

Aula 05 – Sudoku

Programação em Java para a Plataforma Android



Agenda

- Refinando layouts
- Lidando com telas em modo paisagem
- Invocando atividades dentro de atividades
- Usando temas para melhorar layouts
- Depurando via o console de saída
- Carregando uma atividade



Sudoku

- Sudoku é um jogo de raciocínio, cujo objetivo é preencher uma matriz de 9x9 casas com números inteiros entre 1 e 9 inclusive, de tal forma que nenhuma linha, coluna ou diagonal possua números iguais

Este exemplo foi tirado do Capítulo 3 do livro
“Hello, Android (The Pragmatic Programmers)”
por Ed Buenette



Visão geral do jogo



Visão geral do jogo



Para que serve
cada uma dessas
classes?



A primeira atividade

```
package com.cursoandroid.aula05;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class Sudoku extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sudoku);
    }
}
```

O layout principal

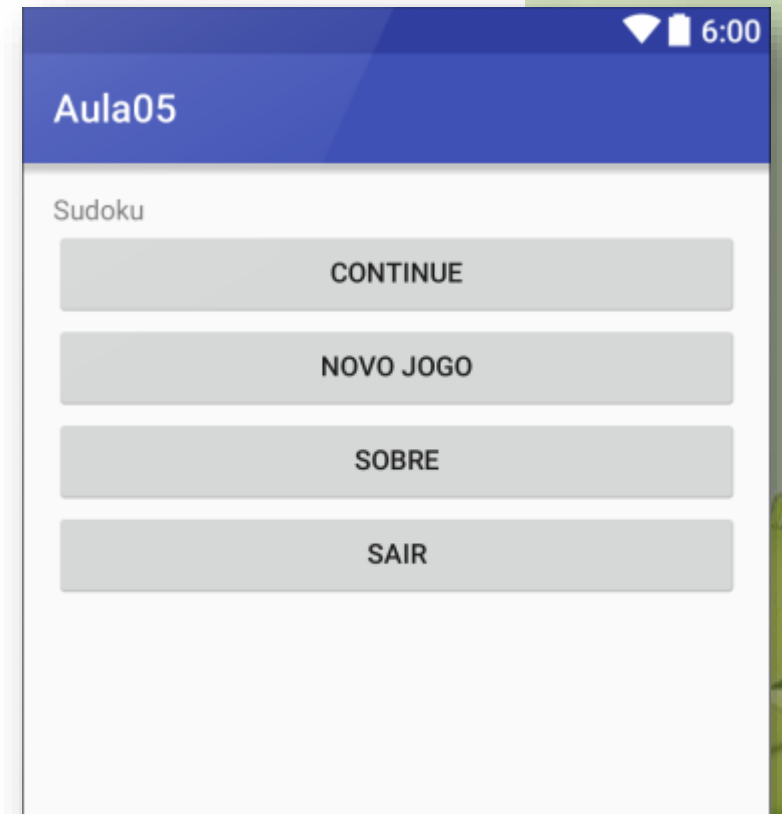
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/main_title" />
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/continue_label" />
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/new_game_label" />
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/about_label" />
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/exit_label" />
</LinearLayout>
```

Você consegue
prever como
será a tela
principal da
nossa aplicação?



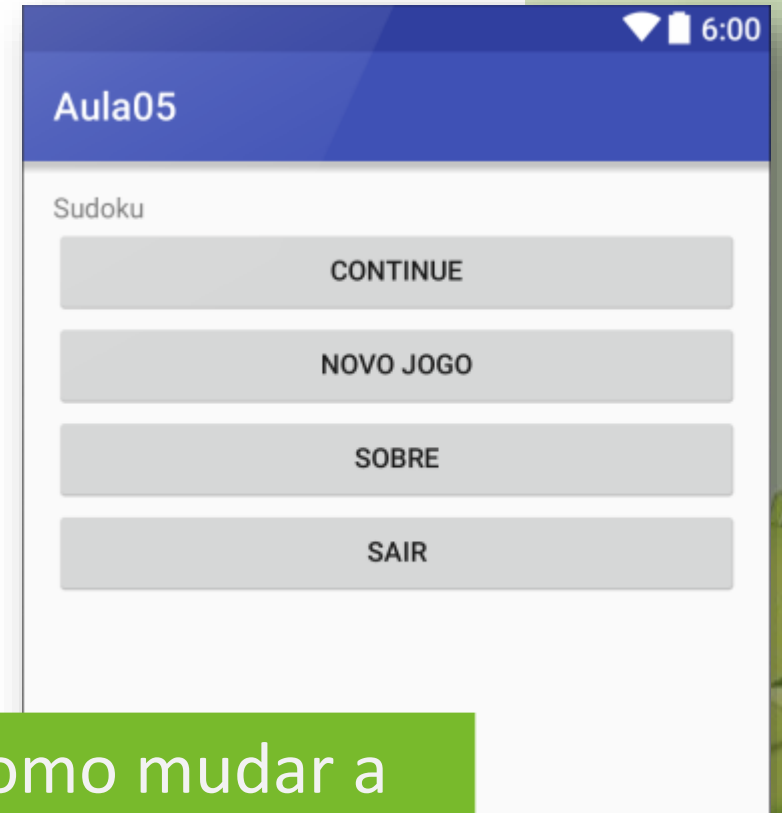
O layout principal

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/main_title" />
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/continue_label" />
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/new_game_label" />
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/about_label" />
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/exit_label" />
</LinearLayout>
```



O layout principal

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/main_title" />
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/continue_label" />
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/new_game_label" />
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/about_label" />
    <Button android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/exit_label" />
</LinearLayout>
```



Como mudar a cor de fundo?

Cores também são recursos

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="background">#AA3F51B5</color>
</resources>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    ...
    android:background="@color/background">
    ...
</LinearLayout>
```



Como adicionar
margens entre os
botões e o
perímetro da tela?

