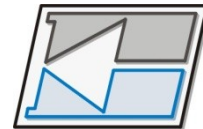


Introdução à Programação Android

Prof. Samir Bonho

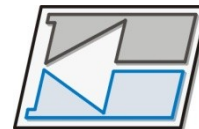
Aula 1

Florianópolis, 19 de Fevereiro de 2014

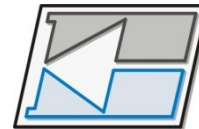


Sumário

- ✓ Apresentação da disciplina
- ✓ Introdução ao ambiente Android
- ✓ Fontes: Lecheta

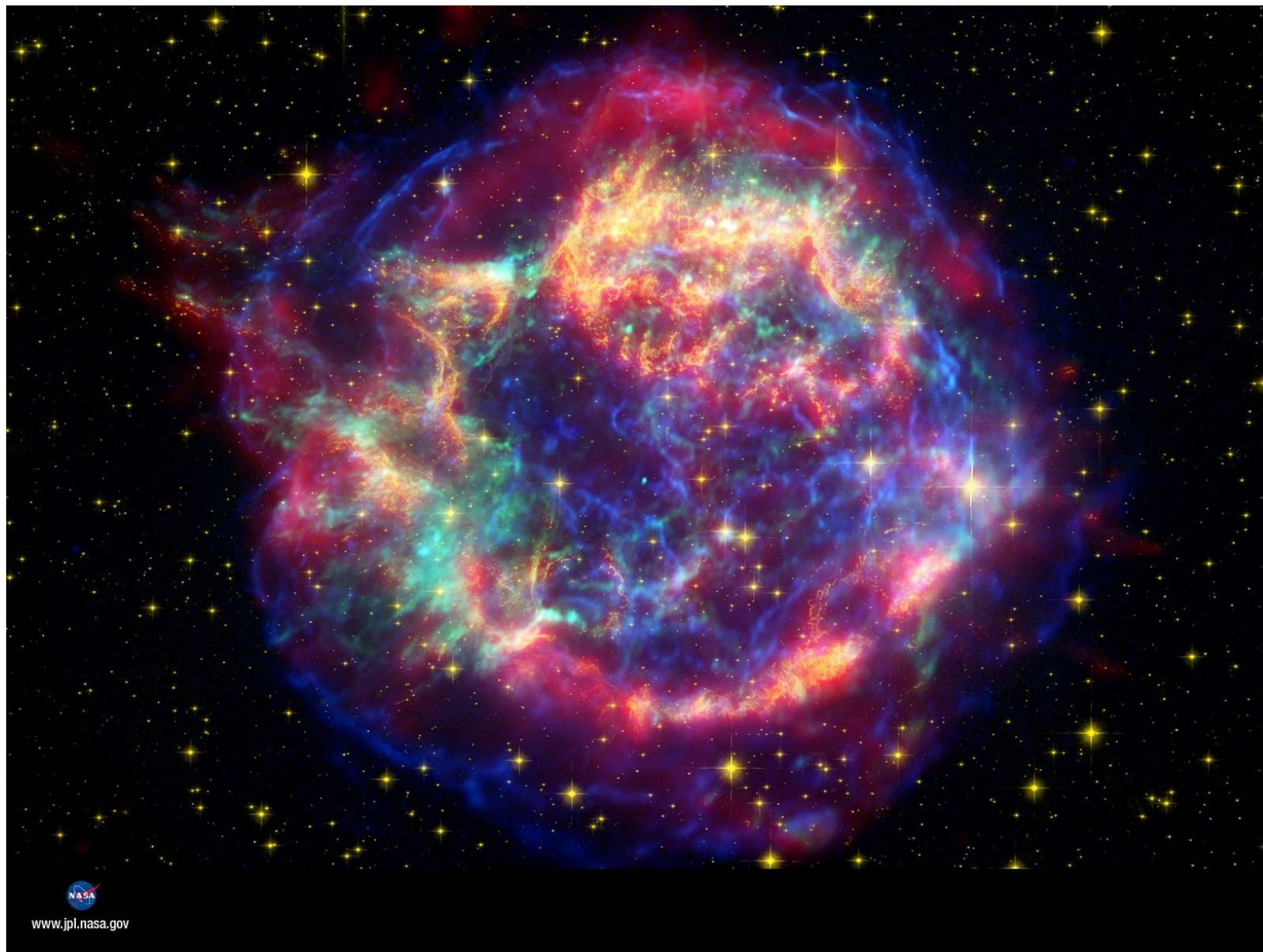
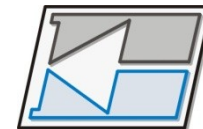


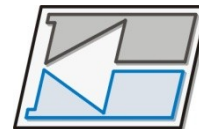
Apresentação da disciplina



Plano de ensino

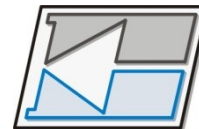
- ✓ *Datas*
- ✓ *Avaliações*
- ✓ *Projeto*



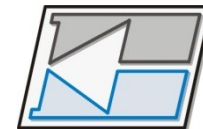


Regras gerais

- ✓ *Proibido comer dentro de sala de aula*
- ✓ *Telefones celulares*
- ✓ *Organização*
- ✓ *Bom senso*



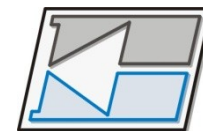
Introdução



Introdução

- Situações onde a internet é acessada no celular (Fonte: Estudo Especial Mobile: 4.600 entrevistas (25 de abril a 2 de maio de 2012))

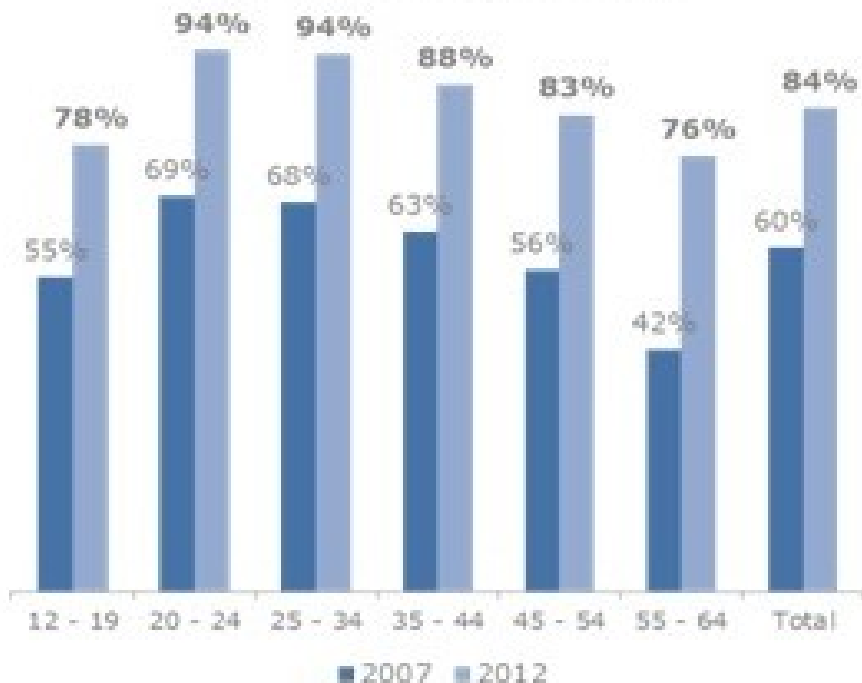




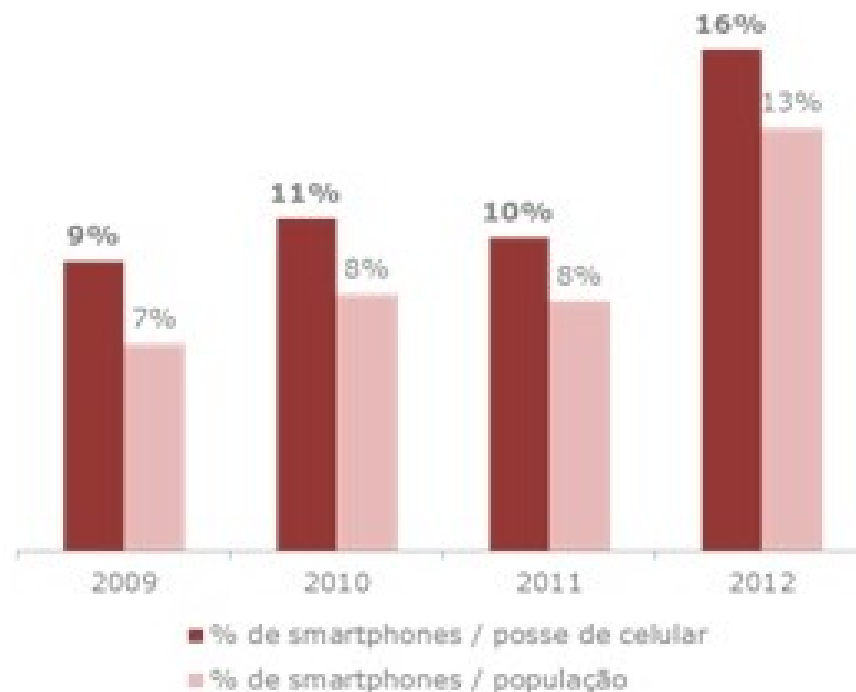
Introdução

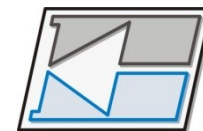
- (Fonte: Estudo Especial Mobile: 4.600 entrevistas (25 de abril a 2 de maio de 2012))

Posse de celular por idade



% de posse de Smartphone

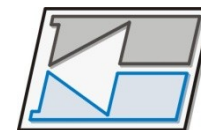




Introdução

- Conteúdo acessado por meio da internet no celular (Fonte: Estudo Especial Mobile: 4.600 entrevistas (25 de abril a 2 de maio de 2012))





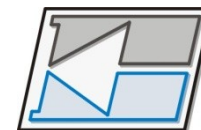
Introdução

- Números do mercado global: Marcas

Worldwide Smartphone Sales to End Users by Vendor in 3Q13 (Thousands of Units)

Company	3Q13 Units	3Q13 Market Share (%)	3Q12 Units	3Q12 Market Share (%)
Samsung	80,356.8	32.1	55,054.2	32.1
Apple	30,330.0	12.1	24,620.3	14.3
Lenovo	12,882.0	5.1	6,981.0	4.1
LG Electronics	12,055.4	4.8	6,986.1	4.1
Huawei	11665.7	4.7	7,804.3	4.5
Others	102941.8	41.1	70206.8	40.9
Total	250,231.7	100.0	171,652.7	100.0

Source: Gartner (November 2013)



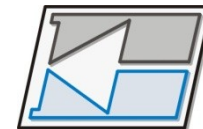
Introdução

- Números do mercado global: O.S

Worldwide Smartphone Sales to End Users by Operating System in 3Q13 (Thousands of Units)

Operating System	3Q13 Units	3Q13 Market Share (%)	3Q12 Units	3Q12 Market Share (%)
Android	205,022.7	81.9	124,552.3	72.6
iOS	30,330.0	12.1	24,620.3	14.3
Microsoft	8,912.3	3.6	3,993.6	2.3
BlackBerry	4,400.7	1.8	8,946.8	5.2
Bada	633.3	0.3	4,454.7	2.6
Symbian	457.5	0.2	4,401.3	2.6
Others	475.2	0.2	683.7	0.4
Total	250,231.7	100.0	171,652.7	100.0

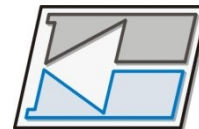
Source: Gartner (November 2013)



