



UNIVERSIDADE NOVE DE JULHO - UNINOVE
PROJETO EM COMPUTAÇÃO APLICADA

EXCLUÍDOS OS DADOS SOBRE OS AUTORES EM ATENDIMENTO A
LGPD - LEI GERAL DE PROTEÇÃO DE DADOS

PROJETO EM COMPUTAÇÃO APLICADA

São Paulo
2024

**EXCLUÍDOS OS DADOS SOBRE OS AUTORES EM ATENDIMENTO A
LGPD - LEI GERAL DE PROTEÇÃO DE DADOS**

PROJETO EM COMPUTAÇÃO APLICADA

Projeto apresentado a Universidade Nove de
Julho - UNINOVE, como parte dos requisitos obrigatórios para obtenção do título de Bacharel em Ciencia da Computação.

Prof. Orientador: Hebert Bratefixe Alquimim

**São Paulo
2024**

RESUMO

Neste projeto, desenvolvemos uma aplicação móvel composta por um frontend em React Native e Expo Go, criando assim um aplicativo multiplataforma para Android e para IOS. O objetivo é criar um aplicativo para ajudar os usuários a gerenciar seus cronogramas de medicação de forma prática, garantindo o cumprimento dos horários de doses e reduzindo os riscos de erros na administração dos medicamentos. Com uma interface intuitiva e recursos avançados, o aplicativo oferece uma solução completa para a gestão de medicações, ajudando pacientes a seguirem seus tratamentos de maneira mais segura. A integração com familiares é outro diferencial do aplicativo, permitindo que essas partes envolvidas acompanhem o tratamento e ofereçam ajuda aos usuários. Essa conexão entre paciente e familiares contribui para uma supervisão mais eficiente e para a melhoria da adesão ao tratamento, promovendo segurança e confiança no processo. O desenvolvimento do Hémedios utiliza React Native com Expo, garantindo uma experiência de usuário fluida e interativa. A escolha dessas tecnologias possibilita a criação de um aplicativo mobile, compatível com as plataformas Android e iOS.

Palavras-chave: Expo Go, React Native, Android, IOS, Multiplataforma, Aplicação Móvel

ABSTRACT

In this project, we developed a mobile application consisting of a frontend in React Native and Expo Go, creating an app for both Android and iOS. The goal is to create an app that helps users manage their medication schedules in a practical way, ensuring timely doses and reducing the risk of medication administration errors. With an intuitive interface and advanced features, the app provides a complete solution for medication management, helping patients follow their treatments more safely. Integration with family members is another key feature of the app, allowing them to track the treatment and offer support to the users. This connection between the patient and their family members contributes to more efficient supervision and improved treatment adherence, promoting safety and trust in the process. The development of Hémedios uses React Native with Expo, ensuring a smooth and interactive user experience. The choice of these technologies enables the creation of a mobile app compatible with both Android and iOS platforms.

Keywords: Keyword 1, Keyword 2, Keyword 3, Keyword 4, Keyword 5, Keyword 6.

SUMÁRIO

Lista de Tabelas	7
1 Introdução	8
1.1 Contextualização	8
1.2 Justificativa	8
1.3 Objetivos	8
1.3.1 Geral	8
1.3.2 Específicos	9
2 Fundamentação Teórica	10
2.1 Visão Geral	10
2.2 React Native	10
2.3 ExpoGo	10
2.3.1 Por que essas tecnologias foram escolhidas?	11
2.3.1.1 React Native	11
2.3.1.2 Expo Go	11
3 Metodologia	13
3.1 Desenvolvimento	13
3.2 Visão Geral	13
3.3 Testes	13
3.4 Frontend	13
3.4.1 React Native	14
3.4.2 Expo GO	14
4 Análise dos Resultados	15
4.1 Avaliação	15
4.2 Comparação	15
4.3 Github do projeto	15
5 Imagens da aplicação	16
5.1 telas da aplicação	16
5.2 Tela Principal do Aplicativo	16
5.3 Funcionalidade do Alarme de Medicamento	16
5.4 Detalhes do Layout e Estilo Visual	17
5.5 Tela de novo alarme	17
5.6 Tela de perfil	17

