



# Curso de Desenvolvimento em ambientes Android

Prof. Henrique Yoshikazu Shishido

---

Universidade Tecnológica Federal do Paraná  
Campus Cornélio Procópio  
[shishido@utfpr.edu.br](mailto:shishido@utfpr.edu.br)  
+55 43 3520-3927

# Pré-requisitos

- Lógica de programação e algoritmos
- Conhecimento de orientação a objetos
- Linguagem Java
- Conhecimento de Linux básico



codeup

# Conteúdo

- Contexto do desenvolvimento em Android
- O ambiente de desenvolvimento
- Estrutura de um projeto exemplo
- O primeiro Hello World!
- Interfaces com o usuário
- Internacionalização e localização
- Componentes de interface gráfica
- Temas e imagens background
- O componente List e menus de contexto
- Instalação de aplicações
- Persistência e retorno de activities



android

# Contextualização do ambiente Android

- A plataforma Android é uma adaptação do sistema operacional Linux para tablets, smartphones e PDA
- A plataforma oferece uma máquina virtual otimizada para a execução de *bytecodes* específicos da DVK
- *Dalvik Virtual Machine* ! A DVK não é uma  
Máquina Virtual JAVA!
  - Interpreta apenas *bytecodes* específicos a partir dos *bytecodes* Java convertidos pela ferramenta “DX”



ANDROID

# Dalvik Virtual Machine

Programa.java

```
public class Programa {  
...  
}
```

Java compiler



Programa.class

```
010101001010101  
010101001110101  
010101010101010101
```

Dalvik compiler  
“dx”



Programa.dalvik

```
1010100010010110  
10100101011001  
101001011101001
```

Execution on  
Android Hardware



android

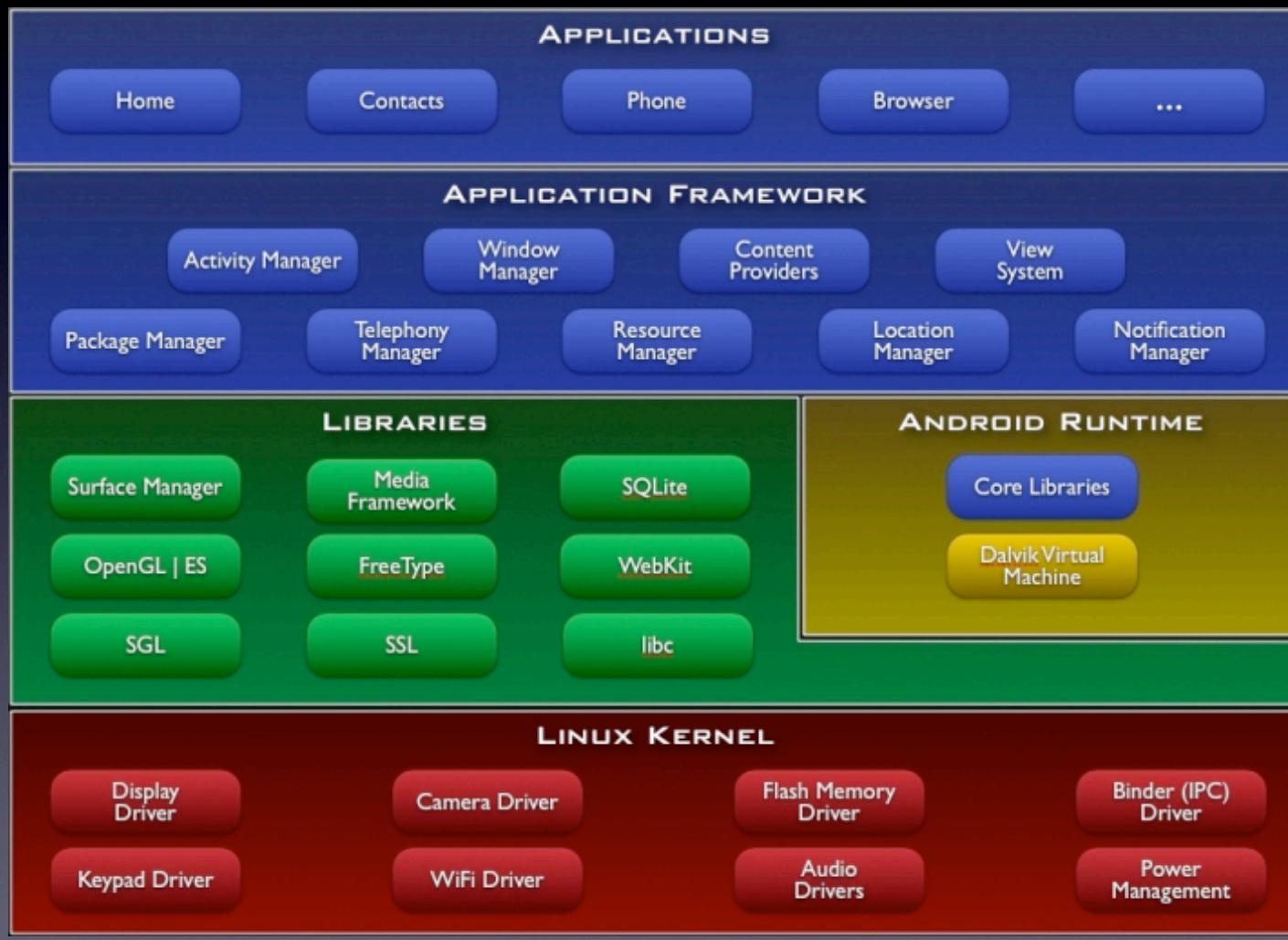
# Contextualização do ambiente Android

- O Google disponibilizou um kit de desenvolvimento (SDK) para a plataforma Android em Java
- Bibliotecas
- Emulador de um hardware Android
- Conversores de *bytecode*



ANDROID

# Arquitetura



android

# Arquitetura

- Cada “programa” é executado por uma instância da Dalvik Virtual Machine, abstraída por um processo no S.O.
- As vantagens da arquitetura Android são:
  - Não há a necessidade de usar linguagens como C/C++
  - Não é preciso fazer *system calls* (chamadas de sistema) para acessar o hardware



Afinal, o que é preciso então para se programar para Android?



