

# **Relatório de Desenvolvimento**

## **Aplicativo de Mensagens Instantâneas**

**Projeto:** Desenvolvimento de um aplicativo de mensagens instantâneas para Android

**Data:** 01/09/2025

### **Equipe 09:**

Gabriel Silva Tassara – 12311BSI218

Guilherme Siqueira Botelho – 12311BSI217

Kevin Martins de Souza Barbosa – 12311BSI216

João Lucas Gonçalves Teixeira – 12311BSI201

Marcos Paulo Oliveira Gomes – 12311BSI231

## **1. Introdução**

Esse relatório apresenta o processo de desenvolvimento do nosso aplicativo de mensagens instantâneas para Android. O objetivo inicial era criar um app funcional com uma diversidade de funções a escolha do usuário, como por exemplo, notificações personalizadas para grupos ou chats específicos, adicionar ou remover contatos, criação de grupos com múltiplos usuários e funções de adição e remoção de usuário dentre outras funções. Para implementação das condições pedidas, utilizamos a conexão com firebase e implementação da arquitetura MVVM (Repository, ViewModel e View), buscando que o usuário final tivesse a melhor experiência.

## 2. Decisões Técnicas

Para a construção do aplicativo utilizamos a IDE Android Studio na linguagem Kotlin utilizando do Jetpack Compose, a versão mínima do Android escolhida para a implementação do projeto foi a API 24 – Nougat/ Android 7.0. Destacamos então as seguintes escolhas técnicas:

- **Arquitetura MVVM (ModelView-ViewModel):** Decidimos usar o padrão MVVM para deixar o código mais organizado. Com ele, separamos tudo em três partes: a **View** (as telas que o usuário vê), o **ViewModel** (que cuida da lógica da tela) e o **Model** (que busca e guarda os dados). Na prática, isso significou que as Activities ficaram mais simples, e a lógica de negócio ficou mais fácil de testar e dar manutenção.
- **Firebase Cloud Messaging (FCM):** Em síntese, o uso do FCM foi definido como um **serviço de entrega**. O FCM cuida de toda a complexidade de rede e bateria para garantir que a mensagem chegue ao dispositivo. Uma vez entregue, o firebase Cloud Messaging a recebe e, em seguida, utiliza as APIs padrão do Android para transformar os dados dessa mensagem em uma notificação visível e interativa para o usuário final
- **Kotlin e Coroutines:** Usamos **Kotlin** por ser a linguagem padrão para Android. Para as tarefas que dependem de rede, como buscar dados do Firebase, adotamos as **Coroutines**. Elas ajudaram a organizar o código assíncrono de uma forma muito mais limpa, o que deixou o código mais legível e fácil de depurar.

## 3. Divisão dos Papéis

Buscando a organização do projeto, foi realizada a designação da responsabilidade de cada membro do grupo de acordo com as principais áreas do projeto.

- **Gabriel Silva Tassara - Desenvolvedor Full-Stack / Arquiteto de Software**
  - Iniciou a base do projeto, criando o repositório entre a interface (UI) e os dados do back-end (lista de conversas)
  - Desenvolveu uma funcionalidade que exige a conexão entre a interface (UI) e os dados do back-end (lista de conversas).
  - Implementou a lógica de back-end para a criação de grupos e gerenciamento de participantes.
  - Ficou responsável pela apresentação final e demonstração, tarefas que exigem um conhecimento global do projeto.

▪ **Guilherme Siqueira Botelho - Desenvolvedor Back-end**

- Definiu a arquitetura de dados criando as classes de modelo (Usuario, Mensagem, Conversa).
- Implementou a funcionalidade central do aplicativo: o envio e recebimento de mensagens em tempo real.
- Trabalhou em lógicas complexas de back-end, como o status das mensagens (enviada, lida) e a adaptação para grupos.
- Finalizou com a revisão e organização do código, uma tarefa focada na qualidade da lógica interna.

▪ **Kevin Martins De Souza Barbosa- Desenvolvedor Front-end/UI-UX**

- Começou com a tarefa de UX, desenhando os Wire frames das telas principais.
- Implementou a interface da tela de chat, o componente mais interativo do aplicativo.
- Integrou funcionalidades visuais e de mídia na interface, como o envio de imagens.
- Concluiu sua participação criando o vídeo de demonstração, focando em como o produto é apresentado visualmente ao usuário.

