

MEDICINETIME: PROTOTIPAÇÃO DE UM APLICATIVO MULTIPLATAFORMA PARA GERENCIAMENTO DE MEDICAMENTO

Leandro Gabriel dos Santos Silverio¹, Prof. Fabricio Gustavo Henrique¹

¹Faculdade de Tecnologia de FATEC Ribeirão Preto (FATEC)

Ribeirão Preto, SP – Brasil

leandro.silverio@fatec.sp.gov.br, fabricio.henrique@fatec.sp.gov.br

Resumo. A gestão eficaz de tratamentos medicamentosos, especialmente entre idosos, é crucial para garantir a segurança e eficácia dos cuidados de saúde. Este artigo aborda a prototipação de um aplicativo destinado a simplificar o controle de medicamentos, visando melhorar a adesão e administração correta dos mesmos. Explora-se a relação entre tecnologia e terceira idade, destacando desafios de inclusão digital e oportunidades de melhoria na qualidade de vida. O aplicativo, chamado MedicineTime, oferece funcionalidades como cadastro de medicamentos, lembretes personalizados e alertas sobre interações medicamentosas. Desenvolvido em Flutter e integrado ao Firebase, o MedicineTime busca não apenas facilitar a gestão de tratamentos, mas também promover uma administração segura e eficaz de medicamentos, contribuindo assim para a saúde e bem-estar dos usuários, com potencial impacto positivo na qualidade de vida da população idosa. O código-fonte e os documentos estão disponíveis em: <https://github.com/legabriel306/tcc-medicinetime>.

Abstract. Effective management of medication treatments, especially among the elderly, is crucial to ensure the safety and efficacy of healthcare. This article addresses the prototype of an application aimed to simplify medication control, improve adherence and correct administration. The relationship between technology and the elderly is explored, highlighting challenges of digital inclusion and opportunities for improving life quality. The application, called MedicineTime, offers features such as medication registration, personalized reminders, and alerts about drug interactions. The app was developed in Flutter and integrated with Firebase, MedicineTime seeks not only to facilitate treatment management but also to promote safe and effective medication administration, thus contributing to users' health and well-being, with the potential to impact positively the elderly life quality. The source code and the documents are available at: <https://github.com/legabriel306/tcc-medicinetime>.

1. Introdução

A evolução da medicina está ligada ao avanço da tecnologia e às constantes descobertas de novos tratamentos para diversas doenças. Com essas descobertas, surgem novas formulações farmacêuticas destinadas a combater os agentes causadores das enfermidades. Para garantir a eficácia e os resultados desejados, é crucial seguir rigorosamente os tratamentos com administração de medicamentos.

O uso inadequado de qualquer medicamento não apenas falha em proporcionar os efeitos desejados ao paciente, mas também pode resultar em sérios danos à saúde. Conforme a Secretaria de Estado da Saúde (SES, 2024) destaca, especialmente em

tratamentos com antibióticos, é imperativo seguir o horário de administração e a duração determinados pelo médico, sob o risco de levar o paciente a infecções generalizadas graves.

A gestão eficaz da ingestão de medicamentos é uma preocupação vital, sobretudo para a população idosa, que muitas vezes enfrenta desafios na manutenção de um regime de medicamentos consistente.

Nesse contexto, este artigo propõe explorar a prototipagem de um aplicativo projetado para atender às necessidades dos idosos, simplificando o controle de seus tratamentos medicamentosos. Por meio de um sistema de lembretes e monitoramento personalizado, este aplicativo visa oferecer uma solução prática e confiável para assegurar que os usuários tomem os medicamentos corretos no momento adequado, melhorando assim a qualidade de vida e o bem-estar daqueles que dependem de tratamentos médicos. Neste artigo, discuti-se em detalhes as funcionalidades, os benefícios e o potencial impacto desse aplicativo na gestão da saúde dos idosos. Para tal, o artigo está organizado da seguinte forma: na Seção 2 são comentadas a fundamentação teórica do trabalho; na Seção 3 são apresentados os materiais e métodos utilizados na pesquisa; na Seção 4 são discutidos os resultados obtidos a partir da análise dos dados; na Seção 5 são apresentadas as conclusões do estudo, juntamente com sugestões para trabalhos futuros.

2. Referencial Teórico

Esta seção oferece a fundamentação teórica para o desenvolvimento deste trabalho, explorando a relação entre tecnologia e a terceira idade.

2.1. Tecnologia e terceira idade

Os idosos vivenciaram a fase de transformação desta sociedade virtual. A população idosa está em constante crescimento, tornando-se um público ativo com interesses pelos artefatos tecnológicos ligados à internet. Entretanto, os idosos não foram socializados com a internet, visto que, a internet está presente na sociedade há poucas décadas (ROCHA, 2018).