- SDK Android
- IDE Java



ANDROID

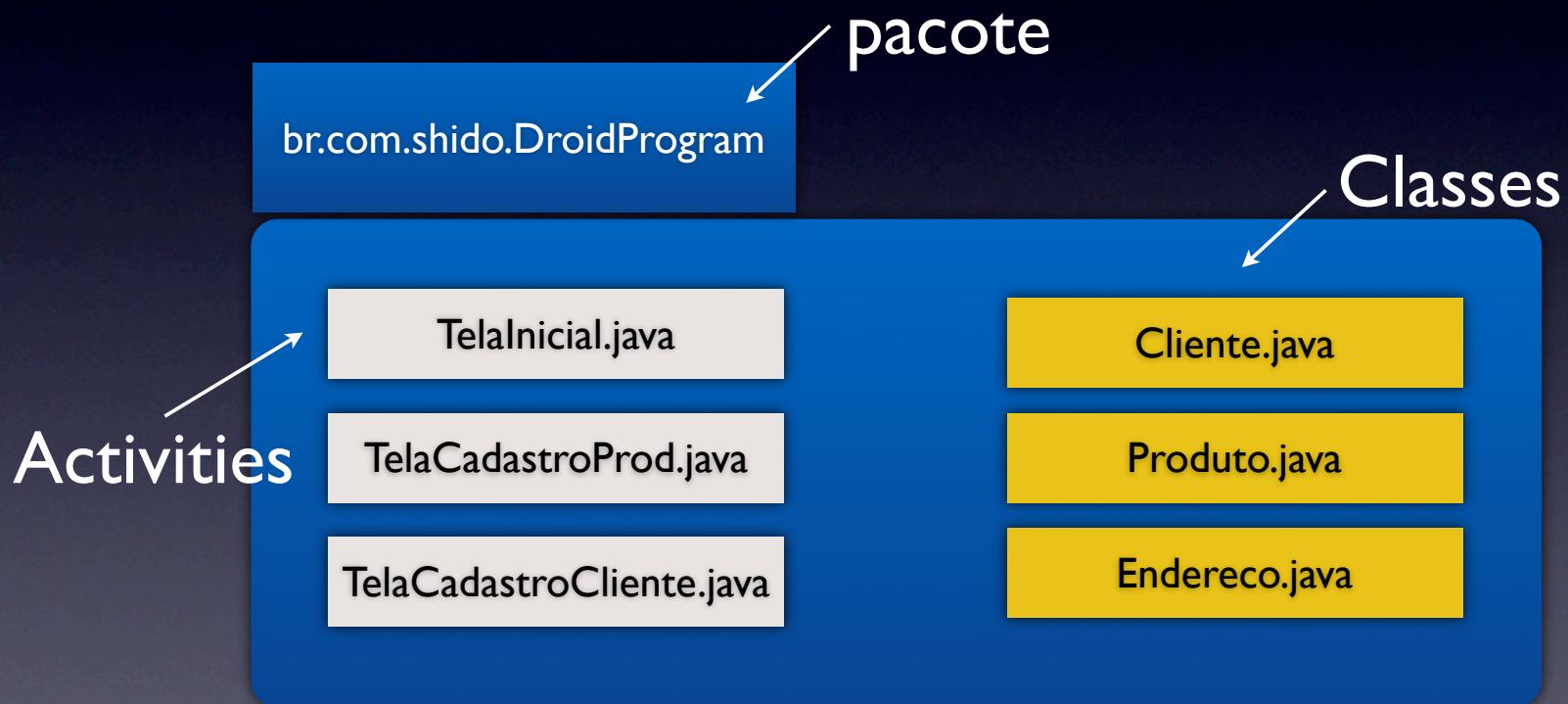
# Anatomia de uma aplicação Android

- **Activities** - telas
- **Services** - processos em execução *background*
- **Content Providers** - interfaces para disponibilização de dados
- **Broadcast Receivers** - elementos capazes de responder por eventos gerais do sistema
- **Eventos** - sinais de origem de usuário ou por componentes do sistema (timer, bloqueio de tela, ...)



ANDROID

# Anatomia de uma aplicação Android



# Componentes Android

- Os componentes Android são implementados por *frameworks* e você os estende para criar seus componentes específicos

Por exemplo:

## View

<http://developer.android.com/reference/android/view/View.html>

O View é um componente para controle de telas para:  
interação do usuário com a tela;  
Existem: **TextView**, **EditView**, etc;



ANDROID

# Componentes Android

- Os **Views** podem gerar e receber eventos que podem ser capturados no código-fonte
- Componentes como **ViewGroups** permitem organizar melhor a tela
- Existem **Views** fornecidas pelo próprio SDK capazes de agregar funcionalidades genéricas



ANDROID

# Activity

i

Uma activity é uma interface para o usuário

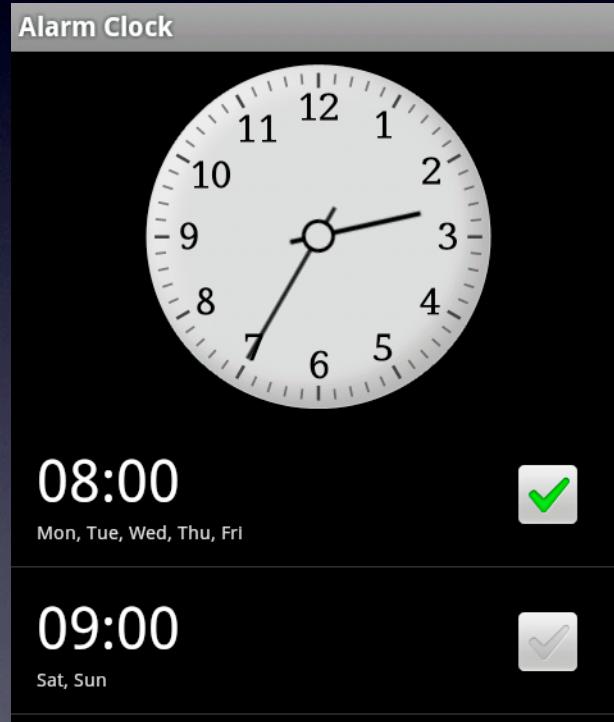
i

Para gerar uma nova Activity é necessário criar um arquivo de layout, contendo um ou mais

