

Programação de Dispositivos Móveis

Aula 04

A Classe Activity

É obrigatório que cada activity do projeto esteja declarada no arquivo AndroidManifest.xml, caso contrário não é possível utilizá-la.

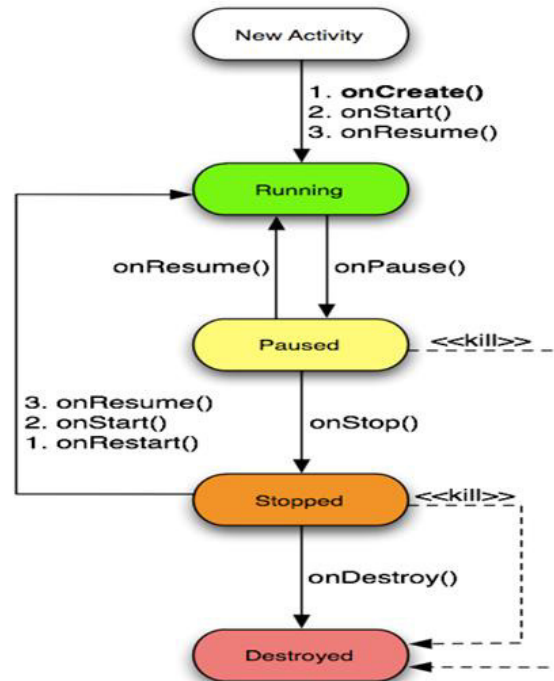
Para declarar a activity é utilizada a tag <activity> que recebe a classe principal, que é sempre relativa ao pacote principal.

A classe Activity 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) e o layout resource). 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().

Um Activity tem essencialmente três estados:

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



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:

<p>Estrutura do projeto:</p> <p>src</p> <ul style="list-style-type: none"> • Contém as classes Java da aplicação. Contém a classe MainActivity que foi criada. 	
---	--

<p>gen</p> <ul style="list-style-type: none"> • Contém a classe R que é gerada automaticamente e permite que a aplicação acesse qualquer recurso como arquivos e imagens. <p>assets</p> <ul style="list-style-type: none"> • Contém arquivos opcionais ao projeto, como por exemplo, uma fonte customizada. <p>res</p> <ul style="list-style-type: none"> • 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. <p>res/drawable</p> <ul style="list-style-type: none"> • Imagens com resoluções diferentes. Como existem diversos celulares Android com resolução de tela diferente, é possível customizar as imagens para ficar com o tamanho exato. <p>res/layout</p> <ul style="list-style-type: none"> • Contém os arquivos XML utilizados para construir as tela. <p>res/values</p> <ul style="list-style-type: none"> • 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.

Criando Aplicação Android (Aula 04)

O aplicativo que vamos desenvolver permitirá o usuário cadastrar nomes e os respectivos sexos e depois listar esses nomes cadastrados.

```
package br.pdm.clinica;
```

```
public class Medico {
    private String crm;
    private String nome;
    private String tipo;

    public Medico (String nome, String crm, String tipo) {
        this.crm = crm;
        this.nome = nome;
        this.tipo = tipo;
    }

    public void setCrm (String crm) {
        this.crm = crm;
    }

    public String getCrm () {
```

```

        return crm;
    }

    public void setNome (String nome) {
        this.nome = nome;
    }

    public String getNome () {
        return nome;
    }

    public void setTipo (String tipo) {
        this.tipo = tipo;
    }

    public String getTipo () {
        return tipo;
    }
}

```

Vamos aprender mais elementos visuais e como se faz a ligação desses elementos do layout xml (*main.xml*, *cadastro.xml* e *lista.xml*) com a linguagem de programação Java.

Vamos criar a nossa aplicação "Aula 04". No Eclipse, faça o seguinte:

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

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

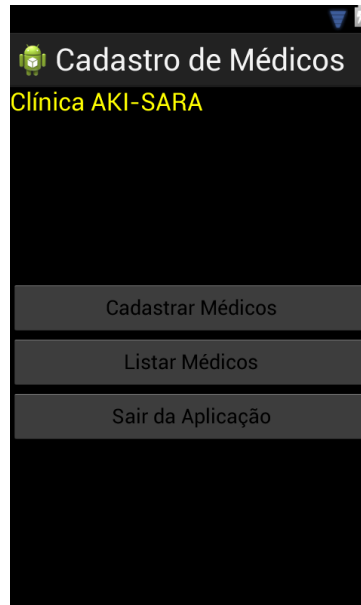
- Project Name: Aula_04
- Application name: Aula 04
- Package name: com.pdm.clinica
- Create Activity: PrincipalActivity
- Min SDK Version: 8 (*pode definir outro tipo de emulador*)
- Clique em "Finish"

