



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DA PARAÍBA**

**CURSO DE BACHARELADO EM ENGENHARIA DE
COMPUTAÇÃO**

RAFAELA BORBA FALCÃO CIRINO

**CHAT ACADÊMICO – PROTÓTIPO DE APLICATIVO DE BATE-
PAPO PARA INSTITUIÇÃO DE ENSINO**

CAMPINA GRANDE – PB

2023

RAFAELA BORBA FALCÃO CIRINO

CHAT ACADÊMICO – PROTÓTIPO DE APLICAÇÃO DE BATE-PAPO PARA INSTITUIÇÃO DE ENSINO

Trabalho de Conclusão de Curso apresentado ao Programa de Graduação de Engenharia de Computação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – Campus Campina Grande - PB, em cumprimento às exigências parciais para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Igor Barbosa da Costa

C578c Cirino, Rafaela Borba Falcão.

Chat acadêmico: protótipo de aplicação de bate-papo para instituição de ensino / Rafaela Borba Falcão Cirino. - Campina Grande, 2023.

23 f. : il.

Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) - Instituto Federal da Paraíba, 2023.

Orientador: Prof.Dr. Igor Barbosa da Costa.

1. Engenharia da Computação - Desenvolvimento de aplicativo
 2. Sistema operacional - Android
 3. Comunicação em tempo real - realtime database
 4. Comunicação por mensagens de texto - firebase cloud messaging
- I. Igor Barbosa da Costa. II. Título.

CDU 004.451

Dedico este trabalho de conclusão de curso ao meu marido, Diego Cirino, por todo incentivo e motivação ao decorrer deste curso. Que possamos trilhar novos e brilhantes caminhos juntos.

Agradecimentos

- Em primeiro lugar Àquele que me deu o dom da vida e salvação, Deus, sem Ele nada seria possível.
- Agradeço ao meu marido por todo incentivo, estímulo e compreensão em toda essa jornada fazendo com que tudo ficasse mais leve.
- Agradeço à toda a minha família por sempre acreditarem e confiarem em mim.
- Agradeço aos amigos do curso que tanto me ajudaram no entendimento das matérias e por todo o companheirismo.
- Agradeço ao professor Igor Barbosa da Costa por toda orientação e prontidão em todas as etapas desse trabalho.
- Agradeço especialmente ao professor César Rocha Vasconcelos por todos os ensinamentos da disciplina que inspirou na elaboração deste projeto.
- Ao Instituto Federal de Educação, Ciência e Tecnologia da Paraíba que me proporcionou todas as ferramentas para que eu pudesse me graduar em Engenharia de Computação fazendo com que eu me sentisse acolhida todos os dias.
- Em especial agradeço à Lara por todos os momentos compartilhados e por ter me dado alegria todos os dias de sua vida.

“Porque dele e por ele, e para ele, são todas as coisas; glória, pois, a ele eternamente. Amém” – Romanos 11:36

Chat Acadêmico – Protótipo de aplicativo de bate-papo para instituição de ensino

Resumo. Com o aumento da demanda por comunicação direta e efetiva na comunidade acadêmica, é fundamental estabelecer mecanismos para preservar a segurança dos dados pessoais. Nesse contexto, foi desenvolvido um aplicativo oficial destinado à comunidade acadêmica do Instituto Federal da Paraíba (IFPB), com o objetivo de fornecer segurança, escalabilidade e centralização na comunicação entre seus membros.

Palavras-chave: *Java, Android, Firebase, Realtime Database, Firebase Cloud Messaging.*

Abstract. *As the demand for direct and effective communication in the academic community continues to grow, it is crucial to establish mechanisms to preserve the security of personal data. In this context, an official application has been developed for the academic community of the Federal Institute of Paraíba (IFPB) with the aim of providing security, scalability, and centralized communication among its members.*

Keywords: *Java, Android, Firebase, Realtime Database, Firebase Cloud Messaging.*

Lista de Figuras

Figura 1 – Tela do Android Studio.	5
Figura 2 – Tela do console do Firebase.	6
Quadro 1 – Requisitos Funcionais.	9
Quadro 2 – Requisitos Não Funcionais.	9
Figura 3 – Tela de <i>splash</i> do aplicativo.	10
Figura 4 – Tela de <i>login</i>	10
Figura 5 – Tela de cadastro.	11
Figura 6 – Tela de perfil e conversas recentes.	11
Figura 7 – Tela de usuários cadastrados.	12
Figura 8 – Tela para adicionar usuário com notificação.	12
Figura 9 – Tela de bate-papo.	13