6	Sobre o site	18
6.1	imagens	18
6.2	Desenvolvimento de site Web	19
6.3	Links	19
7	Conclusões	20
7.1	Conclusões	20
7.2	Contribuições	20
7.3	Trabalhos Futuros	20
8	Bibliografia	21
8.0.1	Haverbeke, Marijn	21
8.0.2	Simpson, Kyle	21
8.0.3	Eisenman, Bonnie	21
8.0.4	Stefanov, Stoyan	21
8.0.5	Dabit, Nader	21
8.0.6	Hunter II, Thomas	21
8.0.7	Johansen, Christian	21
8.0.8	Hunt, Andrew e Thomas, David	22
8.0.9	Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John	22
8.0.10	Fowler, Martin	22
8.1	Documentação e Recursos Online	22
8.1.1	Expo	22
8.1.2	Redux	22
8.1.3	Wieruch, Robin	22
8.1.4	Bevacqua, Nicolas	22

LISTA DE TABELAS

2.1	Comparação das Tecnologias Utilizadas no Projeto	12
-----	--	----

1 INTRODUÇÃO

Resumo do capítulo

Gerenciar medicações diárias pode ser um desafio, ainda mais quando se trata de vários medicamentos com horários e doses específicas, com base nisso, desenvolvemos um aplicativo de medicação que oferece uma solução prática e eficiente. esse aplicativo permite que os usuarios acompanhem sua prescrições, com notificações automaticas garantindo que os medicamentos sejam tomados no hora certa, e assim obtendo uma adesão maior ao tratamento.

1.1 CONTEXTUALIZAÇÃO

A adesão ao tratamento médico é um dos principais desafios enfrentados por pacientes que necessitam de medicamentos contínuos ou em horários específicos, falhar em seguir um corretamento um cronograma de medicação pode resultar em complicações de saúde ou piora no quadro clinico, esse problema chega ser mais comum em idosos, pessoas com doenças crônicas ou pacientes que precisam de medicamentos ao longo do dia. Nesse caso o uso da tecnologia pode auxiliar na gestão de medicação, o aplicativo móvel Hémedio surge como uma ferramenta prática, criado para ajudar os usuarios a organizar suas medicações.

1.2 JUSTIFICATIVA

Hémedios é um aplicativo para ajudar os usuários a criar e gerenciar cronogramas de medicação oferecendo uma solução prática e eficiente. Ele não só facilita o cumprimento dos horários de doses, mas também reduz o risco de erros, como a administração incorreta ou a falta de doses. A personalização de notificações, o registro das medicações e a possibilidade de acompanhar o histórico de uso permitem que os pacientes sigam o tratamento de maneira mais segura, enquanto a integração com médicos e familiares melhora o suporte e a supervisão. Dessa forma, o aplicativo contribui para uma gestão mais eficiente da saúde, promovendo a continuidade do tratamento e melhorando a qualidade de vida do paciente.

1.3 OBJETIVOS

1.3.1 Geral

Desenvolver um aplicativo móvel que ajude os usuarios com o cronograma de medicações, facilitando a adesão dos tratamentos.

1.3.2 Específicos

Implementar um frontend utilizando ReactNative com Expogo, para um desenvolvimento mobile. Desenvolver um aplicativo interativo com ReactNative para oferecer uma experiência de usuário fluida. Validar a eficácia da solução através de testes e feedback de usuários.

2 FUNDAMENTAÇÃO TEÓRICA

Resumo do capítulo

As tecnologias fundamentais escolhidas para o desenvolvimento da aplicação móvel. São elas: ReactNative e Expo go. O React Native foi selecionado devido à sua eficácia na construção de interfaces de usuário dinâmicas e reativas, utilizando uma abordagem componentizada que facilita a manutenção e escalabilidade do projeto, além de ser um framework de uma linguagem de programação que o meu grupo tinha mais familiaridade. O Expo go reduz a complexidade inicial ao fornecer um conjunto pré-configurado de ferramentas e bibliotecas. Não é necessário configurar manualmente o ambiente de desenvolvimento para Android ou iOS, o que facilita o início do projeto, e é um aplicativo que pode ser instalado em um dispositivo móvel (Android ou iOS), permitindo a visualização do aplicativo em tempo real, sem precisar compilar o código nativo ou passar pelo processo de build complexo. Essas tecnologias foram escolhidas com base em sua maturidade, popularidade e capacidade de atender aos requisitos do projeto, proporcionando uma base sólida para o desenvolvimento de uma aplicação móvel moderna e eficiente.

2.1 VISÃO GERAL

Para entendermos por que as tecnologias escolhidas foram fundamentais para o desenvolvimento do projeto, é essencial analisar cada uma delas individualmente.

