Cor de fundo diferente para mensagens recebidas

Uma coisa interessante para fazermos é diferenciar as mensagens que estão na lista, para sabermos se estamos enviando ou recebendo a mensagem. Podemos, por exemplo, colocar uma cor diferente na mensagem que recebemos, só para deixar mais claro.

Para isso, dentro da mensagem podemos carregar também o id do cliente. E se esse id for diferente do id da mensagem, significa que estamos **recebendo** a mensagem, então modificamos a cor de fundo da linha:

@Override  
public View getView(int i, View view, ViewGroup viewGroup) {  
  
 View linha = activity.getLayoutInflater().inflate(R.layout.mensagem, viewGroup, false);  
  
 TextView texto = (TextView) linha.findViewById(R.id.tv\_texto);  
  
 Mensagem mensagem = getItem(i);  
  
 if (idDoCliente != mensagem.getId()) {  
 linha.setBackgroundColor(Color.CYAN);  
 }  
  
 texto.setText(mensagem.getText());  
}

Tanto o **idDoCliente** quanto o **id** da mensagem não existem, então vamos criá-los. Primeiro o **idDoCliente**:

public class MensagemAdapter extends BaseAdapter {  
  
 private List<Mensagem> mensagens;  
 private Activity activity;  
 private int idDoCliente;  
  
 public MensagemAdapter(int idDoCliente, List<Mensagem> mensagens, Activity activity) {  
 this.idDoCliente = idDoCliente;  
 this.mensagens = mensagens;  
 this.activity = activity;  
 }  
  
 // métodos comentados  
}

E o **id** da **Mensagem**, juntamente com o seu *getter*:

public class Mensagem {  
  
 private String texto;  
 private int id;  
  
 public String getText() {  
 return texto;  
 }  
  
 public int getId() {  
 return id;  
 }  
}

Por último, falta o retorno, vamos retornar a **linha**:

@Override  
public View getView(int i, View view, ViewGroup viewGroup) {  
  
 View linha = activity.getLayoutInflater().inflate(R.layout.mensagem, viewGroup, false);  
  
 TextView texto = (TextView) linha.findViewById(R.id.tv\_texto);  
  
 Mensagem mensagem = getItem(i);  
  
 if (idDoCliente != mensagem.getId()) {  
 linha.setBackgroundColor(Color.CYAN);  
 }  
  
 texto.setText(mensagem.getText());  
  
 return linha;  
}

Agora finalmente podemos testar!

## **Testando o MensagemAdapter**

Na classe **MainActivity**, vamos testar o *adapter* que acabamos de fazer. Como temos acesso à ListView? Novamente podemos utilizar o método **findViewById**, mas para isso precisamos adicionar o id na ListView, dentro do arquivo **activity\_main.xml**. A **ListView** ficará assim:

<ListView  
 android:id="@+id/lv\_mensagem"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />

Voltando à **MainActivity**, conseguimos acessar a *view*:

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 ListView listaDeMensagens = (ListView) findViewById(R.id.lv\_mensagem);  
}

Agora precisamos do conteúdo das mensagens que vamos enviar. Para isso vamos criar uma lista de mensagens, com o auxílio do método **Arrays.asList**

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 ListView listaDeMensagens = (ListView) findViewById(R.id.lv\_mensagem);  
  
 List<Mensagem> mensagens = Arrays.asList(new Mensagem(1, "olá alunos de android"), new Mensagem(2, "oi"));  
}

Como na classe **Mensagem** não há esse construtor, vamos criá-lo:

// classe Mensagem  
  
public Mensagem(int id, String texto) {  
 this.id = id;  
 this.texto = texto;  
}

Agora que temos o **ListView**, que é o componente visual que irá exibir as nossas mensagens, e a lista de mensagens que serão enviadas, precisamos converter essa lista de mensagens em componentes que irão aparecer na tela, e quem faz isso para nós é justamente o **MensagemAdapter**:

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 ListView listaDeMensagens = (ListView) findViewById(R.id.lv\_mensagem);  
  
 List<Mensagem> mensagens = Arrays.asList(new Mensagem(1, "olá alunos de android"), new Mensagem(2, "oi"));  
  
 MensagemAdapter adapter = new MensagemAdapter(idDoCliente, mensagens, this);  
}

Vamos criar o atributo **idDoCliente** e inicializá-lo com **1**, assim poderemos ver as cores diferentes. Agora falta atribuir o **adapter** à **ListView**, fazemos isso através do método **setAdapter** da mesma:

public class MainActivity extends AppCompatActivity {  
  
 private int idDoCliente = 1;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 ListView listaDeMensagens = (ListView) findViewById(R.id.lv\_mensagem);  
  
 List<Mensagem> mensagens = Arrays.asList(new Mensagem(1, "olá alunos de android"), new Mensagem(2, "oi"));  
  
 MensagemAdapter adapter = new MensagemAdapter(idDoCliente, mensagens, this);  
  
 listaDeMensagens.setAdapter(adapter);  
 }  
}

Agora podemos executar a aplicação, ver o aplicativo rodando e reparar nas cores diferentes, exatamente como esperávamos!

Para melhorar o layout, no arquivo **mensagem.xml**, vamos alterar o **layout\_height** do **LinearLayout** para **50dp**. E vamos colocar um texto no botão, no arquivo **activity\_main.xml**:

<Button  
 android:layout\_weight="1"  
 android:text="Enviar"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content" />

Com isso o nosso projeto base está pronto! Agora podemos dar continuidade ao curso e aprender as tecnologias que aqui focaremos no restante dos capítulos.