



Programação de Dispositivos Móveis

Guia para Aula Laboratorial 5

Licenciatura em Engenharia Informática

Licenciatura em Informática Web

Sumário

Implementação de simples aplicações que demonstram o funcionamento dos intents, os objetos que abstraem a forma como as várias componentes de uma aplicação Android™ comunicam, e que permitem a definição do fluxo dessas aplicações. São abordados intents implícitos e explícitos, bem como a transmissão de dados a partir da atividade que invoca o intento, para aquela que é invocada.

Pré-requisitos:

Algumas das tarefas enunciadas a seguir requerem o acesso a um sistema com o Android Studio e com o SDK Android™, bem como com a Gradle™ instalados ou, alternativamente, com permissões para instalação e configuração do IDE, *kit* e ferramenta. Serão suficientes permissões para criar diretórios e ficheiros num disco local e para configurar variáveis de sistema, nomeadamente a *path*. É necessário ter acesso a uma versão e imagem da plataforma Android™ ou a um dispositivo físico com o sistema operativo e com a opção de *debug* ativa. É igualmente necessário ter um compilador Java instalado.

1 Preliminares

Preliminaries

O guia laboratorial 2 elabora nos passos necessários a criação e compilação (*build*) de projetos de aplicações para a plataforma Android™ via linha de comandos. Esta abordagem, apesar de não comportar algumas das facilidades oferecidas por ambientes de desenvolvimento integrados, nomeadamente ambientes de edição da interface de utilizador *What You See Is What You Get* (WYSIWYG), permite conhecer em maior profundidade os detalhes de implementação de uma aplicação Android™, mas requer que, após instalação do Android Studio, se atualize e instalem as várias ferramentas do *Software Development Kit* (SDK) Android™¹. Depois do sistema estar devidamente configurado, 4 passos são suficientes para criar um projeto Android™, gerar o ficheiro *.apk* e instalar a aplicação num dispositivo (virtual ou real):

1. Inicializar o dispositivo móvel virtual ou ligar um real ao computador²;
2. Gerar o projeto através do Android Studio;

¹Ver <https://developer.android.com/studio/intro/update>.

²Se o dispositivo for real, tem de ter a opção de depuração ativada.

Programming of Mobile Devices

Guide for Laboratory Class 5

Degree in Computer Science and Engineering

Degree in Web Informatics

Summary

Implementation of simple applications that depict the functioning of intents, which are the objects that abstract the means by which several components of an Android™ application communicate, and that enable the definition of a flow in such applications. This laboratory guide handles implicit and explicit intents, as well as the transmission of data between the invoking and the invoked activities.

3. Compilar o projeto com a ferramenta Gradle™, emitindo o comando `$ gradlew assembleDebug` na raiz do projeto;

4. Instalar a aplicação com um comando semelhante a `$ adb install -r path\NomeApp-debug.apk`.

Tarefa 1 Task 1

Como já vem sendo habitual, a primeira tarefa consiste em iniciar um *Android Virtual Device* (AVD). Para isso, pode emitir o comando `$ emulator`, incluído na pasta *emulator* do SDK, e lançar um AVD. Caso não exista nenhum AVD configurado, crie um³. O ideal será um emulador de uma versão superior à 6.0 do Sistema Operativo (SO).

2 Intentos Implícitos

Implicit Intents

O objetivo desta aula é introduzir o estudo e praticar os objetos que, na plataforma Android™, permitem a comunicação e transição entre componentes de uma ou mais

³O guia laboratorial 1 contém uma breve discussão acerca deste assunto.

aplicações. Estes objetos são conhecidos por *Intents* (intentos) porque refletem a definição de uma operação que ainda precisa ser feita. Esta secção foca-se num dos 2 tipos de intents: os implícitos.

Tarefa 2 Task 2

Depois de se certificar que tem um dispositivo virtual a correr e que tem uma plataforma alvo superior à 4.2 disponível, crie um novo projeto Android™ através do Android Studio™ com as seguintes especificações:

- Nome da aplicação — *Implicit*;
- Domínio — *pmd.di.ubi.pt*;
- Sem suporte para C++ ou Kotlin;
- Deve ser um projeto para *smartphone* out *tablet*, mínimo API 21;
- Com uma Empty Activity chamada *Implicit*;
- Peça para gerar o ficheiro de *layout* (o nome do ficheiro de *layout* deve ser *activity_implicit.xml*);
- Retire qualquer suporte de retrocompatibilidade.

Note que os nomes sugeridos antes devem ser seguidos com rigor, já que deles depende, por vezes, o funcionamento bem sucedido da aplicação a ser desenvolvida.

