



## Faça como eu fiz: Apresentação de posts

### Enviando imagens para o Cloud Storage

Dentro do formulário de post, em `onViewCreated()`, busque a `ImageView` do post e adicione uma imagem a partir de uma URL. Então, modifique a visibilidade do layout do formulário para que a visibilidade seja visível.

Você pode adicionar a imagem da sua preferência.

Em seguida, confira se a imagem é apresentada ao entrar no formulário e, ao clicar no botão de enviar o post, envie apenas a imagem para o Cloud Storage ao invés do post para o Firestore. Teste o envio da imagem, caso obtenha a resposta de sucesso, verifique no console do Firebase se a imagem foi enviada.

Você pode apenas comentar o código, adicionando duas barras (`//`) no início da linha que envia o post para evitar o envio.

### Enviando a imagem do post

Modifique o código de envio de imagem para o Cloud Storage e integre com a arquitetura do App. Para isso faça o seguinte:

- Crie o método no repositório para receber o id do post e imagem.
  - Envie a imagem para o Cloud Storage e vincule a URL de download com o post a partir do id.

- Modifique o `ViewModel` do formulário para chamar o método de envio da imagem após salvar o post.
- Ajuste o fragment do formulário para enviar a imagem como `ByteArray` para o `viewmodel` ao tentar salvar o post.

Após finalizar a implementação, teste o App, tente salvar o post com imagem, confira se a URL de download da imagem é apresentada no post salvo no Firestore.

## Exibindo imagens do post na lista

Modifique o adapter da lista de posts para que exiba a imagem do post. Para isso, faça o seguinte:

- Adicione o campo de imagem como `String?` com valor padrão `null` tanto para o modelo como para o documento de post.
- No adapter, modifique a visibilidade da `ImageView` do item do post para que seja visível se existir imagem no post e não visível caso não exista
- Tente carregar a imagem do post no `ImageView` com o `coil`.

Execute o App e confira se as imagens de posts com imagens são apresentadas.