Introdução

- Oportunidades

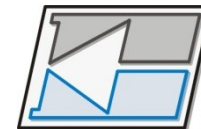




Introdução

- Oportunidades

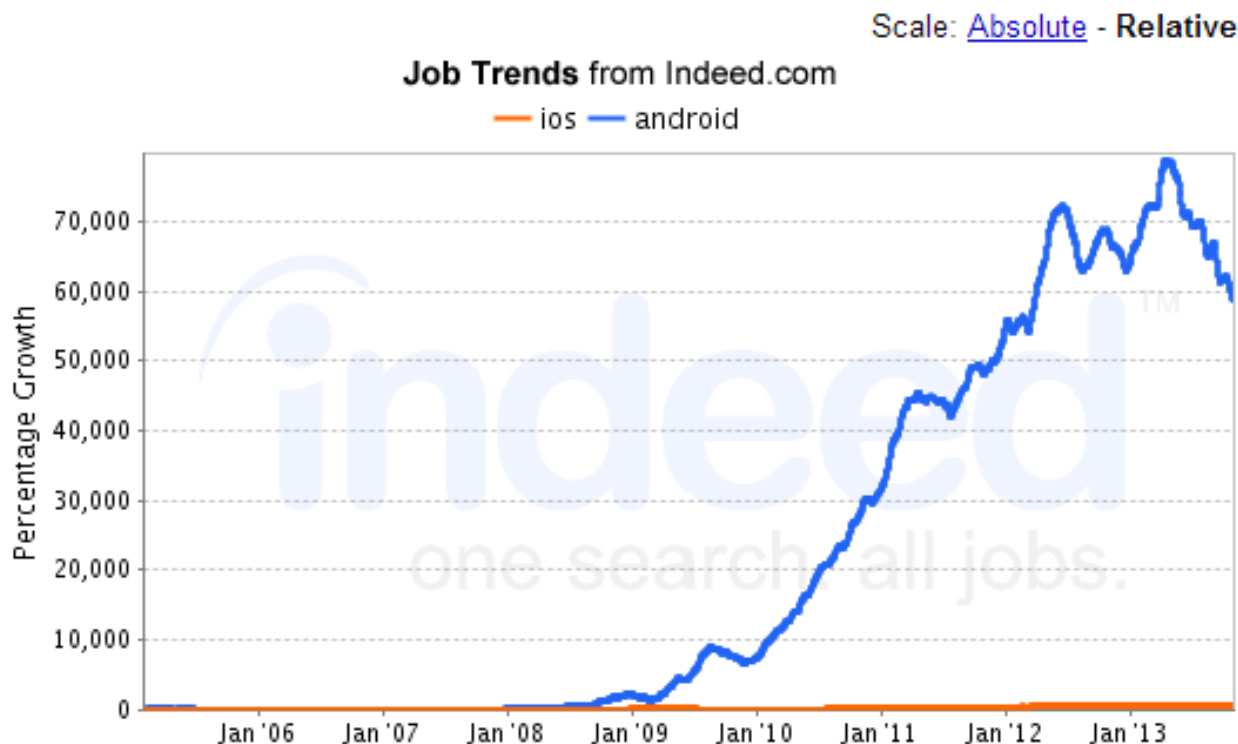




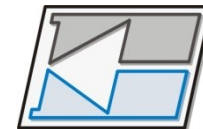
Introdução

- E o trabalho?

ios, android Job Trends



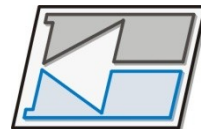
Indeed.com searches millions of jobs from thousands of job sites.
This job trends graph shows relative growth for jobs we find matching your search terms.



Introdução: Vantagens Android

- Android possui algumas vantagens...

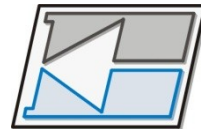




Introdução: Vantagens Android

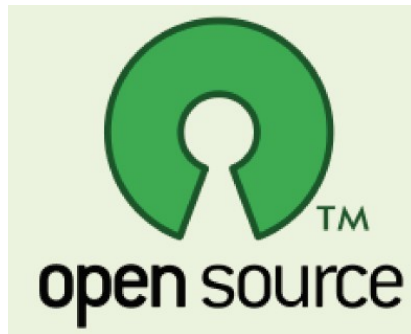
- Dispositivos para todos os bolsos (\$).



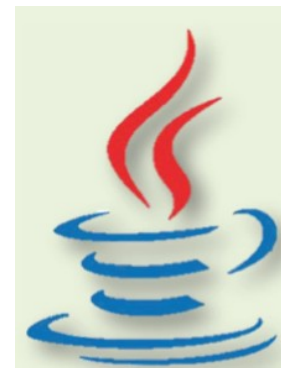


Introdução: Vantagens Android

- Open Source

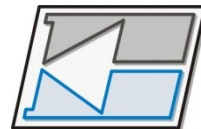


- Desenvolvimento em JAVA



- Programação em Windows, Linux e Mac

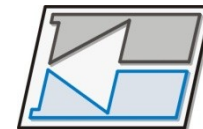




Introdução: Vantagens Android

- Android vai além dos smartphones e tablets:

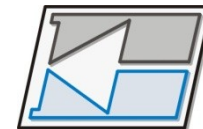




Introdução: Vantagens Android

- Android vai além dos smartphones e tablets:

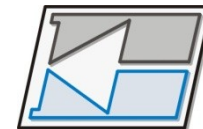




Introdução: Vantagens Android

- Android vai além dos smartphones e tablets:
 - Sony SmartWatch o relógio Android da Sony

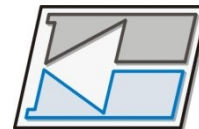




Introdução: Vantagens Android

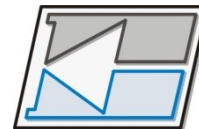
- Android vai além dos smartphones e tablets:



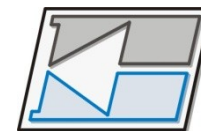


Introdução: Vantagens Android





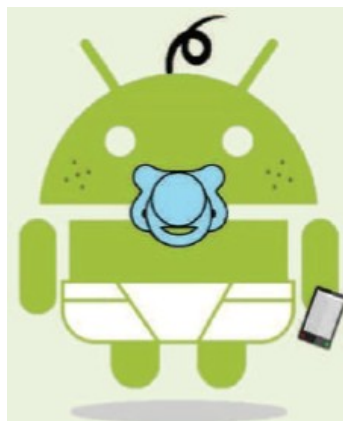
Histórico

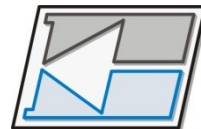


Histórico

Outubro de 2003

- Android, Inc. foi fundada em Palo Alto, California por Andy Rubin, Rich Miner, Nick Sears e Chris White.

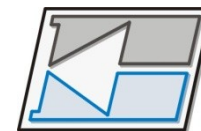




- Google acquire Android Inc.

Julho de 2005



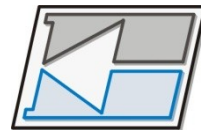


Histórico

- A Open Handset Alliance é formada.



5 de Novembro de 2007

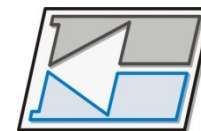


Histórico

- Android Beta SDK é disponibilizado.



12 de Novembro de 2007

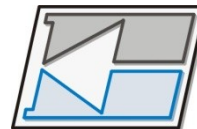


Histórico

- Lançado o primeiro smartphone com Android 1.0, o HTC Dream (G1).

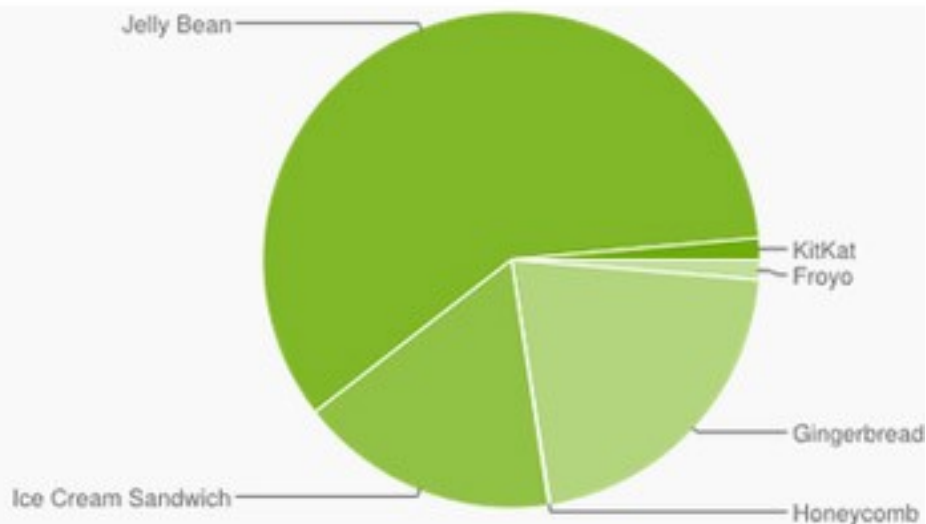


23 de Setembro de 2008



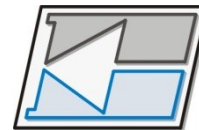
Histórico

Version	Codename	API	Distribution
2.2	Froyo	8	1.3%
2.3.3 - 2.3.7	Gingerbread	10	21.2%
3.2	Honeycomb	13	0.1%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	16.9%
4.1.x	Jelly Bean	16	35.9%
4.2.x		17	15.4%
4.3		18	7.8%
4.4	KitKat	19	1.4%

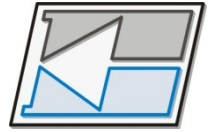


Data collected during a 7-day period ending on January 8, 2014.

Any versions with less than 0.1% distribution are not shown.

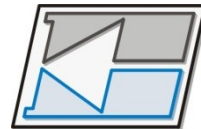


Android



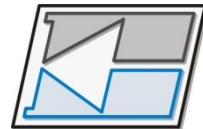
Android: Intro

- O que é??
- ✓ “Android is the world’s most popular mobile platform... Android devices are already smart (and will only get smarter)...” (Google)
- ✓ “Sistema operacional baseado em Linux desenvolvido especialmente para dispositivos móveis” (Wikipedia).



Android: Intro

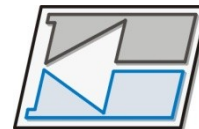
- **Framework** para construção de aplicações!!



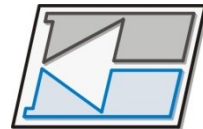
Android: Intro

- Framework para construção de aplicações:
 - ✓ APIs para: Construções gráficas, notificações, navegação, acesso a web, comunicação interprocessos, gerência de memória, rede (wi-fi, 3G, GPRS, BT, GPS, NFC...), etc.
 - ✓ Google Play Store!!
 - ✓ OpenGL -> Jogos!!!

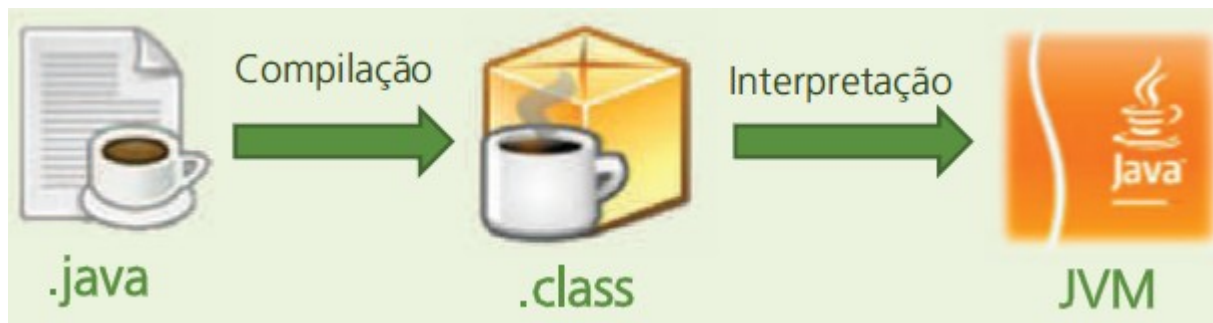


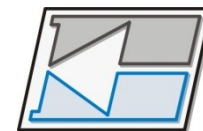


Desenvolvimento

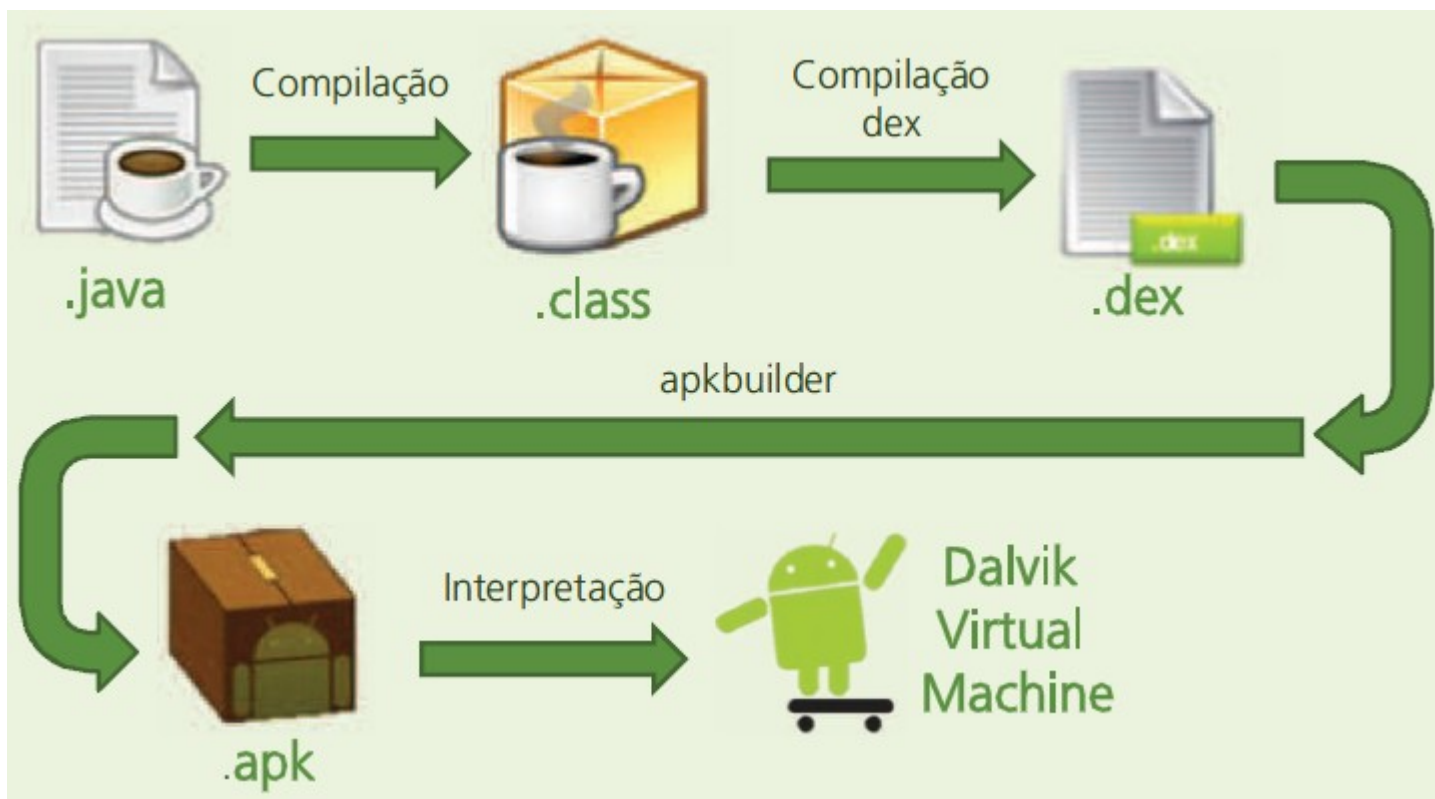


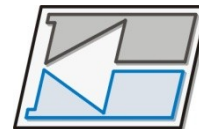
- Processo de desenvolvimento Java tradicional



