


Android

Suportando de Smartphones a Tablets




Tópicos Abordados



- Suportando diversos idiomas
- Suportando diversas versões do Android
- Suportando múltiplos tamanhos de tela
 - Tamanho
 - Orientação
 - Resolução
 - Densidade
- Classificações
 - Tamanho
 - Densidade
 - Orientação
- Qualificadores
- Definindo telas suportadas

Qualificadores



- Toda a preparação de um aplicativo para ser executado em dispositivos heterogêneos passa pelos qualificadores
- Os qualificadores são utilizados pelo Android para decidirem qual resource será utilizado com base nas configurações do dispositivo

```

/res
/drawable-mdpi
/drawable-hdpi
/layout-land
/layout-large
/values-v11
/values-v14

```

Qualificadores

Suportando Diversos Idiomas



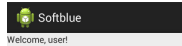
- O aplicativo pode mostrar textos em idiomas diferentes de acordo com a configuração do dispositivo

/res/layout/activity_main.xml

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

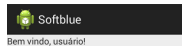
/res/values/strings.xml

```
<string name="txt_msg">
    Welcome, user!
</string>
```



/res/values-pt-BR/strings.xml

```
<string name="txt_msg">
    Bem vindo, usuário!
</string>
```



Exemplos para Diversos Idiomas



/res/values-en/strings.xml

Inglês

/res/values-es/strings.xml

Espanhol

/res/values-fr/strings.xml

Francês

/res/values-fr-CA/strings.xml

Francês do Canadá

/res/values-en-US/strings.xml

Inglês dos EUA

<lingua>

<lingua>-r<país>

IETF BCP 47

Idiomas via Programação



- É possível descobrir o idioma do dispositivo via programação
- O país/idioma é representado por um **Locale**

```
Locale locale = Locale.getDefault();
String country = locale.getCountry();
String language = locale.getLanguage();
```

Suportando Diversas Versões do Android



- No mercado existem dispositivos Android baseados nas mais diversas versões
- Qualificadores podem ser usados para usar diferentes resources

```
AndroidManifest.xml
<application android:theme="@style/AppTheme">
```

```
/res/values-v11/styles.xml
<style name="AppTheme"
    parent="android:Theme.Holo.Light" />
```

API Level 11
(Android 3.0)

```
/res/values-v14/styles.xml
<style name="AppTheme"
    parent="android:Theme.Holo.Light.DarkActionBar" />
```

API Level 14
(Android 4.0)

Minimum Version & Target Version



- É importante especificar as versões do Android suportadas pela sua aplicação
 - Versão mínima (minimum)
 - Versão “alvo” (target)

```
<manifest>
<uses-sdk
    android:minSdkVersion="11"
    android:targetSdkVersion="18" />
</manifest>
```

- Use como target a última versão disponível
 - Para aproveitar os novos recursos e visual
 - O mesmo vale para a compilação

Detectando a Versão via Programação



- A versão utilizada pelo dispositivo Android pode ser obtida via programação

```
int version = Build.VERSION.SDK_INT;
```

Retorna a API Level

- As versões possuem constantes associadas
 - Classe **Build.VERSION_CODES**

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
    //...
}
```

API de Compatibilidade



- A API de compatibilidade permite que versões antigas do Android suportem recursos criados em versões mais novas do Android
- Exemplos
 - Fragments
 - Action bar
 - Notificação
 - etc.

Suportando Múltiplas Telas

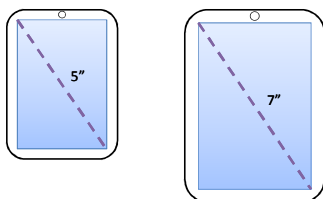


- Dispositivos Android possuem uma série de variações no que diz respeito à tela
 - Tamanho, densidade e orientação diferentes
- O Android tenta sempre adequar a interface da aplicação ao dispositivo
 - Algumas vezes é necessário utilizar layouts diferentes para características de telas diferentes
 - Isto visa aproveitar melhor o espaço da tela e melhorar a experiência do usuário


Tamanho (Size)



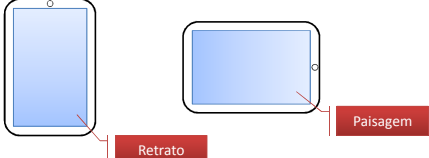
- O tamanho da tela diz respeito ao comprimento da diagonal da tela




Orientação (*Orientation*)



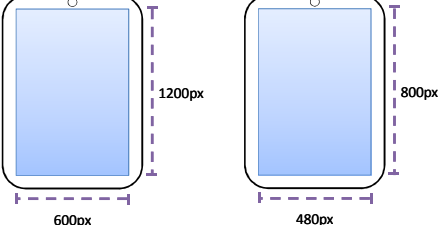
- Pode ser de dois tipos
 - Retrato (*portrait*)
 - A tela tem mais altura do que largura
 - Paisagem (*landscape*)
 - A tela tem mais largura do que altura




Resolução de Tela (*Resolution*)



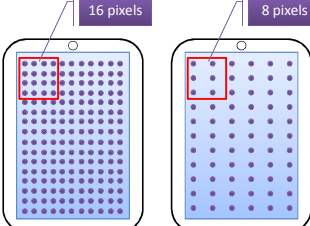
- É o número de pixels existentes na largura (*width*) e altura (*height*) da tela



Densidade da Tela (*Density*)



- Densidade é a quantidade de pixels existentes em uma área da tela



Resolução x Densidade

Softblue

- Ao montar uma interface gráfica, é preciso levar tanto a resolução quanto a densidade em consideração

A mesma imagem aparece de forma diferente porque a unidade *pixel* não leva a densidade em consideração

Density-Independent Pixel

Softblue

- Chamada também de **dip** ou **dp**
- É uma medida que leva em consideração a resolução e a densidade da tela

A unidade *dp* corrige o valor em pixels com base na densidade do dispositivo

Classificação de Tamanho


Softblue

- O Android possui quatro classificações para tamanhos de tela


- Qualificadores podem ser usados para criar layouts com base nesta classificação

```

/res/layout-small/main.xml    /res/layout-large/main.xml
/res/layout-normal/main.xml  /res/layout-xlarge/main.xml
  
```


Nova Classificação de Tamanho


- A partir do Android 3.2 uma nova classificação de tamanho passou a ser utilizada
 - Se a aplicação é voltada ao Android 3.2 como versão mínima, o uso desta nova classificação é recomendada
 - Se a aplicação suporta versões anteriores ao Android 3.2, é preciso usar a classificação baseada em *small*, *normal*, *large* e *xlarge*
- A nova classificação permite identificar diferentes tamanhos de tela de forma mais eficaz

Nova Classificação de Tamanho


- Baseada em um tamanho mínimo de tela, expresso em *dp*
 - Smallest width dp*
 - Tamanho mínimo da tela, que não depende da orientação
 - Width dp*
 - Largura corrente da tela (depende da orientação)
 - Height dp*
 - Altura corrente da tela (depende da orientação)

O *smallest width dp* substitui a classificação de tamanho baseada em *small*, *normal*, *large* e *xlarge*

Smallest Width dp


- Algumas classificações comuns

Smallest Width dp	Exemplo
320	Smartphones
480	Tablets de pequeno porte
600	Tablets de 7"
720	Tablets de 10"

- O qualificador *sw* pode ser utilizado

```
/res/layout-sw320dp/main.xml
```

```
/res/layout-sw480dp/main.xml
```

```
/res/layout-sw600dp/main.xml
```

```
/res/layout-sw720dp/main.xml
```

Width dp & Height dp



- *Width dp* pode ser usado quando o resource depende de uma largura mínima corrente

```
/res/layout-w600dp/main.xml
```

Usado quando houver pelo menos 600dp de largura na tela

- *Height dp* pode ser usado quando o resource depende de uma altura mínima corrente

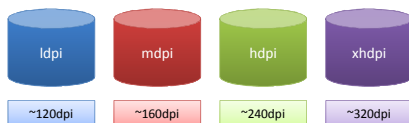
```
/res/layout-h600dp/main.xml
```

Usado quando houver pelo menos 600dp de altura na tela

Classificação de Densidade



- O Android possui quatro classificações principais para densidades de tela

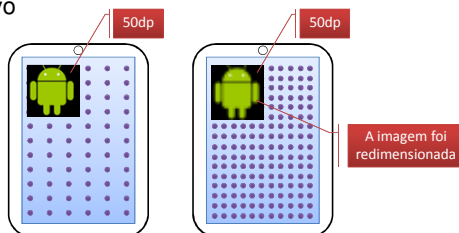


- Esta classificação é utilizada principalmente em resources do tipo *drawable*

Resources Drawable e a Densidade



- Quando uma imagem é criada, ela tem uma resolução fixa
- Ao ser exibida, a imagem vai precisar ser redimensionada de acordo com a densidade do dispositivo




Drawables para Diferentes Densidades


- Para não ter este problema, é possível especificar uma imagem para cada densidade de tela

	ldpi	mdpi	hdpi	xhdpi
Escala	0,75	1,0	1,5	2,0
Resolução	60x75	80x100	120x150	160x200



Drawables para Diferentes Densidades



- Qualificadores podem ser utilizados para referenciar os drawables nas mais diversas densidades

```

/res/drawable-ldpi/imagem.png
/res/drawable-mdpi/imagem.png
/res/drawable-hdpi/imagem.png
/res/drawable-xhdpi/imagem.png

```

Dependendo da densidade do dispositivo, o Android escolhe a imagem correspondente

Classificação de Orientação


- A orientação pode ser retrato ou paisagem
- Os qualificadores **port** e **land** podem ser usados para layouts com orientações diferentes

```

/res/layout-port/layout.xml
/res/layout-land/layout.xml

```

Informações Via Programação



- Tamanho

```
int screenLayout = getResources().getConfiguration().screenLayout;

if ((screenLayout & Configuration.SCREENLAYOUT_SIZE_MASK) ==
    Configuration.SCREENLAYOUT_SIZE_SMALL) {
    // Tamanho small
} else if ((screenLayout & Configuration.SCREENLAYOUT_SIZE_MASK) ==
    Configuration.SCREENLAYOUT_SIZE_NORMAL) {
    // Tamanho normal
} else if ((screenLayout & Configuration.SCREENLAYOUT_SIZE_MASK) ==
    Configuration.SCREENLAYOUT_SIZE_LARGE) {
    // Tamanho large
} else if ((screenLayout & Configuration.SCREENLAYOUT_SIZE_MASK) ==
    Configuration.SCREENLAYOUT_SIZE_XLARGE) {
    // Tamanho xlarge
}
```

Informações Via Programação



- Orientação

```
int orientation = getResources().getConfiguration().orientation;

if (orientation == Configuration.ORIENTATION_PORTRAIT) {
    // Retrato
} else if (orientation == Configuration.ORIENTATION_LANDSCAPE) {
    // Paisagem
}
```

- Densidade

```
int density = getResources().getDisplayMetrics().densityDpi;

if (density == DisplayMetrics.DENSITY_LOW) {
    // Densidade ldpi
} else if (density == DisplayMetrics.DENSITY_MEDIUM) {
    // Densidade mdpi
} else if (density == DisplayMetrics.DENSITY_HIGH) {
    // Densidade hdpi
} else if (density == DisplayMetrics.DENSITY_XHIGH) {
    // Densidade xhdpi
}
```

Aliases para Layouts



- Os aliases podem ser usados para evitar a duplicidade de arquivos de layout



Aliases para Layouts Softblue

- A solução é criar um mapeamento (alias) para o arquivo de layout real

`/res/layout/main_two.xml` 2 colunas

`/res/layout-sw600dp/layout.xml`

```
<resources>
  <item name="main" type="layout">@layout/main_two</item>
</resources>
```

`/res/layout-large/layout.xml`

```
<resources>
  <item name="main" type="layout">@layout/main_two</item>
</resources>
```

R.layout.main

Ordem dos Qualificadores Softblue

- Um resource pode ter mais de um qualificador associado
- Se for este o caso, a ordem dos qualificadores deve ser respeitada

Idioma
(pt_rR, en_rUS)

Smallest Width
(sw600dp, sw320dp)

Available Width
(w600dp, w320dp)

Available Height
(h600dp, h320dp)

Tamanho
(normal, large, xlarge)

Orientação
(port, land)

Densidade
(mdpi, hdpi, xhdpi)

Versão do Android
(v7, v11, v14)

Exemplos

/res/layout-large-land-v7

/res/layout-pt-port

/res/layout-v11-land

/res/layout-sw600dp-en

Resources Padrão Softblue

- É importante que sempre haja um resource padrão
 - O Android usa o resource padrão se não acha uma alternativa melhor de resource
 - Se um resource padrão não for encontrado, vai ocorrer um erro na aplicação

Português →

Inglês →

Espanhol →

Francês →

→ /res/values-pt/...

→ /res/values-en/...

→ /res/values-es/...

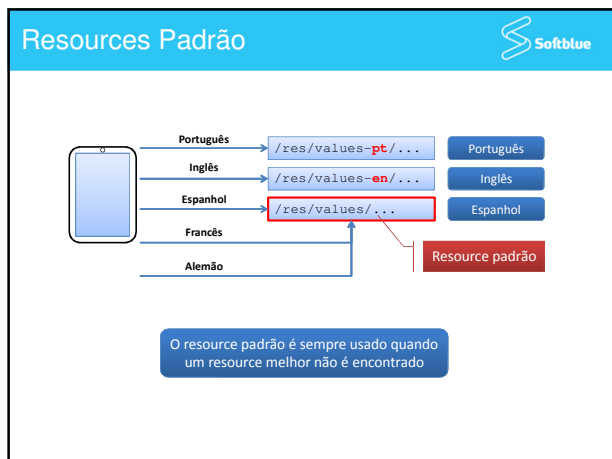
→ ✗

Português

Inglês

Espanhol

Erro



- ## Definindo as Telas Suportadas
- Se a sua aplicação não suporta todos os tipos de telas disponíveis, é preciso definir que tipos de tela ela suporta
 - Lojas como a Google Play Store filtram as aplicações com base nessas restrições de tela
 - Dispositivos com telas não suportadas não poderão baixar a aplicação

A Tag `supports-screens`

- A tag **`supports-screens`** deve ser usada no *AndroidManifest.xml* para determinar as telas suportadas

```
<supports-screens
  android:smallScreens="false"
  android:normalScreens="false"
  android:largeScreens="true"
  android:xlargeScreens="true"
  android:requiresSmallestWidthDp="600" />
```

Antes da API Level 13

Da API Level 13 em diante