

The background features a complex network of thin grey lines connecting various points, forming a web-like structure. Scattered throughout are numerous 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

CHAINS
Conexão entre views

01

**LINHAS GUIA E
BARREIRAS**
UI mais responsivas

02

**OUTROS GROUP
VIEWS**
Relative Layout, Frame
Layout entre outros

03

**ADD
COMPORTAMENTO**
Colorindo o
ColorMyViews

04

Aula 10 AGENDA



01

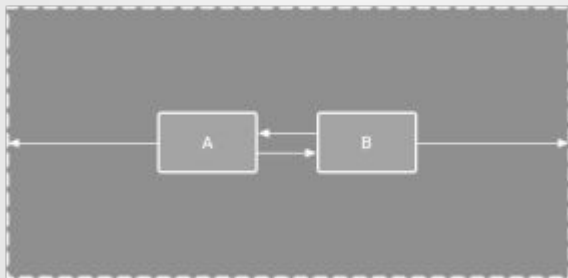
CHAINS



CHAINS

Uma **chain** é um grupo de views vinculadas umas às outras com restrições **bidirecionais**. As views dentro de uma chain podem ser distribuídas vertical ou horizontalmente.

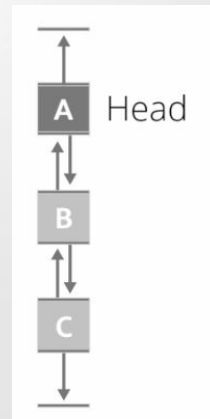
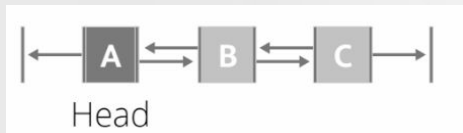
Por exemplo, o diagrama a seguir mostra duas visualizações restritas uma à outra, o que cria uma cadeia horizontal.



CHAINS

Head da chain

A primeira view em uma cadeia é chamada de cabeça da cadeia. Os atributos definidos no topo da cadeia controlam, posicionam e distribuem todas as visualizações na cadeia. Para chains horizontais, a cabeça é a vista mais à esquerda. Para correntes verticais, a cabeça é a vista mais superior. Em cada um dos dois diagramas abaixo, "A" é o chefe da cadeia.



CHAINS

Chain style

Os estilos de cadeia definem a maneira como as vistas encadeadas são espalhadas e alinhadas. Você estiliza uma corrente atribuindo um atributo de estilo de corrente, adicionando peso ou definindo bias nas views. Existem quatro estilos de cadeias:

Spread: este é o estilo padrão. As views são distribuídas uniformemente no espaço disponível, após as margens serem contabilizadas.



CHAINS

Chain style

Spread inside: A primeira e a última views são anexadas ao pai em cada extremidade da cadeia. O resto das visualizações são distribuídas uniformemente no espaço disponível.



Packed: As views são empacotadas juntas, após as margens serem contabilizadas. Você pode então ajustar a posição de toda a corrente mudando o bias da visão da cabeça da corrente.



