



PADO
Labs

INTRODUÇÃO AO DESENVOLVIMENTO MOBILE



Aula II - Views e Layout

Contato



Professor



David Krepsky



david@hausenn.com.br



DKrepsky

LinearLayout Horizontal



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="Item 1"
        android:textColor="@color/white" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="Item 2"
        android:textColor="@color/white" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="Item 3"
        android:textColor="@color/white" />

</LinearLayout>
```

Layout

Item 1Item 2Item 3

LinearLayout Vertical



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="Item 1"
        android:textColor="@color/white" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="Item 2"
        android:textColor="@color/white" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="Item 3"
        android:textColor="@color/white" />

</LinearLayout>
```

Layout

Item 1
Item 2
Item 3

Wrap Content



O tamanho do elemento é o tamanho do conteúdo do elemento

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:background="@color/black"  
    android:text="Item 1"  
    android:textColor="@color/white" />
```



Match Parent



O tamanho do elemento é o tamanho pai

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="@color/black"  
    android:text="Item 1"  
    android:textColor="@color/white" />
```



0dp e Weight com espaços diferentes



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">
```

```
<TextView
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="wrap_content"
    android:background="@color/black"
    android:text="Item 1"
    android:textColor="@color/white" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/black"
    android:text="Item 2"
    android:textColor="@color/white" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/black"
    android:text="Item 3"
    android:textColor="@color/white" />
```

```
</LinearLayout>
```

Layout

Item 1

Item 2Item 3

0dp e Weight com espaços iguais



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <TextView
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="Item 1"
        android:textColor="@color/white" />

    <TextView
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="Item 2"
        android:textColor="@color/white" />

    <TextView
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="Item 3"
        android:textColor="@color/white" />

</LinearLayout>
```



Espaçamento entre os itens



```
<TextView
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="wrap_content"
    android:background="@color/black"
    android:text="Item 1"
    android:textColor="@color/white" />

<Space
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="wrap_content"/>

<TextView
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="wrap_content"
    android:background="@color/black"
    android:text="Item 2"
    android:textColor="@color/white" />

<Space
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="wrap_content"/>

<TextView
    android:layout_width="0dp"
    android:layout_weight="1"
    android:layout_height="wrap_content"
    android:background="@color/black"
    android:text="Item 3"
    android:textColor="@color/white" />
```



Gravity Centro Horizontal



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:gravity="center_horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="Item 1"
        android:textColor="@color/white" />

</LinearLayout>
```

Layout

Item 1

Gravity Centro Vertical



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="Item 1"
        android:textColor="@color/white" />

</LinearLayout>
```

Layout

Item 1

Gravity Centro



```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    android:gravity="center"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/black"
        android:text="Item 1"
        android:textColor="@color/white" />

</LinearLayout>
```

Layout

Item 1



ATIVIDADE I

Catálogo da linha digital
PADO

App ToolBar



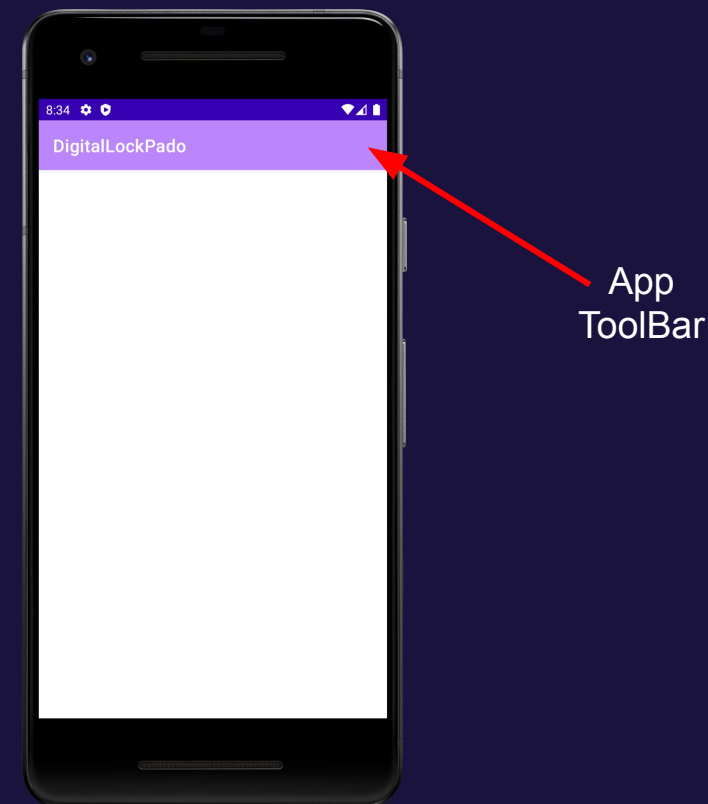
Nosso objetivo é implementar uma ToolBar customizada para o aplicativo PADO.



App ToolBar



A App ToolBar é um View dentro do aplicativo que oferece informações úteis para os usuários e auxilia na navegação, principalmente quando combinada com menus.



App ToolBar



Para customizar a App ToolBar, o primeiro passo é desabilitar a barra padrão do sistema.

No arquivo *res > values > themes > themes.xml*, troque o tema base de ***DarkActionBar*** para ***NoActionBar***



Sem
ToolBar

```
<!-- Base application theme. -->  
<style name="Theme.DigitalLockPado" parent="Theme.MaterialComponents.DayNight.NoActionBar">
```

App ToolBar



Adicione um *View* da classe ***MaterialToolbar*** ao layout, defina as dimensões e restrições

```
<com.google.android.material.appbar.MaterialToolbar
    style="@style/Widget.MaterialComponents.Toolbar.Primary"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">
</com.google.android.material.appbar.MaterialToolbar>
```

