



PADO
Labs

INTRODUÇÃO AO DESENVOLVIMENTO MOBILE



Aula II - Views e Layout

Contato



Professor



David Krepsky



david@hausenn.com.br



DKrepsky

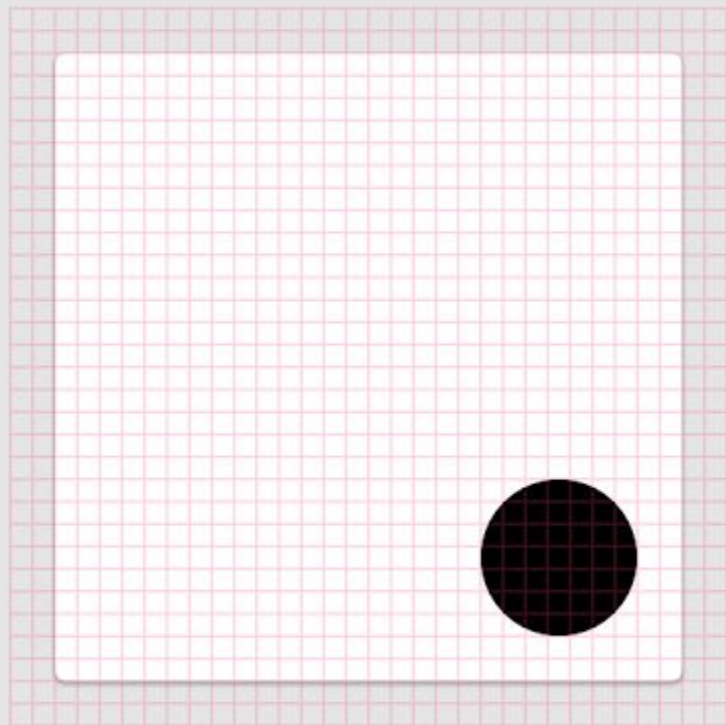
Unidades DP e SP



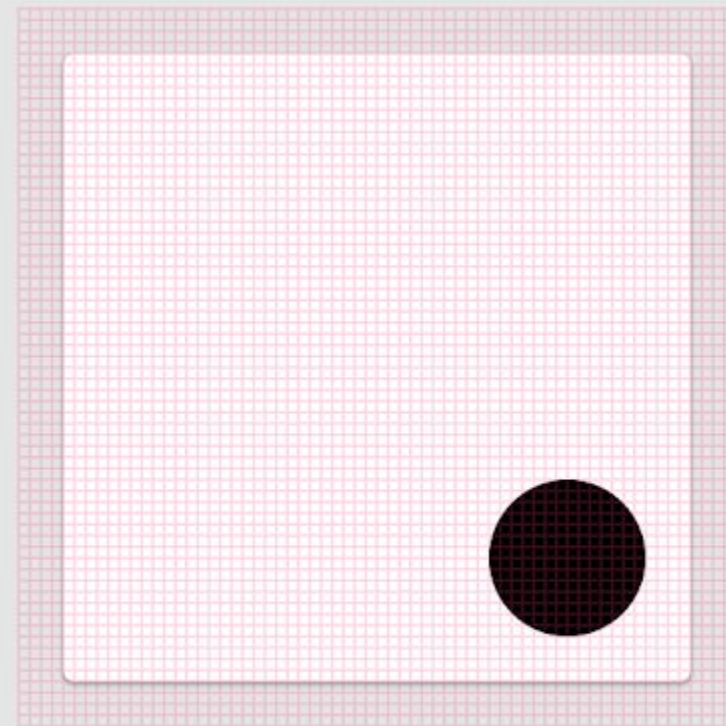
- As telas de alta densidade têm mais pixels por polegada do que as de baixa densidade.
- Como resultado, os elementos da interface do usuário com as mesmas dimensões de pixel aparecem maiores em telas de baixa densidade e menores em telas de alta densidade.

