



# Programação para Android

Aula 02 - Parte 02: string.xml (internacionalização) e layout em diferentes orientações

# Na aula anterior...

## ► LinearLayout

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_margin="20px"  
    android:padding="20px"  
    android:background="#B7A6CD"  
    android:orientation="vertical"
```

```
>
```

## ► Widgets:

```
<ImageView />
```

```
<TextView />
```

```
<EditText />
```



```
    android:text="Proprietário"  
    android:textColor="#5BA4F1"  
    android:textSize="20dp"  
    android:typeface="serif"  
    android:hint="Nome e sobrenome do proprietário"  
    android:textColorHint="#AC43FA"  
    android:inputType="number"
```



# Objetivos

- ▶ Configurar um arquivo string.xml
- ▶ Internacionalizar uma aplicação a partir da configuração de arquivos de idioma
- ▶ Layout na horizontal e vertical



# Parte 01: string.xml e internacionalização

# Internacionalização no Android

- ▶ No desenvolvimento de aplicações para Android NÃO é considerado uma boa prática de programação colocar os textos do aplicativo diretamente na propriedade text conforme exemplificado a seguir.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nome: "
    android:id="@+id/tvNome"
/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Passaporte"
    android:id="@+id/tvPassaporte"
/>
```

# Internacionalização no Android

- ▶ Como boas práticas de programação um aplicativo Android contém um arquivo chamado `string.xml`, neste arquivo são armazenadas os textos exibidos no aplicativo, como por exemplo, o nome do aplicativo, nome de botões, nome de layouts. Deixando todos os textos do aplicativo em um só arquivo, facilita a manutenção além de podermos utilizar este recurso para internacionalizar nosso APP.
- ▶ Para suportar internacionalização, o ambiente Android possui arquivos de “recursos” (resource), em XML, contendo os textos a serem exibidos na aplicação.
- ▶ No arquivo `string.xml` criamos os objetos e em seguida referenciamos em nosso layout.

# Internacionalização no Android

## ► Definição de objetos no arquivo string.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">Internacionalização</string>
    <string name="lb_nome">Nome: </string>
    <string name="lb_passaporte">Passaporte: </string>
    <string name="lb_endereco">Endereço: </string>
    <string name="lb_telefone">Telefone: </string>
    <string name="lb_enviar">Enviar</string>
    <string name="lb_limpar">Limpar</string>

</resources>
```

# Internacionalização no Android

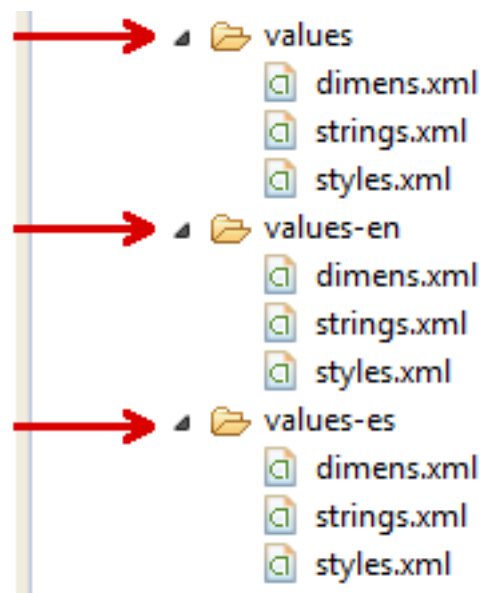
- ▶ Após definido no arquivo string.xml, podemos referenciar os objetos criado através da propriedade text, conforme verificado abaixo:

