

# Desenvolvimento de Interfaces Android

Análise e Desenvolvimento de Software - 2020 / 4

# Compatibilidade do Dispositivo

# Compatibilidade de Dispositivo

O Android foi feito para funcionar em vários tipos diferentes de dispositivos, como smartphones, tablets e televisões.

Como desenvolvedor, a variedade de dispositivos oferece a possibilidade de um enorme público para seu app.

Para que seu aplicativo tenha sucesso em todos esses dispositivos, ele precisa tolerar uma certa variabilidade de recursos e oferecer uma interface do usuário flexível que se adapte a diferentes configurações de tela.

# Compatibilidade de Dispositivo

Para facilitar seu trabalho, o Android oferece um framework de aplicativo dinâmico em que você pode fornecer recursos do aplicativo específicos para a configuração em arquivos estáticos, como diferentes layouts XML para diferentes tamanhos de tela.

Em seguida, o Android carrega os recursos adequados com base na configuração atual do dispositivo.

Com o planejamento do design do app e alguns recursos adicionais, você pode publicar um único pacote de apps (APK) que oferece uma experiência otimizada para o usuário em vários dispositivos.



# Compatibilidade de Tela

# Compatibilidade de Tela

O Android é executado em uma variedade de dispositivos que oferecem diferentes tamanhos e densidades de tela.

O sistema realiza um escalonamento e redimensionamento básico para adaptar sua interface do usuário a diferentes telas.

No entanto, ainda é preciso realizar outras ações para garantir que ela se ajuste perfeitamente a cada tipo de tela.

# Compatibilidade de Tela

Por padrão, o Android redimensiona seu aplicativo para que ele caiba na tela atual.

Para garantir que seu layout seja redimensionado corretamente até mesmo para pequenas variações no tamanho da tela, você deve implementá-lo com flexibilidade em mente.

O princípio básico que você deve seguir é evitar codificar a posição e o tamanho dos componentes da sua IU.

Em vez disso, permita que os tamanhos da visualização se expandam e especifique posições relativas à visualização pai ou a outras visualizações irmãs para que a ordem e os tamanhos que você pretende usar permaneçam iguais mesmo com a mudança do layout.





## Laboratório → Criar Layouts Alternativos

1. Abra o layout padrão e clique em Orientation for Preview na barra de ferramentas.
2. Na lista suspensa, clique para criar uma variante sugerida, como Create Landscape Variant ou clique em Create Other.
3. Se você tiver selecionado Create Other, o Select Resource Directory será exibido. Selecione um qualificador de tela à esquerda e adicione-o à lista de Chosen qualifiers. Quando terminar de adicionar qualificadores, clique em OK. Consulte as seções a seguir para ver informações sobre os qualificadores de tamanho de tela.



# Medidas

# Medidas

É possível especificar largura e altura com medidas exatas, embora não seja recomendável na maioria dos casos. Em geral, usa-se uma destas constantes para definir a largura e a altura:

- **wrap\_content**: instrui a exibição a se redimensionar de acordo com as medidas exigidas pelo conteúdo.
- **match\_parent**: instrui a exibição a assumir o maior tamanho permitido pelo grupo de exibições pais.

**Observação:** ao usar ConstraintLayout, não use match\_parent. Em vez disso, defina a dimensão como 0dp para ativar um comportamento especial chamado de “corresponder restrições”, que geralmente é o mesmo que esperado de match\_parent.

# Medidas

Dispositivos Android não só vêm com vários tamanhos de telas (celulares, tablets, TVs etc.) como as telas também têm diferentes tamanhos de pixel.

