Antes de escrever este livro, Dawn foi autora de outros três da série *Use a Cabeça!,* incluindo o *Use a Cabeça! Estatística* e o *Use a Cabeça! C* — publicados pela Editora Alta Books —, tendo também trabalhado em vários outros da série.

Quando Dawn não estiver trabalhando em livros da série *Use a Cabeça!*, você a encontrará aprimorando suas habilidades de Tai Chi, lendo, correndo, fazendo renda de bilros ou cozinhando. Ela aprecia especialmente passar o tempo com seu maravilhoso marido, David.

David Griffiths começou a programar com 12 anos de idade, quando assistiu a um documentário sobre o trabalho de Seymour Papert. Aos 15, escreveu uma implementação da linguagem de computação LOGO, de Papert. Após estudar matemática pura na universidade, começou a escrever código para computadores e artigos de revista. Trabalhou como *agile coach*, desenvolvedor e garagista, mas não nessa ordem. É capaz de escrever código em mais de 10 linguagens e prosa em apenas uma. Quando não está escrevendo, codificando ou instruindo, passa grande parte de seu tempo livre viajando com sua adorável esposa — e coautora — Dawn.

Antes de escrever este livro, David foi autor de três outros da série *Use a Cabeça!,* incluindo o *Use a Cabeça! Rails* e o *Use a Cabeça! C* — publicados pela Editora Alta Books.

Você pode encontrar os autores no Twitter, em *https://twitter.com/ HeadFirstDroid* — conteúdo em inglês.

Conteúdo (Sumário)

Conteúdo (a coisa real)

Introdução

Seu cérebro no Android. Aqui, você está tentando aprender algo, enquanto ali seu cérebro está fazendo um favor a você, garantindo que o aprendizado não permaneça. Se cérebro está pensando: "É melhor deixar espaço para coisas mais importantes, como quais animais selvagens evitar e se fazer snowboard pelado é uma má ideia." Então, como você engana seu cérebro, fazendo-o pensar que sua vida depende de saber como desenvolver aplicativos Android?

1 Mergulhe

O Android conquistou o mundo.

Todo mundo quer um smartphone ou um tablet, e os dispositivos Android são muito populares. Neste livro, vamos ensiná-lo a **desenvolver seus próprios aplicativos** e vamos começar fazendo com que você construa um aplicativo básico em um Dispositivo Virtual Android. Ao longo do caminho, você vai conhecer alguns dos componentes básicos de todos os aplicativos Android, como as atividades e os layouts.

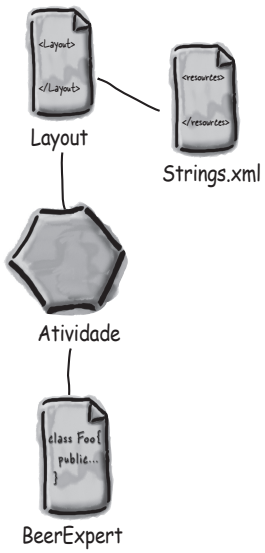
Tudo que você precisa é de algum conhecimento sobre Java...

Construção de aplicativos interativos

2 Aplicativos Que Fazem Algo

A maioria dos aplicativos precisa responder ao usuário de algum modo.

Neste capítulo, você vai ver como pode tornar seus aplicativos *um pouco mais interativos*. Como fazer com que seu aplicativo **faça** algo em resposta ao usuário e **como fazer com que sua atividade e seu layout se comuniquem** como velhos conhecidos. Ao longo do caminho, o levaremos um pouco **mais fundo** sobre **como o Android realmente funciona**, apresentando o **R**, a joia escondida que junta tudo.

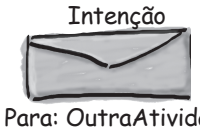


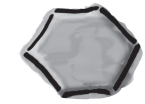
várias atividades e intenções

3 Declare Sua Intenção

A maioria dos aplicativos precisa de mais de uma atividade.

Até agora, vimos apenas aplicativos com uma atividade, o que está bem para aplicativos simples. Mas quando as coisas ficam mais complicadas, ter apenas uma atividade não resolverá. Vamos mostrar **como construir aplicativos com várias atividades** e como fazer seus aplicativos se comunicarem usando **intenções**. Veremos também como as intenções podem ser usadas para **ultrapassar os limites de seu aplicativo e fazer com que atividades em outros aplicativos de seu dispositivo executem ações**. As coisas ficam muito mais poderosas...

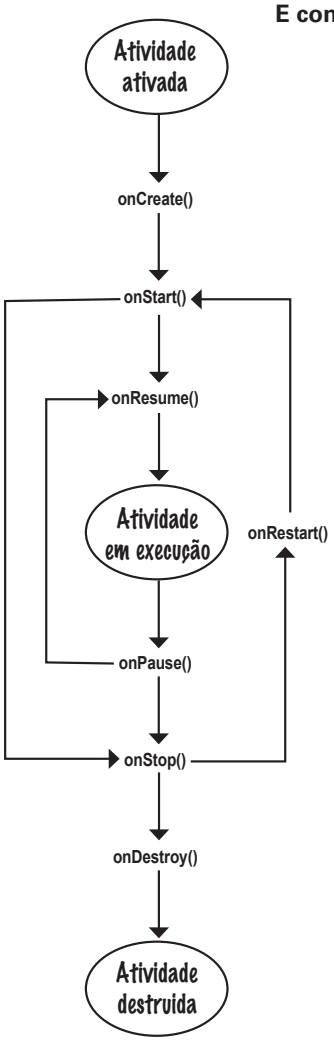




o ciclo de vida da atividade

4 Sendo uma Atividade

As atividades formam a base de todo aplicativo Android. Até aqui, vimos como criar atividades e fizemos uma atividade iniciar outra usando uma intenção. Mas *o que está realmente acontecendo debaixo do capô?* Neste capítulo, vamos nos aprofundar **no ciclo de vida da atividade.** O que acontece quando uma atividade é **criada** e **destruída**? Quais métodos são chamados quando uma atividade *se torna visível e aparece no primeiro plano* e quais são chamados quando ela perde o foco e é oculta? **E como você salva e restaura o estado de sua atividade**?

