

DESENVOLVIMENTO DE UM APLICATIVO DE GERENCIAMENTO DE CHAMADO USANDO FRAMEWORK FLUTTER

Diego Carmino Moretti¹, Karlos Eduardo Cristóvão Silva Nunes¹, Rodrigo Oliveira Plotze¹, Debora Pelicano Diniz¹

¹Faculdade de Tecnologia de FATEC Ribeirão Preto (FATEC)

Ribeirão Preto, SP – Brasil

diego.moretti@fatec.sp.gov.br,
karlos.nunes@fatec.sp.gov.br,
rodrigo.plotze@fatec.sp.gov.br,
debora.diniz2@fatec.sp.gov.br

Resumo. *O presente estudo aborda a falta de organização na abertura de chamados para equipe de tecnologia da informação em uma escola, resultando em perda de produtividade. O trabalho propõe o desenvolvimento de um aplicativo de gerenciamento de chamados, utilizando o framework Flutter e conceitos e técnicas de engenharia de software. O projeto adota o padrão MVC e a Firebase Realtime Database para armazenamento de dados. Embora o aplicativo ainda não esteja em uso, representa uma solução para melhorar o atendimento ao cliente, aplicando conceitos de engenharia de software e Flutter. A intenção é continuar o desenvolvimento no futuro.*

Abstract. *The present study addresses the lack of organization in opening tickets for the information technology team in a school, resulting in a loss of productivity. The work proposes the development of a service request management application, using Flutter framework and concepts and techniques from software engineering. The project adopts the MVC pattern and Firebase Realtime Database for data storage. Although the application is not yet in use, it represents a solution to enhance customer service by applying software engineering and Flutter concepts. The intention is to continue the developing in the future.*

1. INTRODUÇÃO

A falta de organização na abertura de chamados para a equipe de tecnologia da informação pode levar à perda de produtividade e, consequentemente, aumentar os custos para resolução de problemas. Os chamados dessa escola são direcionados por meio de mensagens de WhatsApp ou ligações telefônicas para o técnico responsável. Este procedimento dificulta a organização dos chamados e, consequentemente, atrasa as respostas para a solução deles. Uma ferramenta de controle dos chamados também impacta na gestão da qualidade de entregas feitas pelo atendimento, o que possibilitará formas para a empresa buscar uma melhoria contínua em seus processos.

Este trabalho tem como objetivo descrever e demonstrar o desenvolvimento de um aplicativo de gerenciamento dos chamados para a equipe de tecnologia da informação com objetivo aprimorar a eficiência, a comunicação e a responsabilidade no tratamento e resolução de questões técnicas relacionadas a tecnologia da informação dentro de uma escola, utilizando o framework Flutter e aplicando conceitos de engenharia de software. A estrutura do trabalho está dividida em apresentar o problema em questão, realizar o levantamento de requisitos e, com base neles, utilizar o framework Flutter para desenvolver a aplicação.

O presente trabalho está dividido em 5 seções. Na seção a seguir é apresentado o referencial teórico no qual foi feito um levantamento de informações para apoiar o

desenvolvimento dessa pesquisa. Logo após vem a seção de materiais e métodos, em que constam as informações de como foi feito a escolha do material e como ele foi analisado. A seguir, na seção de resultados são apresentadas as informações recolhidas por meio da metodologia e mostra se os objetivos foram alcançados, na sequência, são feitas as considerações finais, apresentando as dificuldades e encaminhamentos futuros e, por fim, as referências bibliográficas na qual os autores se basearam para a realização deste trabalho.

2. REFERENCIAL TEÓRICO

Aplicativos de gerenciamento de chamados são ferramentas essenciais para o gerenciamento eficaz de solicitações de suporte técnico ou atendimento ao cliente. Neste referencial teórico, serão discutidos alguns conceitos-chave relacionados a esses aplicativos, incluindo suas características, benefícios e uso em diferentes setores.

Os chamados são solicitações feitas por usuário para resolver problemas que eles encontram em seu dia de trabalho (SCHULTZ, 2019). A falta de organização na abertura de chamados pode levar uma perda de produtividade e, conseqüentemente, aumentar os custos para resolver problemas. Gerenciar os chamados dos usuários pode ser complicado, por isso, automatizar tarefas operacionais é essencial para evitar que informações se percam, ou que os serviços não sejam realizados.

Os aplicativos de gerenciamento de chamados ou *tickets* fornecem para as empresas soluções de suporte técnico ou atendimento ao cliente de maneira eficiente e organizada. Segundo Ignaczuk (2020), um sistema de tickets representa a capacidade de converter as solicitações do atendimento ao cliente em um formato numérico dentro da aplicação, permitindo, assim, o rastreamento de todas as interações e o acompanhamento do cliente até a conclusão do chamado.