Resumo dos nomes ao iniciar um projeto:

- Project Name - O nome do projeto que aparecerá no Package Explorer do Eclipse.
- Application Name - Nome que aparecerá no menu do Android.
- Package Name - Nome único do pacote que identifica a aplicação.
- Create Activity - O nome da classe que representa a tela inicial da aplicação.
- Min SDK Version - Tipo de Emulador Android

Código – Aula 04

Os códigos básicos são gerados diretamente a partir do assistente New Project. Em seguida vamos criar a tela proposta:



O arquivo main.xml, responsável pelo layout fica com o seguinte conteúdo: 1 LinearLayout vertical, 1 TextView, 1 LinearLayout vertical e 3 Button.

Listagem 1. main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#000000">

    <TextView
        android:gravity="center"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/titulo"
        android:textColor="#FFFF00"
        android:textAppearance="?android:attr/textAppearanceLarge"/>

    <LinearLayout
        android:gravity="center"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical">

        <Button
            android:id="@+id/buttonAbrirCadastro"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Cadastrar Médicos"/>

        <Button
            android:id="@+id/buttonAbrirListagem"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Listar Médicos"/>

        <Button
            android:id="@+id/buttonSair"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Sair da Aplicação"/>
    </LinearLayout>
</LinearLayout>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

O segundo `LinearLayout` é para posicionar os tres `Buttons` distribuidos no espaço vertical da tela através do atributo `android:layout_height="fill_parent"` que aparece no segundo layout. Os `buttons` ocupam toda a tela devido ao atributo `android:layout_width="fill_parent"` de cada `Button`.

O arquivo `PrincipalActivity.java` tem a classe `PrincipalActivity` que estende a classe `Activity`. Ao estender a classe `Activity` temos acesso à interação com o sistema `Android`. Podemos associar um `Activity` como sendo uma `Tela` do `Android`. Logo se tivermos várias `Telas` precisaremos de vários `Activity`'s.

```
public class PrincipalActivity extends Activity { ... }
```

A nossa classe `PrincipalActivity` vai ter como atributos, os elementos com os quais queremos interagir. Estes são o campo de texto (`EditText`) que guarda o nome e o campo `view` (`TextView`) que exibe o conteúdo do campo texto; o botão que vai preencher o campo `view` com o nome digitado. O elemento `TextView` equivale ao `TLabel` do `Swing`.

```
public static List<Medico> lista = new LinkedList<Medico>();

private Button bSair;
private Button bListar;
private Button bCadastrar;
```

Agora que temos os atributos necessários vamos interagir com o sistema `Android` para isso temos que chamar os elementos que estão no nosso layout `xml`, isto é, no `main.xml`.

Quando essa atividade é iniciada, o método `onCreate` é invocado, passando um `savedInstanceState`. O método `onCreate` é uma substituição do método de classes de atividade de mesmo nome. Ele chama o método `onCreate` da superclasse.

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    bSair = (Button) this.findViewById(R.id.buttonSair);
    bListar = (Button) this.findViewById(R.id.buttonAbrirListagem);
    bCadastrar = (Button) this.findViewById(R.id.buttonAbrirCadastro);
    bCadastrar.setOnClickListener(this);
    bListar.setOnClickListener(this);
    bSair.setOnClickListener(this);
}
}
```

Como sabemos os elementos que aparecem na tela são criados editando os seus códigos ou criando-os através da tela gráfica. Todos os elementos que são referenciados no arquivo `PrincipalActivity.java` devem ser identificados, através dos seus `id`'s criados no arquivo `main.xml`. No arquivo do código `Java` o método `findViewById(<id do elemento>)` faz uma cópia do elemento que aparece na tela para que o mesmo seja manipulado pelo programador.

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.

```
setContentView(R.layout.main);
```

Agora vamos ver quando o botão `Cadastrar` for clicado. Para isso temos de ter um `listener` do botão. Primeiro temos que adicionar o(s) botão(ões) no sistema de atendimento de eventos.

```
bCadastrar.setOnClickListener(this);
bListar.setOnClickListener(this);
```

```
bSair.setOnClickListener(this);
```

Quando um botão for acionado, por exemplo, o evento onClick, o sistema aciona o método **public void** onClick(View v). O evento onClick verifica primeiro qual foi o botão acionado e em seguida passa a efetuar o respectivo o tratamento.