2.2 REACT NATIVE

React Native é um framework de código aberto para o desenvolvimento de aplicativos móveis que utiliza JavaScript e o conceito de componentes do React (uma biblioteca utilizada para a construção de interfaces de usuário.). React Native permite aos desenvolvedores escrever códigos para criar aplicativos móveis nativos para Android e iOS, utilizando a mesma base de código, com grande parte da lógica sendo compartilhada entre as duas plataformas, o que reduz o tempo de desenvolvimento e manutenção, ao mesmo tempo em que mantém um desempenho próximo do nativo e a flexibilidade de usar JavaScript.

2.3 EXPOGO

Expo Go é um aplicativo móvel e uma ferramenta de desenvolvimento que faz parte do Expo, uma plataforma open-source para o desenvolvimento de aplicativos móveis usando React Native. O Expo Go é projetado para facilitar o processo de desenvolvimento e teste de aplicativos móveis, permitindo a interação do aplicativo diretamente

no dispositivo, sem precisar passar por um processo completo de build e deploy a cada alteração. O Expo go funciona como um visualizador para o aplicativo em tempo real, visualizando as alterações feitas no código diretamente no dispositivo.

2.3.1 Por que essas tecnologias foram escolhidas?

2.3.1.1 React Native

A escolha do React Native se deve à sua popularidade e eficácia na construção de interfaces de usuário dinâmicas e reativas. Sua arquitetura componentizada permite uma fácil manutenção e escalabilidade do projeto, enquanto sua performance otimizada garante uma experiência de usuário fluida.

2.3.1.2 Expo Go

Expo Go é uma ferramenta de desenvolvimento para criar aplicativos móveis rapidamente, com um fluxo de trabalho simplificado e muitas funcionalidades prontas para uso, ideal para projetos menores e protótipos. O Expo go é fácil de configurar, oferece feedback instantâneo durante o desenvolvimento, permitindo que você faça ajustes rápidos e veja os resultados de maneira quase imediata.

Tabela 2.1 – *Comparação das Tecnologias Utilizadas no Projeto*

Critério	React Native	Expo Go
Popularidade	Um dos frameworks mais populares para o desenvolvimento de aplicativos móveis, com uma grande comunidade de desenvolvedores e amplo suporte.	Crescimento significativo de adoção, impulsionado pela demanda por uma tipagem mais robusta e segura em projetos JavaScript.
Flexibilidade	Arquitetura componentizada permite a construção de interfaces complexas e reativas, com grande flexibilidade e reusabilidade de código.	Integra-se perfeitamente ao ecossistema JavaScript existente, permitindo uma transição suave para equipes já familiarizadas com JavaScript.
Desempenho	Utiliza Virtual DOM para otimizar a atualização da interface, garantindo um desempenho eficiente mesmo em aplicações de grande escala.	Oferece um sistema de tipos estáticos que ajuda a identificar e corrigir erros em tempo de compilação, reduzindo a ocorrência de bugs e melhorando a performance do código.
Compatibilidade	Compatível com Android e iOS. Ele permite que você escreva código JavaScript para criar aplicativos móveis que rodem de forma nativa nas plataformas Android e iOS, com um único código base	Integra-se facilmente a frameworks e bibliotecas JavaScript populares.

3 METODOLOGIA

Resumo do capítulo

Neste capítulo, descrevemos a abordagem metodológica adotada para o desenvolvimento do projeto, incluindo as tecnologias utilizadas, a estrutura do projeto e as ferramentas de desenvolvimento empregadas. A metodologia seguida neste projeto inclui a análise de requisitos, escolha das tecnologias, desenvolvimento incremental dos componentes, e testes contínuos para garantir a funcionalidade e performance da aplicação.

3.1 DESENVOLVIMENTO

A aplicação foi desenvolvida utilizando uma abordagem modular, onde cada componente foi implementado e testado de forma independente antes de ser integrado ao projeto principal.

3.2 VISÃO GERAL

Fornecer uma estrutura sólida para o desenvolvimento do projeto, delineando os princípios, ferramentas e processos-chave que serão seguidos para alcançar os objetivos estabelecidos.

3.3 TESTES

Os testes foram conduzidos para garantir que todas as funcionalidades da aplicação funcionassem conforme o esperado. Foram realizados testes unitários nos componentes individuais e testes de integração na aplicação como um todo.