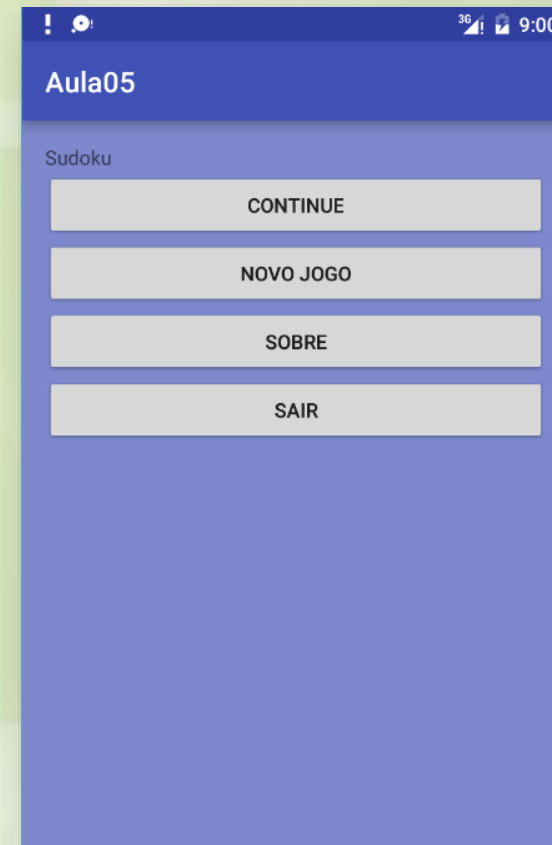


Margens

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    ...
    android:padding="30dp"
    ...
</LinearLayout>
```

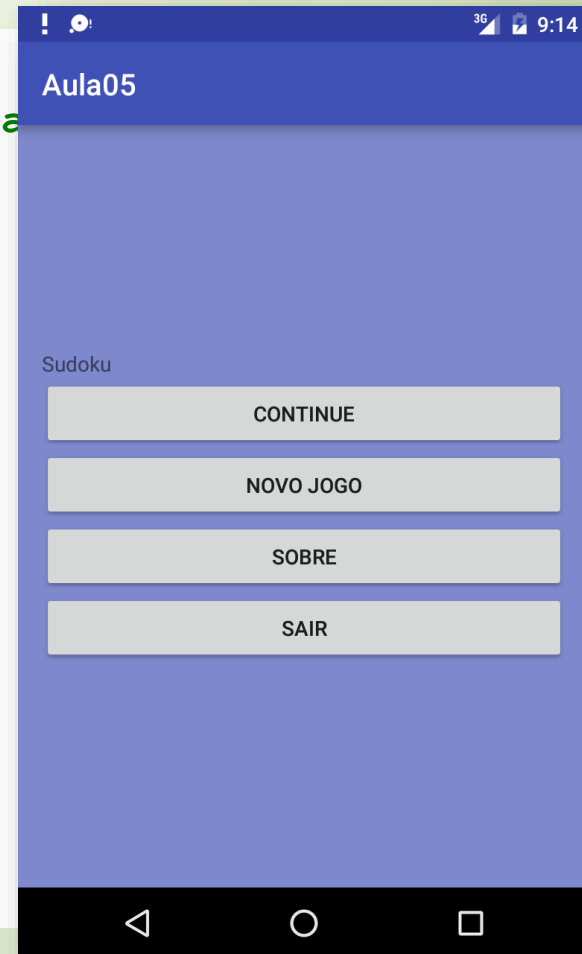
O que são dp's?

Como centralizar
os botões
verticalmente?



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/a
    android:orientation="horizontal"
    ...
    android:background="@color/background">
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_gravity="center">
        <TextView ... />
        <Button ... />
        <Button ... />
        <Button ... />
        <Button ... />
    </LinearLayout>
</LinearLayout>
```

Não dá pra
aumentar o
tamanho do título
não?



Interface final

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    ... >
    <LinearLayout
        ... >
            <TextView android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:text="@string/main_title"
                android:layout_marginBottom="25dp"
                android:textSize="24.5sp"/>
            <Button ... />
            <Button ... />
            <Button ... />
            <Button ... />
        </LinearLayout>
    </LinearLayout>
```

O que é
"sp"?



Unidades de medida

- **px** (pixels): pontos na tela
- **in** (polegadas): tamanho real (medido por régua)
- **mm** (milímetros): tamanho real
- **pt** (pontos): 1/72 de uma polegada
- **dp** (*density-independent pixels*): em uma tela com 160 pontos por polegada, 1dp = 1px. Se forem 320 pontos, então 1dp = 2px, etc
- **sp** (*scale-independent pixels*): mesmo que dp porém ajustável ao tamanho da fonte



Layout de paisagem

- Muitos smartphones possuem um layout para o modo paisagem, chamado quando viramos o aparelho
- Nosso layout de paisagem ficou assim:
- Qual o problema desse layout?
- E como resolvê-lo?



Alternar para o modo paisagem no emulador:

Mac → ctrl + fn + f11

Windows → left-ctrl + f11

Linux → ctrl + f11



Layout de paisagem

- É possível definir um layout de paisagem editando o arquivo `res/layout-land/sudoku_activity.xml`
- Como criar o layout abaixo?



Tabelas