Para proporcionar qualidade de vida a esse grupo de indivíduos, deve-se pensar em formas de oferecer a eles a possibilidade de trabalhar com a tecnologia, fazendo uso da mesma para aprendizagem contínua, por exemplo. Dessa forma, eles poderão se manter atualizados e, até mesmo, atuar no mercado de trabalho. Dentro deste contexto, a tecnologia pode desempenhar um papel importante no apoio aos idosos, ao permitir que eles levem uma vida independente, com acesso a inúmeros recursos, tais como: acesso a banco eletrônico e contato com amigos e familiares por meio de redes sociais digitais. Entretanto, a tecnologia também pode ser uma barreira na vida dos idosos que não possuem familiaridade com mesma, nem condições de se manter atualizado com relação às frequentes inovações (MOL, 2011).

A tecnologia pode melhorar a vida dos idosos, mas também traz desafios como a acessibilidade e a resistência à mudança. O envolvimento digital é essencial para aumentar sua participação na vida social e facilitar as tarefas cotidianas, como o gerenciamento de seus medicamentos. A dosagem correta e a administração adequada são essenciais para a saúde dos idosos. Embora a tecnologia ofereça oportunidades de aprendizagem e independência, é necessário enfrentar os desafios da inclusão digital e capacitar os idosos para utilizarem a tecnologia digital da melhor forma possível (MOL, 2011).

2.2. Definição de interações medicamentosas

Interações medicamentosas são tipos especiais de respostas farmacológicas, em que os efeitos de um ou mais medicamentos são alterados pela administração simultânea ou anterior de outros, ou através da administração concorrente com alimentos (SECOLI, 2008).

As interações medicamentosas representam uma área complexa e multifacetada da farmacologia clínica, com importantes implicações para a prática médica. Compreender os mecanismos subjacentes e os tipos de interações, juntamente com uma avaliação criteriosa dos riscos e benefícios, é fundamental para fornecer cuidados de saúde seguros e eficazes aos pacientes. A conscientização sobre as interações medicamentosas e a adoção de medidas preventivas são essenciais para otimizar os resultados terapêuticos e minimizar os riscos associados ao uso de medicamentos (SECOLI, 2008).

As interações medicamentosas podem ser classificadas de diversas maneiras, incluindo sua natureza (farmacocinética ou farmacodinâmica), mecanismo de ação (inibição ou indução de enzimas metabólicas), gravidade (de leves a graves) e previsibilidade (previsíveis ou imprevisíveis). Além disso, as interações podem ocorrer entre medicamentos prescritos, medicamentos de venda livre, fitoterápicos e até mesmo substâncias presentes na dieta, como alimentos, bebidas alcoólicas, leite, entre outras (SECOLI, 2008).

2.3. Inclusão e acessibilidade

A inclusão e acessibilidade são elementos centrais no design de dispositivos e aplicativos. Um aplicativo inclusivo coloca as pessoas em primeiro lugar, priorizando uma comunicação respeitosa e apresentando conteúdo e funcionalidades de maneiras que todos possam acessar e compreender (APPLE, 2024b). A empatia é fundamental para entender como diferentes perspectivas podem reagir ao conteúdo. Um design inclusivo considera diversas características humanas, como idade, gênero, etnia, habilidades físicas e cognitivas, entre outras.

A acessibilidade vai além de tornar as informações disponíveis apenas para pessoas com deficiências, trata-se de garantir que as informações estejam acessíveis a todos, independentemente de suas capacidades ou situações (APPLE, 2024a). Essa declaração enfatiza a importância de projetar e desenvolver ambientes digitais, produtos e serviços de forma inclusiva, para que possam ser usados por uma variedade de pessoas, independentemente de suas habilidades físicas, cognitivas ou sensoriais. Acesso universal é essencial para promover a igualdade e garantir que todos tenham acesso aos recursos essenciais em um mundo cada vez mais centrado na tecnologia e na informação.

3. Materiais e Métodos

No âmbito do desenvolvimento de aplicações móveis, é importante contar com ferramentas e tecnologias que proporcionem eficiência, desempenho e flexibilidade. Nesse contexto, o presente trabalho adotou um conjunto de ferramentas, cada uma desempenhando um papel importante no processo de desenvolvimento, e elas estão descritas nessa seção.

3.1. Flutter

Flutter é um *framework* que possibilita a criação de aplicativos nativos para dispositivos

Android e iOS com a linguagem de programação Dart. O *framework* compila código Advanced RISC Machine (ARM) nativo para as duas plataformas, entregando um aplicativo com alto desempenho. A tecnologia utiliza o motor de renderização Skia 2D, uma ferramenta extremamente versátil que trabalha com diversas plataformas (*hardwares* e *softwares*). Além de ser usado nos produtos da empresa Google, é também usado no Mozilla FireFox, FireFoxOS entre outros. Ele também é capaz de compilar as telas em tempo real (*hot reload*), agilizando o desenvolvimento de aplicativos (NAPOLI, 2019).

