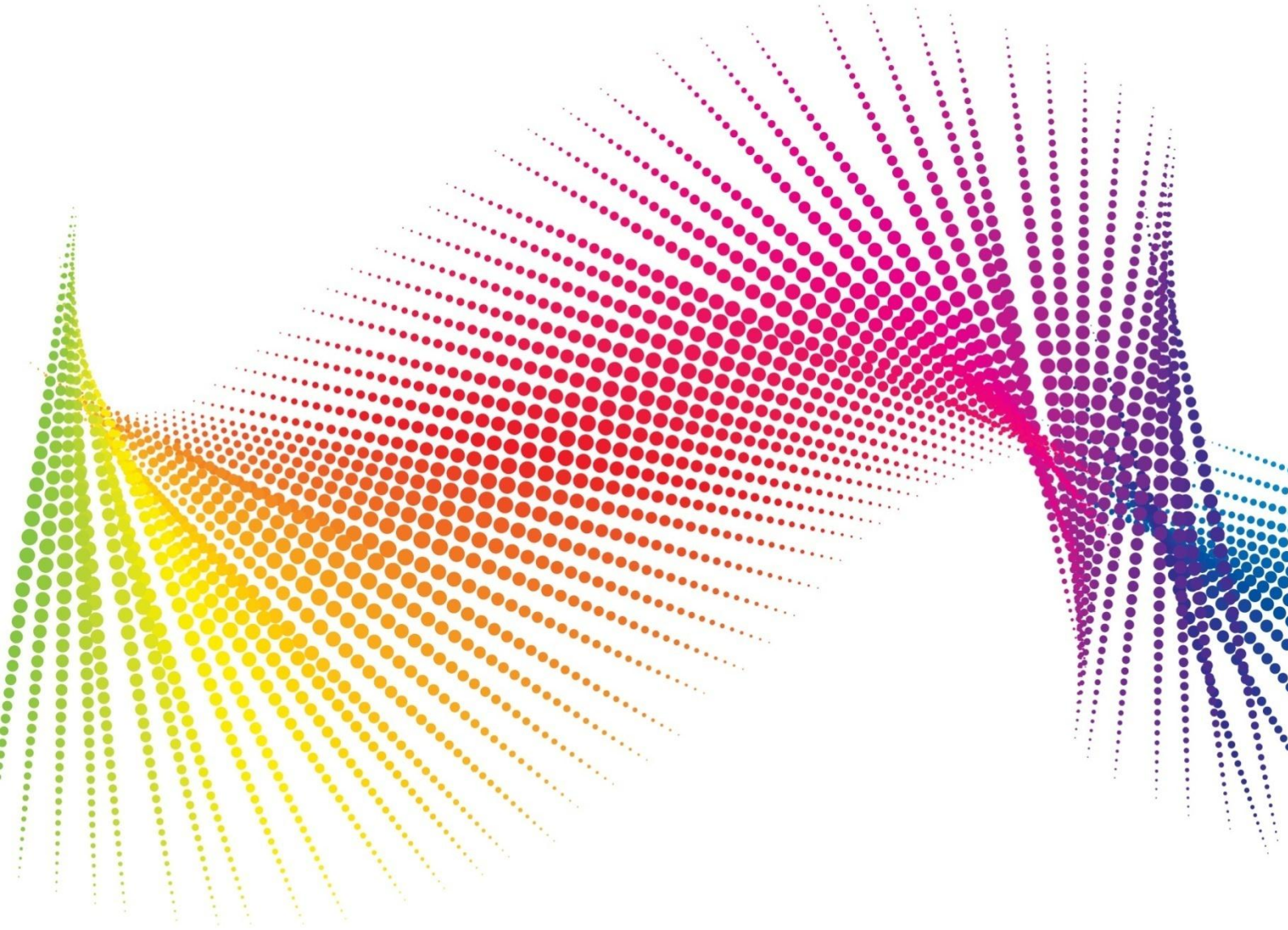
Computação Móvel

Aula 08



Este material é parte integrante da disciplina oferecida pela UNINOVE.
O acesso às atividades, conteúdos multimídia e interativo, encontros virtuais, fóruns de discussão e a comunicação com o professor devem ser feitos diretamente no ambiente virtual de aprendizagem UNINOVE.

Uso consciente do papel.
Cause boa impressão, imprima menos.

Aula 08: Recursos de imagens e strings no Android, localização da aplicação

Objetivo: Demonstrar o funcionamento dos recursos de alocação e utilização de imagens no Android, assim como outro recurso muito interessante que permite a manipulação de strings, possibilitando, com isso, a criação de aplicação que podem selecionar o idioma no qual ela será apresentada.