```
<TableLayout
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:layout_gravity="center"
    android:stretchColumns="*">
    <TableRow>
        <Button
            android:id="@+id/continue_button"
            android:text="@string/continue_label" />
        <Button
            android:id="@+id/new_button"
            android:text="@string/new_game_label" />
    </TableRow>
    <TableRow>
        <Button
            android:id="@+id/about_button"
            android:text="@string/about_label" />
        <Button
            android:id="@+id/exit_button"
            android:text="@string/exit_label" />
    </TableRow>
</TableLayout>
```



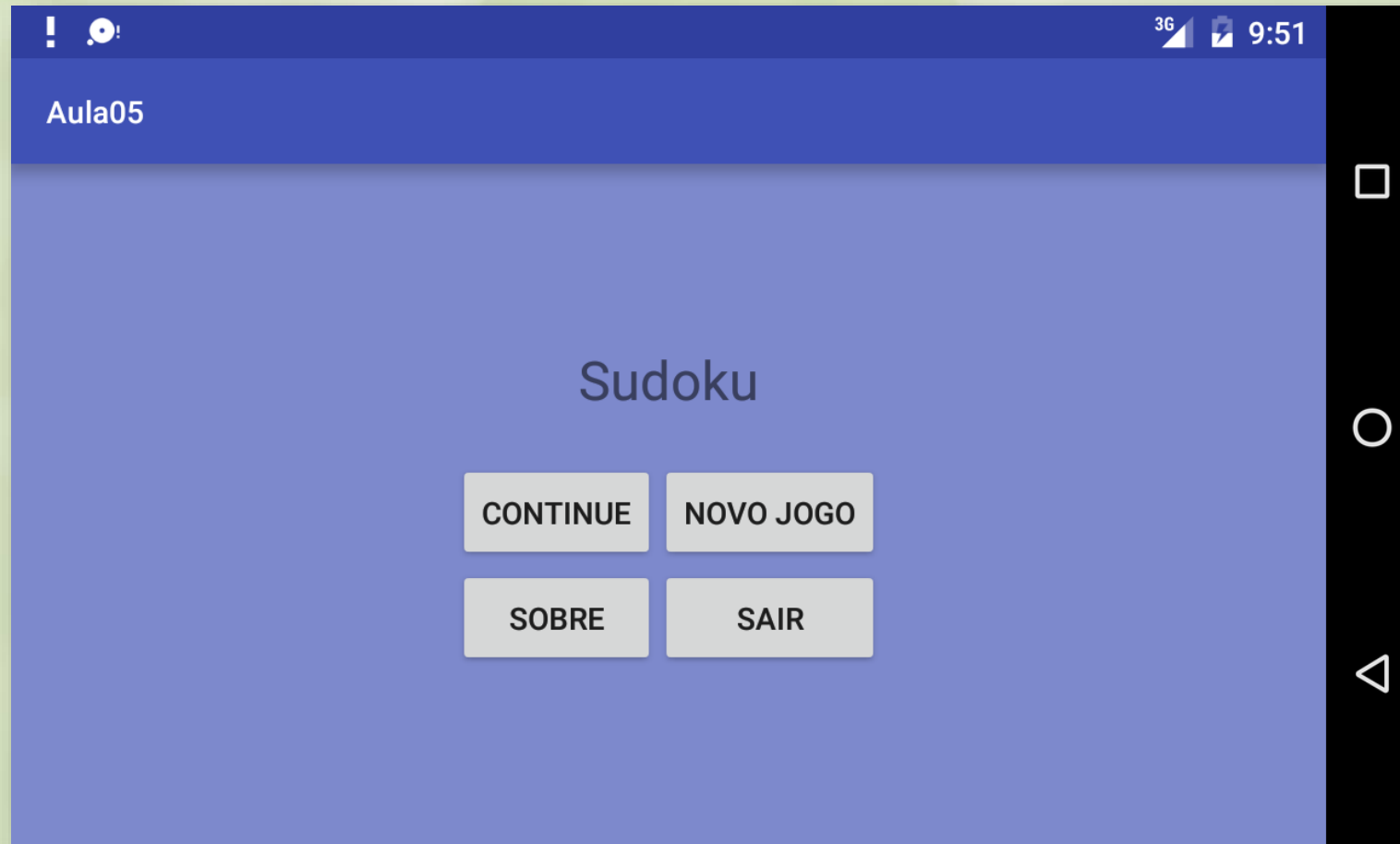
Tabelas

