

Programação para Dispositivos Móveis

Fatec Ipiranga
Análise e Desenvolvimento de Sistemas

Aula 03 – Introdução ao ambiente de desenvolvimento: Eclipse e Android SDK

Dalton Martins

dmartins@gmail.com

São Paulo, Fevereiro, 2012



Configurando o ambiente de trabalho

- O que é necessário para termos um bom ambiente de desenvolvimento?
 - O JDK como as bibliotecas de base do Java:
<http://java.sun.com/javase/downloads/index.jsp>
 - O Android SDK com as bibliotecas de base da plataforma Android:
<http://developer.android.com/sdk/index.html>
 - O Eclipse como ambiente de desenvolvimento:
<http://www.eclipse.org/downloads/>
 - O ADT Plugin para Eclipse, contendo ferramentas para facilitar a criação de projetos Android:
<http://dl.google.com/android/ADT-16.0.1.zip>
 - Para maiores informações sobre instalar a plataforma completa, dê uma olhada aqui: <http://developer.android.com/sdk/installing.html>

Conhecendo as ferramentas para desenvolvimento Android

- Explorando os novos ícones do plugin ADT no Eclipse → vamos abrir o Eclipse e dar uma olhada...
- Conhecendo o **SDK Manager**: é o gerenciador das versões de bibliotecas Android instaladas e a disposição para o desenvolvedor. Esse módulo permite termos várias versões diferentes que podem funcionar em vários equipamentos diferentes;
- Conhecendo o **AVD Manager**: é o gerenciador do tipo de dispositivos que iremos emular para teste, permite configurarmos tamanho do visor, memória RAM, memória de cartão, versão da API, entre outros...