Recursos de imagens

Os recursos de imagens são representados pela classe drawable que é um recurso geral para gráficos que pode desenhar na tela e que também pode recuperar imagens com APIs comogetDrawable(int) ou aplicar em outro recurso de XML utilizando-se o atributo drawable ou icon. Existem vários tipos diferentes de drawables:

- **Arquivo de bitmap –** representa um arquivo bitmap (.png , .jpg ou .gif). A utilização deste recurso cria um objeto BitmapDrawable.
- **Nine Patch-File** – representa um arquivo PNG com regiões elásticas para permitir o redimensionamento da imagem com base no conteúdo. A utilização deste recurso cria um objeto NinePatchDrawable.
- **Layer List** – representa um objeto drawable que gerencia uma série de outros drawables. Estes são desenhados alocados em uma matriz, de modo que o elemento com o maior índice será desenhado em cima. A utilização deste recurso cria um objeto LayerDrawable.
- **State List** – representa um arquivo XML que faz referência a diferentes gráficos de bitmap para estados diferentes (por exemplo, para usar uma imagem diferente quando um botão é pressionado). A utilização deste recurso cria um objeto StateListDrawable.

- **Level List** – representa um arquivo XML que define um drawable que gerencia um número alternativo de imagens, sendo que a cada drawable é atribuído um valor máximo de alternativas. A utilização deste recurso cria um objeto LevelListDrawable.
- **Drawable de Transição** – representa um arquivo XML que define um drawable que pode alternar entre dois recursos drawable. A utilização deste recurso cria um objeto TransitionDrawable.
- **Scale Drawable** – representa um arquivo XML que define um drawable que altera o tamanho de outro basedo no seu valor atual nível. A utilização deste recurso cria um objeto ScaleDrawable.
- **Shape Drawable** – representa um arquivo XML que define uma forma geométrica, incluindo cores e gradientes. A utilização deste recurso cria um objeto ShapeDrawable.

Recursos de strings

Um recurso de string fornece os textos para a sua aplicação, podendo opcionalmente aplicar estilos e formatações ao texto. Existem três tipos de recursos que podem fornecer strings a sua aplicação:

- String – XML recurso que fornece uma única sequência caracteres.
- String Array – XML recurso que fornece um conjunto de strings.
- Qualitity strings (plurais) – recurso XML que carrega strings diferentes para diferentes idiomas de uma mesma palavra ou frase.

Todos os recursos associados a strings são capazes de aplicar alguma marcação de estilo e argumentos de formatação.

String

Uma única cadeia caracteres que pode ser referenciada a partir do aplicativo ou a partir de outros arquivos de recursos (como um layout XML).

sintaxe:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <string
      name="string_name"
      >text_string</string>
</resources>
```

elementos:

<resources>

Obrigatório. Este deve ser o nó raiz.
Sem atributos.

<string>

atributos:

name: nome para a cadeia. Este nome será usado como a identificação do recurso. Para melhor entendimento veja o exemplo 1.

Arquivo XML salvo em res / valores / strings.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <string name="hello">Hello!</string>
</resources>
```

Esta disposição aplica-se XML uma sequência para a View:

```
<TextView
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:text="@string/hello" />
```

Este código do aplicativo recupera uma string:

```
String string = getString(R.string.hello);
```

String Array

Este recurso gera uma matriz de strings que podem ser referenciados a partir de uma aplicação.

Nota: a matriz de strings é um recurso simples que é referenciado usando o valor fornecido no nome de atributo (não o nome do arquivo XML). Como tal, pode combinar recursos matriz de strings com outros recursos simples em um arquivo XML.

Referência de recurso: em Java: . R.array string_array_name

sintaxe:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <string-array
      name="string_array_name">
      <item
         >text_string</item>
   </string-array>
</resources>
```

elementos:

<resources>

Obrigatório. Este deve ser o nó raiz.
Sem atributos.

<string-array>

Define uma matriz de strings. Contendo um ou mais <item> elementos.

atributos:

name: Para melhor entendimento. Este nome será usado como a identificação de recurso para fazer referência à matriz. Para melhor entendimento veja o exemplo 2.

Arquivo XML salvo em res / valores / strings.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="planets_array">
        <item>Mercury</item>
        <item>Venus</item>
        <item>Earth</item>
        <item>Mars</item>
    </string-array>
</resources>
```

Este código aplicativo recupera uma matriz de cadeia:

```
Resources res = getResources();
String[] planets = res.getStringArray(R.array.planets_array);
```

Qualitity Strings (plurais)

Idiomas diferentes possuem regras gramaticais diferentes quanto a formação de plurais. Em Inglês, por exemplo, a quantidade de 1 é um caso especial. Nós escrevemos "um livro", e para qualquer outra quantidade se escreve "n livros". Mesmo sendo a distinção entre o singular e o plural muito comum, outras linguagens não fazem essas distinções. O Android suporta um conjunto completo que pode ser representado por de zero, um, dois, poucos, muitos, e outros.