**ViewGroups**

i

Todos os eventos são gerados pelas Views e pelos botões do aparelho (*button menu, context button, back*)



android

# Service

- Um serviço é um componente capaz de executar operações de longo prazo e em *background*. Exemplos: I/O, serviços de rede, GPS, alertas...
- Vamos pensar em uma aplicação de vendas
  1. Vendedores recebem atualizações de Clientes e acessam através da aplicação
  2. O download automático de atualizações poderia ser feito por um Service de sua aplicação
  3. Um service fica rodando em *background* e pode executar tarefas longas e repetitivas, além de ser acessível por outros programas
- Um media-player faz uso de um Service para continuar tocando a música mesmo que a interface do player seja fechada



android

# Content Provider

- Armazenam e recuperam dados, e permitem disponibilizá-los a outras aplicações
  - Contacts
  - Agenda
- O Android vem com vários *Content Providers*, *contacts info*, e qualquer outro programa pode obter a lista de contatos do usuário (desde que autorizado)
- Aplicações mobile estão muito relacionadas com “mashups”



ANDROID

# Dúvidas?





# O Ambiente Android

Prof. Henrique Yoshikazu Shishido

---

Universidade Tecnológica Federal do Paraná  
Campus Cornélio Procópio  
[shishido@utfpr.edu.br](mailto:shishido@utfpr.edu.br)  
+55 43 3520-3927

# Um aplicativo Android

- Uma aplicação Android precisa ser empacotada para distribuí-la.
- A extensão de uma aplicação Android é “.apk”
- Dentro de um .apk, podemos encontrar:
  - O *hashcode* das pastas é calculado e adicionado no arquivo “Manifest.mf”
  - A pasta “res” armazena os recursos não compiláveis, como: arquivos de layout, imagens, menus, etc
  - Os *bytecodes* Dalvik são armazenados no arquivo “dex”
  - O arquivo “arsc” armazena as *strings* e uma tabela com as entradas do diretório “res” em formato binário

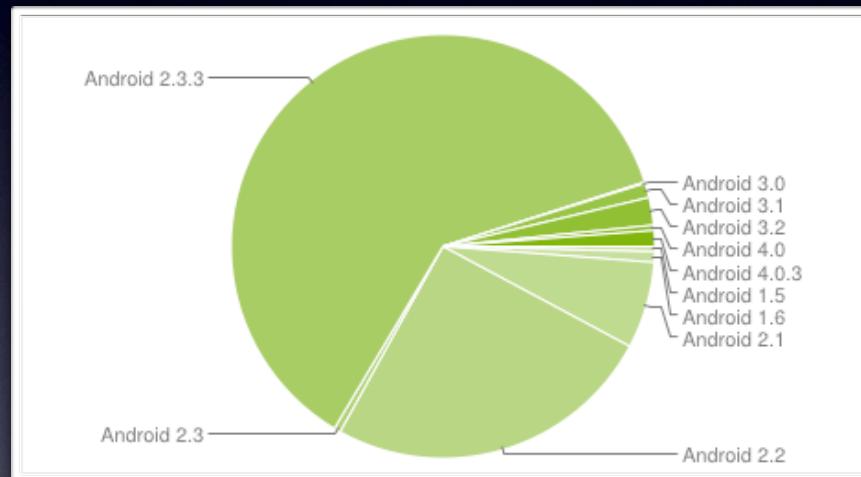


ANDROID

# Versões de plataformas Android

- O Android possui diversas versões de plataformas de desenvolvimento, conforme a tabela a seguir:

Fonte: Google (5 de março de 2012)



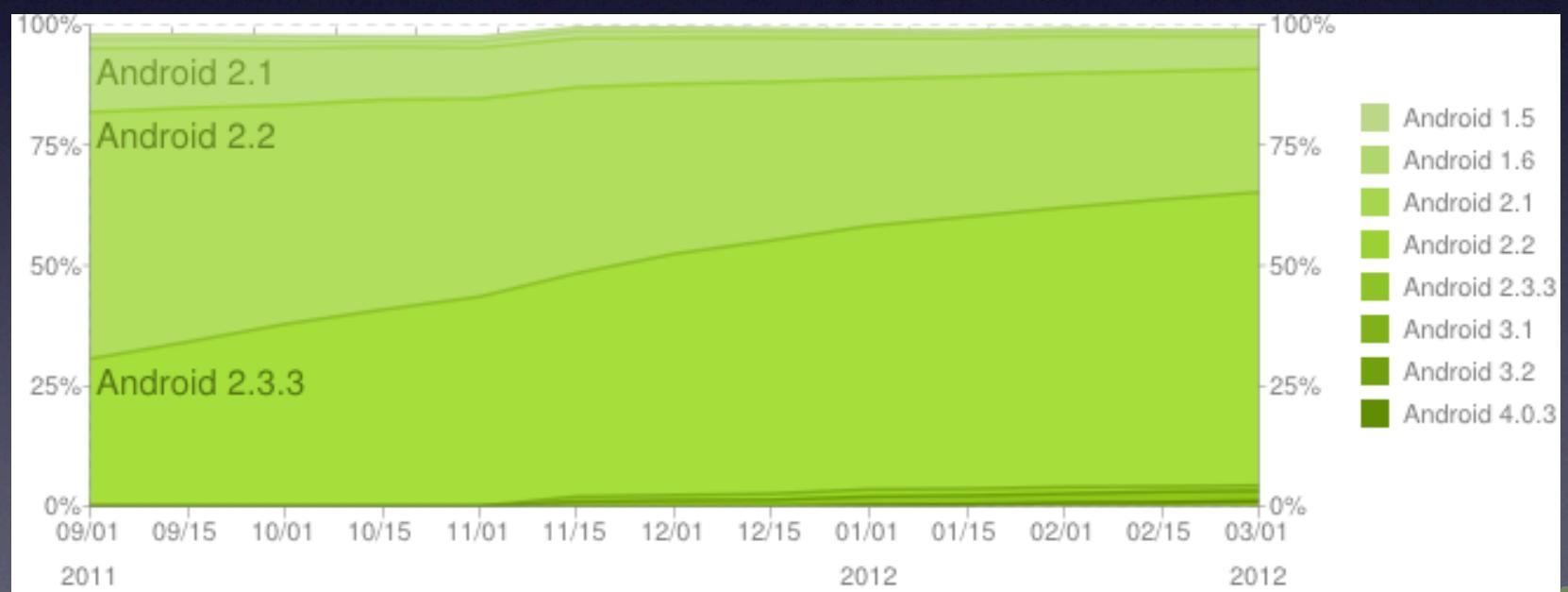
i

- O ideal é desenvolver sob a plataforma onde há o maior número de usuários!

