Injetando Picasso com Dagger

Estamos usando o **Picasso** globalmente através de um *Singleton*, mas há também um *Builder* para criarmos instâncias de Picasso, e assim injetar com **Dagger**.

Picasso picasso = new Picasso.Builder(context).build();

Precisamos ensinar ao Dagger como criar instâncias do Picasso e para isso vamos adicionar um novo método anotado com @Provides no ChatModule:

// ChatModule.java  
  
@Provides  
public Picasso picasso() {  
 Picasso picasso = new Picasso.Builder(???).build();  
 return picasso;  
}

Vamos receber, no construtor, uma instância de Application que será o que usaremos como Context:

public class ChatModule {  
  
 private Application app;  
  
 public ChatModule(Application app) {  
 this.app = app;  
 }  
  
 // getChatService() comentado  
  
 @Provides  
 public Picasso picasso() {  
 Picasso picasso = new Picasso.Builder(app).build();  
 return picasso;  
 }  
  
}

Como vamos usar o Picasso no MensagemAdapter, o Dagger precisa ser capaz de injetar nessa classe e para isso vamos adicionar essa capacidade no ChatComponent:

@Component(modules=ChatModule.class)  
public interface ChatComponent {  
  
 void inject(MainActivity activity);  
 void inject(MensagemAdapter adapter);  
  
}

Como o módulo precisa de uma instância de Application, precisamos indicar ao Dagger que ele deverá usar outra instância ao fazer as injeções. Podemos fazer isso no ChatApplication:

// ChatApplication.java  
  
@Override  
public void onCreate() {  
 component = DaggerChatComponent.builder()  
 .chatModule(new ChatModule(this))  
 .build();  
}

Agora podemos injetar o **Picasso** tanto na **MainActivity** quanto no **MensagemAdapter**, e deixar de utilizá-lo globalmente.