CHAINS

Chain style

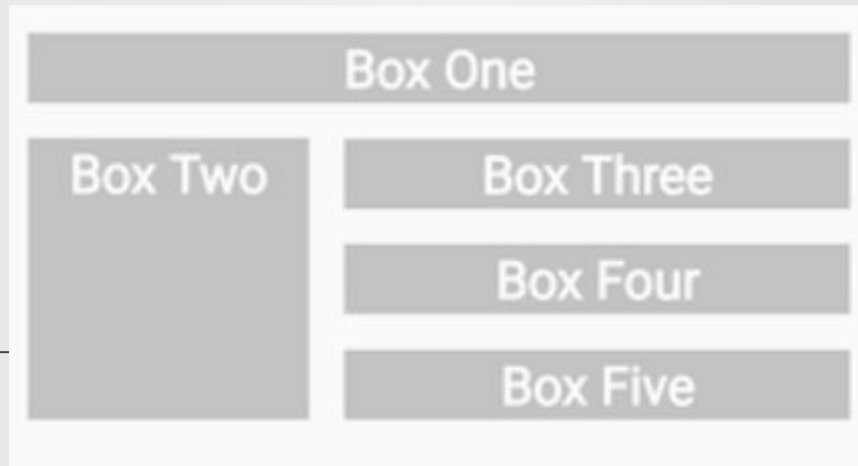
Weighted: as views são redimensionadas para preencher todo o espaço, com base nos valores definidos nos atributos `layout_constraintHorizontal_weight` ou `layout_constraintVertical_weight`. Por exemplo, imagine uma cadeia contendo três visualizações, A, B e C. A visualização A usa um peso de 1. As visualizações B e C usam cada uma um peso de 2. O espaço ocupado pelas visualizações B e C é o dobro da visualização A, como mostrado abaixo






dojo | CHAINS

O efeito que queremos criar é o seguinte:



Repare no posicionamento das caixas três, quatro e cinco.



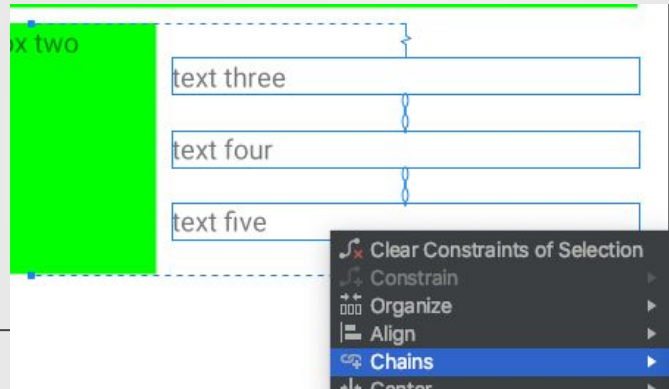
dojo | CHAINS

- Crie as três TextViews (box three, four e five), extraia o conteúdo de texto para strings.xml e estilize

Attribute	Top text view	Middle text view	Bottom text view
ID	box_three_text	box_four_text	box_five_text
text	@string/box_three	@string/box_four	@string/box_five
style	@style/whiteBox	@style/whiteBox	@style/whiteBox

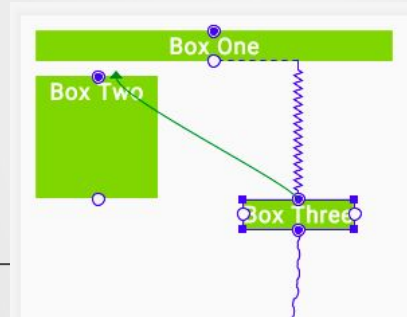
dojo | CHAINS

- Selecione as caixas três, quatro e cinco, clique com o botão direito, chains, create vertical chain



dojo | CHAINS

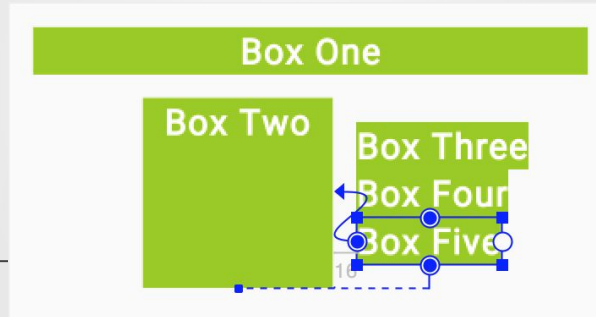
- Adicione uma constraint que se estende do topo da Caixa Três até o topo da Caixa Dois. Isso remove a constraint superior existente e a substitui pela nova constraint. Você não precisa excluir a constraint explicitamente.