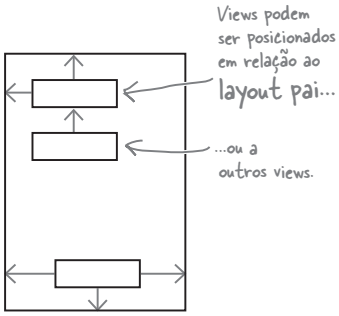


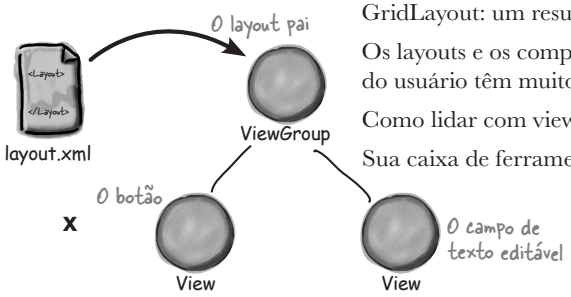
a interface do usuário

5 Aproveite a Vista

Vamos encarar os fatos, você precisa saber como criar excelentes layouts.

Se você está construindo aplicativos que deseja que as pessoas usem, precisa garantir que eles **tenham exatamente a aparência desejada**. Até aqui, apenas arranhamos a superfície no que diz respeito a criar layouts; portanto, é hora de *examinar um pouco mais a fundo*. Vamos apresentar mais **tipos de layout** que podem ser usados e também o conduziremos a um passeio pelos **principais componentes de interface gráfica do usuário** e sobre *como utilizá-los*. Ao final do capítulo, você verá que, mesmo parecendo um pouco diferentes, todos os layouts e componentes de interface gráfica do usuário têm ***mais em comum do que poderia imaginar***.



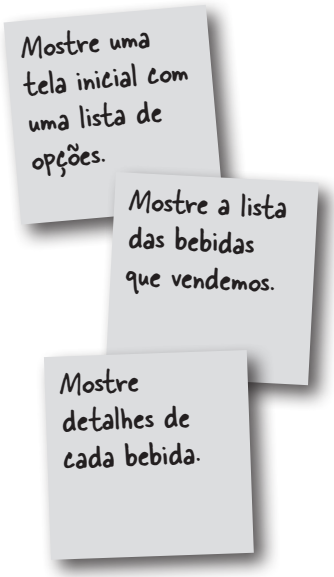


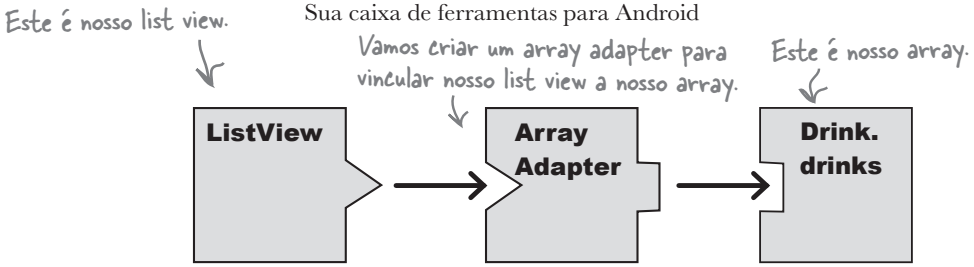
6 list views e adaptadores

Organize-se

Quer saber qual é a melhor estrutura para seu aplicativo Android?

Você aprendeu sobre alguns dos elementos básicos utilizados para construir aplicativos e agora é *hora de se organizar*. Neste capítulo, mostramos como você pode pegar várias ideias e ***estruturá-las em um aplicativo impressionante***. Vamos mostrar como **listas de dados** podem se transformar na parte fundamental do projeto de seu aplicativo e como **vinculá-las** pode criar um *aplicativo poderoso e fácil de usar*. No processo, você vai ver como o uso de **receptores de evento e adaptadores** tornam seu aplicativo mais dinâmico.



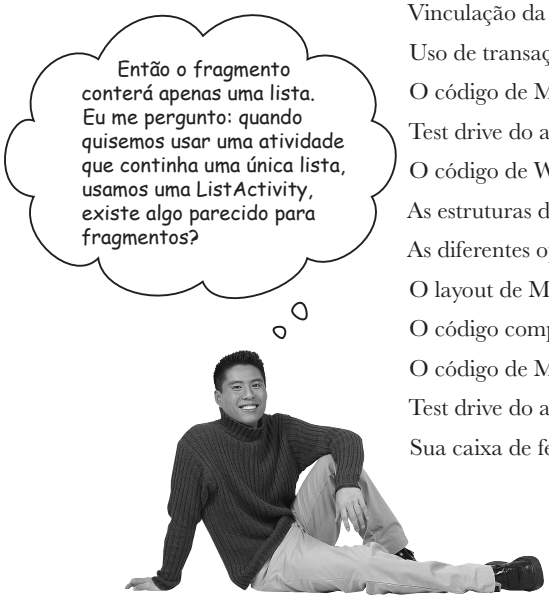


fragmentos

7 Modularize

Você viu como criar aplicativos que funcionam da mesma maneira, independente do dispositivo em que está sendo executado.

Mas, e se você quiser que seu aplicativo tenha *aparência e comportamento diferentes,* dependendo de estar sendo executado em um *celular* ou em um *tablet*? Neste capítulo vamos mostrar como fazer com que seu aplicativo escolha o **layout mais adequado para o tamanho de tela do dispositivo**. Também vamos apresentar os **fragmentos**, uma maneira de criar *componentes de código modulares que podem ser reutilizados por diferentes atividades.*

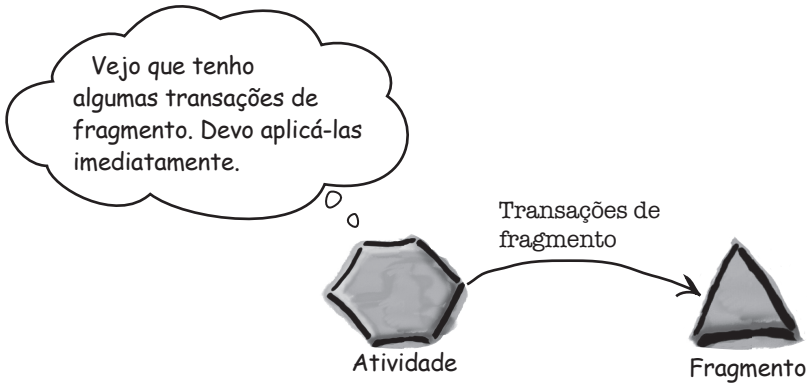


8 Fragmentos aninhados

Lidando com filhos

Já vimos que usar fragmentos em atividades permite reutilizar código e tornar seus aplicativos mais flexíveis.