```
<TableLayout
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:layout_gravity="center"
    android:stretchColumns="*">
    <TableRow>
        <Button
            android:id="@+id/continue_button"
            android:text="@string/continue_label" />
        <Button
            android:id="@+id/new_button"
            android:text="@string/new_game_label" />
    </TableRow>
    <TableRow>
        <Button
            android:id="@+id/about_button"
            android:text="@string/about_label" />
        <Button
            android:id="@+id/exit_button"
            android:text="@string/exit_label" />
    </TableRow>
</TableLayout>
```

E se removermos
esse pedacinho de
texto **aqui**?

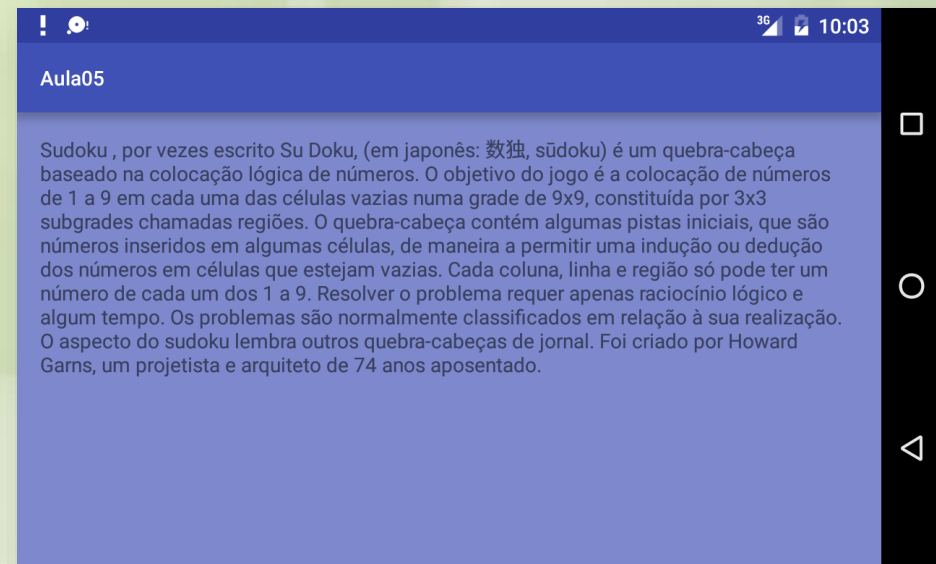
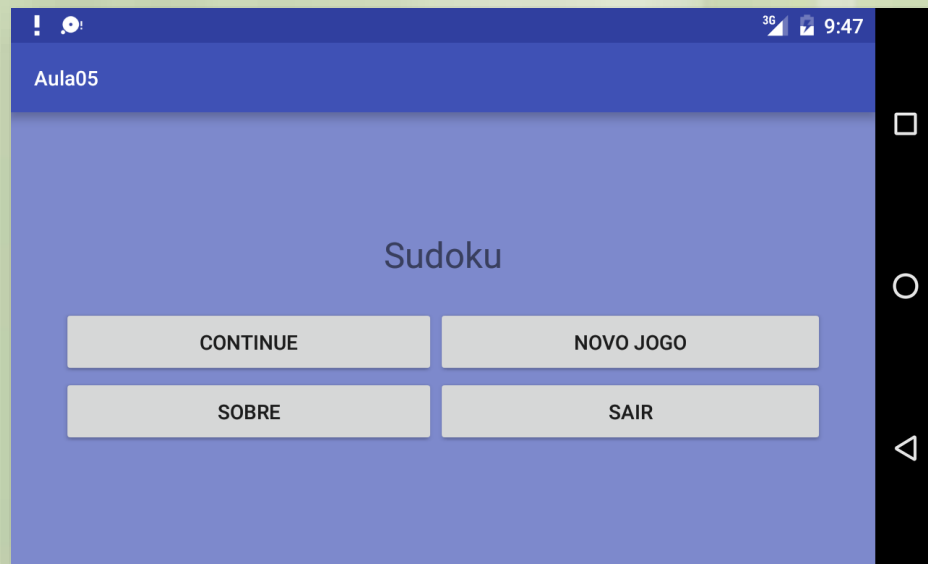


Não fica tão legal



Botão about

- O que poderíamos fazer com o botão about?
 - Outras aplicações possuem uma funcionalidade assim?
 - Como implementá-la?



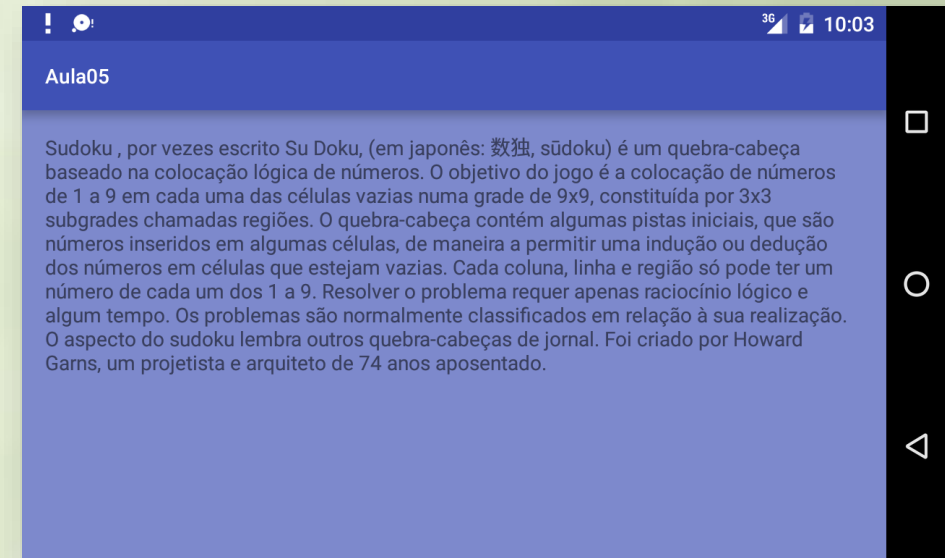
Activity About

```
package com.cursoandroid.aula05;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class AboutActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about);
    }
}
```



Como é o código
do layout de nossa
atividade?



Um layout para About

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:background="@color/background">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/about_text"/>
</ScrollView>
```

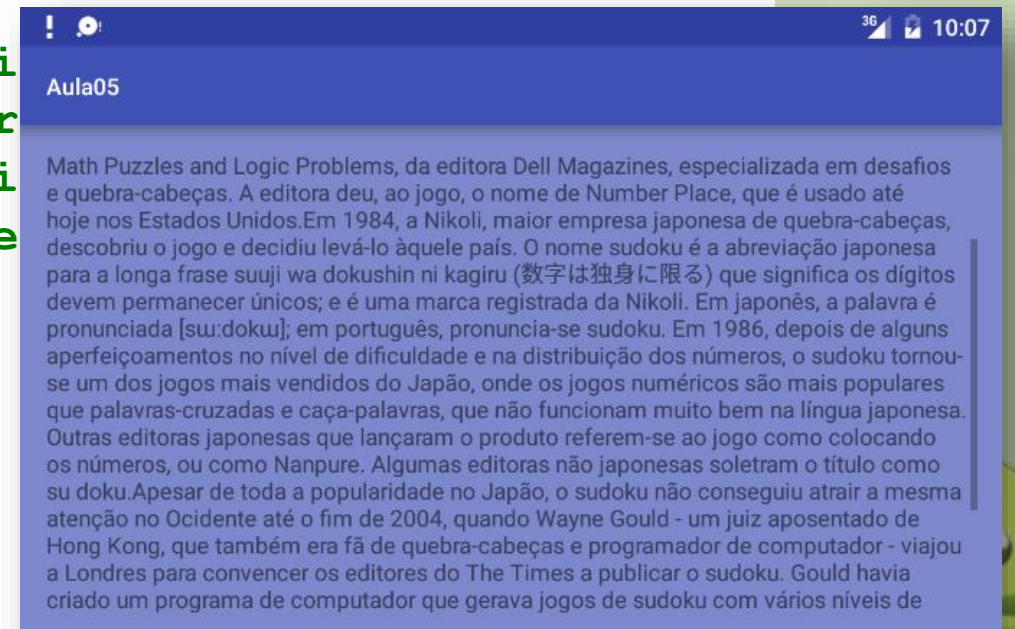
Para que serve
esse tipo de
layout?