Unidades DP e SP

1



2



Unidades DP e SP



- Os pixels escaláveis (sp) têm a mesma função que os pixels independentes de densidade (dp), mas para fontes. O valor padrão de um sp é o mesmo que o valor padrão de um dp.
- A principal diferença entre um sp e um dp é que os sp preservam as configurações de fonte de um usuário. Os usuários que têm configurações de texto maiores para acessibilidade verão que os tamanhos de fonte correspondem às suas preferências de tamanho de texto.

Unidades DP e SP



- Os pixels escaláveis (sp) têm a mesma função que os pixels independentes de densidade (dp), mas para fontes. O valor padrão de um sp é o mesmo que o valor padrão de um dp.
- A principal diferença entre um sp e um dp é que os sp preservam as configurações de fonte de um usuário. Os usuários que têm configurações de texto maiores para acessibilidade verão que os tamanhos de fonte correspondem às suas preferências de tamanho de texto.

Unidades DP e SP



$dp = (largura \text{ em pixels} * 160) / \text{densidade da tela}$

Largura física da tela	Densidade da tela	Largura da tela em pixels	Largura da tela em dps
1,5 e	120	180px 240px 360px	240 dp
1,5 e	160		
1,5 pol	240		

Unidades DP e SP



De modo geral, sempre usar DP para tamanho e posicionamento dos elementos e sempre utilizar SP para o tamanho das fontes.

UI e Views

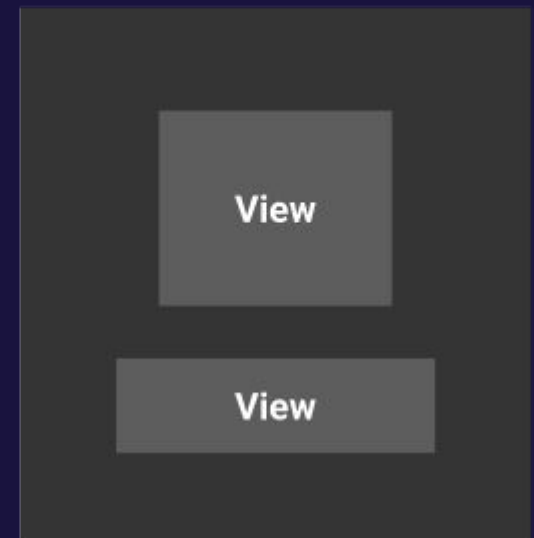


- A Interface do usuário (*User Interface*) é tudo o que você vê na tela do celular, ou seja, texto, imagens, botões, menus e etc.
- Cada um desses elementos é o que chamamos de *View* no Android.
- Por exemplo, um **botão** é um *View* do tipo ***Button***, um **texto** é um *View* do tipo ***TextView***.

ViewGroup ou Containers

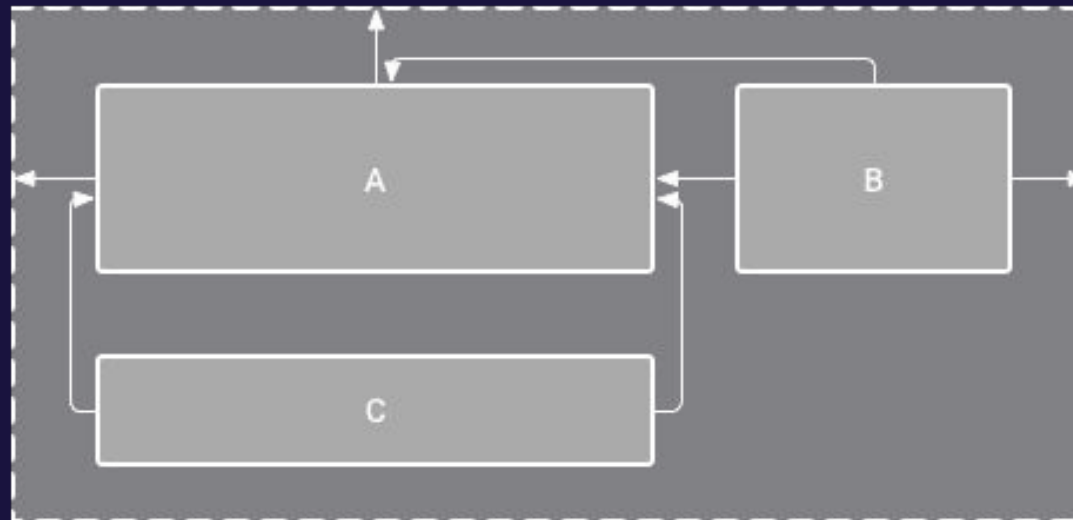
- Para organizar os Views dentro de uma UI, são utilizados grupos de views, que funcionam como containers para os elementos.
- Os ViewGroup podem organizar os views de diversas formas diferentes.

