

# Desenvolvimento de Apps para Mobile

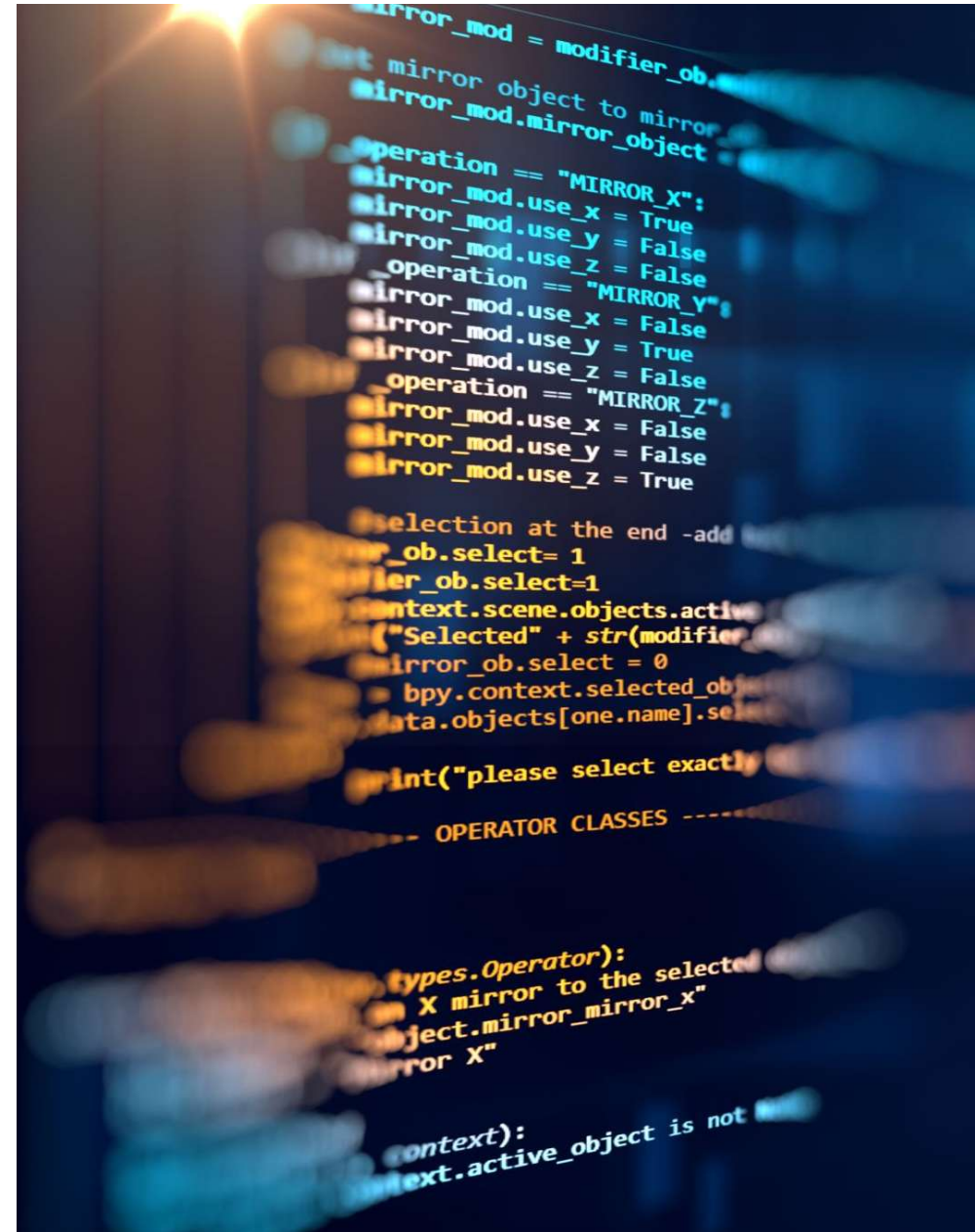
Fabrício Tonetto Londero

Aula 05 - Intents



# Intent

- Uma `Intent` representa uma **mensagem da aplicação** para o **sistema operacional**, solicitando que algo seja realizado;
- Cabe ao **sistema operacional** interpretar essa mensagem e tomar as providências necessárias:
  - Abrir uma aplicação, fazer uma ligação para um determinado número, abrir o navegador em uma determinada página da internet, enviar um email, etc.
- O principal uso é para **navegação e execução de novas activities**.



