Escondendo o teclado

Ainda temos um pequeno problema, toda vez que mudamos a orientação da tela, temos que esconder o teclado para poder enxergar a aplicação. O ideal que a aplicação inicie com o teclado escondido, e quando enviarmos alguma mensagem, o teclado suma também.

## **Iniciando a aplicação sem o teclado**

Então vamos por partes, primeiro para que a aplicação inicie com o teclado escondido, fazemos isso editando o arquivo **manifests/AndroidManifest.xml**, na tag **<activity>**. Ela ficará assim:

<activity android:name=".activity.MainActivity"  
 android:windowSoftInputMode="stateAlwaysHidden">

## **Sumindo com o teclado quando uma mensagem é enviada**

Para sumir com o teclado quando enviamos uma mensagem, temos que implementar essa funcionalidade no método **enviarMensagem**, da classe **MainActivity**:

@OnClick(R.id.btn\_enviar)  
public void enviarMensagem() {  
 chatService.enviar(new Mensagem(idDoCliente, editText.getText().toString())).enqueue(new EnviarMensagemCallback());  
}

Nele, primeiramente limparemos o conteúdo do **editText**:

@OnClick(R.id.btn\_enviar)  
public void enviarMensagem() {  
 chatService.enviar(new Mensagem(idDoCliente, editText.getText().toString())).enqueue(new EnviarMensagemCallback());  
  
 editText.getText().clear();  
}

Agora, após enviar a mensagem, o campo de texto é zerado. Depois precisamos esconder o teclado, mas para isso devemos pegar uma referência de **InputMethodManager**, que é quem gerencia as formas de entrada do Android. Podemos instanciá-lo através do método **getSystemService**, passando a sua constante por parâmetro:

@OnClick(R.id.btn\_enviar)  
public void enviarMensagem() {  
 chatService.enviar(new Mensagem(idDoCliente, editText.getText().toString())).enqueue(new EnviarMensagemCallback());  
  
 editText.getText().clear();  
  
 InputMethodManager inputMethodManager = (InputMethodManager) getSystemService(INPUT\_METHOD\_SERVICE);  
}

Com ele em mãos, escondemos o teclado chamando o método **hideSoftInputFromWindow**, passando o *token* do campo de texto e a flag por parâmetro. Como não utilizaremos a flag, passaremos o valor **0**:

@OnClick(R.id.btn\_enviar)  
public void enviarMensagem() {  
 chatService.enviar(new Mensagem(idDoCliente, editText.getText().toString())).enqueue(new EnviarMensagemCallback());  
  
 editText.getText().clear();  
  
 InputMethodManager inputMethodManager = (InputMethodManager) getSystemService(INPUT\_METHOD\_SERVICE);  
 inputMethodManager.hideSoftInputFromWindow(editText.getWindowToken(), 0);  
}

Podemos agora testar a nossa aplicação. Reparem que ela inicia sem o teclado, e quando enviamos alguma mensagem, ele também some :)