Um layout para About

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_hori
    android:paddingRight="@dimen/activity_hori
    android:paddingTop="@dimen/activity_verti
    android:paddingBottom="@dimen/activity_ve
    android:background="@color/background">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/about_text"/>
</ScrollView>
```

Como a atividade
About deve ser
invocada?



O manifesto

- Para adicionar-se uma nova atividade, é preciso modificar o arquivo AndroidManifest.xml

```
<activity android:name=".AboutActivity"  
          android:label="@string/sobre_o_sudoku">  
</activity>
```

- Quando se cria a atividade através do template do Android Studio, este código é adicionada automaticamente



Relacionamento entre atividades

- Uma atividade pode invocar outra, via objetos chamados intenções (Intent)

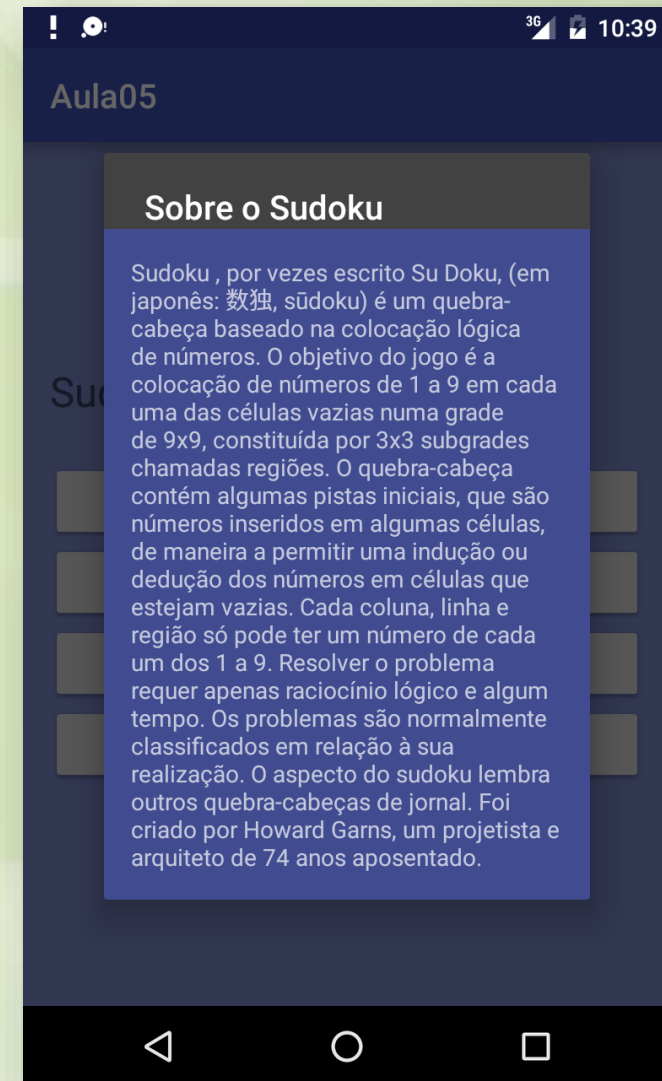
```
public class Sudoku extends AppCompatActivity implements View.OnClickListener {  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_about);  
        View aboutButton = findViewById(R.id.about_button);  
        aboutButton.setOnClickListener(this);  
    }  
    public void onClick(View v) {  
        switch (v.getId()) {  
            case R.id.about_button:  
                Intent i = new Intent(this, AboutActivity.class);  
                startActivity(i);  
                break;  
        }  
    }  
}
```

Pergunta repetida:
Como se dá a amarração
entre botão e evento?



Caixa de dialogo

- Seria mais interessante que a atividade About fosse mostrada como uma caixa de diálogo
- Como podemos modificar a aplicação para obtermos uma caixa de diálogo?



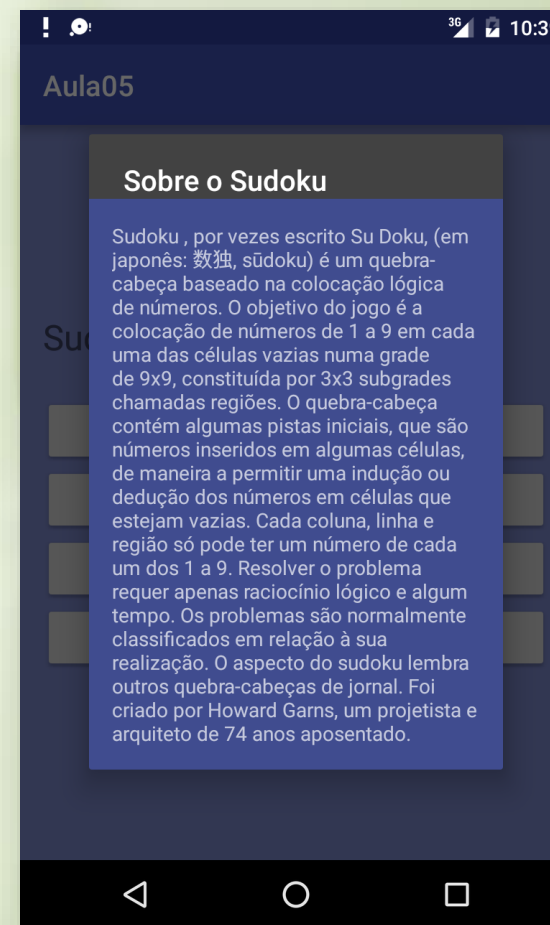
Temas

- O Android permite que desenvolvedores criem Temas
- Temas são como arquivos em formato CSS
- Já existem alguns temas prontos