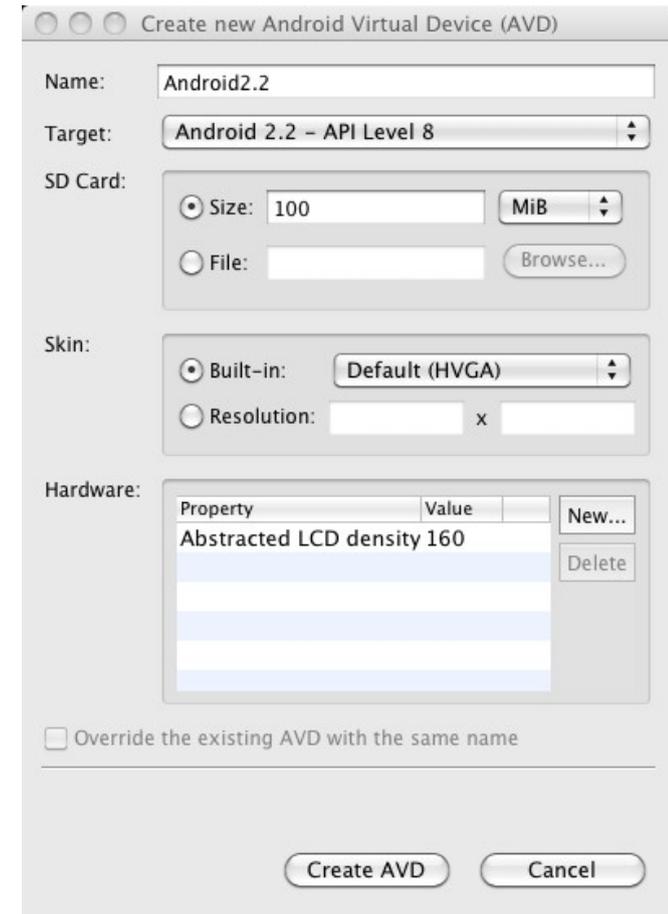
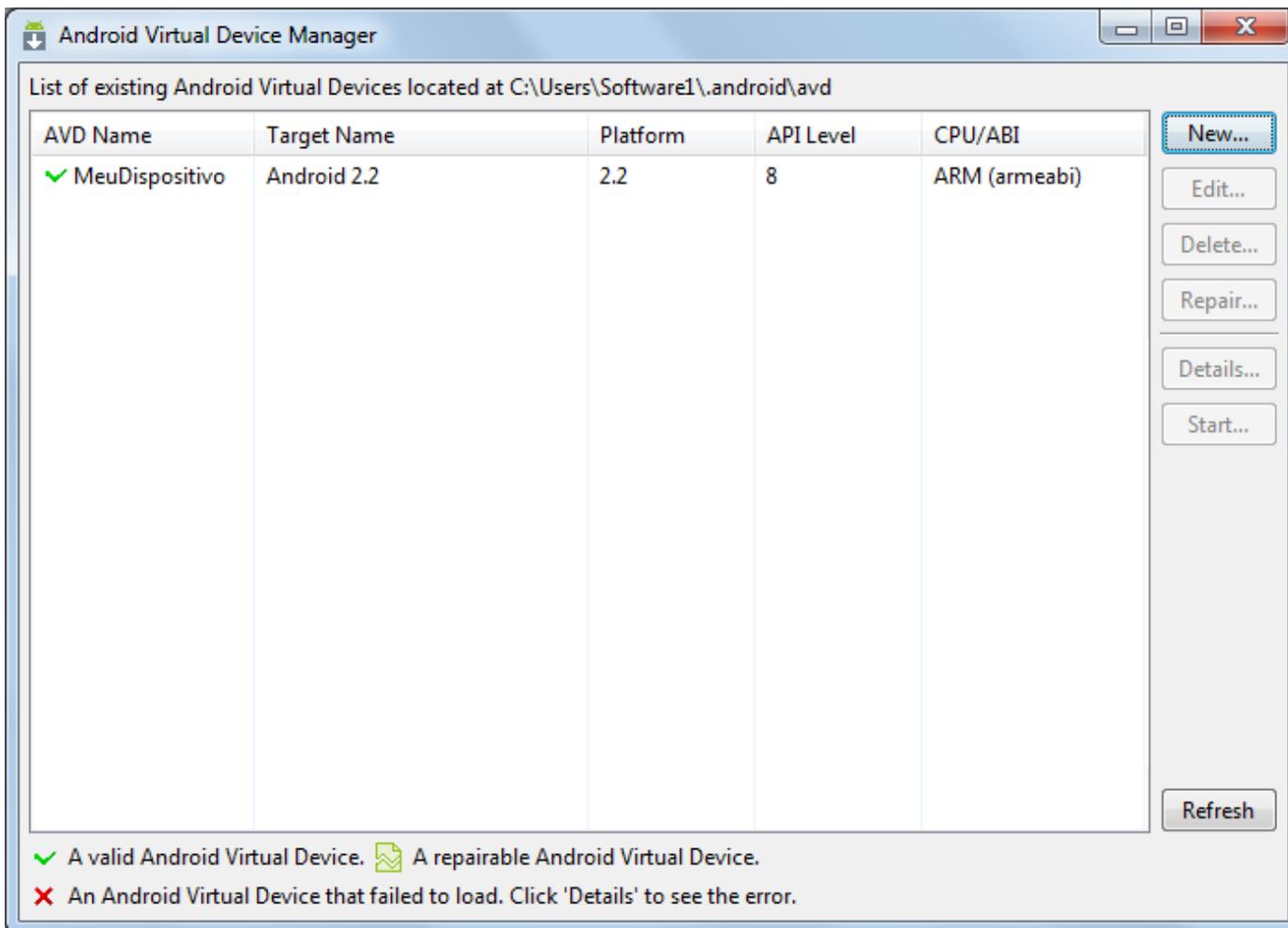
SDK Manager

The screenshot shows the Android SDK Manager window with the following details:

- Window Title:** Android SDK Manager
- SDK Path:** C:\Users\Rafael\Downloads\android-sdk-windows
- Table:** A table listing various packages with columns for Name, API, Rev., and Status.
- Filters:** Checkboxes for 'Updates/New', 'Installed', and 'Obsolete'. A link to 'Select New or Updates' is also present.
- Sort by:** Radio buttons for 'API level' (selected) and 'Repository'. A 'Deselect All' link is also available.
- Buttons:** 'Install packages...' and 'Delete packages...' buttons.
- Progress Bar:** A green progress bar at the bottom.
- Footer:** 'Fetching URL: http://www.echobykyocera.com/download/echo_repository.xml'

Name	API	Rev.	Status
Tools			
Android SDK Tools		16	Installed
Android SDK Platform-tools		10	Not installed
Android 4.0.3 (API 15)			
Android 4.0 (API 14)			
Android 3.2 (API 13)			
Android 3.1 (API 12)			
Android 3.0 (API 11)			
Android 2.3.3 (API 10)			
Android 2.2 (API 8)			
Android 2.1 (API 7)			
Android 1.6 (API 4)			
Android 1.5 (API 3)			
Extras			

