Programa de Dispositivos Móveis

Aula 01

A Plataforma Android

O Android é um conjunto de softwares voltado para dispositivos móveis, como smartphones, tablets, smartbooks ou netbooks. Hoje no mercado existem diversos sistemas operacionais diferentes para celulares e smartphones, o que causa falta de padronização e um enorme esforço ao tentar portar aplicações e utilização de recursos entre estes diferentes modelos e marcas. Android é a resposta da Google para este problema. Trata-se de um sistema operacional open-source baseado em Linux. Ele foi inicialmente desenvolvido pela Google e posteriormente pela OpenHandset Alliance (uma associação comercial composta por mais de trinta empresas de tecnologia e telefonia celular.

Com a variedade de recursos do Android, seria fácil confundi-lo com um sistema operacional desktop. Como já foi dito, o Android é um ambiente em camadas baseado em kernel Linux e que inclui algumas funções. O subsistema da UI inclui:

- Janelas
- Visualizações
- Widgets para a exibição de elementos comuns como caixas de edição, listas e listas suspensas

O Android inclui um navegador incorporável baseado em WebKit, o mesmo mecanismo navegador de software livre equipando o navegador Mobile Safari do iPhone.

O Android tem várias opções de conectividade, incluindo WiFi, Bluetooth e dados wireless através de uma conexão celular (por exemplo, GPRS, EDGE e 3G). Uma técnica popular em aplicativos Android é estabelecer um link com o Google Maps para exibir um endereço diretamente em um aplicativo. O suporte para serviços baseados em locais (como GPS) e acelerômetros também está disponível na pilha de software Android, embora nem todos os dispositivos Android sejam equipados com o hardware necessário. Existe também suporte para câmera.

O Android aborda o desafio dos gráficos com suporte integrado para gráficos em 2-D e 3-D, incluindo a biblioteca OpenGL. O armazenamento de dados do Android inclui o banco de dados SQLite de software livre popular. A Figura 1 mostra uma visualização simplificada das camadas do software Android.

Abaixo seguem as quatro características principais do Android:

- Aplicativos sem fronteiras - Os aplicativos no Android podem acessar funcionalidades essenciais de dispositivos móveis por meio de APIs padrão. Os aplicativos podem anunciar seus recursos para que outros aplicativos os usem.

- Os aplicativos são criados igualmente - Qualquer aplicativo do dispositivo móvel pode ser substituído ou estendido, mesmo componentes essenciais como o discador ou a tela inicial.

- Os aplicativos podem facilmente incorporar a web - Os aplicativos podem facilmente incorporar HTML, Java Script e páginas de estilo. Um aplicativo pode exibir conteúdo da web por meio de uma WebView.

- Os aplicativos podem ser executados em paralelo - O Android é um ambiente multitarefa completo, no qual os aplicativos podem ser executados em paralelo. Um aplicativo pode produzir avisos para chamar a atenção enquanto estiver sendo executado em segundo plano.

Os aplicativos podem ser escritos em JAVA. A maior diferença entre Android e JME (Java Micro Edition) é que o Android não é uma aplicação que roda sobre o sistema operacional do dispositivo móvel; ele é o próprio sistema operacional, então, as aplicações desenvolvidas para o Android só rodam em dispositivos equipados com o sistema operacional Android, não podendo ser instaladas/executadas em nenhuma outra plataforma. Ele permite que os desenvolvedores criem software na linguagem de programação Java. O Android executa as aplicações desenvolvidas para ele através da máquina virtual personalizada chamada Dalvik VM.