- Adicione uma constraint da parte inferior da Caixa Cinco até a parte inferior da Caixa Dois.

dojo | CHAINS

- Crie uma constraint conectando o lado esquerdo da Caixa Três ao lado direito da Caixa Dois. Repita para o Quadro Quatro e o Quadro Cinco, restringindo o lado esquerdo de cada um ao lado direito do Quadro Dois.



- Restrinja o lado direito de cada uma das três visualizações de texto ao lado direito do layout.

dojo | CHAINS

- Usando margins, deixe o posicionamento final das caixas assim:
- Commite as alterações e faça push :)



02

**LINHAS GUIA E
BARREIRAS**

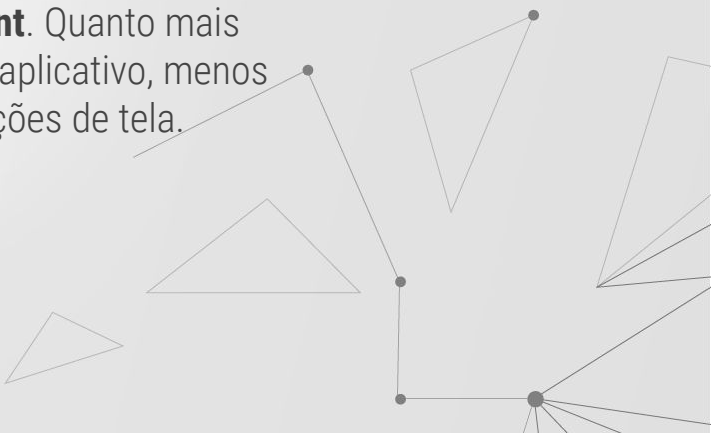




UI RESPONSIVAS

Ao desenvolver aplicativos do mundo real, use **restrições flexíveis para a altura e largura** de seus elementos de interface do usuário, sempre que possível.

Por exemplo, use **match_constraint** ou **wrap_content**. Quanto mais elementos de IU de tamanho fixo você tiver em seu aplicativo, menos adaptável será seu layout para diferentes configurações de tela.





LINHAS GUIA

Você pode adicionar uma linha guia vertical ou horizontal à qual seja possível posicionar as views.

A posição da guia pode ser definida em:

- dp
 - %
 - Borda do layout
-





dojo | LINHAS GUIA

- Ajuste o tamanho da box_two: altura = até uma linha guia posicionada na metade da tela. Largura = até uma linha guia posicionada em 30% da tela
-



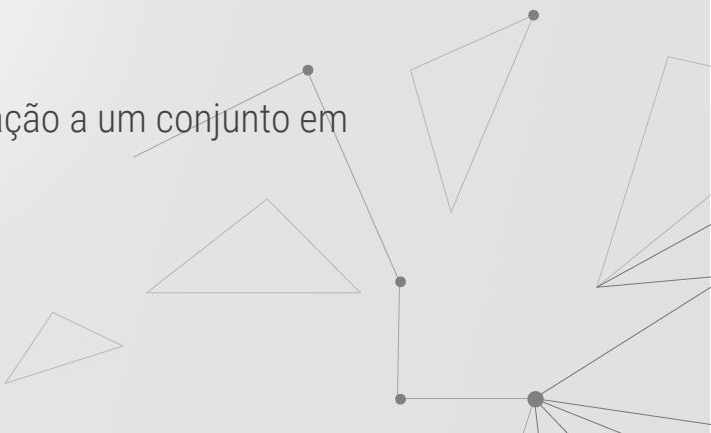


BARREIRAS

Assim como uma linha guia, uma barreira é uma linha invisível à qual pode-se restringir views.

A barreira não define a própria posição; a posição da barreira depende da posição das views contidas nela.

Isso é útil quando você quer restringir uma visualização a um conjunto em vez de a uma visualização específica.