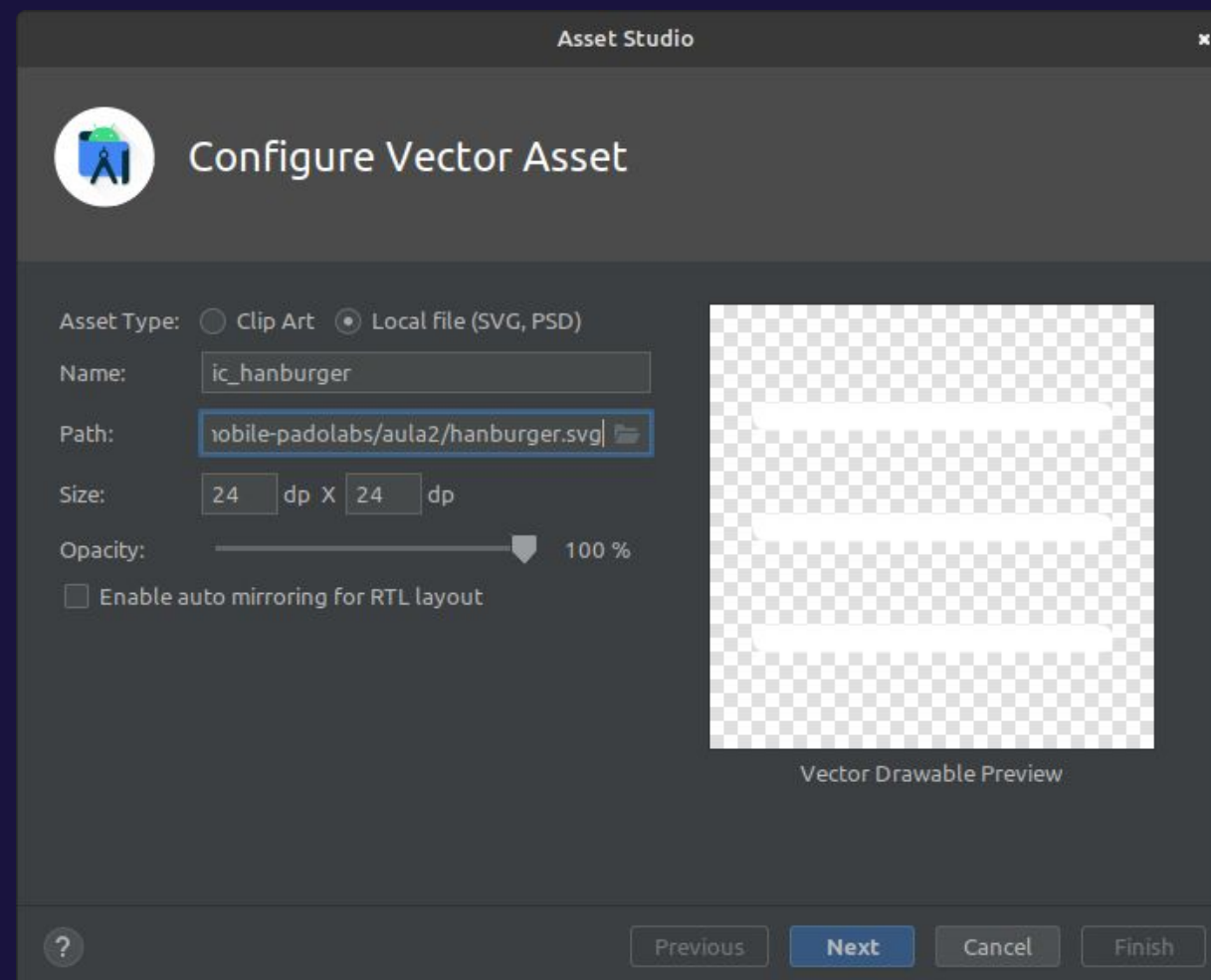


App ToolBar



Adicione os ícones em SVG ao projeto clicando com o botão direito no projeto **New** > **Vector Asset**.

Faça isso para todos os ícones SVG da pasta de imagens disponibilizada.



App ToolBar



Adicione como ícone de navegação a imagem do hamburger.

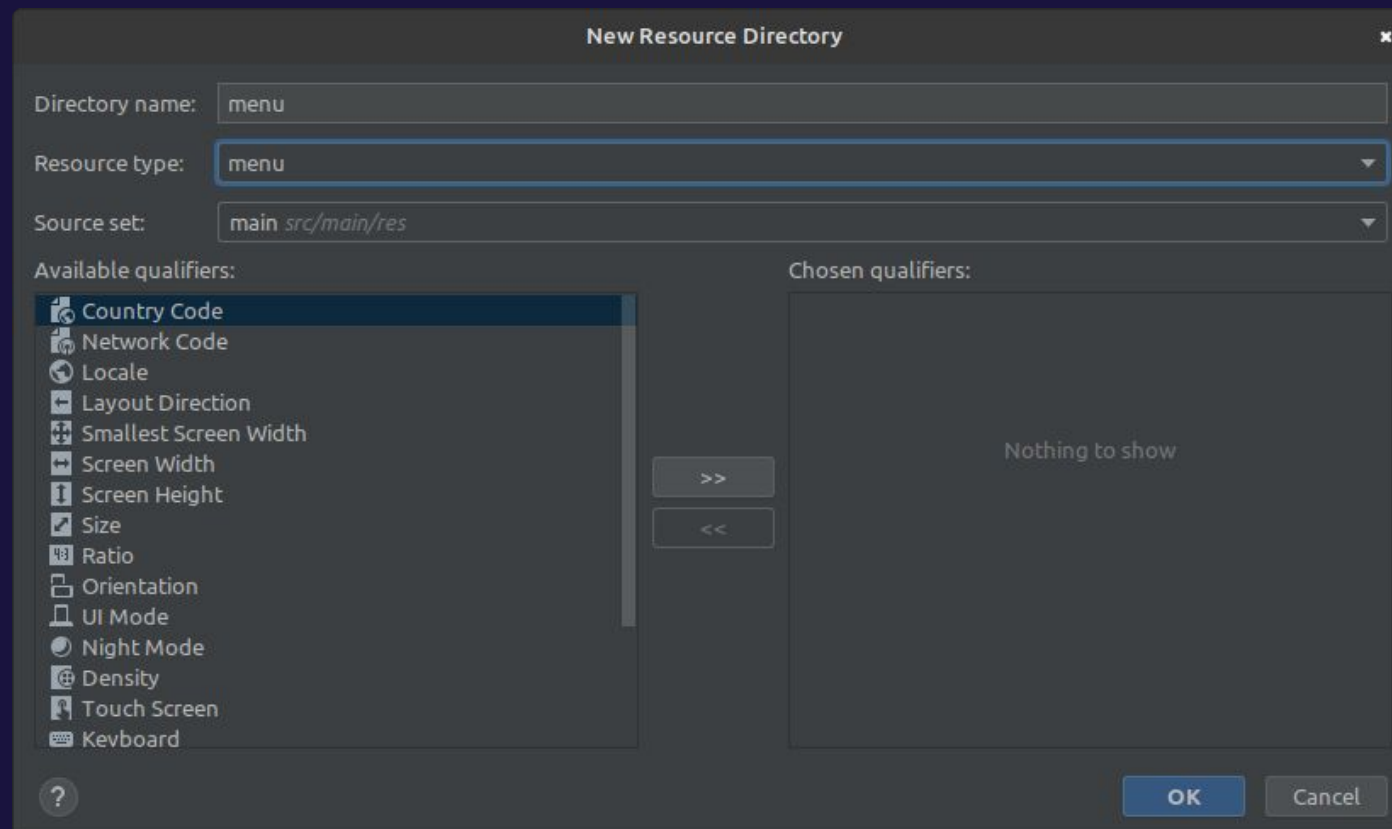
```
<com.google.android.material.appbar.MaterialToolbar
    style="@style/Widget.MaterialComponents.Toolbar.Primary"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:navigationIcon="@drawable/ic_hamburger"
>
</com.google.android.material.appbar.MaterialToolbar>
```



App ToolBar



Adicione um diretório de menus indo em File > New > Android Resource Directory e preencha conforme a imagem abaixo.

