

Relatório: Desenvolvimento de Aplicação de Bate-Papo para Android

2025

Sumário

- 01.** Introdução e visão geral
- 02.** Arquitetura e tecnologias utilizadas
- 03.** Funcionalidades
- 04.** Estrutura do Projeto
- 05.** Dificuldades
- 06.** Conclusão

Introdução e visão geral

O UaiPapo é um aplicativo de mensagens em tempo real desenvolvido para a plataforma Android. O projeto foi criado como parte do curso de Programação de Dispositivos Móveis do Bacharelado em Sistemas de Informação da Universidade Federal de Uberlândia.

O objetivo principal do software é fornecer uma plataforma funcional para comunicação instantânea, permitindo que os usuários se cadastrem, criem canais de bate-papo e troquem mensagens de texto e imagem em tempo real.

Arquitetura e tecnologias utilizadas

- **Linguagem de Programação:** O aplicativo é desenvolvido inteiramente em Java, que também foi utilizado para a interface de usuário.
- **Arquitetura:** O projeto segue o padrão de arquitetura MVVM (Model-View-ViewModel). A separação de responsabilidades é clara, com ViewModels (SignInViewModel, HomeViewModel, ChatViewModel, etc.) gerenciando a lógica de negócios e o estado da UI, enquanto as telas (Screens) em Compose atuam como a camada de visualização.
- **Backend e Banco de Dados (BaaS):** O aplicativo utiliza o Firebase como seu principal serviço de backend:
 - **Firebase Authentication:** Para autenticação de usuários, utilizando número de telefone (OTP).
 - **Firebase Realtime Database:** Para o armazenamento e sincronização em tempo real de mensagens e canais.
 - **Firebase Cloud Messaging:** Para o envio de notificações push quando novas mensagens são recebidas.
- **Armazenamento de Mídia:** Além do Firebase, o projeto integra o Supabase Storage para o upload e armazenamento de imagens enviadas no chat..

Funcionalidades

O aplicativo já possui um conjunto de funcionalidades essenciais para um aplicativo de mensagens:

- Autenticação de Usuários:
 - Tela de cadastro (SignUpScreen) que permite a criação de uma nova conta com nome e número de telefone.
 - Tela de login (SignInScreen) que utiliza o número de telefone para autenticação.
 - Verificação via OTP (One-Time Password) para validar o número de telefone do usuário.
- Tela Principal e Canais:
 - A HomeScreen exibe uma lista de contatos para bate-papo disponíveis.
 - Funcionalidade para adicionar novos grupos de conversa com contatos existentes por meio de um diálogo modal.
- Bate-papo em Tempo Real:
 - A ChatScreen é a tela de conversa, onde os usuários podem trocar mensagens dentro de um canal específico.
 - As mensagens são exibidas em tempo real, com uma diferenciação visual para as mensagens do usuário atual e as dos outros participantes.
- Envio de Imagens:
 - Os usuários podem enviar imagens selecionadas da galeria do dispositivo ou tiradas diretamente pela câmera.
- Notificações:
 - O serviço FirebaseMessageService é configurado para receber e exibir notificações push quando novas mensagens chegam, mantendo os usuários engajados mesmo com o aplicativo em segundo plano.

Estrutura do projeto

O código-fonte está bem organizado em pacotes, seguindo as melhores práticas do desenvolvimento Android:

- `model`: Contém as classes de dados como `Message` e `Channel`, que definem a estrutura dos objetos utilizados no aplicativo.
-
- `ui/feature`: Este pacote é subdividido por funcionalidade (`auth`, `chat`, `home`), cada um contendo suas respectivas telas (`Screen`) e `ViewModels`.
- `ui/theme`: Define o tema do aplicativo, incluindo cores, tipografia e estilos.

Principais dificuldades

Durante o desenvolvimento do UaiPapo, a equipe enfrentou diversos desafios técnicos, as principais dificuldades foram:

- **Custo de Serviços e Busca por Alternativas:** Uma barreira inicial foi o custo associado a certas funcionalidades em plataformas de Backend-as-a-Service (BaaS). Funções específicas ou limites de uso mais altos em serviços como o Firebase eram parte de planos pagos. Isso motivou a equipe a pesquisar e implementar alternativas que iria garantir a viabilidade do projeto sem custos operacionais.
- **Implementação da Autenticação via Telefone:** A configuração da autenticação de usuários via número de telefone (OTP - One-Time Password) com o Firebase Authentication se mostrou um processo complexo. O desafio envolveu não apenas a configuração correta no painel do Firebase, mas também a implementação do fluxo de verificação no lado do aplicativo, gerenciando o envio do código, a verificação, os timeouts e o tratamento de possíveis falhas, garantindo um processo seguro e confiável para o usuário.
- **Configuração de Notificações Push:** Gerar notificações em tempo real no dispositivo do usuário foi um desafio técnico significativo. A implementação exigiu a configuração completa do Firebase Cloud Messaging (FCM), a criação de um serviço (FirebaseMessagingService) no aplicativo para receber as mensagens em segundo plano e a lógica para construir e exibir as notificações de forma correta, respeitando as diferentes versões do sistema Android e suas exigências de canais de notificação.
- **Implementação do Envio de Mídia:** A funcionalidade de enviar imagens no chat apresentou dificuldades relacionadas ao manuseio de arquivos e permissões no Android.
- **Decisões da linguagem:** O projeto foi iniciado em Kotlin e, posteriormente, o grupo optou por utilizar Java.

Grupo

- Rayane Reis Mota - 12311BSI233: UI/UX e Frontend.
- Gabriel Augusto Paiva - 12311BSI245: Backend e Banco de Dados.
- Chloe Anne Scaramal - 12311BSI232: Sincronização de Dados e Cache Offline.
- Vinícius Resende Garcia - 12021BCC027: Arquitetura de Software e Lógica da Aplicação.
- João Pedro Zanetti - 12311BSI230: Funcionalidades Nativas do Android.

Conclusão

O UaiPapo é um projeto acadêmico que demonstra a aplicação prática de tecnologias modernas de desenvolvimento Android. A escolha de Kotlin, Jetpack Compose, Firebase e Hilt resulta em um aplicativo performático, escalável e de fácil manutenção. Com as funcionalidades já implementadas, ele serve como uma excelente base para um aplicativo de mensagens completo.