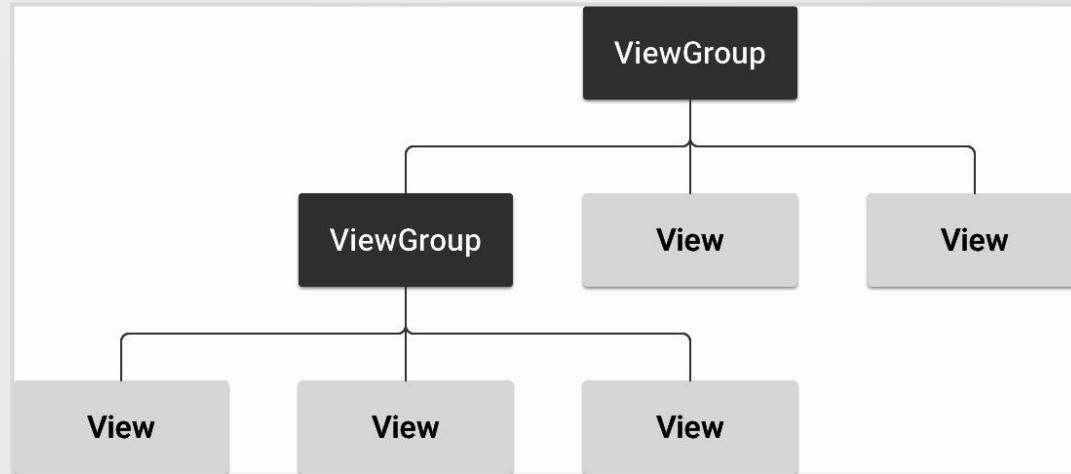


03

OUTROS GROUPVIEWS



LAYOUTS





LAYOUTS

Linear Layout

Views filhas numa única direção: vertical ou horizontal

Constraint Layout

View dispostas de acordo com as relações entre views irmãs e layout pai

Relative Layout

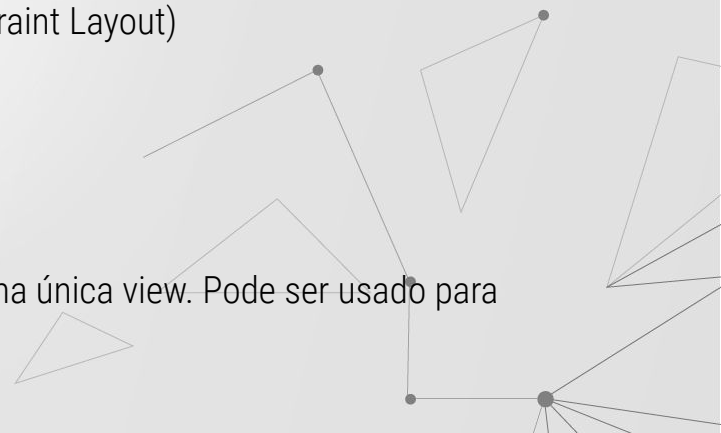
Dispõe views filhas de forma relativa (irmão mais velho do Constraint Layout)

Table Layout

Agrupa views numa estrutura de linhas e colunas

Frame Layout

É um espaço reservado na tela que você pode usar para exibir uma única view. Pode ser usado para estruturar layouts com view sobrepostas



Questão 1

Em um `ConstraintLayout`, qual das opções a seguir descreve as constraints necessárias para manter uma visualização no lugar durante o tempo de execução?

- ☐ Duas restrições horizontais.
 - ☐ Uma restrição vertical.
 - ☐ Pelo menos uma restrição horizontal e uma vertical.
 - ☐ Não há necessidade de restringir a visão.
-



Questão 2

Qual dos seguintes tipos de constraint expande uma visualização apenas o necessário para ajustar seu conteúdo?

- ☐ Wrap content
 - ☐ Match constraint
 - ☐ Fixed constraint
 - ☐ Baseline constraint
-



Questão 3

O inspetor de visualização está disponível apenas para visualizações em _____.