Os aplicativos de chamados podem ser enquadrados como um serviço de tecnologia da informação. De acordo com Pinheiro e Magalhães (2007, p.46),

O gerenciamento de serviços de tecnologia da informação visa alocar adequadamente os recursos disponíveis e gerenciá-los de forma integrada evitando-se a ocorrência de problemas na entrega e na operação dos serviços de Tecnologia da Informação.

O presente trabalho utiliza algumas técnicas de engenharia de software para o desenvolvimento da aplicação, visando um processo de desenvolvimento mais objetivo. Existem diversas aplicações que fazem o gerenciamento de chamados como a Milvus, Movidesk, Zendesk, entre outras empresas, normalmente essas empresas são usadas por grandes corporações.

3. MATERIAIS E MÉTODOS

O trabalho conta com uma revisão bibliográfica de textos da área dos estudos de desenvolvimento de aplicações no framework Flutter em busca de conceitos e definições para elucidar e teorizar o debate aqui proposto e o desenvolvimento prático de um aplicativo utilizando o framework citado. Dentro da engenharia de software foram utilizados os conceitos de análise de requisitos funcionais e não funcionais e o diagrama de caso de uso. Esses conceitos são utilizados para definir a viabilidade e delimitar a arquitetura do projeto. Para produzir a aplicação foram utilizadas algumas ferramentas que auxiliam o desenvolvimento de aplicações e nesta seção iremos discutir as tecnologias que foram utilizadas.

Na modelagem e documentação do sistema da aplicação foi utilizada a UML (Unified Modeling Language – linguagem de modelagem unificada) “contém uma notação robusta para a modelagem e o desenvolvimento de sistemas orientados a objetos e tornou-se um padrão da indústria para o desenvolvimento de software de todos os tipos” (PRESSMAN, 2021) e o programa que foi utilizado para fazer esses modelos foi o Astah UML que é um conjunto de ferramentas que fornece uma solução para desenvolvedores individuais desenvolverem seus modelos (ASTAH, 2023). O Astah UML foi utilizado para elaborar os diagramas de classes, diagramas de caso de uso e o diagrama de atividades presentes neste artigo.

3.1. Diagrama de Caso de Uso

Um diagrama de caso de uso é uma ferramenta para ajudar a visualizar e entender a funcionalidade de um sistema de software. “Os diagramas de caso de uso ajudam a determinar a funcionalidade e as características do software sob o ponto de vista do usuário” (PRESSMAN, 2021). O Diagrama de Casos de Uso do Projeto aqui apresentado está mostrado na Figura 1.