Um aplicativo Android consiste em uma ou mais das classificações a seguir:

- Atividades

    - Um aplicativo que possui uma UI visível é implementado com uma atividade. Quando um usuário seleciona um aplicativo da tela inicial ou de um ativador de aplicativo, uma atividade é iniciada.
- Serviços
    - Um serviço deve ser utilizado para qualquer aplicativo que precise persistir por um longo período de tempo, como um monitor de rede ou um aplicativo de verificação de atualização.
- Provedores de conteúdo
    - Você pode pensar em provedores de conteúdo como um servidor de banco de dados. O trabalho de um provedor de conteúdo é gerenciar o acesso aos dados que persistem, como um banco de dados SQLite. Se seu aplicativo for muito simples, você não precisa necessariamente criar um provedor de conteúdo. Se estiver construindo um aplicativo maior, ou um que disponibilize dados para várias atividades ou aplicativos, um provedor de conteúdo será o meio de você acessar seus dados.
- Receptores de transmissão
    - Um aplicativo Android pode ser ativado para processar um elemento de dados ou para responder a um evento, como o recebimento de uma mensagem de texto.

O Android Runtime e Dalvik VM

Cada aplicação no Android roda em um processo diferente, e em cada processo é criada uma instância da máquina virtual Dalvik. As classes compiladas da aplicação se convertem no bytecode (.class) e depois é transformado para o formato .dex (Dalvik Executable) que é a aplicação do Android. A Dalvik VM (Virtual Machine) interage diretamente com o kernel Linux para execução de funcionalidades subjacentes como threading e gerenciamento de memória de baixo nível, com isto o processo (aplicação) roda dentro do DVM. Diferente de uma aplicação no JVM (Java Virtual Machine), que passa pela máquina virtual, para ser executado.

O Android Development Tools (ADT)

O Android Development Tools (Ferramentas de Desenvolvimento para Android) é um plugin (extensão) para o Eclipse IDE que o deixa preparado para desenvolver aplicações para o Android de forma integrada e simplificada. Ele permite a criação e depuração de aplicações Android de maneira fácil e rápida. Algumas características básicas do plugin:

- Provê uma maneira simples de interação com o emulador, permitindo a inserção de break-points, visualização das threads e processos atuais, entre outros... Diretamente dentro do Eclipse.
- Um assistente para criação de novos projetos, que ajuda a criar rapidamente um novo projeto Android (criando e configurando de forma básica todos os arquivos necessários) e lhe deixando pronto para começar.
- Automatiza e simplifica o processo de compilação e deploy da aplicação Android.
- Provê um editor de código fonte que lhe ajuda a escrever arquivos XMLs válidos para os arquivos de configuração e recursos do Android.
- Permite a geração do pacote *APK*, que poderá ser distribuído para os usuários.

Instalando o Eclipse com ADT Plugin

O Eclipse é um IDE (Integrated Development Environment) para desenvolvimento de aplicações em Java. Ele permite que extensões (plugins) sejam adicionadas a ele, tornando-o mais completo e específico para certas tarefas. Vamos adicionar uma extensão para o Eclipse poder desenvolver aplicações e ter uma série de grandes facilidades (citadas anteriormente) para Android.

Android Virtual Devices (Dispositivos Virtuais do Android)

É um conceito criado a partir do Android 1.5; ele serve para armazenar um conjunto de características, para que o emulador possa simular exatamente a configuração de um dispositivo real. É possível criar várias configurações para testar a aplicação em várias circunstâncias diferentes, por exemplo:

Dispositivos com:
- Quantidade X de memória
- Banda de internet limitada a X kbps/s.

- Suporte a imagens 3D
- Cartão de memória limitado a X Gb
- Teclado virtual
- Gravador de Video e Áudio
- SMS
- Internet G3
- Google Maps
- Entre muitos outros...