Neste capítulo, vamos mostrar *como aninhar um fragmento dentro de outro.* Você vai ver como utilizar o **gerenciador de fragmentos filhos** para domar transações de fragmento incontroláveis. No processo, vai ver por que é tão importante *conhecer as diferenças entre atividades e fragmentos.*



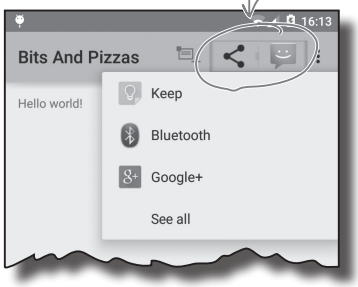
barras de ação

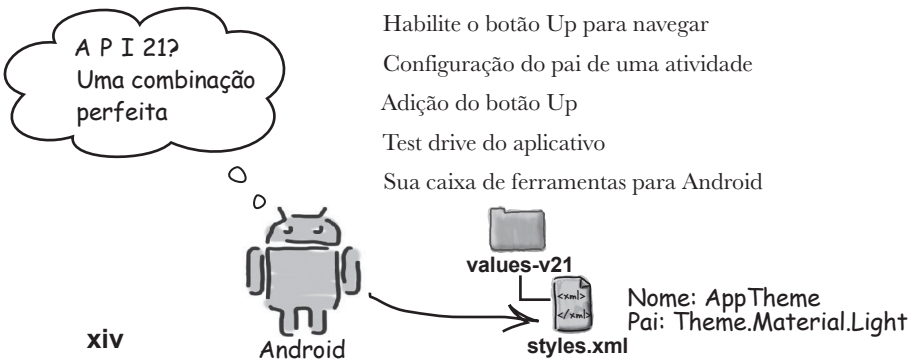
9 Pegando Atalhos

Todo mundo gosta de um atalho.

E, neste capítulo, você vai ver como adicionar atalhos aos seus aplicativos usando **barras de ação**. Vamos mostrar como iniciar outras atividades pela **adição de itens de ação** às suas barras de ação, como compartilhar conteúdo com outros aplicativos usando o **provedor de ação de compartilhamento** e como navegar na hierarquia de seu aplicativo implementando o **botão Up da barra de ação**. No processo, você vai ver como dar aparência e comportamento homogêneos ao seu aplicativo usando **temas** e vai conhecer o pacote de bibliotecas de suporte Android.

É assim que a ação de compartilhamento aparece na barra de ação. Quando você clica nela, ela fornece uma lista de aplicativos que podem ser usados para compartilhar conteúdo.



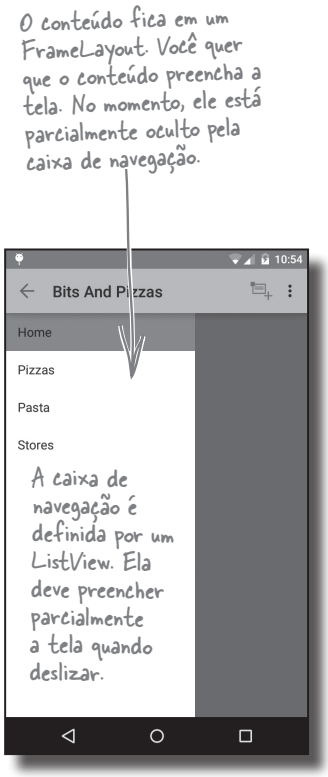


caixas de navegação

10 Indo a Lugares

Os aplicativos são muito melhores quando são fáceis de navegar.

Neste capítulo, vamos apresentar a **caixa de navegação**, um painel deslizante que aparece quando você passa o dedo pela tela ou clica em um ícone na barra de ação. Vamos mostrar como ela é utilizada para exibir *uma lista de vínculos* que o levam para **todos os hubs principais de seu aplicativo**. Você também vai ver como a *troca entre fragmentos* **facilita chegar** a esses hubs e torna **rápido exibi-los**.

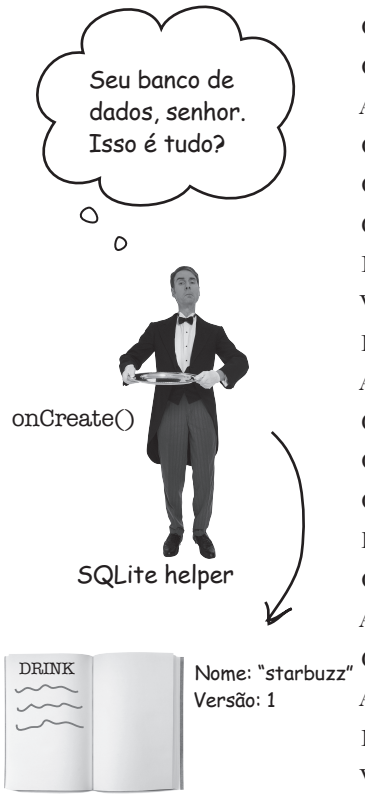


bancos de dados SQLite

11 Ative o Banco de Dados

Se você estiver gravando pontuações mais altas ou salvando tweets, seu aplicativo precisará armazenar dados.

E, no Android, você normalmente mantém seus dados seguros dentro de um **banco de dados SQLite**. Neste capítulo, vamos mostrar como você *cria um banco de dados, adiciona tabelas a ele e o preenche com dados,* tudo com a ajuda do **SQLite helper**. Então, você vai ver como pode *transferir atualizações* de forma limpa para sua estrutura de banco de dados e como *reverter*, caso precise desfazer quaisquer alterações.