Tarefa 3 Task 3

Edite o ficheiro *main.xml* de forma a incluir o botão especificado a seguir no *layout*:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:text="Send Message"
    android:layout_gravity="center"
    android:onClick="sendMessage"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Faça também por **eliminar** a etiqueta de texto que vem por defeito em projetos criados com o Android Studio™. Observe as propriedades do botão e responda às questões seguintes. **Q1.: Qual a aparência deste botão, em termos de altura?**

- ☐ Vai ocupar metade do ecrã.
- ☒ Vai ocupar o ecrã de cima a baixo.
- ☐ Vai ocupar apenas o suficiente para acolher o texto Send Message.

Q2.: Em termos de disposição, onde se vai situar este botão?

- ☐ À esquerda.
- ☒ Ao centro.
- ☐ À direita.
- ☐ Em cima.
- ☐ Ao meio.
- ☐ Em baixo.

Q3.: Só da análise do XML, diria que precisa de implementar algum método na atividade principal para que a aplicação funcione bem?

- ☐ Humm... diria que não.
- ☒ Humm... parece-me que sim, nomeadamente o método com protótipo

```
public void sendMessage(View v)
```

Tarefa 4 Task 4

Procure e abra para edição o ficheiro contendo o código fonte da atividade principal (*implicit*). Coloque o código Java incluído a seguir nesse ficheiro:

```
package pt.ubi.di.pmd.implicit;

import android.app.Activity;
import android.os.Bundle;
import android.view.*;
import android.content.Intent;
import android.widget.*;

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

        public void sendMessage(View v){
            Intent iSendMsg = new Intent(Intent.
                ACTION_SEND);
            iSendMsg.putExtra(Intent.EXTRA_TEXT, "You
                should type the message here!");
            iSendMsg.setType("text/plain");
            // If one wants to make sure that an
            // activity will resolve this intent,
            // the following comment should be removed
            // if ( iSendMsg.resolveActivity(
                getPackageManager()) != null )
                startActivity(iSendMsg);
        }
    }
}
```

No código incluído em cima, são importados vários pacotes, nomeadamente o *android.widget.**. **Q4.: Este *import* é mesmo necessário?**

- ☐ Sim, é mesmo necessário.
- ☒ Não percebo por que é que lá foi incluído.

Q5.: E o *import* de *android.content.Intent*, é necessário?

- ☒ Indubitavelmente.
- ☐ Não percebo por que é que lá foi incluído.

Q6.: Em que pacote/classe é que o método *setContentView()* está definido?

- ☒ *import android.app.Activity*;
- ☐ *import android.os.Bundle*;
- ☐ *import android.view.**;
- ☐ *import android.content.Intent*;
- ☐ *import android.widget.**;

Antes de evoluir para a próxima tarefa, analise o código

e responda ao seguinte: **Q7.: O que é que esta aplicação supostamente deve fazer?**

- ☒ Enviar uma SMS.
- ☒ Enviar um e-Mail.
- ☐ Abrir o *browser*.
- ☒ Enviar uma mensagem para o Facebook.
- ☐ Abrir a calculadora.
- ☐ Nenhuma das anteriores.
- ☐ Todas as anteriores, incluindo a que diz: nenhuma das anteriores.

Tarefa 5 Task 5

Compile, instale e teste o projeto que criou. Confirme se o aspeto da aplicação se coaduna com o que havia respondido anteriormente relativamente ao botão.

Q8.: Por que é que o *intent* que foi criado no âmbito desta parte do guia laboratorial é conhecido por *intento implícito*?

- ☐ Porque a ação que vai produzir é desconhecida.
- ☐ Porque a ação que vai ser feita foi totalmente definida.
- ☒ Porque a atividade que vai tratar da ação especificada não é conhecida à partida.
- ☐ Porque a atividade que vai tratar da ação especificada é conhecida e definida no código.
- ☐ Porque a *intent* tem um parâmetro adicional configurado com o método `putExtra()`.

Q9.: Só para que fique registado, o que é que acontece quando pressiona o botão *Send Message*?

- ☐ É aberta uma calculadora (que estranho!).
- ☐ É mostrada uma caixa de diálogo onde é dada a hipótese de escolher qual a aplicação para onde o fluxo vai evoluir. A atividade onde estava fica desfocada no fundo.
- ☐ É aberta a aplicação que envia SMSs.
- ☐ É aberta a aplicação que envia e-mails.
- ☐ É aberta a aplicação que mostra os mapas.
- ☐ A aplicação é terminada e um erro é mostrado no ecrã.

3 Intentos Explícitos

Explicit Intents

Esta parte do guia laboratorial é focada em intentos explícitos.