Emulador do Android

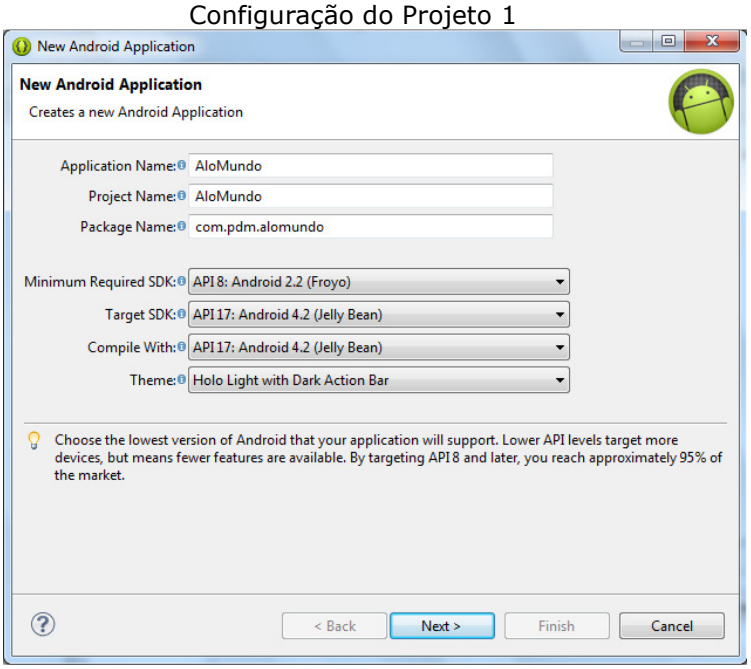
Todas as configurações podem ser combinadas, para que o teste seja preciso. Pense em AVD como rodar sua aplicação em vários dispositivos diferentes: uns mais "possantes" e outros mais "fraquinhos". Isto evita que o desenvolvedor tenha alguns problemas de configuração.
Atenção:Para executar o emulador, é obrigatório informar um AVD existente.Criando um novo AVD (Android Virtual Device)

Criando a Primeira Aplicação Android (AloMundo)

Chegou o momento de criarmos a nossa primeira aplicação "Alo Mundo" em Android. No Eclipse, faça o seguinte:

1. Clique no menu "File" -> "New" e escolha a opção "Other..."
2. Nas opções que surgiram, selecione "Android Application Project" e clique em "Next"

Configuração do Projeto 1



A janela "New Android Application" que surgiu serve para que você digite os dados do projeto que será criado. Preencha-a da seguinte forma:

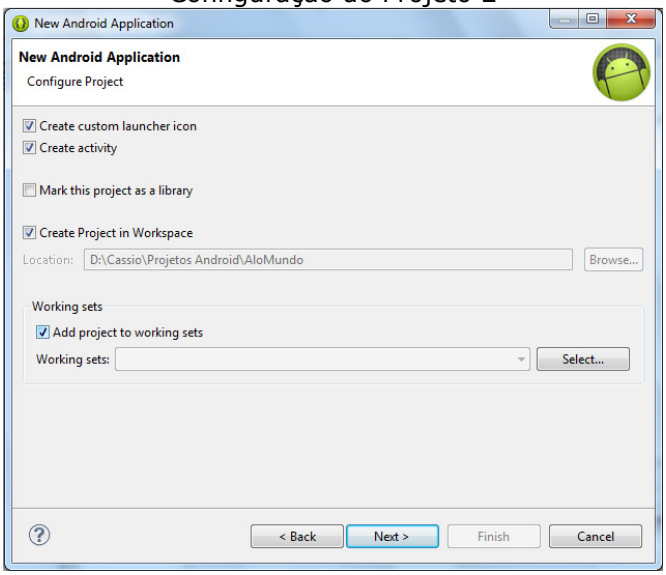
- Application name: AloMundo

- Project Name: AloMundo
- Package name: com.pdm.olamundo
- Minimum Required SDK: API 8: Android 2.2 (Froyo)
- Target SDK: API 17: Android 4.2 (Jelly Bean)
- Compile With: API 17: Android 4.2 (Jelly Bean)
- Theme: Holo Light with Dark Action Bar
- Clique em "Next"

Resumo dos nomes ao iniciar um projeto:

- Application Name - Nome que aparecerá no menu do Android.
- Project Name - O nome do projeto que aparecerá no Package Explorer do Eclipse.
- Package Name - Nome único do pacote que identifica a aplicação.
- Minimum Riquired SDK - Versão mínima do Android necessária para funcionar a sua aplicação
- Target SDK – Versão do Android alvo da sua aplicação
- Compile With – Versão do Android que será utilizada para executar a aplicação
- Theme – Tema da aplicação