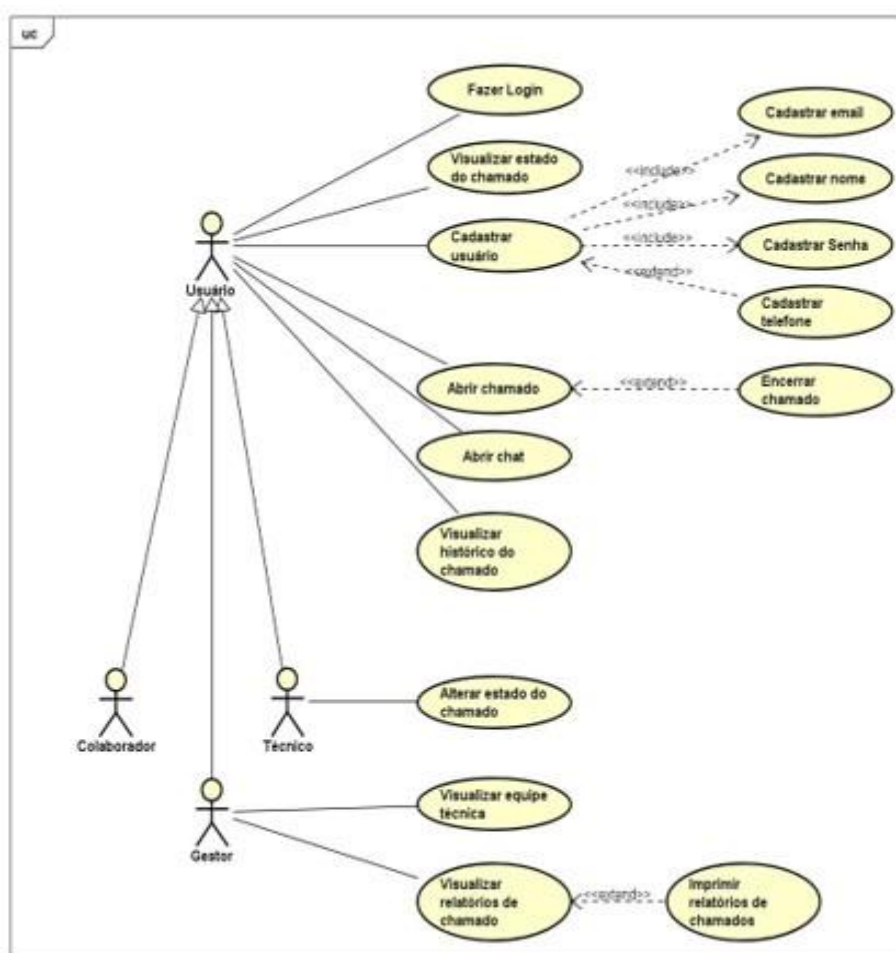


Figura 1 Diagrama de caso de uso

Fonte: (Autoria própria, 2023)

3.2. Diagrama de classe

O diagrama de classe é uma ferramenta visual importante para modelar e documentar a estrutura de um sistema orientado a objetos. O diagrama de classe “fornece uma visão estática ou estrutural do sistema. Ele não mostra a natureza dinâmica das comunicações

entre os objetos das classes no diagrama” (PRESSMAN, 2021). O diagrama de classe desse projeto está demonstrado na Figura 2.

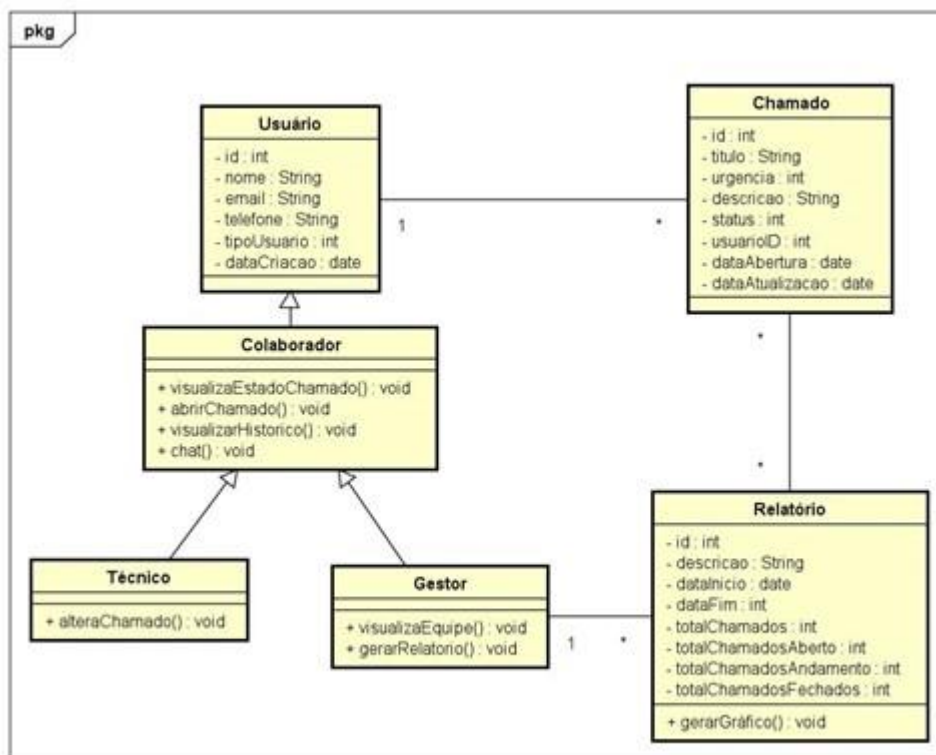


Figura 2 Diagrama de classe
Fonte: (Autoria própria, 2023)

3.3. Diagrama de atividade

“O diagrama de atividade mostra o comportamento dinâmico de um sistema ou parte de um sistema por meio do fluxo de controle entre ações que o sistema executa” (PRESSMAN, 2021). O diagrama de atividade da aplicação está descrito na Figura 3.