3.4 FRONTEND

O frontend é a parte de um sistema de software que os usuários interagem diretamente. Também é conhecido como a interface do usuário (UI). Ele consiste em todos os elementos visíveis e interativos de um aplicativo ou site, incluindo botões, menus, formulários, gráficos, entre outros.

As principais responsabilidades do frontend incluem:

1. Apresentação de Conteúdo: O frontend é responsável por exibir informações e conteúdo de maneira clara e organizada para os usuários.
2. Interatividade: Ele fornece meios para os usuários interagirem com o sistema, como clicar em botões, preencher formulários e navegar por diferentes páginas.

3. Usabilidade: O frontend é projetado para ser intuitivo e fácil de usar, proporcionando uma experiência agradável ao usuário.
4. Estilo e Design: Ele trata do layout, estilo e design visual do aplicativo ou site, garantindo uma aparência atraente e coesa.
5. Performance: O frontend também é responsável por garantir uma resposta rápida e eficiente às interações do usuário, minimizando os tempos de carregamento e proporcionando uma experiência fluida.

3.4.1 React Native

React Native é uma estrutura de aplicativo movel popular, baseada na linguagem JavaScript permite criar um aplicativo para varias plataformas usando a mesma base de código. É mantido pelo Facebook e uma comunidade de desenvolvedores individuais e empresas, chamado de programação hibrida, quando um mesmo código funciona em varias plataformas, contruido com base no React uma biblioteca JavaScript. React Native pode ser usado como uma base no desenvolvimento de aplicativos móveis. (??)

3.4.2 Expo GO

Aplicativo que permite rodar, testar e visualizar a aplicação RactNative durante o desenvolvimento, sem precisar compilar o código nativo, facilitando o processo de desenvolvimento permitindo que seja visto imediatamente as alterações feitas no código sem precisar recompilar e redistribuir o aplicativo.

4 ANÁLISE DOS RESULTADOS

Os resultados do desenvolvimento indicam que a aplicação é capaz de criar cronogramas e notificar os usuarios referentes a medicação, horarios e termino do tratamento, com uma interface intuitiva os resultados foi satisfatorio, atendendo aos requisitos definidos, nosso proximo passo é elevar este produto a novos patamares, até porque no momento o software só funciona em plataformas Andoid e IOS, queremos leva-lo para qualquer plataforma como web.

4.1 AVALIAÇÃO

A aplicação mostrou-se eficiente, porém foram identificados alguns desafios, como a limitação de sobrepor os outros aplicativos do android mostrando a notificação.