Configuração do Projeto 2



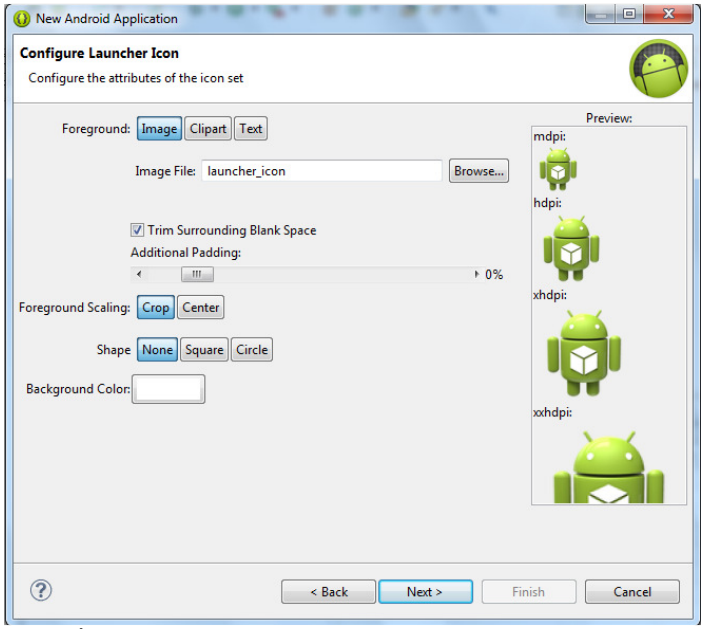
Marcar as opções:

- Create custom launcher icon
- Create activity
- Create Project in Workspace
- Clicar em "Next"

Resumo dos nomes:

- Create custom launcher icon – permite a criação de um ícone personalizado para a aplicação
- Create activity – permite a criação de uma Atividade Principal para a aplicação
- Mark this Project as a library – permite que a aplicação seja utilizada como uma biblioteca
- Create Project in Workspace – faz com que a aplicação seja salva no espaço de trabalho escolhido

Configuração do Ícone da aplicação



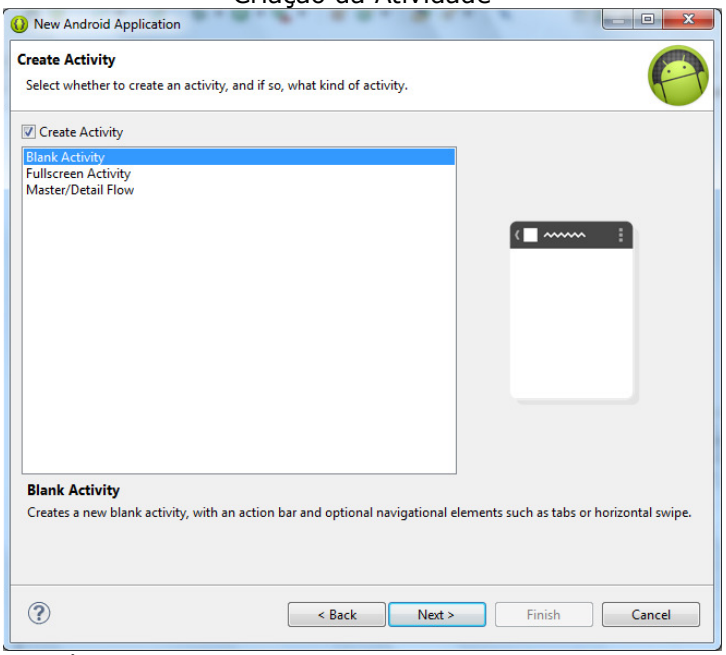
(essa tela só aparecerá se a opção "Create custom launcher icon" da tela anterior estiver marcada)

