

Programação para Dispositivos Móveis

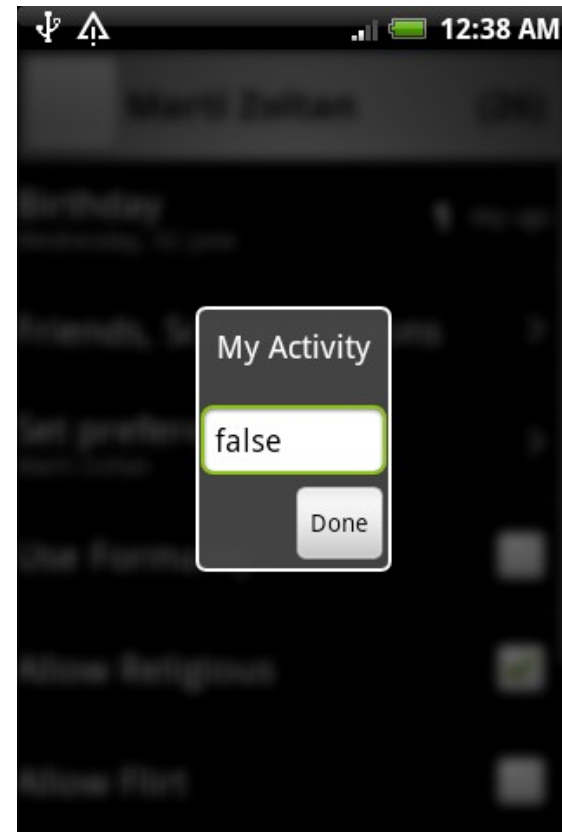
Fatec Ipiranga
Análise e Desenvolvimento de Sistemas

Aula 05 – Programando formularios, activities e intents.

Dalton Martins

dmartins@gmail.com

São Paulo, Março, 2012



Iniciando...

- Para melhor conhecermos as possibilidades de programação do Android, vamos entender melhor como ele organiza seus recursos, suas activities e layouts,
- Vamos iniciar um novo projeto padrao com o Eclipse...
- Agora, vamos abrir o arquivo **AndroidManifest.xml** para conhecer melhor seu modo de organização, alem do arquivo **strings.xml** que cuida de nossos poucos recursos ate aqui e o **main.xml**, que organiza o layout da tela.

AndroidManifest.xml

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

    <uses-sdk android:minSdkVersion="15" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >

        <activity
            android:name=".ProgramandoActivitiesActivity"
            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>
```

strings.xml

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

    <string name="hello">Hello World, ProgramandoActivitiesActivity!</string>
    <string name="app_name">ProgramandoActivities</string>

</resources>
```

main.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="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

</LinearLayout>
```

Fazendo algumas mudanças

- Vamos criar agora um novo recurso string para ser exibido em nossa tela. Para isso, teremos:
 - Incluir o recurso no arquivo strings.xml;
 - Incluir a visualização do recurso no arquivo main.xml;
 - Não será necessário nenhuma alteração no nosso código fonte, pois ele deixa o gerenciamento da estrutura de layout e dos conteúdos que serão exibidos a cargo dos arquivos xml. Veremos como isso funciona...

Novo arquivo strings.xml

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

    <string name="hello">Hello World, ProgramandoActivitiesActivity!</string>
    <string name="app_name">ProgramandoActivities</string>

    <string name="teste">Testando a criacao de um novo recurso!</string>

</resources>
```

Novo arquivo main.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="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/teste" />

</LinearLayout>
```


Introdução aos recursos de interação com usuário

- Vejamos agora como começar a manipular recursos para ampliar as possibilidades de interação com usuário em nossos programas.
- Vamos começar por entender melhor as possibilidades de exibição de texto que o Android nos oferece;
- O Android permite criarmos links contextuais em relação ao como os textos podem ser exibidos quando tiverem links especiais marcados:
 - Web;
 - E-mail;
 - Telefone;
 - Mapas...

Introdução aos recursos de interação com usuário

- Vejamos um exemplo.
- Vamos trabalhar com uma nova versão do arquivo main.xml e strings.xml, onde faremos algumas alterações para recursos de links de texto.

