

Curso Android para *Iniciantes*



Lana Mesquita
Cristiane Mayara
Brunno Melo

Contatos



- Lana Mesquita
lanabeatriz.mesquita@gmail.com
- Cristiane Mayara
cristiane.mayara@gmail.com
- Brunno Melo
brunnomelo@gmail.com
- Grupo Facebook:
<https://www.facebook.com/groups/CursoAndroidCAET2014>

Ementa de hoje

Aula 4: Realizar ações

1. Intent
2. Shared Preferences
3. OptionMenu
4. Prática: implementar Intent e SharedPreferences

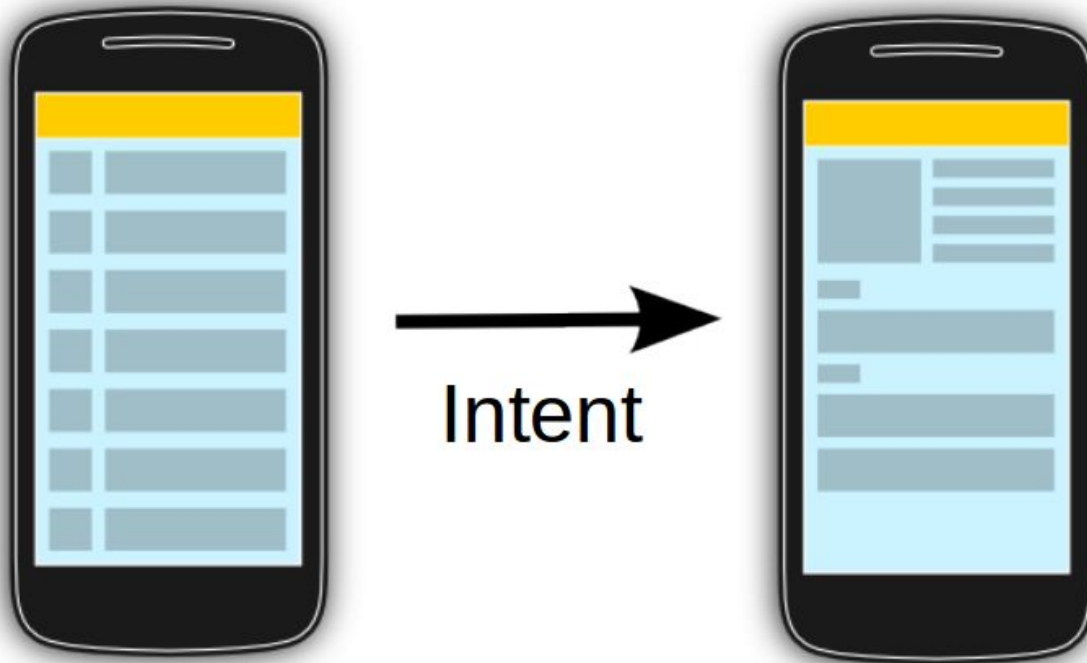


Intent

- “Intenção”
- É uma descrição abstrata de uma operação a ser executada
- “Pedidos” enviados ao S.O. Android
- Enviar e recuperar dados
- Iniciar uma nova activity
- Abrir uma pagina Web
- Realizar uma chamada telefônica



Intent



Intent

- Discar um número:

```
Intent intent = new Intent (Intent.ACTION_DIAL, Uri.parse("tel:93675359"));
startActivity(intent);
```

- Abrir um página Web:

```
Intent intent = new Intent (Intent.ACTION_VIEW, Uri.parse("http://codeandroid.org"));
startActivity(intent);
```

Intent

- Lançar outra Activity:

```
Intent intent = new Intent (this, ProximaActivity.class);  
startActivity(intent);
```

- Lançar outra Activity com dados:

```
Intent intent = new Intent (this, ProximaActivity.class);  
intent.putExtra("title", "Curso Android para Iniciantes");  
startActivity(intent);
```

Intent

- Outras ações
- Intent implícito
- Intent filter
- Categorias



Shared Preferences

- “Preferências compartilhadas”
- Formar de salvar as preferências do usuário
- Armazena dados de tipo primitivo
- Funciona no formato chave-valor



Shared Preferences



Shared Preferences

- `getPreferences()`

```
SharedPreferences myPreferences = getPreferences(Context.MODE_PRIVATE);
```

- `getSharedPreferences()`

```
SharedPreferences myPreferences =  
getSharedPreferences("Identificador", Context.MODE_PRIVATE);
```