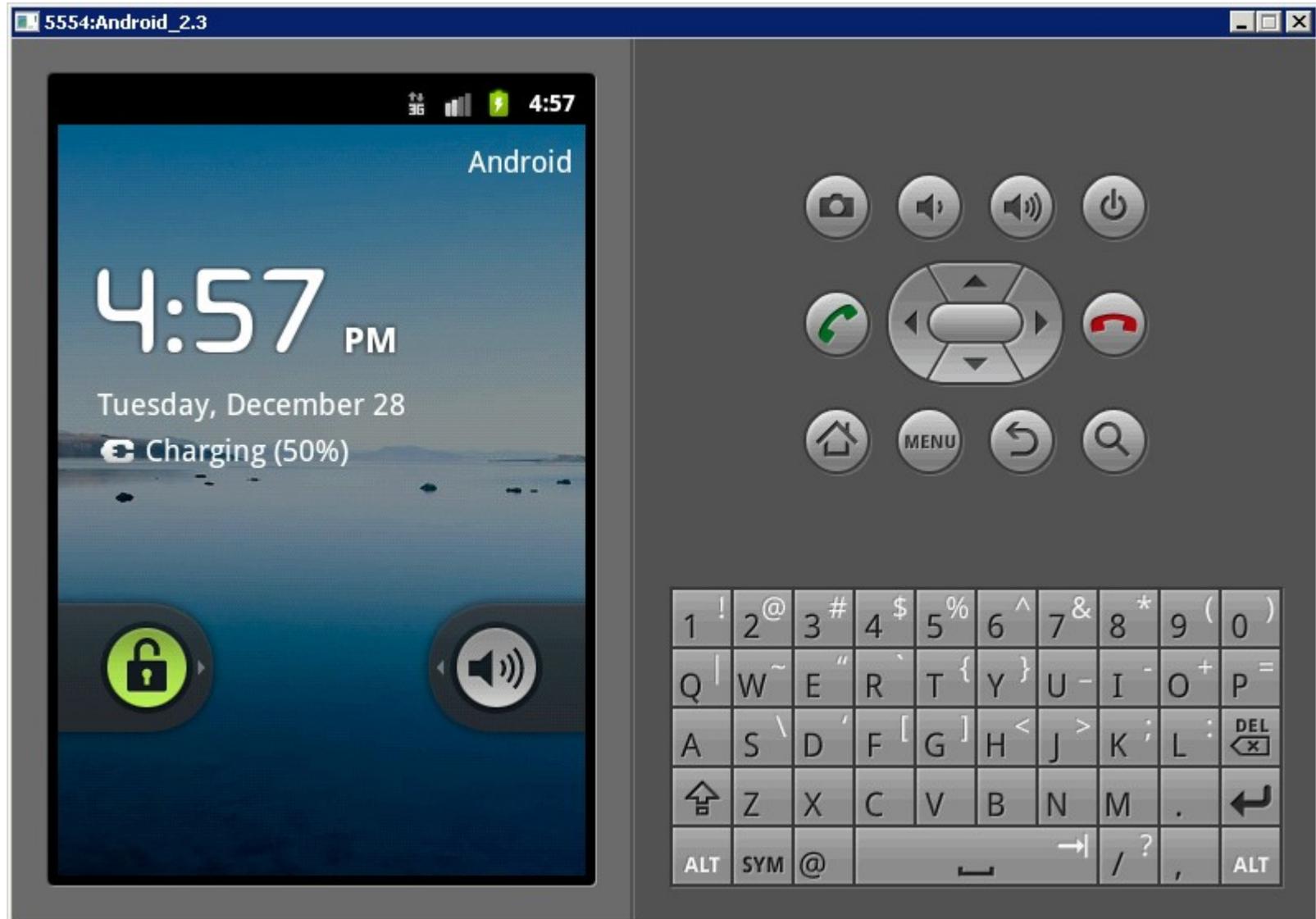
AVD Manager



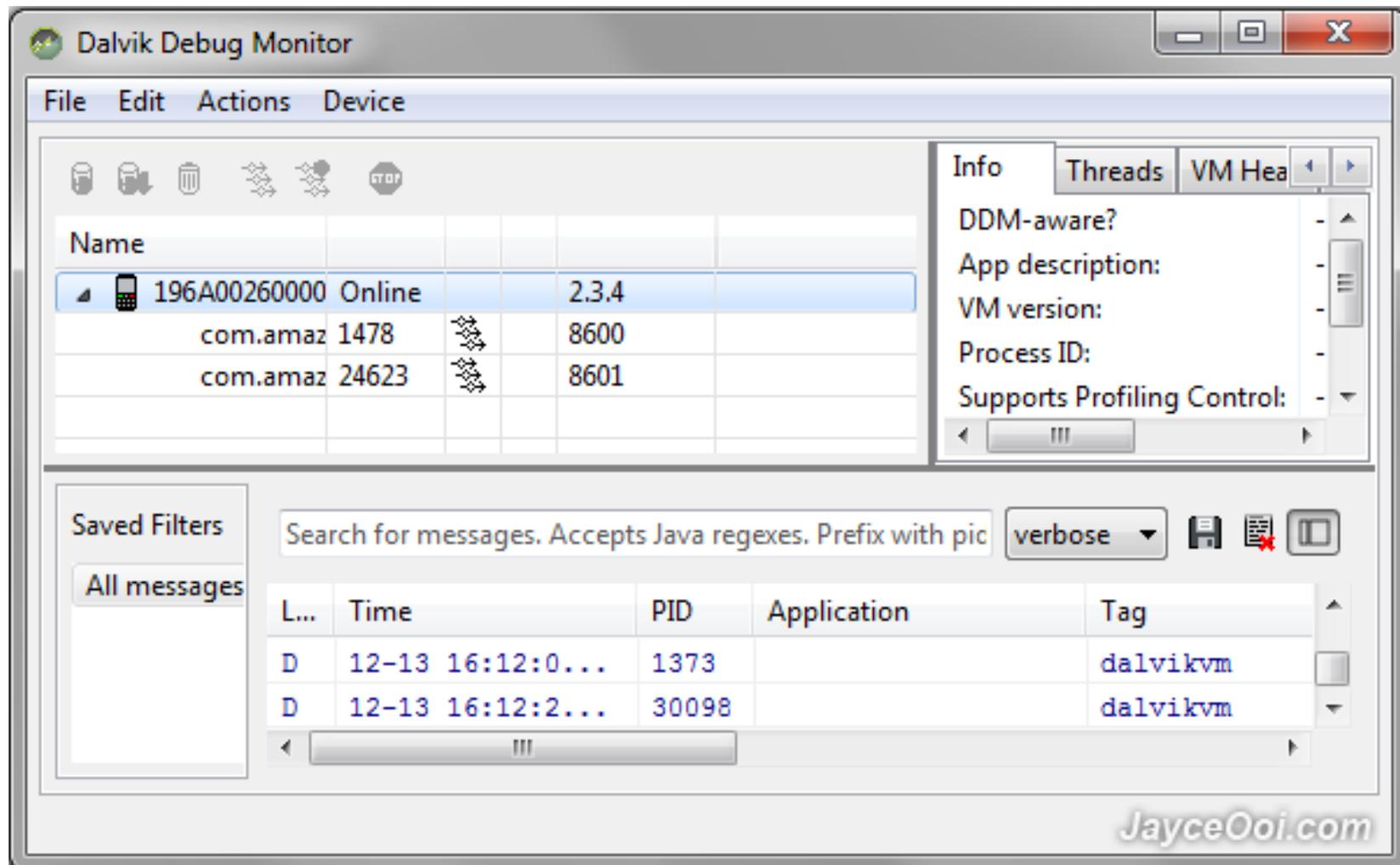
Conhecendo as ferramentas para desenvolvimento Android

- Conhecendo o **Android Emulator**: é o emulador de um celular Android.
- Conhecendo o **DDMS (Dalvik Debug Monitor Server)**: é uma ferramenta que funciona em modo de comando e também é integrada com o Eclipse. Ela permite acesso direto ao dispositivo no qual estamos testando nosso código, seja real ou emulado. Utilizamos o DDMS para visualizar e gerenciar processos, threads rodando no dispositivo, visualizar a pilha de dados, debugar processos e outras funções.
- Conhecendo o **Android Hierarchy Viewer**: é uma ferramenta visual que ilustra a relação entre componentes de layout, ajuda os desenvolvedores a projetar e debugar interfaces.