```
public void onClick(View v) {
    if(v.getId()==bCadastrar.getId()) {
        Intent myIntent = new Intent(v.getContext(), Cadastro.class);
        startActivity(myIntent);
    }

    if(v.getId()==bListar.getId()) {
        Intent myIntent = new Intent(v.getContext(), Listagem.class);
        startActivity(myIntent);
    }

    if(v.getId()==bSair.getId()) {
        Toast.makeText(getApplicationContext(), "SAINDO" ,
Toast.LENGTH_SHORT).show();
        finish();
    }
}
} // fim do evento onClick
```

Listagem 2. PrincipalActivity.java

```
package com.pdm.clinica;

import java.util.LinkedList;
import java.util.List;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class PrincipalActivity extends Activity implements OnClickListener {

    public static List<Medico> lista=new LinkedList<Medico>();
    private Button btnCadastrar;
    private Button btnListar;
    private Button btnSair;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        btnCadastrar=(Button) this.findViewById(R.id.buttonAbrirCadastro);
        btnListar=(Button) this.findViewById(R.id.buttonAbrirListagem);
        btnSair=(Button) this.findViewById(R.id.buttonSair);

        btnCadastrar.setOnClickListener(this);
        btnListar.setOnClickListener(this);
        btnSair.setOnClickListener(this);
    }

    public void onClick(View v) {
        if (v.getId() == btnCadastrar.getId()) {
```

```

        Intent myIntent = new Intent(v.getContext(), Cadastro.class);
        startActivity(myIntent);
    }
    if (v.getId() == btnListar.getId()) {
        Intent myIntent = new Intent(v.getContext(), Listagem.class);
        startActivityForResult(myIntent, 0);
    }
    if (v.getId() == btnSair.getId()) {
        finish();
    }
}
}

```

Listagem 3. strings.xml

```

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

    <string name="app_name">Cadastro de Médicos</string>
    <string name="action_settings">Settings</string>
    <string name="titulo">Clínica AKI-SARA</string>

</resources>

```

Listagem 4. cadastro.xml



```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#000000" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cadastro de Médicos"
        android:textColor="#FFFF00"
        android:textAppearance="?android:attr/textAppearanceLarge"/>

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"

```

```

    android:layout_height="wrap_content"
    android:text="CRM"
    android:textColor="#FFFF00"
    android:textAppearance="?android:attr/textAppearanceLarge"/>

```

```
<EditText
```

```

    android:id="@+id/editTextCrm"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#FFFFFF">
    <requestFocus></requestFocus>

```

```
</EditText>
```

```
<TextView
```

```

    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nome"
    android:textColor="#FFFF00"
    android:textAppearance="?android:attr/textAppearance"/>

```

```
<EditText
```

```

    android:id="@+id/editTextNome"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#FFFFFF"/>

```

```
<RadioGroup
```

```

    android:id="@+id/radioGroup1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

```

```
    <RadioButton
```

```

        android:checked="true"
        android:id="@+id/radioResidente"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Residente"
        android:textColor="#FFFF00"/>

```

```
    <RadioButton
```

```

        android:id="@+id/radioContratado"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Contratado"
        android:textColor="#FFFF00"/>

```

```
</RadioGroup>
```

```
<LinearLayout
```

```

    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal"
    android:weightSum="1">

```

```
    <Button
```

```

        android:id="@+id/buttonOk"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="0.5"
        android:text="OK"/>

```

```
<Button
```



```

        android:id="@+id/buttonFecharCadastro"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_weight="0.5"
        android:text="Fechar"/>

```

```
</LinearLayout>
```

```
</LinearLayout>
```

Listagem 5. Cadastro.java

```

package br.pdm.clinica;

import br.pdm.clinica.R;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Toast;

public class Cadastro extends Activity implements OnClickListener {
    private Button btnOk, btnFechar;
    private EditText edtCrm, edtNome;
    private RadioGroup rgGrupo;

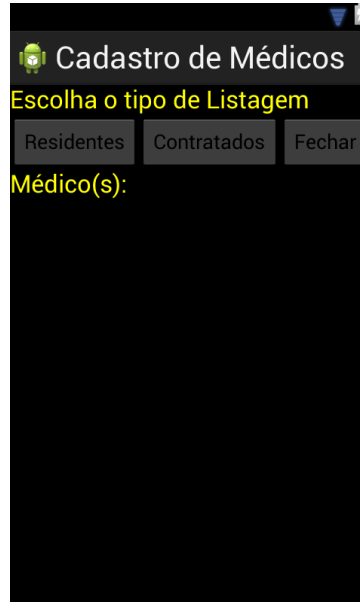
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.cadastro);
        btnOk=(Button) this.findViewById(R.id.buttonOk);
        btnFechar=(Button) this.findViewById(R.id.buttonFecharCadastro);
        edtCrm=(EditText) this.findViewById(R.id.editTextCrm);
        edtNome=(EditText) this.findViewById(R.id.editTextNome);
        rgGrupo=(RadioGroup) this.findViewById(R.id.radioGroup1);
        btnOk.setOnClickListener(this);
        btnFechar.setOnClickListener(this);
    }