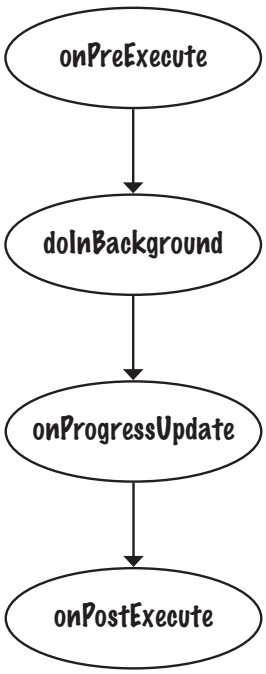


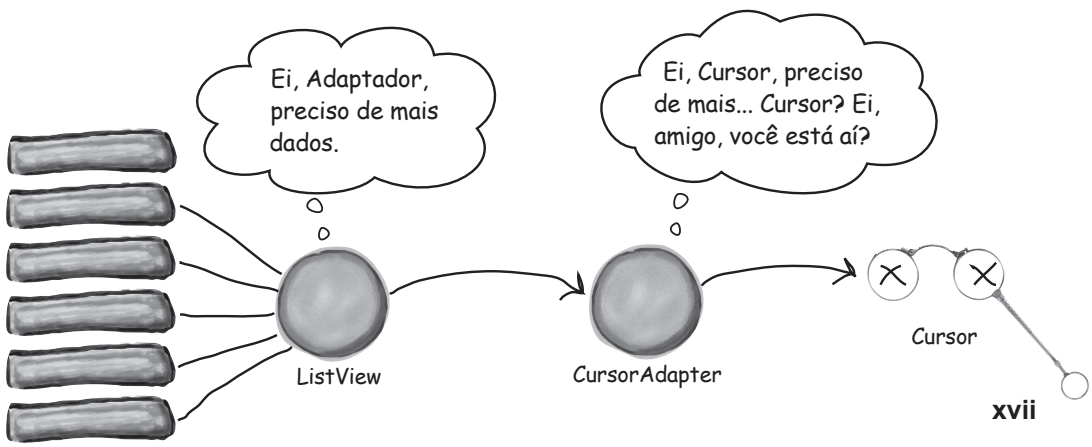
cursores e tarefas assíncronas

12 Conexão com Banco de Dados

Então, como você conecta seu aplicativo a um banco de dados SQLite?

Até aqui, vimos como criar um banco de dados SQLite usando um SQLite helper. O próximo passo é fazer suas atividades acessá-lo. Neste capítulo, você vai saber como utilizar **cursores** para *obter dados do banco de dados, como acessar cursores e como obter dados deles.* Então, vai descobrir como utilizar **cursor adapters** para conectá-los aos list views. Por último, você vai descobrir como o fato de escrever *código multithread* eficiente com **AsyncTaks** mantém seu aplicativo veloz.



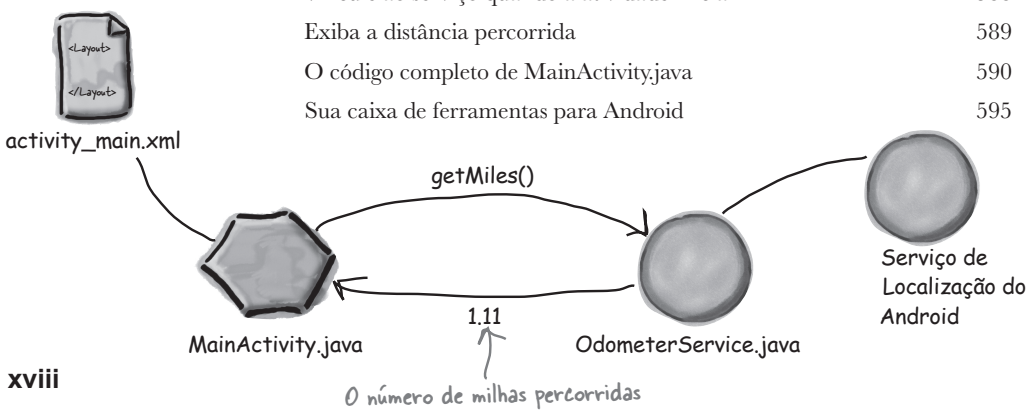


serviços

13 Ao Seu Serviço

Existem algumas operações que você quer manter em funcionamento, independente de qual aplicativo tenha o foco.

Por exemplo, se você começa a tocar uma melodia em um aplicativo de música, provavelmente espera que, ao trocar para outro aplicativo, ela continue tocando. Neste capítulo, você vai ver como usa **serviços** para lidar com situações como essa. No processo, vai ver como usa alguns dos **serviços internos do Android**. Examinaremos como você mantém seus usuários informados com o *serviço de notificação* e como o *serviço de localização* pode dizer onde você está.

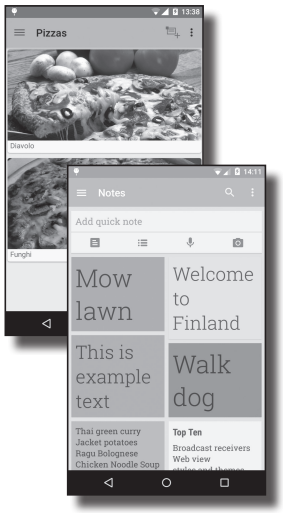


material design

14 A Vida em um Mundo Material

Com a API nível 21, a Google introduziu o Material Design.

Neste capítulo, vamos ver **o que é Material Design** e como fazer seus aplicativos se ajustarem a ele. Começaremos apresentando os **card views**, os quais podem ser reutilizados em seu aplicativo para se obter *aparência e comportamento uniformes*. Em seguida, vamos apresentar o **recycler view**, o amigo flexível do list view. No processo, você vai ver como **criar seus próprios adaptadores** e como mudar completamente a aparência de um recycler view com apenas *duas linhas de código*.