Sumário

1	Introdução	1
1.1	Relevância do Tema	2
1.2	Objeto de Pesquisa	2
1.2.1	Contextualização do Problema de Pesquisa.	2
1.2.2	Hipótese	2
1.3	Objetivos de Estudo	2
1.3.1	Objetivo Geral	2
1.3.2	Objetivos Específicos	2
1.4	Justificativa.	3
1.5	Estrutura do Trabalho.	3
2	Referencial Teórico	4
2.1	Java	4
2.2	Android	4
2.3	Github	5
2.3.1	Firebase	6
2.3.2	Cloud Firestore	6
2.4	API Firebase Cloud Messaging – FCM	7
2.5	Trabalhos Relacionados	7
3	Metodologia	8
4	Resultados	8
4.1	Requisitos e Protótipos de Telas	8
4.1.1	Requisitos	8
4.1.2	Protótipos de Telas	9
5	Conclusão e Melhorias Futuras	13
5.1	Conclusão	13
5.2	Melhorias Futuras	14
	Referências	15

1. Introdução

Este capítulo apresenta considerações iniciais do presente trabalho, sua relevância, objeto de pesquisa, objetivos de estudo e justificativa, finalizando, com a apresentação da estrutura dos capítulos seguintes.

1.1 Relevância do Tema

Com o crescente uso de tecnologia no meio acadêmico, a comunicação entre os integrantes da comunidade acadêmica tornou-se mais importante e necessária. Os aplicativos de bate-papo passaram a ser muito utilizados para este fim, tornando-se um dos principais meios de comunicação entre os membros da comunidade, consentindo a troca de mensagens instantâneas entre estudantes, professores e demais membros da instituição.

No entanto, com o avanço da tecnologia e a necessidade de se estar cada vez mais próximo, surgiram inúmeros fatores que precisam ser levados em consideração quando dessa aproximação tecnológica, tal como compartilhamento de dados sensíveis, dentre outros. Desde um número de telefone pessoal a uma localização compartilhada em tempo real, esses dados em mão de pessoas com objetivos maliciosos, pode causar danos irreparáveis, e adiciona-se a este fator o compartilhamento de dados sensíveis, como número de telefone pessoal, que podem ser evitados de serem compartilhados.

O desenvolvimento de um aplicativo de bate-papo acadêmico pode contribuir para o aumento na segurança no que tange ao compartilhamento de dados sensíveis na comunicação e colaboração entre a comunidade acadêmica. O diferencial deste projeto é a obrigatoriedade de se cadastrar no aplicativo apenas usando o e-mail acadêmico, sem com isso, compartilhar dados pessoais como o número de telefone. Espera-se que este trabalho contribua para o desenvolvimento de ferramentas que promovam a comunicação entre a comunidade acadêmica e a colaboração entre seus membros, melhorando desta forma a qualidade do ensino, pois a comunicação seria mais efetiva e centralizada em um único meio de comunicação, e a comunicação dos integrantes e aprendizagem sem deixar de ressaltar a segurança e o compartilhamento de dados pessoais.

1.2 Objeto de Pesquisa

1.2.1 Contextualização do Problema de Pesquisa

O problema em análise aborda a importância de manter dados pessoais sigilosos na comunidade acadêmica, sem interferir na vida pessoal dos usuários. É comum o uso de aplicativos de bate-papo como meio de comunicação entre estudantes, professores e demais membros da instituição, porém, o compartilhamento de dados sensíveis como número de telefone, por exemplo, pode gerar problemas de segurança e privacidade.

1.2.2 Hipótese

A hipótese é de que um aplicativo de bate-papo que permita o registro apenas com o e-mail acadêmico pode facilitar a comunicação entre os usuários, reduzindo a espera por resposta e centralizando os usuários em uma única plataforma. Essa solução pode contribuir para a melhoria na comunicação e colaboração entre a comunidade acadêmica, garantindo a segurança e privacidade dos dados pessoais.

1.3 Objetivos de Estudo

Nesta seção serão apresentados os objetivos gerais e específicos do trabalho com o fim de atingir o resultado esperado.

1.3.1 Objetivo Geral

O objetivo geral deste trabalho é o desenvolvimento de um protótipo de aplicativo móvel que permita que membros da comunidade acadêmica, que se cadastraram, sejam capazes de se comunicarem através de troca de mensagens instantâneas, dentre outros fatores sem necessariamente haver a troca de dados pessoais, no caso, o número de telefone.