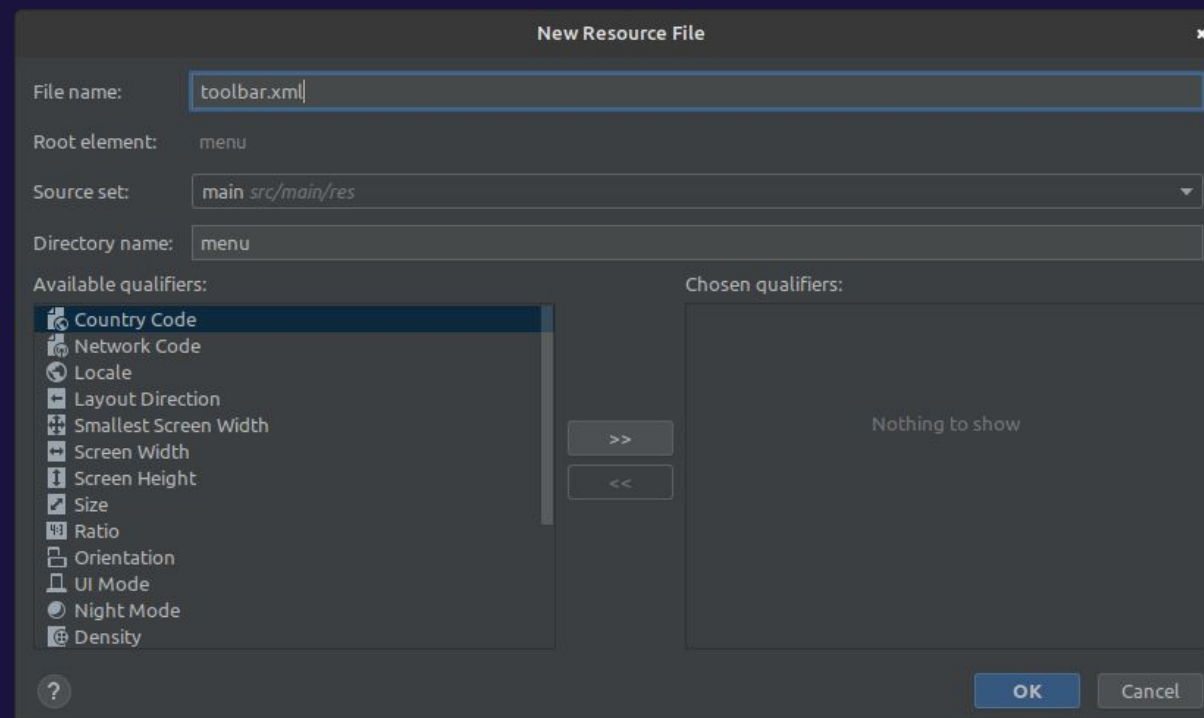
A screenshot of the "New Resource Directory" dialog box in Android Studio. The dialog has a title bar with a close button. It contains three input fields at the top: "Directory name:" with the text "menu", "Resource type:" with a dropdown menu showing "menu", and "Source set:" with a dropdown menu showing "main src/main/res". Below these fields are two lists. The "Available qualifiers:" list on the left contains 16 items, each with a small icon: Country Code, Network Code, Locale, Layout Direction, Smallest Screen Width, Screen Width, Screen Height, Size, Ratio, Orientation, UI Mode, Night Mode, Density, Touch Screen, and Keyboard. The "Country Code" item is currently selected. Between the two lists are two buttons: ">>" and "<<". The "Chosen qualifiers:" list on the right is empty and contains the text "Nothing to show". At the bottom right of the dialog are "OK" and "Cancel" buttons. A help icon (?) is located at the bottom left.

App ToolBar



Clique com o botão direito no diretório de menu (res > menu) e adicione um novo arquivo de menu (New > Menu Resource File).

Nomeie o arquivo para *toolbar.xml*



App ToolBar



Cada ícone é um item do menu, assim adicione 2 itens, um para as notificações e outro para o carrinho.

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto">
4      <item
5          android:id="@+id/menu_item_notifications"
6          android:contentDescription="Bell icon"
7          android:icon="@drawable/ic_bell"
8          android:title="Notifications"
9          app:showAsAction="ifRoom" />
10     <item
11         android:id="@+id/menu_item_cart"
12         android:contentDescription="Cart Icon"
13         android:icon="@drawable/ic_cart"
14         android:title="Cart"
15         app:showAsAction="ifRoom" />
16 </menu>
```

Linha importante!
Permite adicionar os ícones
sem o texto na toolbar

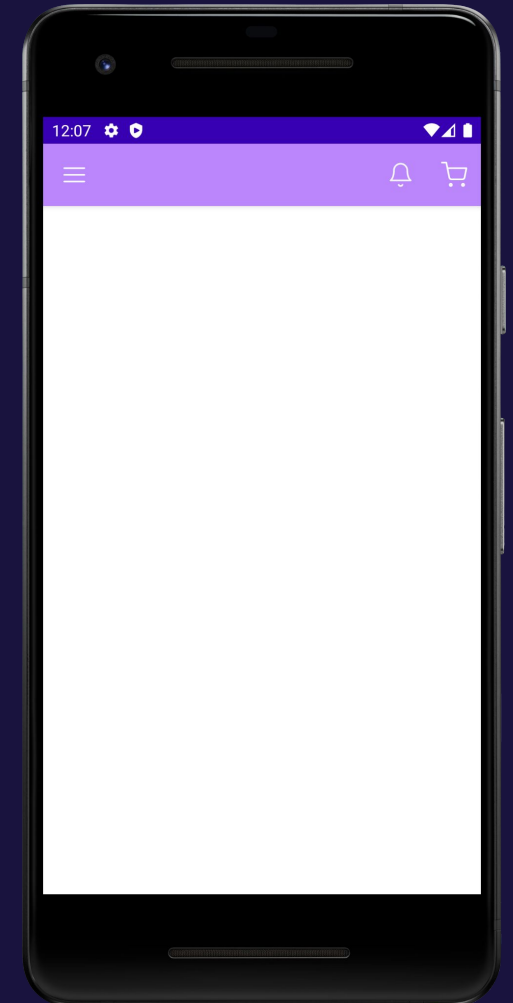
App ToolBar



Adicione o menu a ToolBar

```
<com.google.android.material.appbar.MaterialToolbar
    style="@style/Widget.MaterialComponents.Toolbar.Primary"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:menu="@menu/toolbar"
    app:navigationIcon="@drawable/ic_hamburger">

</com.google.android.material.appbar.MaterialToolbar>
```



App ToolBar



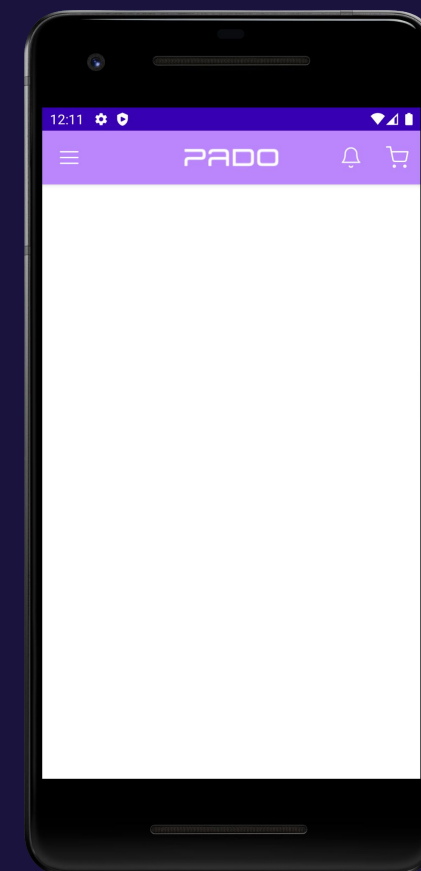
Adicione um *ImageView* para a imagem do logo

Centralize e escale para que fique bonito :D

```
<com.google.android.material.appbar.MaterialToolbar
    style="@style/Widget.MaterialComponents.Toolbar.Primary"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:menu="@menu/toolbar"
    app:navigationIcon="@drawable/ic_hamburger">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:scaleX="0.6"
        android:scaleY="0.6"
        android:src="@drawable/logo"
        android:contentDescription="@string/content_description_logo_pado" />

</com.google.android.material.appbar.MaterialToolbar>
```



App ToolBar



Para adicionarmos *EventListeners* aos eventos da ToolBar, adicione um id.