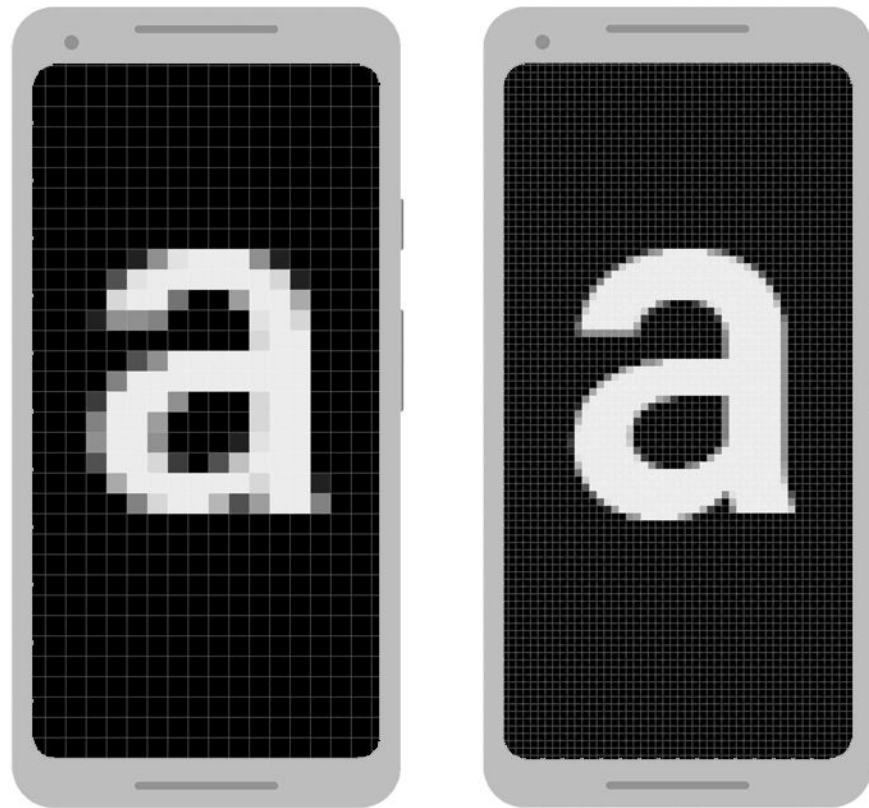
Enquanto um dispositivo tem 160 pixels por polegada quadrada, outro encaixa 480 pixels no mesmo espaço.

Se você não considerar essas variações em densidade de pixels, o sistema pode dimensionar suas imagens (resultando em imagens borradas) ou as imagens podem aparecer em um tamanho completamente errado.

# Medidas

A primeira armadilha a ser evitada é usar pixels para definir distâncias ou tamanhos.

Definir dimensões com pixels é um problema, porque telas diferentes têm densidades pixels diferentes, de maneira que o mesmo número de pixels pode corresponder a diferentes tamanhos físicos nos vários dispositivos.



# Medidas

Para preservar o tamanho visível da sua interface do usuário em telas com diferentes densidades, você precisa projetar sua IU com pixels de densidade independente (**dp**) como unidade de medida.

Um **dp** é uma unidade de pixel virtual aproximadamente do tamanho de um pixel em uma tela de densidade média (160 dpi, a densidade "básica").

O Android converte esse valor no número apropriado de pixels reais para cada densidade.



# Medidas

Você precisa usar pixels escalonáveis (**sp**) como suas unidades para definir tamanhos de texto, mas nunca para tamanhos de layout.

Por padrão, a unidade **sp** é do mesmo tamanho que a **dp**, mas ela é redimensionada com base no tamanho de texto preferencial do usuário.

# Medidas

**px** → Correspondente ao número de pixels (pontos) da tela e a recomendação é evitar utilizar px e apenas usar em casos bem específicos e raros.

**dip / dp** → Density-independent Pixels: essa unidade é relativa à resolução da tela e deve ser usada sempre que for definir medidas de componentes visuais. Por exemplo se a resolução da tela é de 160 dpi, significa que um dp representa 1 pixel em um total de 160.

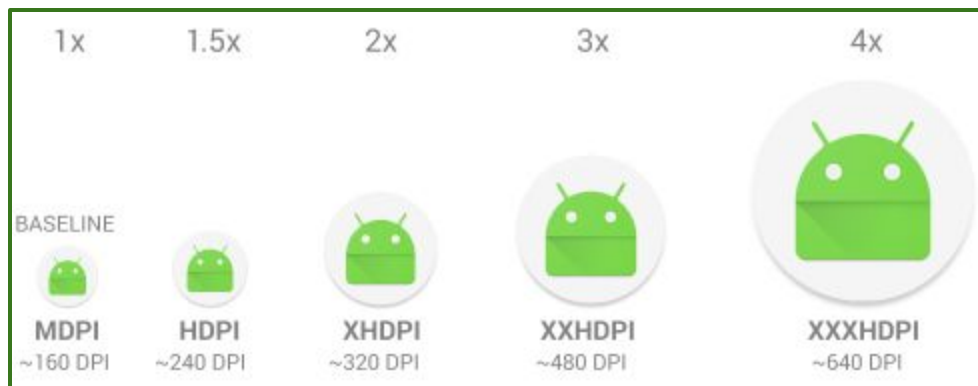
**sp** → Scale-independent Pixels: considera o tamanho da fonte que o usuário está utilizando. É recomendado que use essa unidade somente quando especificar o tamanho de uma fonte, para que esta seja automaticamente ajustada conforme as preferências da tela do usuário.