- ☐ Um ConstraintLayout
 - ☐ Um ConstraintLayout ou LinearLayout
 - ☐ Qualquer ViewGroup
 - ☐ Um LinearLayout
-





REVISÃO

Questão 4

Uma chain é um grupo de views vinculadas entre si por _____.

- ☐ Restrições superior e inferior
 - ☐ Restrições bidirecionais
 - ☐ Restrições direita e esquerda
 - ☐ Restrições de linha de base
-



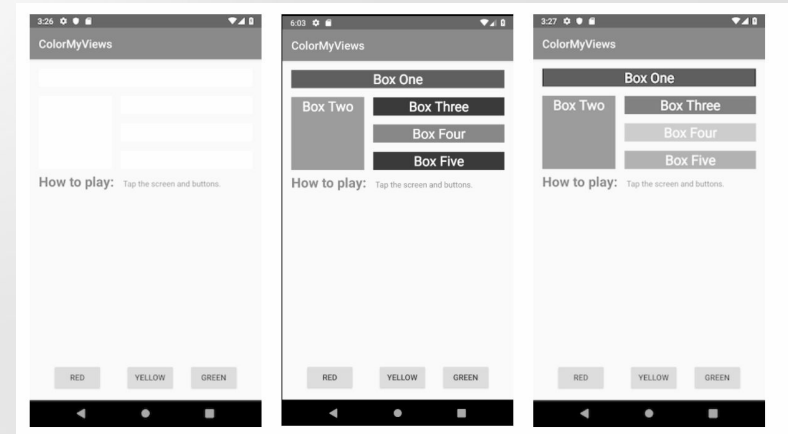
04

ADD COMPORTAMENTO

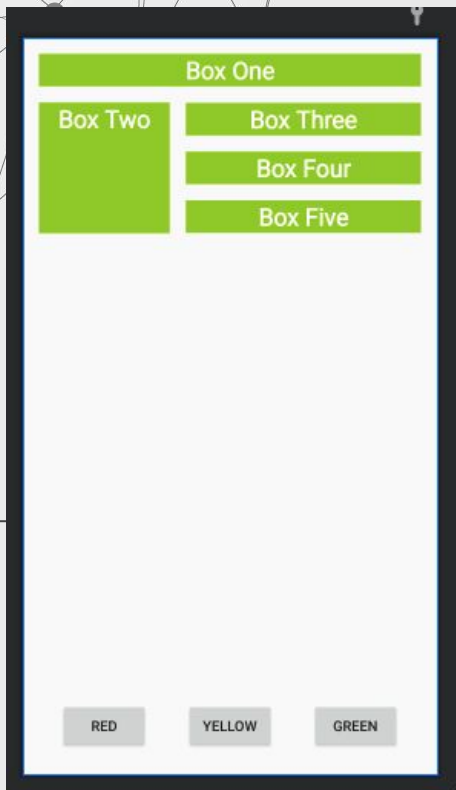


COLOR MY VIEWS

O aplicativo ColorMyViews é inspirado no artista holandês Piet Mondrian. Ele inventou um estilo de pintura chamado neoplasticismo, que usa apenas linhas verticais e horizontais e formas retangulares em preto, branco, cinza e cores primárias.



dojo / BOTÕES



- Adicione ao layout 3 botões
 - `redButton`
 - `yellowButton`
 - `greenButton`
 - Configure os botões para ficarem alinhados entre si verticalmente
 - Configure os botões para que a distância entre eles seja sempre distribuída igualmente
-
- Por fim, conecte os botões com a parte inferior da tela, adicionando uma margem de 24dp

dojo / BOTÕES

- Crie as seguintes cores no arquivo colors.xml
 - `<color name="grey">#BABABA</color>`
 - `<color name="green">#12C700</color>`
 - `<color name="red">#E54304</color>`
 - `<color name="yellow">#FFC107</color>`
- Cora os botões de acordo com seu respectivo nome