1.3.2 Objetivos Específicos

São objetivos específicos do desenvolvimento do aplicativo de bate-papo:

- Permitir a visualização dos usuários do aplicativo;
- Permitir o envio de notificações para usuários ausentes, bem como notificações de erros relacionados a cadastro ou falha no *login*;

- Desenvolvimento de um banco de dados não relacional utilizando o Firebase.

1.4 Justificativa

Este trabalho, na forma de um protótipo de aplicativo consolida os conceitos adquiridos durante a graduação em Engenharia da Computação, aplicando-os ao desenvolvimento de um aplicativo de bate-papo para uso acadêmico. Através do uso da linguagem Java, suas ferramentas e frameworks, pretende-se demonstrar a versatilidade dessa linguagem no desenvolvimento de aplicações móveis.

A necessidade de comunicação segura entre a comunidade acadêmica, sem a troca de dados pessoais, é fundamental para melhorar a comunicação e incentivar a troca de informações e dúvidas relacionadas à instituição. Reunir toda a comunidade acadêmica em um aplicativo oficial da instituição pode ser uma forma de solucionar o problema, visto que aplicativos de mensagens instantâneas comerciais não suportam grupos com uma quantidade superior a 1.024, como é o caso do WhatsApp.

Portanto, a criação de um protótipo de aplicativo *Android*, o sistema operacional mais utilizado, responsável por gerenciar conversas entre os participantes, é justificável e essencial para melhorar a comunicação entre professores, alunos e funcionários da instituição. Uma vez que se apresenta como uma ferramenta importante para incentivar a troca de conhecimento e esclarecimento de dúvidas, concentrando a comunicação entre os participantes cadastrados.

1.5 Estrutura do Trabalho

Este artigo é composto por várias seções, sendo esta a primeira. Nela são apresentadas as considerações iniciais e contextualização do trabalho, incluindo os objetivos e a justificativa.

A seção 2 trata sobre o referencial teórico estabelecido para o desenvolvimento deste projeto. Ela refere-se às tecnologias e ferramentas utilizadas bem como trabalhos relacionados ao tema.

Seção 3 são apresentadas a metodologia adotada na formulação deste trabalho visando alcançar os objetivos propostos.

Seção 4 especifica os requisitos para o trabalho, sendo eles funcionais e não funcionais, bem como os protótipos das telas desenvolvidas para o aplicativo.

Por fim, na seção 5 encontram-se a conclusão e melhorias propostas para possibilidade de continuidade no futuro.

2. Referencial Teórico

Nesta seção serão abordadas as ferramentas, linguagem de programação, tipo de banco de dados e API (*Application Programming Interface*) utilizada para envio de mensagens entre plataformas no desenvolvimento do protótipo de aplicativo.

2.1 Java

Java é uma linguagem de programação orientada a objetos que foi desenvolvida na década de 90 pela *Sun Microsystems* e atualmente é mantida pela Oracle Corporation desde 2008 (DEITEL, 2016). Por ser uma linguagem consolidada, multiplataforma, escalável e orientada a objetos, há uma maior facilidade em criar aplicativos modulares e robustos. É uma linguagem de programação com grande popularidade no desenvolvimento de aplicativos móveis, com uma vasta gama de bibliotecas Java que podem ser usadas para desenvolver qualquer tipo de aplicativo, muitos deles podendo ser utilizadas na plataforma Android (LESIECKI, 2001).

2.2 Android

O Android é um sistema operacional baseado em Linux e multiusuário, onde cada aplicativo é executado em uma máquina virtual (*Virtual Machine*) isolada de outros aplicativos. O sistema é compatível com uma ampla variedade de dispositivos e recursos. (Estevão, 2018)