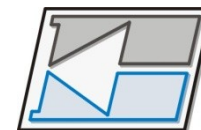


- Processo de desenvolvimento Android

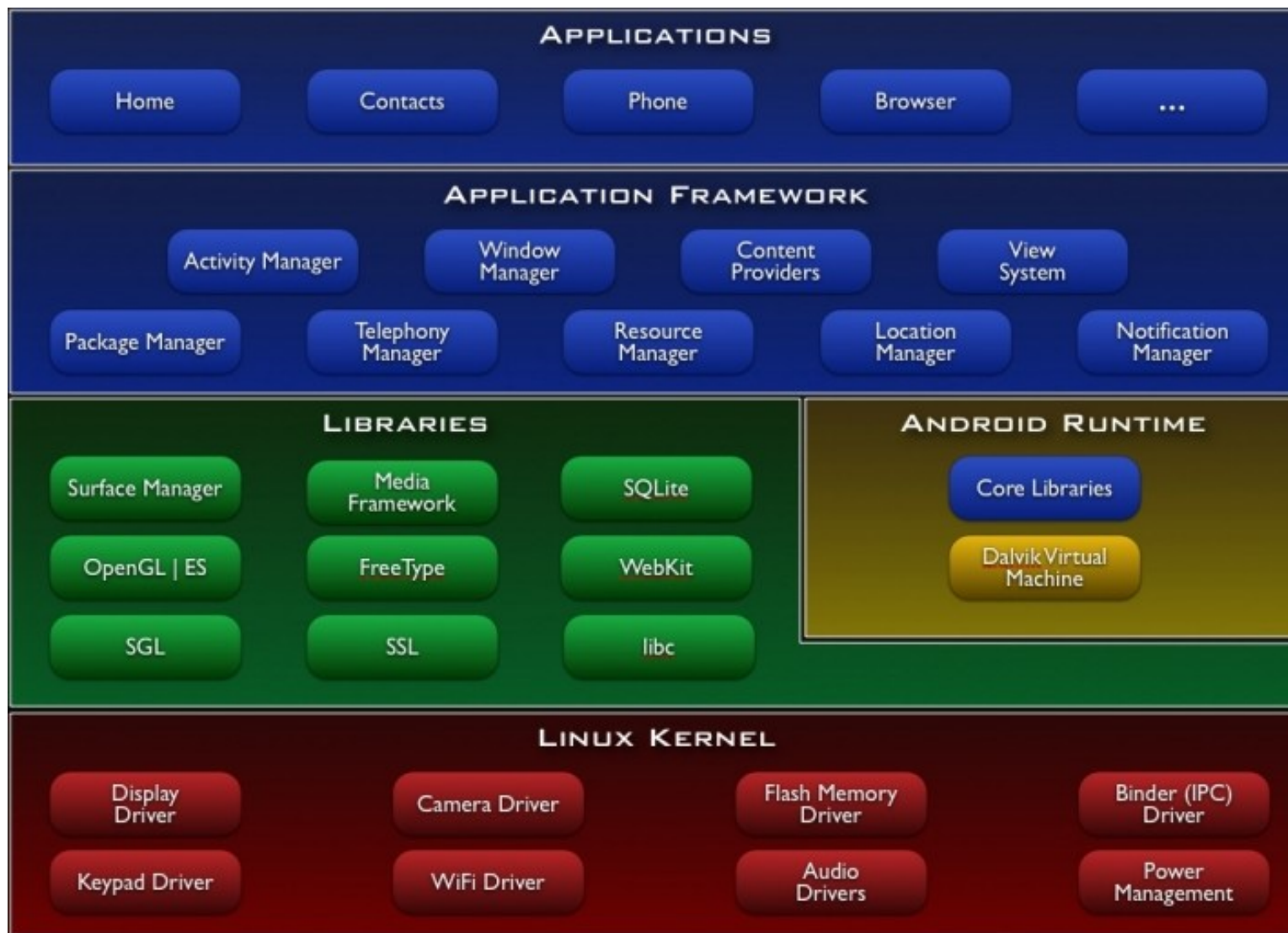


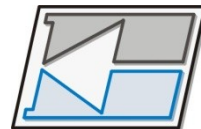


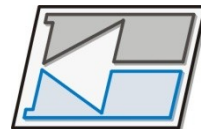
Arquitetura

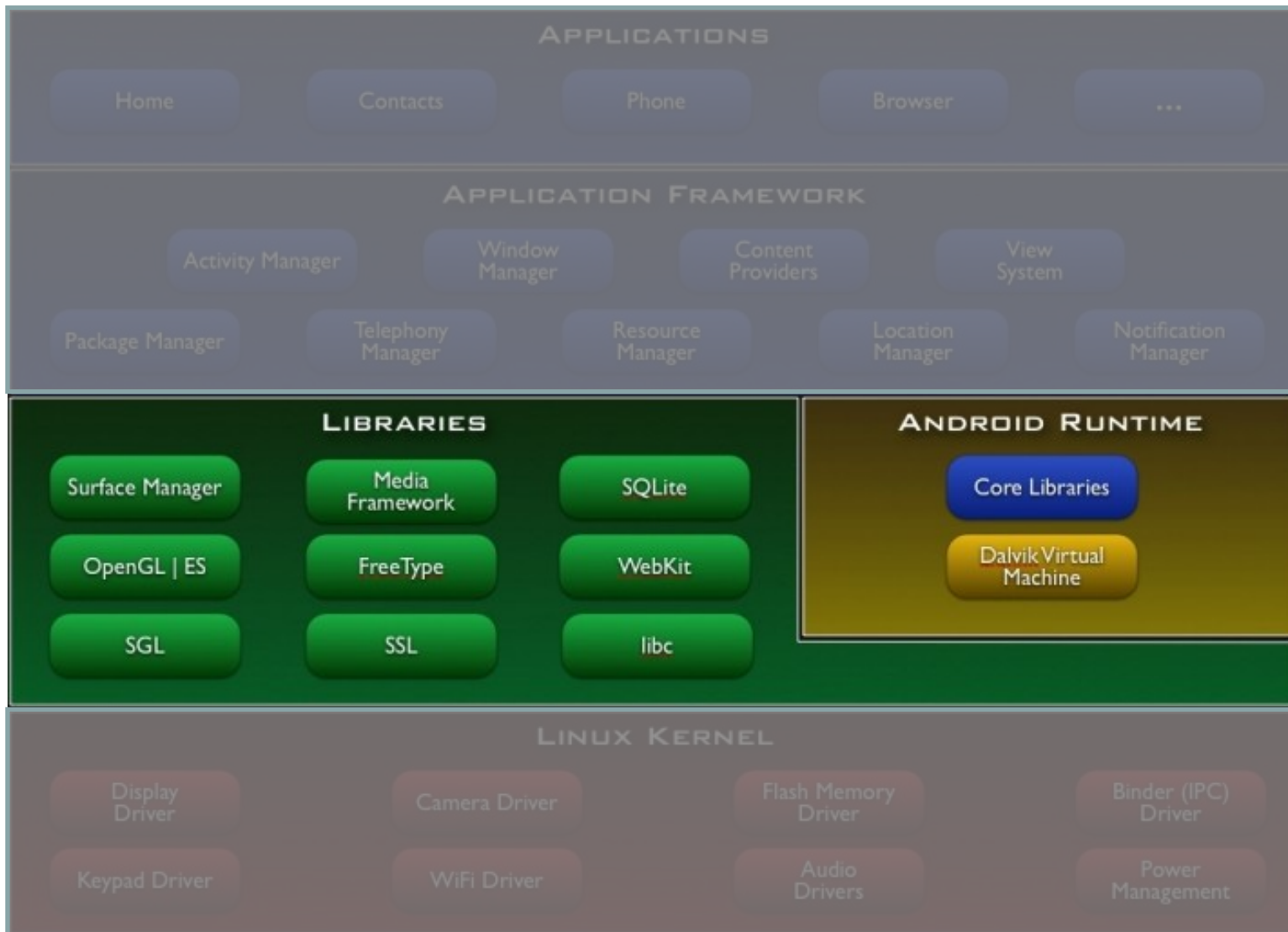
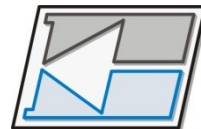


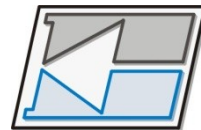
Android: Arquitetura





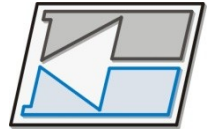




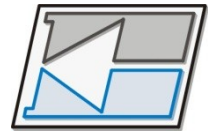


Android: Arquitetura



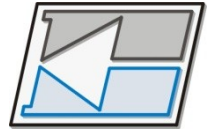


- É código-aberto!!
- ✓ Posso modificar o código e criar um Android só para mim?
- ✓ Posso criar aplicações para áreas específicas? (automotiva, médica, prospecção mineral, etc) ?
- ✓ Posso estender e criar aplicações específicas para as áreas em que trabalho???



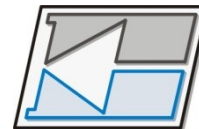
Android: Arquitetura

- ✓ E se eu precisar de alguma funcionalidade que não está presente no framework Android? Como faço?
- ✓ Por Exemplo: Conectar minha aplicação Android a uma rede CAN de um automóvel para obter dados em RT do mesmo. É possível?



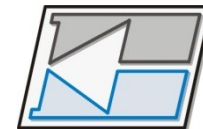
Android: Arquitetura

- **Android** Native Development Kit (NDK)
 - ✓ Conjunto de ferramentas que permite ao desenvolvedor incluir em suas aplicações componentes em código-nativo (C/C++).
 - ✓ Principais vantagens:
 - Reuso de código; e
 - Em alguns casos, aumento na performance.