- `getDefaultSharedPreferences()`

Shared Preferences

- Editor

```
SharedPreferences.Editor myEditor = myPreferences.edit();
```

- Put

- myEditor.putBoolean(“chave”, valor);
- myEditor.putFloat(“chave”, valor);
- myEditor.putInt(“chave”, valor);
- myEditor.putLong(“chave”, valor);
- myEditor.putString(“chave”, valor);

- Commit

```
editor.commit();
```

Shared Preferences

- Exemplo inserção de preferências:

```
SharedPreferences myPreferences =  
getSharedPreferences("com.sample", Context.MODE_PRIVATE);  
SharedPreferences.Editor myEditor = myPreferences.edit();  
myEditor.putBoolean("Music", true);  
myEditor.putInt("Volume", 30);  
myEditor.putString("Nome", valor);  
myEditor.putFloat("Altura", 1.78);  
editor.commit();
```

Shared Preferences

- Get
 - `myPreferences.getBoolean("chave", valor_padrão);`
 - `myPreferences.getInt("chave", valor_padrão);`
 - `myPreferences.getFloat("chave", valor_padrão);`
 - `myPreferences.getLong("chave", valor_padrão);`
 - `myPreferences.getString("chave", valor_padrão);`

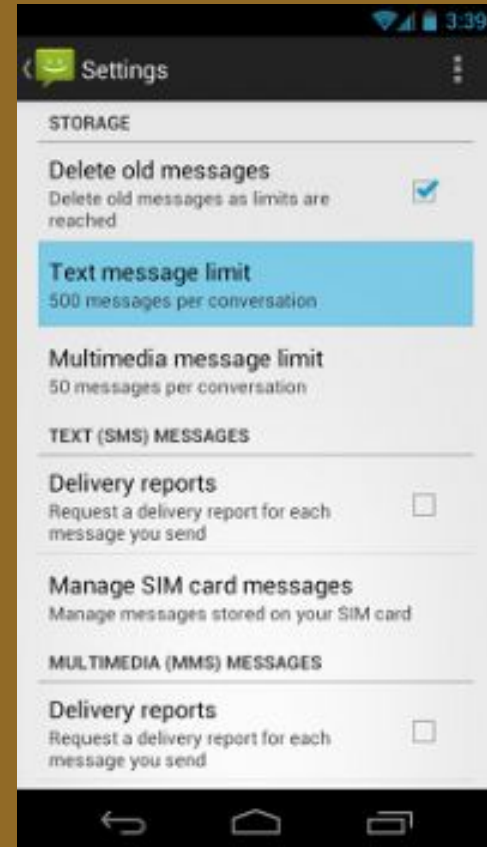
Shared Preferences

- Exemplo acesso de preferências:

```
SharedPreferences myPreferences = getPreferences("com.sample", Context.MODE_PRIVATE);  
musica = myPreferences.getBoolean("Musica", true);  
volume = myPreferences.getInt("Volume", 100);  
nome = myPreferences.getString("Nome", "Usuário");  
altura = myPreferences.getFloat("Altura", 1.65);
```

Shared Preferences

- PreferenceScreen
 - Tipo de tela própria para configurações/menu
 - Parecido com um xml de layout
- Componentes
 - CheckBoxPreference
 - DialogPreference
 - ListPreference
 - EditTextPreference
 - MultiSelectPreference
 - RingtonePreference
 - SwitchPreference
 - TwoStatePreference