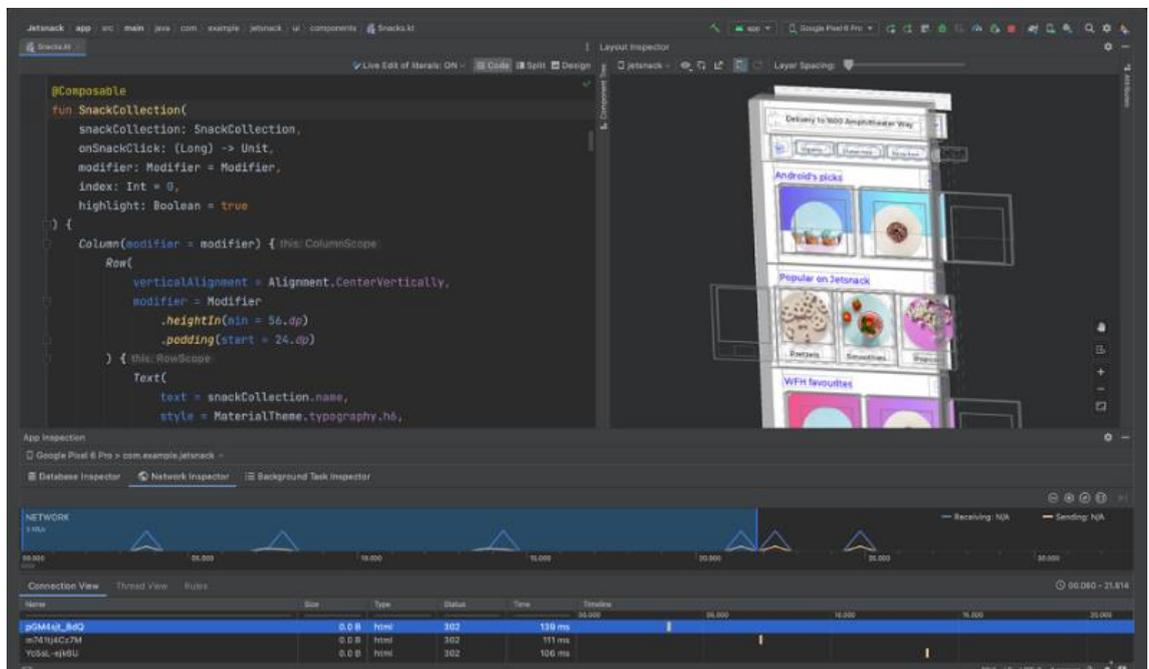
Para desenvolver aplicativos nativos para a plataforma Android, é utilizado o kit de desenvolvimento de software Android SDK. A linguagem de programação Java é comumente utilizada para escrever aplicativos que são executados na ART (*Android Runtime*) e Dalvik, máquinas virtuais personalizadas projetadas para rodar em dispositivos *Android* com um *kernel* Linux.

O Android é uma plataforma popular devido ao fácil acesso ao mercado por meio da loja de aplicativos *Google Play Store*. Além disso, ser *open-source* permite personalizar o sistema operacional para atender às necessidades específicas do desenvolvedor.

Neste projeto, foi escolhido o Android nativo por ser mais rápido e confiável, permitindo trabalhar com ferramentas de baixo nível e oferecendo a melhor experiência ao usuário, como notificações *push*. Para garantir uma experiência de usuário ideal, é necessário seguir as melhores práticas.

O Android Studio, desenvolvido pela Google e apresentado na Figura 1, é a IDE (*Integrated Development Environment*) oficial utilizada neste projeto e oferece diversos recursos para desenvolver aplicativos para dispositivos Android de todos os tipos.

Figura 1: Tela do Android Studio



Fonte: Android Studio (2023)

2.3 Github

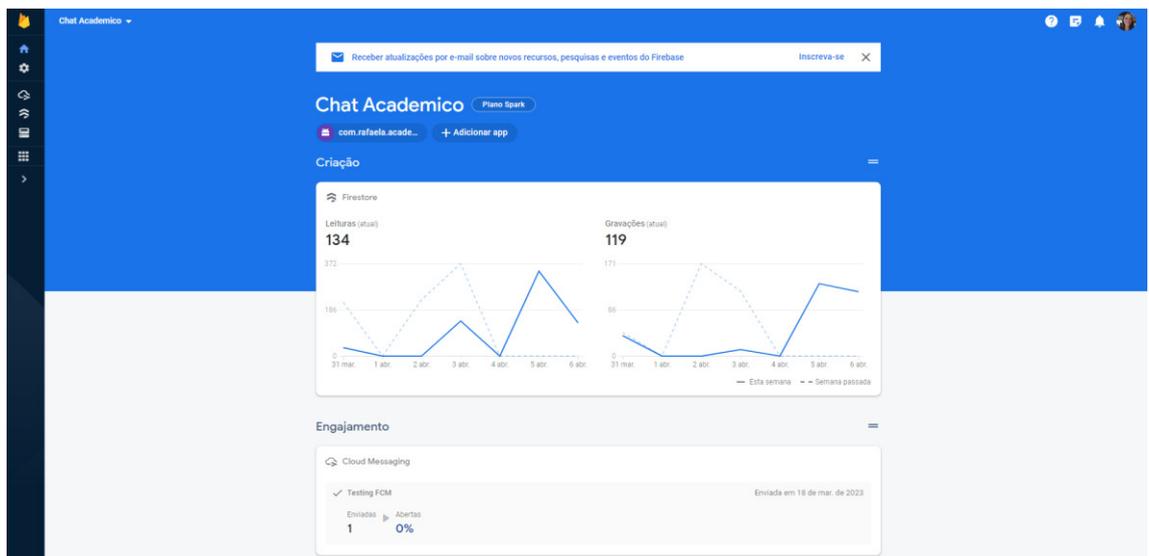
O Github é uma plataforma de hospedagem de código-fonte que utiliza o Git, um *software open source* mantido pelo Linux, para o controle de versão. Ele permite que equipes colaborem de forma efetiva em projetos, possibilitando o