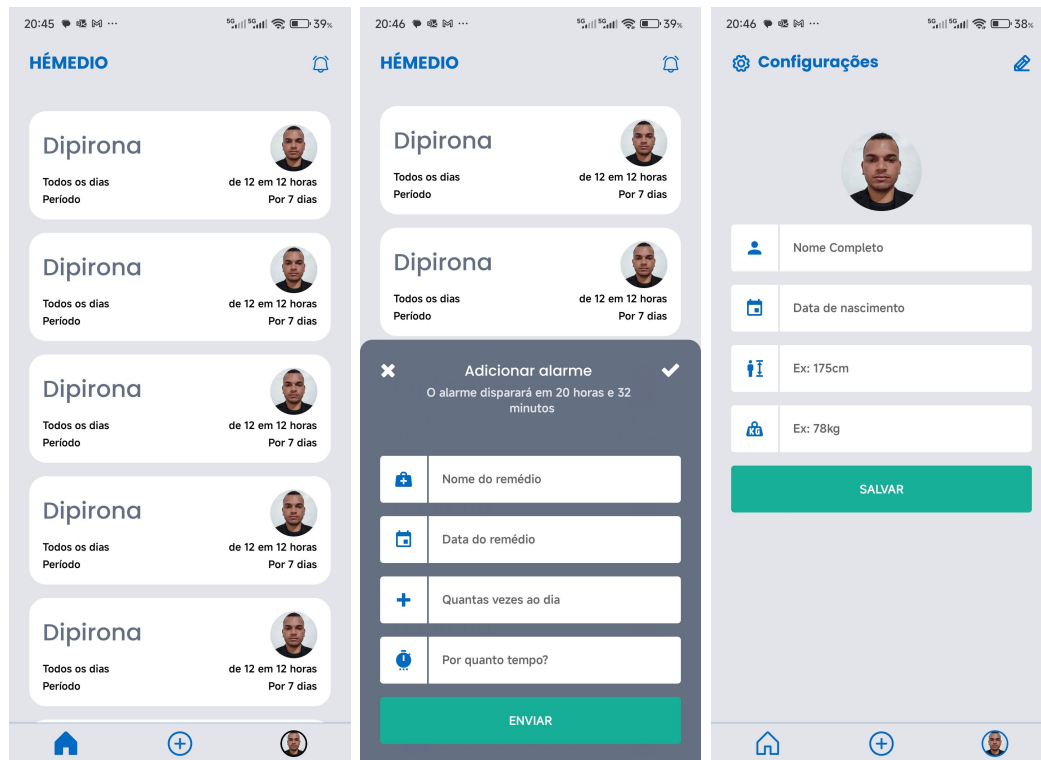
4.2 COMPARAÇÃO

Os resultados obtidos estão alinhados com os objetivos propostos. A aplicação oferece uma interface intuitiva e funcional, permitindo a criação de cronogramas para medicações e tratamentos.

4.3 GITHUB DO PROJETO

5 IMAGENS DA APLICAÇÃO

5.1 TELAS DA APLICAÇÃO



5.2 TELA PRINCIPAL DO APLICATIVO

A tela principal exibe a interface inicial do aplicativo "Hémedio", com elementos como o logo e um ícone de campainha, que simboliza o conceito central de um aplicativo de alarmes para lembrar o usuário sobre a hora de tomar medicamentos.

A tela usa um "ScrollView" para garantir que o conteúdo da tela seja rolável, acomodando diversos elementos de forma flexível. A organização visual e a ausência de barras de rolagem visíveis melhoram a experiência do usuário.

5.3 FUNCIONALIDADE DO ALARME DE MEDICAMENTO

Exibir as informações detalhadas sobre o medicamento que será administrado, juntamente com a programação de alarme para o lembrete diário. Este capítulo descreve o layout específico da interface do alarme para um medicamento, que pode ser editado ou ajustado conforme necessário.

5.4 DETALHES DO LAYOUT E ESTILO VISUAL

Este capítulo descreve como o layout e os estilos são definidos para criar uma interface agradável e intuitiva, permitindo que o usuário interaja de forma eficiente com os alarmes e os lembretes de medicamentos.

5.5 TELA DE NOVO ALARME

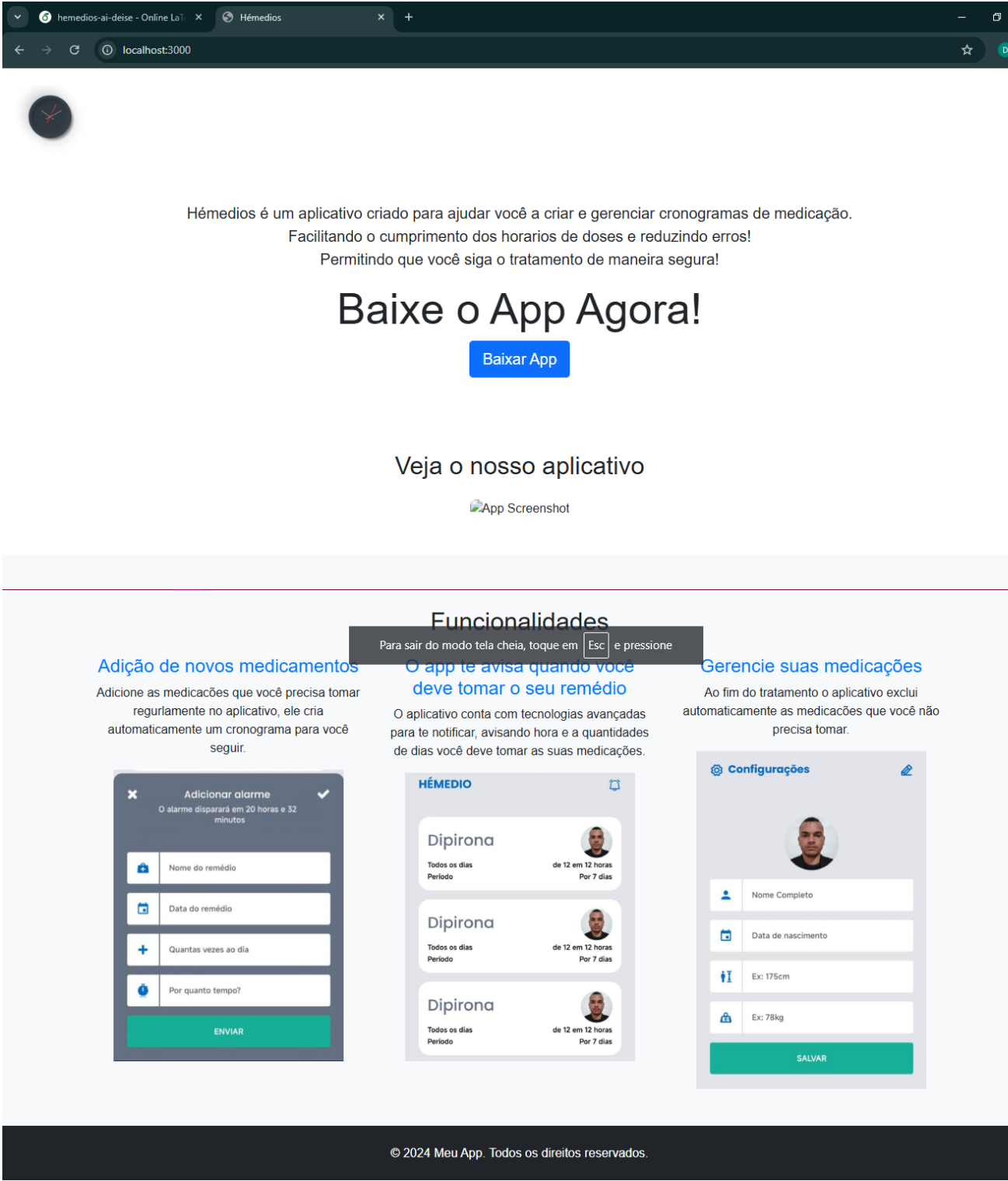
Este componente de alarme permite que o usuário insira informações sobre um medicamento e configure um alarme com os detalhes fornecidos. O BottomSheet oferece uma maneira prática de fazer isso sem sobrecarregar a interface principal do aplicativo. O uso de react-hook-form garante uma validação eficiente dos campos e facilita o manuseio dos dados do formulário.