Shared Preferences

- Exemplo

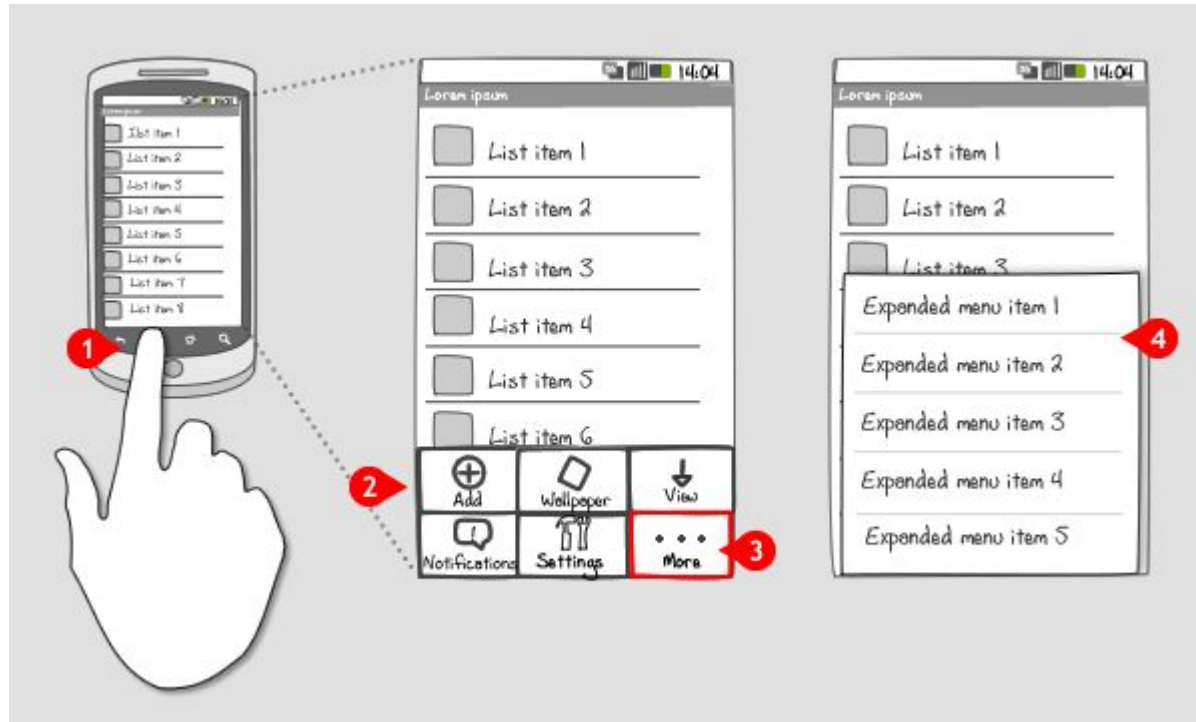
```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android"
>
    <CheckBoxPreference
        android:key="@string/pref_notificacao"
        android:title="@string/pref_notificacao_titulo"
        android:summary="@string/pref_notificacao_sumario"
        android:defaultValue="true" />
    <RingtonePreference
        android:dependency="notif"
        android:key="@string/pref_ringtone"
        android:ringtoneType="notification"
        android:showDefault="true"
        android:showSilent="true"
        android:summary="@string/pref_ringtone_sumario"
        android:title="@string/pref_ringtone_titulo" />
</PreferenceScreen>
```

Menu Option

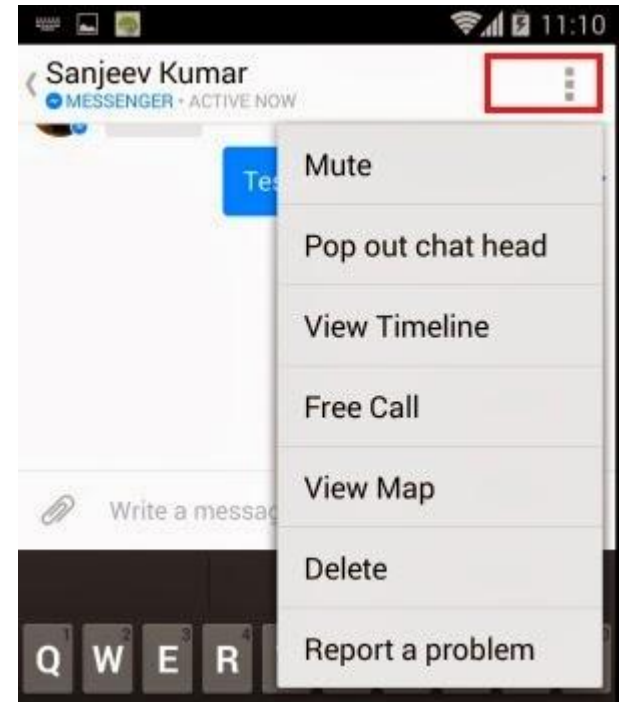
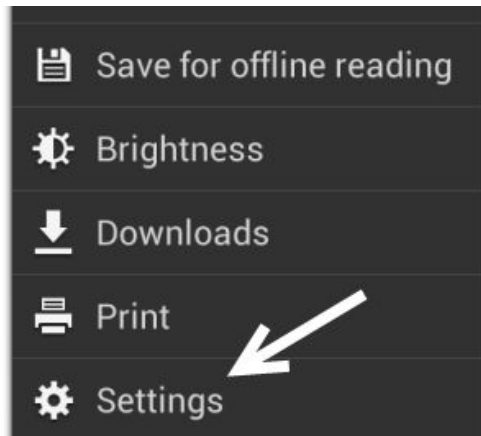
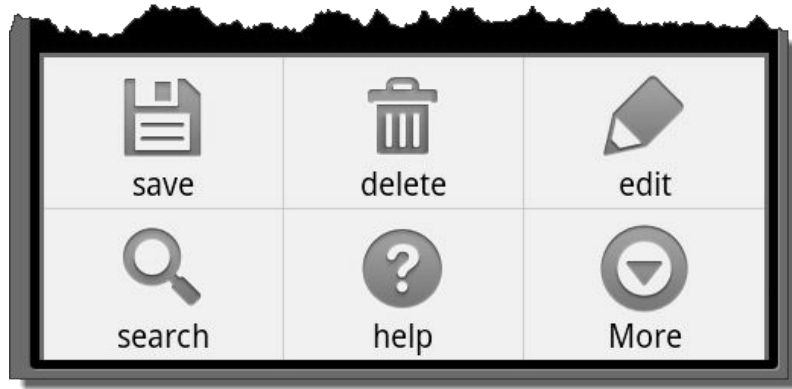
- Ações que são relevantes no contexto da activity corrente
- Exemplos: “Pesquisar”, “Compor email”, “Configurações”
- Mostrado ao pressionar o botão Menu, para versões de Android 2.3 ou menores
- Para versões 3.0 ou maiores o botão menu é *deprecated* (não presente em alguns dispositivos), podendo ser incorporado na *action bar*