As regras para decidir qual caso usar para um determinado idioma e quantidade pode ser muito complexa, por isso Android fornece métodos como getQuantityString() para selecionar o recurso adequado.

localização do arquivo: res/valores/filename.xml

Sendo <plurals> o nome do elemento que será usado como a identificação de recurso.

referência de recurso: Em Java: . R.plurals plural_name

sintaxe:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <plurals
        name="plural_name">
        <item
            quantity=["zero" | "one" | "two" | "few" | "many" | "other"]
            >text_string</item>
    </plurals>
</resources>
```

elementos:

<resources>

Obrigatório. Este deve ser o nó raiz.
Sem atributos.

<plurals>

Uma coleção de strings, de que, uma string é fornecido, dependendo da quantidade de algo. Contém um ou mais <item> elementos.

atributos:

name: Um nome para o par de cordas. Este nome será usado como a identificação de recurso.

<item>

Uma string simples ou múltipla. O valor é uma referência aos recursos de cadeia. Deve ser herdada de um elemento do tipo <plurals>.

atributos:

Quatity: Palavra-chave. Um valor que indica quando esta cadeia deve ser usada. Os valores válidos, com não exaustiva de exemplos entre parênteses:

Valor	Descrição
Zero	Quando o idioma requer tratamento especial do número 0 (como em árabe).
Um	Quando o idioma requer tratamento especial de números como um (como acontece com o número 1 em línguas inglesa e mais outro, em russo, qualquer número que termina em 1, mas não

termina em 11 é nesta classe).

Dois	Quando o idioma requer tratamento especial de números como dois (como em galês).
Poucos	Quando a língua requer tratamento especial de "pequenas" (como números com 2, 3, e 4, em Checa, ou números terminando 2, 3, ou 4, mas não de 12, 13, ou 14, em polaco).
Muitos	Quando o idioma requer tratamento especial de "grandes" os números (como números que terminam com 11-99 em maltês).
Outro	Quando a linguagem não requer tratamento especial da quantidade determinada.

Para melhor entendimento veja o exemplo 3.

Arquivo XML salvo em res / valores / strings.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <plurals name="numberOfSongsAvailable">
      <item quantity="one">One song found.</item>
      <item quantity="other">%d songs found.</item>
   </plurals>
</resources>
```

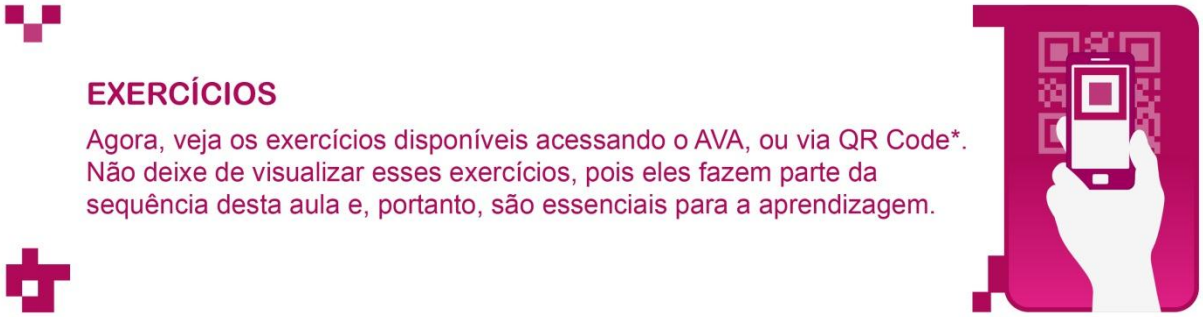
Arquivo XML salvo em res / valores-pl / strings.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <plurals name="numberOfSongsAvailable">
      <item quantity="one">Znaleziono jedną piosenkę.</item>
      <item quantity="few">Znaleziono %d piosenki.</item>
      <item quantity="other">Znaleziono %d piosenek.</item>
   </plurals>
</resources>
```

Código Java:

```
int count = getNumberOfsongsAvailable();
Resources res = getResources();
String songsFound = res.getQuantityString(R.plurals.numberOfSongsAvailable, count, count);
```

Para termos certeza de que entendemos a aula, vamos realizar uma atividade sobre interface do usuário.



***** O QR Code é um código de barras que armazena links às páginas da web. Utilize o leitor de QR Code de sua preferência para acessar esses links de um celular, tablet ou outro dispositivo com o plugin Flash instalado.

Próxima aula

Na próxima aula continuaremos a entender os componentes presentes na estrutura de uma aplicação Android, falaremos sobre filtros de mensagens, processos, theads e o ciclo de vida de uma aplicação.

REFERÊNCIAS

LECHETA, Ricard R. *Android* – *ap*renda a criar aplicações para dispositivos móveis com o Android SDK*. 2*. ed. São Paulo: Novatec, 2010.

ROGERS, Rick; et al. *Desenvolvimento de aplicações Android*. 1. ed. São Paulo: Novatec, 2009.