```
<com.google.android.material.appbar.MaterialToolbar
    style="@style/Widget.MaterialComponents.Toolbar.Primary"
    android:id="@+id/mainToolBar"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:menu="@menu/toolbar"
    app:navigationIcon="@drawable/ic_hamburger">
```

App ToolBar



No arquivo MainActivity.kt, adicione um handler para o evento `setNavigationOnClickListener`.

Utilize um toast para identificar quando o botão foi pressionado.

```
val mainToolBar = findViewById<MaterialToolbar>(R.id.mainToolBar)
mainToolBar.setNavigationOnClickListener { it: View!
    Toast.makeText(context: this,
        text: "Hamburger pressionado!!!",
        Toast.LENGTH_SHORT)
        .show()
}
```

App ToolBar



```
mainToolBar.setOnMenuItemClickListener { menuItem ->
    when (menuItem.itemId) {
        R.id.menu_item_notifications -> {
            Toast.makeText(
                context: this,
                text: "Ir para a página de notificações!!!",
                Toast.LENGTH_SHORT
            ).show()

            true ^setOnMenuItemClickListener
        }
        R.id.menu_item_cart -> {
            Toast.makeText(
                context: this,
                text: "Ir para a página do carrinho!!!",
                Toast.LENGTH_SHORT
            ).show()

            true ^setOnMenuItemClickListener
        }
        else -> false ^setOnMenuItemClickListener
    }
}
```

Adicione um handler para o evento `setOnMenuItemClickListener` para tratar os cliques no carrinho e notificações.

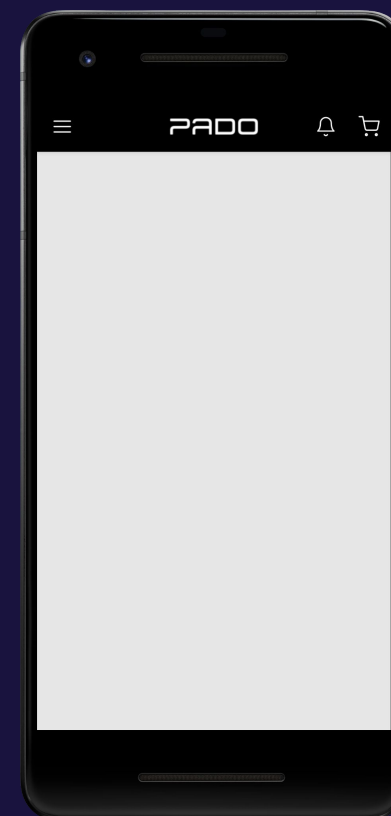
Utilize um toast para identificar quando o botão foi pressionado.

App ToolBar



O último passo é ajustar a cor do background

```
<com.google.android.material.appbar.MaterialToolbar
    android:id="@+id/mainToolBar"
    style="@style/Widget.MaterialComponents.Toolbar.Primary"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:background="@color/black"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:menu="@menu/toolbar"
    app:navigationIcon="@drawable/ic_hamburger">
```



Página com Scroll

Como podemos ver no FIGMA, o layout da página inicial não cabe no espaço disponível na tela.

Para que o usuário possa rolar a tela e visualizar todo o conteúdo, devemos adicionar um View do tipo ***ScrollView***, abaixo da ToolBar.

Todo o conteúdo da nossa página será adicionado dentro do ***ScrollView*** a partir desse ponto.

Página com Scroll

Note que a restrição Top do ScrollView começa no fim da ToolBar. Isso evita que parte do conteúdo seja encoberto pela barra.

```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:fillViewport="true"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/mainToolBar"
    app:layout_constraintBottom_toBottomOf="parent">

</ScrollView>
```

Página com Scroll

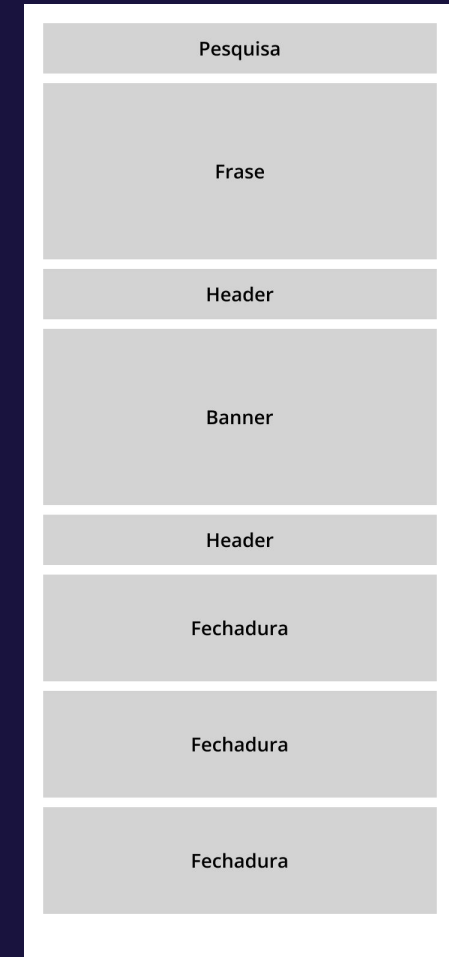
Com o ScrollView adicionado, quando o conteúdo da nossa página não couber na tela, será possível rolar o conteúdo para baixo.

Contudo, o ScrollView não é um elemento de layout, então devemos adicionar um ViewGroup dentro do ScrollView.

Para o layout de nossa página vamos analisar a estrutura do conteúdo da tela.

Página com Scroll

Analizando o conteúdo, vemos que o layout básico da página é o ***LinearLayout*** configurado para vertical.



Página com Scroll

Após adicionar o `VerticalLayout`, podemos começar a adição dos outros elementos.

```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/mainToolBar">

    <LinearLayout

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:orientation="vertical">

        </LinearLayout>
    </ScrollView>
```

Página com Scroll



Para melhorar a visualização do app, adicionamos um padding de 16dp em todo o LinearLayout.

