**Android Studio**

**Implementação para experiência de usuário e Interface**

1. **Adicionando em build.grade(app)** 👇

// Adicionando viewBinding na linha 34 e 35 como verdadeiro

**Ficando assim**

* **LINHA 34:buildFeatures {**

**35: viewBinding true**

1. **Adicionando em activity\_MAIN.XML** 👇

!--Na linha 10 foi colocado um android:id="@+id/tvHello" para recuperar -->  
<!-- na MainActivity incluindo um identificador-->

**Ficando assim**

13: android:id="@+id/tvHello"

**3.0Adicionando em MAINActivity.kt** 👇

// Criando activity usando o biding nas linhas 11, 18, 19 e 21 e comenta a linha 16

Linha 10// binding Private para ActivityMainBinding  
11 private lateinit var binding: ActivityMainBinding

linha18. binding = ActivityMainBinding.inflate(*layoutInflater*) //Criando o elemento binding

19. setContentView(binding.*root*) // chamando set com elemento root  
  
21. binding.tvHello.*text* = "Teste"

1. **Adicionando em me.dio.simulator a classe Java ficando assim** 👇

* **MainActivityJava e ai adiciona o conteúdo** 👇
* package me.dio.simulator;  
    
  import android.os.Bundle;  
  import android.widget.TextView;  
    
  import androidx.annotation.Nullable;  
  import androidx.appcompat.app.AppCompatActivity;  
    
  import me.dio.simulator.databinding.ActivityMainBinding;  
    
  public class MainActivity extends AppCompatActivity {  
    
   // binding Private para ActivityMainBinding  
   private ActivityMainBinding binding;  
    
   @Override  
   protected void onCreate(@Nullable Bundle savedInstanceState) {  
   super.onCreate(savedInstanceState);  
    
   binding = ActivityMainBinding.*inflate*(getLayoutInflater()); //Criando o elemnto binding  
   setContentView(binding.getRoot());  
    
   }  
  }

1. Usando o **SwipeRefreshLayout no Android EX**

**Adicionando em build.grade(app)** **E alterando da linha a 45 para essa** 👇

Linha42. implementation '**androidx**.core:core-ktx:1.7.0'

Linha43. implementation '**androidx**.appcompat:appcompat:1.4.0'

Linha44. implementation '**androidx**.constraintlayout:constraintlayout:2.1.2'

**Linha.45 implementation 'androidx.swiperefreshlayout:swiperefreshlayout**:1.1.0**'**

Linha46. implementation 'com.google.android.material:material:1.6.1'

1. **Alterações de Cores primárias e secundárias das paletas**

* **Cfme desejar seguindo regras das cores**

1. **Alterações de ícones e Temas**

**Botão direito e vai criar um**

**🡪image ou vector asset**

**O Image asset para png vai em clip arte clica no bonequinho e pesquisa pelo que precisa**

**O vector asset serve para svg, pois consegue se adequar ao tamanho.**

1. **Protótipos**

**Depois vai Em constraintLayout(que são regras de posicionamento) vai incluir:**

* **SwipeRefreshLayout**
* **Recicler Views**
* **Adicionando em activity\_MAIN.XML** 👇
* <!-- adicionando layout de atualização de deslizamento linha 10 a 17 -->  
  <androidx.swiperefreshlayout.widget.SwipeRefreshLayout  
   android:id="@+id/SRMatches"  
   android:layout\_width="0dp"  
   android:layout\_height="0dp"  
   app:layout\_constraintBottom\_toBottomOf="parent"  
   app:layout\_constraintEnd\_toEndOf="parent"  
   app:layout\_constraintStart\_toStartOf="parent"  
   app:layout\_constraintTop\_toTopOf="parent" />
* <!-- adicionando layout de recycler views linha 19 a 23 -->  
   <androidx.recyclerview.widget.RecyclerView  
   android:id="@+id/RVMatches"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent" />
* <!-- adicionando botão e acrescentando linha 30 e 31 -->
* Linha30. android:contentDescription="Botão de Simulação de Partidas. "
* linha31. android:clickable="true"
* **🡪Depois vc clica na lampada pra fazer o extratc e altera o nome e em values coloca o x no português e values)**
* **🡪E logo após em cima do texto criado da ctrl clique e já vai adionar em values (português e inglês)**
* **Continuando la em cima no app clica com botão direito em new / vector asset e muda a configurqação do botão conforme teu projeto**

1. **Criação de Layouts**

**Vai com botão direito na pasta layout**

* **E clica layout resources file**
* **Ao abrir o arquivo vai alterar para gerar um cardview**
* <?xml version="1.0" encoding="utf-8"?>  
  <androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  
   android:layout\_width="match\_parent"  
   android:layout\_height="match\_parent">  
    
  </androidx.cardview.widget.CardView>
* **Em design vai fazer o seguinte:**
* **Em layout Vai puxar ConstraintLayout**
* **Em widgets vai puxar imageviews e alterar para 120dp altura e largura**
* **Em Common eu puxo textView.**