Tarefa 6 Task 6

Crie um novo projeto Android™ através do Android Studio™ com as seguintes especificações:

- Nome da aplicação — `Explicit1`;
- Domínio — `pmd.di.ubi.pt`;

- Sem suporte para C++ ou Kotlin;
- Deve ser um projeto para *smartphone* out *tablet*, mínimo API 21;
- Com uma `Empty Activity` chamada `Explicit1`;
- Peça para gerar o ficheiro de *layout* (o nome do ficheiro de *layout* deve ser `activity_explicit1.xml`);
- Retire qualquer suporte de retrocompatibilidade.

Note que os nomes sugeridos antes devem ser seguidos com rigor, já que deles depende, por vezes, o funcionamento bem sucedido da aplicação a ser desenvolvida.

Tarefa 7 Task 7

Modifique o *layout* da aplicação que é criado por defeito de modo a conter o botão seguinte:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Start Calculator"
    android:onClick="startCalculator"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

Faça também por **eliminar** a etiqueta de texto que vem por defeito em projetos criados com o Android Studio™.

A definição do botão incluída parece sugerir que o botão vai estar centrado na interface gerada pela atividade.

Q10.: Esta *assunção* parece-lhe válida?

- ☐ Assim à primeira vista, não vejo porque não, mas já sei que o Prof. cá mete estas questões só para nos dar a volta.
- ☒ Assim à primeira vista, não vejo porque não.
- ☐ Não. Esta *assunção* é errada.

Justifique a sua resposta anterior, para quando depois for estudar, cá estar tudo explícito:

Tarefa 8 Task 8

Dada a definição anterior do botão, deve ser necessária a implementação de uma função de tratamento do evento `onClick`. Procure e abra para edição o ficheiro que implementa a atividade principal (`explicit1.java`) e declare essa função. **Q11.: Só mesmo por curiosidade, como se chama a classe que implementa a atividade principal?**

- ☐ Não sei, terei de abrir primeiro o ficheiro `.java`.

- ☐ Chama-se `main` e estende a classe `Activity`.
- ☒ Então: tem de se chamar, obrigatoriamente, `Explicit1`.
- ☐ Chama-se pelo nome.

No corpo dessa função, coloque o seguinte:

```
Intent iCalc = new Intent();
iCalc.setAction(Intent.ACTION_MAIN);
iCalc.setComponent(new ComponentName("com.android.
    calculator2", "com.android.calculator2.Calculator
"));
startActivity(iCalc);
```

Dado o código incluído antes, e comparando com os *imports* que o ficheiro que implementa a atividade principal já contém, selecione, em baixo, os nomes dos pacotes que ainda precisa incluir no ficheiro `explicit1.java`:

- ☐ `import android.app.Activity;`
- ☐ `import android.os.Bundle;`
- ☒ `import android.view.*;`
- ☒ `import android.content.Intent;`
- ☒ `import android.content.ComponentName;`

Tarefa 9 Task 9

Compile, instale e teste a aplicação criada anteriormente.

Q12.: Só para que fique registado, o que é que acontece quando pressiona o botão **Start Calculator**?

- ☒ É mesmo aberta uma calculadora! E esta, hein?
- ☐ É mostrada uma caixa de diálogo onde é dada a hipótese de escolher qual a aplicação para onde o fluxo vai evoluir. A atividade onde estava fica desfocada no fundo.
- ☐ É aberta a aplicação que envia e-mails (que estranho).
- ☐ É aberto um *browser* com dois *tabs* que imediatamente são direcionados para os dois links especificados no código.
- ☐ A aplicação é terminada e um erro é mostrado no ecrã.

No código incluído em cima está uma linha de código com o seguinte trecho `ComponentName("com.android.calculator2",`

Q13.: O que é que significam as duas **strings** neste trecho?

- ☐ São dois *links* que o Android™ abre no *browser* configurado por defeito.
- ☒ A primeira é o nome do pacote da calculadora que vem de fábrica com o Android™; a segunda é o nome da atividade principal da calculadora.
- ☐ A primeira é o nome da atividade principal da calculadora; a segunda é o nome do pacote da calculadora que vem de fábrica com o Android™.
- ☒ Estas duas strings definem sem sombra de dúvida o componente para onde a intenção aponta.

Q14.: Quando usada como um adjetivo, como é o caso em intento explícito, a palavra **explícito** deve ou não ser acentuada?

- ☒ Deve ser.
- ☐ Não deve ser.

Procure no dicionário (e.g., <http://www.priberam.pt/DLP0>) a definição da palavra antes discutida, e inclua-a no espaço seguinte:

Q15.: A atividade desenvolvida envia dados para a atividade recetora do intento?

- ☒ Não.
- ☐ Sim, nomeadamente: _____

Tarefa 10 Task 10

Até este ponto, todas as aplicações desenvolvidas eram constituídas por apenas uma atividade. A parte restante deste guia sugere a criação de uma atividade adicional, que irá inclusive receber dados da atividade que a invocar. Pode construir um projeto de raiz ou simplesmente alterar o anterior, sendo a descrição seguinte orientada para a alteração daquele que acabou de ser feito.

