



# Desenvolvimento de Software para Dispositivos Móveis

## Aula 8 - Customizar Widgets - RecyclerView - Material Design



Professor: Anderson Almada

# Checkbox

---

- Na pasta res, entre na pasta drawable
- Crie um layout (botão direito do mouse)
  - New -> Drawable resource file
- Em file name, especifique o nome que você quiser
  - customcheckbox

# Checkbox

---

- Como é um checkbox, é preciso customizar as opções de checked ou unchecked dele.
- Especifique como conteúdo da tag selector, os seguintes itens:
  - `<item android:state_checked="true" android:drawable="" />`
  - `<item android:state_checked="false" android:drawable="" />`

# Checkbox

---

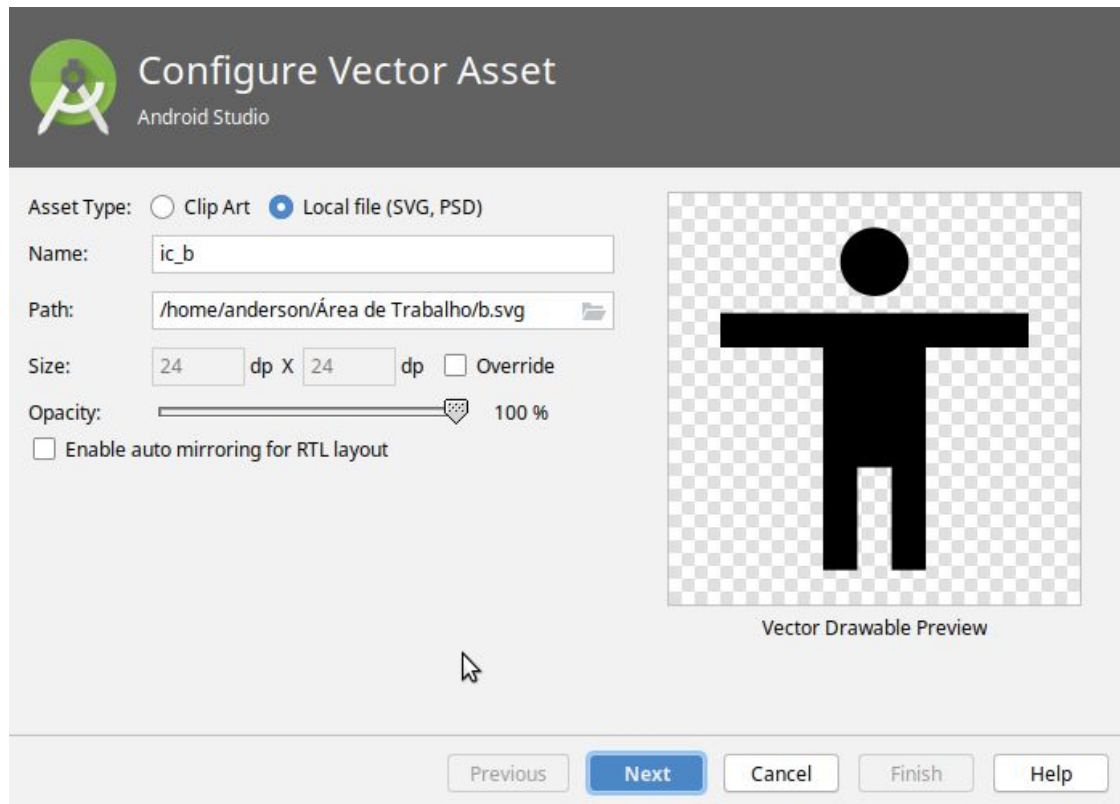
- Agora é necessário colocar os ícones na pasta drawable. Para isso, busque pelos itens que desejar. Sugestão:
  - <https://material.io/resources/icons/?style=baseline>

# Checkbox

---

- Agora use a própria ferramenta para fazer a integração do svg (ou png) com o projeto.
- Para isso, clique com o botão direito em cima de app e escolha:
  - New -> Vector Asset