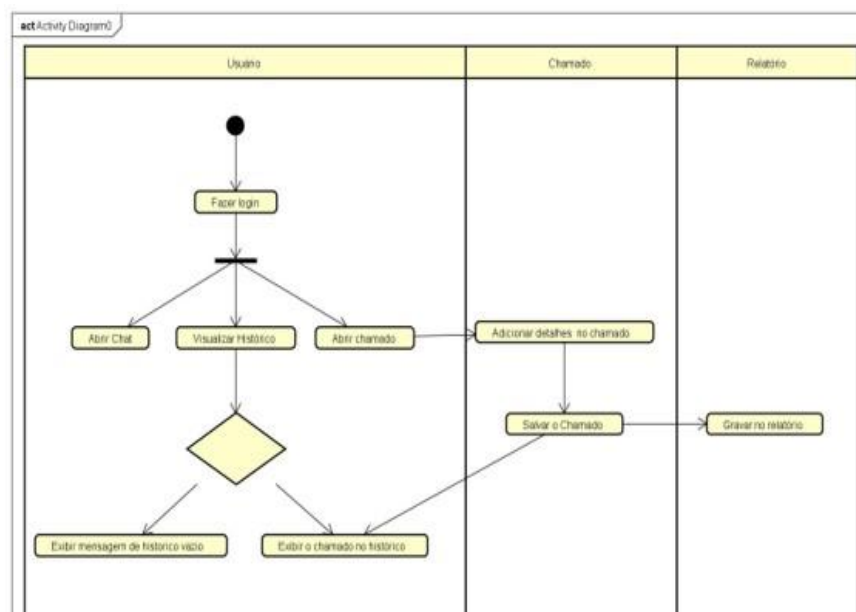


Figura 3 Diagrama de classe
Fonte: (Autoria própria, 2023)

3.4. Prototipação

De acordo com Sommerville (2011) um protótipo é uma versão inicial de um sistema de software, usado para demonstrar conceitos, experimentar opções de projeto e descobrir mais sobre o problema e suas possíveis soluções. Para a prototipação foi utilizada a ferramenta Figma (FIGMA, 2023). Os protótipos das telas para essa aplicação estão demonstrados na Figura 4.

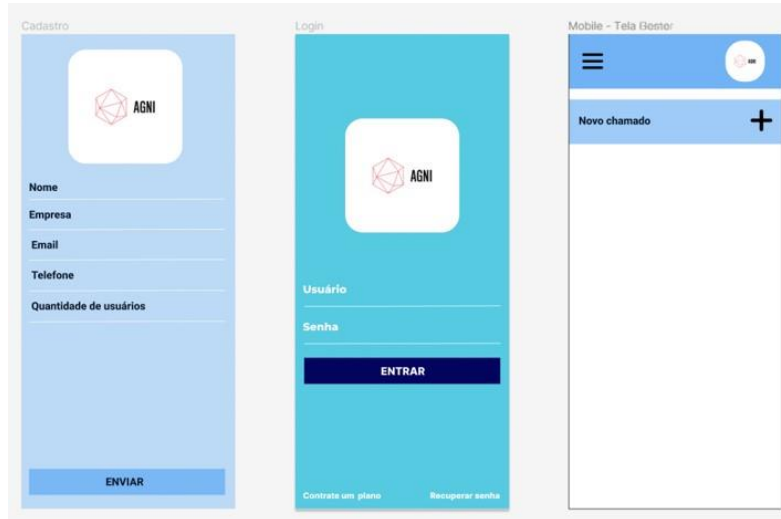


Figura 4 Prototipação
Fonte: (Autoria própria, 2023)

3.5. Framework Flutter

O framework usado para desenvolver o aplicativo foi o Flutter que é o kit de ferramentas de interface do usuário portátil do Google para criar aplicativos e compilados nativamente para dispositivos móveis, Web e desktop a partir de uma única base de código (FLUTTER, 2023). O Flutter usa como linguagem de programação a Dart que é uma linguagem de programação de alto nível, orientada a objetos, desenvolvida pela Google em 2011.

3.6. Firebase Realtime Database

Para o armazenamento de dados para o esse projeto foi utilizado o banco de dados não relacional. O Firebase Realtime Database é um banco de dados NoSQL hospedado na nuvem. Os dados são armazenados como JSON e sincronizados em tempo real para cada cliente conectado. (FIREBASE, 2023). As coleções relacionadas a esse projeto são as coleções chamado, usuário e relatório. Como demonstrados nas Tabelas 1, 2 e 3.

Tabela 1. Coleção chamado

| Coleção | Chamado | |
|------------------|----------------|---|
| Campo | Tipos de Dados | Descrição |
| id | int | identificador único do chamado |
| título | String | Título do chamado |
| descricao | String | Descrição do chamado |
| status | Int | Status atual do chamado(aberto, em andamento e concluído) |
| data_abertura | Timestamp | Data de abertura do chamado |
| data_atualizacao | Timestamp | Data da última atualização do chamado |
| urgencia | Int | Define a prioridade do chamado |

| | | |
|---------------|--------|--|
| usuarioID | String | Define o usuário que criou o chamado |
| responsavelID | String | Define o técnico que responde ao chamado |

Fonte: (Autoria própria,2023)