O Flutter oferece uma série de vantagens notáveis para o desenvolvimento de aplicativos nativos para Android e iOS (FLUTTER, 2024). Além de seus excelentes recursos nesse aspecto, o Flutter possui uma variedade de outras atrações que o diferenciam no cenário de desenvolvimento móvel. Um desses pontos fortes é a velocidade ágil de tradução que se destaca das demais tecnologias disponíveis. Isso resulta em um processo de desenvolvimento mais eficiente e eficaz.

Outra característica é o foco na reutilização de código. Isto é possível graças ao conceito de *widgets*. Isso permite que a interface seja decomposta em seus componentes (FLUTTER, 2024). Essa abordagem facilita a criação de elementos como botões, listagens, telas e animações que são partes essenciais da experiência do usuário.

Os *widgets*, que constituem a base desta abordagem, são representações de componentes individuais como botões de animações complexos. É implementado como uma classe em Dart, uma linguagem que segue o paradigma de programação orientada a objetos (DART, 2023). Esses *widgets* têm a capacitância única de descrever sua própria exibição.

No contexto do Flutter, existem dois tipos principais de *widgets*: *statefull* e *stateless*. *Widgets* sem estado ou *stateless* não precisam persistir ou alterar seu estado durante todo o seu ciclo de vida. Ele recebe todos os dados necessários quando é criado e depende dos componentes que o utilizam para se renderizar (FLUTTER, 2024). Por exemplo, um botão com texto estático será excluído automaticamente quando a tela onde ele se encontra for excluída.

Em contraste, os *statefull widgets* suportam um estado mutável e independente. Isso permite que os *widgets* armazenem e gerenciem dados. Este estado pode ser alterado usando eventos cronometrados, como ações do usuário ou respostas externas da API (FLUTTER, 2024).

A composição da UI (interface de usuário) do Flutter segue a lógica de uma árvore de *widgets* onde cada elemento possui as informações necessários para renderizar a tela. O processo de renderização ocorre de forma sequencial, percorrendo a árvore de *widgets* da raiz até as folhas. Nesse processo, o Flutter coleta as restrições de tamanho e posição de cada elemento, determinadas pelos *widgets* pais. Em seguida, é realizado um processo reverso para definir os valores reais de renderização, como cores, tamanhos e posições (FLUTTER, 2024).

3.2. Dart

A linguagem Dart é um projeto de código aberto criado pelo Google. Conforme destacado por Walrath e Ladd (2012), sua origem está na necessidade de simplificar o desenvolvimento de aplicações web complexas, proporcionando velocidade no carregamento e na inicialização, com o intuito de garantir uma experiência de usuário superior. Dart, desenvolvido com foco na criação de aplicações web, oferece uma série de recursos que incluem programação assíncrona, manipulação de dados, codificação e

decodificação de informações, protocolos de comunicação na internet, programação concorrente e uma ampla variedade de bibliotecas essenciais para o desenvolvimento (DART, 2023).

Dart é uma linguagem otimizada para cliente para desenvolver aplicativos rápidos em qualquer plataforma. Seu objetivo é oferecer a linguagem de programação mais produtiva para desenvolvimento multiplataforma, combinada com uma plataforma de execução flexível para estruturas de aplicativos (DART, 2023).

3.3. Firebase

O FireBase foi escolhido pela sua integração e compatibilidade com Flutter, que facilita o processo de desenvolvimento, e possui diversas ferramentas para agilizar a codificação. É uma plataforma de desenvolvimento de app que ajuda na construção e crescimento de apps. Apoiado pela Google e confiado por milhões de negócios ao redor do mundo (FIREBASE, 2024).

Além disso, é uma plataforma simples de usar e possui diversos serviços, como o Firebase Authentication. O objetivo do Firebase Authentication é facilitar o desenvolvimento de um sistema de autenticação seguro, além de melhorar a experiência de *login* e integração para os usuários finais. Ele oferece uma solução de identidade completa, compatível com contas de e-mail/senha, autenticação por telefone, *login* do Google, Twitter, Facebook, GitHub e outros (FIREBASE, 2024).

3.4. Engenharia de requisitos

A engenharia de requisitos é pautada acerca da prototipação gráfica, diagramação do software e análise esmiuçada de detalhes técnicos. Esta é uma parte importantíssima de um projeto que busca ter alta qualidade em seu produto final pois é a partir dos requisitos levantados que podem ser definidas as capacidades e limitações presentes em um software (WIEGERS, 2022).

Dividindo-se em quatro etapas distintas, a engenharia de requisitos começa com o Levantamento de Requisitos, no qual entrevistas com usuários, clientes e *stakeholders* são conduzidas. Este estágio visa garantir que o *software* atenda de maneira integral às necessidades reais do que está sendo desenvolvido. A análise de requisitos sucede o levantamento, em que as informações são minuciosamente examinadas e validadas, dando continuidade apenas às que são verdadeiramente relevantes para o *software* (WIEGERS, 2022).