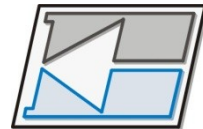


Ferramentas necessárias



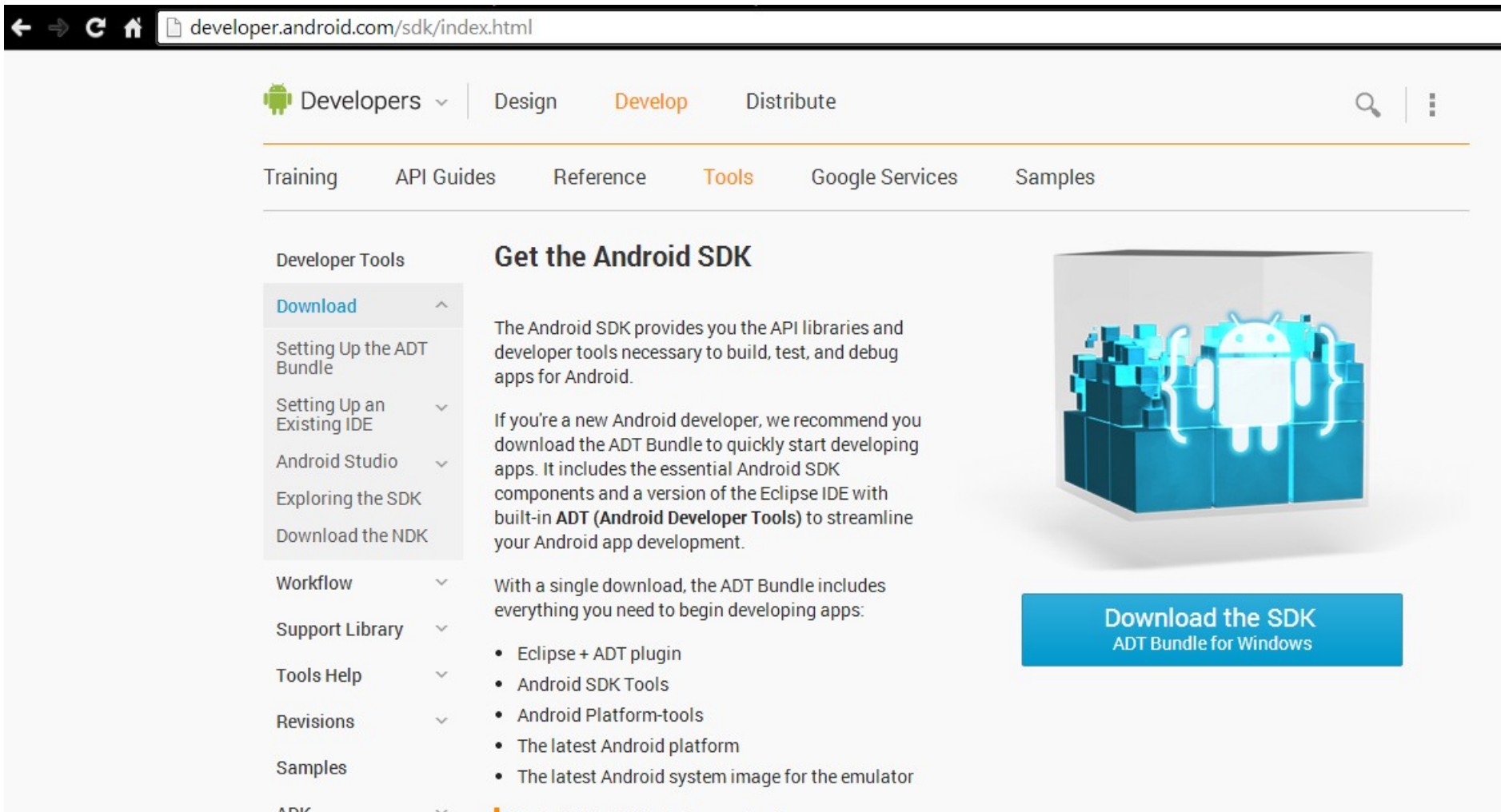
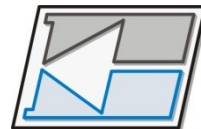


Instalação/Configuração ADT (Android Developer Tools)



Instalação/Configuração ADT

- ADT = Bundle de ferramentas para desenvolvimento Android.
 - ✓ Eclipse + plugins ADT
 - ✓ Android SDK e ferramentas (tools)
 - ✓ Ferramentas da plataforma Android
 - ✓ Última plataforma Android disponível
 - ✓ Versão mais atualizada do emulador Android
 - ✓ <http://developer.android.com/sdk/index.html>



The screenshot shows the Android Developer website. The browser address bar displays `developer.android.com/sdk/index.html`. The navigation bar includes links for **Developers**, **Design**, **Develop** (highlighted), and **Distribute**. Below this, there are links for **Training**, **API Guides**, **Reference**, **Tools** (highlighted), **Google Services**, and **Samples**.

The main content area is titled **Get the Android SDK**. It features a sidebar on the left with a 'Developer Tools' section containing links like 'Download', 'Setting Up the ADT Bundle', 'Setting Up an Existing IDE', 'Android Studio', 'Exploring the SDK', and 'Download the NDK'. The main text explains that the Android SDK provides API libraries and developer tools for building, testing, and debugging apps. It recommends downloading the ADT Bundle for new developers, which includes the essential Android SDK components and a version of the Eclipse IDE with built-in ADT (Android Developer Tools).

A large 3D graphic of the Android robot made of blue blocks is shown on the right. Below it, a blue button reads **Download the SDK** with the subtitle **ADT Bundle for Windows**.

Developer Tools

- Download** ^
- Setting Up the ADT Bundle
- Setting Up an Existing IDE ^
- Android Studio ^
- Exploring the SDK
- Download the NDK

Workflow ^

Support Library ^

Tools Help ^

Revisions ^

Samples

ADK ^

Get the Android SDK

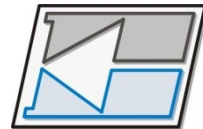
The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in **ADT (Android Developer Tools)** to streamline your Android app development.

With a single download, the ADT Bundle includes everything you need to begin developing apps:

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

Download the SDK
ADT Bundle for Windows



Instalação/Configuração ADT


- Plataforma e versão do Android
 - ✓ Variedade de modelos = Variedade de versões de Android.
 - ✓ Novos dispositivos tendem a possuir versões mais novas do Android.
 - ✓ Compatibilidade
 - Cada plataforma tem um código identificador, chamado de API level:
 - API level 1: Plataforma 1.0;
 - API level 2: Plataforma 1.1;
 - - ...
 - - API level 9: Plataforma 2.3;

Android SDK Manager

Packages Tools

SDK Path: C:\Users\Samir\AppData\Local\Android\android-sdk

Packages

 Name	API	Rev.	Status
Tools			
<input checked="" type="checkbox"/> Android SDK Tools		20.0.3	Update available: rev. 22
<input checked="" type="checkbox"/> Android SDK Platform-tools		14	Update available: rev. 16.0.2
API 19			
<input checked="" type="checkbox"/> Intel x86 Atom System Image	19	1	Not installed
<input checked="" type="checkbox"/> Google APIs	19	2	Not installed
API 18			
Android 4.2.2 (API 17)			
Android 4.1 (API 16)			
<input type="checkbox"/> Documentation for Android SDK	16	2	Installed
<input checked="" type="checkbox"/> SDK Platform	16	2	Update available: rev. 4
<input type="checkbox"/> Samples for SDK	16	1	Installed
<input checked="" type="checkbox"/> ARM EABI v7a System Image	16	2	Update available: rev. 3
<input type="checkbox"/> Intel x86 Atom System Image	16	1	Not installed
<input type="checkbox"/> Mips System Image	16	4	Not installed
<input checked="" type="checkbox"/> Google APIs	16	2	Update available: rev. 3
<input type="checkbox"/> Sources for Android SDK	16	2	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)			

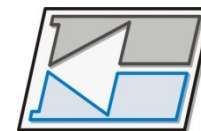
Show: ☒ Updates/New ☒ Installed ☐ Obsolete Select [New](#) or [Updates](#)

Sort by: ☒ API level ☐ Repository [Deselect All](#)

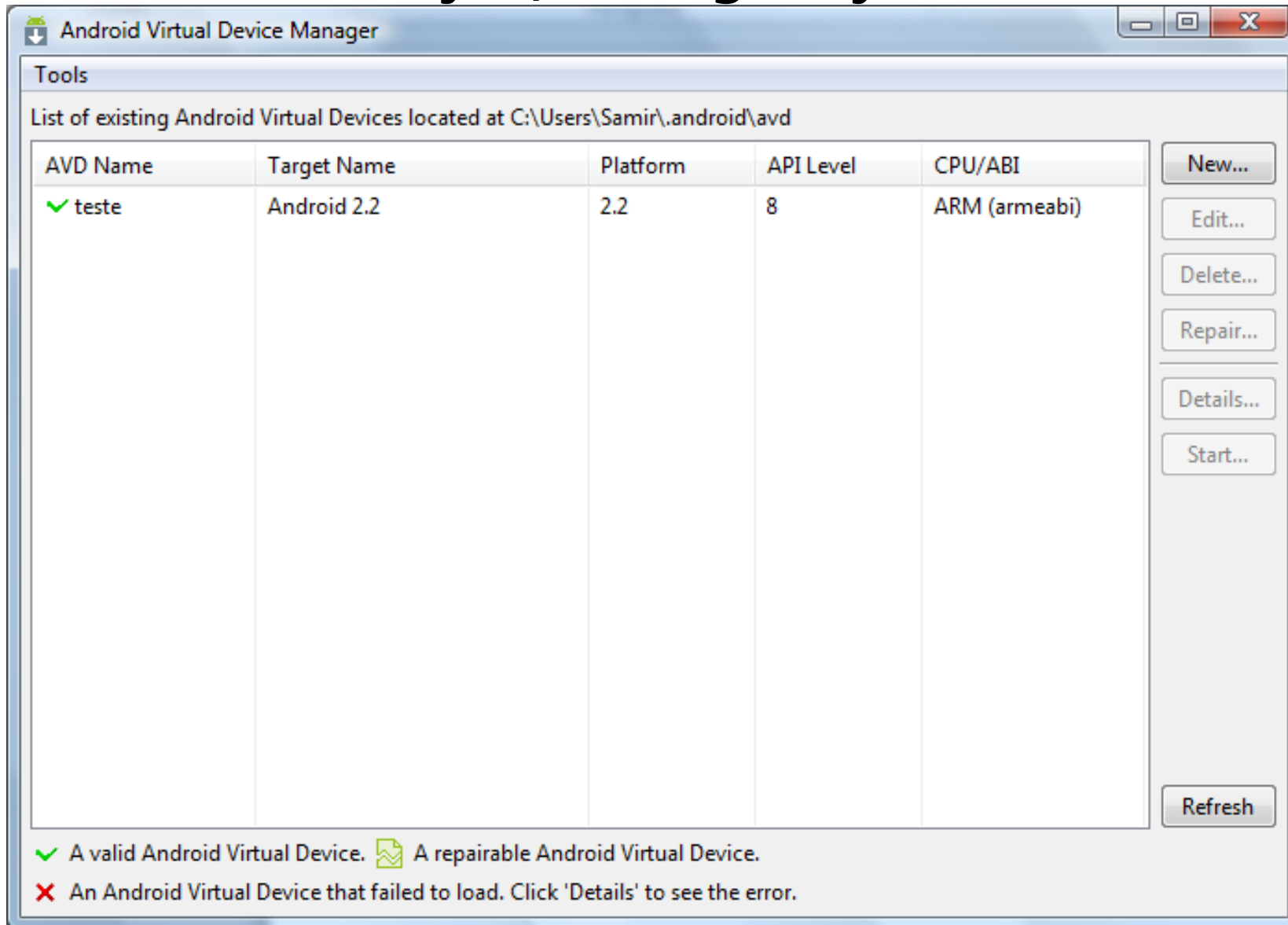
Install 9 packages...

Delete 6 packages...

Done loading packages.



Instalação/Configuração ADT



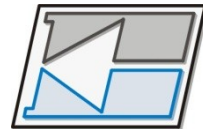
Tools

List of existing Android Virtual Devices located at C:\Users\Samir\.android\avd

AVD Name	Target Name	Platform	API Level	CPU/ABI
✓ teste	Android 2.2	2.2	8	ARM (armeabi)

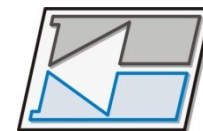
New...
Edit...
Delete...
Repair...
Details...
Start...
Refresh

✓ A valid Android Virtual Device. ⚠ A repairable Android Virtual Device.
✗ An Android Virtual Device that failed to load. Click 'Details' to see the error.



Instalação/Configuração ADT

- Criando e configurando um AVD
 - ✓ AVD = Android Virtual Device
 - ✓ Emulador que simula exatamente a configuração de um dispositivo real.
 - ✓ Possibilita a configuração de diversos parâmetros relativos ao dispositivo.
 - ✓ No momento da criação, **observar** o nível de API suportado pelo AVD.



Instalação/Configuração ADT

- Criando e configurando um AVD
✓ AVD Manager

Create new Android Virtual Device (AVD)

Name: myPhone

Target: Android 2.2 - API Level 8

CPU/ABI: ARM (armeabi)

SD Card:

☒ Size: 128 MiB

☐ File: Browse...