ViewGroup



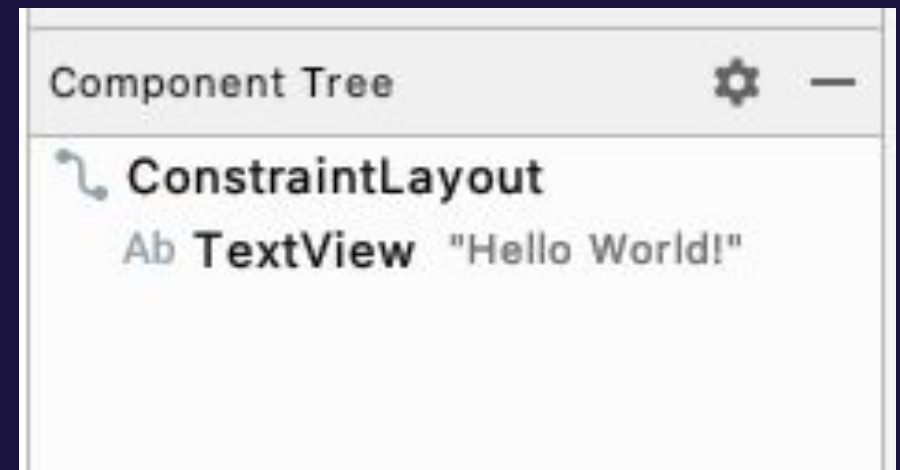
ConstraintLayout

- O nome Constraint significa restrições, ou seja, é um grupo que realiza o layout da UI de acordo com uma série de restrições de posicionamento para os elementos View.



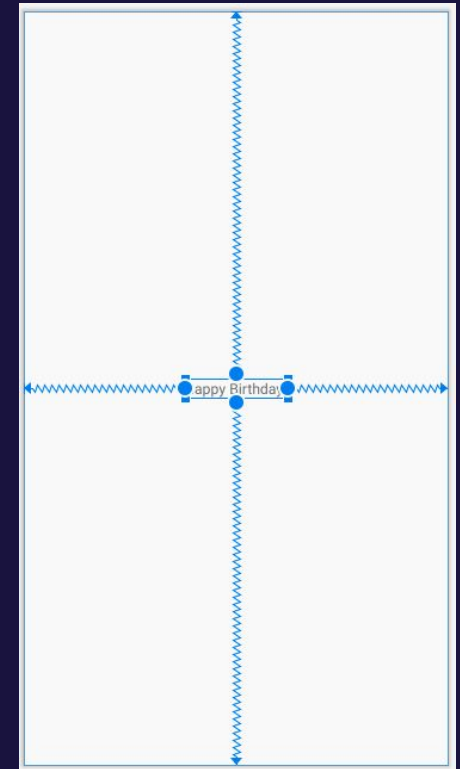
ConstraintLayout

- O ViewGroup do tipo ConstraintLayout é um dos tipos mais utilizados no Android.
- Na imagem ao lado, temos um TextView dentro de um grupo do tipo ConstraintLayout.



Constraint (Restrições)

- Para que um elemento tenha seu posicionamento definido, o mesmo deve ter suas restrições na horizontal e na vertical definidas.