A terceira etapa, Especificação de Requisitos, envolve a criação da documentação que compreende os requisitos que passaram pela análise. Por fim, a Verificação e Validação de Requisitos engloba a implementação dos requisitos especificados, seguida por testes para assegurar que estão sendo cumpridos conforme estimado (WIEGERS, 2022).

3.5. UML

A Linguagem de Modelagem Unificada (UML) da OMG (*Object Management Group*) ajuda a especificar, visualizar e documentar modelos de sistemas de software, incluindo sua estrutura e design, de uma forma que atende a todos esses requisitos (UML, 2015), desempenha um papel crucial ao estabelecer uma linguagem padrão para a diagramação de *software*. Esta linguagem versátil possibilita a criação de modelagens para sistemas de qualquer complexidade. Dentre os diversos tipos de diagramas da UML,

os mais notáveis incluem o de classes, que representa a estrutura estática do sistema; casos de uso, que descreve as funcionalidades do ponto de vista do usuário; sequência, que ilustra a dinâmica de comportamento do *software*; atividade, que representa o fluxo de controle do sistema; e componentes, que exhibe a estrutura física do sistema.

Os diagramas apresentados neste artigo foram elaborados utilizando o Astah UML (2023) em sua versão gratuita para estudantes, uma ferramenta que possibilita a criação de diagramas de software.

3.6. Figma como ferramenta de prototipação

O Figma é uma ferramenta de design colaborativa e poderosa para equipes. As ferramentas de prototipagem do Figma facilitam a criação e o compartilhamento de protótipos interativos, de alta fidelidade, sem necessidade de código. Pertencente a empresa Figma, Inc., lançada em 2016 por Dylan Field e Evan Wallace (FIGMA, 2023a).

O Figma é uma plataforma versátil que permite a modelagem e edição de páginas, softwares e conteúdos para tecnologia, sendo especialmente eficaz no aprimoramento de UI e UX (experiência do usuário). Esta ferramenta se destaca por ser baseada em nuvem, o que possibilita a edição em qualquer local com acesso à internet. Além disso, oferece recursos colaborativos, facilitando projetos com múltiplos usuários, e possui uma ampla gama de funcionalidades disponíveis para o usuário. No contexto do artigo, foi utilizada a versão gratuita do Figma, que atendeu de maneira eficiente às necessidades do projeto (FIGMA, 2023b).

4. Resultados

Nesta seção serão apresentados alguns dos diagramas UML feitos a partir dos requisitos do sistema, os protótipos de alta fidelidade, assim como uma breve explicação do fluxo de uso do programa.

4.1. Elicitação de requisitos e documentação

O documento de elicitação de requisitos para o aplicativo que foi prototipado neste projeto pode ser encontrado no GITHUB do projeto: <<https://github.com/legabriel306/tcc-medicinetime/blob/main/documento%20de%20requisitos.docx>>.

A ferramenta Astah UML foi usada para desenvolver os documentos UML e a documentação do aplicativo. Os diagramas de Atividade para o Usuário, Caso de Uso e Classe foram criados para o aplicativo prototipado neste projeto. Eles estão disponíveis do **GitHub**: <https://github.com/legabriel306/tcc-medicinetime>.

4.2. Aplicativo MedicineTime

O MedicineTime é um aplicativo voltado para idosos que monitora as interações entre medicamentos. Para facilitar o controle dos tratamentos medicamentosos, a aplicação fornece alertas sobre possíveis interações entre os medicamentos prescritos. O aplicativo oferece lembretes personalizados para a administração correta dos medicamentos por meio de uma interface fácil de usar. Isso ajuda a garantir que os tratamentos sejam seguros e eficazes.

4.3. Implementação do aplicativo MedicineTime

O aplicativo contém sete telas já implementadas. O objetivo principal é a gestão de medicamentos, que inclui telas para *login* e cadastro de usuários, bem como para exibir,

adicionar, excluir e editar medicamentos. As telas foram desenvolvidas em Flutter, integradas ao Firebase. O código-fonte está disponível em: <https://github.com/legabriel306/tcc-medicinetime/tree/main/app/Medicine-Time>.

O aplicativo conta com a tela de *login* (Figura 1A) que apresenta dois campos de preenchimento. O usuário deve digitar seu endereço de e-mail previamente cadastrado no campo 'E-mail' e a senha definida no campo 'Senha'. O usuário deve clicar no botão 'Entrar' após fornecer seus dados de *login*. O aplicativo redirecionará o usuário para a tela do menu (Figura 3A) caso os dados forem válidos. A integração do Flutter com o Firebase permitiu a implementação do sistema de autenticação de *login*.

Na tela de *login*, o botão 'sobre o aplicativo' redireciona o usuário para uma tela que descreve o objetivo do aplicativo (Figura 2B), o botão 'cadastrar' redireciona o usuário para a tela de cadastro (Figura 1B), e o botão 'Recuperar senha' redireciona o usuário para a tela 'esqueceu a senha' (Figura 2A).