Platform	Codename	API Level	Distribution
<a href="#">Android 1.5</a>	Cupcake	3	0.4%
<a href="#">Android 1.6</a>	Donut	4	0.8%
<a href="#">Android 2.1</a>	Eclair	7	6.6%
<a href="#">Android 2.2</a>	Froyo	8	25.3%
<a href="#">Android 2.3 - Android 2.3.2</a>	Gingerbread	9	0.5%
<a href="#">Android 2.3.3 - Android 2.3.7</a>		10	61.5%
<a href="#">Android 3.0</a>	Honeycomb	11	0.1%
<a href="#">Android 3.1</a>		12	1.1%
<a href="#">Android 3.2</a>		13	2.1%
<a href="#">Android 4.0 - Android 4.0.2</a>	Ice Cream Sandwich	14	0.4%
<a href="#">Android 4.0.3</a>		15	1.2%

# Versões de plataformas Android

- Pode-se observar que a versão 2.3.3 (Gingerbread) ganhou maior espaço no mercado no primeiro trimestre de 2012
- Em um período de 1 ano, pode haver a migração de uma versão para outra



Fonte: Google (5 de março de 2012)

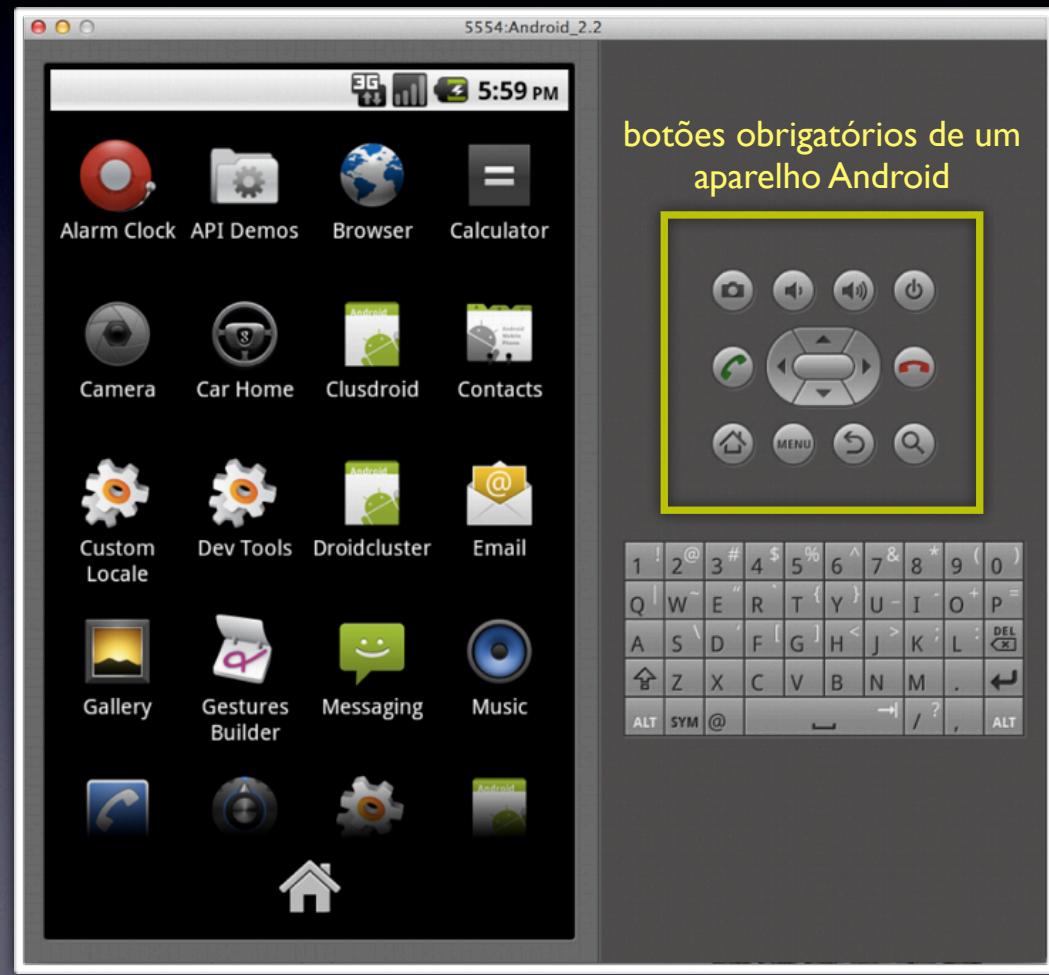


ANDROID

# Android Emulator

O Android Emulator permite executar e testar os aplicativos desenvolvidos sem a necessidade de um aparelho real

A experiência de uso do emulador é idêntica ao de um aparelho real, exceto à sensação do *touch screen*



# Localização de aplicativos no dispositivo

- Os aplicativos são armazenados nos dispositivos em dois locais:
  - No diretório `/data/app` (apk)
  - e em `/data/dalvik-cache` (arquivos .dex)





# Android Market

- A distribuição de aplicativos é realizada através do Android Market
- Existem dois métodos de distribuição:
  - Gratuito
  - Pago
- “O mercado de aplicativos mobile deve crescer para cerca de 50 bilhões até 2012”. (GetJars)
- Para disponibilizar um aplicativo no Android Market é preciso seguir algumas regras apresentadas no eslaide a seguir