Snapshot:

☒ Enabled

Skin:

☒ Built-in: Default (WVGA800)

☐ Resolution: x

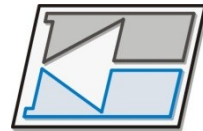
Hardware:

Property	Value	
Abstracted LCD density	240	
Max VM application hea...	24	

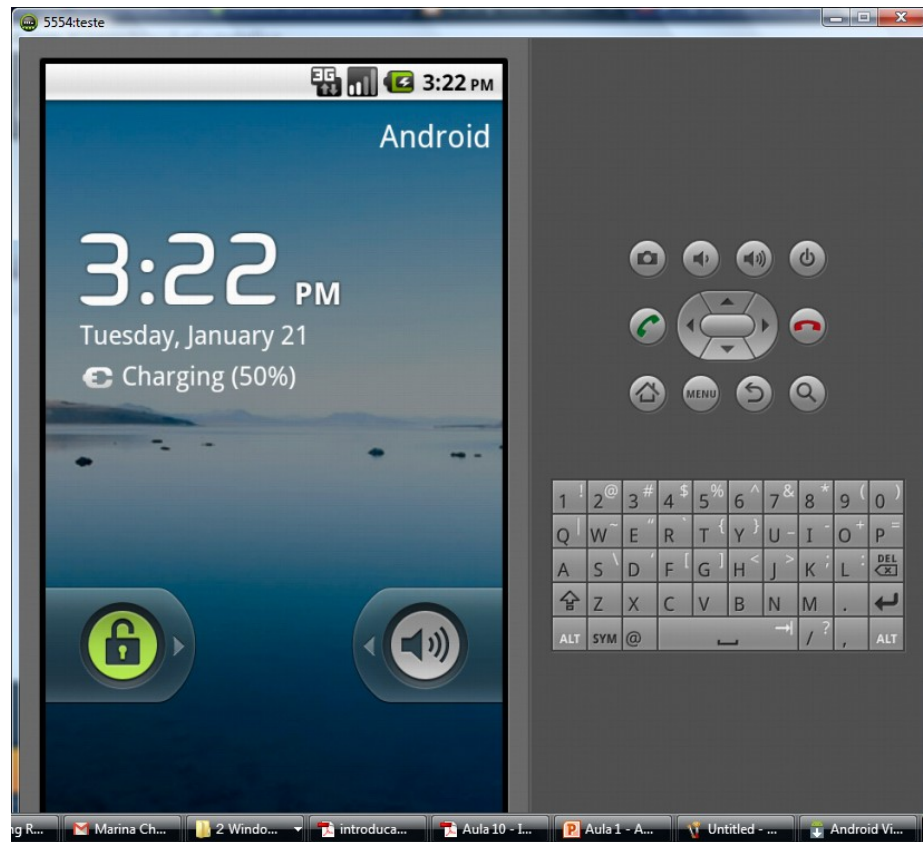
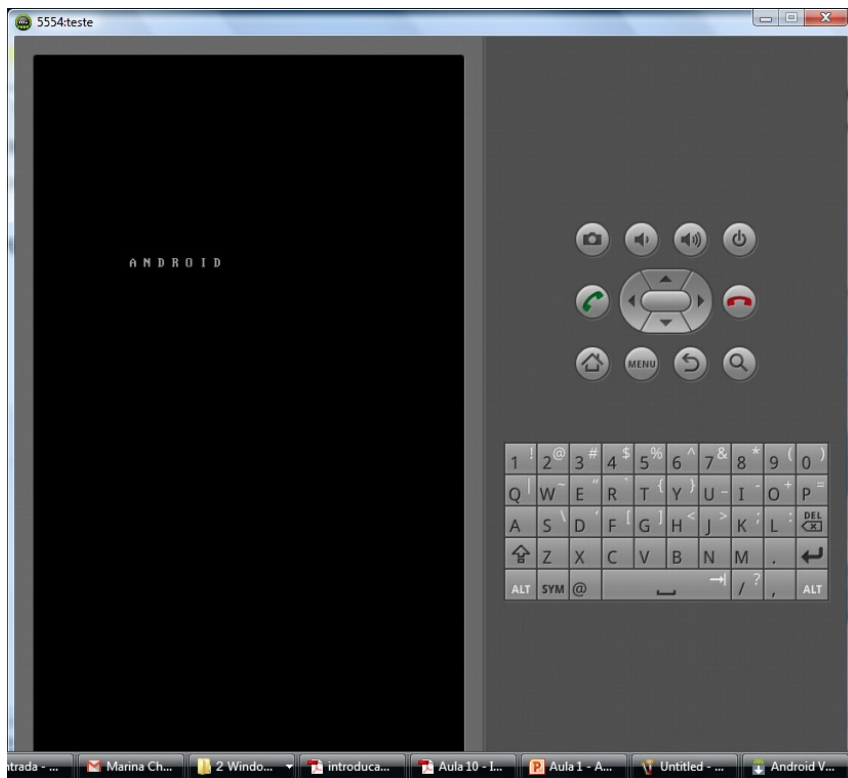
New... Delete

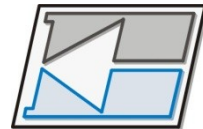
☐ Override the existing AVD with the same name

Create AVD Cancel



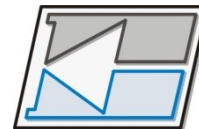
✓ AVD Manager





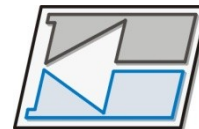
Instalação/Configuração ADT

- Aplicações Android
 - ✓ .apk = Extensão das aplicações Android.
 - ✓ Nada mais é que um arquivo compactado com todos os recursos a serem utilizados pela aplicação.

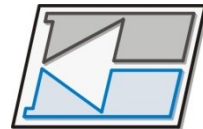


Instalação/Configuração ADT





Primeira Aplicação Android

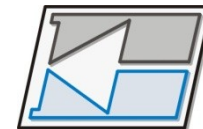


Android: Primeira Aplicação

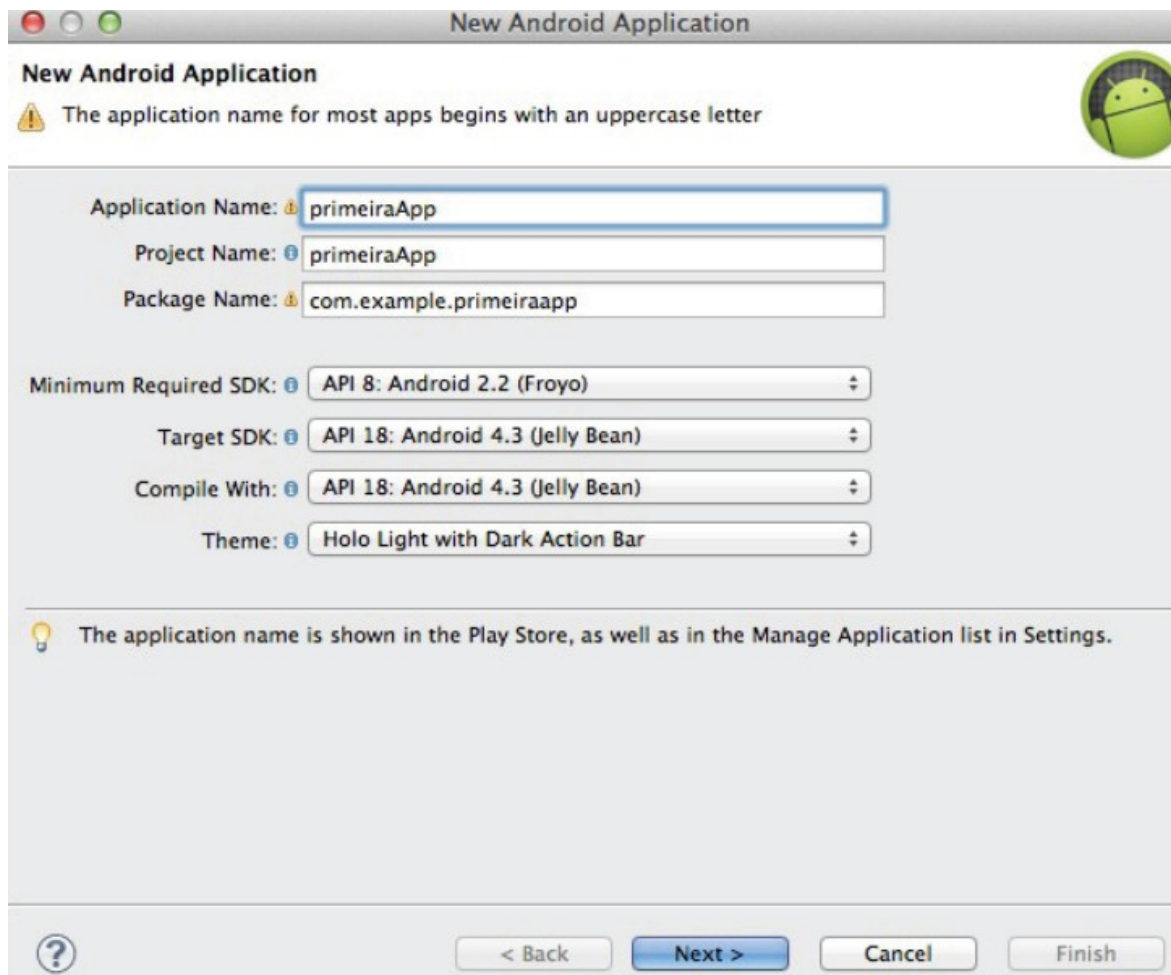
- Criar um novo projeto/app
 - ✓ File -> New -> Android Application Project.
 - ✓ Preencher:
 - . Application Name: primeiraApp
 - . Project Name: primeiraApp
 - . Package Name: com.example.primeiraapp
 - . Ajustar parâmetros da SDK.

INSTITUTO FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA

Android: Primeira Aplicação



- Criar um novo projeto/app



New Android Application

The application name for most apps begins with an uppercase letter

Application Name: primeiraApp

Project Name: primeiraApp

Package Name: com.example.primeiraapp

Minimum Required SDK: API 8: Android 2.2 (Froyo)

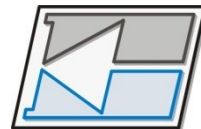
Target SDK: API 18: Android 4.3 (Jelly Bean)

Compile With: API 18: Android 4.3 (Jelly Bean)

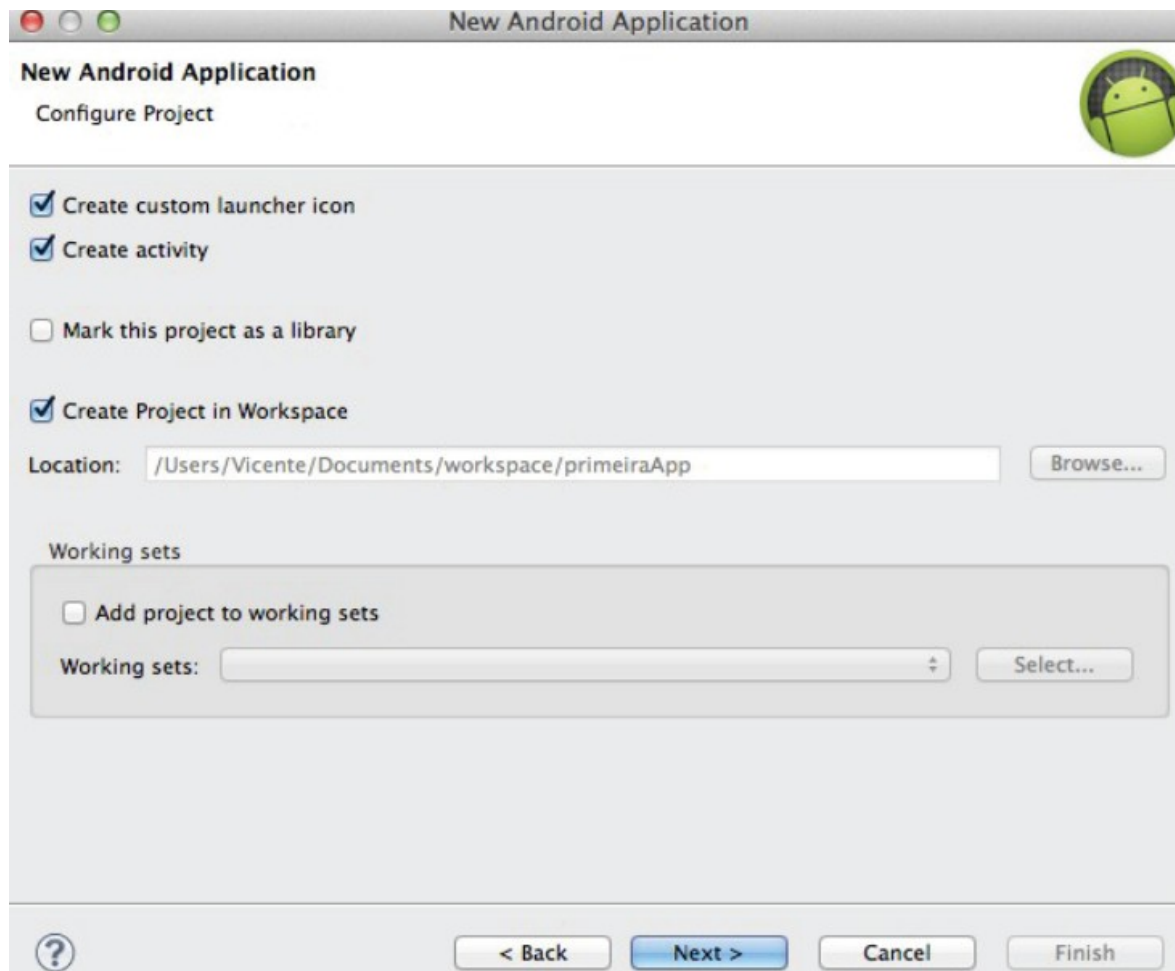
Theme: Holo Light with Dark Action Bar

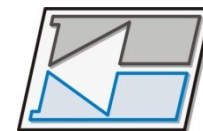
The application name is shown in the Play Store, as well as in the Manage Application list in Settings.