Tabela 2. Coleção Usuário

| Coleção | Usuário | |
|--------------|----------------|---|
| Campo | Tipos de Dados | Descrição |
| Id | int | identificador único do usuário |
| nome | String | Nome completo do usuário |
| Email | String | Endereço de email do usuário |
| Senha | String | Senha do usuário |
| data_criacao | timestamp | Data e hora da criação do usuário |
| tipo_usuario | Int | Define o tipo do usuário (colaborador, técnico ou gestor) |

Fonte: (Autoria própria, 2023)

Tabela 3. Coleção Relatório

| Coleção | Relatório | |
|-----------------------------|----------------|--|
| Campo | Tipos de Dados | Descrição |
| Id | int | identificador único do relatório |
| Tipo | int | Tipo do relatório (diário, semanal ou mensal) |
| data_inicio | timestamp | Data e hora do início do período de tempo abrangido pelo relatório |
| data_fim | timestamp | Data e hora de fim do período abrangido pelo relatório |
| total_chamados | int | número total de chamados registrados durante o período do relatório |
| total_chamados_em_aberto | int | número total de chamados em aberto registrados durante o período do relatório |
| total_chamados_em_andamento | int | número total de chamados em andamento registrados durante o período do relatório |
| total_chamados_fechados | int | número total de chamados fechados registrados durante o período do relatório |

Fonte: (Autoria própria, 2023)

4. RESULTADOS

O levantamento de requisitos e as interfaces de interação com o usuário foram concluídos juntamente com a modelagem do Banco de Dados. Utilizamos o padrão *MVC* para a arquitetura do projeto a escolha do padrão *MVC* na arquitetura do projeto nos permite separar as responsabilidades entre o *Model*, *View* e *Controller*, promovendo a modularidade, flexibilidade e manutenibilidade do sistema. Essa abordagem facilita o desenvolvimento colaborativo e promove a reutilização de código, resultando em uma aplicação bem estruturada e de qualidade (FREEMAN, 2007).

A tela inicial da aplicação vide a Figura 5a apresenta dois campos de inserção de dados, um para o nome do usuário e outro para a senha. Além disso, possui três botões: "Entrar", "Cadastre-se" e "Recuperar senha". Ao clicar no botão "Cadastre-se", o usuário é direcionado para a tela de cadastro (Figura 5b), na qual pode preencher quatro campos

de dados: nome, telefone, e-mail e senha. Após preencher esses campos, o usuário pode clicar no botão "Cadastrar" para salvar as informações. Se a ação for bem-sucedida, é exibido um resumo das informações inseridas como mostrado na Figura 6a.