5.6 TELA DE PERFIL

O código implementa uma tela de perfil de usuário em React Native, utilizando o pacote 'react-hook-form' para gerenciamento de formulário e validação de campos. A tela é composta por um cabeçalho com ícones de configurações e edição, seguidos por uma imagem de perfil que é carregada a partir de uma URL estática do GitHub. Abaixo da imagem, há quatro campos de entrada que permitem ao usuário editar seu nome, data de nascimento, altura e peso, com validações para garantir que todos os campos obrigatórios sejam preenchidos. Cada campo possui um ícone correspondente, como "account" para o nome, "calendar" para a data de nascimento, "human-male-height-variant" para altura e "weight-kilogram" para o peso. Esses campos são validados com mensagens de erro caso o usuário não preencha as informações corretamente. Quando o usuário pressiona o botão de salvar, os dados são exibidos no console e o formulário é resetado, permitindo ao usuário reiniciar o preenchimento se necessário. O botão de salvar também possui uma funcionalidade de pressionamento longo para limpar os campos. O design é simples, com uma imagem de perfil circular e um layout organizado, utilizando o tema de cores definido no objeto 'theme' para garantir consistência visual. A funcionalidade e a aparência da tela são pensadas para uma experiência de usuário clara e eficiente ao editar informações pessoais.

6 SOBRE O SITE

6.1 IMAGENS



6.2 DESENVOLVIMENTO DE SITE WEB

Com o aumento do número de pessoas que necessitam de tratamentos médicos com múltiplos medicamentos, surge a necessidade de ferramentas que facilitem o gerenciamento adequado dessas medicações. O "Hémedios" é um aplicativo web desenvolvido com o objetivo de ajudar os usuários a criar e gerenciar cronogramas de medicação, garantindo que os horários de doses sejam seguidos corretamente e que erros sejam minimizados. Este capítulo aborda a estrutura e o desenvolvimento do site do aplicativo, detalhando as funcionalidades e as tecnologias utilizadas.

Estrutura do Projeto O projeto foi desenvolvido utilizando Node.js com a framework Express para o backend, enquanto o frontend é composto por uma página HTML que carrega o conteúdo visual do aplicativo.

O site desenvolvido para o aplicativo "Hémedios" tem como objetivo facilitar a gestão de tratamentos médicos através de uma interface simples e funcional. Com o uso de Node.js e Bootstrap, a aplicação oferece um layout responsivo e acessível, proporcionando uma experiência de usuário agradável e eficiente.

