

The background features a complex network of thin grey lines and dots, forming a web-like structure. Scattered throughout are various triangles of different sizes and orientations, some with solid black dots at their vertices. The overall aesthetic is modern and technical.

PROGRAMA DEV VENTURE

Desenvolvimento Android

GIT
Revisão de conceitos

01

ORGANIZAÇÃO
Strings, Dimensions,
Styles

02

LINEAR LAYOUT
Entendendo estrutura de
layouts

03

SCROLLVIEWS

Layouts mais
responsivos

04

Aula 08 AGENDA



Atividades:



- Desafio HackerRank (Link no Github) - 15/06
 - Koans (Link no Github) - 19/06
 - Tapa no Visual da App Lance Dados - 19/06
 - Desafio Calculadora - 16/06
-



01

GIT

Sistema de Controle de Código Fonte

O QUE É GIT

Git é um Sistema de Controle de Código Fonte (do inglês Source Code Management).

É uma ferramenta importante para nos apoiar na gestão de históricos dos documentos (entre várias outras coisas)

Arquivo:
curso-android.txt

18:35 - 06/07/2021
| Marcella Criou o
arquivo

18:40 - 06/07/2021
| Marcella
adicionou um título

18:47 - 06/07/2021
| Marcella
adicionou uma
imagem

18:50 - 06/07/2021
| Marcella coloriu o
texto de cinza

GIT

O QUE É GIT

Linux:

```
$ sudo yum install git-all
```

```
$ sudo apt-get install git-all
```

Mac OS:

```
$ git
```

<http://git-scm.com/download/mac>

Windows:

<http://git-scm.com/download/win>




O QUE É GIT

git init

Num terminal (Git Bash)

- `git init myproject`
- `cd myproject`
- `git config user.email "seu email"`
- `git config user.name "seu nome"`

- Pasta myproject **é criada**
 - Dentro desta pasta a **pasta .git passa a existir**
 - **Tudo que fizermos** dentro desta pasta pode ser versionado pelo Git.
 - Definimos para o projeto git nossa autoria
- 



O QUE É GIT

git status

Num terminal (Git Bash)

- git status
- touch primeiroarquivo.txt
- git status

- **Status:** alguma novidade na minha pasta? Algo que eu queira gravar?

O QUE É GIT

```
marcelasouza@2045-1B: ~/Personal Projects/testgit/myproject
~/Pe/t/myproject git master git status
branch master

commits yet

nothing to commit (create/copy files and use "git add" to track)
~/Pe/t/myproject git master
```

```
On branch master
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    primeiroarquivo.txt

nothing added to commit but untracked files present (use "git add" to track)
```



O QUE É GIT

git add

Num terminal (prompt comando)

- git add .
- git status

- add: coloca as mudanças feitas numa espécie de área intermediária (stage área) antes delas serem permanentemente registradas
- Você pode adicionar arquivos específicos ou tudo que foi modificado (.)

O QUE É GIT

git add

```
On branch master
```

```
No commits yet
```

```
Changes to be committed:
```

```
(use "git rm --cached <file>..." to unstage)
```

```
new file:   primeiroarquivo.txt
```

O QUE É GIT

git commit

Num terminal (prompt comando)

- git commit -m "Meu primeiro commit"
- git status

- commit: grava permanentemente as alterações no histórico (mas dá pra desfazer se precisar)
- É comum em projeto termos padrões de como escrever a mensagem do commit:
 - Padronizar a língua
 - Escrever a mensagem no imperativo ("Cria primeiro arquivo" ao invés de "Criando primeiro arquivo", "Criei primeiro arquivo", "Primeiro arquivo criado" etc)
 - Um commit deve ser relacionado à uma única mudança. A mensagem de descrição deve ser curta.

Conventional Commits

O QUE É GIT

git commit

```
🍏 ~/Pe/t/myproject git master +1 git commit -m "Meu primeiro commit"
[master (root-commit) 2388dc0] Meu primeiro commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 primeiroarquivo.txt

🍏 ~/Pe/t/myproject git master git status ✓ 11:47:13 ☺
On branch master
nothing to commit, working tree clean

🍏 ~/Pe/t/myproject git master ✓ 11:47:27 ☺
```

Conventional Commits



O QUE É GIT

git log

Num terminal (prompt comando)

- git log

- log: mostra histórico dos commits

Conventional Commits



O QUE É GIT

git log

```
commit 2388dc024f419609c21145bc797bb056e3c62946 (HEAD -> master)
```

```
Author: Marcella Souza <marcella.souza@ifood.com.br>
```

```
Date: Sat May 8 11:47:13 2021 -0300
```

```
    Meu primeiro commit
```

```
(END)
```


Conventional Commits



DESAFIO

Num terminal (prompt comando)