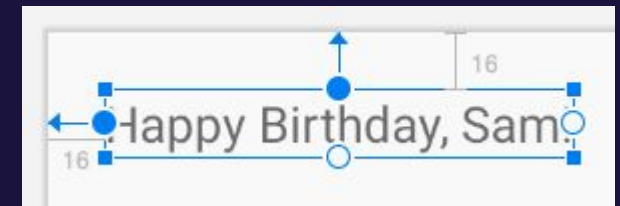
Constraint (Restrições)

- Quando um elemento não possui informações suficientes sobre o seu posicionamento, um erro aparece no component tree, informando se o problema é na horizontal ou vertical.



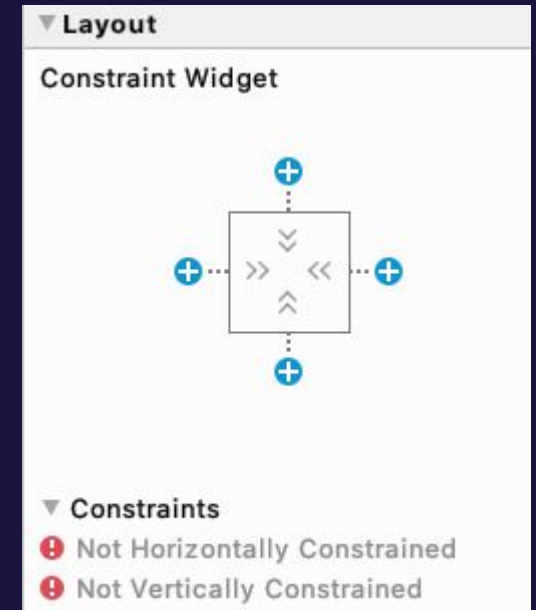
Constraint (Restrições)

- Uma forma de adicionar as restrições, é através do editor de telas, selecionando o elemento e utilizando os círculos em branco, arraste e solte o ponto de referência.
- As linhas de restrições funcionam como molas, que puxam o elemento na direção da seta,



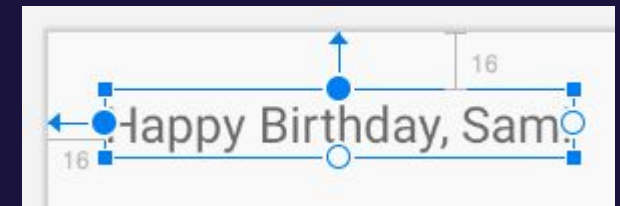
Constraint (Restrições)

- Uma outra forma de adicionar restrições é através do painel de atributos, na opção de layout.
- Ao clicar no +, é adicionado automaticamente uma restrição naquela direção.



Constraint (Restrições)

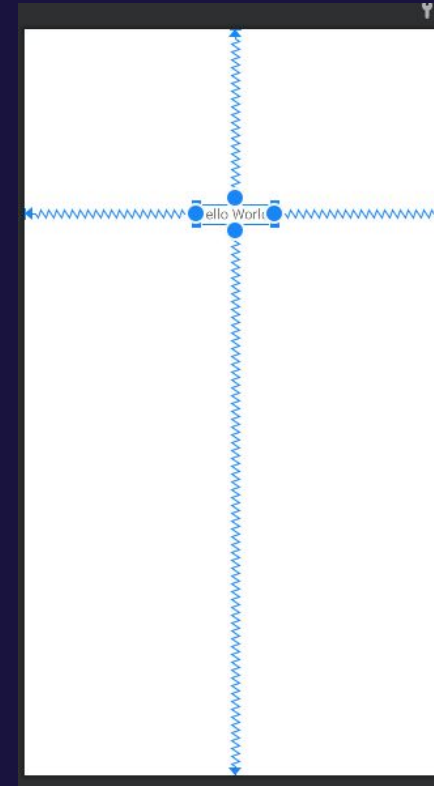
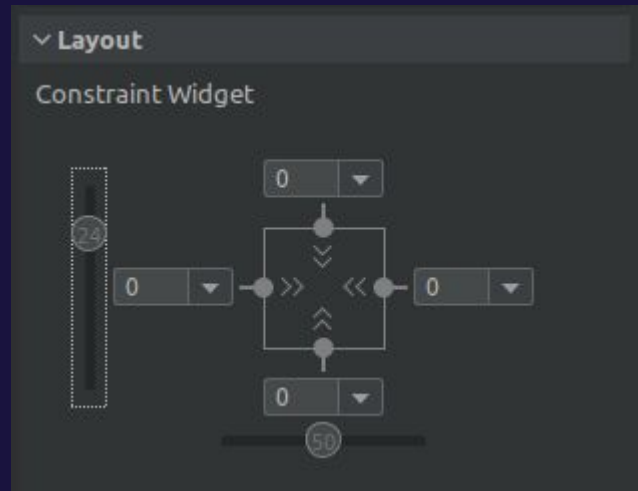
- Além da restrição, podemos adicionar uma margem, para garantir um espaçamento no elemento.
- Na figura ao lado, temos um TextView, com restrições Top e Left e 16dp de margem em ambos os lados.
- As margens podem ser adicionadas no painel de atributos, nas opções de layout.