Android Emulator



DDMS – Dalvik Debug Monitor Server



Android Hierarchy Viewer

Devices Load View Hierarchy Display View Invalidate Request Layout

Operation	Duration (ms)
measure	3.342
layout	0.667
draw	24.693

The screenshot displays the Android Hierarchy Viewer interface. On the left, a view hierarchy tree is shown with the following structure:

- PhoneWindow\$DecorView #0@43aba7d0 NO_ID
 - LinearLayout #0@43aba90 NO_ID
 - FrameLayout #1@43abd20 id/content
 - LinearLayout #0@43abd200 NO_ID (highlighted in blue)
 - TextView #3@43abe160 id/fourthTextView
 - TextView #0@43abd458 id/firstTextView
 - TextView #4@43abe5c0 id/fifthTextView

On the right, a preview window titled "android.widget.LinearLayout@43abd200" is shown. It has a toolbar with "On White", "On Black", and "Show Extras" options. The preview area displays the rendered text:

Wow
Some
Text
More Text
And Still More

Testando o ambiente de desenvolvimento

- A melhor maneira de termos certeza de que o ambiente está instalado corretamente é testarmos com uma aplicação exemplo;
- Iremos fazer isso rodando o clássico jogo Cobras, onde iremos:
 - Criar um novo projeto Android;
 - Criar um novo perfil AVD;
 - Criar uma configuração de lançamento para o novo projeto.