< Back Next > Cancel Finish



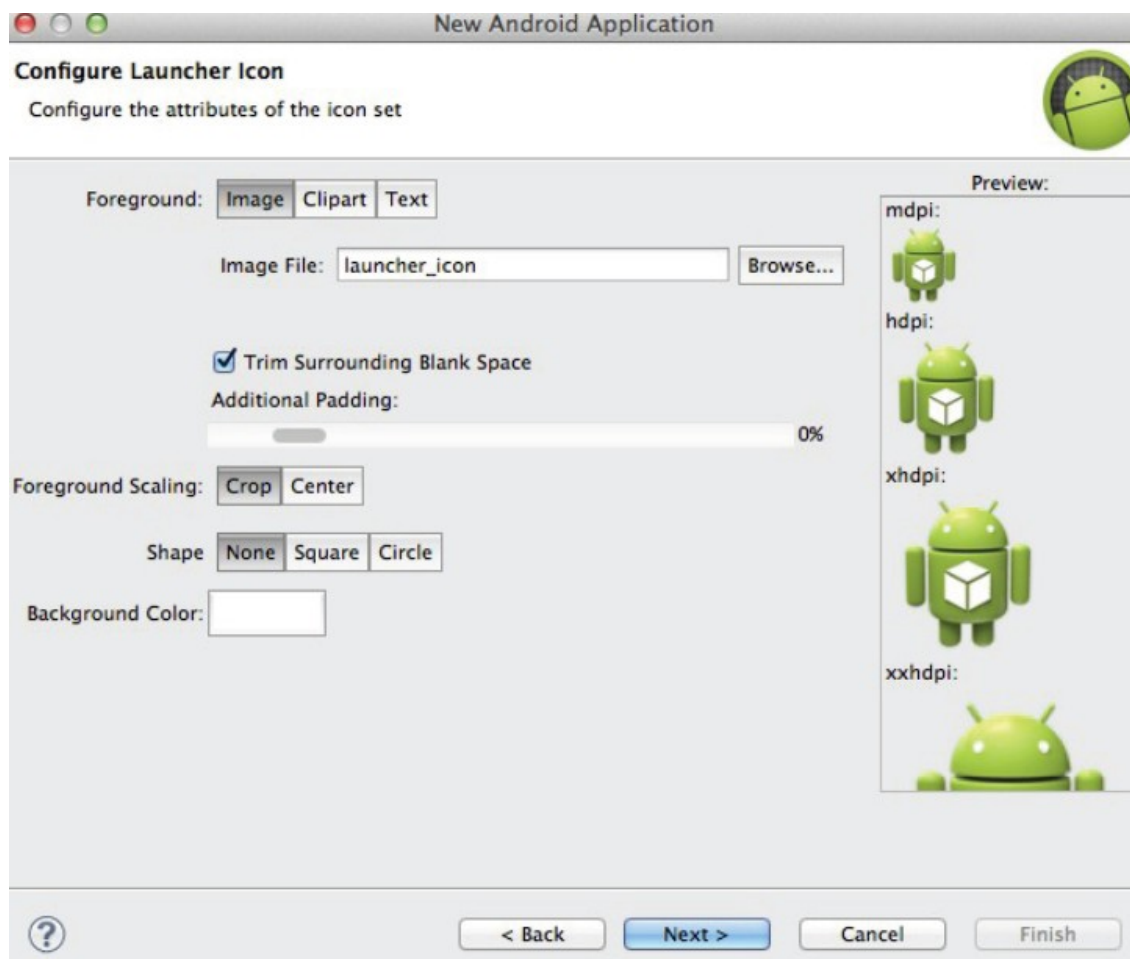
- Criar um novo projeto/app

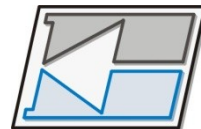




Android: Primeira Aplicação

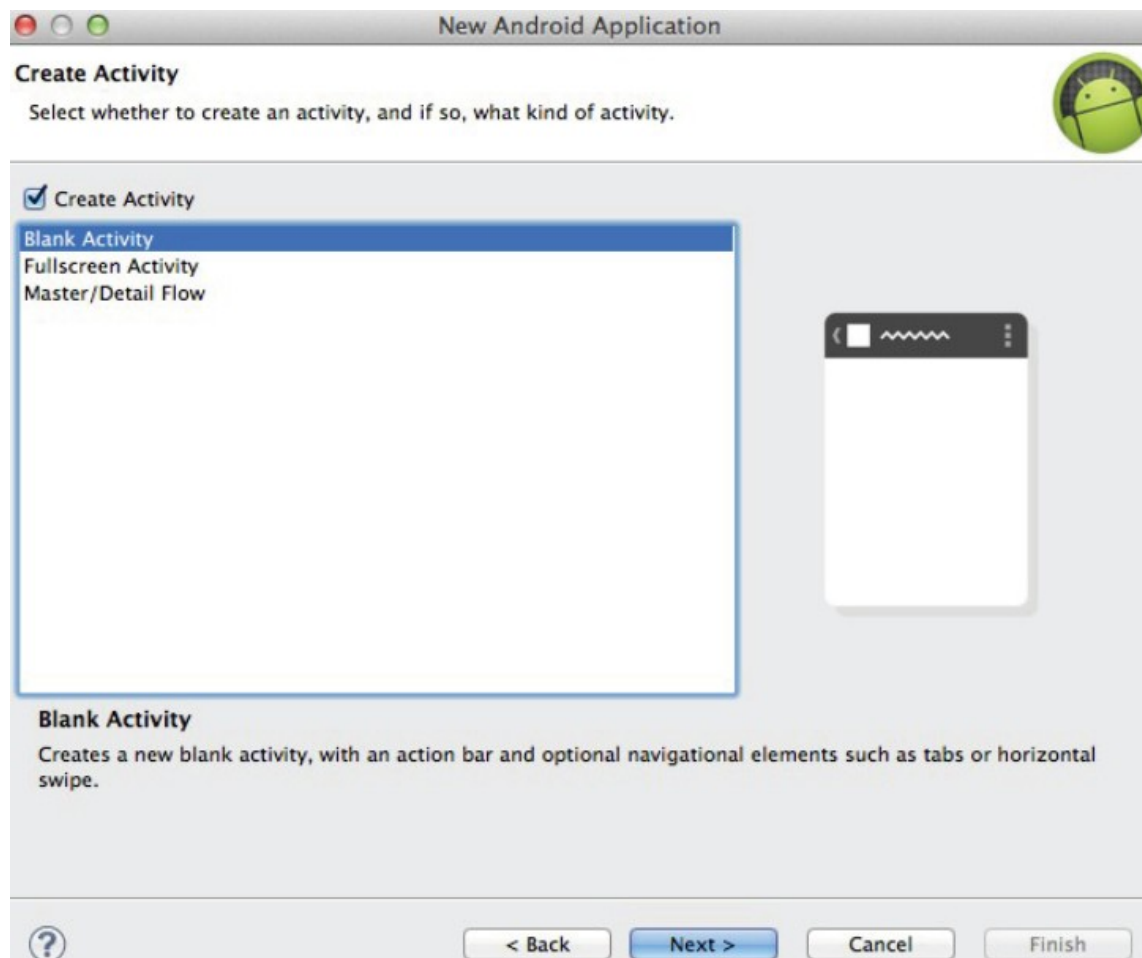
- Criar um novo projeto/app

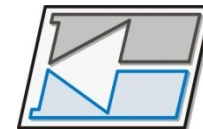




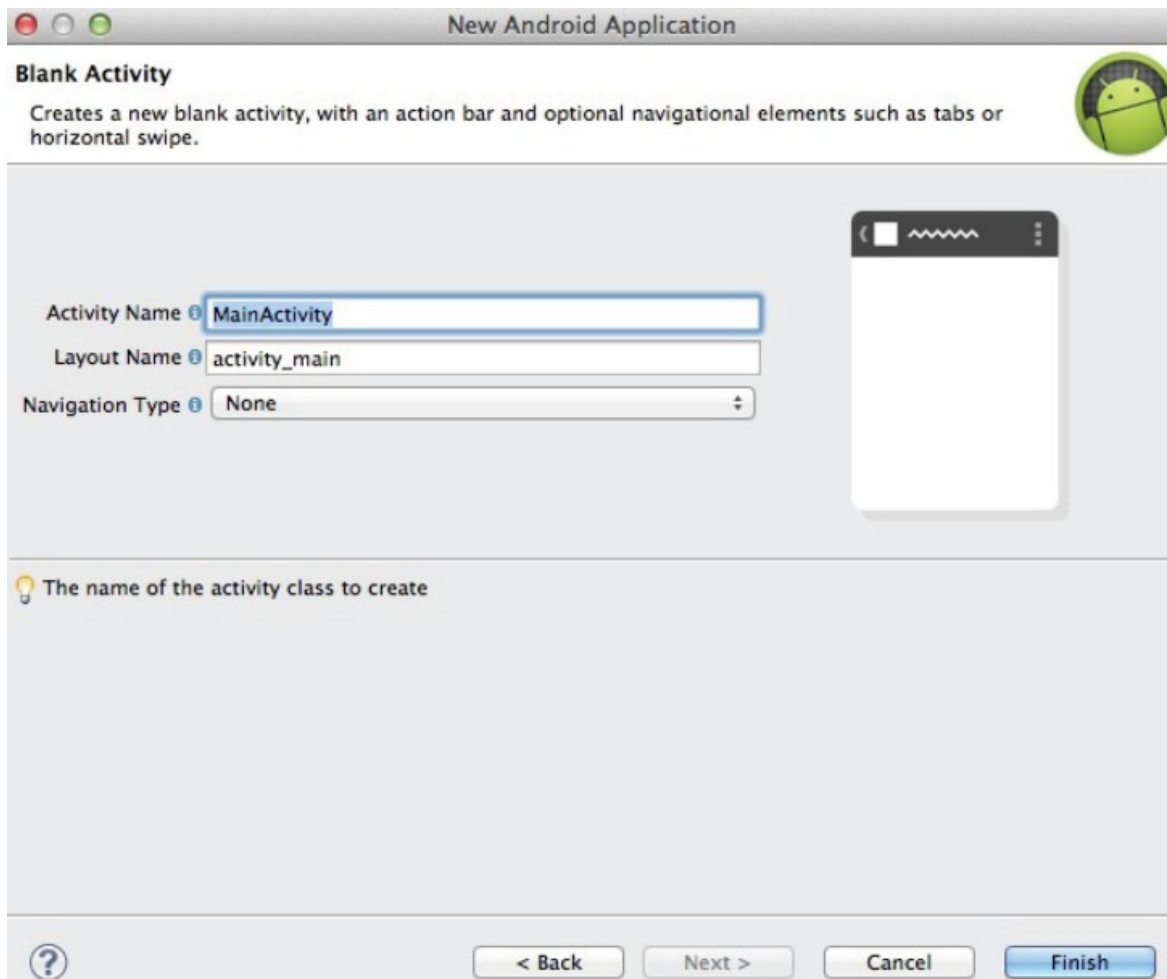
Android: Primeira Aplicação

- Criar um novo projeto/app





- Criar um novo projeto/app



New Android Application

Blank Activity

Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.