ANDROID

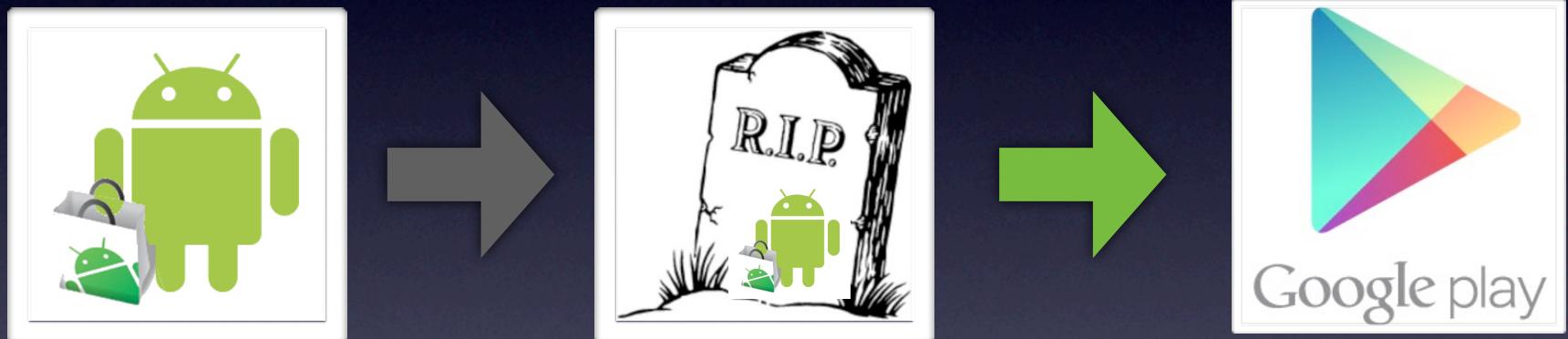


# Exigências de distribuição no Android Market

- Para publicar um aplicativo é necessário:
  1. Pagar U\$25,00 pelo registro (realizado somente 1 vez)
  2. Deve-se enviar pelo menos dois screenshots da aplicação com resolução de 320x480 pixels
  3. Enviar o “.apk” da aplicação
  4. Enviar um ícone de resolução 512x512 pixels
  5. Assinar a aplicação com uma chave criptográfica com validade superior a 22/10/2033
  6. Definir os atributos “`android:icon`” e “`android:label`” no elemento `<application>` no arquivo “`AndroidManifest.xml`”. Por fim, definir os atributos “`android:versioncode`” e “`android:versionName`” no elemento `<manifest>`.



# Porém, o Android Market...





# Google Play!

- O Google Play veio para realizar a fusão entre:
  - Android Market
  - Google Music
  - Google eBook Store

?

Por um acaso, esse conjunto de serviços te faz lembrar de algum outro aplicativo?

# O ambiente de desenvolvimento

- Para desenvolver qualquer aplicativo Android é preciso das seguintes ferramentas:
  - IDE Eclipse (interface para codificação)
  - Android SDK Manager (bibliotecas e emulador)
  - Android Developer Toolkit (ADT) (*plugin* para integração com a IDE Eclipse)



# Download das ferramentas...

- IDE Eclipse



[Windows \(32-bit\)](#)

[Windows \(64-bit\)](#)



[Linux \(32-bit\)](#)

[Linux \(64-bit\)](#)



[Mac OS X \(32-bit\)](#)

[Mac OS X \(64-bit\)](#)



ANDROID

# Download das ferramentas...

- Android SDK



[Windows](#)



[Linux](#)



[Mac OS X](#)



ANDROID

# Instalação e configuração do ambiente

- Após baixar o Eclipse e o Android SDK, siga os seguintes passos:
  1. Descompacte a IDE Eclipse na pasta desejada
  2. Instale o Android SDK Manager
  3. Execute o Android SDK Manager e marque a versão da plataforma que deseja instalar conforme o eslaide posterior



ANDROID

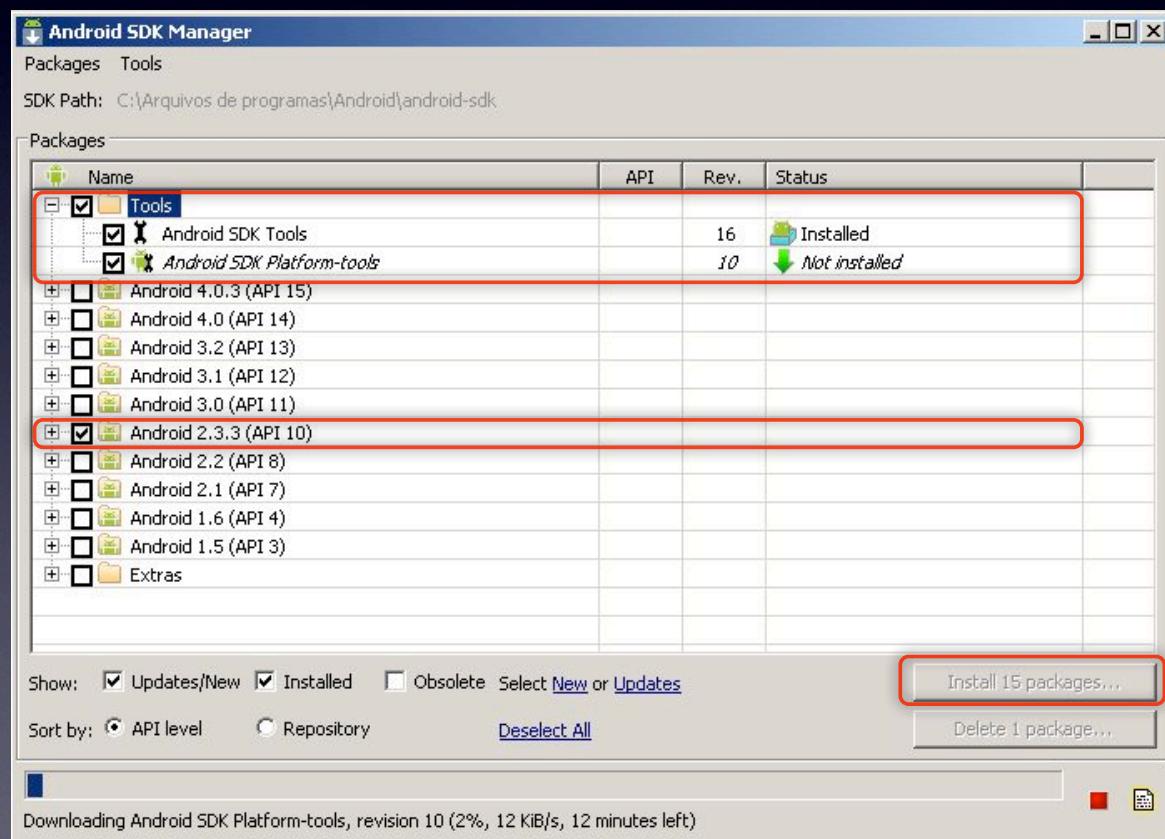
# Instalação e configuração do ambiente

Instruções para o Android SDK Manager:

1. Marque a categoria “Tools”
2. Marque a plataforma Android 2.3.3
3. Clique em “Install Packages”



O processo de download da SDK é longo...