A tela de cadastro de usuário (Figura 1B) apresenta quatro campos para preenchimento. O usuário deve inserir seu nome no campo 'Nome', o endereço de e-mail no campo 'E-mail', a senha no campo 'Senha' e confirmar a senha novamente no campo 'Confirmar Senha'.

Além desses campos, haverá um botão 'Cadastrar' que o usuário pode clicar após preencher todos os campos necessários. Ao clicar neste botão, os dados inseridos serão enviados para o sistema para processamento. Se os dados fornecidos forem válidos e corresponderem aos critérios de validação, o usuário será registrado no sistema com sucesso.

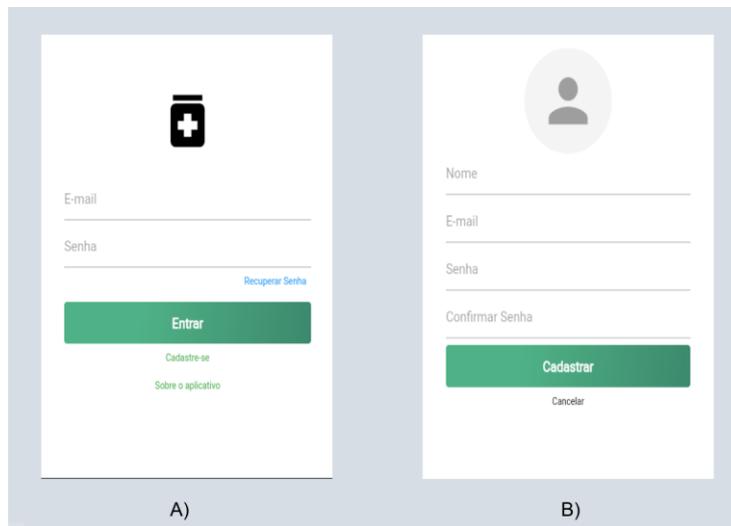


Figura 1. A) Tela de *login*; B) Tela de cadastro do usuário
Fonte: (Autoria própria, 2023)

Na tela de redefinição de senha (Figura 2A), o usuário tem a opção de redefinir sua senha. O usuário deve inserir seu endereço de e-mail registrado anteriormente no campo 'E-mail'. O usuário receberá um e-mail contendo um link para redefinir sua senha após clicar no botão 'Enviar'.

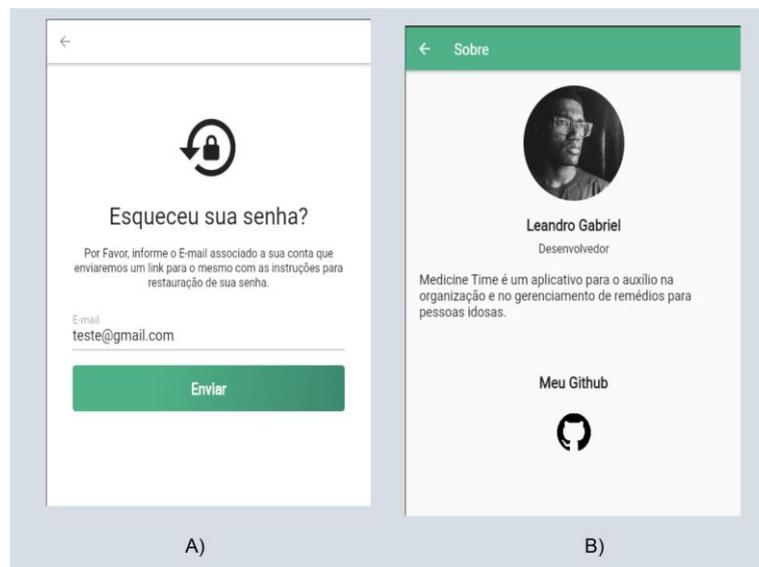


Figura 2. A) Tela de redefinir senha; B) Tela Sobre.
Fonte: (Autoria própria, 2023)

A tela principal (Figura 3A) exibe os medicamentos cadastrados pelo usuário. Aqui, é possível excluí-los ao clicar no botão com o ícone de lixeira no canto direito ou editá-los ao clicar diretamente sobre o medicamento desejado. O usuário é redirecionado para a tela de configurações (Figura 4) ao clicar no nome do usuário no canto superior direito ou no ícone ao lado. Para adicionar novos medicamentos, basta clicar no botão de adição no canto inferior direito, que o redirecionará para a tela de cadastro de medicamentos (Figura 3B).

A tela de cadastro de medicamento (Figura 3B) possui três campos para preenchimento: o campo 'Nome', onde o usuário deve inserir o nome do medicamento; o campo 'Dose', para especificar a dose por ingestão; e o campo 'Intervalo', que indica o intervalo de tempo entre cada ingestão do medicamento. Ao clicar em 'Salvar', o medicamento será salvo e aparecerá na tela principal (Figura 3A).