    public void onClick(View v) {
        if(v.getId() == btnOk.getId()) {
            String texto;
            int radioId = rgGrupo.getCheckedRadioButtonId();
            if (radioId != -1) {
                RadioButton rbSelecao=(RadioButton)findViewById(radioId);
                texto=rbSelecao.getText().toString();
                MainActivity.Lista.add(new
Medico(edtNome.getText().toString(), edtCrm.getText().toString(), texto));
                edtNome.setText("");
                edtCrm.setText("");
                rgGrupo.clearCheck();

                Toast.makeText(getApplicationContext(),
edtNome.getText().toString() + "cadastro: " + texto, Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(getApplicationContext(), "Não cadastrado!",
Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

}

Listagem 6. listagem.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#000000" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Escolha o tipo de Listagem"
        android:textColor="#FFFF00"
        android:textAppearance="?android:attr/textAppearanceLarge"/>

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:weightSum="1">

        <Button
            android:id="@+id/buttonResidentes"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="0.3"
            android:text="Residentes"/>

        <Button
            android:id="@+id/buttonContratados"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="0.3"
            android:text="Contratados"/>

        <Button
            android:id="@+id/buttonFecharLista"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="0.3"

```

```

        android:text="Fechar"/>
    </LinearLayout>

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Médico(s):"
            android:textColor="#FFFF00"
            android:textAppearance="?android:attr/textAppearanceLarge"/>

        <TextView
            android:bufferType="normal"
            android:id="@+id/textViewTipo"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:textColor="#FFFF00"
            android:textAppearance="?android:attr/textAppearanceLarge"/>

    </LinearLayout>

    <TextView
        android:id="@+id/textViewListagem"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="#FFFF00"
        android:textAppearance="?android:attr/textAppearanceMedium"/>

</LinearLayout>

```

Listagem 7. Listagem.java

```

package br.pdm.clinica;

import br.pdm.clinica.R;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class Listagem extends Activity implements OnClickListener {
    private Button btnFechar;
    private Button btnResidentes;
    private Button btnContratados;
    private TextView txtTipo;
    private TextView txtLista;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.Listagem);
        btnFechar=(Button) this.findViewById(R.id.buttonFecharLista);
        btnResidentes=(Button) this.findViewById(R.id.buttonResidentes);
        btnContratados=(Button) this.findViewById(R.id.buttonContratados);
        txtTipo=(TextView) this.findViewById(R.id.textViewTipo);
    }
}

```

```

txtLista=(TextView) this.findViewById(R.id.textViewListagem);
btnFechar.setOnClickListener(this);
btnResidentes.setOnClickListener(this);
btnContratados.setOnClickListener(this);
}

public void onClick(View v) {
    if (v.getId() == btnFechar.getId()) {
        Toast.makeText(getApplicationContext(), "Fechando...",
Toast.LENGTH_SHORT).show();
        finish();
    }
    if (v.getId() == btnResidentes.getId()) {
        txtLista.setText(null);
        txtTipo.setText("Residentes");
        for (int i = 0; i < PrincipalActivity.lista.size(); i++) {
            Medico med = (Medico) PrincipalActivity.lista.get(i);
            if(PrincipalActivity.lista.get(i).getTipo().equals("Residente"))
{
                String reg="Nome: " + med.getNome() + " - CRM: " +
med.getCrm() + "\n";
                txtLista.append(reg);
            }
        }
    }
    if (v.getId() == btnContratados.getId()) {
        txtLista.setText(null);
        txtTipo.setText("Contratados");
        for (int i = 0; i < PrincipalActivity.lista.size(); i++) {
            Medico med = (Medico) PrincipalActivity.lista.get(i);
            if(PrincipalActivity.lista.get(i).getTipo().equals("Contratado"))
{
                String reg="Nome: " + med.getNome() + " - CRM: " +
med.getCrm() + "\n";
                txtLista.append(reg);
            }
        }
    }
}
}

```

Listagem 8. AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.pdm.clinica"
    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="br.pdm.clinica.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>
    <activity android:name="br.pdm.clinica.Cadastro" />
    <activity android:name="br.pdm.clinica.Listagem" />
</application>
```

```
</manifest>
```

Fazendo Deploy da Aplicação para o Emulador

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** "Aula_04" 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 da view **Console** do Eclipse. Quando estiver terminado, a aplicação surgirá na tela do emulador.