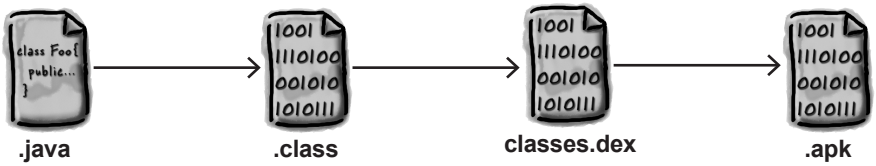


art

O Android Runtime

Os aplicativos Android precisam ser executados em dispositivos com processadores de baixo poder de processamento e muito pouca memória.

Os aplicativos Java podem ocupar muita memória e, como são executados dentro de sua própria Máquina Virtual Java (JVM), podem demorar bastante para iniciar ao serem executados em máquinas de baixo poder de processamento. O Android lida com isso não usando a JVM para seus aplicativos. Em vez disso, ele usa uma máquina virtual muito diferente, chamada Android runtime (ART). Neste apêndice, vamos ver como o ART faz com que seus aplicativos Java funcionem bem em um dispositivo pequeno e de baixo poder de processamento.

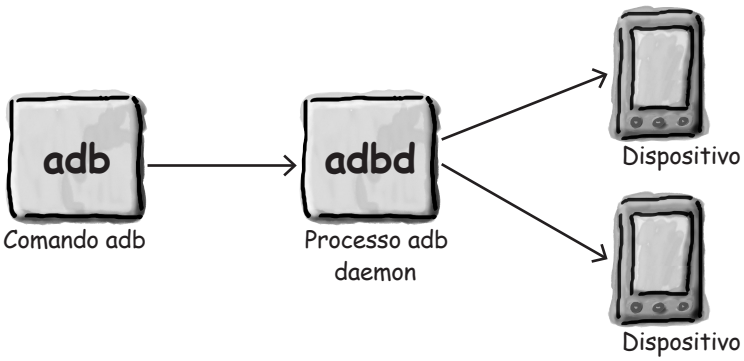


adb

O Android Debug Bridge

Neste livro, nos concentramos no uso de um IDE para todas as suas necessidades de Android.

Mas existem ocasiões em que usar uma ferramenta de linha de comando pode ser muito útil, como naqueles casos em que o Android Studio não consegue enxergar seu dispositivo Android, mas você simplesmente *sabe* que ele está lá. Neste capítulo, apresentaremos o Android Debug Bridge (ou adb), uma ferramenta de linha de comando que você pode usar para se comunicar com o emulador ou com dispositivos Android.

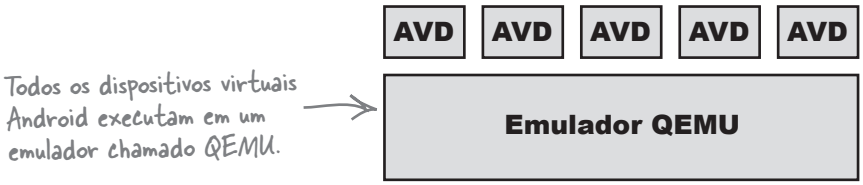


o emulator

iii O Android Emulator

Já se sentiu como se estivesse gastando todo seu tempo esperando pelo emulador?

Não há dúvida de que usar o emulador Android é útil. Ele permite ver como seu aplicativo será executado em dispositivos que não são os físicos a que você tem acesso. Mas, às vezes, ele pode se sentir um pouco... preguiçoso. Neste apêndice, vamos explicar por que o emulador pode parecer lento. Melhor ainda, vamos dar algumas dicas que aprendemos para **acelerá-lo**.



sobras

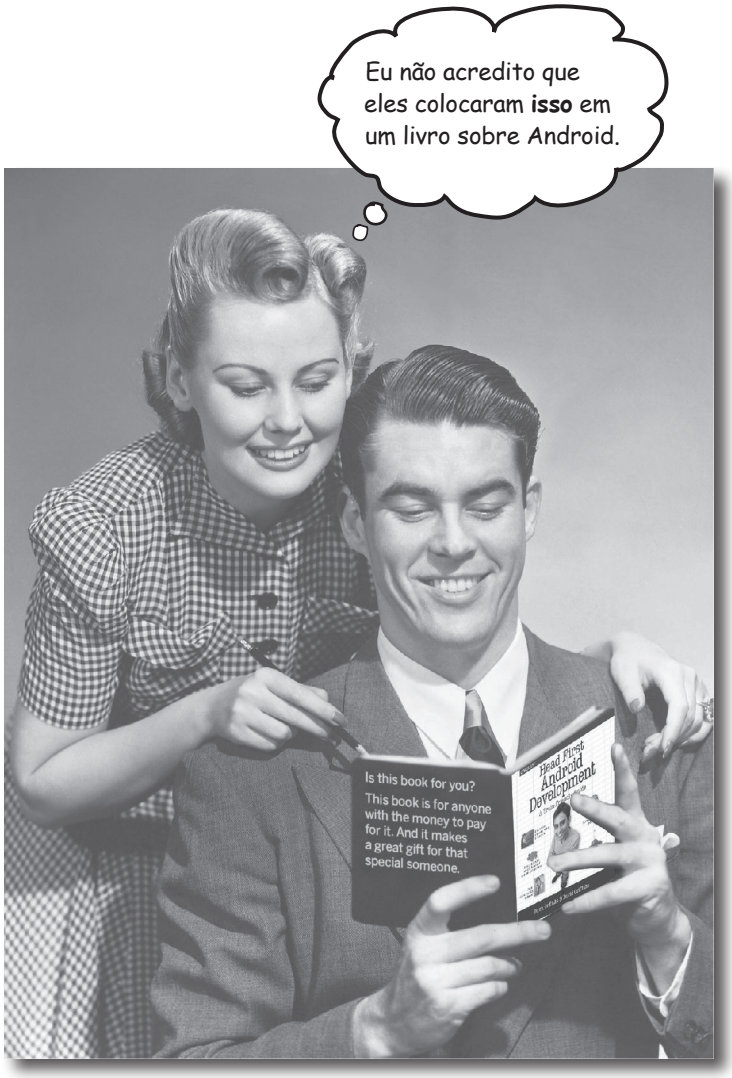
iv As Dez Mais (que não abordamos)

Mesmo depois de tudo isso, ainda há um pouco mais.