# Checkbox



# Checkbox

---

- Depois de escolher a opção e clicar em **next**, um arquivo será criado com o nome que você escolheu (na imagem anterior ic\_b) com extensão xml.
- Faça isso para os dois ícones (checked e unchecked)
- Agora, basta importar os dois xml no arquivo customcheckboxbox.xml

# Checkbox

- O conteúdo do arquivo customcheckbox.xml deve ficar parecido com esse:

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_checked="true" android:drawable="@drawable/ic_a" />
    <item android:state_checked="false" android:drawable="@drawable/ic_b"/>
</selector>
```



# Checkbox

---

- Agora você pode criar normalmente um checkbox no layout da activity e fazer a importação do customcheckbox.xml
- Para importar, use o atributo:
  - **android:button**
- O resultado é o seguinte:

# Checkbox

---

<CheckBox

```
    android:id="@+id/checkbox"
    android:layout_width="match_parent"
    android:layout_height="130px"
    android:button="@drawable/customcheckbox"
    android:text="CheckBox" />
```

## RecyclerView < Sdk 29

---

- Importe no build.gradle (Module: app)

```
dependencies {  
    ...  
    implementation 'com.android.support:recyclerview-v7:28.0.0'  
}
```

- Clique em **Sync now** para sincronização do projeto com as dependências

# RecyclerView >= Sdk 29

---

- Importe no build.gradle (Module: app)