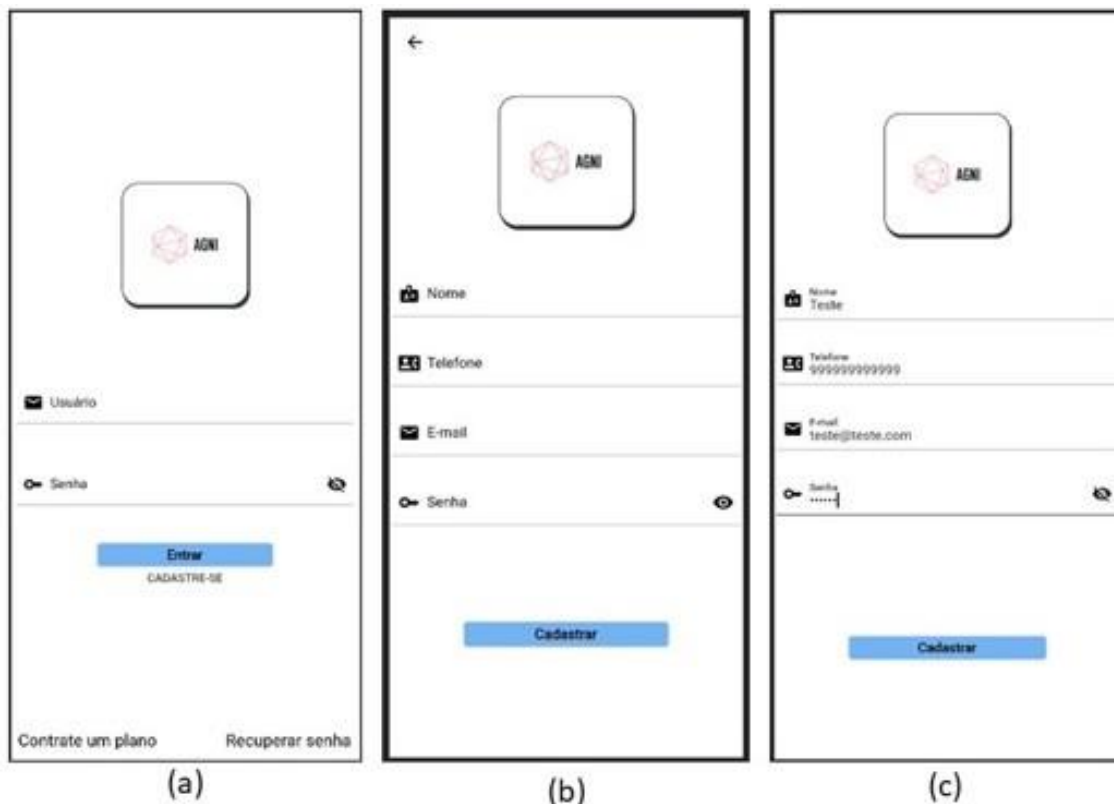


Figura 5 Telas do aplicativo
Fonte: (Autoria própria, 2023)

Ao clicar no botão "Recuperar senha", o usuário deve informar o e-mail cadastrado para receber o protocolo de recuperação de senha igual ao demonstrado na Figura 6b.

Ao acessar a aplicação, o usuário é direcionado para a tela de Home Figura 6c, que exibe um menu no canto superior esquerdo com informações do usuário, como a foto e o e-mail. Além disso, a tela possui quatro botões: "Home", "Histórico", "Sobre" e "Sair" como mostra a Figura 7a. O botão "Home" retorna o usuário para a tela inicial da aplicação. O botão "Histórico" lista todos os chamados feitos pelo usuário Figura 8a. O botão "Sobre" exibe informações sobre a aplicação e links para acessar informações sobre os desenvolvedores Figura 8b. Já o botão "Sair" desconecta o usuário da aplicação.

Ao clicar no botão "Novo Chamado", presente na tela de Home, o usuário é direcionado para uma nova tela com três campos de inserção de dados: título, urgência e descrição Figura 7b. Ao preencher esses campos, o usuário pode clicar no botão "Enviar" para enviar o chamado.

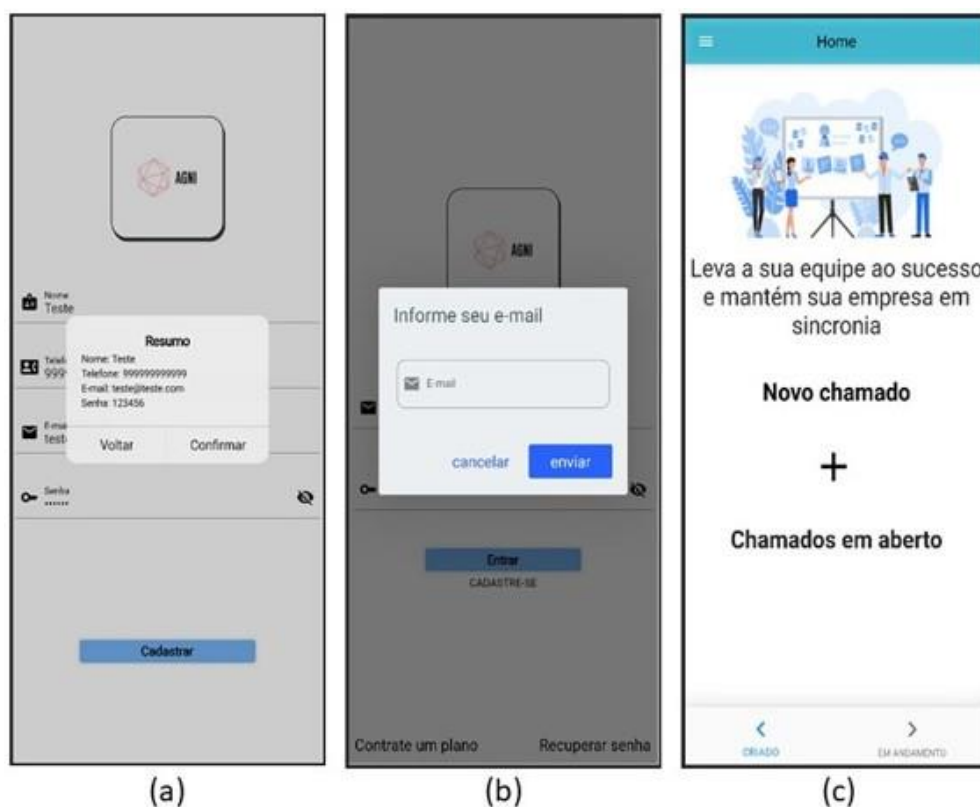


Figura 6 Telas do aplicativo
Fonte: (Autoria própria, 2023)