# Instalação e configuração do ambiente

4. Execute o Eclipse e clique em “Help” => “Install new software...”

5. Clique no botão “Add...” e adicione os seguintes dados:

Name:ADT

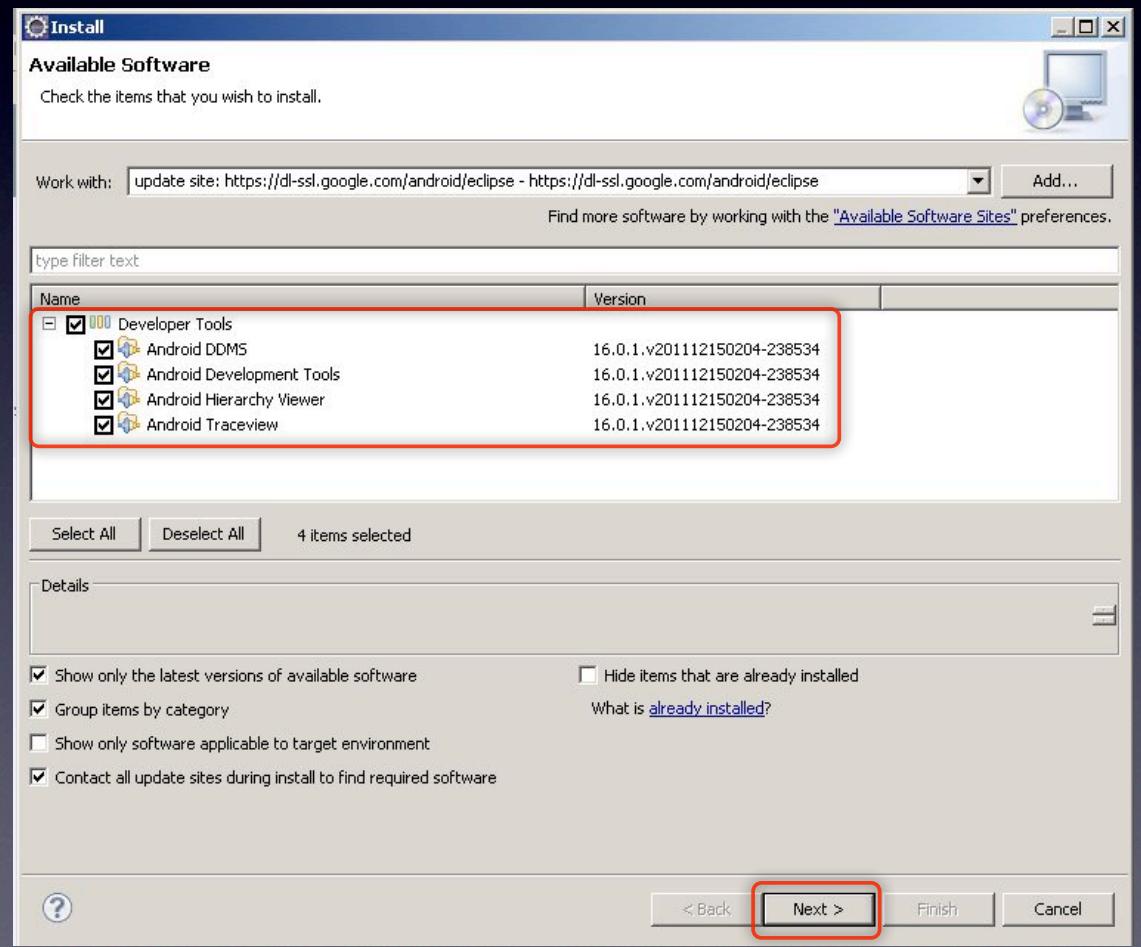
Location: <https://dl-ssl.google.com/android/eclipse>

6. Marque a categoria completo de “Developer Tools” (conforme figura)

7. Clique em “Next”



O processo de download do ADT também é longo... tenha paciência



# Instalação e configuração do ambiente

Pronto! Assim que completado os passos anteriores, o seu ambiente Android já estará pronto para o desenvolvimento!





# Estrutura de um projeto Android

Prof. Henrique Yoshikazu Shishido

---

Universidade Tecnológica Federal do Paraná  
Campus Cornélio Procópio  
[shishido@utfpr.edu.br](mailto:shishido@utfpr.edu.br)  
+55 43 3520-3927

# Por que o Eclipse?

- A maioria dos desenvolvedores utilizam a IDE Eclipse
- O Eclipse é uma IDE *open source*
- O Eclipse possui o Android Development Toolkit (ADT)
- A maioria dos materiais são baseados no Eclipse



# Android Version x API Level

- **ATENÇÃO:** Existem muitas versões de Android disponíveis!
- Não será qualquer aplicação que poderá ser executada em qualquer dispositivo Android
- Conforme vimos na aula passada, a versão mais utilizada é a Android 2.3 - *Gingerbread*, seguida da versão 2.2 - *Froyo*



# Android Version x API Level

- Uma aplicação Android pode estabelecer quais os requisitos de sistema necessários, como:
  - **android: minSdkVersion**: menor API suportada
  - **android: maxSdkVersion**: maior API suportada
  - **android:targetSdkVersion**: para qual API sua aplicação foi desenvolvida
- Não se esqueça que você deve utilizar uma plataforma AVD (*Android Virtual Device*) compatível com o nível da API