Achamos que você precisa saber apenas mais algumas coisas. Não nos sentiríamos bem as ignorando e queríamos oferecer a você um livro que pudesse levantar sem precisar de muita malhação na academia. Antes de deixar o livro de lado, **leia estes petiscos do início a fim**.

como usar este livro

Introdução



Nesta seção responderemos a pergunta inquietante: "Então, POR QUE eles colocaram isso em um livro sobre Android?"

A quem se destina este livro?

Se você responder "sim" para todas estas perguntas:

1. Você já sabe programar em Java?
2. Quer dominar o desenvolvimento de aplicativos Android, criar o próximo software de sucesso, fazer uma pequena fortuna e se retirar-se para sua própria ilha particular?
3. Você prefere fazer as coisas e aplicar o que aprendeu, em vez de ouvir alguém proferir uma palestra sem parar durante horas?

OK, talvez isto seja um pouco absurdo. Mas, você precisa começar em algum lugar, certo?

este livro é para você.

Quem provavelmente deve fugir deste livro?

Se você responder "sim" para alguma destas perguntas:

1. Você está procurando uma introdução rápida ou um livro de referência para desenvolver aplicativos Android?
2. Você preferiria ter as unhas do pé arrancadas por 15 macacos estridentes do que aprender algo novo? Você acredita que um livro sobre Android deveria abordar *tudo*, especialmente todo o material obscuro que jamais usará e, se isso levar o leitor às lágrimas no processo, tanto melhor?

este livro **não** é para você.

[Observação do Marketing: este livro é para qualquer um que tenha cartão de crédito...]

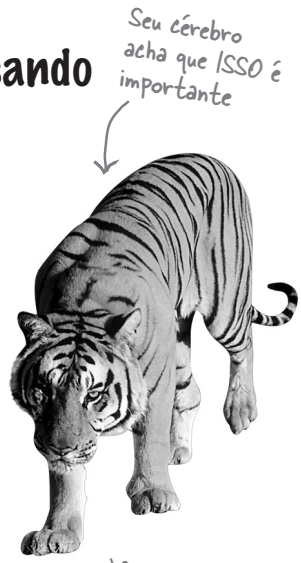
Sabemos o que você está pensando

"Como pode isto ser um livro sério sobre desenvolvimento de aplicativos Android?"

"Qual é a ideia de todos esses elementos gráficos?"

"Posso realmente *aprender* assim?"

Sabemos o que o seu cérebro está pensando



Seu cérebro anseia por novidades. Está sempre buscando, examinando, *esperando* por algo incomum. Ele foi feito assim e isso ajuda você a continuar vivo.

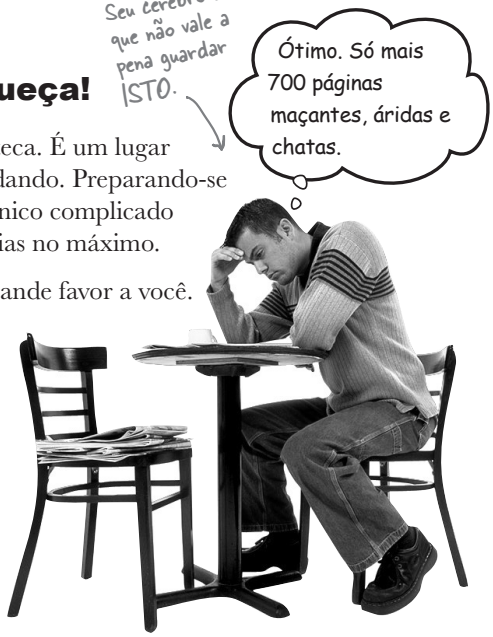
Então o que o seu cérebro faz com toda a rotina e coisas comuns e normais que você encontra? Tudo o que *pode* fazer para que elas parem de interferir no *verdadeiro* trabalho do cérebro — gravar o que *interessa*. Ele não se preocupa em preservar as coisas chatas; elas nunca passam no filtro "é óbvio que isso não é importante".

Como seu cérebro *sabe* o que é importante? Suponha que você esteja fazendo uma caminhada durante o dia e um tigre pula na sua frente — o que acontece dentro da sua cabeça e do seu corpo?

Fogo nos neurônios. Emoções se intensificam. Explosão química.

E é assim que seu cérebro sabe...

Isto deve ser importante! Não esqueça!



Mas imagine que você esteja em casa ou em uma biblioteca. É um lugar seguro, quente, fora do alcance de tigres. Você está estudando. Preparando-se para uma prova. Ou tentando aprender algum tema técnico complicado que seu chefe acha que vai demorar uma semana, dez dias no máximo.

Só um problema. Seu cérebro está tentando fazer um grande favor a você. Está tentando garantir que esse assunto, *obviamente* nada importante, não entulhe recursos escassos. Recursos mais bem utilizados armazenando as coisas *realmente* importantes. Como tigres. Como perigo de incêndio. Como você nunca deveria ter postado aquelas fotos da festa em seu perfil no Facebook. E não há um jeito fácil de dizer ao seu cérebro: "ei, cérebro, muito obrigado, mas não importa o quanto este livro é maçante e como meu registro emocional na escala Richter é baixo no momento, eu realmente *quero* guardar todas essas coisas."

1 início

Mergulhe

O Android conquistou o mundo.

Todo mundo quer um smartphone ou um tablet, e os dispositivos Android são muito populares. Neste livro, vamos ensiná-lo a **desenvolver seus próprios aplicativos** e vamos começar fazendo com que você construa um aplicativo básico em um Dispositivo Virtual Android. Ao longo do caminho, você conhecerá alguns dos componentes básicos de todos os aplicativos Android, como as **atividades** e os **layouts**. **Tudo que você precisa é de algum conhecimento sobre Java...**

Bem-vindo à Androidville

Android é a plataforma móvel mais popular do mundo. Na última contagem, havia mais de *um bilhão* de dispositivos Android ativos no mundo, e esse número está aumentando rapidamente.