Activity Name: MainActivity

Layout Name: activity_main

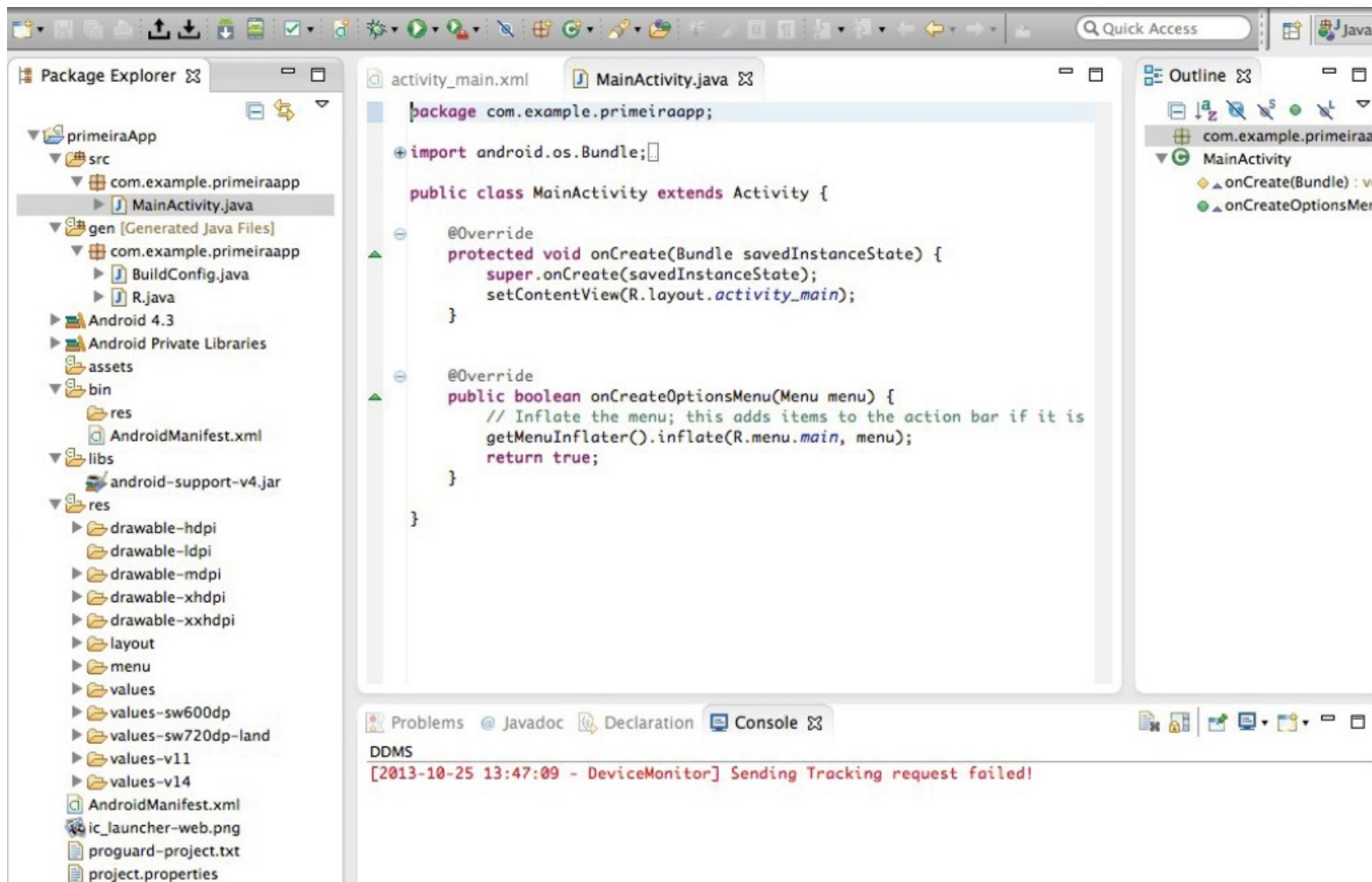
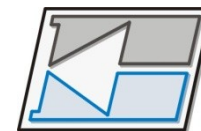
Navigation Type: None

The name of the activity class to create

< Back Next > Cancel Finish

INSTITUTO FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA

Android: Primeira Aplicação



The screenshot displays the Android Studio IDE interface. The Package Explorer on the left shows the project structure for 'primeiraApp', including the 'src' directory with 'com.example.primeiraapp' and 'MainActivity.java'. The main editor window shows the code for 'MainActivity.java', which extends 'Activity' and implements 'onCreate' and 'onCreateOptionsMenu'. The Outline on the right shows the class hierarchy. The Console at the bottom displays a message: '[2013-10-25 13:47:09 - DeviceMonitor] Sending Tracking request failed!'.

```
package com.example.primeiraapp;

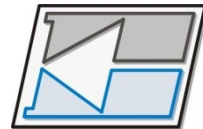
import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

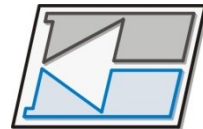
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

DDMS
[2013-10-25 13:47:09 - DeviceMonitor] Sending Tracking request failed!



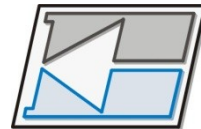
Android: Primeira Aplicação

- Estrutura do projeto
 - ✓ **src**: Contém as classes em Java e a MainActivity.java.
 - ✓ **gen**: Contém classe R.java gerada automaticamente. Permite que a aplicação acesse os recursos (arquivos, imagens, etc) utilizando as constantes dessa classe.
 - ✓ **assets**: Arquivos opcionais ao projeto. Ex.: Fonte customizada.
 - ✓ **res**: Contém os recursos utilizados pela aplicação (imagens, layouts, sons, etc). Possui três diferentes sub-pastas: *drawable*, *layout* e *values*.



Android: Primeira Aplicação

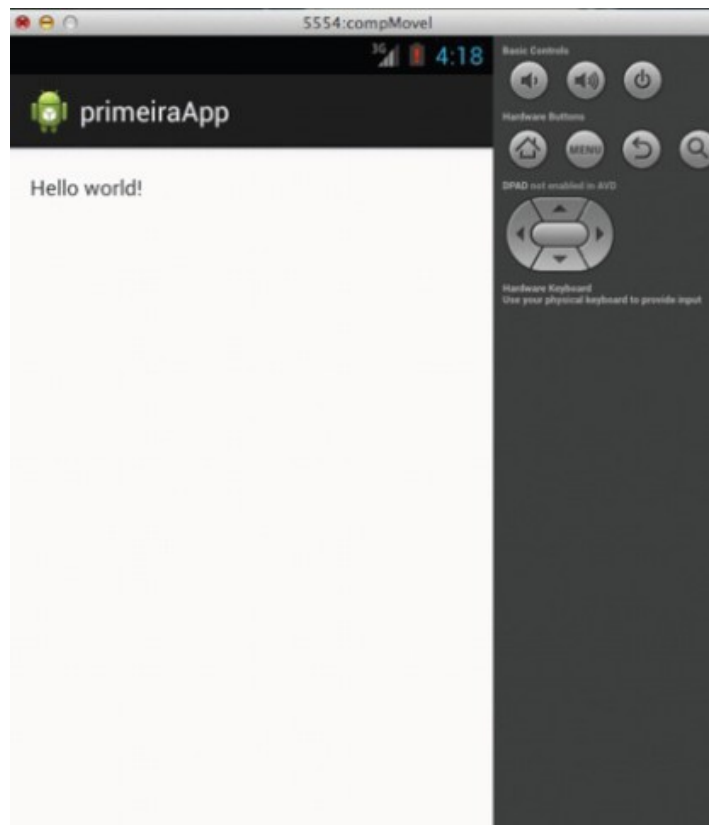
- Estrutura do projeto
 - ✓ **drawable**: Contém as imagens usadas na aplicação. É possível separar as imagens em pastas de acordo com o tamanho e resolução de cada dispositivo.
 - ✓ **layout**: Contém os arquivos XML de *layouts* para a construção das telas da aplicação.
 - ✓ **values**: Contém os arquivos XML utilizados para a internacionalização da aplicação.
 - ✓ **OBS**: Cada arquivo (imagens, audio, XML, etc) dentro da pasta *res* possui uma referência na classe R, que é gerada automaticamente.

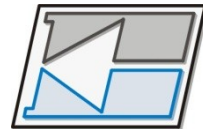


Android: Primeira Aplicação

- Execução:

✓ Run -> Run As -> Android Application.



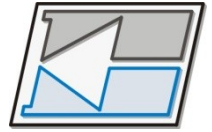


Android: Primeira Aplicação

- Arquivos Gerados no Projeto:

✓ /res/layout/activity_main.xml

- . Arquivo XML que define a interface gráfica da tela.
- . Por padrão, o arquivo contém uma tag `<TextView>` que exibe um texto de “Hello world”.
- . Atentar para o valor de `android:text="@string/hello_world"`;
- . Padrão de layout é definido pela tag `<RelativeLayout>`

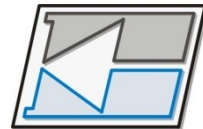


Android: Primeira Aplicação

- Arquivos Gerados no Projeto:

✓ `/res/layout/activity_main.xml`

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
</RelativeLayout>
```



Android: Primeira Aplicação

- Arquivos Gerados no Projeto:

- ✓ `/res/values/strings.xml`

- . Arquivo XML que contém as mensagens da aplicação para organizar os textos em um único arquivo centralizado.

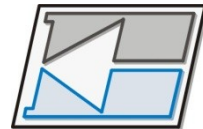
- . Facilita a internacionalização da aplicação.

- . `/res/values-en/strings.xml`

- . `/res/values-pt/strings.xml`

- ...

- . Inicialmente contém a string do hello world e o nome da aplicação (chave `app_name`).



Android: Primeira Aplicação

- Arquivos Gerados no Projeto:

✓ /res/values/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

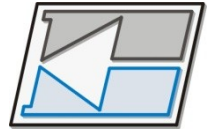
```
<resources>
```

```
    <string name="app_name">primeiraApp</string>
```

```
    <string name="action_settings">Settings</string>
```

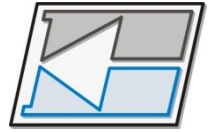
```
    <string name="hello_world">Hello world!</string>
```

```
</resources>
```



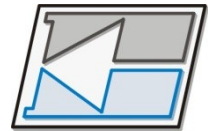
Android: Primeira Aplicação

- Arquivos Gerados no Projeto:
 - ✓ `/gen/R.java`
- Contém constantes para facilitar o acesso aos recursos do projeto (XMLs de layout, strings, etc).
- Classe é gerada automaticamente pelo ADT a cada vez que um novo recurso é adicionado ao projeto.
- **Importante:** Nunca alterar manualmente a classe R.java.



Android: Primeira Aplicação

- Arquivos Gerados no Projeto:
 - ✓ **MainActivity.java**
 - **Classe principal** do projeto. Representa a tela inicial da aplicação.
 - Toda e qualquer *activity* deve estender da classe `android.app.Activity`.
 - `android.app.Activity` representa uma tela da aplicação. É responsável por controlar o estado e os eventos da tela.
 - `android.app.Activity` apenas representa a tela, não sendo responsável pelo desenho de qualquer elemento.



Android: Primeira Aplicação

- Arquivos Gerados no Projeto:

- ✓ MainActivity.java

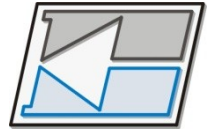
```
package com.example.primeiraapp;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends Activity {

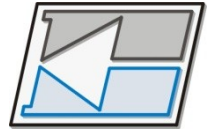
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```



Android: Primeira Aplicação

- Arquivos Gerados no Projeto:
 - ✓ **MainActivity.java**
- Método **protected void onCreate(Bundle savedInstanceState)** precisa ser implementado obrigatoriamente.
- **protected void onCreate(Bundle savedInstanceState)** é chamado automaticamente pelo Android quando a tela é criada.
- Elementos gráficos são desenhados na tela através da classe View: TextView, ImageView, etc.
- Criação da tela (e dos elementos gráficos) pode se dar através de arquivos XML ou “programaticamente”.



Android: Primeira Aplicação

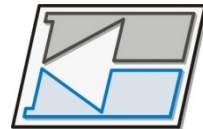
- Arquivos Gerados no Projeto:

- ✓ **MainActivity.java**

Dentro da classe `android.app.Activity` existem duas assinaturas para o método `setContentView()`.

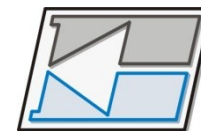
- 1ª: ID indicando algum recurso de algum arquivo XML.
- 2ª: Argumento do tipo `View`.

- `setContentView()` é o método que faz a ligação entre a *Activity* e a *View* (responsável por desenhar a interface gráfica na tela).
- **Importante:** Verificar o `setContentView()` presente em `onCreate()` da `MainActivity.java`.



Android: Primeira Aplicação

- Arquivos Gerados no Projeto:
 - ✓ **AndroidManifest.xml**
 - Principal arquivo do projeto.
 - Dentre as diversas informações, vemos que a *MainActivity* foi configurada como a principal do projeto:
 - Descreve a versão mínima e *target* da SDK.
 - Ícone, versão da aplicação, nome da aplicação, temas, ...

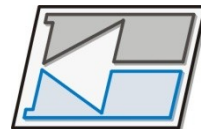


Android: Primeira Aplicação

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.primeiraapp"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="18" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.primeiraapp.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Android: Primeira Aplicação

- Arquivos Gerados no Projeto:

✓ **AndroidManifest.xml**

- Notar uso das referências a itens visuais.

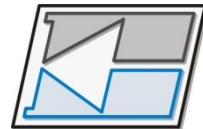
...

`android:icon="@drawable/ic_launcher"`

...

- A notação com o caractere '@' é utilizada sempre que for necessário acessar um recurso dentro de um XML.
- Se necessário fazer uso de um recurso à partir de uma classe Java, então classe R deve ser utilizada.

Objetivo	XML	Classe R
Acessar icon.png localizada na pasta drawable.	@drawable/icon	R.drawable.icon
Acessar msg hello localizada no arquivo strings.xml	@string/hello	R.string.hello

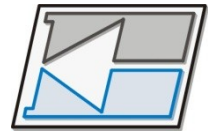


Android: Primeira Aplicação

- Listando aplicações instaladas e navegando no dispositivo.

✓ **ADB (Android Debug Bridge):** Interface de debug do Android.

- `$ adb devices`: Listagem dos dispositivos conectados no computador.
- `$ adb shell`: Acesso à shell do dispositivo.
- `$ adb shell ls`: Listagem do sistema dos arquivos presentes na raiz do root filesystem.
- `. $ adb shell ls /data/app`: Listagem das apps instaladas no sistema.



Instalação/Configuração ADT

- Instalando aplicações Android no dispositivo

✓ Acessar:

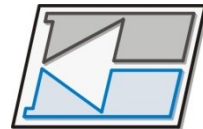
[https://code.google.com/p/zxing/downloads/detail?](https://code.google.com/p/zxing/downloads/detail?name=BarcodeScanner4.31.apk)

[name=BarcodeScanner4.31.apk](https://code.google.com/p/zxing/downloads/detail?name=BarcodeScanner4.31.apk)✓

Baixar:

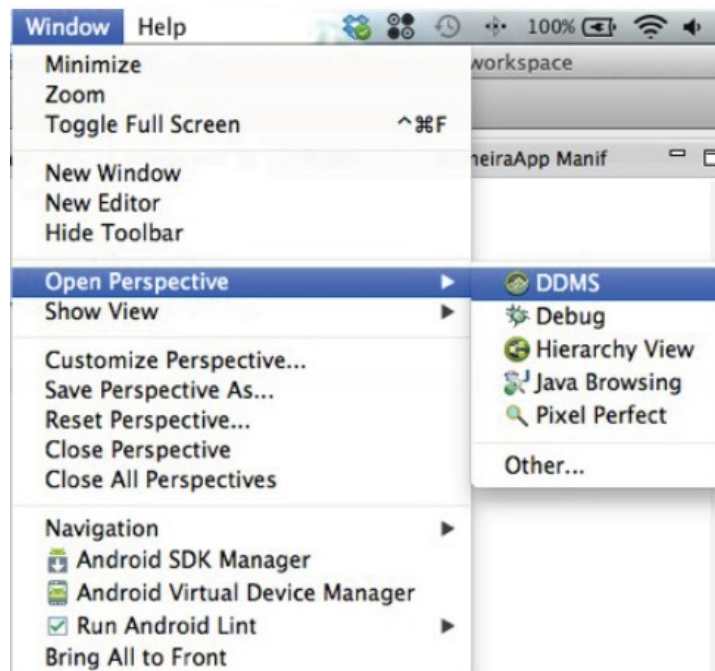
BarcodeScanner4.31.apk

✓ C:\<adt_dir>\sdk\platform-tools\adb install BarcodeScanner4.31.apk



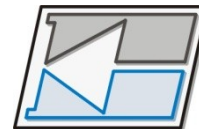
Android: Primeira Aplicação

- Perspectiva DDMS do ADT:
 - ✓ **DDMS** (*Dalvik Debug Monitor Service*): Provê controle do dispositivo conectado.
 - ✓ Acesso a informações úteis para o *debug* de sistemas.



INSTITUTO FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA

Android: Primeira Aplicação



DDMS - Eclipse

File Edit Refactor Navigate Search Project Run Window Help

Quick Access

Java DDMS

Devices

Name	State	Process	Port
teste [emulator-5554]	Online	teste [2.2,...	
android.process.m		170	8600
com.android.emai		203	8601
com.android.musi		229	8602
jp.co.omronsoft.o		108	8603
com.android.launc		119	8604
com.android.quick		222	8605
com.android.mms		184	8606
com.android.phon		116	8607
com.android.alarm		156	8608
com.android.proti		213	8609
system_process		58	8610
android.process.ac		149	8611
com.android.settir		121	8612

Threads Heap Allocation Tracker Network Statistics File Explorer Emulator Control

Telephony Status

Voice: home Speed: Full

Data: home Latency: None

Telephony Actions

Incoming number:

☐ Voice

☐ SMS

Message:

Call Hang Up

Location Controls

Manual GPX KML

☒ Decimal

☐ Sexagesimal

Longitude -122,084095

Latitude 37,422006

Send

LogCat

Saved Filters

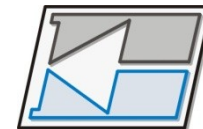
All messages (no filters)

Search for messages. Accepts Java regexes. Prefix with pid, app, tag; or text: to limit scope.

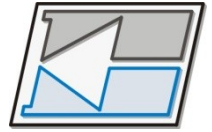
verbose

L...	Time	PID	TID	Application	Tag	Text
------	------	-----	-----	-------------	-----	------

DDMS - Eclipse

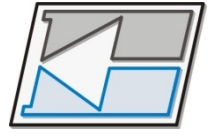


Activity e View



Activity e View

- Activity
 - ✓ Representa uma tela da aplicação.
 - ✓ A Tela é composta por vários elementos visuais, no Android representados por objetos da classe `android.view.View`.
 - ✓ *Activity* define a tela, controla sua máquina de estados, a passagem de parâmetros de uma tela para outra, etc.
 - ✓ Exibição de elementos em uma *Activity* : *View*.



Activity e View

- View

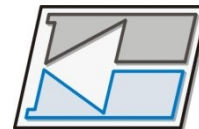
- ✓ Pode ser um componente gráfico simples (botão, checkbox, imagem, etc).

- ✓ Ligação entre a *View* e *Activity*:

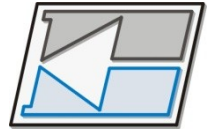
```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);}
```

- Arquivo **activity_main.xml** será utilizado como *view* principal da tela.

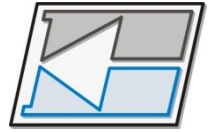


Componentes



Componentes

- Classe R
 - É criada automaticamente pelo ADT e contém ponteiros para os recursos do projeto.
 - - Sempre que um novo arquivo é adicionado a uma das pastas dos *resources*, uma nova constante é criada na classe R.
 - ✓ **Exemplo:** Uma figura icone.png é inserida na pasta drawables dos recursos do projeto.
 - ✓ Automaticamente uma constante `R.drawable.icone` é criada na classe R.
 - ✓ Analogamente, `R.layout.activity_main` acessa o arquivo de *layout* chamado `activity_main.xml`.

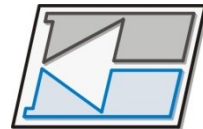


Componentes

- *AndroidManifest.xml*:

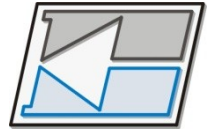
- ✓ Base da aplicação **Android**.
- ✓ **Obrigatoriamente**, cada *Activity* do projeto tem que estar declarada dentro deste arquivo.
- ✓ Pelo menos uma *Activity* deve existir no projeto - a *Activity* principal, caso algum componente visual seja incluído.

```
<activity  
  android:name="com.example.camera_hardware.MainActivity"  
    android:label="@string/app_name" >  
  <intent-filter>  
    <action android:name="android.intent.action.MAIN" />  
    <category android:name="android.intent.category.LAUNCHER" />  
  </intent-filter>  
</activity>
```



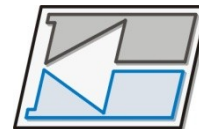
Componentes

- *AndroidManifest.xml*:
 - ✓ A tag `<intent-filter>` é utilizada para customizar a forma como a *activity* será iniciada.
 - ✓ Ação **MAIN** significa que a *Activity* pode ser iniciada isoladamente, como ponto inicial da aplicação.
 - ✓ A categoria **LAUNCHER** indica que a *Activity* estará disponível para o usuário na tela inicial junto a outras apps de usuário.

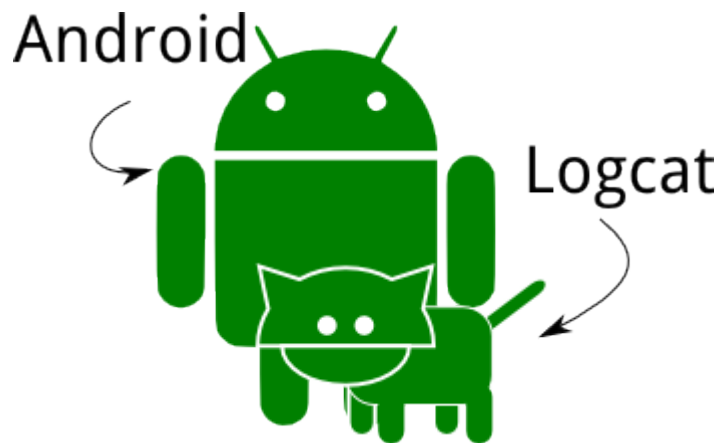


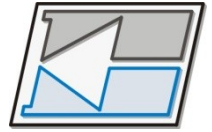
Exercícios

- a) Criar um novo layout xml
- b) Passar o texto do *android: text* para o arquivo `/res/values/strings.xml`.
- c) Alterar a função void `onCreate(Bundle)` para fazer referência a esse novo *layout*.



Logcat: o salvador

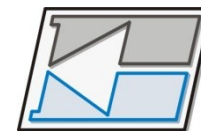




Logcat

- Introdução

- ✓ Em Java, para imprimir uma mensagem na tela faz-se uso do comando `System.out.println()`.
- ✓ Por que no Android tal comando não funciona?
 - Na máquina virtual Dalvik o `stdout` e `stderr` redirecionam sua saída para a pasta `/dev/null`.
- ✓ No Android utiliza-se a classe `android.util.Log` para criar logs de informação, debug, alertas e erros.
- ✓ Implementação e suporte a diversas categorias de logs.
- ✓ Logs gerados com a classe `android.util.Log` são direcionados para a ferramenta LogCat: **Window -> Show View -> Other->Logcat**



Problems Javadoc Declaration Console Emulator Control LogCat Lint Warnings

Search for messages. Accepts Java regexes. Prefix with pid; app; tag; or text; to limit scope. verbose Save Print Copy Download

Saved Filters + - 📄
All messages (no filters)

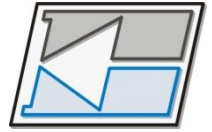
Level	Time	PID	TID	Application	Tag	Text
D	01-31 14:22:1...	2026	2051		BatteryService	update start
D	01-31 14:22:1...	2026	2051		BatteryService	level:34 scale:100 status:2 perature: 348 technology: Li icon:17303270 invalid charge 165
D	01-31 14:22:1...	2026	2051		BatteryService	turn on LED for charging
E	01-31 14:22:1...	14483	14483		MtpService	In MTPAPP onReceive:android.
E	01-31 14:22:1...	14483	14483		MtpService	battPlugged Type : 2
D	01-31 14:22:1...	2127	2127		STATUSBAR-BatteryContro...	onReceive() - ACTION_BATTERY
D	01-31 14:22:1...	2127	2127		STATUSBAR-BatteryContro...	onReceive() - level:34
D	01-31 14:22:1...	2127	2127		STATUSBAR-BatteryContro...	onReceive() - plugged:2
D	01-31 14:22:1...	2127	2127		STATUSBAR-BatteryContro...	onReceive() - BATTERY_STATUS
D	01-31 14:22:1...	2127	2127		STATUSBAR-PhoneStatusBar	ACTION_BATTERY_CHANGED
D	01-31 14:22:1...	2127	2127		STATUSBAR-PhoneStatusBar	NORMAL_BATTERY
E	01-31 14:22:1...	2026	2173		Watchdog	!@Sync 3254

Problems Javadoc Declaration Console LogCat

Search for messages. Accepts Java regexes. Prefix with pid; app; tag; or text; to limit scope.

Saved Filters + - 📄
All messages (no filters) (1803)
com.example.appnum5 (Session F)
com.example.appnum6 (Session F)

Level	Time	PID	TID	Application	Tag	Text
V	11-01 14:13:44.121	20164	20164	com.example.appnum6	COMP_MOVE1	log verboso
D	11-01 14:13:44.121	20164	20164	com.example.appnum6	COMP_MOVE1	log de debug
I	11-01 14:13:44.121	20164	20164	com.example.appnum6	COMP_MOVE1	log de informacao
W	11-01 14:13:44.121	20164	20164	com.example.appnum6	COMP_MOVE1	log de warning
E	11-01 14:13:44.121	20164	20164	com.example.appnum6	COMP_MOVE1	log de erro



Logcat

• Exercícios

- Crie uma aplicação e coloque ao longo do código (activity principal) os Logs abaixo.
- Inicie o emulador, execute o aplicativo e veja os Logs criados utilizando o Logcat:

//Verbose

```
Log.v(CATEGORIA, "log verbose");
```

//Debug

```
Log.d(CATEGORIA, "log de debug");
```

//Info

```
Log.i(CATEGORIA, "log de informacao");
```

//Warning

```
Log.w(CATEGORIA, "log de warning");
```

//Erro

```
Log.e(CATEGORIA, "log de erro");
```