gerenciamento de tarefas e problemas relacionados ao projeto, rastreamento de mudanças e colaboração em tempo real (MATOLA, 2020). Essa plataforma é considerada essencial para desenvolvedores e foi escolhida para este trabalho devido a essas vantagens.

2.4 Firebase

Firebase é uma plataforma móvel do Google que conta com serviços de computação em nuvem de *back-end* e plataformas de desenvolvimento de aplicativos móveis sendo possível realizar o escalonamento automático (Firebase 2021). Hospeda banco de dados não relacionais (NoSQL). Pode-se visualizar como exemplo da tela do console do Firebase na Figura 2. No console podemos encontrar dados em tempo real da aplicação como a coleção de usuários cadastrados e conversas realizadas em tempo real, bem como o sistema de mensageria da aplicação e configurações gerais para o projeto armazenado na plataforma.

Figura 2: Tela do console do Firebase



Fonte: *print screen* do console plataforma Firebase (2023)

2.4.1 Cloud Firestore

O Cloud Firestore é um banco de dados NoSQL hospedado na nuvem que permite armazenar e sincronizar dados em tempo real entre usuários, se

baseia nos resultados do Real Database, banco de dados original do Firebase (Firebase, 2023). Ele utiliza tecnologias de streaming para lidar com cargas de trabalho que sofrem mudanças constantes, como as salas de bate-papo deste projeto. Os dados são armazenados em formato JSON (*JavaScript Object Notation*) e sincronizados em tempo real.

2.5 API Firebase Cloud Messaging – FCM

É uma solução de mensagens multiplataforma que permite entregar mensagens de forma confiável e sem custo. Pode-se transferir uma carga útil de até 4 KB para um aplicativo cliente para o caso de uso de mensagens instantâneas, por exemplo. Com o uso da API FCM (*Firebase Cloud Messaging Application Programming Interface*), é possível enviar notificações e direcionar mensagens (Firebase, 2023).

2.6 Trabalhos Relacionados

Foram analisadas aplicações que fornecem serviço similar ao tema, visando de forma principal o entendimento e a execução do projeto. Exemplos de aplicativos analisados: *WhatsApp*, *Telegram* e *Signal*.

O *WhatsApp* é o aplicativo mais popular, tendo se juntado ao *Facebook* em 2014, atualmente *Meta*, e possui mais de dois bilhões de usuários em mais de 180 países. É usado como aplicativo de mensagens instantâneas, chamadas de voz e vídeo, bem como envio de documentos através de uma conexão com a internet.

O *Telegram* é um serviço de mensagens instantâneas baseado em *cloud* e pode ser usado na web ou como aplicativo móvel. Tem seu foco na segurança e velocidade. Os usuários podem fazer chamadas em vídeo, voz, enviar arquivos de qualquer tipo e figurinhas.

Signal é um aplicativo criptografado multiplataforma desenvolvido pela *Signal Foundation*, organização sem fins lucrativos. É considerado um dos aplicativos mais eficientes, tanto em segurança quanto em privacidade, sendo adotado pelo Congresso dos Estados Unidos e Comissão Europeia.

3. Metodologia

O método para o desenvolvimento ágil deste projeto será o *Scrum*. Dentre outras metodologias existentes, o *Scrum* é um *framework* para gerenciamento de projetos tendo ênfase no desenvolvimento de *software* e garantindo assim que prazos, metas e objetivos do projeto sejam alinhados e conhecidos em todas as fases. O *Scrum* tem foco no projeto e visa em melhorias constantes baseadas em *feedbacks* agregando mais produtividade nos processos.

4. Resultados

Serão abordados neste tópico requisitos, sendo este divididos em funcionais e não funcionais e protótipos das telas que foram desenvolvidas.