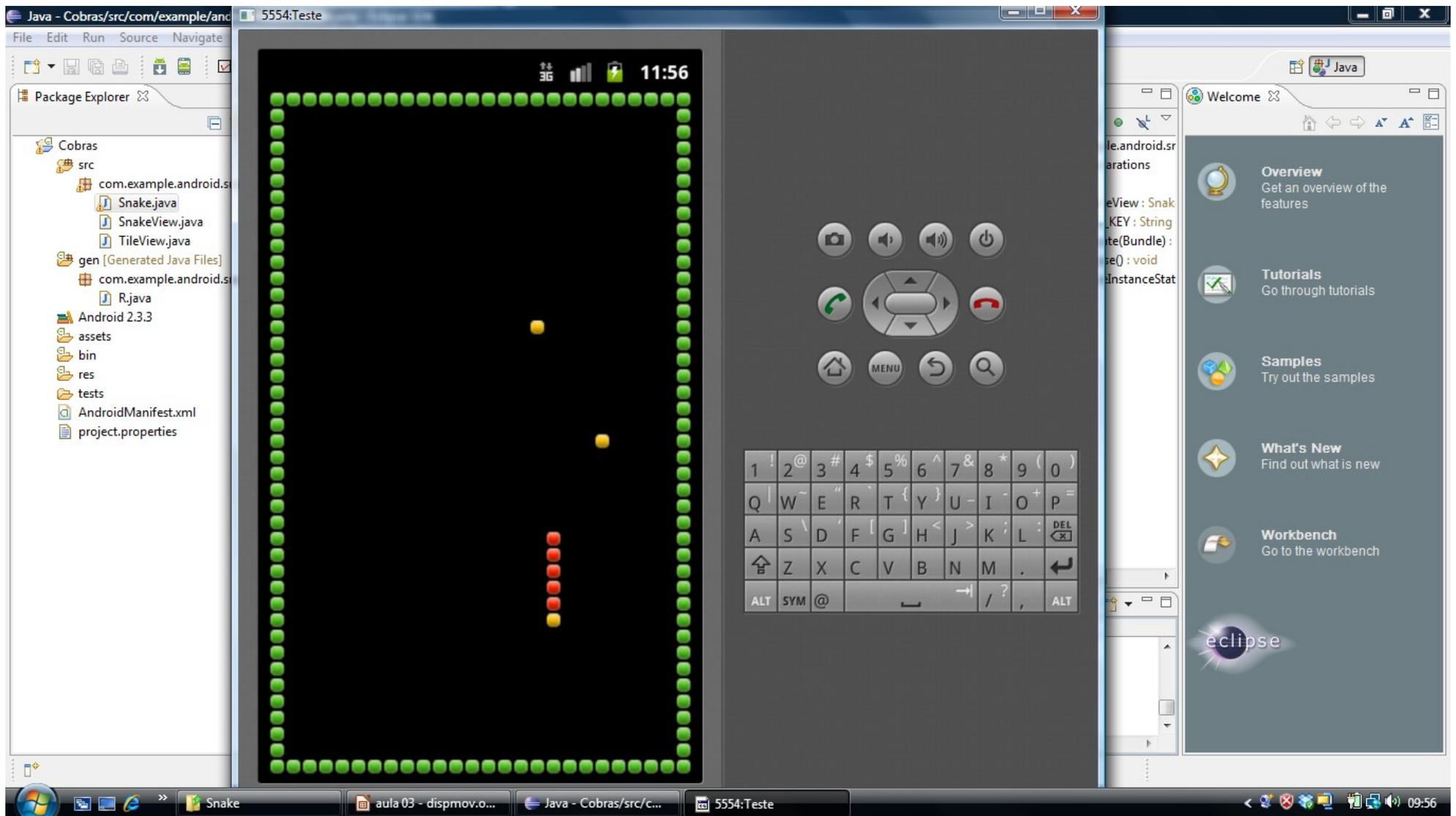
Configurando a aplicação Cobras

- Va no menu File → New → Project;
- Escolha Android → Android Project;
- Escolha a opção de criar projeto já existente;
- Navegue ate o diretorio onde se encontram os exemplos do Android SDK e escolha o diretorio Snake;
- Defina o nome da aplicação. Ex: Cobras.
- Preencha o nome do pacote com:
`com.example.android.snake`;

Criando um novo AVD para o projeto Cobras

- O próximo passo será criarmos um novo dispositivo para suportar a emulação do projeto:
 - Vamos abrir o AVD Manager e configura-lo localmente;
 - Agora, vamos criar uma configuração base para podermos executar a aplicação:
 - Será onde descreveremos que configurações de emulador queremos utilizar e qual o ponto de entrada para nossa aplicação;
 - Vamos ao menu Run → Run Configurations → Android: ali iremos selecionar o nome da configuração, o projeto e na aba Target, iremos selecionar a configuração AVD que criamos anteriormente;
 - Agora, vamos rodar o exemplo indo ao ícone Run → Run configurations → selecione a configuração que criamos e escolha o botão Run. Aguarde enquanto o simulador inicia...

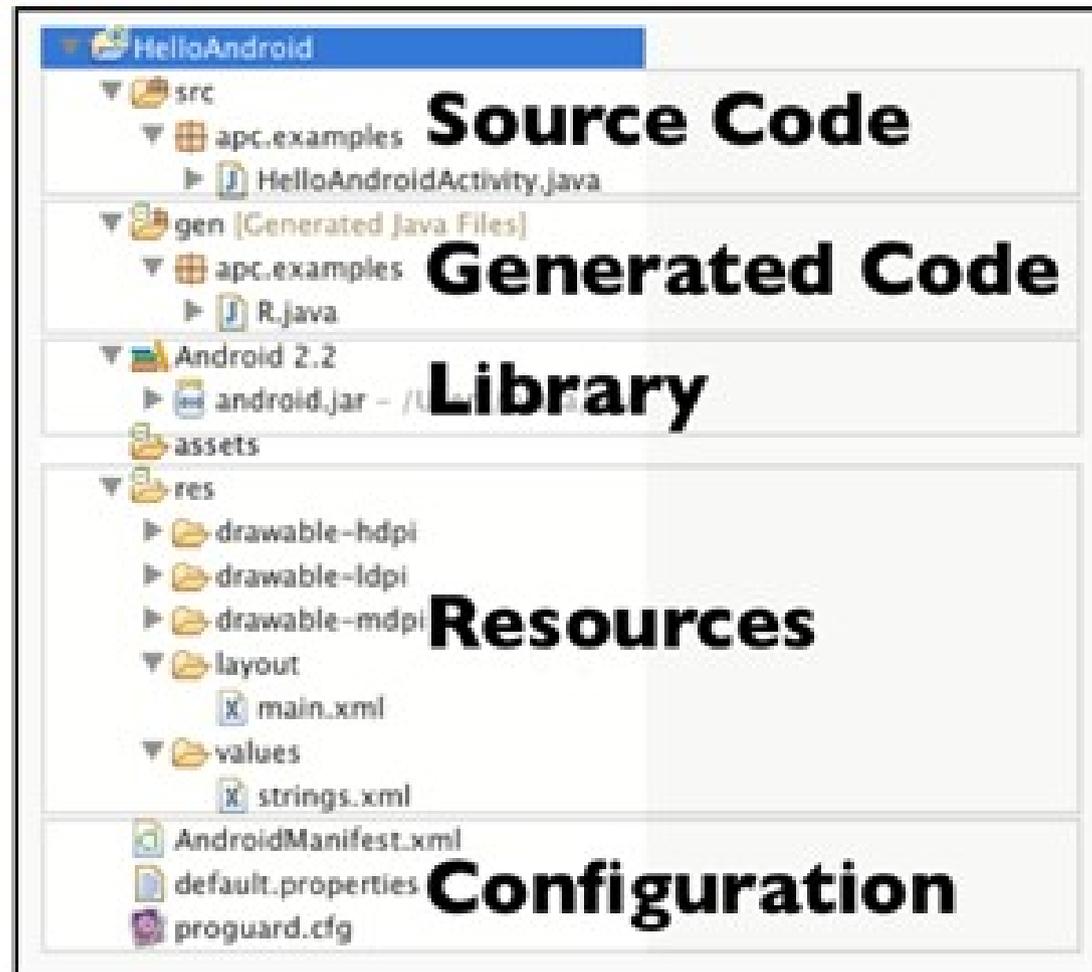
Emulador rodando....



