

ANDROID INTENTS



“640K é mais memória do que qualquer pessoa vai precisar.” (Bill Gates, 1981)

- **INTENTS**
- **COMO UTILIZAR INTENTS?**
- **INTENT FILTERS**
- **THREADS, HANDLERS E RUNNABLE**



INTENTS

- São mensagens, enviadas/recebidas para/do sistema operacional Android, para ativar três dos principais componentes do núcleo de uma aplicação: *Activities*, *Services* e *Broadcast Receivers*;
- É um pacote(*bundle*) de informações que interessam ao componente que receberá a mensagem;
- Tipo de objeto da classe **android.content.Intent**;
- Algumas das possíveis construções de uma Intent:
 - **Intent()**: cria uma intent vazia;
 - **Intent(String action, Uri uri)**: cria uma intent para uma determinada ação e URI;
 - **Intent(Context packageContext, Class<?> cls)**: Cria uma intent para um componente específico.



- Uma intent contém as seguintes informações:
 - **ComponentName:** O nome do componente que deverá manipular a intent;
 - **Action:** Uma string indicando a ação a ser executada;
 - **Data:** A URI(*Uniform Resource Identifier*) e o MIME(*Multipurpose Internet Mail Extensions*) dos dados a serem tratados;
 - **Category:** Uma string contendo informações adicionais sobre o tipo de componente a manipular a intent;
 - **Extras:** Informação adicional, no formato chave-valor(par), a ser enviada para o componente manipulador da Intent;
 - **Flags:** Normalmente utilizados para instruir o Android de como carregar a activity.



INTENTS - ACTIONs

- Ações que podem ser definidas no Intent:
 - Ação a ser executada;
 - Ação, em caso de broadcast intent, a ser reportada.

CONSTANTE	COMPONENTE ALVO	AÇÃO
ACTION_CALL	activity	Inicia uma chamada de telefone.
ACTION_EDIT	activity	Exibe dados para edição pelo usuário.
ACTION_MAIN	activity	Startup de uma activity principal sem entrada e saída de dados.
ACTION_SYNC	activity	Sincronização de dados servidor para dispositivo.
ACTION_BATTERY_LOW	broadcast receiver	Aviso de que a bateria tem pouca carga.
ACTION_HEADSET_PLUG	broadcast receiver	Um fone de ouvido foi plugado ou desplugado.
ACTION_SCREEN_ON	broadcast receiver	A tela foi ativada/ligada.
ACTION_TIMEZONE_CHANGED	broadcast receiver	A configuração de timezone foi alterada.

INTENTS - CATEGORYs

- Algumas constantes de Categorys prontas:

CONSTANTE	SIGNIFICADO
CATEGORY_BROWSABLE	A atividade alvo pode, de forma segura, ser invocada pelo browser para exibir os dados referenciados por um link – por exemplo, uma imagem ou e-mail.
CATEGORY_GADGET	A atividade pode ser embarcada dentro de outra atividade que hospeda gadgets.
CATEGORY_HOME	A atividade apresenta a tela inicial do Android.
CATEGORY_LAUNCHER	A atividade pode ser a atividade inicial de uma tarefa e está listada no topo do carregador de aplicações.
CATEGORY_PREFERENCE	A atividade alvo é um painel de preferências.



INTENTS - RESOLUÇÃO

- A resolução de Intents pelo Android, acontece de duas formas:
- **Explícita:** Intents explícitas designam seus alvos pelo nome do componente. Este tipo intent é utilizada tipicamente para passagem de mensagens internas em uma aplicação;
- **Implícita:** Intents que não nomeiam seus alvos. São utilizadas para ativar recursos de outras aplicações.



COMO PODEMOS UTILIZAR INTENTS?

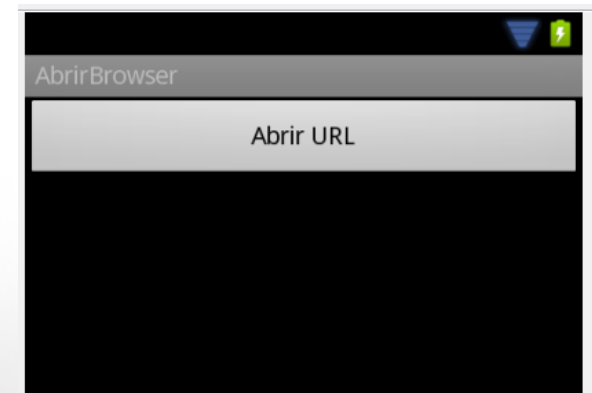
- Abrir novas telas:
startActivity(intent);
- Ligar para algum número de telefone;
- Abrir o browser com alguma url;
- Exibir endereços, localização ou rota no Google Maps;
- Abrir o Google Play para fazer download de algum aplicativo;



EXEMPLOS – ABRINDO O BROWSER

```
11 public class AbrirBrowserActivity extends Activity {
12     /** Called when the activity is first created. */
13     private final String urlUnoesc = "http://www.unoesc.edu.br";
14
15     @Override
16     public void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.main);
19
20         Button btnAbrirURL = (Button)findViewById(R.id.btnAbrirURL);
21         btnAbrirURL.setOnClickListener(new OnClickListener() {
22             public void onClick(View v) {
23                 // criando um objeto do tipo URI para abrir no browser
24                 Uri uri = Uri.parse(urlUnoesc);
25                 Intent it = new Intent(Intent.ACTION_VIEW, uri);
26                 startActivity(it);
27             }
28         });
29     }
30 }
```

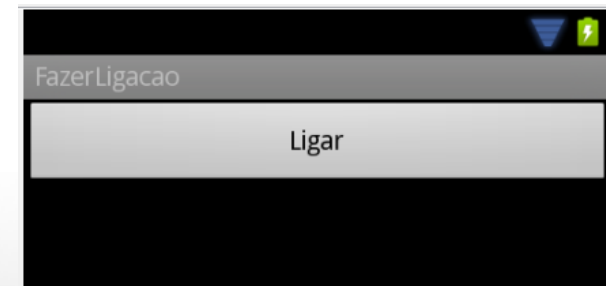
Criando uma intent
com um ACTION_VIEW
para uma URI contendo
uma URL.



EXEMPLOS – FAZENDO LIGAÇÕES

```
11 public class FazerLigacaoActivity extends Activity {
12     /** Called when the activity is first created. */
13     private final String telefone = "tel:0140498888888";
14
15     @Override
16     public void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.main);
19
20         Button btnLigar = (Button)findViewById(R.id.btnLigar);
21         btnLigar.setOnClickListener(new OnClickListener() {
22             public void onClick(View v) {
23                 // criando um objeto do tipo URI para realizar ligação
24                 Uri uri = Uri.parse(telefone);
25                 Intent it = new Intent(Intent.ACTION_VIEW, uri);
26                 startActivity(it);
27             }
28         });
29     }
30 }
```

Criando uma intent
com um ACTION_VIEW
para uma URI de
telefone.



EXEMPLOS – VISUALIZANDO UM CONTATO DA AGENDA

```
11 public class VisualizarContatoActivity extends Activity {
12     /** Called when the activity is first created. */
13     private final String contato = "content://com.android.contacts/contacts/1";
14
15     @Override
16     public void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.main);
19
20         Button btnVisualizar = (Button)findViewById(R.id.btnVisualizar);
21         btnVisualizar.setOnClickListener(new OnClickListener() {
22             public void onClick(View v) {
23                 // criando um objeto do tipo URI para realizar ligação
24                 Uri uri = Uri.parse(contato);
25                 Intent it = new Intent(Intent.ACTION_VIEW, uri);
26                 startActivity(it);
27             }
28         });
29     }
30 }
```

Criando uma intent com um ACTION_VIEW para uma URI de provedor de conteúdo de contatos.



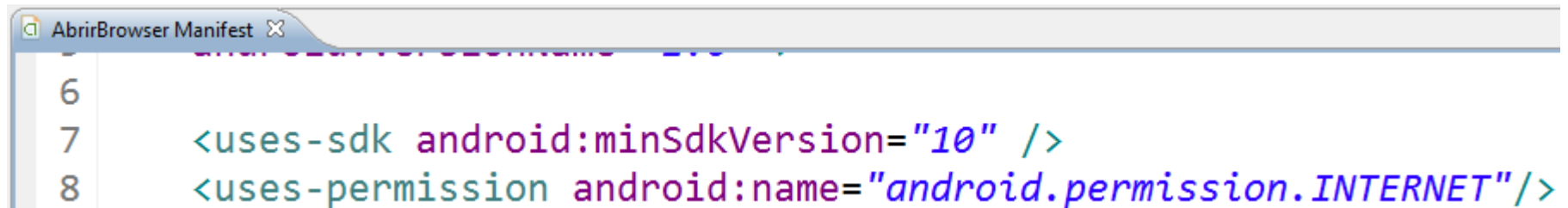
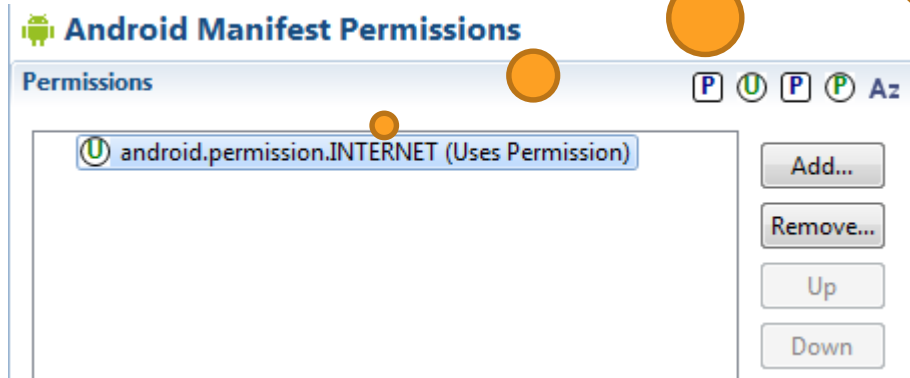
PRECISAMOS DE PERMISSÃO – ANDROIDMANIFEST.XML