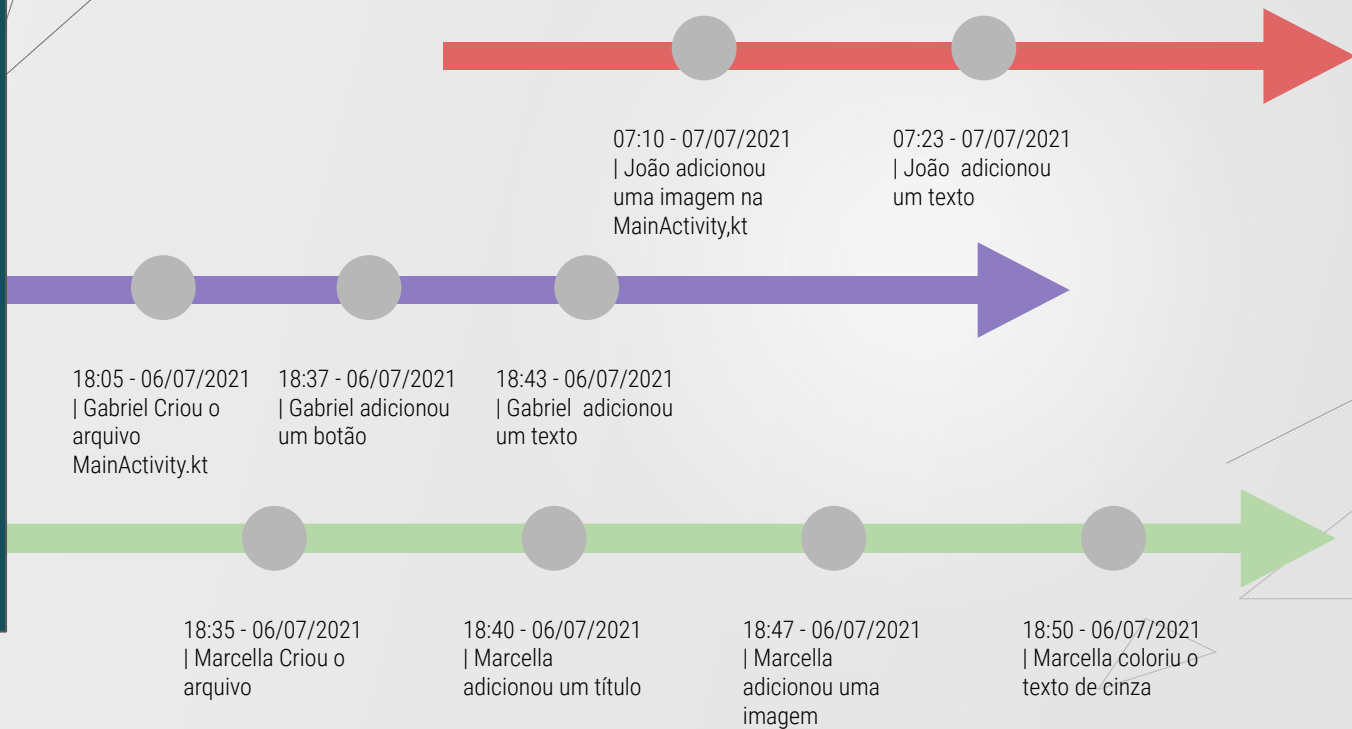
- nano primeiroarquivo.txt

- Adicione no arquivo o texto: meu nome é "seu nome aqui"
 - Salve as alterações
 - Verifique o status do seu repositório
 - Adicione o arquivo no stage
 - Faça um commit
 - Verifique o histórico do seu repositório
- 

GIT


TRABALHO DISTRIBUÍDO

Repositório do time





GIT **TRABALHO DISTRIBUÍDO**

- Necessidade de uma "fonte da verdade"
 - Mudanças locais devem ser sincronizadas com a fonte da verdade para que outras pessoas consigam acessá-las
- 

GITHUB

GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git.






GITHUB

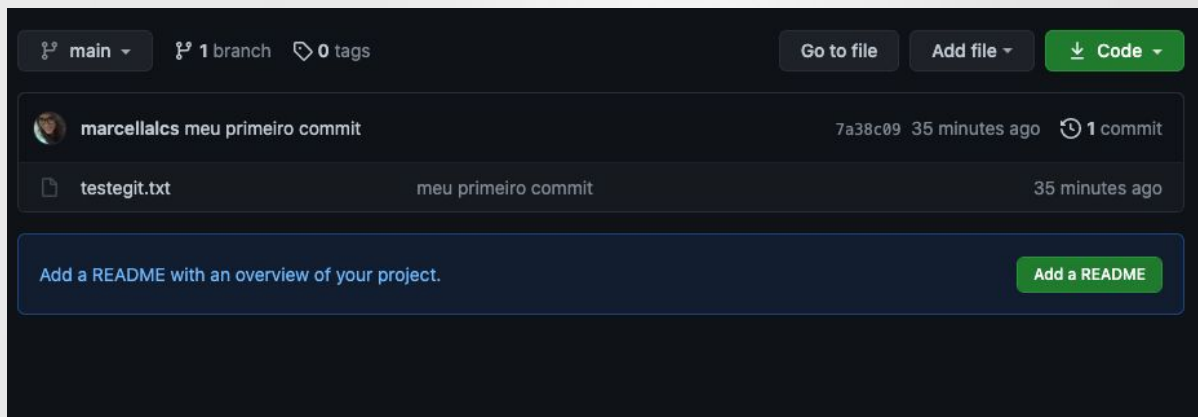
- Crie no Github um repositório
- Sincronize seu repositório local com o repositório remoto que criamos

```
git remote add origin  
https://github.com/suaconta/myproject.git  
git branch -M main  
git push -u origin main
```



GITHUB

Você vai precisar autorizar o acesso ao repo remoto para o push dar certo



02

ORGANIZAÇÃO





ABOUT MY PET

No aplicativo AboutMyPet, você pode mostrar fatos interessantes sobre seu pet ou algum animal que você goste. Este aplicativo exibe um nome, duas imagens e um texto descritivo.