```
<activity android:name=".AboutActivity"
    android:label="@string/sobre_o_sudoku"
    android:theme="@style/Theme.AppCompat.Dialog">
</activity>
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_about);

    setFinishOnTouchOutside(true);
}
```



Adicionando configurações

Menu de preferências:

Adicione um menu à sua aplicação, de tal forma que o usuário possa configurar o ambiente de jogo

- O usuário deveria ser capaz de escutar música enquanto brinca
- E ele poderia pedir dicas para a nossa aplicação
- Essas opções podem ser configuradas via um menu de configuração

Começemos pelo menu. Layout?



string.xml

- Vamos criar algumas strings que nos auxiliarão na construção do menu:

```
<string name="settings_label">Configurações...</string>  
<string name="settings_title">Preferências do Sudoku</string>  
<string name="settings_shortcut">s</string>  
<string name="music_title">Música</string>  
<string name="music_summary">Tocar música de fundo</string>  
<string name="hints_title">Dicas</string>  
<string name="hints_summary">Mostrar dicas durante o jogo</string>
```

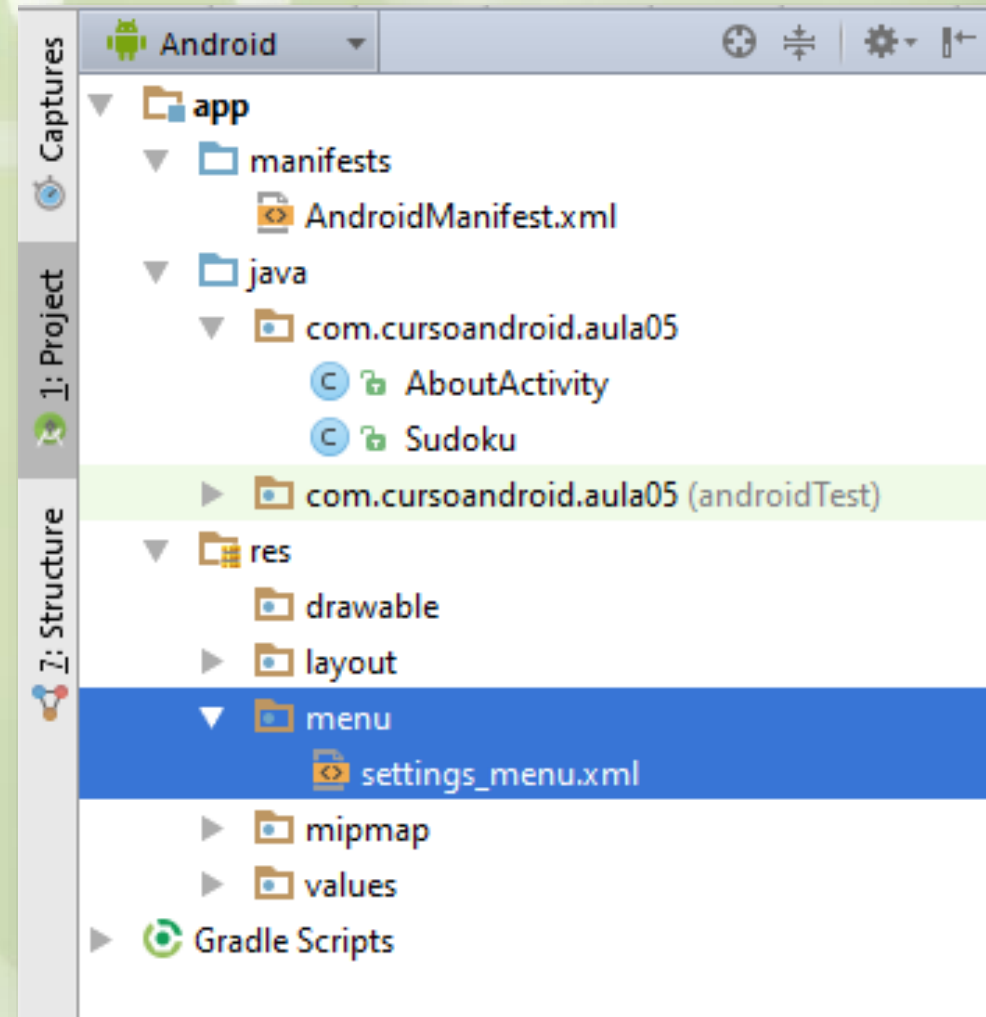
Novamente: por que criar strings em string.xml? Quais strings devem ficar lá?

Tudo bem, mas vocês lembram como inserir o menu em nossa aplicação?



O layout do menu

- O ambiente de desenvolvimento Android define uma pasta menu, em que é definido o layout do menu da aplicação



O layout do menu

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
<item
    android:id="@+id/settings"
    android:title="@string/settings_label"
    android:alphabeticShortcut="@string/settings_shortcut" />
</menu>
```

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
<item
    android:id="@+id/settings"
    android:title="Configurações"
    android:alphabeticShortcut="s" />
</menu>
```

Será que
isso daria
certo?



Ativando o menu

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.settings_menu, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.settings:
            startActivity(this, new Intent(Prefs.class));
            return true;
    }
    return false;
}
```

Como deveria ser o layout de **Prefs**?

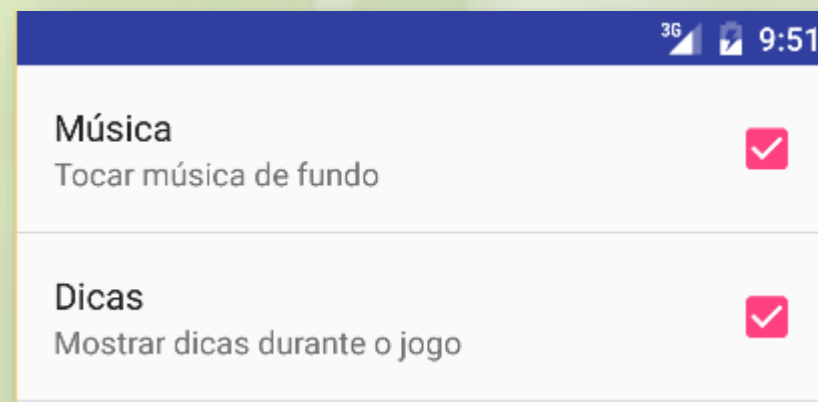
E como deveria ser essa atividade **Prefs** que modifica as preferências do usuário?