Android é uma plataforma de código-fonte aberto abrangente, baseada no Linux e promovida pelo Google. É um poderoso framework de desenvolvimento, contendo tudo que é preciso para se construir excelentes aplicativos, usando uma mistura de Java e XML. Além disso, permite implantar esses aplicativos em uma grande variedade de dispositivos — telefones celulares, tablets e muito mais.

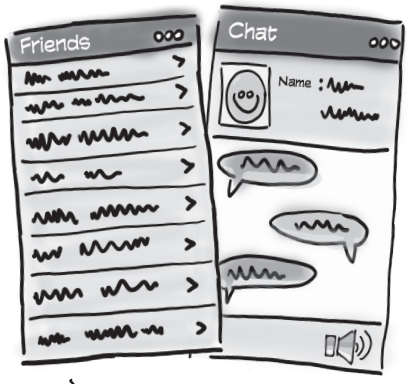
Então, o que compõe um aplicativo Android típico?

Vamos construir nossos aplicativos Android usando uma mistura de Java e XML. Explicaremos as coisas pelo caminho, mas você precisará ter um bom entendimento de Java para extrair o máximo deste livro.

Layouts definem a aparência de cada tela

Um aplicativo Android típico é composto de uma ou mais telas. A aparência de cada tela é definida por um layout. Normalmente, os layouts são definidos usando-se XML e podem incluir componentes de interface do usuário, como botões, campos de texto e rótulos.

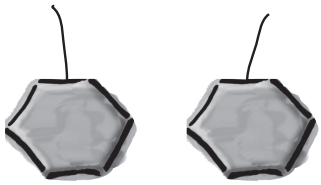
Os layouts informam ao Android sobre a aparência das telas em seu aplicativo.



Código Java define o que o aplicativo deve fazer

Os layouts definem apenas a *aparência* do aplicativo. O que o aplicativo *faz* é definido pelo código Java. Uma classe especial do Java, chamada **atividade**, decide o layout a ser usado e diz ao aplicativo como responder ao usuário. Como exemplo, se um layout inclui um botão, é preciso escrever código em Java na atividade para definir o que o botão deve fazer quando for pressionado.

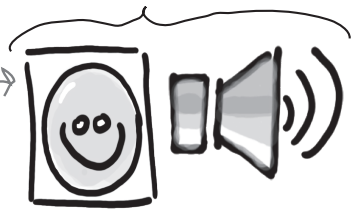
As atividades definem o que o aplicativo deve fazer.



Às vezes também são necessários recursos extras

Além do código Java e dos layouts, os aplicativos Android frequentemente precisam de recursos extras, como arquivos de imagem e dados de aplicativo. Você pode adicionar quaisquer arquivos extras necessários ao aplicativo.

Os recursos podem incluir arquivos de som e imagem.



Na verdade, os aplicativos Android são constituídos simplesmente de vários arquivos em diretórios específicos. Quando se constrói um aplicativo, todos esses arquivos são empacotados juntos, fornecendo um aplicativo que pode ser executado em um dispositivo.

A plataforma Android dissecada

Não se preocupe se parece que existem coisas demais para entender.

Neste estágio estamos apenas fornecendo uma visão geral do que existe na plataforma Android. Vamos explicar os diferentes componentes com mais detalhes quando for necessário.

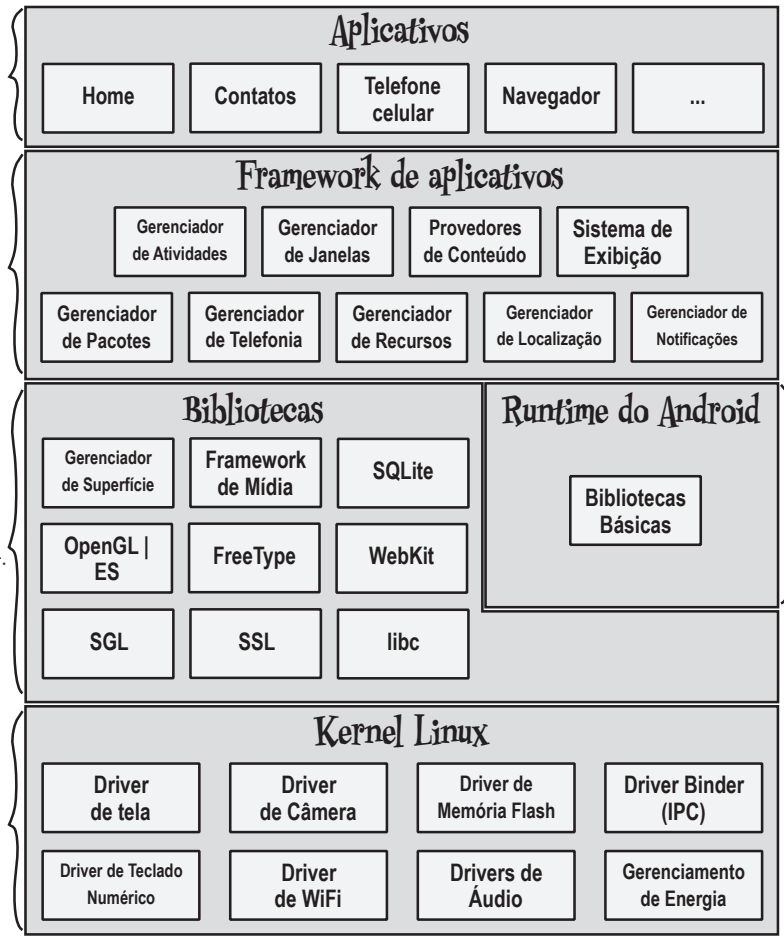
A plataforma Android é constituída de vários componentes diferentes. Isso inclui aplicativos básicos, como Contatos, um conjunto de APIs para ajudá-lo a controlar a aparência e o comportamento de seu aplicativo, além de toda uma coleção de arquivos e bibliotecas de apoio. Vamos dar uma rápida olhada em como tudo isso se encaixa:

O Android vem com um conjunto de aplicativos básicos como Contatos, Calendário, Mapas e um navegador.

Ao construir seus aplicativos, você tem acesso às mesmas APIs usadas pelos aplicativos básicos. Essas APIs são usadas para controlar a aparência e o comportamento de seu aplicativo.