```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:fillViewport="true"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/mainToolBar">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="16dp">

    </LinearLayout>
</ScrollView>
```

Barra de pesquisa

Após adicionar o `VerticalLayout`, podemos começar a adição dos outros elementos.

O primeiro elemento no nosso design é a barra de pesquisa.

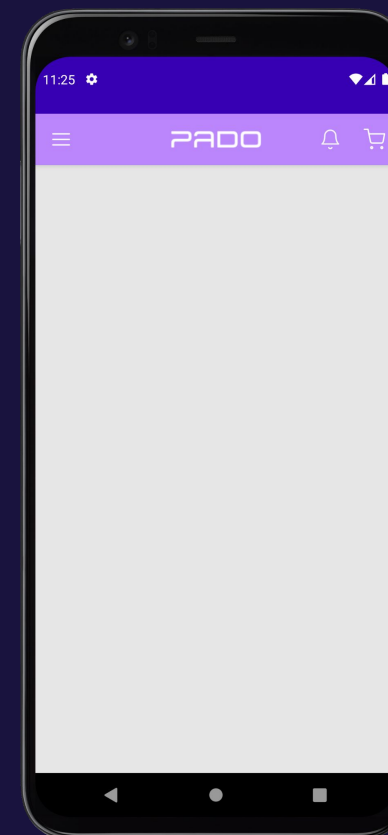


Barra de pesquisa



Para visualizarmos melhor os elementos, mude o background da aplicação para a cor #E6E6E6.

```
2 <androidx.constraintlayout.widget.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:background="#e6e6e6"
9     tools:context=".MainActivity">
```

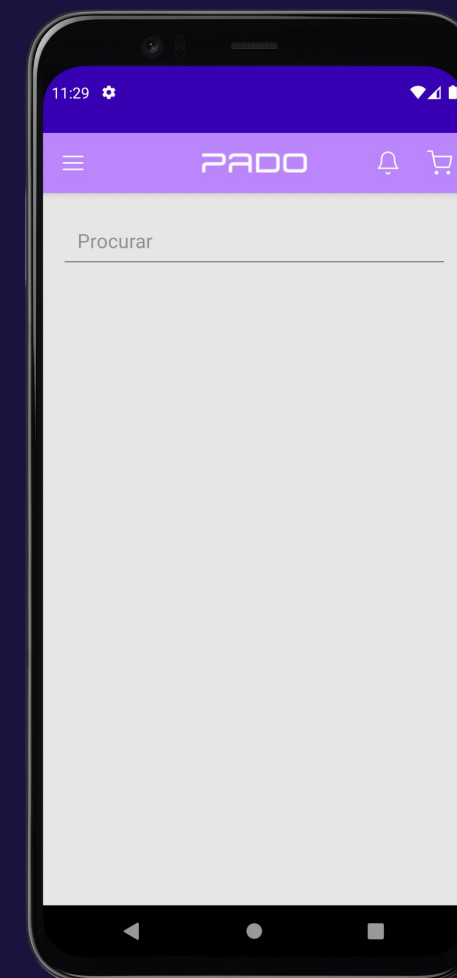


Barra de pesquisa



Adicione um EditText. Note que o background é transparente e não há ícone.

```
<EditText  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Procurar"  
    android:padding="16dp" />
```

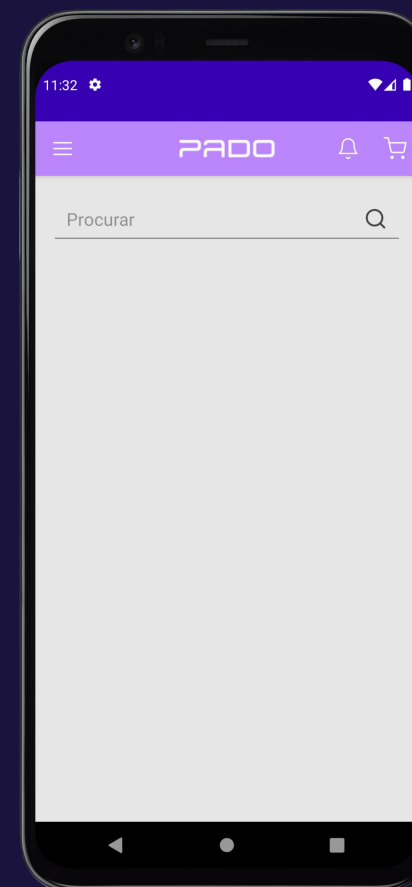


Barra de pesquisa



Para adicionar o ícone de lupa, podemos utilizar a propriedade `android:drawableEnd`

```
<EditText  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:drawableEnd="@drawable/ic_search"  
    android:hint="Procurar"  
    android:padding="16dp" />
```



Barra de pesquisa

Para adicionar um background com cantos arredondados, primeiro precisamos criar um arquivo de desenho XML.

Clique com o botão direito na pasta *drawable* e selecione a opção *New > Drawable Resource File*.

Barra de pesquisa



Preencha o formulário conforme a imagem abaixo:

A screenshot of the "New Resource File" dialog box in an IDE. The dialog has a dark theme. At the top, the title bar says "New Resource File" with a close button. Below the title bar, there are four input fields: "File name:" with the text "bg_input", "Root element:" with the text "shape", "Source set:" with a dropdown menu showing "main src/main/res", and "Directory name:" with the text "drawable". Below these fields, there are two sections: "Available qualifiers:" on the left and "Chosen qualifiers:" on the right. The "Available qualifiers:" section contains a list of qualifiers with icons: Country Code, Network Code, Locale, Layout Direction, Smallest Screen Width, Screen Width, Screen Height, Size, Ratio, Orientation, UI Mode, Night Mode, and Density. The "Country Code" item is currently selected. Between the two sections are two buttons: ">>" and "<<". The "Chosen qualifiers:" section is currently empty and contains the text "Nothing to show". At the bottom of the dialog, there is a question mark icon on the left and "OK" and "Cancel" buttons on the right.

Barra de pesquisa

Abra o arquivo bg_input.xml e adicione:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <shape xmlns:android="http://schemas.android.com/apk/res/android"
3      android:shape="rectangle">
4      <solid android:color="#ffffff" />
5      <corners android:radius="10dp" />
6  </shape>
7
```

O EditText terá um
background
retangular

Preenchido com cor
branca

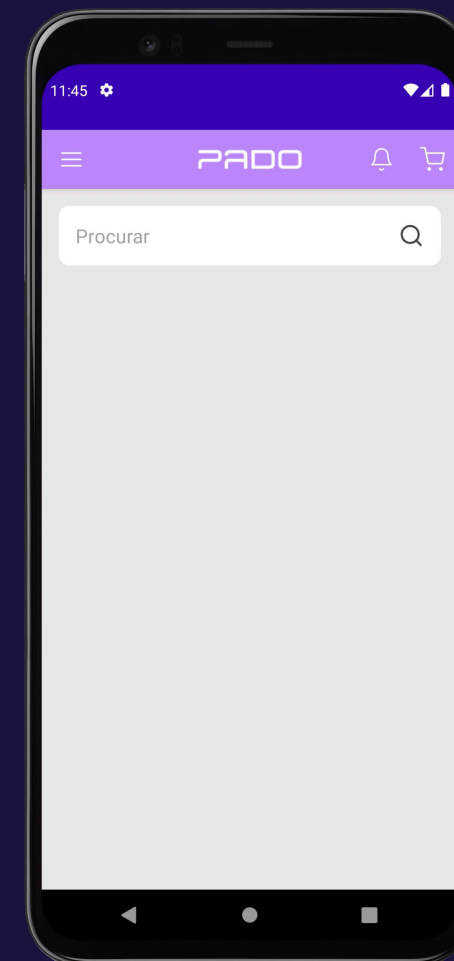
E uma borda com raio de 10dp

Barra de pesquisa



Adicione o background que criamos ao EditText:

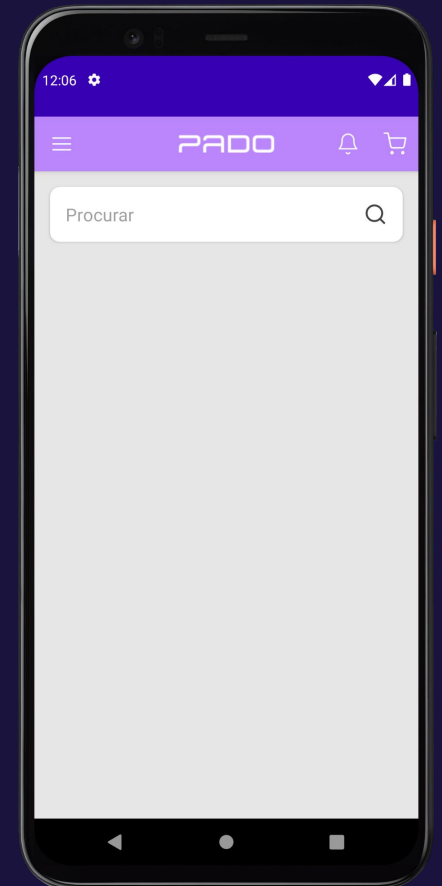
```
<EditText  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:background="@drawable/bg_input"  
    android:drawableEnd="@drawable/ic_search"  
    android:hint="Procurar"  
    android:padding="16dp" />
```



Barra de pesquisa

Para adicionar sombra, podemos utilizar a propriedade **android:elevation**:

```
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/bg_input"
    android:drawableEnd="@drawable/ic_search"
    android:elevation="2dp"
    android:hint="@string/search"
    android:padding="16dp" />
```



Barra de pesquisa

A propriedade **android:elevation** pode ser utilizada em qualquer View. Porém é necessário que o elemento pai tenha padding e a opção **android:clipToPadding="false"**

Barra de pesquisa

Adicione
`android:clipToPadding="false"`
ao `LinearLayout` para o correto
funcionamento da elevação.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:clipToPadding="false"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@drawable/bg_input"
        android:drawableEnd="@drawable/ic_search"
        android:elevation="2dp"
        android:hint="@string/search"
        android:padding="16dp" />