4.1 Requisitos e Protótipos de Telas

4.1.1 Requisitos

Entende-se como requisitos as definições de uma propriedade, comportamento ou restrições de operação que um produto ou serviço deve atender (WTHREEX.COM, 2014).

Foram separados os requisitos funcionais e não funcionais do projeto. Os requisitos funcionais especificam o que o sistema deverá fazer, ou seja, todos os problemas e necessidades que devem ser atendidos e resolvidos pelo sistema por meio de funções ou serviços. Já os requisitos não funcionais são todos aqueles relacionados à forma como o software tornará realidade o que se planejou. Ou seja, descreve como o software será desenvolvido e os requisitos funcionais no que será feito. Os requisitos funcionais e não funcionais estão representados nos Quadros 1 e 2 respectivamente.

Quadro 1 – Requisitos Funcionais

ID	Descrição O sistema deverá:
RF001	Apresentar tela de <i>login</i> com usuário e senha e opção de cadastro para usuário
RF002	Exibir na tela principal, após <i>login</i> , usuário conectado e histórico de conversas
RF003	Permitir acesso a lista de usuários cadastrados
RF004	Permitir ao usuário buscar um usuário
RF005	Permitir ao usuário enviar mensagens
RF006	Permitir que novo usuário se cadastre no sistema
RF007	Permitir que novo usuário só possa se cadastrar no sistema usando e-mail acadêmico
RF008	Permitir que usuário só possa se cadastrar inserindo os dados completos: imagem de perfil, nome, e-mail e senha

Fonte: Elaborado pela autora (2023)

Quadro 2 – Requisitos Não Funcionais

ID	Descrição O sistema deverá:
RNF001	Só pode ser acessado através de <i>login</i> e senha de usuários já cadastrados
RNF002	Tem disponibilidade restrita a acadêmicos do IFPB
RNF003	Foi desenvolvido utilizando a linguagem Java, com arquivos XML (<i>Extensible Markup Language</i>) que são compostos por layouts, <i>strings</i> e imagens em <i>bitmap</i>
RNF005	Sua compatibilidade se dará para o sistema operacional Android
RNF006	Seus dados serão armazenados no Firebase em formato NoSQL

Fonte: Elaborado pela autora (2023)

4.1.2 Protótipos de Telas

Ao iniciar o aplicativo, abre-se a tela de *splash*, de acordo com a Figura 3, que é uma tela que sobrepõe o aplicativo enquanto ele carrega e nela estão o nome do aplicativo e nome da autora.

Figura 3: Tela de *splash*



Fonte: Autoria própria, código disponibilizado no Github (2023)

Em seguida, o aplicativo é aberto com a tela de *login*, o que pode ser visualizado na Figura 4. Nesta tela o usuário acessará sua conta através do seu e-mail e senha ou criar uma conta caso seja novo usuário.

Figura 4: Tela de *login*

A tela de login do aplicativo Chat Acadêmico. O fundo é branco. No topo, o texto 'Bem Vindos' está em bold, seguido por 'Login para continuar' em uma fonte menor. Abaixo, há dois campos de entrada de texto cinza: 'Email' e 'Senha'. Abaixo dos campos, há um botão de login em azul escuro com o texto 'ENTRAR' em branco. No rodapé, há um link 'Criar Nova Conta' em uma fonte pequena.

Fonte: Autoria própria, código disponibilizado no Github (2023)

Ao clicar em criar nova conta, o usuário é direcionado para a tela de cadastro, como pode ser detalhado na Figura 5, a qual solicita uma imagem de perfil, nome completo, e-mail acadêmico e senha, que deverá ser confirmada para que ambas coincidam.

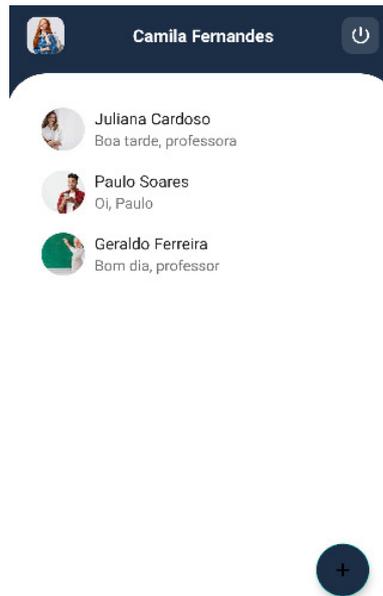
Figura 5: Tela de cadastro

A tela de cadastro, intitulada "Criar Nova Conta", apresenta um formulário centralizado. No topo, há um botão circular com o texto "Adicionar Foto". Abaixo dele, há quatro campos de entrada de texto, cada um com um rótulo: "Nome Completo", "Email", "Senha" e "Repetir a Senha". Na base do formulário, um botão escuro com o texto "INSCREVER" em letras maiúsculas brancas permite a criação da conta. Logo abaixo do botão, o texto "Já possui conta? Entrar" oferece uma opção para usuários existentes.