```
dependencies {  
    ...  
    implementation 'androidx.recyclerview:recyclerview:1.1.0-beta04'  
}
```

- Clique em **Sync now** para sincronização do projeto com as dependências

# RecyclerView

---

- Download ícones
  - more e delete
  - [https://github.com/andersonalmada/ds\\_mobile/tree/master/icons](https://github.com/andersonalmada/ds_mobile/tree/master/icons)
  - Extraia e copie e cole o resultado (res) na pasta do projeto
- Crie um layout na pasta res/layout
  - customrecycler
    - <https://pastebin.com/2K5UAAMa>

# RecyclerView

---

- Configure o build.gradle (Module: app)

- android {

- defaultConfig {

- ...

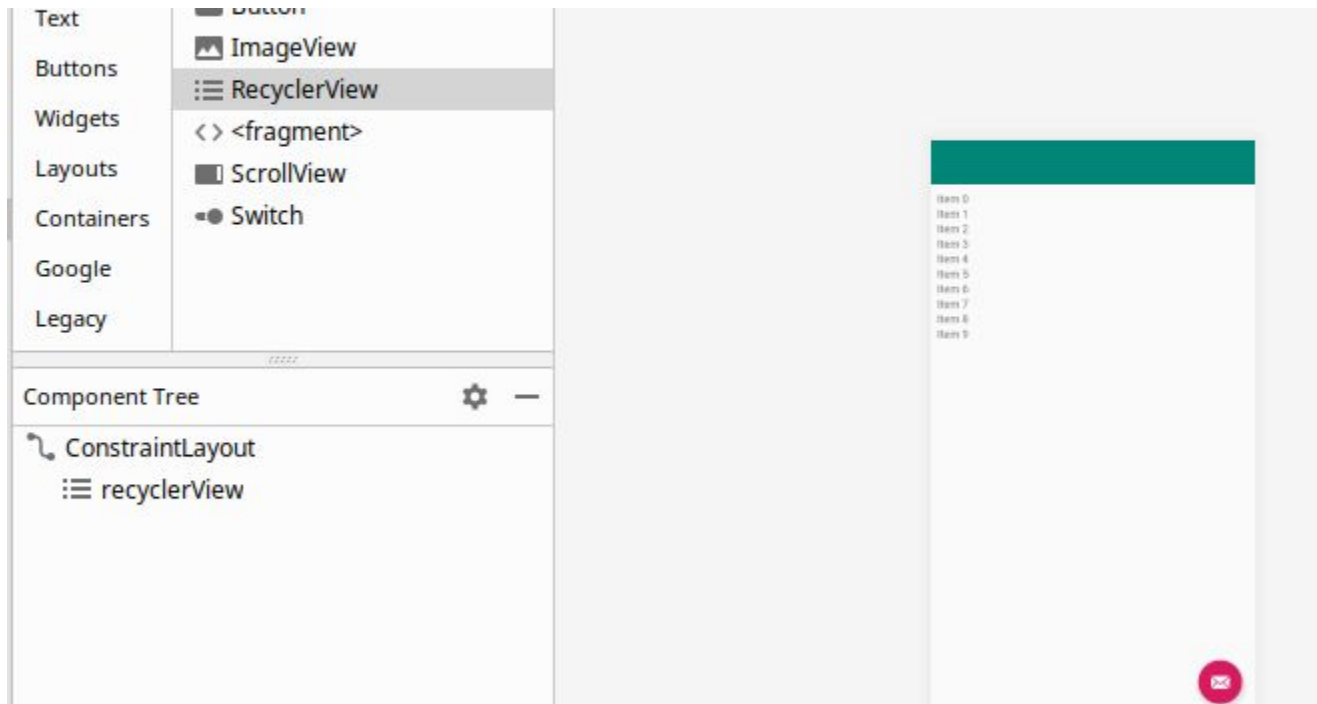
- vectorDrawables.useSupportLibrary = true**

- }

- }

# RecyclerView

- No layout da activity, coloque o RecyclerView e um id para ele



# RecyclerView

---

- Crie um modelo
  - UserModel
    - Name, Age, City
    - <https://pastebin.com/xpWBfKSP>

**Model** - modelo de dados utilizado



# RecyclerView

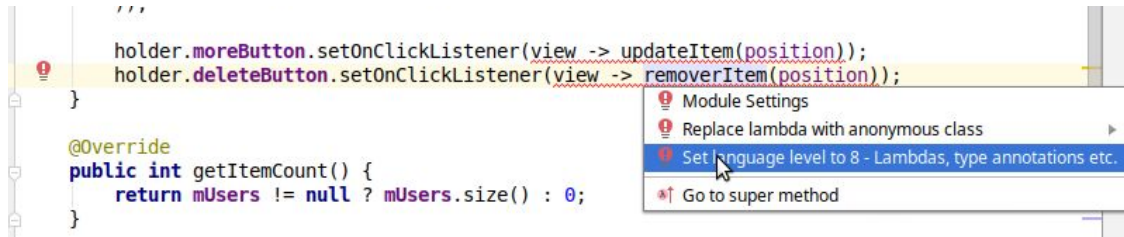
---

- Crie um holder para cada linha
  - LineHolder
    - <https://pastebin.com/11xYNKM5>

**Holder** - mantém as referências da *view* ajudando a reciclagem

# RecyclerView

- Crie um adapter para cada linha
  - LineAdapter
    - <https://pastebin.com/5jfEYNuD>
  - Observação: configure explicitamente para **java 8** (por conta do lambda)



**Adapter** - adaptador para o mapeamento de dados em view

# RecyclerView

---

- Configure o layout do RecyclerView na Activity

```
RecyclerView recyclerView;  
private LinearLayoutManager mAdapter;
```

- No método onCreate

```
recyclerView = (RecyclerView) findViewById(R.id. recyclerView);  
LinearLayoutManager layoutManager = new LinearLayoutManager(this);  
recyclerView.setLayoutManager(layoutManager);  
mAdapter = new LinearLayoutManager(new ArrayList<>(0));  
recyclerView.setAdapter(mAdapter);
```

# RecyclerView

- Configure o FloatingActionButton