Offset

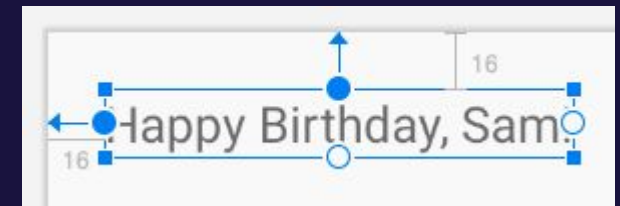
- Quando adicionamos restrições em ambas as direções, em cima e em baixo ou direita e esquerda, o Android nos permite ajustar o posicionamento do componente através da barra de offset.

Offset



Margem

- Além da restrição, podemos adicionar uma margem, para garantir um espaçamento no elemento.
- Na figura ao lado, temos um TextView, com restrições Top e Left e 16dp de margem em ambos os lados.
- As margens podem ser adicionadas no painel de atributos, nas opções de layout.



Padding



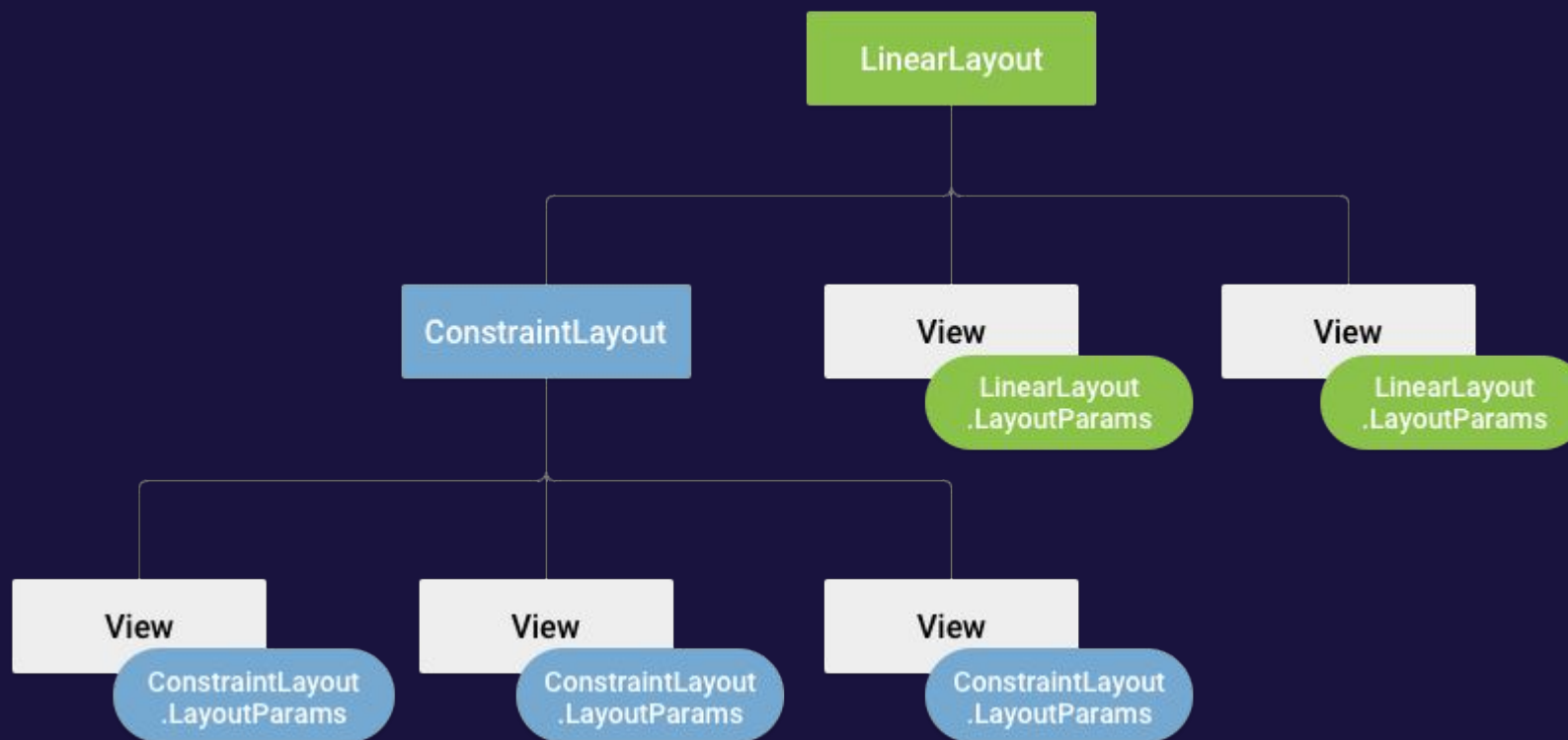
- Similar ao HTML, podemos adicionar padding aos elementos de modo a criar espaço interno no View.

> numeric		
onClick		
orientation		
outlineProvider		
overScrollMode		
▼ padding	[20dp, 20dp, 20dp, 20dp, 20d	
padding	20dp	
paddingStart	20dp	
paddingLeft	20dp	
paddingTop	20dp	
paddingEnd	20dp	
paddingRight	20dp	
paddingBottom	20dp	
password	—	
phoneNumber	—	
pointerIcon		
privateImeOptions		

Outros tipos de layout

- Para criarmos UIs mais complexos, o Android nos permite utilizar outros tipos de layout e suas combinações.
- Os layouts (ViewGroups) e elementos (Views) são arranjados em uma estrutura do tipo Árvore.

Outros tipos de layout



Linear Layout

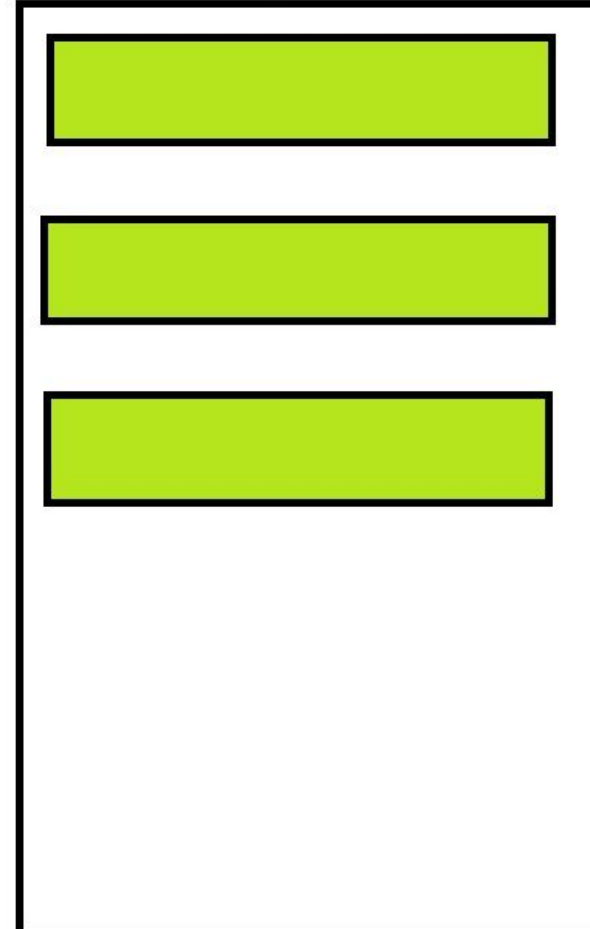
- LinearLayout é um grupo de visualizações que alinha todos os filhos em uma única direção vertical ou horizontal.
- Você pode especificar a direção do layout com o atributo *android:orientation*.