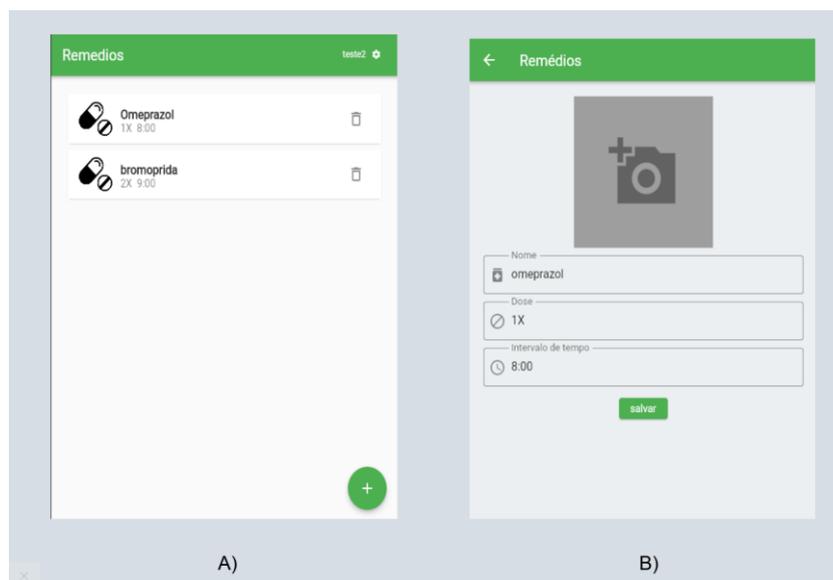


Figura 3. A) Tela principal; B) Tela de cadastro de remédios

Fonte: (Autoria própria, 2023)

A tela de configurações (Figura 4) exibe o nome, e-mail e foto do usuário, previamente cadastrados. Nesta tela, há um botão 'Alterar Nome', que permite ao usuário editar o nome cadastrado. Além disso, o botão 'Logout' permite que o usuário encerre a sessão e seja redirecionado para a tela de *login* (Figura 1A).

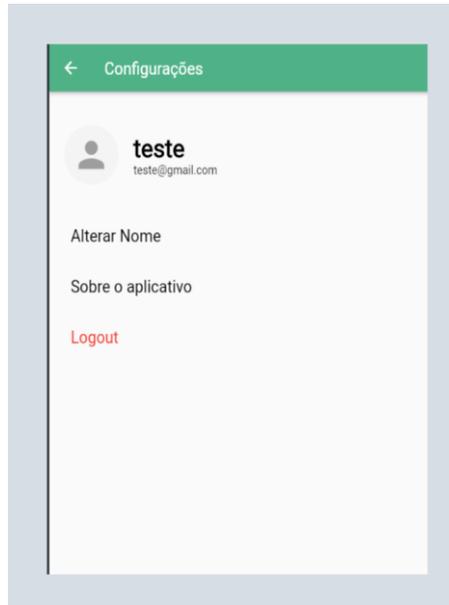


Figura 4. Tela de configurações
Fonte: (Autoria própria, 2023)

4.4. Prototipação do MedicineTime

Além das telas já implementadas e descritas no tópico anterior, foram desenhados alguns protótipos adicionais utilizando a ferramenta Figma (FIGMA, 2023a).

Uma perspectiva para uma próxima versão é a inclusão de contatos do usuário, permitindo que outras pessoas acompanhem e auxiliem no tratamento. A tela de contatos (Figura 5B) exibe os contatos previamente cadastrados. Ao clicar no botão '+' no canto inferior direito, o usuário é redirecionado para a tela de cadastro de contatos (Figura 5A), onde pode adicionar novos contatos. Para excluir um contato, basta clicar no ícone de lixeira no canto direito do respectivo contato. Além disso, é possível editar um contato clicando sobre ele.