A caixa de preferências

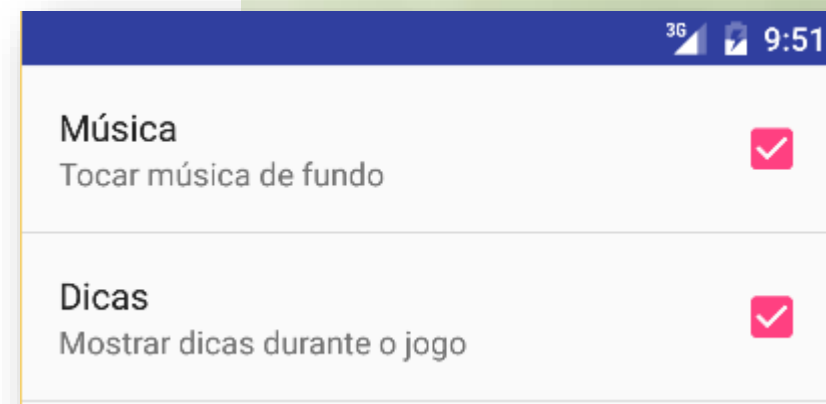
- Podemos usar botões de verificação!

Mas como
implementar
esse layout?



Caixas de preferência

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <CheckBoxPreference
        android:key="music"
        android:title="@string/music_title"
        android:summary="@string/music_summary"
        android:defaultValue="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <CheckBoxPreference
        android:key="hints"
        android:title="@string/hints_title"
        android:summary="@string/hints_summary"
        android:defaultValue="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</PreferenceScreen>
```



E a implementação
da atividade em si?



A atividade Prefs

```
import android.os.Bundle;
import android.preference.PreferenceActivity;
import android.preference.PreferenceFragment;

public class Prefs extends PreferenceActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getFragmentManager().beginTransaction()
            .replace(android.R.id.content, new MyPreferenceFragment()).commit();
    }

    public static class MyPreferenceFragment extends PreferenceFragment {
        @Override
        public void onCreate(final Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            addPreferencesFromResource(R.xml.prefs);
        }
    }
}
```

O que é **esse** método aqui?

E o que está faltando para usarmos a atividade?



Adicionar a atividade ao Manifesto

```
<activity android:name=".Prefs"  
    android:label="@string/settings_title">  
</activity>
```

Níveis de dificuldade:

Quando o usuário clicar em “Novo jogo”, ele deverá receber um menu de contexto, com três opções de dificuldade: fácil, médio e difícil

Seria legal um pouco de música não? Bom, isso fica para uma próxima aula.

Por hora, vamos nos preocupar com o botão “Novo jogo”.



Registrando um evento

- Precisamos de um identificador para o botão que começará o jogo:

activity_sudoku.xml

```
<Button  
    android:id="@+id/new_button" android:layout_width="fill_parent"  
    android:layout_height="wrap_content" android:text="@string/new_game_label" />
```

- E precisamos de um escutador de eventos para o novo botão:

Sudoku.java

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    ...  
    View novoJogoButton = findViewById(R.id.about_button);  
    novoJogoButton.setOnClickListener(this);  
}
```



Níveis de dificuldade

Sudoku.java

```
public void onClick(View v) {
    switch (v.getId()) {
        ...
        case R.id.new_button:
            iniciarNovoJogo();
            break;
    }
}

private void iniciarNovoJogo() {
    new AlertDialog.Builder(this)
        .setTitle(R.string.new_game_title)
        .setItems(R.array.difficulty, new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int difficulty) {
                novoJogo(difficulty);
            }
        }).show();
}

private void novoJogo(int difficulty) {
    Log.d("novoJogo", "selecionou o " + difficulty);
}
```

Dificuldade

Fácil

Médio

Difícil

SAIR

Existem alguns
recursos que
precisamos definir.



Mais Strings e um Arranjo

Strings.xml

```
<string name="new_game_title">Dificuldade</string>
<string name="easy_label">Fácil</string>
<string name="medium_label">Médio</string>
<string name="hard_label">Difícil</string>
```

values/array.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <array name="difficulty">
        <item>@string/easy_label</item>
        <item>@string/medium_label</item>
        <item>@string/hard_label</item>
    </array>
</resources>
```

O que acontece
quando clicamos no
botão “Novo jogo”:

