# **Adicionando uma imagem para identificar o usuário**

*Começando deste ponto? Você pode fazer o* [*DOWNLOAD*](https://s3.amazonaws.com/caelum-online-public/android-chat/stages/capitulo-6.zip) *do projeto completo do capítulo anterior e continuar seus estudos a partir deste capítulo.*

As mensagens do nosso chat são apresentadas ao usuário com o componente ListView, mas ainda não conseguimos saber direito a quem pertence as mensagens que aparecem na lista. As cores ajudam a diferenciar, mas podemos deixar ainda mais claro usando imagens pra identificar os usuários. Poderíamos facilmente usar a câmera para tirar uma foto do usuário ou importar da galeria, mas isso foge um pouco do nosso foco. O que faremos então, é usar um serviço da Web que fornece imagens aleatórias baseadas em uma identificação. Vamos começar?

Para mudar o layout da tela, precisamos modificar o arquivo **res/layout/activity\_main.xml**, adicionando um **<ImageView />** no layout:

<ImageView  
 android:id="@+id/iv\_avatar\_usuario"  
 android:layout\_width="50dp"  
 android:layout\_height="50dp" />

E na MainActivity, podemos pegar a sua referência:

public class MainActivity extends AppCompatActivity {  
  
 // outros atributos  
  
 @BindView(R.id.iv\_avatar\_usuario)  
 ImageView avatar;  
  
 // restante do código comentado  
  
}

## **Mudando o layout do item da lista**

Como já vimos anteriormente, para personalizar o layout do item da lista, basta criar um *Adapter* customizado que apresente para o Android o layout que queremos. No nosso projeto já possuímos nosso próprio *adapter*, o MensagemAdapter, então iremos apenas modificar o layout da linha.

Atualmente só mostramos o texto da mensagem, para colocar a imagem, precisamos colocar o **<ImageView />** no layout. Para isso, vamos deixar o arquivo **res/layout/mensagem.xml** da seguinte forma:

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="horizontal"  
 android:layout\_width="match\_parent"   
 android:layout\_height="match\_parent">  
  
 <ImageView  
 android:id="@+id/iv\_avatar\_mensagem"  
 android:layout\_margin="10dp"  
 android:layout\_width="40dp"  
 android:layout\_height="40dp" />  
  
  
 <TextView  
 android:id="@+id/tv\_texto"  
 android:textSize="20sp"  
 android:paddingLeft="6dp"  
 android:paddingTop="6dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
  
</RelativeLayout>

## **Injetando componentes visuais no Adapter com ButterKnife**

Nesse ImageView queremos mostrar a imagem que buscaremos através da API, e para isso, precisamos manipular esse componente no MensagemAdapter. Como podemos pegar uma instância dele? Podemos usar o método findViewById de View que já conhecemos, mas já vimos que o findViewById aumenta a complexidade do código e com isso diminuindo a visibilidade. Para nos ajudar com essas questões, usamos o **ButterKnife** para injetar os componentes visuais.

Para usá-lo no MensagemAdapter, vamos colocar as referências aos componentes visuais como atributos da classe:

public class MensagemAdapter extends BaseAdapter {  
  
 private List<Mensagem> mensagens;  
 private Activity activity;  
 private Integer idDoCliente;  
  
 @BindView(R.id.tv\_texto)  
 TextView texto;  
  
 @BindView(R.id.iv\_avatar\_mensagem)  
 ImageView avatar;  
  
 // ...  
  
}

Agora, ainda no MensagemAdapter, podemos apagar a seguinte linha"

TextView texto = (TextView) linha.findViewById(R.id.tv\_texto);

Precisamos ainda dizer ao ButterKnife em qual classe ele deve fazer as injeções. Até então usávamos a Activity, agora queremos que a injeção seja feita no MessageAdapter.

Vamos usar o método estático **bind** da classe ButterKnife:

// MensagemAdapter.java   
  
@Override  
public View getView(int i, View view, ViewGroup viewGroup) {  
  
 ButterKnife.bind(this);  
  
 View linha = activity.getLayoutInflater().inflate(R.layout.mensagem, viewGroup, false);  
  
 Mensagem mensagem = getItem(i);  
  
 if(message.getId() != id) {  
 mensagem.setBackgroundColor(Color.CYAN);  
 }  
  
 texto.setText(mensagem.getTexto());  
  
 return mensagem;  
}

Toda Activity tem internamente um View que representa os componentes da tela e por isso podemos fazer findViewById. No caso do *Adapter*, precisamos passar a instância do View separadamente. E para isso podemos usar uma sobrecarga do método **bind**, que recebe a classe onde o ButterKnife irá injetar e a instância de View:

// MensagemAdapter.java   
  
@Override  
public View getView(int i, View view, ViewGroup viewGroup) {  
  
 View linha = activity.getLayoutInflater().inflate(R.layout.mensagem, viewGroup, false);  
  
 ButterKnife.bind(this, linha);  
  
 Mensagem mensagem = getItem(i);  
  
 if(message.getId() != id) {  
 mensagem.setBackgroundColor(Color.CYAN);  
 }  
  
 texto.setText(mensagem.getTexto());  
  
 return mensagem;  
}

Agora já temos os componentes prontos para serem utilizados! A partir do próximo vídeos começaremos a trabalhar com a imagem em si.