dojo | PRIMEIRA TELA

- Criar novo projeto, AboutMyPet a partir de uma activity vazia
 - Execute o projeto no emulador
 - Inicie um repositório no Github
 - Faça o primeiro push
-



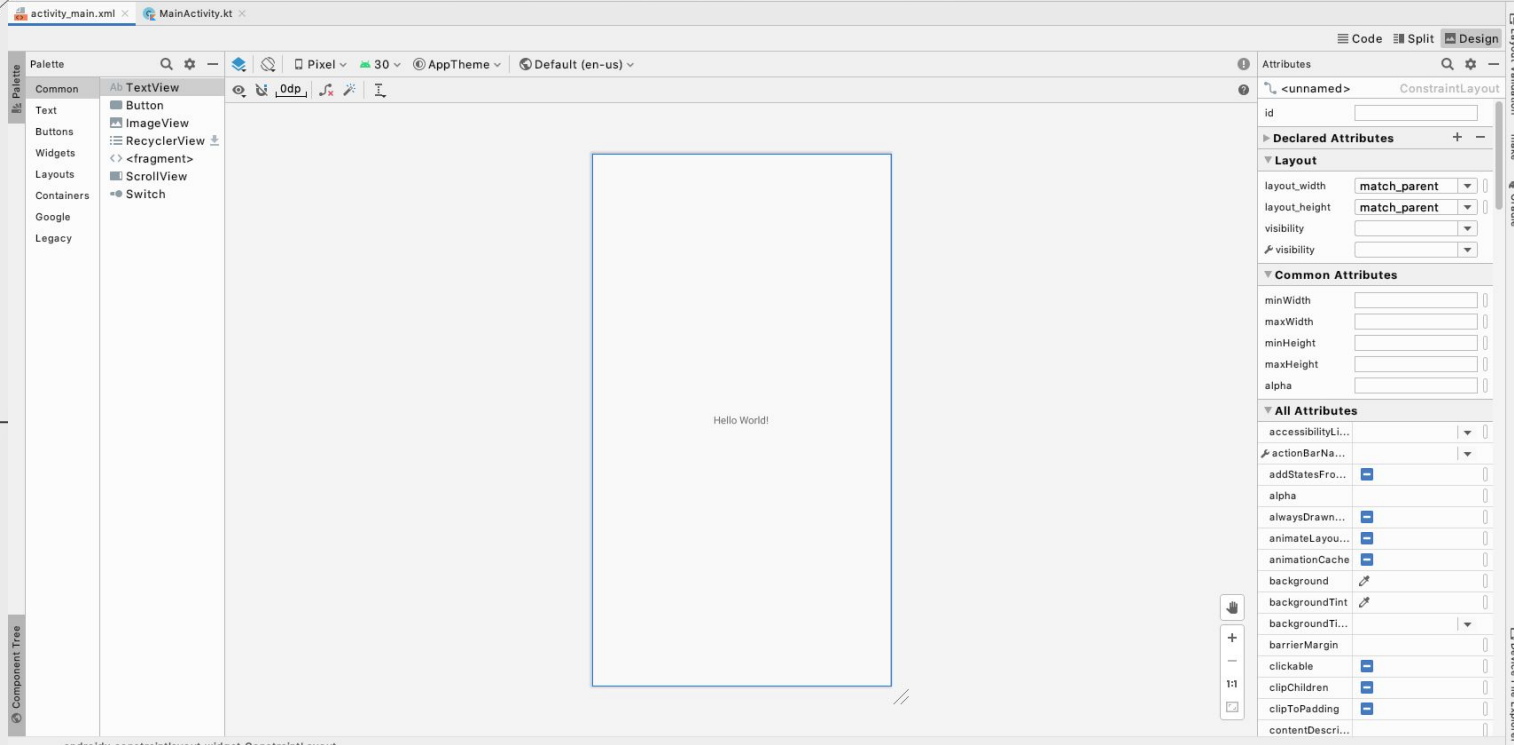


NOVO PROJETO

O template de activity vazia cria uma única activity, **MainActivity.kt**. O template também cria um arquivo de layout chamado **activity_main.xml**. O arquivo de layout tem **ConstraintLayout** como seu **ViewGroup** raiz, e tem um único **TextView** como seu conteúdo.