## Abrir navegador e fazer chamadas - AndroidManifest.xml

```
<uses-feature  
    android:name="android.hardware.telephony"  
    android:required="false" />  
  
<uses-permission android:name="android.permission.CALL_PHONE" />  
<uses-permission android:name="android.permission.INTERNET" />
```

# Abrindo o browser

- Abrir um browser através do Android é uma tarefa bem simples através da classe Intent;
- Para isso, utilizaremos uma ação `Intent.ACTION_VIEW`:

```
Uri uri = Uri.parse("http://www.google.com.br");  
Intent it = new Intent(Intent.ACTION_VIEW, uri);  
startActivity(it);
```

# Abrindo o browser

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Digite o endereço do
site e pressione o botão"
/>
<EditText
    android:id="@+id/campoEndereco"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>
<Button
    android:id="@+id/botaoOk"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=" Visualizar"
/>
```

# Abrindo o browser

```
public void onCreate(Bundle bundle) {

    super.onCreate(bundle);
    setContentView(R.layout.main);

    final Button button = (Button) findViewById(R.id.botaoOk);
    button.setOnClickListener(new Button.OnClickListener() {

        public void onClick(View v) {
            EditText campoEndereco = (EditText)
            findViewById(R.id.campoEndereco);

            String endereco = campoEndereco.getText().toString();
            Uri uri = Uri.parse(endereco);

            Intent intent = new Intent(Intent.ACTION_VIEW, uri);
            startActivity(intent);
        }

    });
}
```

# Fazendo uma ligação

- Em algumas **aplicações** pode ser preciso **automatizar a ligação** para um determinado número de telefone.
- Para isso, basta criar uma mensagem com a ação `Intent.ACTION_CALL`:

```
...  
Uri uri = Uri.parse("tel:99999999");  
Intent it = new Intent(Intent.ACTION_CALL, uri);  
startActivity(it);  
...
```

# Fazendo uma ligação

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Digite o telefone
e pressione o botão"
/>
<EditText
    android:id="@+id/campoTelefone"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>
<Button
    android:id="@+id/botaoOk"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Telefonar"
/>
```



# Fazendo uma ligação

```
public void onCreate(Bundle bundle) {

    super.onCreate(bundle);
    setContentView(R.layout.main);

    final Button button = (Button) findViewById(R.id.botaoOk);
    button.setOnClickListener(new Button.OnClickListener() {

        public void onClick(View v) {

            EditText campoTelefone = (EditText) findViewById(R.id.
campoTelefone);

            String endereco = campoTelefone.getText().toString();
            Uri uri = Uri.parse("tel:"+telefone);
            Intent intent = new Intent(Intent.ACTION_CALL, uri);
            startActivity(intent);

        }

    });
}
```

# Fazendo uma ligação

- Para o exemplo anterior também é necessário solicitar permissão, mas desta vez para chamadas telefônicas;
- Para isso, abra o arquivo `AndroidManifest.xml`, pressione o botão `Add` e adicione a permissão para `android.permission.CALL_PHONE`

# Visualização de contatos

- Para exibir as informações de um contato cadastrado na agenda, utilizaremos a ação `Intent.ACTION_VIEW`;
- O padrão para identificar um contato é `content://com.android.contacts/contacts/1`, onde o número 1 indica o id do contato a ser visualizado:

```
...  
Uri uri = Uri.parse("content://com.android.contacts/contacts/1");  
Intent it = new Intent(Intent.ACTION_VIEW, uri);  
startActivity(it);  
...
```