- Não mecha em nada clique direto em "Next"

Resumo dos nomes:

- Foreground – se o ícone será um(a) imagem, animação ou texto
- Foreground Scaling – se utilizara a imagem/animação/texto, puro(a) ou centralizado
- Shape – formato utilizado no ícone: nenhum, quadrado ou circular
- Background Color: cor de fundo

Criação da Atividade



(essa tela só aparecerá se a opção "Create Activity" da tela Configuração do Projeto 2 estiver marcada)

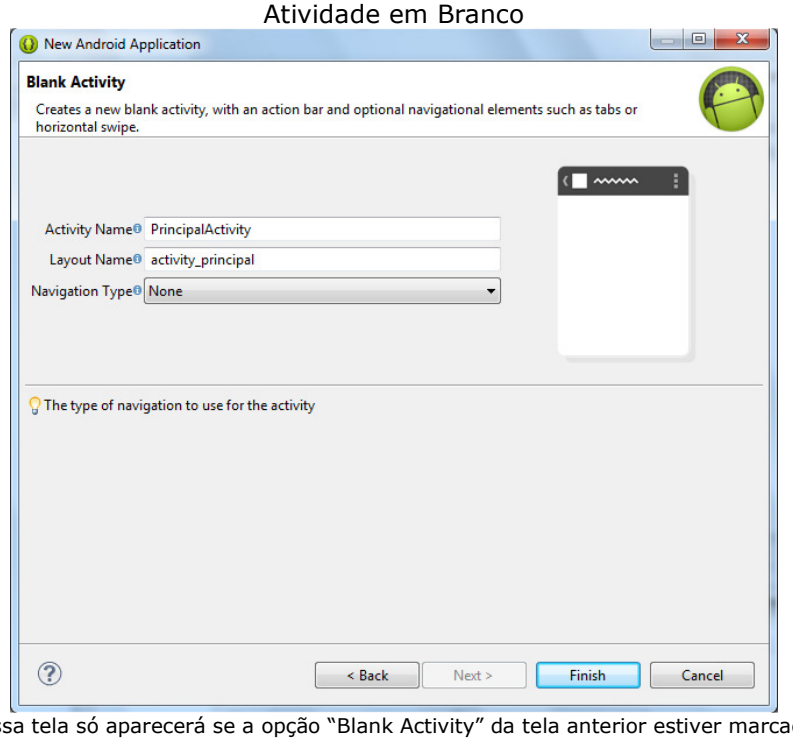
- Escolha a primeira opção, **Blank Activity**, e clique em "Next"

Resumo dos nomes:

- Blank Activity – atividade em branco, com barra de ação e elementos de navegação óptica, tais como abas(*tabs*) e "rolagem" horizontal

- Fullscreen Activity – cria uma atividade que alterna entre a barra de ação da atividade e a IU do sistema de acordo com a interação do usuário
- Master/Detail Flow - cria uma atividade dividida em duas colunas, uma contendo uma lista de objetos e outra os detalhes de cada objeto

Atividade em Branco



(essa tela só aparecerá se a opção "Blank Activity" da tela anterior estiver marcada)

Preencha a tela acima da seguinte forma:

- Activity Name – PrincipalActivity
- Layout Name – activity_principal
- Navigation Type – None
- Clique em "Finish"

Resumo dos nomes:

- Activity Name – nome da atividade
- Layout Name – nome do *layout* da atividade
- Navigation Type – tipo de navegação utilizado na atividade: lista, abas fixas e rolaveis

Diretório de Recursos

Estes são os diretórios de recursos dentro de uma aplicação Android. Todos os recursos da aplicação, tais como, textos, imagens, sons, vídeos, etc. Devem obrigatoriamente estar dentro deste diretório "*res*" e em seus respectivos sub-diretórios. Segue uma explicação detalhada de cada diretório:

Estrutura do projeto:

src
- Contém as classes Java da aplicação.Contém a classe PrincipalAloMundo que foi criada.

gen
- Contém a classe R que é gerada automaticamente e permite que a aplicação acesse qualquer recurso como arquivos e imagens.