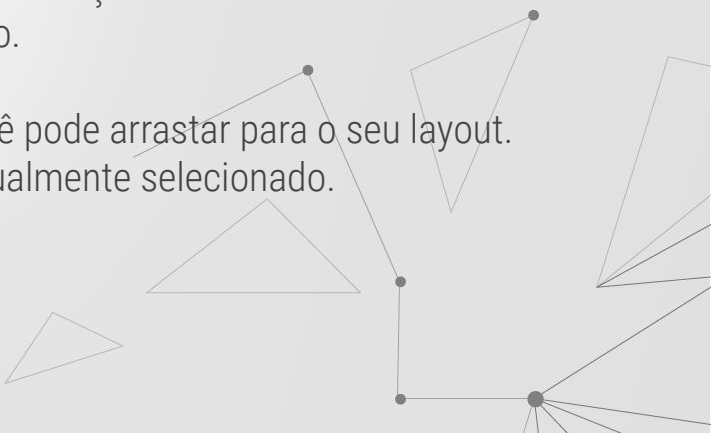


dojo | LAYOUTEDITOR





LAYOUTEDITOR

1. Editor de design: exibe uma representação visual do layout da tela. O editor de design é a parte principal do Editor de Layout.
 2. Barra de ferramentas: fornece botões para configurar a aparência do seu layout no editor de design e para alterar alguns atributos do layout:
 3. Use o Design para uma visualização do mundo real do seu layout.
 4. Use o Blueprint para ver apenas os contornos de cada visualização.
 5. Use Design + Blueprint para ver os dois modos lado a lado.
-
6. Paleta: fornece uma lista de Views e ViewGroups que você pode arrastar para o seu layout.
 7. Atributos: Mostra os atributos da View ou ViewGroups atualmente selecionado.
 8. Exibe a hierarquia de layout como uma árvore de views.
- 

Personalize o tema do seu projeto

- Defina um esquema de cores para seu projeto:
 - <https://material.io/resources/color>
 - Exporte as cores e as incorpore no seu projeto
 - Atualize seu tema para usar as cores criadas
-





dojo | LAYOUTEDITOR

Defina os seguintes atributos para o TextView através do painel Attributes

- ID: petName
 - Text: "nome do pet"
 - textAppearance > textSize: 20sp
 - textAppearance > textColor: @android:color/black
 - Commitar alterações
-

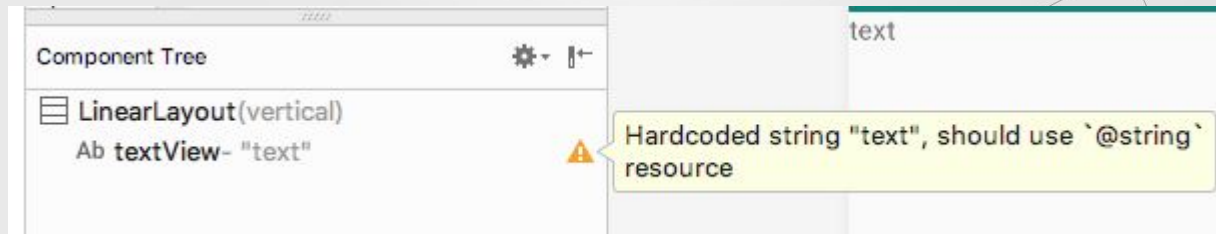


dojo | STRING RESOURCE

Na Árvore de componentes, próximo a TextView, você notará um ícone de aviso. Para ver o texto de aviso, clique no ícone ou aponte para ele, conforme mostrado na imagem abaixo.



- Adicione o resource *pet_name*
- Abra arquivo strings.xml
- Commitar alterações





STRINGS RESOURCES

Separar o conteúdo utilizado na sua aplicação do layout/código ajuda muito na gestão desses recursos.

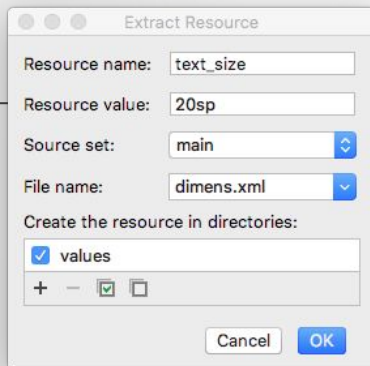
A internacionalização da sua aplicação pode acontecer de forma muito mais simples se você extrair TODOS os textos para o arquivo strings.xml
O reuso e a manutenção desse conteúdo também acontece de forma mais fácil.



dojo | DIMENSION RESOURCE

Da mesma forma que extraímos o nome da string, vamos extrair o tamanho do texto.

- Repare no arquivo `dimens.xml`
- Commitar alterações





DIMENSION RESOURCES

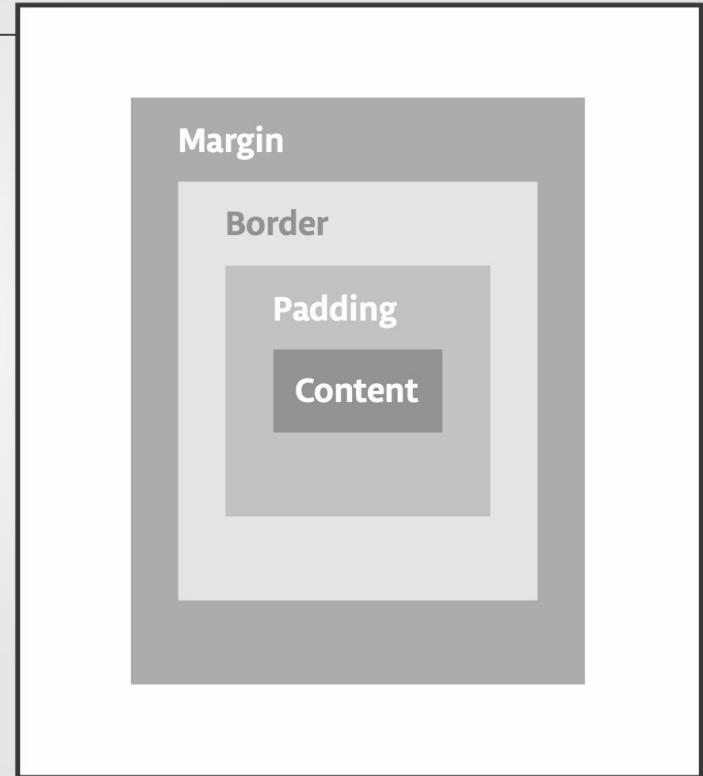
Separar as propriedades de layout utilizadas otimiza o uso e gestão.