Menu Option



Menu Option Examples



Creating a simple Menu Option



- Para especificar menu option para uma activity, os itens devem ser definidos no xml como mostrado abaixo.
- O xml deve ser incluído na pasta **res/menu/** (se a pasta não estiver criada, crie)
- No xml:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/my_item_1"
        android:icon="@drawable/menu_icon_1"
        android:title="Item 1"
        android:showAsAction="ifRoom"/>
    <item android:id="@+id/my_item_2"
        android:icon="@drawable/menu_icon_2"
        android:title="Item 2" />
</menu>
```

Creating a simple Menu Option



Atributos de Item:

- **android:id** O id do recurso que é exclusivo para o item, o que permite que o aplicativo ser capaz de reconhecer o item quando o usuário selecioná-lo.
- **android:icon** Uma referência a um drawable para usar como ícone do item.
- **android:title** Uma referência a uma string para usar como título do item.
- **android:showAsAction** Especifica quando e como este item deve aparecer como um item de ação na ActionBar.

Creating a simple Menu Option



- Em versões 3.0, o sistema chama o método `onCreateOptionsMenu()` para criação do menu option.
- Na classe java:

```
// Inicializar o arquivo de menu xml
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.my_option_menu, menu);
    return true;
}
```

Creating a simple Menu Option



- Ainda na classe java:

```
// Tratar seleção do item

@Override

public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {

        case R.id.my_item_1:

            itemOneAction();

            return true;

        case R.id.my_item_2:

            itemTwoAction();

            return true;

        default:

            return super.onOptionsItemSelected(item);