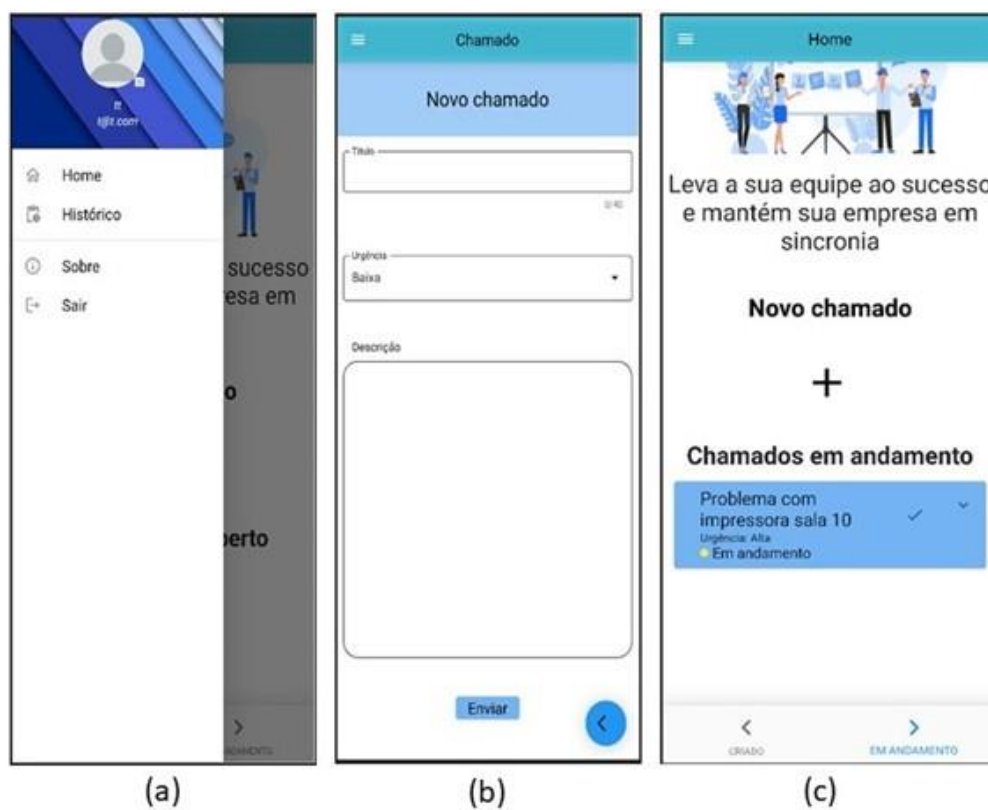


Figura 7 Telas do aplicativo
Fonte: (Autoria própria, 2023)

Ainda na tela de Home, existem dois botões na parte inferior da aplicação: "Criados" e "Em andamento". Ao clicar no botão "Criados", é exibida uma lista com todos os chamados criados pelo usuário demonstrado na Figura 7c. Já ao clicar no botão "Em andamento", é exibida uma lista com todos os chamados em andamento feitos pelo usuário.