Modificar de forma uniforme essas propriedades fica muito mais fácil.
Reaproveitar essas propriedades também



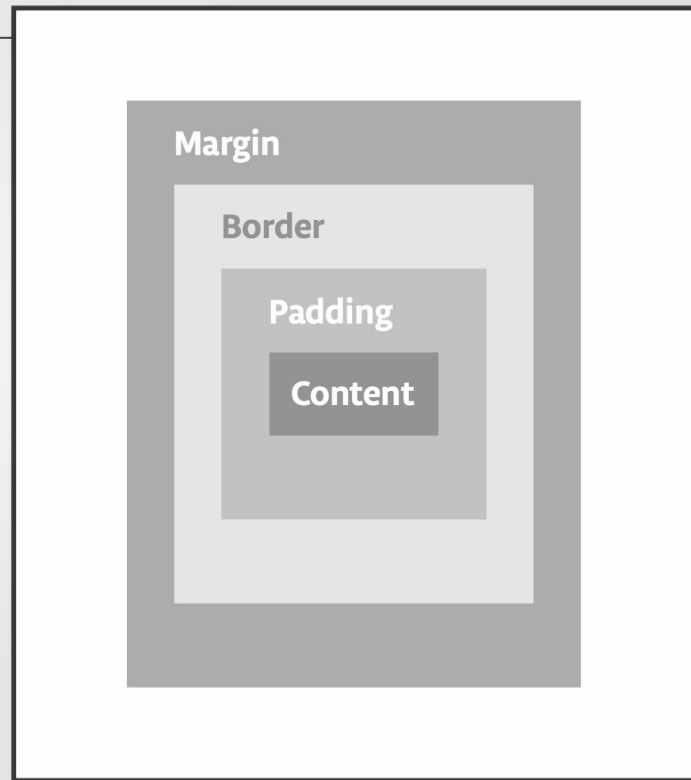
PADDING

Padding é o espaço dentro dos limites de uma view ou elemento. É o espaço entre as bordas da view e o conteúdo da view.



MARGIN

Margem é o espaço adicionado fora das bordas da view. É o espaço da borda da view até seu pai.





dojo | DIMENSION RESOURCE

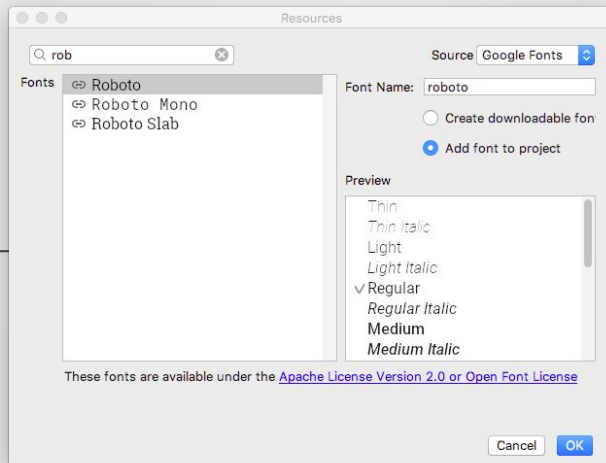
Defina o padding e a margin do TextView a partir dos resources

- small_padding = 8dp
 - default_margin = 16dp
-



dojo | FONT

- Defina a fontFamily como roboto






STYLE

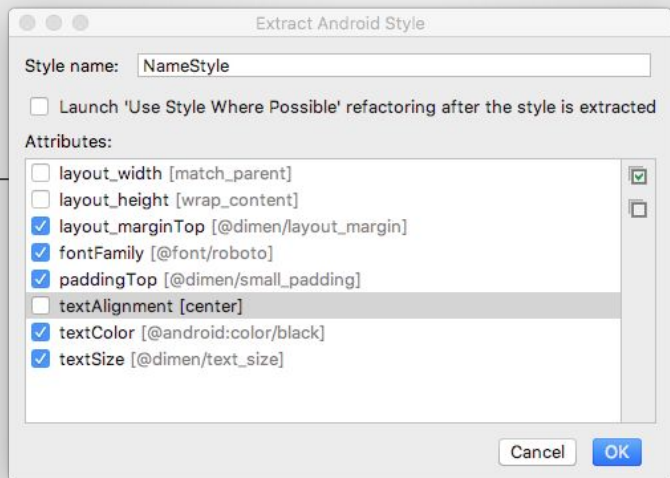
Um **style** é uma coleção de atributos que especificam a aparência e o formato de uma view. Um style pode incluir cor da fonte, tamanho da fonte, cor de fundo, preenchimento, margem e outros atributos comuns.

Você pode extrair a formatação da view do texto do nome em um style e reutilizar o estilo para qualquer número de views em seu aplicativo. Reutilizar um style dá ao seu aplicativo uma aparência consistente quando você tem várias views. Usar styles também permite que você mantenha esses atributos comuns em um único local.



dojo | STYLE

- Extraia os estilos aplicados no componente TextView
- Clique com o botão direito do mouse em TextView na árvore de componentes e selecione Refatorar> Extrair estilo.
- Commitar alterações



03

LINEAR LAYOUT






IMAGE VIEW

A maioria dos aplicativos Android do mundo real consiste em uma combinação imagens, textos e aceitar entrada do usuário na forma de texto ou eventos de clique.