```
04-11 10:41:35.970 15602-15602/com.cursoandroid.aula05 D/novoJogo: selecionou o 2
04-11 10:41:35.970 15602-15629/com.cursoandroid.aula05 E/Surface: getSlotFromBuffer
04-11 10:41:35.980 15602-15602/com.cursoandroid.aula05 W/InputEventReceiver: Attempt
04-11 10:41:37.850 15602-15629/com.cursoandroid.aula05 W/EGL_emulation: eglSurfaceA
04-11 10:41:37.850 15602-15629/com.cursoandroid.aula05 W/OpenGLRenderer: Failed to
```

Ainda o manifesto

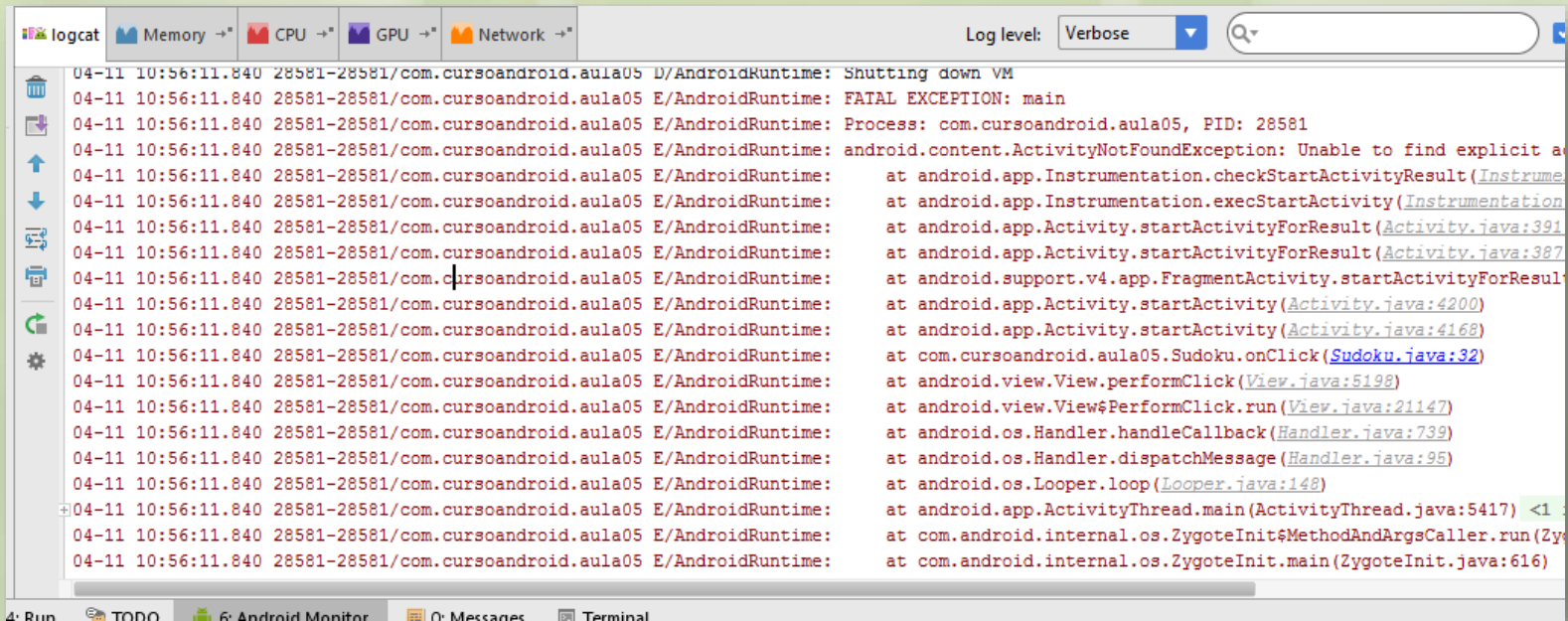
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.cursoandroid.aula05" >
    <application android:allowBackup="true" android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name" android:supportRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".Sudoku" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".AboutActivity"
            android:label="@string/sobre_o_sudoku"
            android:theme="@style/Theme.AppCompat.Dialog"
        </activity>
        <activity android:name=".Prefs" android:label="@string/settings_title">
        </activity>
    </application>
</manifest>
```

O que aconteceria se
apagássemos a declaração
da atividade no manifesto?

Usando o console

- Observaríamos a seguinte mensagem no console:

android.content.ActivityNotFoundException: Unable to find explicit activity class {com.cursoandroid.aula05/com.cursoandroid.aula05>AboutActivity}; have you declared this activity in your AndroidManifest.xml?



The screenshot shows the Android Studio interface with the Logcat window open. The log level is set to 'Verbose'. The log shows a fatal exception: 'android.content.ActivityNotFoundException: Unable to find explicit activity class {com.cursoandroid.aula05/com.cursoandroid.aula05>AboutActivity}; have you declared this activity in your AndroidManifest.xml?'. The exception is thrown from the main thread of the process 'com.cursoandroid.aula05' (PID: 28581). The stack trace includes the following methods: 'Instrumentation.checkStartActivityResult', 'Instrumentation.execStartActivity', 'Activity.startActivityForResult', 'FragmentActivity.startActivityForResult', 'Activity.startActivity', 'Activity.startActivity', 'Sudoku.onClick', 'View.performClick', 'View\$PerformClick.run', 'Handler.handleCallback', 'Handler.dispatchMessage', 'Looper.loop', 'ActivityThread.main', 'ZygoteInit\$MethodAndArgsCaller.run', and 'ZygoteInit.main'.

```
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 D/AndroidRuntime: Shutting down VM
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime: FATAL EXCEPTION: main
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime: Process: com.cursoandroid.aula05, PID: 28581
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime: android.content.ActivityNotFoundException: Unable to find explicit a
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.app.Instrumentation.checkStartActivityResult(Instrume
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.app.Instrumentation.execStartActivity(Instrumentation
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.app.Activity.startActivityForResult(Activity.java:391)
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.app.Activity.startActivityForResult(Activity.java:387)
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.support.v4.app.FragmentActivity.startActivityForResult
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.app.Activity.startActivity(Activity.java:4200)
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.app.Activity.startActivity(Activity.java:4168)
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at com.cursoandroid.aula05.Sudoku.onClick(Sudoku.java:32)
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.view.View.performClick(View.java:5198)
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.view.View$PerformClick.run(View.java:21147)
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.os.Handler.handleCallback(Handler.java:739)
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.os.Handler.dispatchMessage(Handler.java:95)
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.os.Looper.loop(Looper.java:148)
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at android.app.ActivityThread.main(ActivityThread.java:5417) <1
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(Zyg
04-11 10:56:11.840 28581-28581/com.cursoandroid.aula05 E/AndroidRuntime:     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:616)
```



Terminando o Jogo

- Implemente o tratador de eventos que seria ativado caso clicássemos sobre o botão “Sair”

```
public void onClick(View v) {  
    switch (v.getId()) {  
        case R.id.about_button:  
            Intent i = new Intent(this, AboutActivity.class);  
            startActivity(i);  
            break;  
        case R.id.new_button:  
            iniciarNovoJogo();  
            break;  
        case R.id.exit_button:  
            this.finish();  
            break;  
    }  
}
```



Dúvidas

