

Programação de Dispositivos Móveis

Aula 03

A Classe Activity

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.

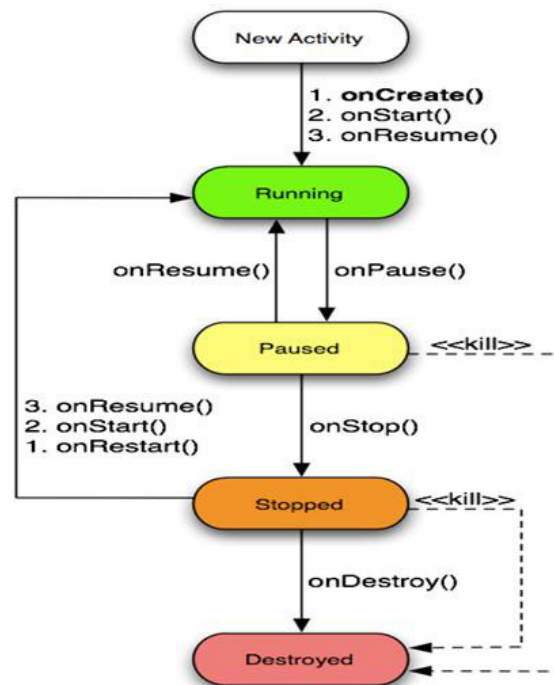
É obrigatório que cada activity do projeto esteja declarada no arquivo AndroidManifest.xml, caso contrário não é possível utilizá-la. Para declarar a Activity é utilizada a tag <activity> que recebe a classe principal, que é sempre relativa ao pacote principal.

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 subdiretórios. Segue uma explicação detalhada de cada diretório:

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

<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 telas. <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 (Aula03)

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

```
package br.pdm.aula_03.modelo;
```

```
public class Pessoa {

    private String nome;
    private String sexo;

    public Pessoa() {}

    public Pessoa(String nome, String sexo) {
        this.nome = nome;
        this.sexo = sexo;
    }

    public String getNome() {
        return nome;
    }
}
```

```

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

    public String getSexo() {
        return sexo;
    }

    public void setSexo(String sexo) {
        this.sexo = sexo;
    }
}

```

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 03". 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 03
- Application name: Aula_03
- Package name: br.pdm.aula_03
- Create Activity: PrincipalActivity
- Min SDK Version: 15 (*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 – Aula03

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 alinhamento vertical, 1 alinhamento horizontal, 1 TextView e 3 Buttons.

Listagem 1. main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#000000" >

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

    <LinearLayout
        android:id="@+id/LinearLayout1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:weightSum="1" >

        <Button
            android:id="@+id/buttonAbrirCadastro"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="0.28"
            android:text="Cadastrar" />

        <Button
            android:id="@+id/buttonAbrirListagem"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="0.28"
            android:text="Listar" />

        <Button
            android:id="@+id/buttonSair"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="0.28"
            android:text="Sair" />

    </LinearLayout>

</LinearLayout>
```

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 da 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<Pessoa> LstNomes = new LinkedList<Pessoa>();
```

```
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.buttonListar);
    bCadastrar = (Button) this.findViewById(R.id.buttonCadastrar);
    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() == 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);
        startActivity(myIntent);
    }
    if (v.getId() == btnSair.getId()) {
        Toast.makeText(getApplicationContext(), "SAINDO",
Toast.LENGTH_SHORT).show();
        finish();
    }
}
```

Listagem 2. PrincipalActivity.java

```

package br.pdm.aula_03;

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

import br.pdm.aula_03.modelo.Pessoa;

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

public class PrincipalActivity extends Activity implements OnClickListener{

    public static List<Pessoa> listaPessoas = new LinkedList<Pessoa>();

    private Button btnSair;
    private Button btnListar;
    private Button btnCadastrar;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        btnSair = (Button) this.findViewById(R.id.buttonSair);
        btnListar = (Button) this.findViewById(R.id.buttonAbrirListagem);
        btnCadastrar = (Button) this.findViewById(R.id.buttonAbrirCadastro);

        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);
            startActivity(myIntent);
        }
        if (v.getId() == btnSair.getId()) {
            Toast.makeText(getApplicationContext(), "Saindo...",
Toast.LENGTH_SHORT).show();
            finish();
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.principal, menu);
        return true;
    }
}

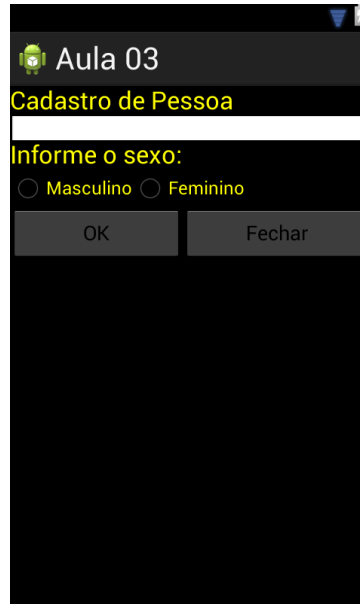
```

Listagem 3. strings.xml

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

    <string name="titulo">Agenda Pessoal</string>
    <string name="app_name">Aula 03</string>
    <string name="action_settings">Settings</string>