```
<string name="teste">Testando a criação de um novo recurso!  
Agora vamos começar a incorporar recursos de links nos textos.  
Por exemplo, meu blog tem o endereço: http://daltonmartins.blogspot.com  
Meu email: dmartins@gmail.com  
Meu telefone: (11) 3333-3333  
Address: 620 Eighth Avenue New York, NY 10018</string>
```

```
<TextView  
    android:id="@+id/TextViewContextual"  
        android:layout_width="fill_parent"  
        android:layout_height="wrap_content"  
        android:text="@string/teste"  
    android:autoLink="web|email|phone|map" />
```

Criando um pequeno formulario

- Os recursos de interaçao com usuario são construidos diretamente nos arquivos XML de layout em nosso diretorio de recursos;
- Em geral, para cada activity que implementarmos, iremos associar um arquivo de layout XML com a disposiçao dos elementos que serao visualizados pelos usuarios;
- Vejamos agora alguns exemplos de elementos de interface e o modo de organizaçao de sua sintaxe em XML:

Criando um pequeno formulario

- Vamos inserir o codigo abaixo no arquivo main.xml

```
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Nome:" />

    <EditText
        android:id="@+id/nome"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```

Criando um pequeno formulario

- Agora, vamos acrescentar um botao ao final da tela...

```
<Button
```

```
    android:id="@+id/enviar"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:text="Enviar" />
```

Exemplo rodando...

The screenshot displays a presentation software interface with a central window showing a mobile application. The application is titled "ProgramandoActivities" and has a status bar at the top showing "3G", signal strength, battery, and the time "12:43".

The main content of the app is a "Hello World" message: "Hello World, ProgramandoActivitiesActivity! Testando a criação de um novo recurso! Agora vamos começar a incorporar recursos de links nos textos. Por exemplo, meu blog tem o endereço: <http://daltonmartins.blogspot.com> Meu email: dmartins@gmail.com Meu telefone: (11) 3333-3333 Address: [620 Eighth Avenue New York, NY 10018](#)".

Below the text is a form with a label "Nome:" followed by a text input field. At the bottom of the form is a button labeled "Enviar".

To the right of the main content is a navigation panel with several icons: a camera, a speaker, a power button, a green phone icon, a central directional pad, a red phone icon, a home button, a "MENU" button, a back arrow, and a search icon.

Below the navigation panel is a QWERTY keyboard layout.

On the left side of the presentation software, a "Slides" pane shows a list of slides. Slide 14 is currently selected and highlighted with a blue border. The slide content includes a title "Clique para adicionar o título" and a bullet point "Clique para adicionar texto".

On the right side, a "Tarefas" (Tasks) pane is visible, showing a list of tasks: "Páginas mestre", "Layouts", "Modelos de tabela", "Animação personalizada", and "Transição de slides".

The bottom of the screen shows the Windows taskbar with several open applications: "Gmail - Entra...", "SENAC", "Android Prog...", "Projeto Peda...", "aula 05 - disp...", "Apresentação...", "Java - Progra...", and "5554:Teste". The system clock shows "10:18".