6.3 LINKS

Para acessar o site hospedado [Clique aqui](#)

Para acessar o video de apresentação [Clique aqui](#)

7 CONCLUSÕES

7.1 CONCLUSÕES

O projeto atingiu seus objetivos principais, demonstrando que é possível criar um aplicativo mobile eficiente para gerenciar medicações e criar cronogramas para auxiliar os pacientes. A integração do React Native com o Expo Go, mostrou-se uma solução viável e eficaz.

7.2 CONTRIBUIÇÕES

Este trabalho contribui para a área de desenvolvimento mobile e farmaceutica, mostrando como tecnologias modernas podem ser utilizadas para resolver problemas e auxiliar pacientes.

7.3 TRABALHOS FUTUROS

Sugestões para trabalhos futuros incluem a otimização de performance para processamento de dados e a adição de mais funcionalidades, como o usuário compartilhar o cronograma com familiares, para ser realizado o acompanhamento por mais de um usuário.

8 BIBLIOGRAFIA

8.0.1 Haverbeke, Marijn

Eloquent JavaScript, 3ª edição.

8.0.2 Simpson, Kyle

You Don't Know JS (2014). Publicado por O'Reilly Media. Disponível em: <<https://github.com/getify/You-Dont-Know-JS>>. Acesso em: 29 nov. 2024.

8.0.3 Eisenman, Bonnie

Learning React Native, 1ª edição (2015). Publicado por O'Reilly Media. Disponível em: <<https://www.oreilly.com/library/view/learning-react-native/9781491929022/>>. Acesso em: 29 nov. 2024.

8.0.4 Stefanov, Stoyan

React – A Biblioteca JavaScript (2018). Publicado por O'Reilly Media. Disponível em: <<https://www.oreilly.com/library/view/react/9781491954628/>>. Acesso em: 29 nov. 2024.

8.0.5 Dabit, Nader

React Native in Action (2017). Publicado por Manning Publications. Disponível em: <<https://www.manning.com/books/react-native-in-action>>. Acesso em: 29 nov. 2024.

8.0.6 Hunter II, Thomas

React and React Native with Redux (2020). Publicado por Packt Publishing. Disponível em: <<https://www.packtpub.com/product/react-and-react-native-with-redux/9781838985979>>. Acesso em: 29 nov. 2024.

8.0.7 Johansen, Christian

Test-Driven Development with JavaScript (2015). Publicado por O'Reilly Media. Disponível em: <<https://www.oreilly.com/library/view/test-driven-development-with/9781783281951/>>. Acesso em: 29 nov. 2024.

8.0.8 Hunt, Andrew e Thomas, David

The Pragmatic Programmer, 2ª edição (2019). Publicado por Addison-Wesley. Disponível em: <<https://pragprog.com/titles/tpp20/the-pragmatic-programmer/>>. Acesso em: 29 nov. 2024.

8.0.9 Gamma, Erich; Helm, Richard; Johnson, Ralph; Vlissides, John

Design Patterns: Elements of Reusable Object-Oriented Software (1994). Publicado por Addison-Wesley. Disponível em: <<https://www.oreilly.com/library/view/design-patterns-elements/0201633612/>>. Acesso em: 29 nov. 2024.

8.0.10 Fowler, Martin

Refactoring: Improving the Design of Existing Code, 2ª edição (2018). Publicado por Addison-Wesley. Disponível em: <<https://www.oreilly.com/library/view/refactoring-improving-the/9780134757599/>>. Acesso em: 29 nov. 2024.

8.1 DOCUMENTAÇÃO E RECURSOS ONLINE

8.1.1 Expo

Expo Docs (2024). Disponível em: <<https://docs.expo.dev/>>. Acesso em: 29 nov. 2024.

8.1.2 Redux

Redux Docs (2024). Disponível em: <<https://redux.js.org/>>. Acesso em: 29 nov. 2024.

8.1.3 Wieruch, Robin

The Road to React (2024). Disponível em: <<https://www.roadtoreact.com/>>. Acesso em: 29 nov. 2024.

8.1.4 Bevacqua, Nicolas

Practical Modern JavaScript (2019). Publicado por Self-published. Disponível em: <<https://github.com/mjavascript/Practical-Modern-JavaScript>>. Acesso em: 29 nov. 2024.