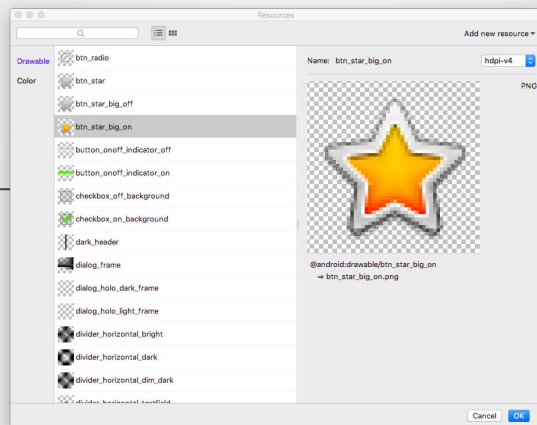
Um ImageView é uma view para exibir recursos de imagem. Por exemplo, um ImageView pode exibir recursos de bitmap, como arquivos PNG, JPG, GIF ou WebP, ou pode exibir um recurso Drawable, como um desenho vetorial.

Existem recursos de imagem que vêm com o Android, como exemplos de ícones, avatares e planos de fundo.



dojo | IMAGEVIEW

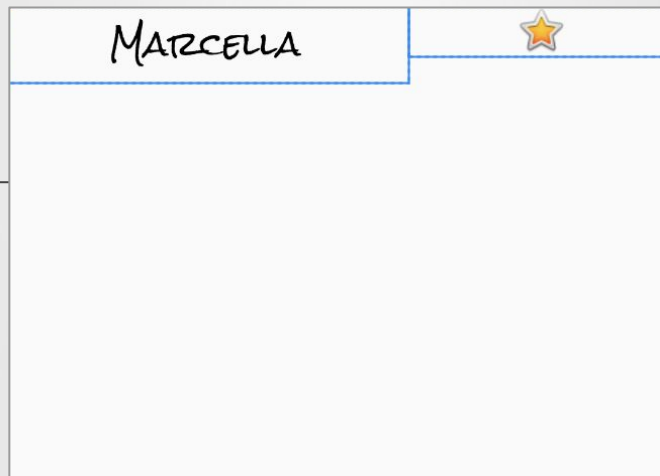
- Adicione a imagem da estrelinha logo abaixo do TextView





dojo | IMAGEVIEW

- Builde a app
- Repare no resultado
- Por que isso aconteceu?
- O que podemos fazer para chegar na disposição abaixo?





VIEWGROUPS

Um ViewGroup é um elemento de interface que pode conter elementos de interface filhos, que são outros elementos e/ou grupos de elementos. As chamadas views que compõem um layout são organizadas como uma hierarquia de views com um grupo de views como raiz.





LINEAR LAYOUT

Todos os filhos de um `LinearLayout` são empilhados um após o outro. Portanto, uma lista vertical terá somente um filho por linha, independentemente da largura, e uma lista horizontal terá altura de apenas uma linha (a altura do filho mais alto, mais preenchimento).

Um `LinearLayout` respeita margens entre filhos e a gravidade (alinhamento à direita, no centro ou à esquerda) de cada filho.



LINEAR LAYOUT

**Vertical
LinearLayout**



**Horizontal
LinearLayout**





ALTURA & LARGURA

Layout_width & Layout_height São os atributos que determinam o tamanho da view.

Eles podem receber os valores:

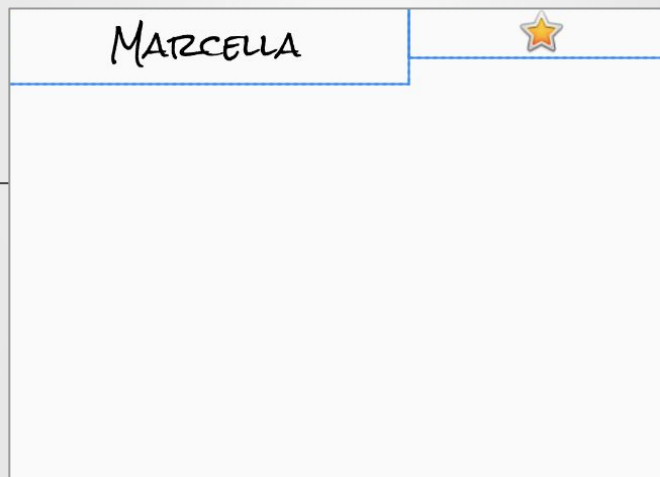
- **match_parent**: o tamanho da view será tão grande quanto a view mãe
- **wrap_content**: o tamanho da view será tão grande quanto o conteúdo dentro dela
- **0dp & weight/height**: o tamanho da view será grande quanto o tamanho disponível e o peso que essa view tem no layout. (essa dimensão deverá ser aplicada a altura se a orientação da view for vertical e a largura se a orientação for horizontal)





dojo | IMAGEVIEW

- Altere o groupview root para Linear Layout
- Ajuste a orientação para Horizontal