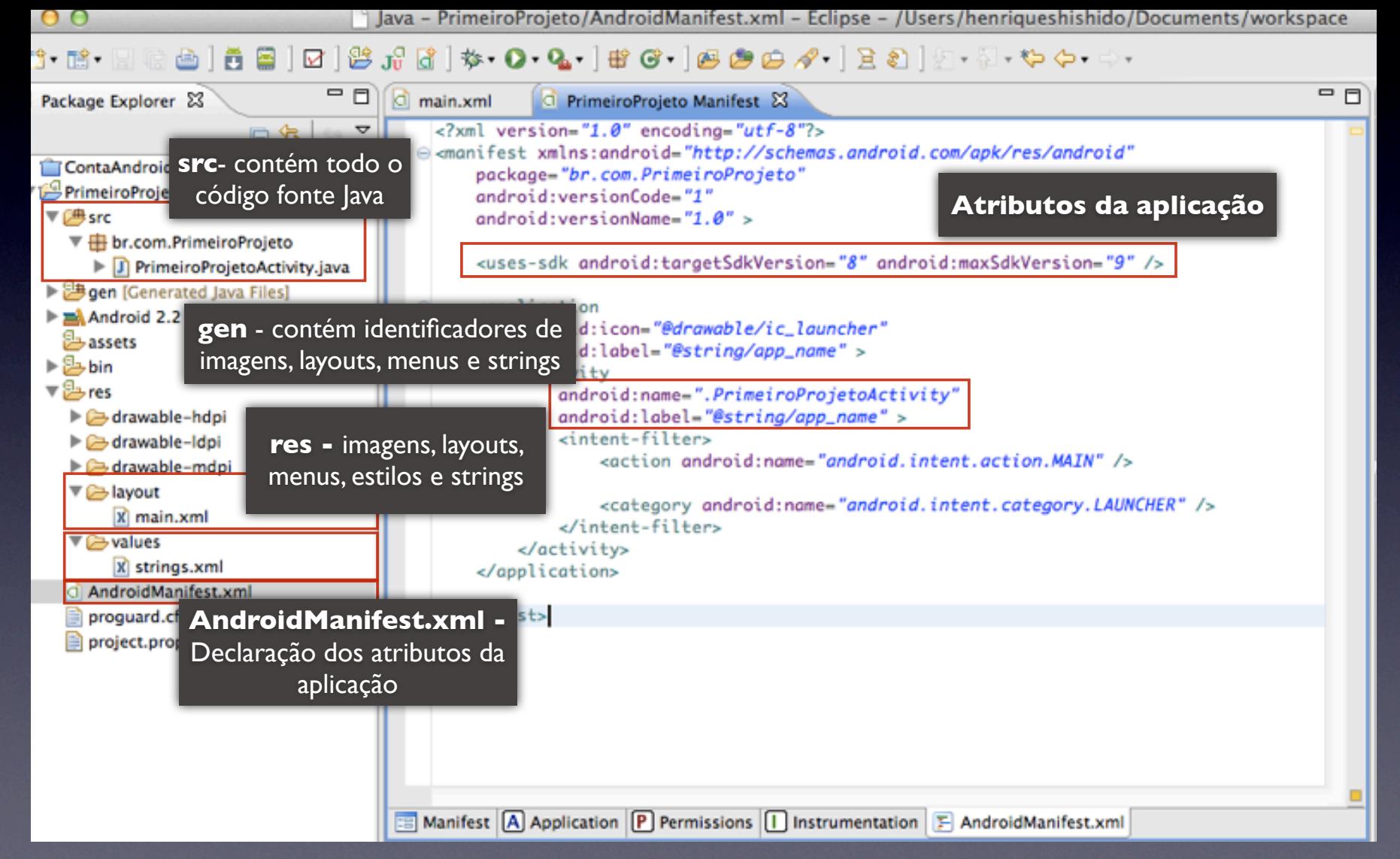


# Baixando um projeto Android

- Para começarmos a nos familiarizar com a IDE, baixe o seguinte projeto no [Moodle](#).
- Descompacte o projeto na pasta “.../workspace”
- Execute o Eclipse!



# Analizando um projeto

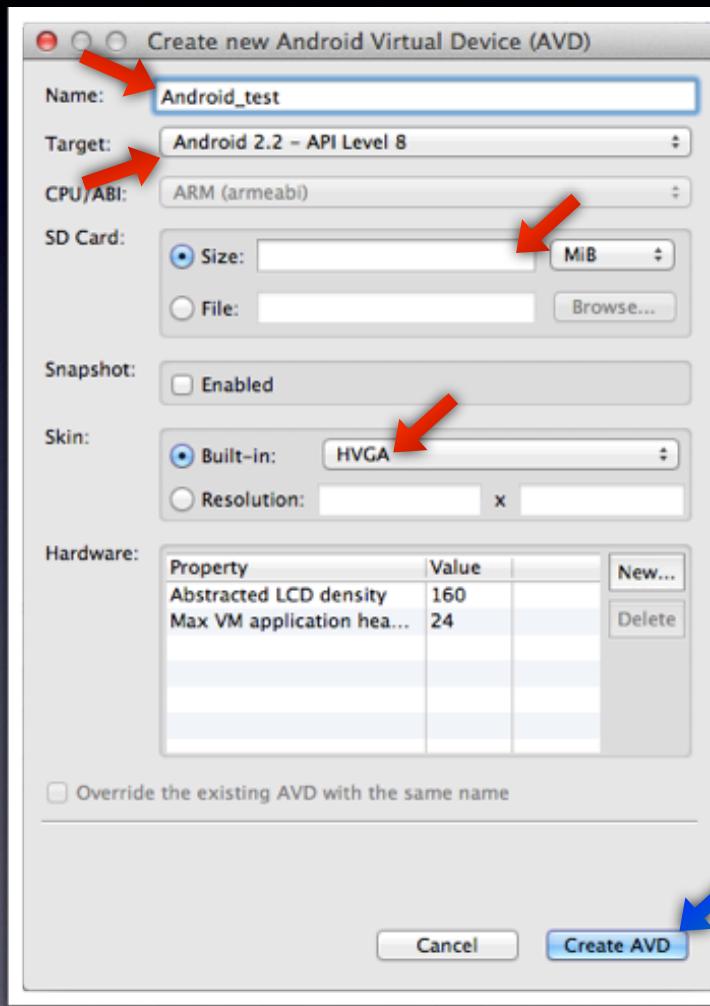


# Criando o dispositivo virtual Android

- Antes de executarmos um projeto, é necessário criar um dispositivo virtual (AVD)
  1. Clique no menu “Window” e selecione AVD Manager
  2. Clique em “New”
  3. Defina o nome da máquina em “Name”
  4. Escolha a versão da API Level em “Target” (sugiro Android 2.2)
  5. Defina o tamanho do cartão SD virtual (sugiro 64MB)
  6. Escolha a resolução do dispositivo (sugiro HVGA - 320x480)
  7. Finalize cliando em “Create AVD”