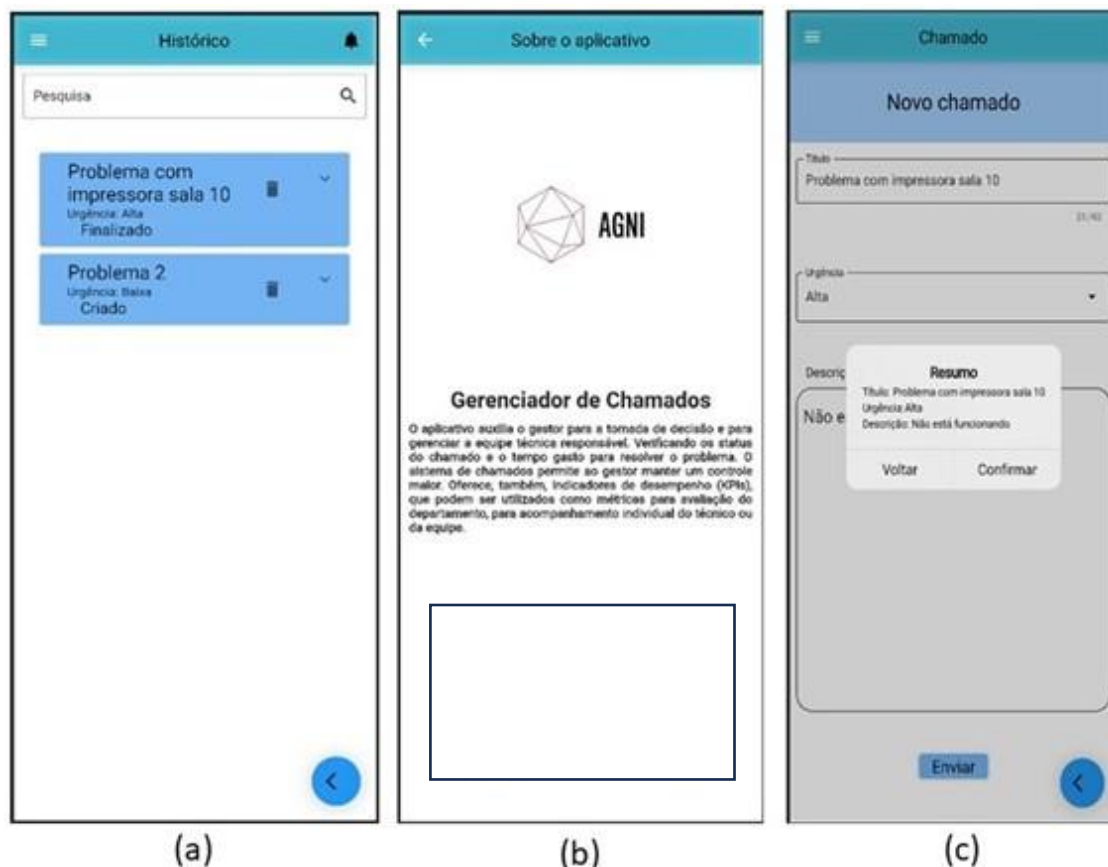


Figura 8 Telas do aplicativo
Fonte: (Autoria própria, 2023)

5. CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo apresentar o processo de desenvolvimento de um aplicativo para gerenciamento de chamados para a equipe de Tecnologia da Informação de uma escola. A construção do projeto foi iniciada com a elicitação de requisitos, em seguida pela modelagem do software e, por fim, a prototipação do aplicativo. O aplicativo busca melhorar a organização e eficiência no atendimento ao cliente, resolvendo problemas de abertura de chamados desorganizada. Foi utilizado Firebase Realtime Database para armazenar os dados e foi adotado o padrão MVC para a arquitetura do projeto. Protótipos foram criados no Figma para validar as interfaces de usuário. Durante a execução do trabalho, constatou-se a dificuldade de encontrar material de referência relacionado a chamados ou tickets. Essa situação destacou um desafio significativo no decorrer do processo de pesquisa e desenvolvimento do projeto em questão.

Os resultados demonstraram o desenvolvimento de um aplicativo completo para gerenciamento de chamados, o qual permite controlar eficientemente as demandas da área de Tecnologia da Informação. O framework Flutter, juntamente com a linguagem de programação Dart, está sendo aplicado para transformação dos protótipos em um

aplicativo operacional. Os próximos passos do projeto é validar o funcionamento do aplicativo junto a uma escola, permitindo identificar possíveis problemas e adicionar novas funcionalidades a partir das percepções dos usuários.

REFERÊNCIAS

- ASTAH. Astah Products. 2023. Disponível em: <https://astah.net/products/>. Acesso em: 26 mar. 2023.
- FREEMAN, Eric; FREEMAN, Elisabeth. Use a Cabeça!: Padrões de Projetos. 2. ed. Rio de Janeiro: Editora Alta Books, 2007.
- FIREBASE. Documentação do Firebase. Disponível em: <https://firebase.google.com/docs/database?hl=pt-br>. Acesso em: 20 mai. 2023.
- FLUTTER. Flutter FAQ. Disponível em: <https://docs.flutter.dev/resources/faq>. Acesso em: 26 mar. 2023.
- IGNACZUK, Carolina. Vantagens, funcionalidades e como escolher um sistema de tickets de suporte. 2020 . Disponível em: <https://conteudo.movidesk.com/sistema-de-tickets-suporte/#dos-beneficios-aos-recursos-tudo-o-que-voce-queria-saber-sobre-sistemas-de-atendimento>. Acesso em: 20 mai. 2023.
- PINHEIRO, Walfrido Brito; MAGALHÃES, Ivan Luizio. Gerenciamento de serviços de TI na prática: uma abordagem com base na ITIL. São Paulo: Novatec, 2007.
- PRESSMAN, Roger S; MAXIM, Bruce R. Engenharia de Software: Uma Abordagem Profissional. 9. ed. Porto Alegre: AMGH, 2021.
- SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo: Pearson Education do Brasil, 2011.
- SCHULTZ, Felix. Sistema de chamados: como funciona e porque você deve contratar. 2019. Disponível em: <https://blog.milvus.com.br/sistema-de-chamados/>. Acesso em: 20 mai. 2023.