Criando nossa primeira aplicação

- Vamos criar um novo projeto no Eclipse → ele criara toda a estrutura de arquivos e diretorios necessarios;
- Observando as configurações no Eclipse...
- Toda a aplicação Android possui um conjunto de arquivos e diretorios que são o nucleo de seu funcionamento. Esses arquivos são criados por padrao pelo Eclipse. Vejamos quais são eles...

Estrutura dos arquivos



Significado dos arquivos

- **AndroidManifest.xml** : arquivo de descrição global da aplicação. Define suas capacidades, permissões e como irá rodar;
- **Diretorio src**: onde todo o código fonte fica armazenado. Inclui o primeiro ponto de entrada da aplicação;
- **Diretorio gen**: onde recursos autogerados pelo plugin Android ficam armazenados;
- **Diretorio res**: onde todos os recursos de uma aplicação ficam armazenados: animações, imagens, arquivos de layout, arquivos XML, strings, dados em geral;
- **Diretorio res/drawable-***: onde fica o ícone em várias resoluções da aplicação;
- **Diretorio res/layout/main.xml**: onde fica o arquivo xml que define o layout da aplicação;
- **Diretorio res/values/strings.xml**: onde ficam os recursos de string da aplicação;
- **Diretorio assets**: onde ficam pedaços de dados que não queremos gerenciar como recursos.

Criando uma configuração de execução da aplicação

- Vamos ao menu Run → run configurations;
- Vamos agora criar uma configuração para debugar a aplicação:
 - Run → debug configurations
- Vamos configurar a aba Target como manual, permitindo que possamos escolher o AVD quando formos rodar a aplicação;
- Lembre-se de pressionar o botão Menu no emulador para destravar o Android!

