

Universidade Federal de Uberlândia - UFU
Bacharelado em Sistemas de Informação
Programação para Dispositivos Móveis - 2025

Integrantes do grupo: Eduardo de Oliveira Araujo, Maria Clara Silva Borges,
Lucas Silva Figueiredo, Paulo Daniel Forti da Fonseca, Pedro Henrique Lopes
Duarte

Relatório Colaborativo Atividade 5

Uati Zapi - Aplicativo de mensagens instantâneas

Esse relatório detalha o processo de desenvolvimento do nosso projeto do aplicativo Uati Zapi. Ao longo do código, implementamos mensagens instantâneas para a plataforma Android. O documento deste relatório, aborda as principais decisões tomadas sobre as técnicas de design, a divisão e papel para cada membro da equipe e os desafios que encontramos ao longo do processo de desenvolvimento.

1. Decisões tomadas:

- Nós optamos por fazer uma integração com a plataforma do Google Firestore, no qual possibilitou a simplificação do processo de desenvolvimento do aplicativo
- Utilizamos o Firestore como base de um banco de dados NoSQL para armazenar as informações dos usuários em tempo real em relação as mensagens escritas, sendo crucial para a funcionalidade principal do aplicativo
- Utilizamos o Firebase Authentication para uma implementação rápida e segura de vários usuários no aplicativo, utilizando o método de login através do e-mail, senha e número de telefone.
- Utilizamos o Firebase Storage, para o armazenamento de arquivos de mídia, como as fotos de perfil dos usuários.
- A arquitetura do aplicativo Uati Zapi, foi baseada em Activities para cada uma das telas presentes, com o uso eficiente de RecyclerViews e Adapters, como a exibição de contatos, membros do grupos e mensagens.
- Os dados no Firestore, consistem em três collections principais, sendo eles o users, chatrooms e as subcoleções de mensagens dentro de cada chatroom. Para a funcionalidade de busca, as mensagens foram estruturadas com "searchKeywords", permitindo assim uma consulta eficiente e rápida no aplicativo.
- A interface do usuário foi construída com componentes do próprio Android, no qual o estado UI é atualizado em tempo real através do "SnapshotListeners" do Firestore, no

qual a cada chamada, é atualizada na tela automaticamente.

- Buscamos também implementar a busca de mensagens em todas as conversas e a capacidade de silenciar notificações para chats específicos, no qual possibilita um controle melhor para cada usuário do aplicativo.

2. Papeis dos membros da equipe:

- Eduardo de Oliveira Araujo (Desenvolvedor Frontend e UI/UX):

o Responsabilidades: Iniciou o processo da implementação da interface do usuário e da experiência visual do aplicativo. Foi responsável também pela criação dos layouts básicos das telas iniciais, lógica de autenticação e pela ilustração da tela de chat.

- Lucas Silva Figueiredo (Desenvolvedor do Core de Mensagens e Funcionalidades Offline):

o Responsabilidades: Funcionalidades centrais da comunicação do chat, listagem e criação de conversas, sistema de envio e recebimento de mensagens e a funcionalidade de sincronização offline para que o projeto funcione sem uma conexão constante.

- Maria Clara Silva Borges (Desenvolvedora de Funcionalidades Sociais e de Grupo dos usuários):

o Responsabilidades: Funcionalidade de interação do usuário com a plataforma, como a busca de contatos e conversas existentes no app, o sistema de status do usuário (online//offline), a criação de chats em grupo de usuários e implementação de imagens e vídeos no aplicativo.

- Paulo Daniel Forti da Fonseca (Desenvolvedor de Funcionalidades de Mídia e Chat):

o Responsabilidades: Funcionalidade complementares e mídia, no qual desenvolveu o envio de imagens, o recurso de mensagens fixadas e o sistema de filtro de mensagens.

- Pedro Henrique Lopes Duarte (Especialista em Backend Firebase e Autenticação):

o Responsabilidades: Funcionalidade de estrutura de dados do projeto, como a configuração do projeto no Firebase, a definição da arquitetura do banco de dados no projeto, a implementação da lógica de cadastro e login de usuários e o desenvolvimento do sistema de notificação push implementada.

- TODOS OS MEMBROS:

o Responsabilidades: Os membros do grupo colaboraram para fazer o planejamento, os testes e as implementações difíceis ao longo do projeto. Ocorreram diversas reuniões de alinhamento, tratativas de bugs e erros afim de obtermos a conclusão do projeto no github.

3. Principais Dificuldades:

- Sincronização e autenticação com o Firebase e FireStore, no qual a garantia de que as mensagens e os status fossem atualizados instantaneamente, foi trabalhoso.
- A lógica de implementação da remoção de um usuário em um grupo, a garantia de que a lista de membros, fossem atualizadas corretamente, e a inconsistência dos códigos no processo de produção do projeto.
- A busca de mensagens no FireStore com uma modelagem de dados eficiente.
- Enfrentamos dificuldade sobre as questões de versões no Android Studio, a sincronização de dados de mensagens para online e offline.
- Sincronização com o tópico do trabalho "Recuperação de Senha".
- Visualização prévia de imagens e vídeos.

4. Uso das LLMs:

- Utilizamos as LLMs para auxílio nas depurações e correções de códigos, a fim de otimizar o tempo de desenvolvimento do projeto, com o resultado de um código linear e legível. O uso estratégico das LLMs, favoreceu a produtividade e a qualidade do código entregável. O foco foi a utilização com um assistente de programação (como o Gemini na própria interface do Android Studio), no qual a depuração ágil e resolução de erros complexos, garantiu uma maior legibilidade do programa. Assim, o uso dessas inteligências artificiais, contribuiu para um código mais modular com as regras de programação e possui uma facilidade de manutenção.