```
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/btEnviar"
    android:text="@string/lb_enviar"
/>
<Button
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/btLimpar"
    android:text="@string/lb_limpar"
/>
```



# Internacionalização no Android

- ▶ Para realizar internacionalização de nossas aplicações, configuramos um arquivo string.xml para cada idioma previsto, mas isto deve ser feito copiando a pasta values e em seguida renomeando esta pasta com a extensão do idioma (en para inglês, es para espanhol, fr para francês, etc.)
- ▶ Veja abaixo:



# Internacionalização no Android

- Depois de criado as pastas para cada idioma, configuramos os objetos criados no arquivo string.xml com a respectiva tradução. Exemplo:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">Internacionalization</string>
    <string name="lb_nome">Name: </string>
    <string name="lb_passaporte">Passport: </string>
    <string name="lb_endereco">Address: </string>
    <string name="lb_telefone">Phone number: </string>
    <string name="lb_enviar">Send</string>
    <string name="lb_limpar">Clear</string>

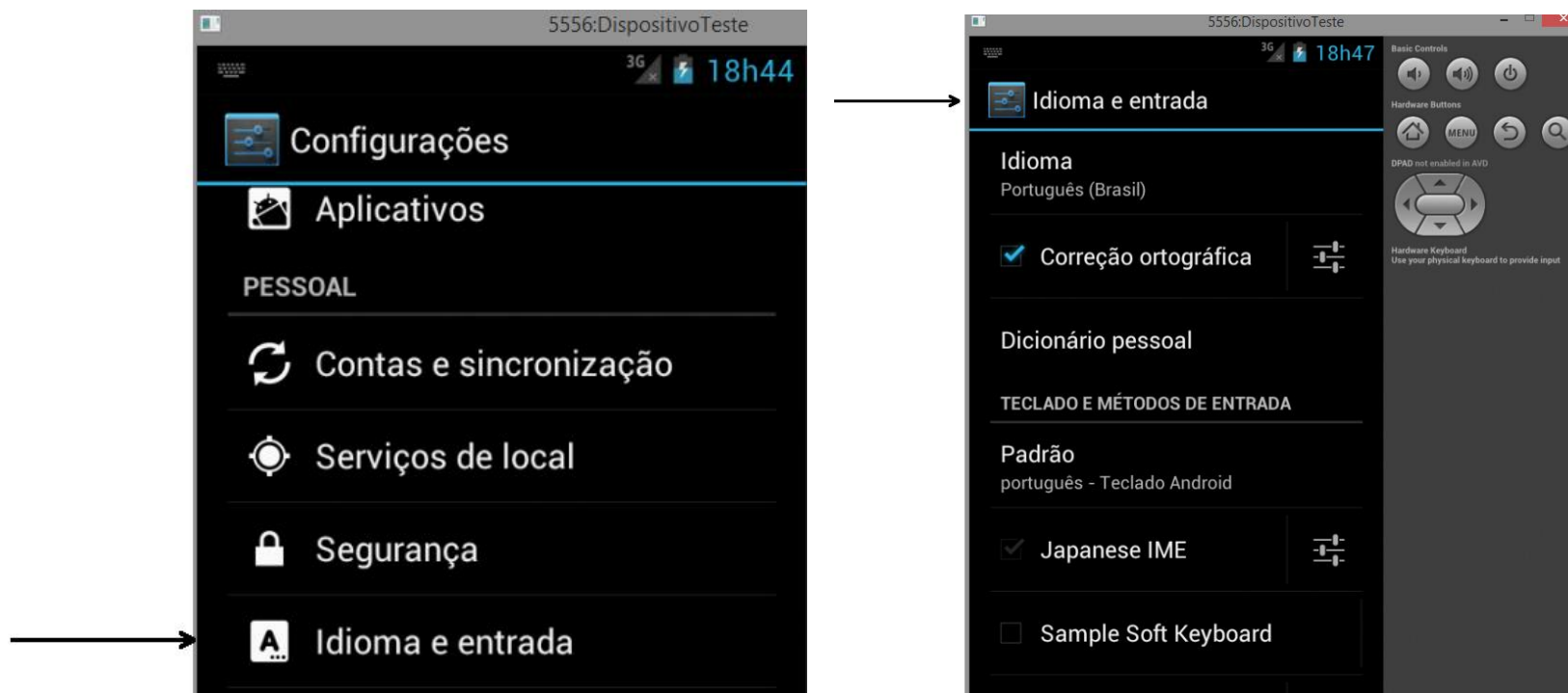
</resources>
```

# Internacionalização no Android

- ▶ De acordo com o idioma do dispositivo ele vai saber qual pasta irá buscar. Por exemplo, se o smartphone de um usuário estiver no idioma inglês ele vai buscar pela pasta values-en , se o estiver no idioma espanhol ele vai buscar na pasta values-es. Este processo é feito automaticamente pelo sistema operacional Android.
- ▶ Caso não exista um idioma específico o Android irá utilizar o arquivo definido na pasta values.
- ▶ Para testar a aplicação, iremos acessar a Configuração do dispositivo e alterar o idioma para algum que foi configurado.