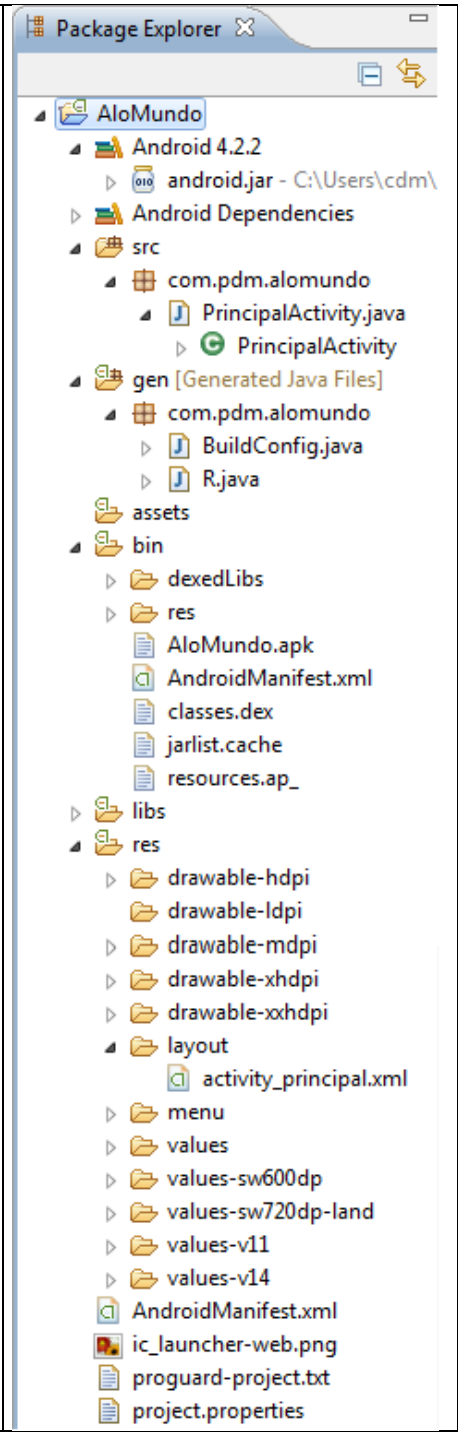
assets
- Contém arquivos opcionais ao projeto, como por exemplo, uma fonte customizada.

res
- Pasta que contém os recursos da aplicação, como imagens, layouts de tela e arquivos de internacionalização (imagens de tela, de botões, ícones etc.). Tem três subpastas: drawable, layout e values.

res/drawable
- Imagens com resoluções diferentes. Como existem diversos celulares Android com resolução de tela diferentes, é possível customizar as imagens para ficar com o tamanho exato.

res/layout
- Contém os arquivos XML utilizados para construir as tela.

res/values
- Contém todos os recursos relacionados a textos como os arquivos XML utilizados para a internacionalização da aplicação.
- Descrição de cores, estilos, etc.



Tipos de Arquivos do Android

O Android possui os seguintes formatos de arquivos:

- **.dex** - São os bytecodes das classes Java compiladas.
- **.apk** - É a aplicação Android completa, empacotada, pronta para ser instalada em um dispositivo móvel, é gerado na pasta **bin**. Semelhante a um arquivo JAR; contém todos os recursos, todos os arquivos *.dex* e todos os arquivos de configuração e identificação necessários para instalação e execução da aplicação Android em um dispositivo móvel compatível.

Estrutura Básica de uma Aplicação Android

Neste momento já temos a nossa aplicação "Alo Mundo" criada no Eclipse. Precisamos verificar a estrutura das aplicações Android para que saibamos exatamente aonde colocar cada recurso e o motivo das coisas estarem no lugar em que estão.