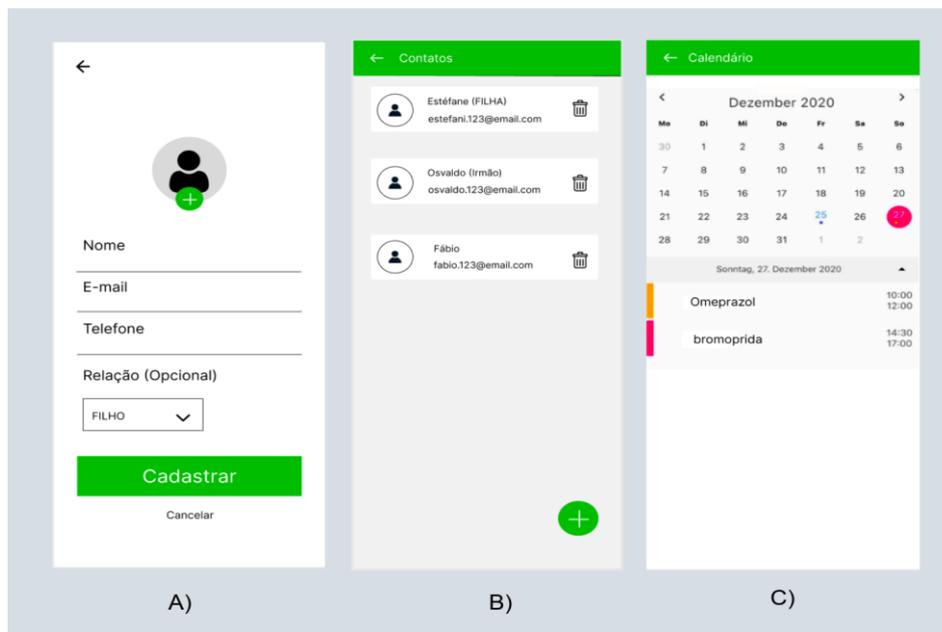


Figura 5. A) Tela de cadastro de contatos; B) Tela de contatos; C) Tela do calendário
Fonte: (Autoria própria, 2023)

A tela de cadastro de contato contém quatro campos de preenchimento: o campo ‘nome’, onde o usuário deve inserir o nome do contato; o campo ‘E-mail’ que é o preenchimento do endereço de e-mail do contato; o ‘Telefone’, onde o usuário deve inserir o número de telefone do contato; o campo “relação”, que é um campo opcional e define a relação que o usuário possui com o contato.

Ao clicar no botão 'Cadastrar' na tela, o contato será adicionado. Se todas as informações obrigatórias forem preenchidas corretamente, uma mensagem de confirmação será exibida. Caso contrário, o aplicativo exibirá uma mensagem de erro indicando o campo que precisa ser corrigido. Ao clicar no botão 'Cancelar', a operação de cadastro será cancelada e você retornará à tela de contatos sem salvar nenhuma alteração.

Com o objetivo de facilitar o gerenciamento diário, o calendário (Figura 5C) apresenta os medicamentos a serem tomados em cada dia. O usuário pode clicar em uma data específica para visualizar a lista completa de medicamentos para aquele dia, incluindo nome, e horário da próxima dose.

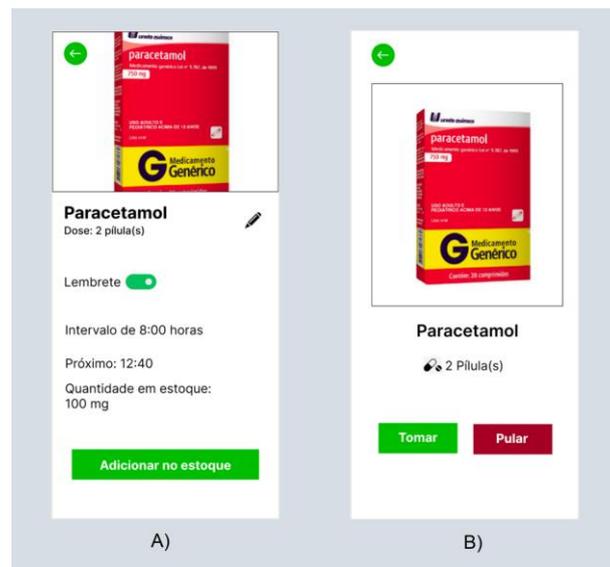


Figura 6. A) Tela de detalhes do medicamento; B) Tela de lembrete
Fonte: (Autoria própria, 2023)

Ao clicar no medicamento na tela principal, o usuário é direcionado para a tela de detalhes (Figura 6A), que apresenta todas as informações importantes sobre o medicamento cadastrado. Isso inclui a dose, o intervalo entre as doses, a quantidade restante no estoque e a hora do próximo lembrete.

Para adicionar mais unidades do medicamento ao estoque, basta clicar no botão 'Adicionar no estoque'.

O usuário também pode gerenciar os lembretes de medicação através do botão ao lado do campo 'Lembrete'. Com um simples toque, é possível ativar ou desativar os lembretes para cada medicamento.

Na tela de lembrete (Figura 6B), o usuário encontra uma interface clara e intuitiva com a imagem, o nome e a dosagem do medicamento em destaque. Para registrar a ingestão do medicamento, basta clicar no botão 'Tomar'. Caso não tenha realizado a ingestão no momento, o botão 'Pular' permite adiar o lembrete. Ambos os botões registram a ação do usuário, garantindo um acompanhamento preciso do tratamento.

5. Considerações Finais

Para garantir qualidade e atender aos requisitos, o MedicineTime foi prototipado usando processos padrões de engenharia de software. Toda a documentação do sistema está disponível no GitHub.