# Criando o dispositivo virtual Android

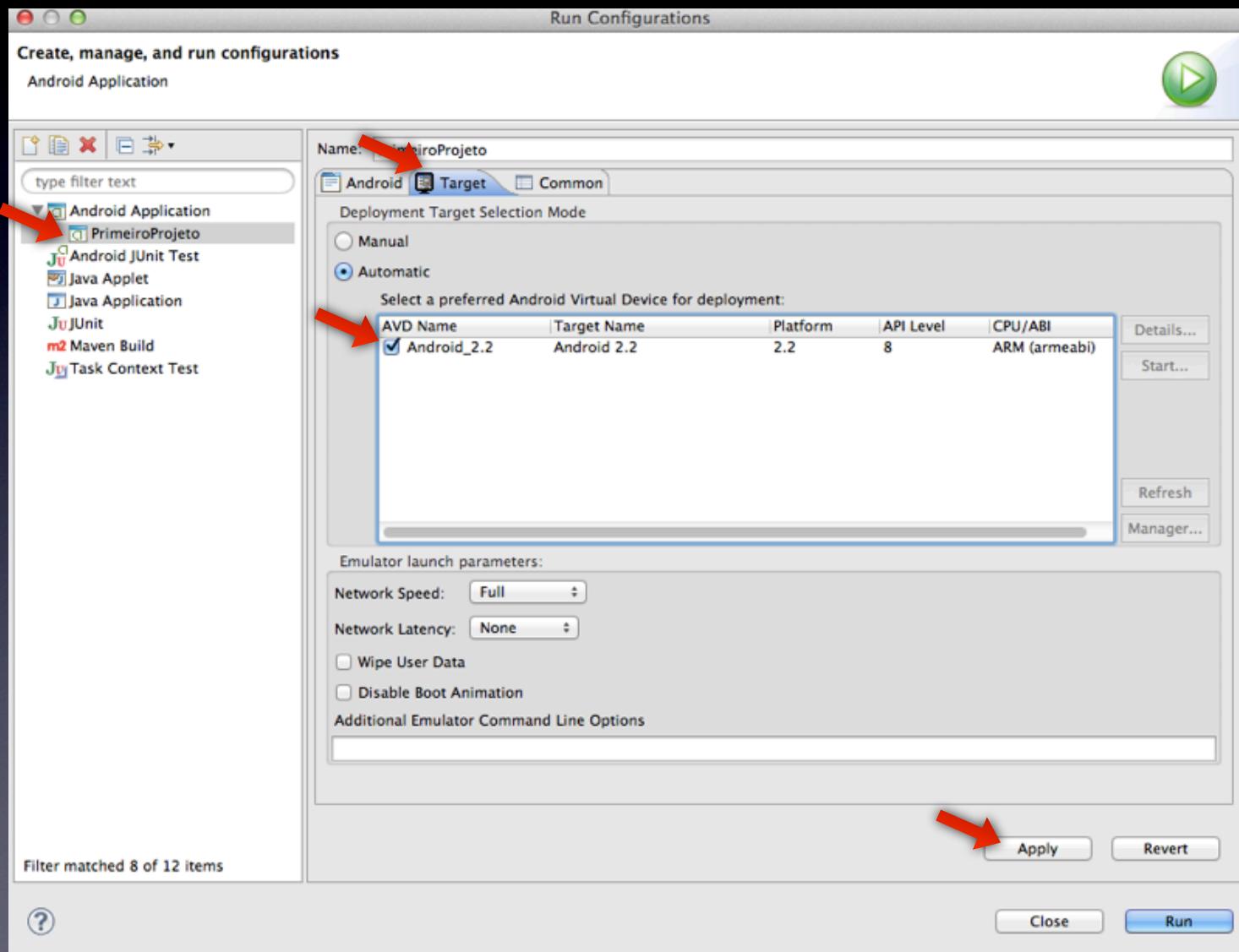


# Executando a aplicação...

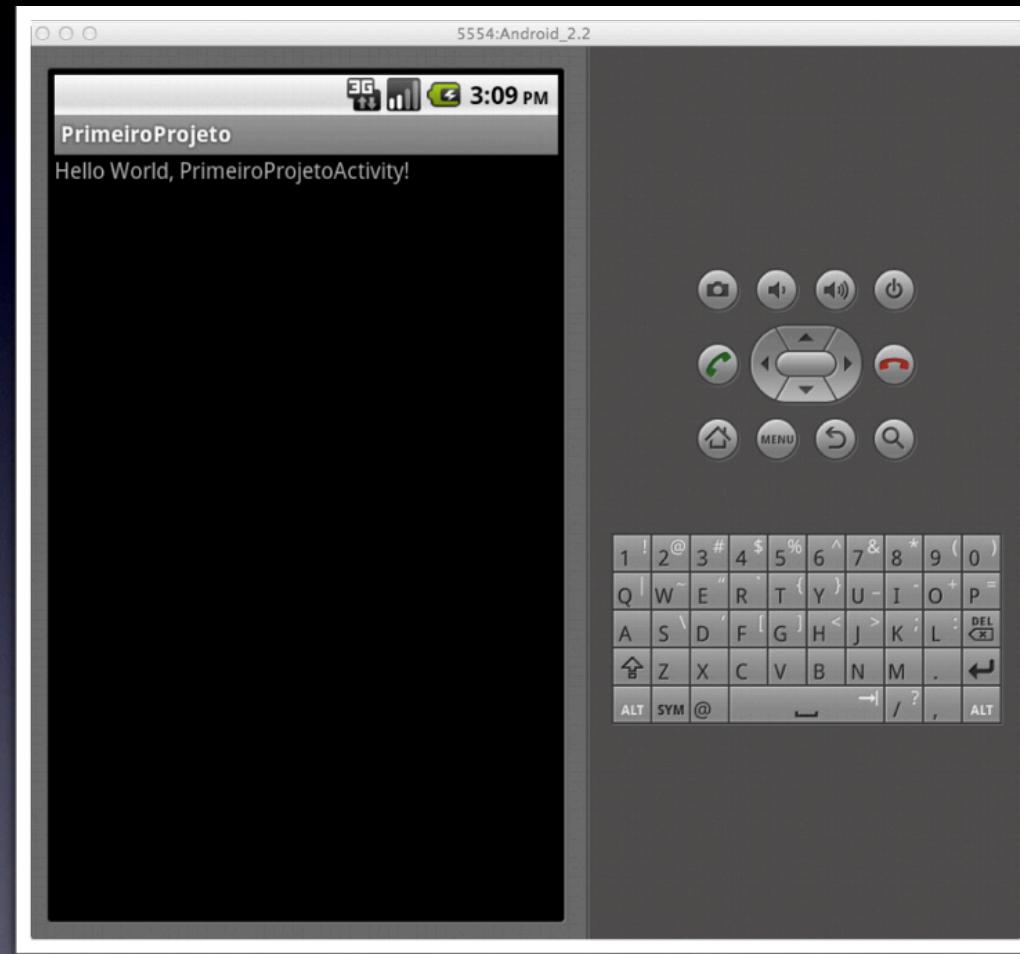
1. Após criar um novo dispositivo virtual, abra o projeto
2. Na barra lateral “*Project Explorer*”, clique com o botão direito sobre o projeto “*PrimeiroProjeto*” e selecione a opção “*Run as...*” => “*Run Configurations...*”
3. Selecione o “*PrimeiroProjeto*” em “*Android Application*” e abra a guia “*Target*”. Nela, você poderia escolher em qual AVD executar a sua aplicação
4. Para confirmar as alterações, clique em “*Apply*”
5. E, por fim, clique em “*Run*” para executar a sua aplicação



# Executando a aplicação...



# Projeto sendo executado!



ANDROID

**Sugestão:**Aproveite para navegar e aprender um pouco mais sobre o ambiente Android



# Dúvidas?