A Classe Activity

A classe Actvity é similar a classe JFrame do Swing representa basicamente uma tela. Quase todas as atividades interagem com o usuário, então ela se responsabiliza por criar a janela na qual você coloca a interface com o usuário. Uma tela é composta de vários elementos visuais, os quais são representados pela classe android.view.View (através do método setContentView(View)). Geralmente uma Atividade é apresentada ao usuário em fullscreen, mas nada impede que ela seja utilizada de outra maneira, como em telas flutuantes. Ao criar uma Atividade, sempre precisamos implementar dois métodos:

- onCreate(Bundle) – É aonde você inicia sua Atividade e define a UI (com setContentView(View). Assim como você pode obter os widgets (elementos de tela) que você precisará manejar, através do método findViewById(int).
- onPause() – É quando o usuário sai da sua aplicação. É neste momento que todas as informações necessárias devem ser persistidas. É importante notar que todas as Atividades devem estar definidas em AndroidManifest.xml para que seja possível inicializá-las através do método Context.startActivity().

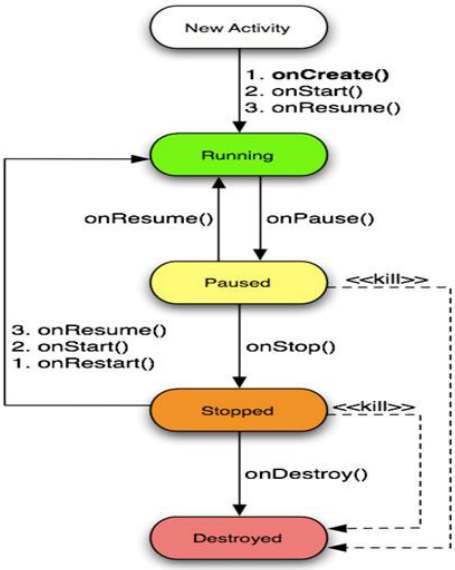
```
package com.pdm.alomundo;

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

public class PrincipalActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Um Activity tem essencialmente três estados:

1. Ele está ***active ou running***
2. Ele está ***paused*** ou
3. Ele está ***stopped***



Ciclo de Vida de um Activity (voltaremos a estudar com mais detalhes nas próximas aulas)

O Arquivo AndroidManifest.xml

Todas as aplicações Android devem ter um arquivo AndroidManifest.xml (exatamente com este nome) no seu diretório raiz. Ele armazena as informações essenciais sobre a aplicação que está sendo desenvolvida, contém as informações de configuração necessárias para ser instalado corretamente no dispositivo. Algumas informações que ele armazena são, por exemplo, os nomes de classes necessários e os tipos de eventos que o aplicativo está pronto para processar, permissões necessárias que o aplicativo precisa para execução, o nome do pacote da aplicação, componentes, atividades, serviços etc. Ele define também as informações de permissão da aplicação Android, por exemplo, acesso a internet, acesso a disco etc. Tal segurança declarativa ajuda a reduzir a probabilidade de um aplicativo perigoso causar danos em seu dispositivo.

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

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=" com.pdm.alomundo.PrincipalActivity"
            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>
```

Utilizando os Componentes Visuais do Android

O Android possui muitos componentes visuais com muitas propriedades e podem ser utilizados de várias maneiras.

Para acessar o layout visual da nossa aplicação "Alo Mundo", clique duas vezes sobre o arquivo "main.xml " dentro de " *res/layout*" (cada Activity terá o seu próprio layout); surgirá então o nosso layout dentro do Eclipse e uma aba do lado esquerdo com os componentes visuais disponíveis que podem ser inseridos no nosso layout.

res/layout/activity_principal.xml

O arquivo main.xml localizado na pasta layout define a interface gráfica da tela. Por padrão, ao criar o projeto esse arquivo contém uma tag <TextView> para exibir um texto. Essa tag define o atributo android:text="@string/hello", que utiliza uma mensagem identificada pela chave hello localizada no arquivo strings.xml. Existe também a tag <LinearLayout> que define como a tag <TextView> vai ser montada. O sinal @ é utilizado para identificar um arquivo XML.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="wrap_parent"
        android:layout_height="wrap_content"
```