Comece por criar um novo ficheiro chamado `Activity2.java` dentro da diretoria `src/main/java/pt/ubi/di/pmd/explicit1`. Este ficheiro terá o código que implementa esta segunda atividade. Abra o ficheiro com um editor de texto (e.g., `notepad++`), e considere colocar no seu conteúdo uma implementação semelhante à que se segue:

```
package pt.ubi.di.pmd.explicit1;

import android.app.Activity;
import android.os.Bundle;
import android.view.*;
import android.widget.*;
import android.content.Intent;

public class Activity2 extends Activity {
    TextView oTV;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.main2);
            oTV = (TextView) findViewById(R.id.textview1);
        };

        Intent iCameFromActivity1 = getIntent();
        oTV.setText(iCameFromActivity1.
            getStringExtra("string1"));
    }
    public void endActivity(View v){
        finish();
    }
}
```

```
}
```

Q16.: O que é que pode concluir da observação do código?

- ☐ Pouca coisa.
- ☐ Vá, uma coisa ou outra, mas nada de relevante.
- ☐ Há ali uns detalhes...
- ☒ Que esta atividade parece inicializar um objeto `Intent` através do método `getIntent()` e não a partir de um construtor.
- ☒ Que é instanciada uma `TextView` e depois preenchida com algo que vem dentro da `Intent`, nomeadamente numa variável geral com designação `string1`.
- ☒ Que, já agora, parece relevante referir que parece haver ali um método que não é usado, provavelmente por ser uma função *callback* para um evento *onClick* definido no XML.
- ☒ Que, ainda neste contexto, parece interessante mencionar que há ali um outro método que nunca tinha visto (`finish()`). Vou já ver o que faz...

Q17.: Qual o ficheiro XML que define o *layout* desta segunda atividade?

- ☐ E como é que eu vou saber isso?
- ☐ `main.xml`
- ☐ `main2.xml`
- ☐ `layout.xml`
- ☒ É um ficheiro que ainda não existe.

Tarefa 11 Task 11

Note que depois de definir uma nova atividade, é necessário criar o seu *layout* (i.e., um ficheiro XML) e declará-la no `AndroidManifest.xml`. Esta tarefa consiste, portanto, em adicionar **no local correto** a seguinte linha no ficheiro mencionado:

```
<activity android:name="Activity2" />
```

Tarefa 12 Task 12

Crie o ficheiro XML `main2.xml` na pasta `res/layout`

e faça por este conter os elementos incluídos a seguir:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=". Activity2">

    <TextView

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        app:layout_constraintRight_toRightOf="parent
```

```
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="End Activity"
    android:onClick="endActivity"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@id/textview1" />

</android.support.constraint.ConstraintLayout>
```

Q18.: Da análise do código que antecede o XML, falta algum atributo em algum elemento incluído em cima?

- ☒ Sim, falta... e vou já tratar disso.
- ☐ Parece-me que falta, mas não sei o que fazer.
- ☐ Não falta nada.

Tarefa 13 Task 13

De seguida, há que modificar a atividade principal de maneira a que aponte para a que criou recentemente. Comece pelo ficheiro `activity_explicit1.xml`, alterando-o de forma a conseguir que a atividade principal mostre mais um botão com o texto `Start Second Activity`. Quando clicado, este botão deve despoletar a rotina `startActivity()`. Provavelmente, será necessário adicionar algo semelhante ao seguinte (**não se esqueça de definir o `id` do botão anterior...**):

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Start Second Activity"
    android:onClick="startActivity"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@id/button1" />
```


Tarefa 14 *Task 14*

Passa agora para o ficheiro `explicit1.java`, **adicionando** o método seguinte no **local correto**:

```
public void startActivity(View v){
    Intent iActivity =
        new Intent(this, Activity2.class);

    iActivity.putExtra("string1",
        "If you were able to do this, you are FABULOUS!");
    startActivity(iActivity);
}
```

Q19.: Como é que se instancia um intento para uma atividade local?

- ☐ Através do construtor `Intent(Context, Class)`, que aceita o contexto da classe atual e o nome do componente destino.
- ☐ Através do método `putExtra("Class", "Target")`, em que a primeira string define o nome da classe atual e a segunda o nome da componente destino.
- ☐ Através do método `startActivity(Intent)`.

Tarefa 15 *Task 15*

Finalmente, compile, instale e teste a aplicação. **Q20.: Funcionou?**

- ☐ Como não podia deixar de ser.
- ☐ Não percebo o porquê de tanto alarido.

Use o espaço seguinte para relatar o funcionamento da aplicação, discutindo, por exemplo, a proveniência da mensagem que é mostrada na segunda atividade.