Fonte: Autoria própria, código disponibilizado no Github (2023)

Ao entrar sendo usuário já cadastrado, ou como usuário novo, a próxima tela é a de perfil, de acordo com a Figura 6, onde aparecem conversas recentes e botão para adicionar nova conversa.

Figura 6: Tela de perfil e conversas recentes



Fonte: Autoria própria, código disponibilizado no Github (2023)

Ao clicar no botão de adicionar, em concordância com a Figura 7, pode-se ver os usuários cadastrados e escolher um novo usuário para iniciar nova conversa.

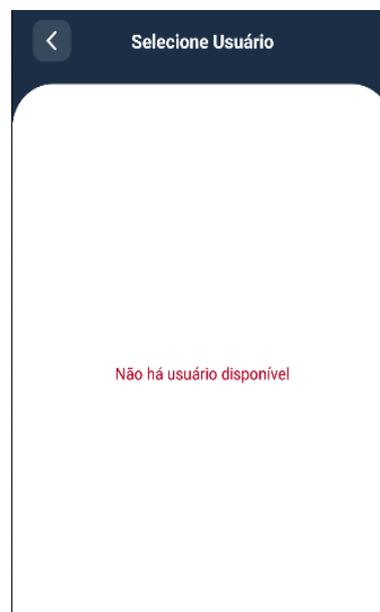
Figura 7: Tela de usuários cadastrados



Fonte: Autoria própria, código disponibilizado no Github (2023)

Caso não tenham usuários cadastrados no sistema, aparece uma notificação, conforme a Figura 8.

Figura 8: Tela para adicionar usuário com notificação



Fonte: Autoria própria, código disponibilizado no Github (2023)

Por fim, a tela de conversas, segundo a Figura 9. Onde fica o histórico de conversas e cada mensagem há a data de envio e hora, contendo também o status do destinatário.

Figura 9: Tela de bate-papo



Fonte: Autoria própria, código disponibilizado no Github (2023)

5. Conclusão e Melhorias Futuras

5.1 Conclusão

O trabalho tem como objetivo o desenvolvimento de um aplicativo de mensagens instantâneas para uso da comunidade acadêmica do IFPB, incentivando desta forma, a prática de comunicação adequada aos padrões acadêmicos.

Além de permissões de cadastro, também foi possível a realização de autenticação de senhas de cadastro. Há também autenticação ao tentar acessar o aplicativo, verifica se está cadastrado e se o e-mail se adequa aos padrões da instituição, bem como notificações quando o usuário está saindo da sua conta, quanto notificações caso algum campo no cadastro não esteja preenchido.

O usuário tem acesso a lista de usuários cadastrados na plataforma, podendo iniciar uma conversa mesmo que o destinatário esteja ausente. Também pode visualizar o histórico de conversas na página de perfil, onde ficam localizados seus dados e abaixo, o histórico de bate-papo. Caso o usuário não

esteja *online*, ele receberá uma notificação no seu aparelho de que recebeu uma mensagem no Chat Acadêmico.

Por fim, o projeto foi configurado com o *Firebase* que armazena os dados que utilizam um banco de dados não relacionais, *Realtime Database*, bem como utiliza os serviços de nuvem do *Cloud Firestore*, facilitando e armazenando os dados da aplicação e envio de notificações. O uso da *Api FCM* possibilita que seja fácil que a aplicação seja escalonada futuramente e entrega mensagens de forma confiável e sem custo.

5.2 Melhorias Futuras

O aplicativo foi desenvolvido como um protótipo para inspirar o desenvolvimento de um projeto final que deverá ser usado como um aplicativo de mensagens instantâneas para a comunidade acadêmica do IFPB. Desta forma, o projeto futuro a ser finalizado deverá ter conectividade com o SUAP (Sistema Unificado de Administração) para que seja verificado se o usuário que está se cadastrando está regularmente matriculado no IFPB.