# Internacionalização no Android

- Configuração do idioma do dispositivo



# Internacionalização no Android

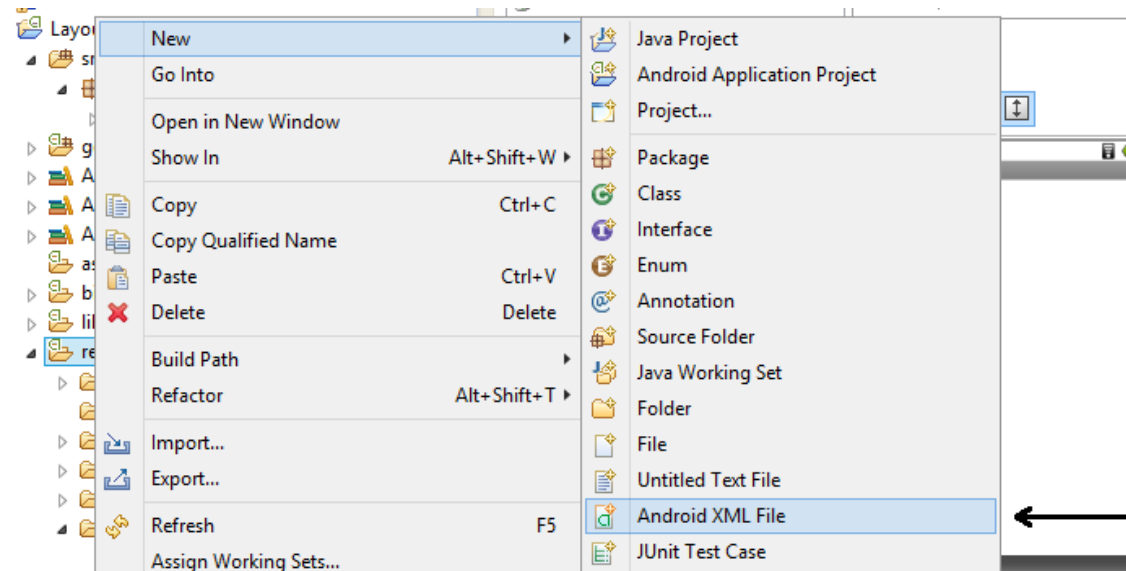
## ► Resultado

Internacionalization	Internacionalización	Internacionalização
Name:	Nombre:	Nome:
<input type="text"/>	<input type="text"/>	<input type="text"/>
Passport:	Pasaporte:	Passaporte:
<input type="text"/>	<input type="text"/>	<input type="text"/>
Address:	Dirección:	Endereço:
<input type="text"/>	<input type="text"/>	<input type="text"/>
Phone number:	Teléfono:	Telefone:
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Send"/>	<input type="button" value="Enviar"/>	<input type="button" value="Enviar"/>
<input type="button" value="Clear"/>	<input type="button" value="Limpiar"/>	<input type="button" value="Limpar"/>

## Parte 02: Layout na vertical e horizontal

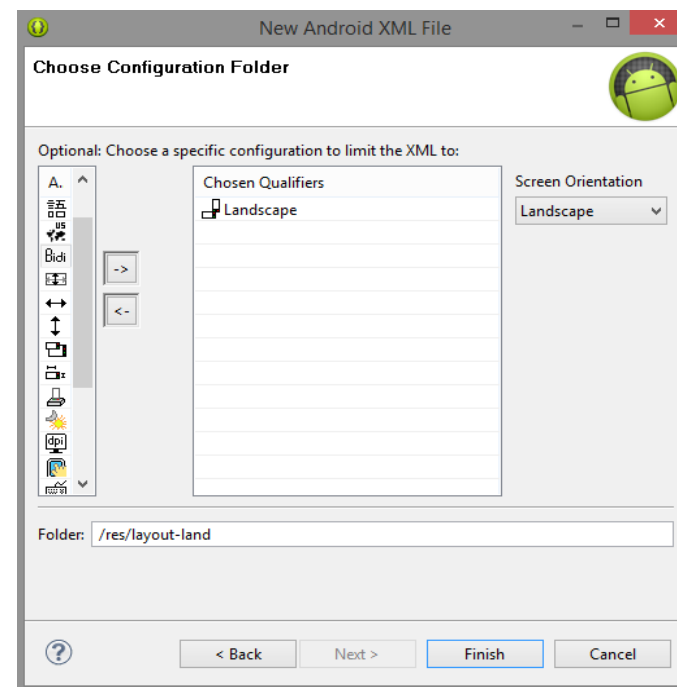
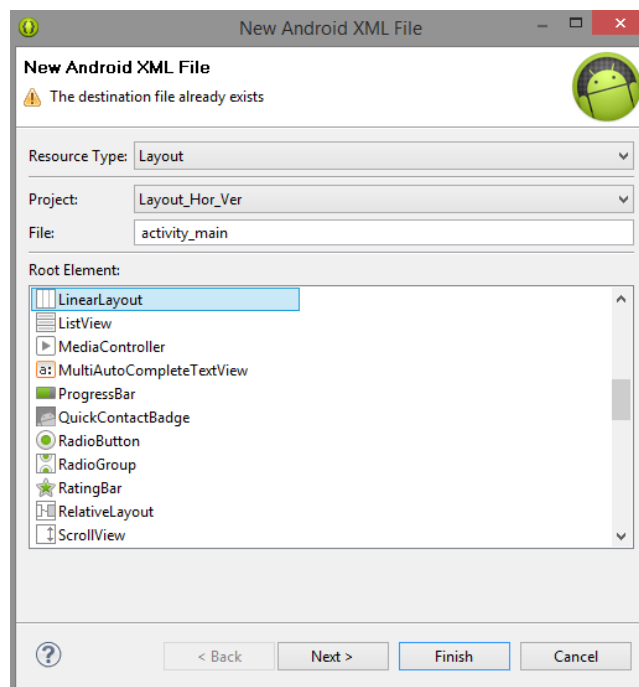
# Layout na vertical e horizontal