- Se o Android não permitir executar as tarefas solicitadas, *peça permissão ...*
- Isso pode ser feito pelo **AndroidManifest.xml**



PERMISSÕES – ANDROIDMANIFEST.XML

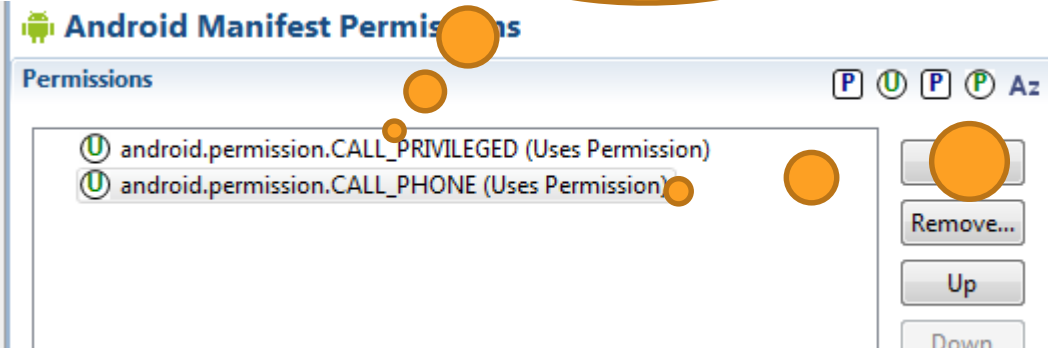
Permite que a aplicação crie/abra sockets de rede.



PERMISSÕES – ANDROIDMANIFEST.XML

Permite que a aplicação inicie uma chamada telefônica, incluindo chamadas de emergência, sem passar pela interface do discador para confirmar a ligação

Permite que a aplicação inicie uma chamada telefônica sem passar pela interface do discador para confirmar a ligação



```
FazerLigacao Manifest
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="br.edu.android.fazerligacao"
4     android:versionCode="1"
5     android:versionName="1.0" >
6
7     <uses-sdk android:minSdkVersion="10" />
8     <uses-permission android:name="android.permission.CALL_PRIVILEGED"/>
9     <uses-permission android:name="android.permission.CALL_PHONE"/>
```



PERMISSÕES – ANDROIDMANIFEST.XML

Permite ler os dados dos contatos do usuário.



The screenshot displays the Android Studio interface. At the top, the 'Android Manifest Permissions' panel is visible, showing a list of permissions. The first permission listed is 'android.permission.READ_CONTACTS (Uses Permission)', which is highlighted with an orange circle. To the right of this list are three buttons: 'Add...', 'Remove...', and 'Up'. Below the permissions panel, the 'VisualizarContato Manifest' editor is open, showing the XML code for the AndroidManifest.xml file. The code is as follows:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="br.edu.android.visualizarcontato"
4     android:versionCode="1"
5     android:versionName="1.0" >
6
7     <uses-sdk android:minSdkVersion="10" />
8     <uses-permission android:name="android.permission.READ_CONTACTS"/>
```

EXECUTANDO UMA ACTIVITY

- Para abrir uma nova tela/activity também utiliza-se uma Intent
 - Cria-se o objeto intent passando a atividade que está chamando e depois a atividade que será chamada

```
Intent itLogin = new Intent(SplashActivity.this,  
    LoginActivity.class);  
startActivity(itLogin);
```



EXECUTANDO UMA ACTIVITY – PASSANDO PARÂMETROS

- Utiliza-se um objeto do tipo **android.os.Bundle**
- Adiciona-se o parâmetro combinando nome e valor
- Para adicionar ao Intent pode ser utilizado método **Intent.putExtras** ou **Intent.putExtra(valor, tipo de dado)**