Também será necessário que haja distinção entre professores, alunos e funcionários da instituição, para que não tenha confusão entre usuários. Além disso, pode-se adicionar a formação de grupos no chat para que os professores possam enviar mensagens direcionadas a uma turma em específico e que os alunos possam criar uma sala exclusiva para aquela turma em específico.

Acredita-se que a estrutura do aplicativo foi elaborada visando que hajam melhorias futuras para que seja complementado o que foi feito. O desenvolvimento do projeto poderá continuar visando um produto que seja capaz de entregar à comunidade acadêmica um produto finalizado com condições de beneficiar aos componentes da instituição.

REFERÊNCIAS

ANDROID. Conceitos e padrões. 2023. Disponível em:

<https://developer.android.com/design/index.html?hl=pt-br> Acesso em 20 de Março de 2023.

ANDROID STUDIO. Download e informações oficiais. 2023. Disponível em:

<https://developer.android.com/studio> Acesso em 10 de Janeiro de 2023.

Código do projeto disponibilizado em:

<https://github.com/rafaelacirino/AcademicoChat> Acesso em 17 de Abril de 2023.

Console do Firebase. 2023. Disponível em:

<https://console.firebase.google.com/u/0/> Acesso em 15 de Abril de 2023.

DEITEL, PAUL; DEITEL, HARVEY. **Java: Como Programar**. Pearson Universidades, 2016.

ESTEVIÃO, **Conceitos Básicos Para Programar Android [2018]**. Disponível em: <https://www.devmedia.com.br/conceitos-basicos-para-programar-para-android/40112> Acesso em 10 de Fevereiro de 2023.

FIREBASE. **Escolher Um Banco de Dados: Cloud Firestore ou Realtime Database. 2023**. Disponível em: <https://firebase.google.com/docs/firestore/rtdb-vs-firestore?hl=pt-br> Acesso em 17 de Abril de 2023.

FIREBASE. Documentação oficial. 2023. Disponível em:

<https://firebase.google.com/docs/firestore?hl=pt-br> Acesso em 15 de Fevereiro de 2023.

FIREBASE. **Firestore Cloud Messaging. 2023**. Disponível em: <https://firebase.google.com/docs/cloud-messaging?hl=pt-br> Acesso em 29 de Março de 2023.

LESIECKI, NICHOLAS, **Java Tools For Extreme Programming: Mastering Open**

Source Tools, Including Ant, JUnit, and Cactus, John Willey & Sons, 2001.

MATOLA, Rodrigo, **Github: O Que É E Como Usar**. Disponível em: <https://blog.geekhunter.com.br/github-o-que-e-como-usar/> Acesso em 25 de Fevereiro de 2023.

MARTIN, ROBERT C., **Arquitetura Limpa: O Guia do Artesão Para Estrutura e Design de Software**. Alta Books, 2019.

MARTIN, ROBERT C., **Desenvolvimento Ágil Limpo: De Volta às Origens.**

Alta

Books, 2020.

RAMPAZZO, L. (2005). **Metodologia científica.** Edições Loyola.

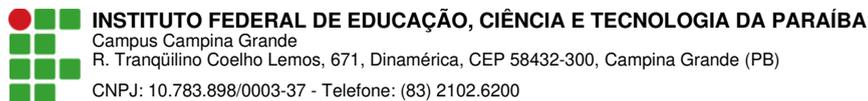
TEJADA, Zoiner, **Dados Não Relacionais e NoSQL.** Microsoft. Disponível em:

<https://learn.microsoft.com/pt-br/azure/architecture/data-guide/big-data/non-relational-data>. Acesso em 20 de Novembro de 2022.

TELEGRAM. 2023. Disponível em: <https://telegram.org/k/> Acesso em 30 de Março de 2023.

SIGNAL. 2023. Disponível em: https://signal.org/pt_BR/ Acesso em 30 de Março de 2023.

WHATSAPP. 2023. Disponível em: https://www.whatsapp.com/?lang=pt_br Acesso em 30 de Março de 2023.



Documento Digitalizado Restrito

Versão Final do TCC

Assunto: Versão Final do TCC
Assinado por: Rafaela Cirino
Tipo do Documento: Resumo
Situação: Finalizado
Nível de Acesso: Restrito
Hipótese Legal: Informação Pessoal (Art. 31 da Lei no 12.527/2011)
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- **Rafaela Borba Falcão Cirino, ALUNO (201911250040) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE**, em 31/05/2023 10:43:14.

Este documento foi armazenado no SUAP em 31/05/2023. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 842247
Código de Autenticação: 5b70db49f1