▪ **Marcos Paulo Oliveira Gomes- Desenvolvedor Back-end e DevOps**

- Realizou a configuração inicial da infraestrutura na nuvem, habilitando os serviços do Firebase.
- Implementou o fluxo de autenticação de usuários, uma tarefa de back-end crítica para a segurança.
- Configurou as notificações push, integrando o app com um serviço de nuvem para engajamento do usuário.
- Foi responsável pelos testes completos, correções e pela geração do pacote final (APK), garantindo a qualidade e a "entrega" do produto.

- **João Lucas Gonçalves Teixeira:** Desenvolvedor de Software e Analista de Documentação
  - Iniciou e finalizou a documentação e o relatório do projeto, registrando decisões, papéis e dificuldades.
  - Implementou funcionalidades que envolvem tanto UI quanto lógica, como a tela de edição de perfil e o logout.
  - Desenvolveu funcionalidades de back-end importantes, como o status online/offline e a busca por contatos.

## 4. Principais Dificuldades Encontradas

Durante o desenvolvimento do projeto enfrentamos algumas situações de dificuldades em questão de implementação e resolução de bugs.

- **Implementação de upload de mídias no aplicativo:**

Durante o desenvolvimento do nosso aplicativo de mensagens, um dos conjuntos de desafios mais significativos e instrutivos surgiu na implementação da funcionalidade de envio de mídias, como imagens e outros arquivos.

O primeiro obstáculo foi a gestão de permissões de acesso no sistema Android. Além de ser necessário declarar as devidas permissões no arquivo `AndroidManifest.xml`, foi preciso implementar o sistema de solicitação de permissões em tempo de execução. Isso exigiu a criação de um fluxo de usuário que lida com a solicitação de acesso à galeria ou aos arquivos do dispositivo e trata os casos em que o usuário nega a permissão, garantindo que o aplicativo não falhe e informe o usuário adequadamente.

- **Sincronização de dados:**

Superada a etapa de permissões, enfrentamos a complexidade na sincronização de dados e na otimização do desempenho. O processo de enviar uma mídia não é uma única ação, ele envolve o upload do arquivo para o serviço de armazenamento em nuvem, a obtenção de um link de acesso e, só então, o envio desse link como uma mensagem no banco de dados para que os outros usuários possam visualizá-lo. Garantir a sincronização de estado, para que o remetente visse um status de "carregando" e o destinatário só recebesse a mensagem quando a mídia estivesse pronta exigiu um tratamento cuidadoso da lógica assíncrona.

- **Optimização do tempo de envio de mídia:**

Aliado a isso, um desafio crucial foi a otimização do tempo de envio. O upload de arquivos de mídia em seu tamanho original se mostrou um processo lento e com alto consumo de dados. Até o presente momento não encontramos uma solução que seja satisfatória para o problema.

## **5. Conclusão**

No final das contas, o desenvolvimento do nosso app de mensagens possui algumas funcionalidades principais implementadas e conseguimos criar uma plataforma que funciona bem no Android, alcançando a maioria dos objetivos que desejados no começo.

O desempenho tem suas bases em decisões técnicas tomadas, por exemplo, usar a arquitetura MVVM que ajudou a organizar o código e facilitou a manutenção. O Firebase também foi essencial, principalmente para a parte das mensagens, infelizmente não conseguimos implementar as notificações. Usar Kotlin com Coroutines para organizar as tarefas de rede, o código ficou bem mais limpo e fácil de mexer.

A divisão de tarefas que a gente fez no grupo também foi um ponto chave, porque cada um pode focar na sua parte e o trabalho andou mais rápido.

O projeto apresentou uns desafios bem grandes, principalmente na hora de implementar o envio de mídias. Os problemas com permissão no Android e a sincronização dos dados se mostraram complexos. A lentidão no envio das mídias é um problema que a gente ainda não conseguiu resolver de um jeito bom, e até agora não achamos uma solução ideal para isso.

Para concluir, podemos dizer que o projeto foi uma experiência de aprendizado muito grande. A gente aprendeu mais funcionalidades do Android, a usar serviços na nuvem e a trabalhar em equipe. Mesmo com os desafios, conseguir entregar o aplicativo funcionando mostrou que o grupo conseguiu colocar em prática o que a gente aprendeu.