- ▶ Quando desenvolvemos aplicativos móveis precisamos criá-lo de modo que o layout mantenha-se perfeito nas diferentes orientações de tela. Em Android esse trabalho não é tão difícil quanto em outras ferramentas.
- ▶ Depois de criado o projeto no Eclipse clique com o botão direito na pasta “res” (onde ficam todos os recursos do aplicativo como, imagens, xml, layouts), em “New” e em “Android XML File”, como mostrado na imagem abaixo:



# Layout na vertical e horizontal

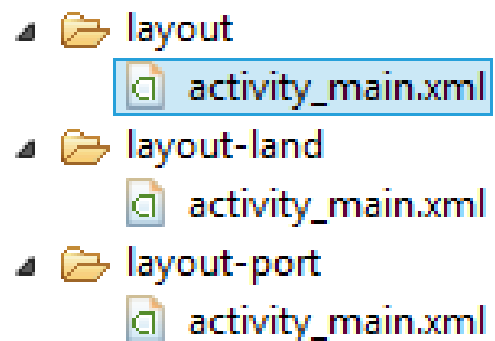
- ▶ Na tela de criação do xml marque o radio-button “**Layout**”, em **Available Qualifiers** escolha **orientation** mande para o lado direito “**Chosen Qualifiers**” e no combo-box “**Screen Orientation**” defina **Portrait**. Com isso você perceberá que uma nova pasta será criada com o nome de “**layout-port**”. Faça o processo novamente sendo que desta vez escolha no combo-box a opção **Landscape** e mais uma nova pasta será criada com o nome de “**layout-land**” como nas imagens abaixo:





# Layout na vertical e horizontal

- ▶ Com isso temos mais duas pastas no nosso projeto. Para que a troca de layouts funcione é preciso criar um layout xml com o mesmo nome em ambas as pastas e o android reconhecerá como o mesmo layout para orientações diferentes.



# Layout na vertical e horizontal

- No xml dos layout escolhemos uma imagem diferente para o background da orientação horizontal e outra para a orientação vertical.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/layout_horizontal"
    android:orientation="vertical" >
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@drawable/layout_vertical" >
</LinearLayout>
```

# Layout na vertical e horizontal

- Resultado dos layouts nas orientações vertical e horizontal



# Atividade prática avaliativa

- ▶ Em um hospital, na ala de internamento, as enfermeiras em cada turno verificam alguns sinais vitais dos pacientes: pressão arterial, batimentos cardíacos e , temperatura do corpo. Também é necessário anotar o nome do paciente e número do leito.
- ▶ Com os recursos já estudados, desenvolva uma APP que permita auxiliar as enfermeiras neste processo, de forma que possa substituir as anotações numa ficha de papel por um APP no Android. Após preenchimento a enfermeira deve clicar em um botão e enviar os dados.
- ▶ Utilize TextView, EditText e Button para montar a interface gráfica
- ▶ Utilize o recurso de string para disponibilizar sua APP em pelo menos 2 idiomas de livre escolha.
  - ▶ **DESENVOLVER EM SALA DE AULA!**
  - ▶ **Tempo disponível para realização: Até 2 h/a**
  - ▶ **DUPLA!**

# Na próxima aula...

- Manipulando dados e programando eventos em uma Activity.