    }

}
```


Prática

- Primeira tela (Layout):

- utilizar uma TableLayout com 3 elementos:

- <TextView...

- <TableRow...

- <TableRow...

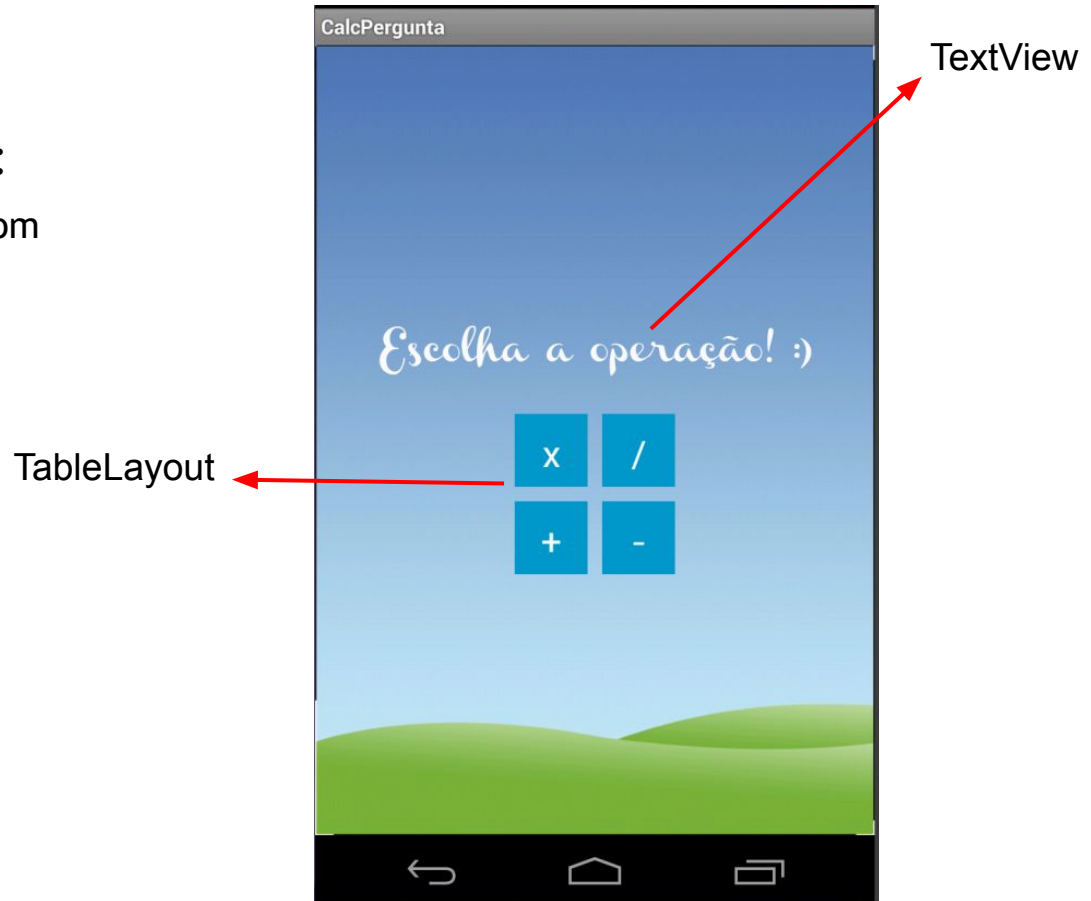
- utilizar parâmetro

android:gravity="center"

- usar parâmetros

android:layout_margin

android:padding







Prática

- Primeira tela (Java):
 - escolhe a operação
 - implementar 4 botões
 - criar um método `callActivity()` onde será chamada a próxima Activity (ver slide 7)
 - criar uma variável `SharedPreferences` para salvar a operação (+,-,x, /)
 - setar o texto com a fonte Channel
 - Utilizar um Log para verificar a operação chamada



Prática

- Primeira tela (Java):
 - criar um Log.v para retornar a operação selecionada

Search for messages. Accepts Java regexes. Prefix with pid:, app:, tag: or text: to limit scope.							verbose ▾				
L...	Time	PID	TID	Application	Tag	Text					
V	06-07 06:22:5...	1242	1242	br.ifce.cursoan...	CALC	Realizando jogo da multiplicação					
I	06-07 06:23:1...	1242	1242	br.ifce.cursoan...	Choreographer	Skipped 44 frames! The application may be doing too much work on i [0]ts main thread.					
V	06-07 06:24:1...	1242	1242	br.ifce.cursoan...	CALC	Realizando jogo da soma					
I	06-07 06:25:5...	1242	1242	br.ifce.cursoan...	Choreographer	Skipped 30 frames! The application may be doing too much work on i [0]ts main thread.					
I	06-07 06:34:0...	1242	1242	br.ifce.cursoan...	Choreographer	Skipped 33 frames! The application may be doing too much work on i [0]ts main thread.					

Prática

- Segunda tela:
 - tela já implementada na aula passada
 - criar um método setResult com o resultado da soma
 - criar um método para mostrar a operação escolhida



Prática

- Desafio:
 - consertar o bug: caso o EditText esteja vazio e você clique em Conferir o app quebra!
 - criar um layout para landscape, utilizando os qualifiers:
 - + layout-land
 - + layout-port



Dicas

- Uma Activity deve sempre implementar o método `onCreate()`, que deve chamar o método `setContentView(view)` para definir o layout
- Três passos importantes para criar uma Activity:
 1. Criar uma Activity (ex: `TesteActivity.java`)
 2. Declarar no Manifest:

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

3. Criar um layout para a activity (ex: `teste_layout`)



Dicas

Chamar e setar um Widget:

- criar o layout