```
Intent it = new Intent(this, ActivityDois.class);  
it.putExtra("brasil", checkbox1.getText());  
it.putExtra("argentina", checkbox2.getText());  
it.putExtra("futebol", radiobutton1.getText());  
it.putExtra("tango", radiobutton2.getText());  
startActivity(it);
```



EXECUTANDO UMA ACTIVITY – PASSANDO PARÂMETROS

- Para recuperar os parâmetros(ActivityDois.class)

```
Intent it = getIntent();  
  
Bundle extras = getIntent().getExtras();  
  
t3.setText(extras.getString("brasil"));  
t1.setText(extras.getString("argentina"));
```



INTENT FILTERS

- Para informar ao sistema quais intents(atividades, serviços ou receptores broadcast) implícitas ele pode manipular podemos definir um ou mais filtros de intent;
- Os intent filters servem para filtrar intents não desejadas;
- Para intents explícitas os intent-filters não são consultados;
- Um intent filter é uma instância de **android.content.IntentFilter**.



INTENT FILTERS

- De forma geral os intent filters são sempre configurados no arquivo `AndroidManifest.xml`;
- Para declarar um intent filter utiliza-se elementos do tipo **<intent-filter>**;
- Um intent filter não pode ser utilizado como recurso de segurança.



INTENT FILTERS - EXEMPLOS

- **Action Test**

```
<intent-filter . . . >
    <action android:name="com.example.project.SHOW_CURRENT" />
    <action android:name="com.example.project.SHOW_RECENT" />
    <action android:name="com.example.project.SHOW_PENDING" />
    . . .
</intent-filter>
```



INTENT FILTERS - EXEMPLOS

- Category Test

```
<intent-filter . . . >
    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />
    . . .
</intent-filter>
```



INTENT FILTERS - EXEMPLOS

- Data test

```
<intent-filter . . . >
    <data android:mimeType="video/mpeg" android:scheme="http" . . . />
    <data android:mimeType="audio/mpeg" android:scheme="http" . . . />
    . . .
</intent-filter>
```



INTENT FILTERS - EXEMPLOS

- Iniciando uma aplicação

```
<intent-filter . . . >
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

Indica que a activity deve ser a primeira a ser carregada.

Indica que o aplicativo deve aparecer no carregador de aplicativos do Android.



Diário de
Classe Mó...



THREADS/RUNNABLES

- Threads são pequenos processos que podem ser carregados/executados no Android;
- O Android executa a aplicação principal em uma Thread, chamada de Main;
- Para evitar realizar operações assíncronas e não bloqueantes, como: acesso a recursos na internet, atualização de barra de progresso; é interessante utilizar Threads.



THREADS/RUNNABLES - EXEMPLO

- Para disparar uma Thread podemos utilizar a seguinte lógica:

```
Thread thread = new Thread(new Runnable() {  
    public void run() {  
        // escreva aqui o que será executado ...  
    }  
});  
// iniciando a Thread  
thread.start();
```
- A interface **Runnable** representa um comando que pode ser executado;
- O método **run()** é o método que deve ser sobrescrito e que será executado ao iniciar a Thread.



ANDROID HANDLERS

- Um handler do Android é um objeto que permite enviar uma mensagem(processo ou runnable) associados a uma Thread;
- Handlers são objetos do tipo **android.os.Handler**;
- Handlers podem ser utilizados de 2 formas: agendar runnables ou mensagens para serem executadas em determinado ponto no futuro; ou para enfileirar uma ação a ser executada em uma Thread diferente.



ANDROID HANDLERS - EXEMPLO

- Exemplo de uso de um Handler para carregar uma tela splash por 5 segundos:

```
Handler h = new Handler();  
h.postDelayed(new Runnable() {  
    public void run() {  
        Intent itLogin = new Intent(SplashActivity.this,  
                                     LoginActivity.class);  
        startActivity(itLogin);  
        finish();  
    }  
}, 5000);
```



REFERÊNCIAS

- **Intents and Intent Filters.** Disponível em: <http://developer.android.com/guide/topics/intents/intent-filters.html>. Acesso em: 12 jul. 2012.
- **LECHETA, Ricardo R.** Google android: aprenda a criar aplicações para dispositivos móveis com o android SDK. 2. ed., rev. e ampl. São Paulo: Novatec, 2010. 608 p. ISBN 9788575222447.
- **MURPHY, Mark L.** Beginning android. New York, USA: Apress, 2009. xxii, 361 p. ISBN 9781430224198.
- **Processes and Threads.** Disponível em: <http://developer.android.com/guide/components/processes-and-threads.html>. Acesso em: 27 jul. 2012.