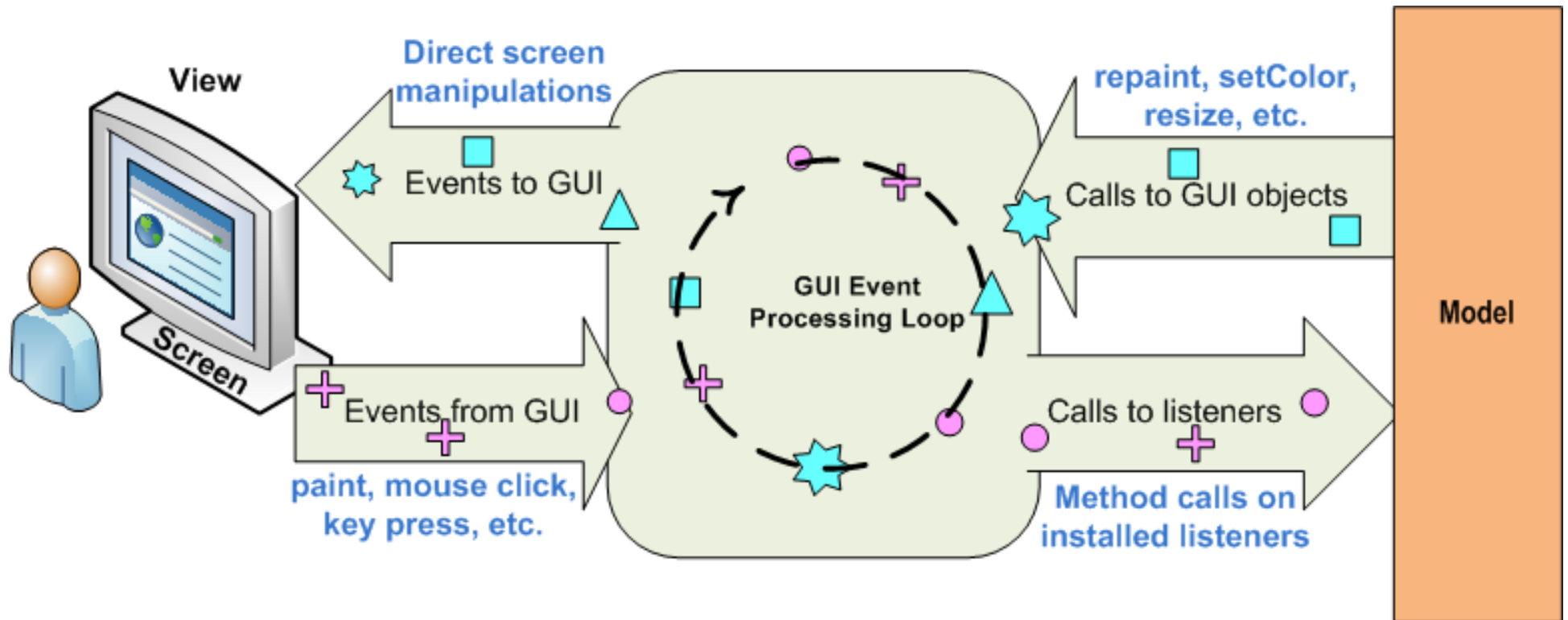
Transitando de atividades

- Agora que construímos um pequeno formulário e estamos mais ambientados com a estrutura de programação do Android, vamos experimentar como podemos usar as Intents para fazermos transições de Activity;
- Para isso, teremos de implementar os seguintes passos:
 - Criar uma nova activity;
 - Declarar a activity no arquivo AndroidManifest.xml;
 - Criar um novo layout XML para essa activity;
 - Criar um objeto Intent para operar a transição:
 - Na Activity inicial → implementar o botão;
 - Na Activity de transição → implementar o botão.

Entendendo melhor eventos em Java

- Quando transitamos de uma Activity para outra, precisamos implementar alguma interface de interação com o usuário que capture um determinado evento;
- A partir da ocorrência desse evento, será acionada uma nova Activity através do envio de uma mensagem via um Intent;
- Logo, será fundamental entendermos melhor como funciona a gestão de eventos no ambiente Java...

Esquema de funcionamento de eventos em Java



Observação: sempre teremos de implementar o que chamaremos daqui por diante de **listeners**, que funcionam como metodos que ficam a espera de que um determinado evento aconteça para poderem reagir com o resultado esperado.

Criando o novo arquivo de layout

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textotransicao"
        android:text="Nova atividade em operação!"
        android:textSize="18sp"></TextView>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/botaotransicao"
        android:text="Voltar"></Button>

</LinearLayout>
```

Declarando a Activity no arquivo AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.android.aulas"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="10" />
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >

        <activity
            android:name=".ProgramandoActivitiesActivity"
            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="TransicaoActivity"></activity>

    </application>
</manifest>
```

Modificando a Activity principal

```
package com.android.aulas;

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

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

        Button button = (Button) findViewById(R.id.enviar);
        button.setOnClickListener(this);
    }

    public void onClick(View src) {
        Intent i = new Intent(this, TransicaoActivity.class);
        startActivity(i);
    }
}
```

Criando uma nova Activity

- Para isso, vamos operar contando com ajuda do Eclipse:
 - Vamos no diretorio /src e clicamos com o botao direito do mouse no pacote com.android.aulas;
 - Escolhemos a opção criar nova classe Java;
 - Escolhemos o nome da nova classe;
 - O Eclipse criara uma estrutura basica de classe para podermos implementar, como o codigo abaixo.

```
package com.android.aulas;  
  
public class TransicaoActivity {  
  
}
```

Criando uma nova Activity

- Teremos de ajustar o código da classe padrão para que ela possa estender uma Activity e operar dentro do modelo da plataforma Android. Vejamos como ficara esse código:

Criando uma nova Activity

```
package com.android.aulas;

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

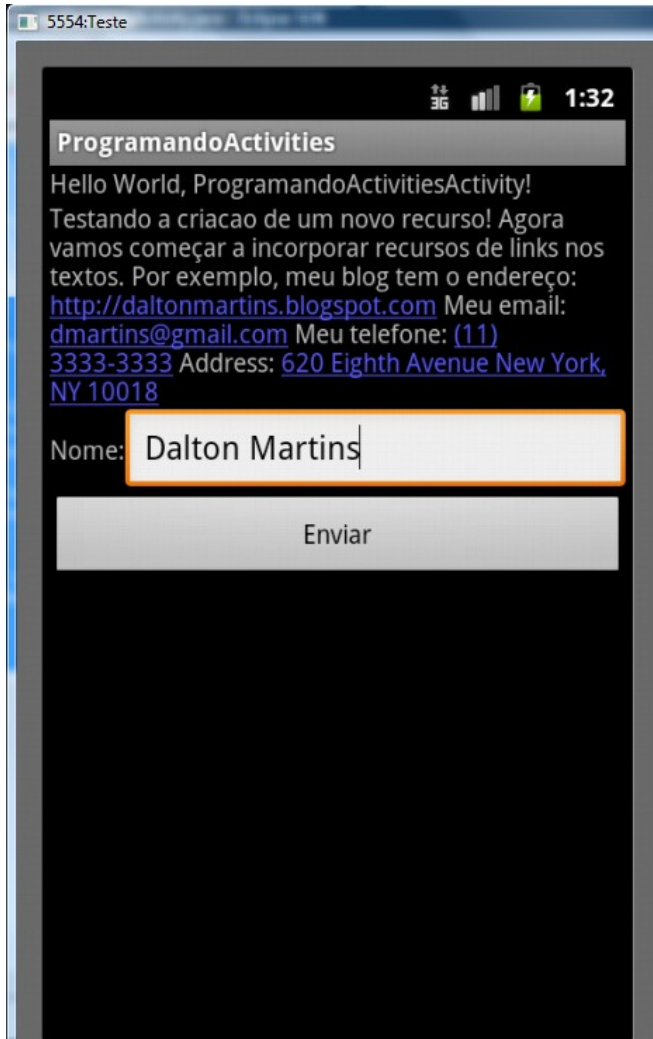
public class TransicaoActivity extends Activity implements OnClickListener {

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

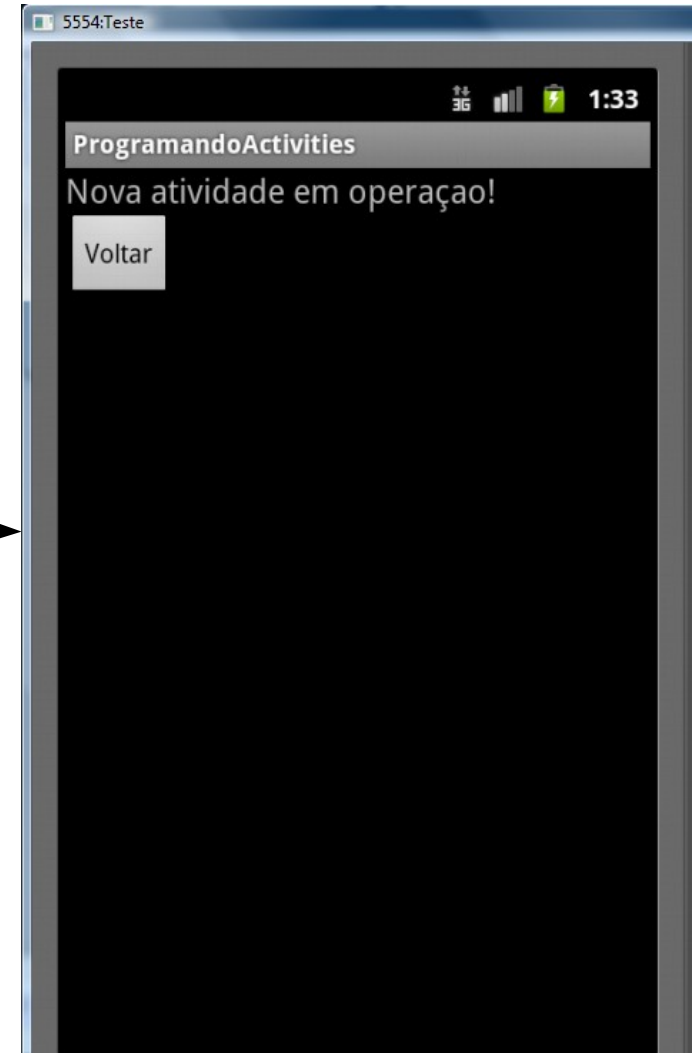
        Button button = (Button) findViewById(R.id.voltar);
        button.setOnClickListener(this);
    }

    public void onClick(View src) {
        Intent i = new Intent(this, ProgramandoActivitiesActivity.class);
        startActivity(i);
    }
}
```

Colocando tudo junto



As activities
podem transitar
entre si.



Exercicio

- Crie agora um programa para Android que implemente os seguintes requisitos:
 - Faça a transicao entre 3 activities;
 - Em cada uma delas ofereça um botao para avançar para a próxima ou voltar para primeira;
 - Coloque na tela alguns campos de texto, informando em qual Activity o usuario se encontra;
 - Coloque tambem campos para recolher informacao a acada Activity.