```
        android:text="@string/hello_world" />


</LinearLayout>
```

res/values/strings.xml

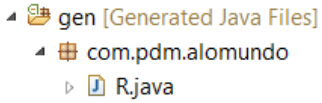
O arquivo string.xml localizado na pasta values contém a mensagens da aplicação. Por padrão, contém o nome da aplicação que digitamos ao criar o projeto e a mensagem que aparece na tela principal definida pelo arquivo main.xml.

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

    <string name="app_name">AloMundo</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>

</resources>
```

A Classe "R.java"



Esta classe é o "coração" do sistema Android. Ela representa, em forma de atributos Java, todos os recursos da sua aplicação que estão dentro dos diretórios explicados acima. Ela é gerada e atualizada automaticamente e não deve ser editada manualmente; o Eclipse fará isto automaticamente para você. Por exemplo, se tivermos dentro do diretório "res/drawable" a imagem "icon.png"; podemos acessá-la de dentro da nossa aplicação Android com a seguinte expressão:

R.drawable.icon

onde "R" é a classe, "drawable" é o diretório e "icon" é o nome do recurso. Isto serve para quaisquer recursos presentes dentro dos diretórios de recursos.

Fazendo Deploy da Aplicação "AloMundo" para o Emulador

Agora que já conhecemos a estrutura de uma aplicação Android, vamos voltar a nossa aplicação "Alô Mundo".

Primeiramente, vamos alterar a perspectiva do Eclipse para "Java EE"; para isto, clique em "Window" -> "Open Perspective" -> "Other...", selecione a opção " Java EE (default) " e clique em "Ok".

Para compilarmos a nossa aplicação e fazermos o deploy dela para o emulador, faça os seguintes passos:

1. Certifique-se de que o emulador está em execução.
2. Clique com o botão esquerdo sobre **projeto** "ProjetoAloMundo" do lado esquerdo da tela.

3. Selecione a opção "Run As" e "Android Application" A aplicação será compilada, empacotada e instalada no emulador do Android.

Acompanhe o progresso da operação através do Console do Eclipse. Quando estiver terminado, a aplicação surgirá na tela do emulador.

Ao clicar o item AloMundo aparece a tela abaixo.

Na Prática

Listagem 1. PrincipalAloMundo.java

```
package com.pdm.alomundo;

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

public class PrincipalActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

O código básico é gerado diretamente a partir do assistente New Project:

- Ele faz parte de um pacote Java chamado com.pdm.alomundo.
- Ele possui duas importações:
  - Uma para a classe de atividade
  - Uma para a classe de pacote configurável
- Quando essa atividade é iniciada, o método onCreate é invocado, passando um savedInstanceState. Não se preocupe com esse pacote configurável em relação aos nossos propósitos; ele será utilizado quando uma atividade for suspensa e depois continuada.
- O método onCreate é uma substituição do método de classes de atividade de mesmo nome. Ele chama o método onCreate da superclasse.
- Uma chamada para setContentView() associa o layout da UI definido no arquivo main.xml. Tudo que estiver no main.xml e no strings.xml é automaticamente mapeado para as constantes definidas no arquivo de origem R.java. Nunca edite esse arquivo diretamente, já que ele é alterado em cada compilação.

A configuração do arquivo AndroidManifest.xml para o aplicativo AloMundo é mostrada abaixo.

Listagem 2. AndroidManifest.xml para PrincipalAloMundo

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

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=" com.pdm.alomundo.PrincipalActivity"
            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>
```

Listagem 3. Layout do AloMundo (activity_principal.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="wrap_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
```

```
</LinearLayout>
```

Listagem 4. Conteúdo do strings.xml

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

    <string name="app_name">AloMundo</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>

</resources>
```

O layout contém um único TextView, que é, de fato, apenas uma parte de um texto estático; ele não é editável.