</LinearLayout>
```

Headline

O próximo passo é a adição do texto “O MUNDO CADA VEZ MAIS DIGITAL”.

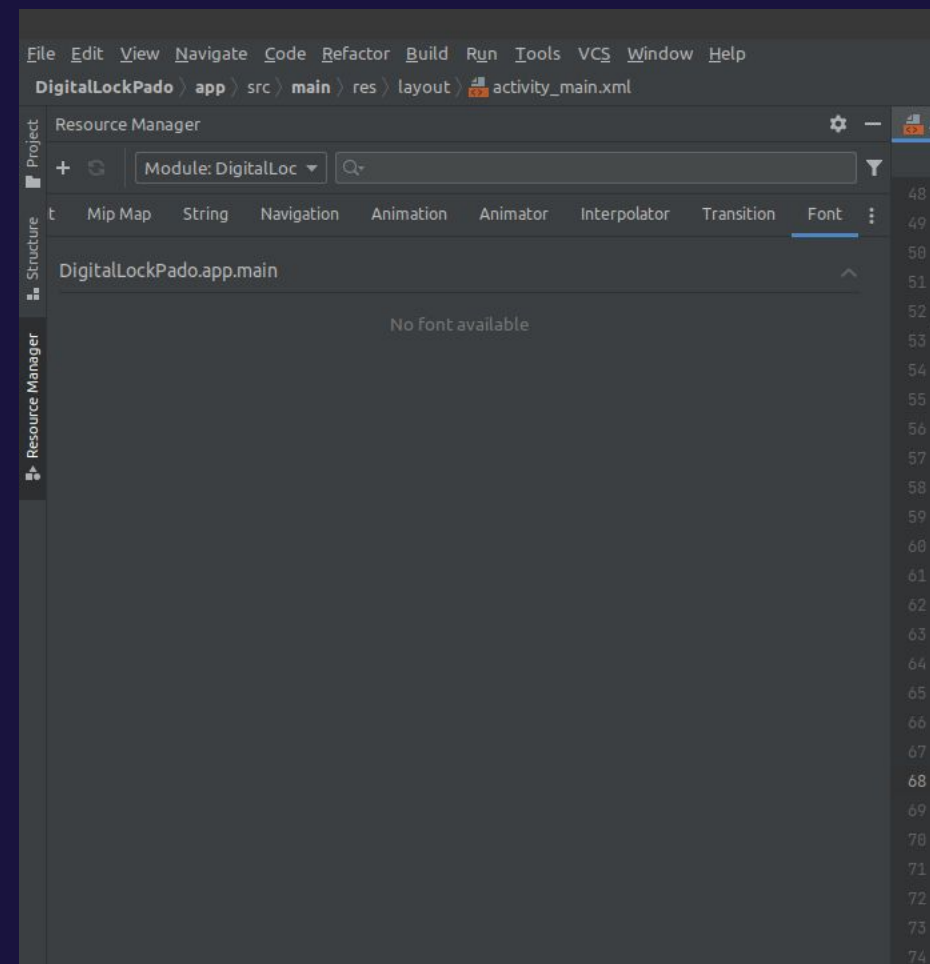
Note que nesse texto temos quebra de linha e fonte bold.



Headline

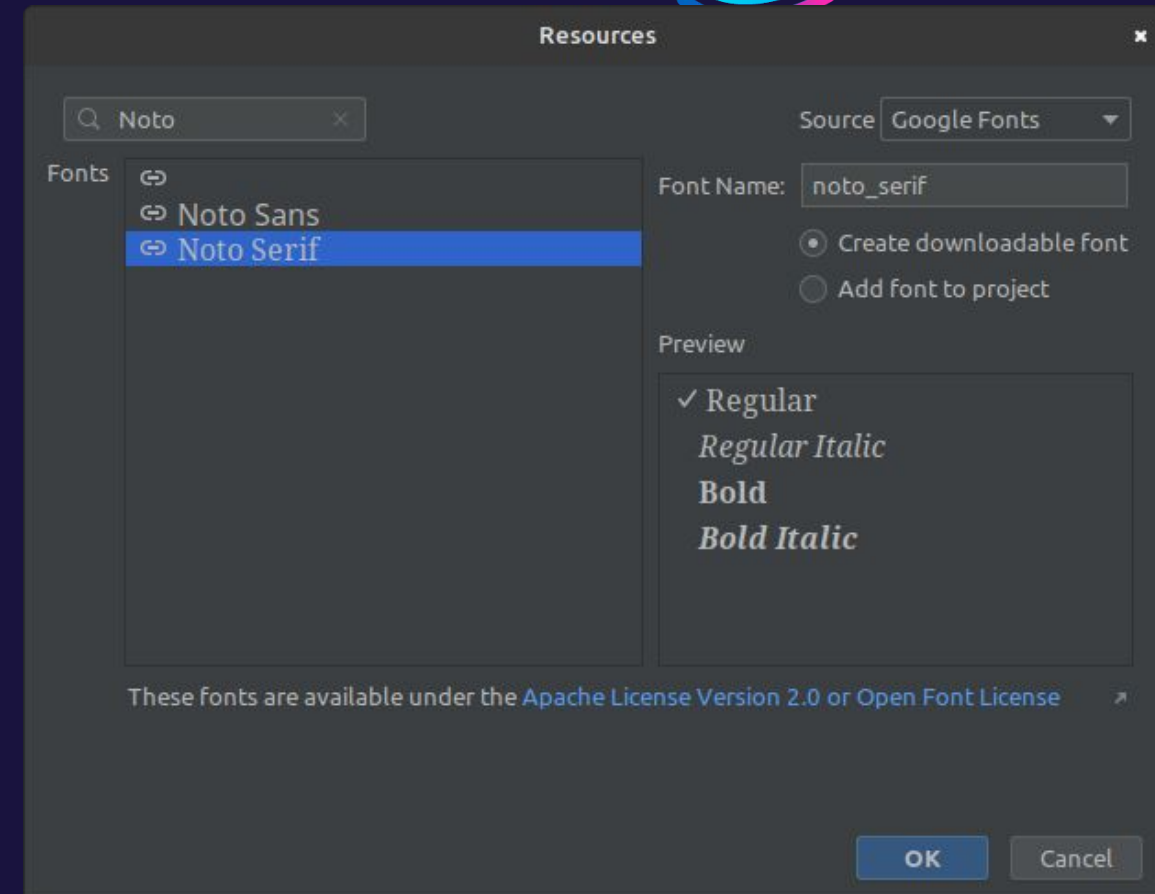


Para adicionar a fonte utilizada no text, abra a aba *Resource Manager* e selecione a tab *Font*.



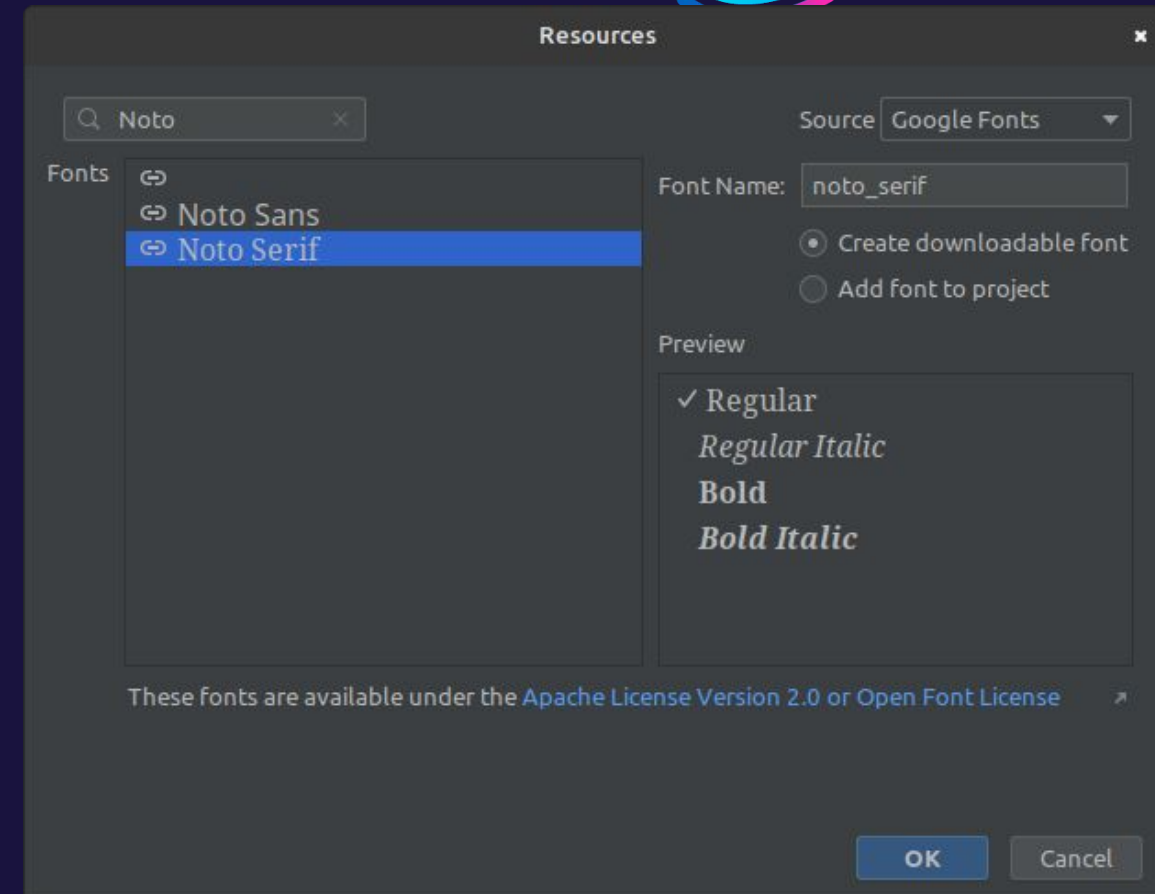
Headline

Clique no símbolo + > *more fonts*
e procure pela fonte Noto Serif.
Adicione a versão Regular.



Headline

Clique no símbolo + > *more fonts*
e procure pela fonte Noto Serif.
Adicione a versão Regular.



Headline



Abra o arquivo *res > values > strings.xml* e adicione o texto:

```
<string name="o_mundo_cada_voz_mais_digital">O MUNDO\nCADA VEZ MAIS\n<b>DIGITAL</b></string>
```

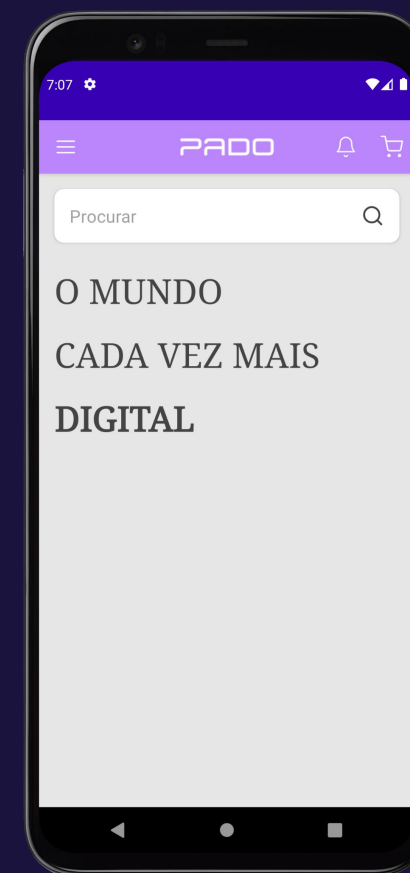
Note o uso do `\n` para quebra de linha e `` para adicionar o **bold** no texto

Headline



Voltando ao arquivo de Layout, adicione o **TextView**

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="26dp"
    android:layout_marginBottom="26dp"
    android:fontFamily="@font/notes_serif"
    android:lineHeight="60sp"
    android:text="@string/o_mundo_cada_vez_mais_digital"
    android:textColor="#444444"
    android:textSize="36sp"
    tools:targetApi="p" />
```