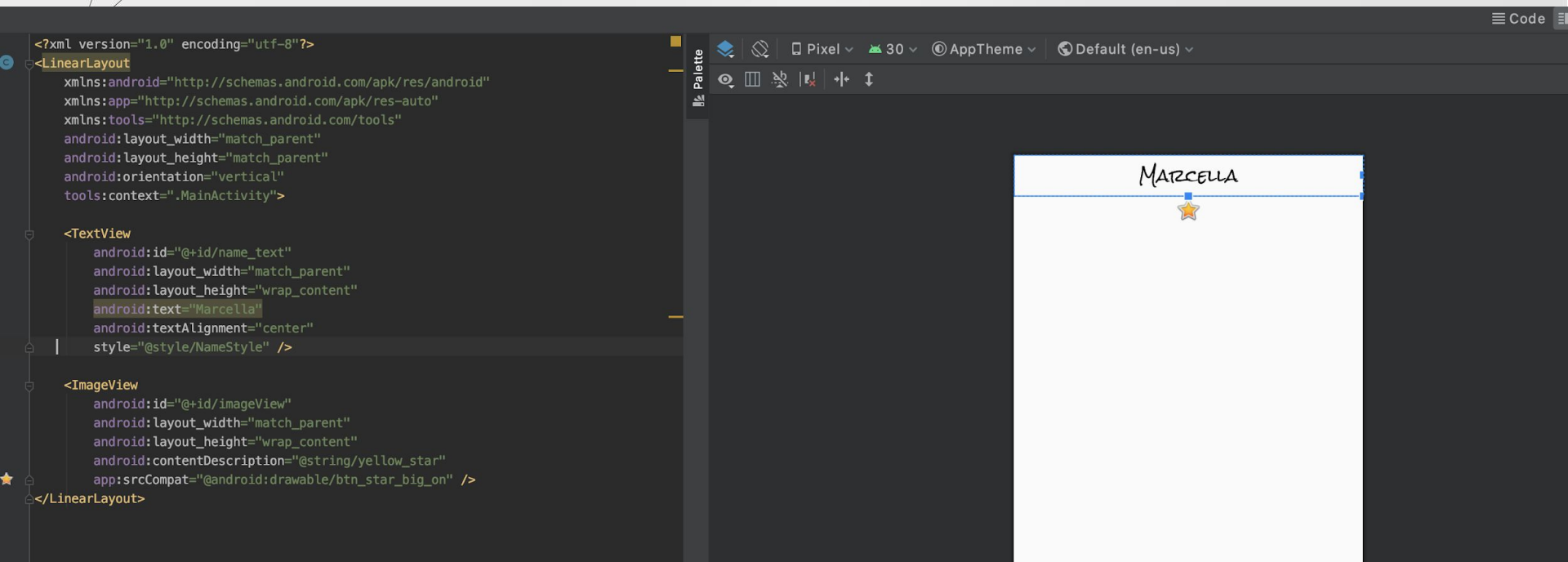
GRAVITY

- Para controlar como o *LinearLayout* alinha todas as views que contém, defina um valor para o atributo ***gravity***.
 - Para controlar como a view filha se posiciona em relação à view mãe, defina um valor para o atributo ***layout_gravity***
-



dojo | IMAGEVIEW

- Ajuste para obter o resultado abaixo:





dojo | IMAGEVIEW

- Renomeie o ImageView
 - Para renomear o id do ImageView, clique com o botão direito em "@ + id / imageView" e selecione Refactor> Rename.
 - Na caixa de diálogo Renomear, defina o id como @+id/star_image. Clique em Refatorar.
 - Commitar alterações
-





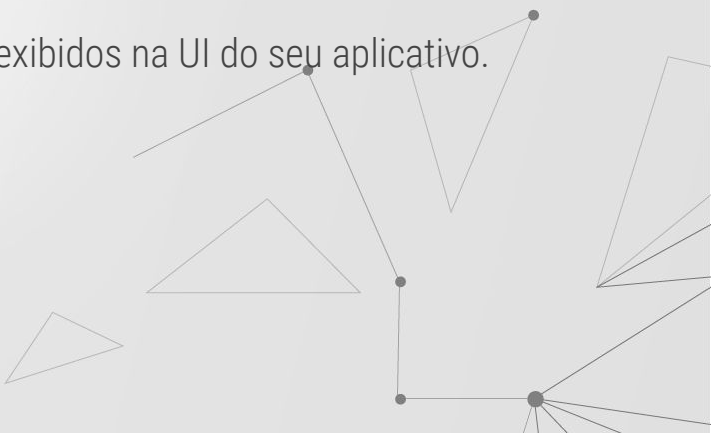
CONTENT DESCRIPTION

Na guia Design, na Árvore de componentes, clique no ícone de aviso ao lado de `star_image`. O aviso é para uma `contentDescription` ausente, que os leitores de tela usam para descrever as imagens para o usuário.





dojo | CONTENT DESCRIPTION

- No painel Atributos, clique nos três pontos ... ao lado do atributo `contentDescription`. A caixa de diálogo Recursos é aberta.
 - Na caixa de diálogo Recursos, selecione Adicionar novo recurso> Novo valor de string. Defina o campo Nome do recurso como `yellow_star_description` e defina o campo valor do recurso como "*estrela amarela*". Clique OK.
 - Use o painel Atributos para adicionar uma margem superior de 16 dp (que é @dimen / layout_margin) à `yellow_star`, para separar a imagem da estrela do nome.
 - Execute seu aplicativo. Seu nome e a imagem da estrela são exibidos na UI do seu aplicativo.
-
- Ative o leitor de tela e veja o resultado
 - Commitar alterações
- 