# Medidas

Para fornecer boa qualidade gráfica em dispositivos com densidades de pixels diferentes, você precisa oferecer várias versões de cada bitmap no seu app: uma para cada intervalo de densidade em uma resolução correspondente.

Se não fizer isso, o Android precisará dimensionar seu bitmap para que ele ocupe o mesmo espaço visível em cada tela, resultando em artefatos de com desfoque.





Qualificador de densidade	Descrição
<b>ldpi</b>	Recursos para telas de baixa densidade ( <i>ldpi</i> ) (cerca de 120 dpi).
<b>mdpi</b>	Recursos para telas de média densidade ( <i>mdpi</i> ) (cerca de 160 dpi). Essa é a densidade básica.
<b>hdpi</b>	Recursos para telas de alta densidade ( <i>hdpi</i> ) (cerca de 240 dpi).
<b>xhdpi</b>	Recursos para telas de densidade extra-alta ( <i>xhdpi</i> ) (cerca de 320 dpi).
<b>xxhdpi</b>	Recursos para telas de densidade extra-extra-alta ( <i>xxhdpi</i> ) (cerca de 480 dpi).
<b>xxxhdpi</b>	Recursos para telas de densidade extra-extra-extra-alta ( <i>xxxhdpi</i> ) (cerca de 640 dpi).
<b>nodpi</b>	Recursos para todas as densidades. Esses são recursos independentes de densidade. O sistema não dimensiona recursos marcados com esse qualificador, independentemente da densidade da tela atual.
<b>tvdpi</b>	Recursos para telas entre mdpi e hdpi; aproximadamente 213 dpi. Esse não é considerado um grupo de densidade "principal". Ele é destinado principalmente a televisões, e a maioria dos apps provavelmente não precisa dele. Fornecer recursos mdpi e hdpi é o suficiente para a maioria dos apps, e o sistema os dimensionará conforme for apropriado. Se julgar necessário fornecer recursos tvdpi, dimensione-os para um fator de 1,33*mdpi. Por exemplo, uma imagem de 100 x 100 px para telas mdpi precisa ter 133 x 133 px para tvdpi.

# Idiomas e Culturas

# Idiomas e Culturas

Os apps incluem recursos que podem ser específicos para determinada cultura. Por exemplo, um app pode incluir strings próprias de uma cultura que são traduzidas para o idioma da localidade atual.

É recomendável manter os recursos específicos da cultura separados do restante do app.

O Android resolve recursos específicos de idioma e cultura com base na configuração de localidade do sistema.

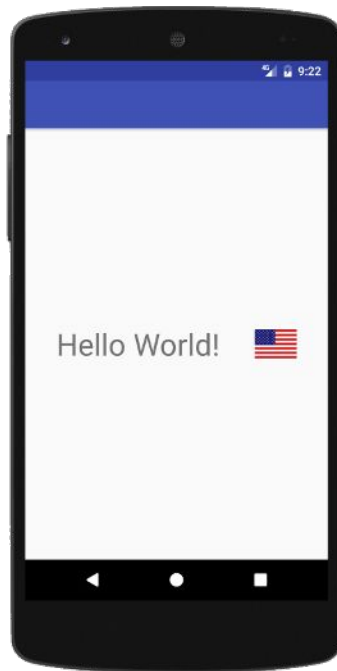
Você pode oferecer compatibilidade com diferentes localidades usando o diretório de recursos no seu projeto Android.



# Idiomas e Culturas

Você pode especificar recursos adaptados à cultura das pessoas que usam seu app e fornecer qualquer tipo de recurso apropriado para o idioma e a cultura dos seus usuários.

Por exemplo, a captura de tela a seguir mostra um app exibindo recursos drawable e de string na localidade padrão do dispositivo (en\_US) e em espanhol (es\_ES).





## **Laboratório** → Criar Compatibilidade de Idioma

1. Criar strings.xml para Brasil, Estados Unidos e Espanha.
2. Criar pastas em drawable para os três países e carregar as bandeiras de cada um.

OBS: Pegue as bandeiras do aplicativo FlagQuiz que fizemos anteriormente.