# **Injeção problemática**

Roberto, aluno da Alura, experimentou o ButterKnife em seu projeto. Vejamos seu código:

public class MainActivity extends Activity {  
  
 @BindView(R.id.lv\_mensagens)  
 private ListView listaDeMensagens;  
  
 @BindView(R.id.et\_texto)  
 private EditText editText;  
  
 @BindView(R.id.btn\_enviar)  
 private Button button;  
  
 // restante do código comentado  
  
}

No entanto, o projeção de injeção dos componentes da view nos atributos da ManActivity não funcionou.

Consegue identificar a causa do problema? Não precisa responder aqui. Quebre a cabeça um pouco e ao clicar em avançar você verá a resposta do instrutor.

### **Opinião do Instrutor**

O problema é que o ButterKnife exige que os atributos não sejam privados. É um preço a se pagar para conseguirmos escrever menos com o ButterKnife.

============================================================

Trabalhando com listeners

## **Gepeto tinha o seguinte listener em seu código:**

## **button.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { chatService.enviar(new Mensagem(idDoCliente, editText.getText().toString())).enqueue(new EnviarMensagemCallback()); } });**

## **No entanto, ele simplificou este código através do uso do ButterKnife. Qual das opções abaixo cria corretamente um novo método em sua *Activity* que corresponde ao código anterior, mas usando o ButterKnife?**

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

O ButterKnife nos ajuda a escrever menos código, inclusive de listeners que é um código tão comum em aplicações Android.

========================================================

# 