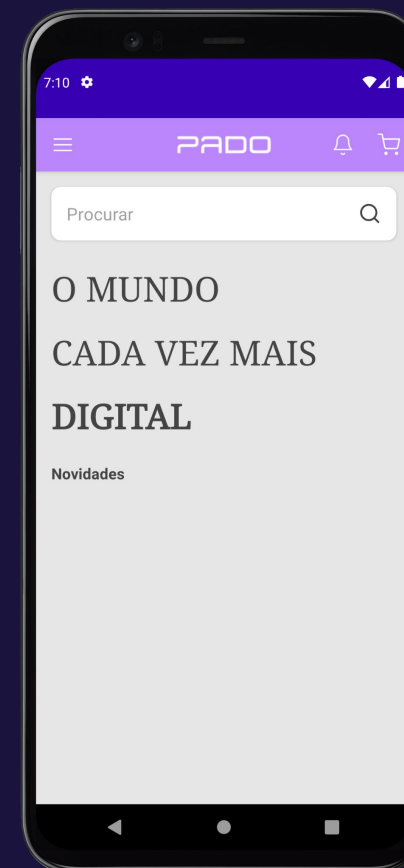


Headline



Adicione o TextView de novidades

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="10dp"
    android:text="@string/novidades"
    android:textColor="#444444"
    android:textSize="16sp"
    android:textStyle="bold" />
```



Banner



Adicione o elemento **CardView** e dentro dele um **ImageView** com a imagem do banner

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:cardCornerRadius="24dp">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:adjustViewBounds="true"
        android:scaleType="fitCenter"
        android:src="@drawable/banner" />

</androidx.cardview.widget.CardView>
```

Cantos arredondados do
CardView

Ajusta o tamanho
da imagem

Escala a imagem mantendo a
proporção

Banner

Repare que o ***CardView*** possui atributo para editar os cantos arredondados e por padrão já possui elevação



Banner



Adicione a imagem do banner aos drawables copiando o arquivo e colando na pasta drawables.

Aproveite e já adicione as outras imagens também.

Lista de Produtos

Comece adicionando o TextView
Produtos

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="10dp"  
    android:layout_marginBottom="10dp"  
    android:text="@string/produtos"  
    android:textColor="#444444"  
    android:textSize="16sp"  
    android:textStyle="bold" />
```





Lista de Produtos

Para criar os itens da lista de produtos, podemos utilizar o ***CardView***, pois o mesmo já fornece o canto arredondado e elevação.

Lista de Produtos

Analizando a estrutura do item, podemos separar o conteúdo do **CardView** em 2 grupos, utilizando um **LinearLayout** horizontal.



FDE-600W

FECHADURA DIGITAL DE EMBUTIR A
PROVA D'ÁGUA E COM CONECTIVIDADE

R\$1.500,00

IMG
(25%)

CONTEÚDO
(75%)

Lista de Produtos



Adicione o CardView

```
<androidx.cardview.widget.CardView
    android:id="@+id/fde600wCard"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="5dp"
    app:cardCornerRadius="8dp">

</androidx.cardview.widget.CardView>
```

Lista de Produtos

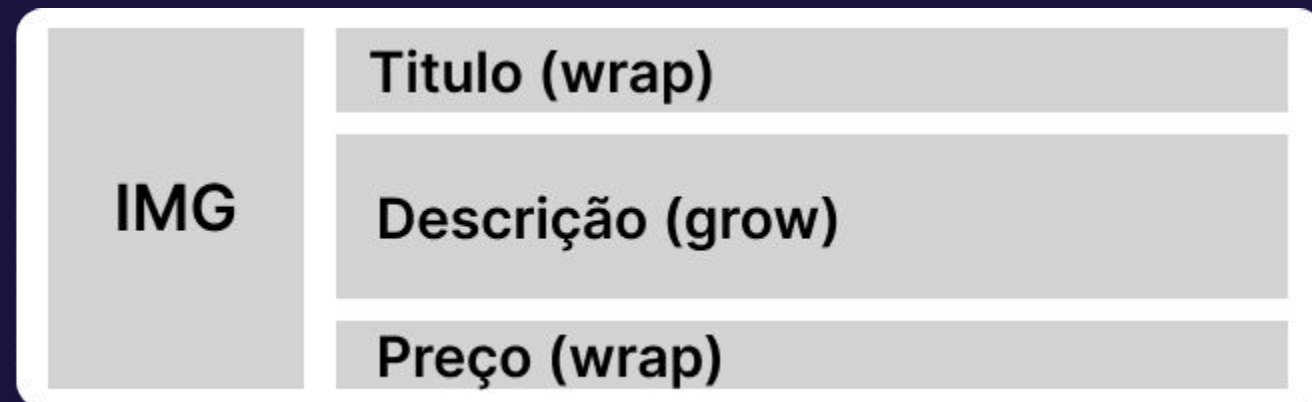
Dentro do **CardView**, adicione um **ImageView** e um **LinearLayout**. Utilize o **weight** para ajustar os tamanhos em 25% e 75%, respectivamente

```
<ImageView
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:layout_weight="0.25"
    android:scaleType="fitCenter"
    android:src="@drawable/fde600w" />

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="0.75"
    android:orientation="vertical">
```

Lista de Produtos

A segunda parte do item, podemos utilizar 3 **TextViews**, sendo o primeiro com altura `wrap_content`, o segundo para crescer até ocupar todo o espaço e o terceiro com `wrap_content`



Lista de Produtos



Adicione o primeiro **TextView**

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="5dp"
    android:text="@string/fde_600w_nome"
    android:textColor="#444444"
    android:textSize="16sp"
    android:textStyle="bold"
    tools:targetApi="p" />
```

Lista de Produtos



Adicione o segundo **TextView**

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:text="@string/fde_600w_descricao"
    android:textColor="#444444"
    android:textSize="12sp"
    tools:targetApi="p" />
```


Lista de Produtos



Adicione o terceiro **TextView**

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/fde_600w_preco"
    android:textColor="#444444"
    android:textSize="12sp"
    android:textStyle="bold"
    tools:targetApi="p" />
```

Lista de Produtos



Por último, adicione a navegação para a página de detalhes da fechadura

```
val fde600wCard = findViewById<CardView>(R.id.fde600wCard);  
fde600wCard.setOnClickListener { it: View!  
    startActivity(Intent( packageContext: this, Fde600W::class.java))  
}
```

Lista de Produtos



Confira o resultado final no emulador



Atividade II



Adicione a fechadura FDE-300W a lista de produtos e crie a página de detalhe da mesma.

Subir no GitHub até a próxima Aula



PADO
Labs