# Visualização de contatos

- Para o exemplo a seguir também é necessário solicitar permissão, mas desta vez para chamadas visualização de contatos;
- Para isso, abra o arquivo `AndroidManifest.xml` e adicione a permissão para `android.permission.READ_CONTACTS`

# Visualização de contatos

```
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Digite o ID do contato
e pressione o botão"
/>
<EditText
    android:id="@+id/campoID"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>
<Button
    android:id="@+id/botaoOk"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Visualizar"
/>
```

# Visualização de contatos

```
public void onCreate(Bundle bundle) {  
    super.onCreate(bundle);  
    setContentView(R.layout.main);  
    final Button button = (Button) findViewById(R.id.botaoOk);  
    button.setOnClickListener(new Button.OnClickListener() {  
  
        public void onClick(View v) {  
            EditText campoID = (EditText) findViewById(R.id.campoID);  
            String id = campoTelefone.getText().toString();  
            Uri uri = Uri.parse("content://com.android.contacts/contacts/"+id);  
            Intent intent = new Intent(Intent.ACTION_VIEW, uri);  
            startActivity(intent);  
        }  
    });  
}
```

# Visualização de todos os contatos

- Para exibir uma lista de todos os contatos cadastrados na agenda, utilizaremos a ação `Intent.ACTION_PICK`;
- O padrão para visualizar a lista de contatos é `content://com.android.contacts/contacts/`

```
...  
Uri uri = Uri.parse("content://com.android.contacts/contacts/");  
Intent it = new Intent(Intent.ACTION_PICK, uri);  
startActivity(it);  
...
```

# Visualização de todos os contatos

```
public void onCreate(Bundle bundle)
{
    super.onCreate(bundle);
    setContentView(R.layout.main);
    final Button button = (Button) findViewById(R.id.botaoOk);
    button.setOnClickListener(new Button.OnClickListener() {

        public void onClick(View v) {
            Uri uri = Uri.parse("content://com.android.contacts/contacts/");
            Intent intent = new Intent(Intent.ACTION_PICK, uri);
            startActivity(intent);
        }

    });
}
```



# Outros exemplos

- Solicitar ao Google Maps que faça uma busca por um endereço:

```
Uri uriGeo = Uri.parse("geo:0,0?q=Floriano+Peixoto,Curitiba");  
Intent it = new Intent(Intent.ACTION_VIEW, uriGeo);  
startActivity(it);
```

- Solicitar ao Google Maps que exiba o mapa em uma coordenada específica:

```
Uri uriGeo = Uri.parse("geo:-25.443195,-49.280977");  
Intent it = new Intent(Intent.ACTION_VIEW, uriGeo);  
startActivity(it);
```

# Outros exemplos

- Solicitar ao Google Maps que exiba a rota entre duas coordenadas:

```
String partida = "-25.443195,-49.290877";
String destino = "-25.442207,-49.278403";
String url =
    "http://maps.google.com/maps?f=d&saddr="+partida+"&daddr="+destino+"&hl=pt";
Uri uriGeo = Uri.parse(url);
Intent it = new Intent(Intent.ACTION_VIEW, uriGeo);
startActivity(it);
```

- Tocar um arquivo mp3 da web:

```
Uri link = Uri.parse(" https://cdn.freesound.org/previews/130/130809_1648170-1q.mp3");
Intent it = new Intent(Intent.ACTION_VIEW, link);
it.setType("audio/*");
startActivity(it);
```

# Exercícios

- 1) Faça uma aplicação que de 3 opções ao usuário de busca no Google Maps. A primeira baseada no endereço digitado, a segunda baseada na coordenada específica e a terceira para gerar a rota entre dois endereços.
- 2) Faça uma pesquisa e descubra como utilizar uma Intent para enviar um SMS e para enviar um email através de uma aplicação para Android.