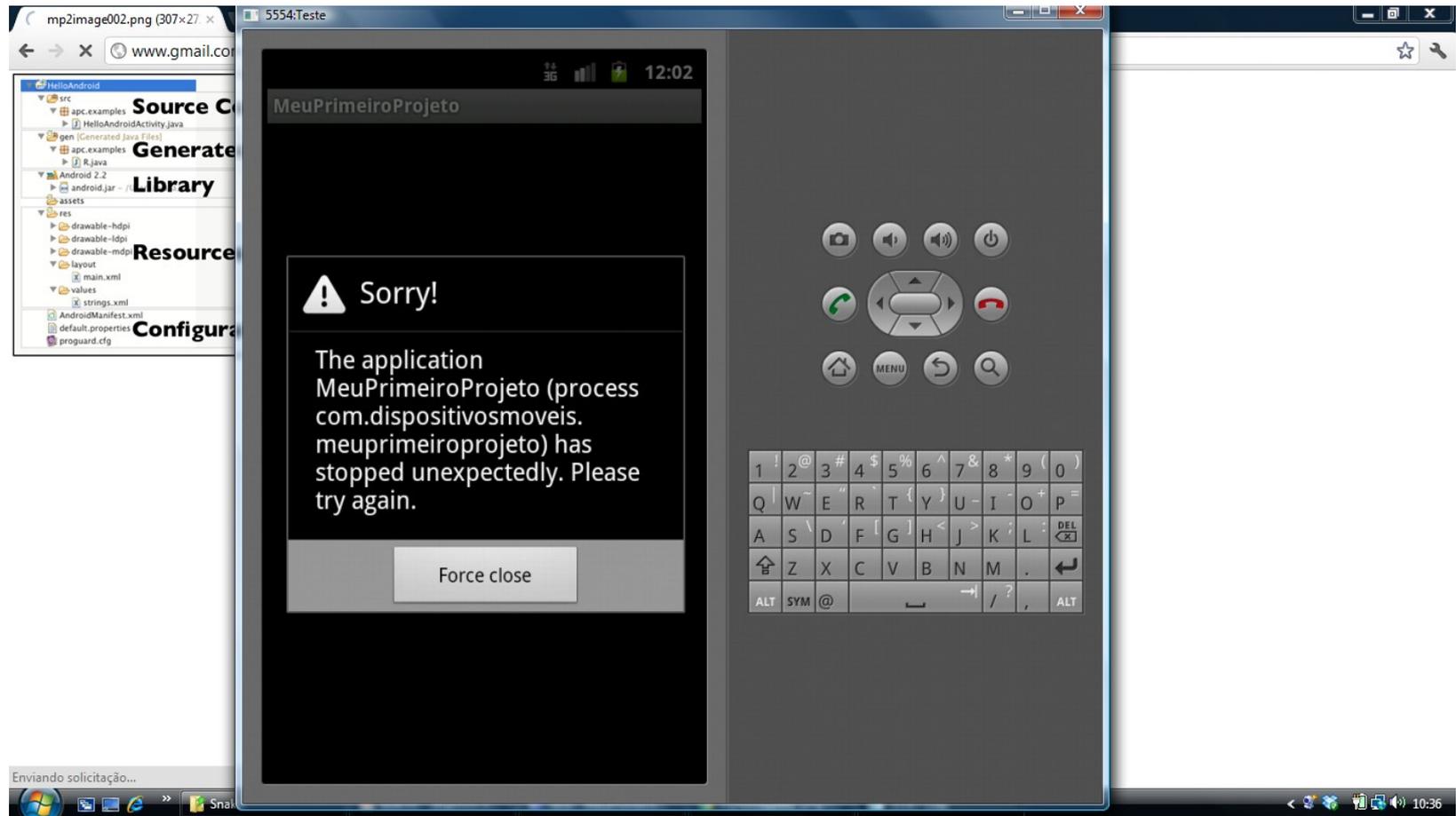
Aprendendo a debugar um programa

- Antes de avançarmos sera muito util entendermos melhor como debugar um programa;
- Vamos forçar um erro no programa. No arquivo de seu codigo fonte, crie esse novo metodo. Nao esqueça de chamar esse metodo dentro de *oncreate()*:

```
public void forceError() {  
    if(true) {  
        throw new Error("Ooooooppppssss!!!");  
    }  
}
```

Aprendendo a debugar um programa

- Vamos rodar a aplicação e ver o que acontece...



Aprendendo a debugar um programa

- Vamos agora rodar a configuração de Debug que criamos anteriormente;
- O Eclipse vai solicitar uma mudança de perspectiva no modo que organiza suas janelas. Isso vai nos facilitar a termos acesso a informações importantes para debugarmos o projeto;
- Vamos analisar algumas janelas, com foco especial na LogCat → ali podemos observar com mais detalhes o que esta acontecendo...
- Vamos criar um ponto de Debug na linha onde o metodo `forceError()` aparece no metodo `onCreate()` → fazemos isso clicando na extrema esquerda da linha e selecionando **Toggle Breakpoint** ! Vejamos o que ocorre...
- Vamos mudar de perspectiva e dar uma olhada no modo DDMS.

Exercicios

- De uma olhada em outros exemplos que estão disponíveis no SDK Android.
- Escolhe alguns e tente criar projetos que permitam rodá-los, de modo que você possa ir se familiarizando melhor com o funcionamento da plataforma que iremos utilizar ao longo do curso.