O MedicineTime será uma excelente ferramenta para ajudar as pessoas a administrar e organizar seus medicamentos. Funcionando como um facilitador para a consistência do tratamento, o aplicativo será especialmente útil para os idosos, auxiliando-os na administração correta e no controle eficaz de seus medicamentos. Ao proporcionar lembretes, alertas de interações medicamentosas e acompanhamento detalhado, MedicineTime contribuirá para a adesão ao regime de tratamento prescrito, reduzindo o risco de esquecimentos e erros na dosagem. Como resultado, o aplicativo não apenas melhora a qualidade de vida dos usuários, mas também ajuda a administrar medicamentos de maneira mais segura e eficaz.

O uso de Flutter com Firebase foi uma escolha adequada a projeto desenvolvido, pois o *framework* atendeu todas as necessidades, incluindo um banco de dados dinâmico em tempo real. Isso permitiu funcionalidades de dados em tempo real e sincronizadas para todos os usuários. Além disso, o desenvolvimento único de aplicativos para Android, iOS e Web aumentou a acessibilidade. O Figma também foi útil na prototipação, sendo flexível, com suporte da comunidade e permitindo colaboração em tempo real de forma intuitiva. No entanto, o principal desafio encontrado é a complexidade de criar uma UI e UX adequadas para idosos. A UI/UX deve ser cuidadosamente projetada levando em consideração as necessidades específicas desse público-alvo.

A criação e implementação de outras telas e recursos que foram prototipados e apresentados na documentação são algumas das perspectivas futuras do MedicineTime. A tradução de componentes conceituais para interfaces funcionais e a integração das funcionalidades planejadas são parte disso. Após essa etapa, os testes serão iniciados, incluindo testes com usuários reais, para determinar quais mudanças e melhorias seriam necessárias.

6. Referências

- APPLE. (2024). Human Interface Guidelines: Accessibility. Disponível em: <<https://developer.apple.com/design/human-interface-guidelines/accessibility>>. Acesso em: 20 mar. 2024a.
- APPLE. (2024). Human Interface Guidelines: Inclusion. Disponível em: <<https://developer.apple.com/design/human-interface-guidelines/inclusion>>. Acesso em: 20 mar. 2024b.
- DART.DEV. Dart Overview. Disponível em: <<https://dart.dev/overview>>. Acesso em: 1 set. 2023.
- FIREBASE, F. Authentication and Cloud Firestore. Firebase, 2022. Disponível em: <<https://firebase.google.com/docs/firestore?hl=pt-br>> Acesso em: 25 Jun. 2024.
- FIGMA, Documentação do Figma (2023). Disponível em: <<https://www.figma.com/prototyping/>>. Acesso em: 30 out. 2023a..
- FIGMA, Documentação do Figma (2023). Disponível em: <<https://www.figma.com/design/>>. Acesso em: 30 out. 2023b.
- FLUTTER, F. Documentação Flutter. Flutter, 2024. Disponível em: <<https://docs.flutter.dev/>> Acesso em: 25 jun. 2024.
- MOL, M. A. Recomendações de usabilidade para interface de aplicativos para smartphones com foco na terceira idade, 2011. Disponível em: <https://bib.pucminas.br/teses/Inform%C3%A1tica_MolAM_1.pdf>. Acesso em 25 jun. 2024.
- NAPOLI, Marco L. Beginning Flutter: A Hands On Guide to App Development. Beginning Flutter: A Hands On Guide to App Development, 2019. Disponível em: <<https://books.google.com.br/books?hl=pt-BR&lr=&id=ex-tDwAAQBAJ&oi=fnd&pg=PR21&dq=NAPOLI+2019+flutter&ots=YWJv9ZccHj&sig=JQnZbDmFylAtvsqMILzFGgWvL6A#v=onepage&q&f=false>>. Acesso em: 10 ago. 2023.
- ROCHA, Manoela Ferreira. EXPERIÊNCIAS DE IDOSOS USUÁRIOS DE INTERNET, 2018.

<[https://repositorio.unisc.br/jspui/bitstream/11624/2405/1/Manoela Ferreira Rocha.pdf](https://repositorio.unisc.br/jspui/bitstream/11624/2405/1/Manoela_Ferreira_Rocha.pdf)>. Acesso em: 20 ago. 2023.

SECRETARIA DE ESTADO DA SAÚDE (SES). Secretaria da Saúde conscientiza a população sobre o uso racional de medicamentos. Disponível em: <<https://saude.se.gov.br/secretaria-da-saude-conscientiza-a-populacao-sobre-o-uso-racional-de-medicamentos/>>. Acesso em: 15 mai. 2024.

SECOLI, SILVIA REGINA. Interações medicamentosas: fundamentos para a prática clínica da enfermagem, 2008. Revista Da Escola De Enfermagem Da USP, 35(1), 28–34. <<https://doi.org/10.1590/S0080-62342001000100005>>. Acesso em: 20 nov. 2024.

UML, What is UML, 2005. Disponível em <<https://www.uml.org/what-is-uml.htm>>. Acesso em 28 de out. de 2023.

WIEGERS, Karl E.; BACH, Joy L (2022). Engenharia de Requisitos: Uma Abordagem Prática. Elsevier.