Debaixo do framework de aplicativos está um conjunto de bibliotecas C e C++. Essas bibliotecas são expostas por meio das APIs de framework.

Debaixo de tudo está o kernel Linux. O Android conta com o kernel para drivers e também para serviços básicos, como segurança e gerenciamento de memória.



O runtime Android vem com um conjunto de bibliotecas básicas que implementam a maior parte da linguagem de programação Java. Cada aplicativo Android é executado em seu próprio processo.

A boa nova é que todas as poderosas bibliotecas Android são expostas por meio das APIs no framework de aplicativos. São essas as APIs que você usa para criar excelentes aplicativos Android. Para começar, basta algum conhecimento sobre Java e uma ótima ideia para um aplicativo.

Vamos fazer o seguinte

Então, vamos mergulhar no assunto e criar um aplicativo Android básico. Precisamos fazer apenas algumas coisas:

1. **Configurar um ambiente de desenvolvimento.**
   Precisamos instalar o Android Studio, o qual inclui todas as ferramentas necessárias para se desenvolver aplicativos Android.

2. **Construir um aplicativo básico.**
   Usando o Android Studio, vamos construir um aplicativo simples que exibirá um exemplo de texto na tela.

3. **Executar o aplicativo no emulador Android.**
   Vamos usar o emulador interno para ver o aplicativo funcionando.

4. **Alterar o aplicativo.**
   Por último, vamos fazer alguns ajustes no aplicativo criado no passo 2, e executá-lo novamente.

Não existem Perguntas Idiotas

P: Todos os aplicativos Android são desenvolvidos em Java?

R: Também é possível desenvolver aplicativos Android em outras linguagens, mas a verdade é que a maioria dos desenvolvedores utiliza Java.

P: Quanto eu preciso conhecer de Java para desenvolver aplicativos Android?

R: Você precisa ter experiência em Java SE. Se estiver se sentindo "enferrujado", sugerimos que consiga um exemplar do livro *Use a cabeça! Java*, de Kathy Sierra e Bert Bates.

P: Preciso conhecer Swing e AWT?

R: O Android não usa Swing nem AWT; portanto, não se preocupe se não tiver experiência na interface do usuário da área de trabalho Java.

início
Configure o ambiente
Construa o aplicativo
Execute o aplicativo
Altere o aplicativo

Seu ambiente de desenvolvimento

Java é a linguagem mais popular para o desenvolvimento de aplicativos Android. Os dispositivos Android não executam arquivos *.class* e *.jar*. Em vez disso, para aumentar a velocidade e o desempenho da bateria, os dispositivos usam seus próprios formatos otimizados de código compilado. Isso significa que não é possível usar um ambiente de desenvolvimento Java normal — também são necessárias ferramentas especiais para converter seu código compilado em um formato Android, para implantá-los em um dispositivo e para permitir a depuração do aplicativo, uma vez que esteja executando.

Tudo isso faz parte do **SDK Android**. Vamos dar uma olhada no que está incluso.

O Android SDK

O SDK (Software Development Kit) Android contém as bibliotecas e ferramentas necessárias para o desenvolvimento de aplicativos Android:

Plataforma do SDK
Há uma dessas para cada versão de Android.

Ferramentas do SDK
Ferramentas para depuração e teste, além de outros utilitários práticos. Existe também um conjunto de ferramentas dependentes da plataforma.

Amostras de aplicativos
Se quiser exemplos de código práticos para ajudá-lo a entender como usar algumas das APIs, as amostras de aplicativos podem ser úteis.

Documentação
Para que você possa obter a documentação da API mais recente offline.

Suporte para Android
APIs extras que não estão disponíveis na plataforma padrão.

Faturamento do Google Play
Permite integrar serviços de faturamento em seu aplicativo.

Esses são apenas alguns dos pontos principais.

O Android Studio é uma versão especial de IntelliJ IDEA

IntelliJ IDEA é um dos IDEs mais populares para desenvolvimento com Java. O Android Studio é uma versão de IDEA que inclui uma versão do SDK Android e ferramentas extras de interface do usuário para ajudá-lo em seu desenvolvimento de aplicativos.

Além de fornecer um editor e o acesso às ferramentas e bibliotecas do SDK Android, o Android Studio oferece modelos que podem ser usados para ajudar a criar novos aplicativos e classes, além de tornar fácil fazer coisas como empacotar seus aplicativos e executá-los.

Instale o Java

Configure o ambiente
Construa o aplicativo
Execute o aplicativo
Altere o aplicativo

O Android Studio é um ambiente de desenvolvimento com Java; portanto, você precisa garantir que a versão correta de Java esteja instalada em sua máquina.

Primeiro, verifique os requisitos de sistema do Android Studio para ver quais versões do JDK (Java Development Kit) e do JRE (Java Runtime Edition) são necessárias. Os requisitos de sistema podem ser vistos aqui:

Às vezes, a Oracle e o Google mudam seus URLs. Se esses URLs não funcionarem, faça uma pesquisa.

http://developer.android.com/intl/pt-br/sdk/index.html#Requirements

Quando souber quais versões do JDK e do JRE são necessárias, pode obtê-las e instalá-las (conteúdo em inglês):

http://www.oracle.com/technetwork/java/javase/downloads/index.html

Então, instale o Android Studio

Se esse URL tiver mudado, procure o Android Studio em developer.android.com. (conteúdo em inglês)

Quando o Java estiver funcionando, baixe o Android Studio do seguinte link (conteúdo em inglês):

https://developer.android.com/sdk/installing/index.html?pkg=studio

Essa página também contém instruções de instalação. Siga as instruções para instalar o Android Studio em seu computador. Quando tiver instalado o Android Studio, abra-o e siga as instruções para adicionar as ferramentas do SDK e as bibliotecas de apoio mais recentes.

Não incluímos instruções de instalação neste livro, pois elas podem ficar desatualizadas muito rapidamente. Siga as instruções online e tudo correrá bem.

Quanto terminar, você deverá ver a tela de boas-vindas do Android Studio. Você está pronto para construir seu primeiro aplicativo Android.

Esta é a tela de boas-vindas do Android Studio. Ela inclui um conjunto de opções para coisas que podem ser feitas.