Linear Layout

Linear Layout Horizontal



Linear Layout Vertical



Space

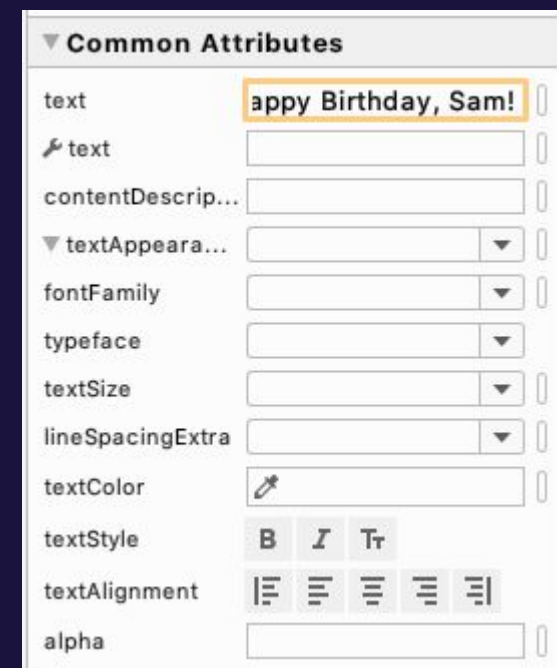


- O Space permite adicionar espaçamento entre itens dentro de um layout.

Outros Views



- ImageView: permite adicionar imagens ao aplicativo.
- CardView: é um card para visualização de itens, muito útil quando se quer mostrar itens com bordas arredondadas.



Criando uma nova tela

- Para criar uma nova tela, adicione uma nova activity ao projeto.
- Botão direito no projeto -> New -> Activity -> Empty Activity

Navegando para uma nova tela

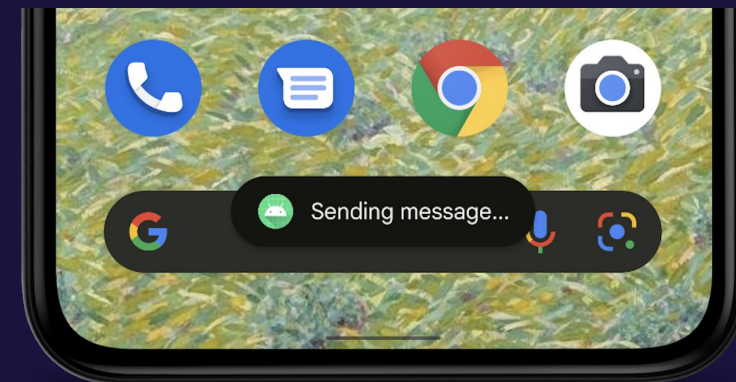


```
startActivity(Intent(this, Pagina2::class.java));
```

Bonus: Toasts



- Toasts permite uma forma simples de comunicar informações temporárias ao usuário.
- Link da documentação: [Android Developer | Toasts](#)





ATIVIDADE I

Catálogo da linha digital
PADO

Atividade I



Crie um aplicativo que sirva como catálogo para a linha de fechaduras digitais da PADO.

[FIGMA](#)

Subir no GitHub até a próxima Aula



PADO
Labs