1. **Conforme projeto**

**Em imageView vai acrescentar ao final da linha**

<string name="imagem\_do\_time\_mandante">Imagem do time Mandante</string>

* **🡪E logo após em cima do texto criado da ctrl clique e já vai adionar em values (português e inglês)**

**Em todos textView**

<!-- Alteração na linha 43,53,69,90 de android:text para tools-->  
<TextView  
  
 tools:text="TextView" />

**Depois vai em Code 🡪 Reformat Code eos tools do textViews mudados vão pro fim da linha**

1. **Alteração em themess.xml**

|  |
| --- |
| <resources xmlns:tools="http://schemas.android.com/tools"> |
|  | <!-- Base application theme. --> |
|  | <!-- https://m3.material.io/libraries/mdc-android/getting-started --> |
|  | <style name="Theme.Banklineandroid" parent="Theme.Material3.Light"> |
|  | <!-- https://m3.material.io/libraries/mdc-android/color-theming --> |
|  | <item name="colorPrimary">@color/primary\_600</item> |
|  | <item name="colorPrimaryContainer">@color/primary\_100</item> |
|  | <item name="colorOnPrimaryContainer">@color/primary\_900</item> |
|  | <item name="colorPrimaryInverse">@color/primary\_200</item> |
|  | </style> |
|  | <!-- Interesting alternative to Material 3: https://material-foundation.github.io/material-theme-builder --> |
|  | </resources> |

1. **Alteração em Dimens.xml**

**App\_bar\_height para 180dp**

**Ao término vai no app clica com botão direito em git**

**E faz o commit directory:**

**🡪 Abaixo digita: “Desenvolvimento dos layouts de listagem e detalhe de partidas.**

1. **Realizaçoes Partidas em json no github**

* **Cria um novo repositório**
* **Código da partida com:**
* **descrição,**
* **local da partida,**
* **nome dos times,**
* **quem é mandante e visitante**
* **força do time**

|  |
| --- |
| [ |
|  | { |
|  | "descricao": "Eliminatórias Copa 2022", |
|  | "local": { |
|  | "nome": "Maracanã", |
|  | "imagem": "https://cdn.pixabay.com/photo/2016/03/29/17/07/maracana-1288818\_1280.jpg" |
|  | }, |
|  | "mandante": { |
|  | "nome": "Brasil", |
|  | "forca": 5, |
|  | "imagem": "https://www.bandeirasnacionais.com/data/flags/normal/br.png" |
|  | }, |
|  | "visitante": { |
|  | "nome": "Argentina", |
|  | "forca": 5, |
|  | "imagem": "https://www.bandeirasnacionais.com/data/flags/normal/ar.png" |
|  | } |
|  | }, |
|  | { |
|  | "descricao": "Eliminatórias Copa 2022", |
|  | "local": { |
|  | "nome": "Olympic Stadium Hernando Siles", |
|  | "imagem": "https://upload.wikimedia.org/wikipedia/commons/9/9d/Hernando\_Siles\_Stadium\_-\_La\_Paz.jpg" |
|  | }, |
|  | "mandante": { |
|  | "nome": "Bolívia", |
|  | "forca": 2, |
|  | "imagem": "https://www.bandeirasnacionais.com/data/flags/normal/bo.png" |
|  | }, |
|  | "visitante": { |
|  | "nome": "Brasil", |
|  | "forca": 5, |
|  | "imagem": "https://www.bandeirasnacionais.com/data/flags/normal/br.png" |
|  | } |
|  | }, |
|  | { |
|  | "descricao": "Eliminatórias Copa 2022", |
|  | "local": { |
|  | "nome": "Maracanã", |
|  | "imagem": "https://cdn.pixabay.com/photo/2016/03/29/17/07/maracana-1288818\_1280.jpg" |
|  | }, |
|  | "mandante": { |
|  | "nome": "Brasil", |
|  | "forca": 5, |
|  | "imagem": "https://www.bandeirasnacionais.com/data/flags/normal/br.png" |
|  | }, |
|  | "visitante": { |
|  | "nome": "Colombia", |
|  | "forca": 3, |
|  | "imagem": "https://www.bandeirasnacionais.com/data/flags/normal/co.png" |
|  | } |
|  | } |
|  | ] |

1. **Em pages vc vai instalar um domínio e depois de realizar o deply em actions**

**Vai ficar assim colocando no final**

**🡪 /matches.json:**

[**https://rafarz76.github.io/matches-simulator-api/matches.json**](https://rafarz76.github.io/matches-simulator-api/matches.json)

* **No android vai criar um domain em java**
* **No site vai digitar para criar a classe e os códigos do json do github**

[**https://www.jsonschema2pojo.org/**](https://www.jsonschema2pojo.org/)

**Depois tira getters e setters cria o nome da classe e clica em previows**

**Incluir retrofit no build.gradle(module: Simulator.app)**

// Retrofit (HTTP Client): https://square.github.io/retrofit  
implementation 'com.squareup.retrofit2:retrofit:2.9.0'  
implementation 'com.squareup.retrofit2:converter-gson:2.9.0'