</resources>
```

Listagem 4. cadastro.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

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

    <EditText
        android:id="@+id/editTextNome"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
        <requestFocus></requestFocus>
    </EditText>

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Informe o sexo: "
        android:textAppearance="?android:attr/textAppearanceLarge">
    </TextView>

    <RadioGroup
        android:id="@+id/radioGroupGrupo"
        android:layout_width="wrap_content"
```

```

    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <RadioButton
        android:id="@+id/radioMasculino"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Masculino">
    </RadioButton>

    <RadioButton
        android:id="@+id/radioFeminino"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Feminino">
    </RadioButton>

</RadioGroup>

<LinearLayout
    android:id="@+id/LinearLayout1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:weightSum="1">

    <Button
        android:id="@+id/buttonCadastrarPessoa"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="0.50"
        android:text="OK">
    </Button>

    <Button
        android:id="@+id/buttonFecharCadastro"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="0.50"
        android:text="Fechar">
    </Button>

</LinearLayout>

</LinearLayout>

```

Listagem 5. Cadastro.java

```

package br.pdm.aula_03;

import br.pdm.aula_03.modelo.Pessoa;
import android.os.Bundle;
import android.app.Activity;
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 {

    EditText edtNome;
    RadioGroup rgGrupo;

```



```

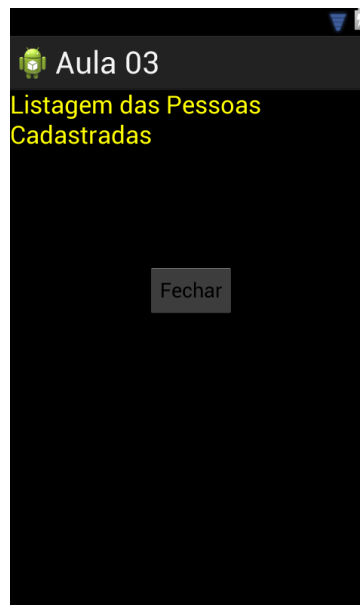
RadioButton rbMasculino, rbFeminino;
Button btnCadastrar, btnFechar;

@Override
public void onCreate (Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.cadastro);
    edtNome = (EditText) findViewById(R.id.editTextNome);
    rgGrupo = (RadioGroup) findViewById(R.id.radioGroupGrupo);
    rbMasculino = (RadioButton) findViewById(R.id.radioMasculino);
    rbFeminino = (RadioButton) findViewById(R.id.radioFeminino);
    btnCadastrar = (Button) findViewById(R.id.buttonCadastrarPessoa);
    btnFechar = (Button) findViewById(R.id.buttonFecharCadastro);
    btnCadastrar.setOnClickListener(this);
    btnFechar.setOnClickListener(this);
}

public void onClick (View v) {
    if (v.getId() == btnFechar.getId()) {
        finish();
    }
    if (v.getId() == btnCadastrar.getId()) {
        Pessoa pessoa = new Pessoa();
        pessoa.setNome(edtNome.getText().toString());
        int id = rgGrupo.getCheckedRadioButtonId();
        if (id != -1) {
            RadioButton rbSelecionado = (RadioButton) findViewById(id);
            pessoa.setSexo(rbSelecionado.getText().toString());
            PrincipalActivity.ListaPessoas.add(pessoa);
            Toast.makeText(getBaseContext(), "Pessoa cadastrada!",
Toast.LENGTH_LONG).show();
        }
    }
}
}
}

```

Listagem 6. listagem.xml



```

<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="Listagem das Pessoas Cadastradas"
    android:textColor="#FFFF00"
    android:textAppearance="?android:attr/textAppearanceLarge"/>

<EditText
    android:id="@+id/editTextLista"
    android:inputType="textMultiLine"
    android:isScrollContainer="true"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:lines="4"
    android:scrollbars="vertical"
    android:textColor="#FFFF00"/>

<Button
    android:id="@+id/buttonFecharListagem"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="Fechar"/>

</LinearLayout>

```

Listagem 7. Listagem.java

```

package br.pdm.aula_03;

import br.pdm.aula_03.modelo.Pessoa;
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.Toast;

public class Listagem extends Activity implements OnClickListener {

    private Button btnFechar;
    private EditText edtLista;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.Listagem);
        btnFechar = (Button) this.findViewById(R.id.buttonFecharListagem);
        edtLista = (EditText) this.findViewById(R.id.editTextLista);
        btnFechar.setOnClickListener(this);

        for (int i = 0; i < PrincipalActivity.ListaPessoas.size(); i++) {
            Pessoa p1 = (Pessoa) PrincipalActivity.ListaPessoas.get(i);

            String reg = p1.getNome() + " - " + p1.getSexo() + "\n";
            edtLista.append(reg);
        }

        public void onClick(View v) {
            if (v.getId() == btnFechar.getId()) {
                Toast.makeText(getApplicationContext(), "Fechando...",
                    Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

```

        finish();
    }
}
}

```

Listagem 8. AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.pdm.aula_03"
    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.aula_03.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.aula_03.Cadastro" />
        <activity android:name="br.pdm.aula_03.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 03" 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.