```
FloatingActionButton fab = findViewById(R.id. fab);  
fab.setOnClickListener(view -> insertItem());
```

- Crie o método de insertItem

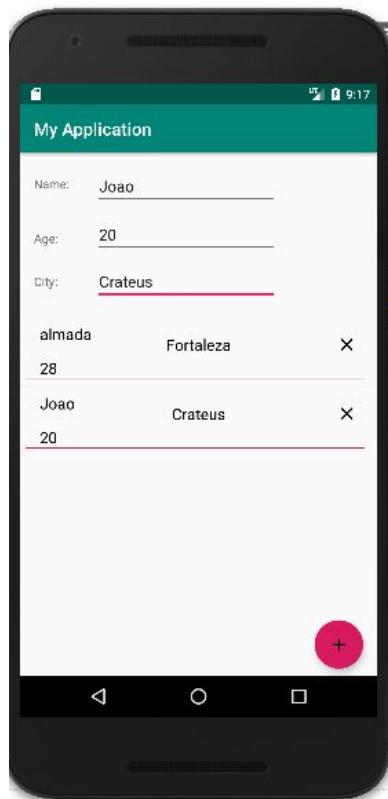
```
public void insertItem() {  
    UserModel model = new UserModel();  
    model.setName( "Anderson" );  
    model.setAge( 28 );  
    model.setCity( "Fortaleza" );  
    mAdapter.insertItem(model);  
}
```

# RecyclerView

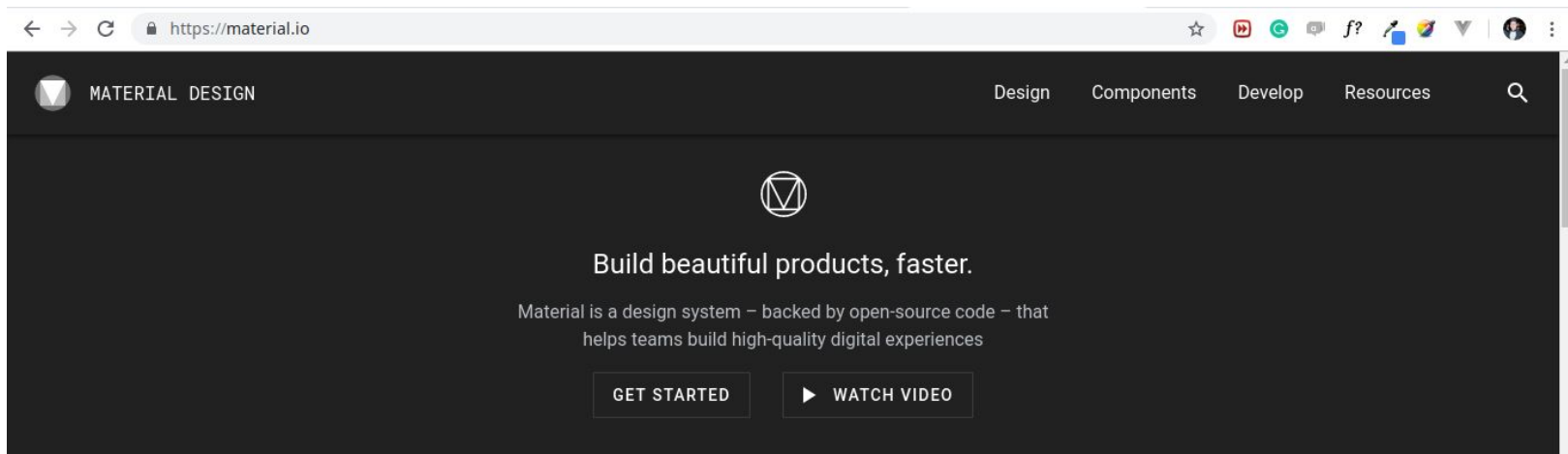
---

- Código completo da MainActivity
  - <https://pastebin.com/u6jrf4aF>

# Exercício

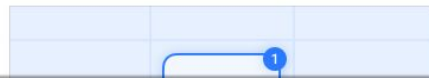


# Material Design



## Design guidance and code

Use our most popular design and development resources to jumpstart your latest project



# Link importante

---

- <https://material.io/resources/icons/?style=baseline>
- [https://material.io/resources/color/#!/](https://material.io/resources/color/#!)
- <https://medium.com/android-dev-br/listas-com-recyclerview-d3f41e0d653c>





# Dúvidas??

E-mail: [almada@crateus.ufc.br](mailto:almada@crateus.ufc.br)