```
<TextView
    android:id="@+id/inicial_tv_selecionar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/inicial_selecionar" />
```

- Chamar o setContentView antes de buscar a variável

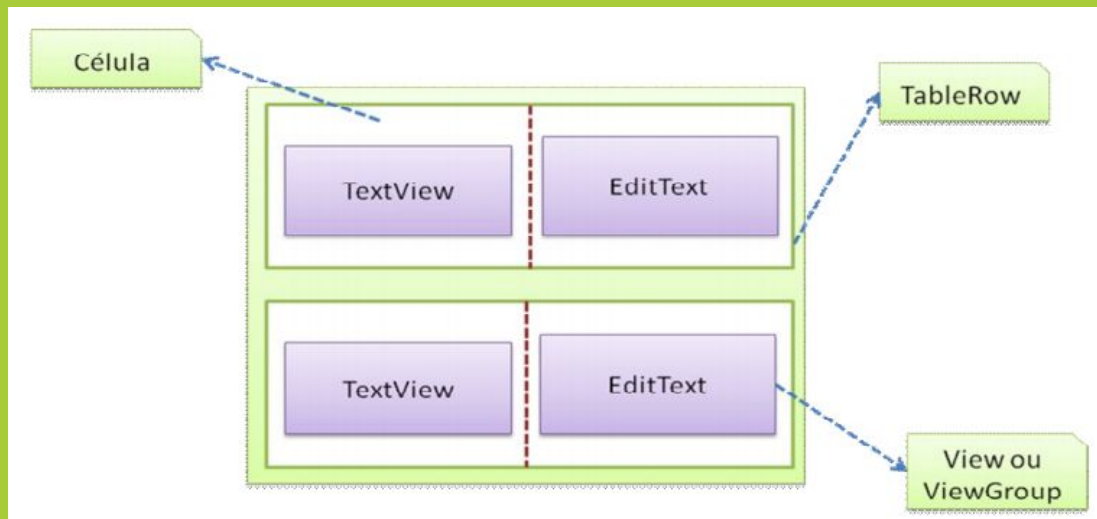
```
TextView tv_escolha;
onCreate(){
    setContentView(R.layout.inicial_layout);
    ...
    tv_escolha = (TextView) findViewById(R.id.inicial_tv_selecionar);
    ...
    tv_escolha.setTypeface(type); ... }
```



Dicas

TableLayout

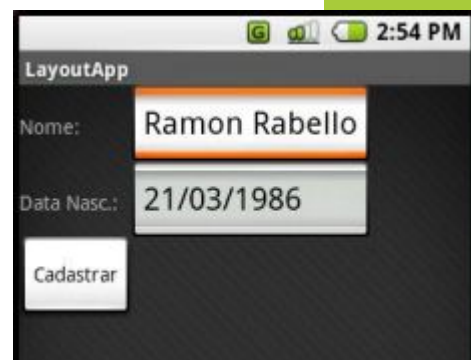
O TableLayout comporta seus filhos em linhas e colunas. Cada filho é representado pelo componente TableRow (um tipo de LinearLayout horizontal)



Dicas

TableLayout

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="wrap_content"
    android:layout_height="fill_parent">
    <TableRow>
        <TextView android:text="Nome:" />
        <EditText android:text="Ramon Rabello" />
    </TableRow>
    <TableRow>
        <TextView android:text="Data Nasc.:" />
        <EditText android:text="21/03/1986" />
    </TableRow>
    <TableRow>
        <Button android:text="Cadastrar" />
    </TableRow>
</TableLayout>
```



Dicas

Button

```
// Button Multiplicar  
bt_multiplicar.setOnClickListener(new OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        prefs.edit().putInt("option", 3).commit();  
        Log.v("CALC", "Realizando jogo da multiplicação");  
        callActivity();  
    }  
});
```



Dicas

setResult()

```
private int setResult(int number1, int number2) {  
    int result = 0;  
    if (option == 1){  
        result = number1 + number2;  
        tvOption.setText(" + ");  
    }  
    else if (option == 2){  
        result = number1 - number2;  
        tvOption.setText(" - ");  
    }  
    else if (option == 3){  
        result = number1 * number2;  
        tvOption.setText(" * ");  
    }  
    else if (option == 4){  
        result = number1 / number2;  
        tvOption.setText(" / ");  
    }  
    return result;  
}
```



Dicas

parseFloat()

- Método para transformar uma string em Double

```
// pega o resultado que o usuário digitou no EditText
double int_result = Double.parseDouble(et_result.getText()
                                     .toString());
```