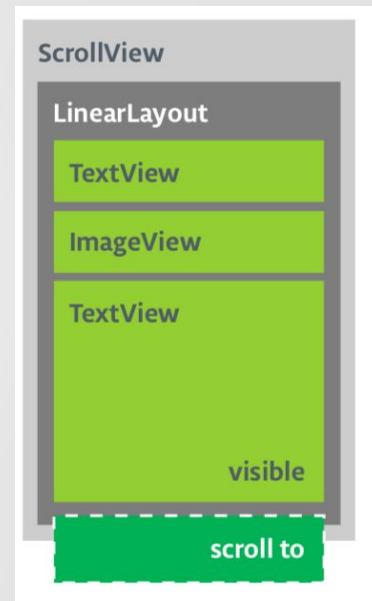
03

SCROLLVIEW



SCROLLVIEW

Um **ScrollView** é um **ViewGroup** que permite que a hierarquia de visualização colocada dentro dele seja rolada. Uma visualização de rolagem pode conter apenas uma outra View, ou ViwGroup, como filho. A view filho é comumente um LinearLayout. Dentro de um LinearLayout, você pode adicionar outras views.





dojo | SCROLLVIEW

- Nesse app, a área com a descrição do seu pet será "rolável".
 - Adicione um ScrollView logo abaixo da estrela
 - Adicione um TextView, de id: text_pet_description e crie uma string de resource para preencher o conteúdo
 - Adicione nessa TextView um style com as seguintes propriedades
 - Id: style_bio
 - fontFamily: Roboto
 - textSize: 14sp
-





dojo | SCROLLVIEW

- Preencha a string `text_pet_description` com uma breve descrição do seu pet (pelo menos 3 linhas)

Dicas:

- Use `\n` para indicar uma quebra de linha.
 - Se você usar um apóstrofo, deverá escapar dele com uma barra invertida.
 - Para texto em negrito use ` ... ` e para texto em itálico use `<i> ... </i>`. Por exemplo: "Este texto está `` negrito `` e este texto está `<i>` itálico `</i>`."
-





dojo | SCROLLVIEW

- Execute o app e observe o resultado
 - Existe algum outro detalhe visual que podemos melhorar?
 - Commitar alterações
-





REVISÃO

Questão 1

Qual das opções a seguir é um grupo de exibição?

- ☐ EditText
 - ☐ LinearLayout
 - ☐ TextView
 - ☐ Botão
-





REVISÃO

Questão 2

Qual das seguintes hierarquias de visualização não é válida?

- LinearLayout > TextView, TextView, ImageView
 - ScrollView > LinearLayout > TextView, Botão, Botão, ScrollView > TextView
 - TextView > TextView, ImageView, ScrollView
-





REVISÃO

Questão 3

Estilos são recursos definidos em styles.xml. Usando estilos, você pode definir cores, fontes, tamanho do texto e muitas outras características de uma visualização. Verdadeiro ou falso?

- ☐ Verdadeiro
 - ☐ Falso
-





REVISÃO

Questão 4

Um ScrollView é um grupo de visualização que pode conter qualquer número de visualizações ou grupos de visualização como seus filhos. Verdadeiro ou falso?

- ☐ Verdadeiro
 - ☐ Falso
-





REVISÃO

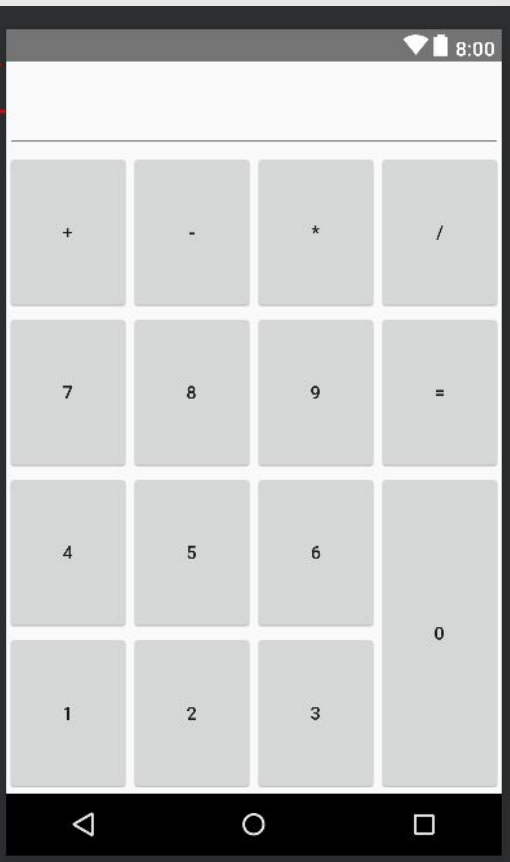
Questão 5

Qual elemento da IU pode ser usado para exibir imagens em seu aplicativo?

- ☐ TextView
 - ☐ ImageView
 - ☐ Botão
 - ☐ ScrollView
-



MARGIN



DESAFIO

Palavras chave da aula de hoje:

- LINEAR LAYOUT
- SCROLLVIEW
- CONTENT DESCRIPTION
- VIEWGROUP
- RESOURCES
- STYLES

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution.