



Emerson Leonardo Oliveira de Lira

Minha UBS: aplicativo para facilitar o acesso a serviços disponibilizados em Unidades Básicas de Saúde

**Recife
2024**

Emerson Leonardo Oliveira de Lira

Minha UBS: aplicativo para facilitar o acesso a serviços disponibilizados em Unidades Básicas de Saúde

Monografia apresentada ao curso de bacharelado em Sistemas de Informação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE
Departamento de Estatística e Informática
Curso de Bacharelado em Sistemas de Informação

Orientador: Cleviton Vinicius Fonseca Monteiro

**Recife
2024**

Dados Internacionais de Catalogação na Publicação
Sistema Integrado de Bibliotecas da UFRPE
Bibliotecário(a): Auxiliadora Cunha – CRB-4 1134

L768m Lira, Emerson Leonardo Oliveira de.
Minha UBS: aplicativo para facilitar o acesso a serviços disponibilizados em Unidades Básicas de Saúde / Emerson Leonardo Oliveira de Lira. – Recife, 2024. 41 f.; il.
Orientador(a): Cleviton Vinicius Fonseca Monteiro. Trabalho de Conclusão de Curso (Graduação) –
Universidade Federal Rural de Pernambuco, Bacharelado em Sistemas da Informação, Recife, BR-PE, 2024.
Inclui referências.
1. Aplicativos móveis. 2. Saúde pública. 3. Design thinking. I. Monteiro, Cleviton Vinicius Fonseca, orient. II. Título

CDD 004

Emerson Leonardo Oliveira de Lira

Minha UBS: aplicativo para facilitar o acesso a serviços disponibilizados em Unidades Básicas de Saúde

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado em: 09 de Outubro de 2024.

BANCA EXAMINADORA

Cleviton Vinicius Fonseca Monteiro (Orientador)
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Silvana Bocanegra
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Recife
2024

Resumo

O Sistema Único de Saúde é o sistema que garante à população brasileira acesso aos serviços de saúde de forma gratuita pela rede pública. A principal forma de acesso a esses serviços é a partir das unidades básicas de saúde, popularmente conhecidas como "postinhos". Os responsáveis nessas unidades por facilitar esse acesso aos moradores são os agentes comunitários, de saúde e de endemias, que atuam como pontes entre os moradores e as unidades de saúde. Com ferramentas para auxiliar o planejamento de estratégias para atuar na região e facilitar a comunicação com os moradores, o Minha UBS propõe, através de um aplicativo, reduzir as barreiras entre a população e os serviços públicos de saúde.

Palavras-chave: desenvolvimento mobile, saúde pública, design thinking.

Abstract

The Unified Health System is the system that guarantees the Brazilian population access to health services free of charge through the public network. The main way of accessing these services is from basic health units, popularly known as "postinhos", Those responsible in these units for facilitating this access for residents are community, health and endemic diseases, agents who act as bridges between residents and health units. With tools to help plan strategies to operate in the region and facilitate communication with residents, Minha UBS proposes, through an application, to reduce barriers between the population and public health services.

Keywords: mobile development, public health care, design thinking.

Lista de Ilustrações

Figura 1 - Etapas do Modelo em Cascata	24
Figura 2 - Protótipo de Tela de Famílias e moradores	27
Figura 3 - Protótipo de Tela de Notícias	27
Figura 4 - Protótipo de Tela de informações do posto	27
Figura 5 - Protótipo de Tela de informações do usuário	27
Figura 6 - Arquitetura do Spring	29
Figura 7 - Arquitetura do Flutter	29
Figura 8 - Diagrama Entidade-Relacionamento do Banco de Dados	31
Figura 9 - Tela de Cadastro	32
Figura 10 - Tela de Login	32
Figura 11 - Tela de Notícias	33
Figura 12 - Tela de Listagem de Moradores	34
Figura 13 - Tela de Listagem de Famílias	34
Figura 14 - Tela de Visitas	35
Figura 15 - Menu de Registro de Visita	36
Figura 16 - Confirmar Registro de Visita	36
Figura 17 - Registro de Caso	36
Figura 18 - Confirmar Registro de Caso	36
Figura 19 - Filtro de Casos	37
Figura 20 - Mapa de Calor dos casos	37

Lista de Tabelas

Tabela 1 - Tabela de Requisitos Funcionais do Aplicativo	25
Tabela 2 - Definição da Arquitetura Spring.	29
Tabela 3 - Definição da Arquitetura Flutter.	30

Lista de abreviaturas e siglas

API	Interface de programação de aplicações
ACS	Agente Comunitário de Saúde
ACE	Agente Comunitário de Endemias
SUS	Sistema Universal de Saúde
UBS	Unidade Básica de Saúde

Sumário

1. Introdução	11
1.1. Objetivos	12
1.1.1. Objetivo Geral	12
1.1.2. Objetivo Específico	12
1.2. Justificativas	12
1.3. Organização do Trabalho	13
2. Referencial Teórico	13
2.1. Unidades Básicas de saúde	13
2.2. Teoria Fundamentada nos dados	14
2.3. Tecnologias Utilizadas	15
2.3.1. Flutter	15
2.3.2. Spring	15
2.3.3. Android Studio	16
2.3.4. Visual Studio Code	17
2.4. Trabalhos Relacionados	18
2.5. Aplicativos Relacionados	19
2.5.1. Meu SUS Digital	19
2.5.2. e-SUS Território	20
2.5.3. e-SUS AC	20
3. Materiais e Métodos	21
3.1. Ferramentas Utilizadas	21
3.1.1. Design Thinking	21
3.1.1.1. Inspiração	21
3.1.1.2. Ideação	22
3.1.1.3. Implementação	23
3.2. Modelo de Desenvolvimento em Cascata	23
3.3. Análise dos Dados	24
4. Resultados	25
4.1. Requisitos Funcionais	25
4.2. Prototipação	26
4.3. Projeto de Sistema	28
4.4. Implementação	32
4.5. Validação do público-alvo	38
5. Conclusão	38
5.1. Trabalhos Futuros	39
6. Referências	40

1. Introdução

O Sistema Único de Saúde (SUS) é o sistema que assegura, de forma pública, o acesso a serviços de cuidados com a saúde para toda a população brasileira e estrangeiros que estiverem no Brasil e precisarem de assistência de saúde, de forma integral, igualitária e universal, ou seja, não faz, e nem deve fazer qualquer distinção entre os usuários, garantido pela Constituição de 1988. É o único sistema de saúde pública no mundo que atende mais de 190 milhões de pessoas, com 80% dependendo exclusivamente dele para qualquer atendimento de saúde (UNA-SUS, 2019).

As ações e serviços referentes ao SUS são gerenciados, de forma solidária e participativa, pelos três entes da Federação: a União, os Estados e os municípios. Assim, a rede de serviços consegue atender à atenção primária, média e alta complexidade, os serviços de urgência e emergência, a atenção hospitalar, as ações e serviços das vigilâncias epidemiológica, sanitária e ambiental e assistência farmacêutica (MINISTÉRIO DA SAÚDE, 2021).

O acesso a esses serviços começa muitas vezes nas Unidades Básicas de Saúde (UBS), conhecidas popularmente como postinhos. Elas têm como objetivo atender até 80% dos problemas de saúde da população, sem que haja a necessidade de encaminhamento para outros serviços, como emergências e hospitais e oferecem serviços como fazer curativos, inalações, tomar vacinas, coletas de exames laboratoriais, tratamento odontológico, receber medicação básica, encaminhar para atendimento com especialistas e etc (CONASS, 2019) .

A ponte entre os moradores da região e a UBS são os agentes comunitários de saúde. Eles atuam na prevenção e promoção de saúde dentro dos bairros, com uma equipe multidisciplinar além da elaboração, avaliação, programação e reprogramação dos planos de ações locais de saúde (CEEN, 2021). Porém os agentes comunitários têm o trabalho prejudicado por problemas de comunicação que levam alguns locais a adotar redes sociais como ferramenta para suprir essa necessidade (DIÁRIO GAÚCHO, 2021).

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver um sistema mobile para dispositivos android e iOS, que auxilie na rotina de trabalho dos agentes comunitários, facilitando o planejamento e ações estratégicas voltados para promoção e prevenção de saúde dentro das regiões de atuação das UBS, melhorando assim a qualidade de vida dos residentes dessas comunidades.

1.1.2 Objetivo Específico

- Realizar o levantamento das funções principais para o sistema compreendendo os processos relacionados a comunicação entre população e funcionários dos postos de saúde.
- Prototipação das funcionalidades levantadas para o desenvolvimento da ferramenta.
- Desenvolvimento de uma aplicação mobile, com API e banco de dados, que atenda as necessidades levantadas.

1.2 Justificativas

Uma das funções fundamentais no trabalho do agente comunitário de saúde é o fortalecimento do vínculo entre as pessoas, que vivem no território, usuárias do SUS e os serviços de Atenção Primária de Saúde (APS) oferecidas pelo SUS (CELUPPI et al., 2022).

Parte desse processo de fortalecimento de vínculo envolve coletar informações dessas pessoas para o oferecimento mais efetivo dos serviços de APS. Esse processo de coleta de informações se tornou mais prático com o desenvolvimento do e-SUS território, que embora possua funcionalidades que permitem o mapeamento e gerenciamento de informações das famílias localizadas no território e a sincronização dessas informações com o Prontuário Eletrônico do Cidadão, ainda oferece lacunas que prejudicam o trabalho desses agentes.

Espera-se que a partir desse trabalho, haja redução dos obstáculos existentes aumentando o repertório de ferramentas disponíveis para os agentes comunitários e melhorando o acesso da população à rede de saúde pública.

1.3 Organização do Trabalho

Este trabalho está separado em 6 capítulos, onde: o Capítulo 1 trata da introdução do tema, trazendo também seus objetivos gerais e específicos, o Capítulo 2 fala sobre o referencial teórico utilizado para elaboração do trabalho, o Capítulo 3 descreve os materiais e métodos utilizados como ferramentas, linguagens de programação e ambientes de desenvolvimento, o Capítulo 4 aborda os trabalhos relacionados com esse tema, o Capítulo 5 aborda os resultados obtidos com a aplicação dos métodos e materiais apresentados anteriormente e o Capítulo 6 aborda as perspectivas de trabalhos futuros.

2. Referencial Teórico

2.1 Unidades Básicas de Saúde

As unidades básicas de saúde são as linhas de contato entre os cidadãos e os serviços oferecidos no SUS. Conhecidas popularmente como postinhos de saúde, elas têm como missão fornecer serviços essenciais para a população em questões de promoção de saúde, prevenção de doenças, tratamento de condições básicas, além de encaminhamentos para níveis mais especializados de atendimento (COSTA, 2023).

Dentro das UBS atuam equipes multidisciplinares como médicos, enfermeiros, dentistas, nutricionistas e assistentes sociais, de acordo com as necessidades da região. Além desses profissionais, existem os agentes comunitários, de saúde e de endemias, esses profissionais atuam construindo e mantendo um elo entre a comunidade e a UBS. Os agentes comunitários atuam diretamente nas comunidades planejando estratégias e entendendo as

necessidades dos moradores para facilitar e melhorar o acesso aos serviços de saúde.

A construção desse elo é obtida a partir da comunicação com os moradores, como a ponte de informações entre moradores e o posto. Os agentes comunitários, utilizam os dados coletados, a partir das visitas domiciliares aos moradores, como casos de condições clínicas ou sintomas de alguma doença, para traçar planejamentos estratégicos e ações voltadas para a melhoria da qualidade de vida dos residentes daquela região. Essas ações podem ser através da divulgação de informações que auxiliem os residentes como campanhas de vacinação e dias de consultas com médicos ou atividades de combate à dengue com utilização de larvicidas em locais de difícil acesso (MINISTÉRIO DA SAÚDE, 2009).

2.2 Teoria Fundamentada nos Dados

De acordo com Merriam (2009) a análise de dados é um processo de dar sentido aos dados, algo como determinar a melhor maneira de organizar o material a ser analisado em um relato narrativo das descobertas. Essa análise é estendida pelos pesquisadores para desenvolvimento de categorias para interpretar o significado desses dados e essas categorias tornam-se as conclusões do estudo, construindo assim uma teoria fundamentada nos dados.

Corbin e Strauss (2008) definem que esse processo de elaboração de uma teoria segue uma sequência de passos, que conceituam, definem e desenvolvem as categorias e depois as relacionam por meio de hipóteses ou declarações de relações. Esses passos são:

- Codificação aberta: é a etapa em que os pontos importantes dos dados são evidenciados, agrupados e nomeados considerando algo que represente essa associação comum, descobrindo assim as categorias.
- Codificação axial: é a etapa em que as categorias são relacionadas às subcategorias para geração de explicações mais precisas e completas sobre os fenômenos.
- Codificação seletiva: é a etapa em que as categorias são refinadas e integradas, e assim os dados começam a ganhar forma de teoria.

2.3 Tecnologias Utilizadas

2.3.1 Flutter

O Flutter¹ é um kit de ferramentas de interface alimentado pela plataforma Dart, uma linguagem otimizada para desenvolvimento de aplicativos rápidos e multiplataforma. O Flutter fornece ferramentas e bibliotecas para criar experiências de interface de usuário que rodam em iOS, Android, macOS, Windows, Linux e na web (DART, 2021).

Esse kit de ferramentas é arquitetado a partir de *widgets*, elementos que combinados formam as interfaces, como se o aplicativo fosse um quebra-cabeças e os widgets as peças, e ao encaixar essas peças as imagens que compõem o quebra-cabeças começam a se formar, alguns desses widgets estão mais relacionados a posicionamento de outros widgets na interface da aplicação como: Column, Row, Scaffold ou Stack e outros estão mais relacionados a exibição como: Text, RaisedButton e Switch (MELO, 2019).

Esses widgets podem ser separados em dois grupos: Stateful e Stateless. O que diferencia os widgets nesses grupos é a forma como ele lida as informações alteradas em tempo de execução, essa alteração é chamada de estado, os widgets que conseguem fazer essa alteração são chamados de stateful e fazem essa alteração a partir do método setState. Enquanto os widgets stateless não conseguem realizar essa alteração, não possuem o método setState e exibem as informações na tela de forma estática (PINHEIRO, 2020).

O Flutter transforma o código-fonte em código nativo permitindo acesso a recursos diretos do sistema. Esse acesso fornece ao Flutter uma maior performance e a possibilidade de compilar em mais de uma plataforma (DORIGHELLO, 2022).. Com isso é possível, a partir do mesmo código gerar uma aplicação capaz de executar em um dispositivo móvel, Android ou iOS, e uma aplicação web.

2.3.2 Spring Boot

¹ Flutter documentation. Disponível em: <<https://docs.flutter.dev/>>.

O Spring Boot² É uma ferramenta de software que facilita o uso de estruturas baseadas em Java, que é uma linguagem de programação criada em 1995 orientada a objetos multiplataforma popular no desenvolvimento de aplicativos em todo mundo. Essa ferramenta oferece aos desenvolvedores grandes estruturas de códigos pré-escritas para que possam usar e adaptar às suas necessidades e assim acelerar o processo de desenvolvimento tanto no início, na fase de criação da aplicação quanto no decorrer do desenvolvimento com a fácil reutilização desses componentes (MICROSOFT, 2023).

Utilizando ferramentas já existentes no *Spring Framework*, o Spring Boot ajuda a aumentar a produtividade do desenvolvedor, principalmente no início do desenvolvimento de novos projetos. Como a construção de um novo projeto envolve a escrita de muitas configurações que são sempre as mesmas, o Spring inicia novos projetos com essas configurações já definidas, porém permitindo a alteração delas a qualquer momento (BIANCHI,2015).

Além da facilidade na iniciação de novos projetos, o spring oferece bibliotecas que facilitam o desenvolvimento e manutenção do software como os Spring Starters, que são um grupo de bibliotecas que auxiliam em funções específicas da aplicação, dois exemplos de utilização dessas bibliotecas são o Spring Boot Starter Web³ e o Spring Boot Starter Data JPA⁴ (SAMPAIO, 2021). O Spring Boot Starter Web possui métodos e configurações pré definidas para a construção de aplicações web permitindo assim a comunicação entre o aplicativo e o servidor através do protocolo HTTP, enquanto o Spring Boot Starter Data JPA facilita a construção da camada de persistência, abstraindo assim toda parte de código que envolve operações entre o servidor e o banco de dados, com essa biblioteca é possível definir classes que representam tabelas no banco de dados e realizar operações com a chamada de métodos.

2.3.3 Android Studio

² Spring boot. Disponível em: <<https://spring.io/projects/spring-boot>>.

³ Web. Disponível em <<https://docs.spring.io/spring-boot/reference/web/index.html>>

⁴ Spring Data JPA. Disponível em <<https://spring.io/projects/spring-data-jpa>>

O Android Studio⁵ é o ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de aplicativos para o sistema operacional Android. O Android Studio possui uma série de ferramentas e recursos que possibilitam o aumento da produtividade do desenvolvedor como: Um ambiente unificado que possibilita o desenvolvimento para todos os dispositivos Android, a edição em tempo real para atualizar elementos combináveis em emuladores e dispositivos físicos, modelos de código e integração com GitHub⁶ para ajudar a criar recursos comuns de apps e importar exemplos de código, ferramentas para detectar problemas de desempenho, usabilidade, compatibilidade com versões, entre outros (DEVELOPERS, 2018).

Como uma IDE (Ambiente de desenvolvimento integrado) as ferramentas fornecidas pelo Android Studio auxiliam no desenvolvimento dos códigos responsáveis pela criação das telas do aplicativo, além disso é possível emular o aplicativo em dispositivos físicos e virtuais validando assim o comportamento do aplicativo nesses dispositivos.

2.3.4 Visual Studio Code

O Visual Studio Code⁷ é um editor de código-fonte leve e gratuito personalizável, por meio de extensões, que oferece suporte a muitas necessidades de programação diferentes, dentre essas necessidades existe o suporte para diversas linguagens como: C++, C#, Java, Python, PHP, Go, .NET. Além disso, é possível depurar código fonte diretamente no editor com pontos de interrupção, pilhas de chamadas, console interativo, anexar aplicativo em execução e aumentar suas possibilidades de utilização a partir das extensões oferecidas dentro do próprio editor (MICROSOFT, 2018).

A decisão do editor de código-fonte para o projeto é uma decisão importante que pode tornar mais fácil a produção do código da aplicação, os testes e as futuras manutenções para melhorias de código e soluções de erros. Uma forma de saber se o visual studio é a solução mais adequada para as necessidades do projeto é utilizando os seguintes critérios: Existe um editor de código oficial ou IDE para a

⁵ Android Studio. Disponível em: <https://developer.android.com/studio?hl=pt-br>

⁶ Documentação de introdução ao GitHub. Disponível em: <https://docs.github.com/pt/get-started>.

⁷ Visual Studio Code documentation. Disponível em: <https://code.visualstudio.com/docs>

linguagem de programação ou estrutura que estou usando? O Visual Studio Code dá suporte à linguagem de programação ou à estrutura que estou usando? Quero um editor de código ou IDE com uma pegada pequena? Preciso de integração de controle do código-fonte com meu editor de código ou IDE? (MICROSOFT, 2024). Ao analisar esses critérios é possível entender se o Visual Studio possui os recursos adequados para o desenvolvimento do projeto.

2.4 Trabalhos Relacionados

Ao longo dos anos alguns estudos foram realizados com o objetivo de aprimorar os processos referentes à informação dentro da saúde pública, melhorando assim o acesso da população a esses serviços. Nesta seção serão abordados alguns desses estudos.

Coelho (2023) desenvolveu um aplicativo para dispositivos móveis, que permite que usuários residentes do município de Tijucas, em Santa Catarina, acompanhem informações e disponibilidade de medicamentos fornecidos pela rede pública, além de localizar farmácias próximas com serviço de dispensação. O desenvolvimento do aplicativo é realizado com a utilização do modelo de cascata e é dividido em 5 etapas: Análise de requisitos, projeto, implementação, teste de software e homologação. Com as metodologias adotadas o autor consegue elaborar os requisitos funcionais e não-funcionais do projeto além dos diagramas como o de caso de uso, que exibe algumas funcionalidades do aplicativo. Além disso, com a utilização da linguagem Flutter e de seus pacotes, o autor conseguiu desenvolver, testar o aplicativo e homologar com a geração de um executável .apk.

Oliveira (2021) busca implementar um aplicativo para dispositivos móveis que forneça ao usuário maior praticidade no gerenciamento de suas vacinas e obtenção de informações sobre elas. Esse desenvolvimento foi separado nas etapas de: Planejamento, Pesquisa de Campo, Levantamento de Requisitos, Protótipo, Validação com Usuários e Redacional. Essa separação de etapas se provou eficaz nos resultados onde foram apresentados os requisitos funcionais e não-funcionais, diagrama de casos de uso, modelo de dados e as principais telas do aplicativo, além de ter respostas positivas na entrevista de validação de usuário.

Em SOUZA et al. (2022) almejam trazer praticidade para gestantes através de um aplicativo móvel para acompanhamento do pré-natal, utilizando um estudo metodológico dividido em três etapas: revisão integrativa da literatura utilizando bases de dados bibliográficas, construção do aplicativo utilizando como linguagem de programação principal o java, com utilização do Java Server Faces (JSF), para desenvolvimento das interfaces, Spring para gerenciamento dos módulos da aplicação e Hibernate para mapeamento objeto-relacional com a base de dados, e a validação do conteúdo com 21 especialistas sendo 14 enfermeiros obstetras e 7 profissionais da área de tecnologia de informação e comunicação. O aplicativo desenvolvido possui 111 telas com funcionalidade sobre as fases de pré-natal, parto, puerpério e aleitamento materno, além de oferecer à gestante funcionalidades como: caderneta virtual da gestante, despertador como lembrete de consultas e o menu fale conosco.

2.5 Aplicativos Relacionados

2.5.1 Meu SUS Digital

O aplicativo Meu SUS Digital é o aplicativo oficial do ministério da saúde e a principal porta de acesso digital para os cidadãos aos serviços oferecidos pelo SUS, a partir dele é possível que o usuário consiga informações como Cartão nacional de saúde, resultado de exames laboratoriais, medicamentos dispensados pelo programa Farmácia popular, registros de doação de sangue e acompanhar seu histórico clínico (MINISTÉRIO DA SAÚDE, 2023).

O aplicativo oferece também ao usuário, através da funcionalidade Rede de saúde, mapear estabelecimentos de saúde próximos de sua localização. Porém as informações que são disponibilizadas sobre os estabelecimentos de saúde, incluindo as UBS, se resumem a telefone, endereço, tipo de estabelecimento (público ou privado) e qual órgão administra (municipal, estadual ou federal) .

Apesar do foco do aplicativo ser para usuários da rede pública de saúde, as funcionalidades referentes às unidades de saúde resumem a interação do usuário a uma consulta de informações gerais, para que a partir dessas informações o usuário

consiga entrar em contato com a unidade de saúde e ter acesso a alguma informação específica. Essa falta de funcionalidades específicas para UBS dificulta o acesso a informações e conseqüente dificultando a prevenção e promoção de saúde para as pessoas que utilizam a rede pública.

2.5.2 e-SUS Território

Buscando reestruturar as informações da Atenção Primária à Saúde (APS) em nível nacional, o Departamento de Saúde da Família (DESF) desenvolveu uma estratégia, o e-SUS APS. Essa estratégia busca informatizar os processos da atenção primária à saúde através de 2 softwares e 4 aplicativos, dentre esses aplicativos existem dois que são focados na rotina de trabalho das UBS: o e-SUS território e o e-SUS AC.

Dentre as atividades desempenhadas pelos agentes comunitários existem processos que envolvem o fortalecimento do vínculo entre a comunidade e a UBS e parte desse processo envolve coletar informações dessas pessoas para o oferecimento mais eficaz dos serviços de APS. Almejando facilitar essa coleta de informações foi desenvolvido o e-SUS território, com funcionalidades que permitem o mapeamento e gerenciamento de informações das famílias localizadas no território e a sincronização dessas informações com o Prontuário Eletrônico do Cidadão (PEC), mas apesar de auxiliar a rotina de trabalho dos agentes o aplicativo ainda não permite o registro de informações de saúde dos cidadãos como identificação de sintomas e casos clínicos, e conseqüentemente não é possível obter relatórios sobre essas informações dentro do aplicativo. Além disso, o e-SUS território se restringe a esse mapeamento e coleta de dados não permitindo ao usuário a divulgação de informações sobre a UBS pelo aplicativo e não contribuindo com a promoção de saúde na região.

2.5.3 e-SUS AC

Além das atividades que envolvem gerenciamento de informações dos moradores da região, dentro das UBS também são realizadas as atividades coletivas, que podem ser divididas em atividades coletivas administrativas, como reuniões com a equipe da UBS ou com outras equipes, para planejamento de ações

na região e atividades coletivas em saúde, como palestras educativas sobre saúde e atendimento ou avaliação em grupo, atuando na promoção de saúde dos residentes. O objetivo do e-SUS AC é auxiliar os profissionais das equipes de Atenção Básica a gerenciar essas atividades, sincronizando essas informações com o prontuário eletrônico do cidadão (BRASIL, 2021). Embora o aplicativo ajude os usuários com a organização e gestão dessas atividades coletivas, as funcionalidades são restritas apenas a essa parte do trabalho dos agentes, e delega a divulgação dessas atividades para outros meios como redes sociais. Além disso, as funcionalidades apresentadas não têm assim uma relação direta com os moradores da região nem oferecendo com outros serviços oferecidos na UBS.

3. Materiais e Métodos

3.1 Ferramentas Utilizadas

3.1.1 *Design Thinking*

Design Thinking é uma abordagem centrada no ser humano para a inovação. Ela se baseia no kit de ferramentas do designer para integrar as necessidades das pessoas com as possibilidades da tecnologia, é uma forma de transformar a maneira como produtos e serviços são desenvolvidos. (IDEO, 2022).

Em Brown (2020), o processo de desenvolvimento de uma solução, utilizando *Design Thinking*, é definido como um sistema de espaços que se sobrepõem. Esses espaços são separados em: inspiração, idealização e implementação.

3.1.1 Inspiração

É o espaço dentro do processo onde são realizadas as pesquisas, onde as pessoas relacionadas ao problema são ouvidas com o propósito de entender quais são as necessidades daquelas pessoas. A técnica escolhida como ferramenta de escuta foi a de entrevistas onde seis agentes comunitários de saúde e de endemias responderam um formulário online com as seguintes perguntas:

1. Qual seu nome?
2. Qual unidade de saúde você trabalha?

3. Quais são as suas responsabilidades como agente comunitário de saúde?
4. Atualmente, como você se comunica com os moradores da sua área de atuação?
5. Quais são as ferramentas que você utiliza para essa comunicação com os moradores?
6. Quais as principais demandas de moradores de sua área de atuação em relação à saúde?
7. Quais são os desafios que você enfrenta nesta comunicação?
8. Quais são as informações que você precisa coletar, registrar e/ou compartilhar com os moradores da sua área?
9. Com os profissionais da sua área de atuação, quais suas informações você precisa coletar, registrar e compartilhar?
10. Como é que você registra e armazena essas informações?
11. Como você avalia a qualidade das informações que recebe dos moradores pelos meios de comunicação atuais?
12. Quais as funcionalidades ou recursos, você gostaria que um aplicativo tivesse para facilitar seu trabalho?
13. Como você se atualiza sobre orientações e novidades relacionadas à saúde?

A partir dessas respostas foi utilizada a ferramenta de *open coding*, uma ferramenta para o desmembramento desses dados em texto para obtenção de um novo conhecimento, transformando assim essas respostas em uma lista de requisitos do aplicativo.

3.1.2 Ideação

Nessa etapa é onde as informações coletadas na inspiração são transformadas em algo mais tangível e próximo da solução final, é onde são transformadas em ideias. Dentre as ferramentas disponíveis para essa

transformação a escolhida foi a prototipagem, a partir do Figma⁸, uma ferramenta de design online que une a acessibilidade da web com as funcionalidades de um aplicativo nativo (EBAC, 2023).

3.1.3 Implementação

A partir daqui as ideias obtidas na etapa anterior são transformadas em produto. Nesse caso o produto transformado será um aplicativo para dispositivos móveis, Android ou iOS, utilizando como ambiente de desenvolvimento o Android Studio e o Visual Studio Code, e as linguagens Flutter e Spring.

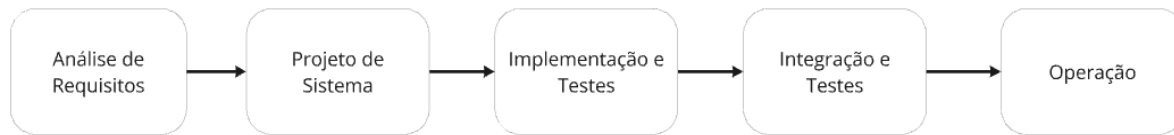
3.2 Modelo de Desenvolvimento em Cascata

O modelo de desenvolvimento de software em cascata foi o primeiro modelo a ser publicado e tem como base processos gerais de engenharia de sistemas. Conhecido pelo encadeamento de suas atividades entre as fases, esse modelo é dirigido a planos, ou seja, ao implementar esse modelo de processo deve-se planejar e programar todas as atividades antes de começar a trabalhar nelas (SOMMERVILLE, 2011).

As etapas do processo de desenvolvimento no modelo cascata podem ser representadas pela Figura 1. Nessas etapas, a Análise de Requisitos é onde as funcionalidades e metas do sistema são estabelecidas e definidas assim como seu funcionamento. O Projeto de Software é onde se planejam os relacionamentos do sistema, estruturas e a arquitetura. Implementação e Testes, é a etapa em que o software começa a ser de fato desenvolvido com rodadas de testes a cada especificação atendida. Na fase de Integração e Testes o software começa a ser testado como um sistema completo, para garantir que os relacionamentos e comunicações estejam de acordo com o especificado. Na Operação o sistema é colocado em uso e novas correções de erro que não tinham sido descobertos.

⁸ <https://www.figma.com/pt-br/design/>

Figura 1 - Etapas do Modelo em Cascata.



Fonte: autoria própria.

Com a elaboração de requisitos bem definidos nas etapas de design thinking, foi montado um escopo de atividades com baixa necessidade de adaptações e incrementos durante o processo de desenvolvimento, formando assim o ambiente ideal para a aplicação do modelo cascata (PRESSMAN; MAXIM, 2019).

3.3 Análise dos Dados

A partir da aplicação do formulário online foram obtidas respostas que revelavam informações interessantes para análise do problema. A partir dessas respostas seguindo as etapas descritas no processo de design thinking foram aplicadas as técnicas de codificação na teoria fundamentada nos dados (STRAUSS; CORBIN, 2008).

Apesar do foco da teoria fundamentada nos dados ser a elaboração de uma teoria, a partir da relação dos códigos, nesse trabalho essas técnicas foram utilizadas para a identificação das necessidades do público-alvo do aplicativo e levantamento dos requisitos para o desenvolvimento do aplicativo. Com a codificação aberta foram identificados códigos como: Whatsapp, Instagram, Notícias do posto, informações do posto, relatório, ações estratégicas, mapeamento, hipertensão e diabetes. Com isso foi possível aplicar a codificação Axial para agrupar esses códigos através da relação entre eles gerando códigos como: Utilização de redes sociais para divulgação de notícias do posto e relatório do mapeamento de casos para planejamento de ações. E por fim foi aplicado a codificação seletiva para refinar esses novos códigos e transformando em requisitos como: Adicionar notícias, criar notícias e relatório de casos clínicos.

A elaboração de requisitos inicia o processo de desenvolvimento em cascata, cujo primeiro passo é a análise de requisitos. A definição dos requisitos de forma

clara e objetiva permitiu a elaboração da arquitetura do software, com o Princípio de Separação de Responsabilidades, onde a divisão de pastas e pacotes dentro do softwares possuem responsabilidades únicas. A fase de implementação do software iniciou assim que a arquitetura e os requisitos foram estabelecidos, nessa fase o desenvolvimento de cada requisito foi realizado separadamente, assim como os testes, e quando foram finalizados, seguiu-se para a fase de integração onde foram implementadas as conexões entre as partes, e o software foi testado por completo. Concluindo com a fase de operação, onde foram gerados os arquivos executáveis do software e validados por um usuário real.

4. Resultados

4.1 Requisitos Funcionais

Um requisito é que define o que um sistema deve fazer e quais são as suas restrições. Os requisitos que se referem à definição do que o sistema deve fazer são chamados de requisitos funcionais e os que definem suas restrições são chamados de requisitos não-funcionais (VALENTE, 2020). Na **Tabela 1**, são descritos os requisitos funcionais do aplicativo, baseados nas necessidades relatadas no formulário.

Tabela 1 - Tabela de Requisitos Funcionais do Aplicativo

ID	Requisito	Descrição
RF1	Ver notícias do posto	O aplicativo deve exibir para o usuário as notícias do posto
RF2	Adicionar notícias ao posto	O usuário deve ser capaz de adicionar notícias ao posto
RF3	Listagem de Moradores	O aplicativo deve exibir ao usuário os moradores cadastrados no posto
RF4	Listagem de Família	O aplicativo deve exibir ao usuário as famílias cadastradas no posto
RF5	Relatório de Visitas	O aplicativo deve exibir ao usuário as visitas realizadas às famílias

RF6	Filtrar as visitas	O usuário deve ser capaz de filtrar por data inicial e data final as visitas realizadas às famílias.
RF7	Cadastro de Visitas	O usuário deve ser capaz de registrar visitas às famílias.
RF8	Registro de Casos	O usuário deve ser capaz de registrar casos de condições clínicas dos moradores.
RF9	Relatório de casos	O aplicativo deve exibir ao usuário os casos de condições clínicas dos moradores, no formato de mapa de calor.

Legenda: RF – Requisito Funcional

Fonte: autoria própria.

4.2 Prototipação

Após a aplicação das etapas de Design Thinking foi possível entender quais são as principais necessidades dos agentes comunitários e como elas poderiam ser transformadas em um aplicativo para dispositivos móveis. Com essas informações foi utilizado o Figma, para prototipar as principais telas do aplicativo.

As Figuras 2,3,4 e 5, são exemplos de como essas informações foram transformadas em protótipos de telas. A informação de moradores cadastrados nos postos e famílias pertencentes originou o protótipo da tela de moradores e famílias, representado na Figura 2, que permite que o agente selecione a forma como deseja visualizar essa informação e filtre para localizar um morador ou uma família com mais facilidade. A Figura 3 se baseou na necessidade de acompanhar e divulgar notícias sobre a unidade básica de saúde, sendo um método de prevenção e promoção de saúde para as pessoas da região.

A Figura 5 mostra a tela inicial de usuário, onde ele encontrará as funcionalidades referentes ao usuário, acessíveis a partir dos itens de menu presentes na tela, como por exemplo o item “Meu Posto”, onde a ideia é que ao ser clicado, direciona o usuário para a tela da Figura 4 onde ele encontrará as funcionalidades referentes ao seu posto.

Figura 2 - Protótipo de Tela de Famílias e moradores.



Fonte: autoria própria.

Figura 4 - Protótipo de Tela de informações do posto.

Figura 3 - Protótipo de Tela de Notícias.



Fonte: autoria própria.

Figura 5 - Protótipo de Tela de informações do usuário.



Fonte: autoria própria.



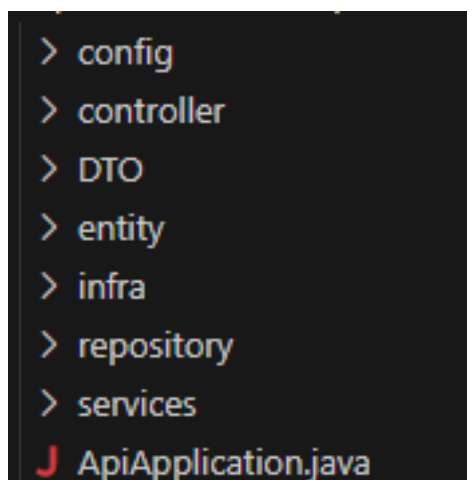
Fonte: autoria própria.

4.3 Projeto de Sistema

Os requisitos definidos serviram para pensar nas linguagens referentes ao aplicativo, onde optou-se por Flutter por oferecer um desenvolvimento multiplataforma e Spring pela simplicidade de comunicação com outras plataformas como o banco de dados e a plataforma desenvolvida pelo Flutter.

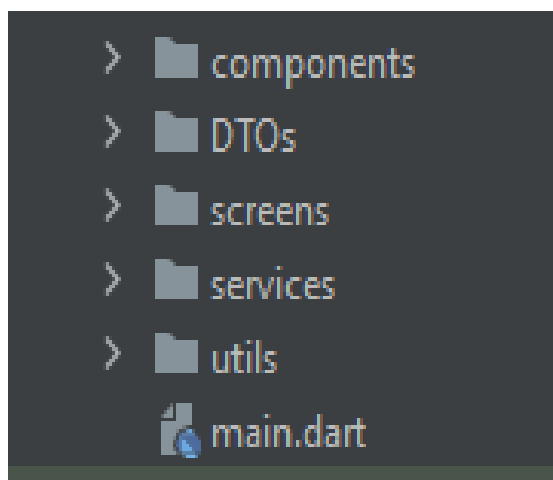
Visando uma maior praticidade em manutenções futuras no software, as classes foram organizadas em pastas de acordo com suas funcionalidades dentro do aplicativo. Essa divisão pode ser observada nas Figuras 6 e 7, com as arquiteturas do Spring e do Flutter, respectivamente, e as Tabelas 2 e 3 apresentam as descrições dessas pastas.

Figura 6 - Arquitetura do Spring



Fonte: autoria própria.

Figura 7 - Arquitetura do Flutter



Fonte: autoria própria.

Tabela 2 - Definição da Arquitetura Spring.

Pasta	Descrição
config	Contém classes de configuração da API.
controller	Contém as classes que lidam com a recepção dos métodos HTTP e retorna o resultado dessas requisições.
DTO	Contém as classes que definem um padrão de um determinado objeto para transferência entre as partes do software.
entity	Contém as classes que representam as entidades no banco de dados como Pessoas, Usuários, entre outros.
infra	Contém classes com métodos e variáveis que funcionam como utilitários dentro do aplicativo.
repository	Contém as classes que fazem operações no banco de dados,

	dependem das classes do entity para essas operações.
services	Contém as classes que realizam requisições HTTP, seguindo o caminho contrário do controller.

Fonte: autoria própria.

Tabela 3 - Definição da Arquitetura Flutter.

Pasta	Descrição
components	Contém as classes elementos que ajudam a formar as telas do aplicativo como Buttons, Texts, entre outros.
DTO	Contém as classes que definem um padrão de um determinado objeto para transferência entre as partes do software.
screens	Contém as classes que formam as telas da aplicação.
services	Contém as classes que realizam requisições HTTP, seguindo o caminho contrário do controller.
utils	Contém classes com métodos e variáveis que funcionam como utilitários dentro do aplicativo.

Fonte: autoria própria.

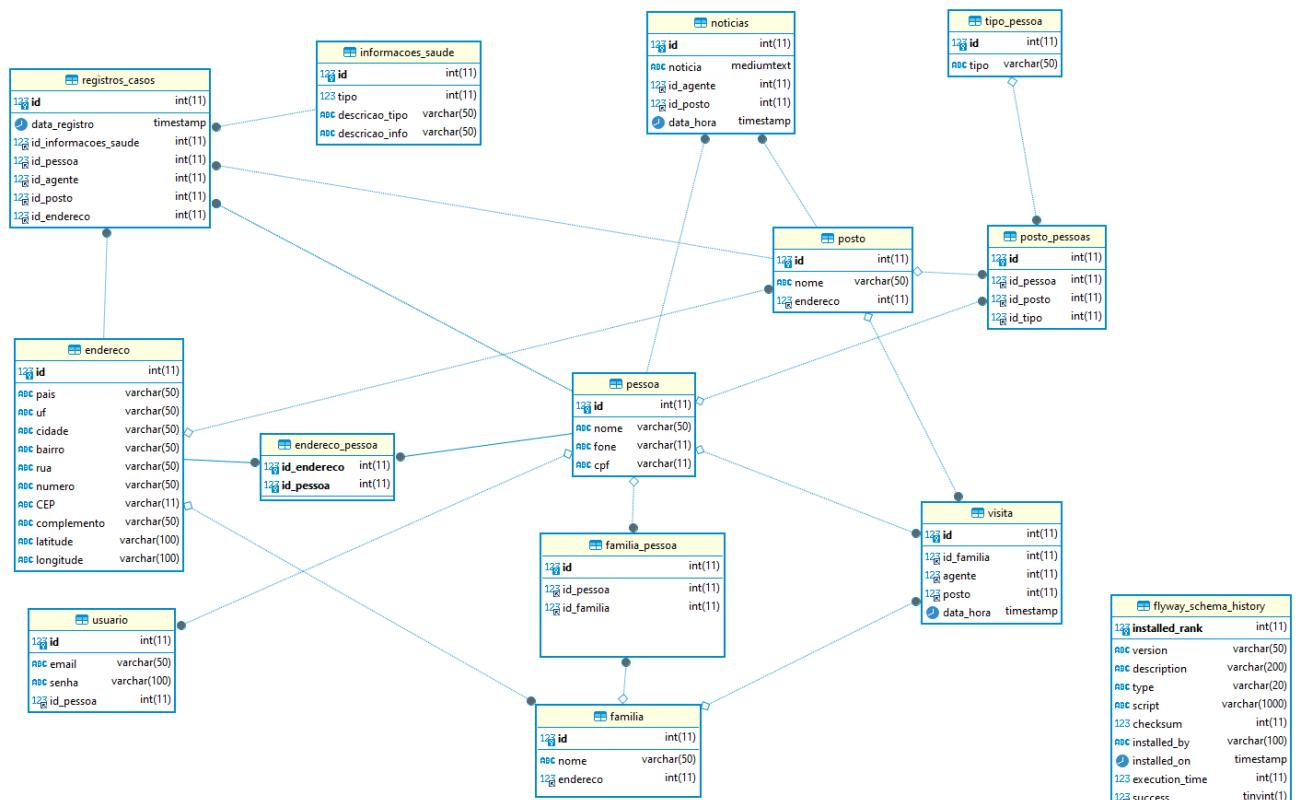
A finalização dos requisitos funcionais, permitiu também planejar a estrutura do banco de dados. Foi possível entender quais entidades seriam necessárias para cumprir os requisitos e como elas iriam se relacionar através de chaves estrangeiras ou tabelas de relacionamento. A Figura 8 representa a arquitetura desse planejamento, com a utilização de um Diagrama de Entidade-Relacionamento, que

é um fluxograma que mostra todos os elementos que compõem o banco de dados, as entidades, seus atributos e seus relacionamentos.

Essas entidades são representações dos objetos utilizados na aplicação como: Pessoas, Endereços e Posto. Sua representação no banco de dados é feita por uma tabela, que pode se relacionar a partir da utilização de outra tabela intermediária, um exemplo desse tipo de relacionamento pode ser visto pelo relacionamento das tabelas Endereco e Pessoa, elas se relacionam através da tabela endereco_pessoa, cada registro em endereco_pessoa, representa uma associação entre uma pessoa e um endereço pelos identificadores.

Outro tipo de relacionamento de tabelas é através da utilização de chaves estrangeiras, quando uma tabela possui nos seus atributos o identificador de um registro em outra tabela e pode ser visto na relação entre postos e endereços no banco de dados, a tabela Posto possui nos seus atributos o identificador da tabela Endereço portanto todo registro na tabela Posto estará associado a um Endereço.

Figura 8 - Diagrama Entidade-Relacionamento do Banco de Dados.

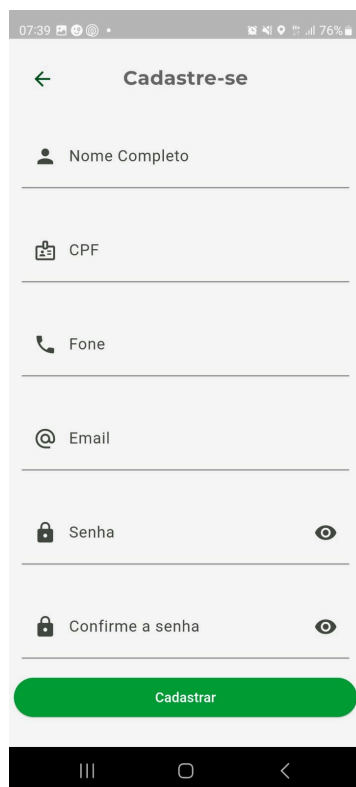


Fonte: autoria própria.

4.4 Implementação

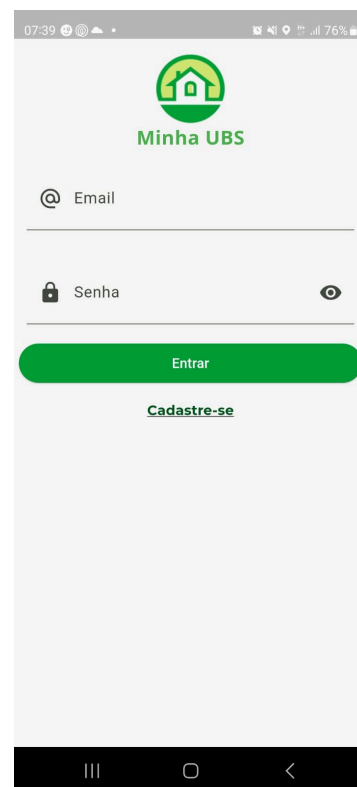
A partir da relação entre as telas do protótipo e os requisitos funcionais, o aplicativo foi desenvolvido. O fluxo de funcionalidades começa a partir do cadastro de usuário representado pela Figura 9, onde ao passar esses dados o aplicativo irá cadastrar o usuário no banco de dados, permitindo que a partir do login e senha ele possa utilizar o sistema como mostra a Figura 10.

Figura 9 - Tela de Cadastro.



Fonte: autoria própria.

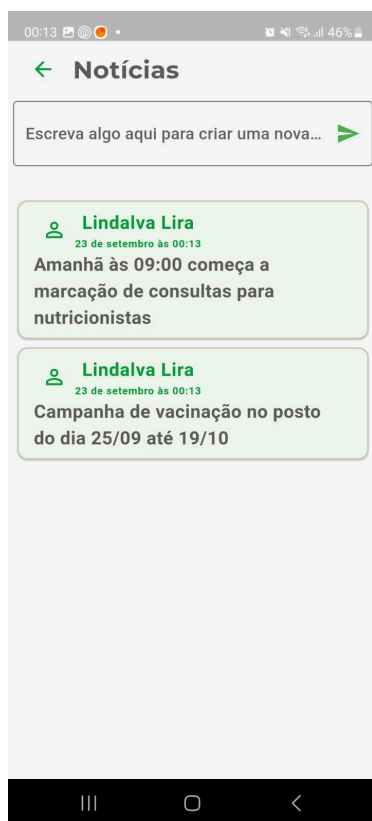
Figura 10 - Tela de Login.



Fonte: autoria própria.

O primeiro requisito RF1 e o RF2 foram atendidos com a implementação da tela apresentada na Figura 8, que permite ao usuário criar e exibir as notícias do posto que está cadastrado.

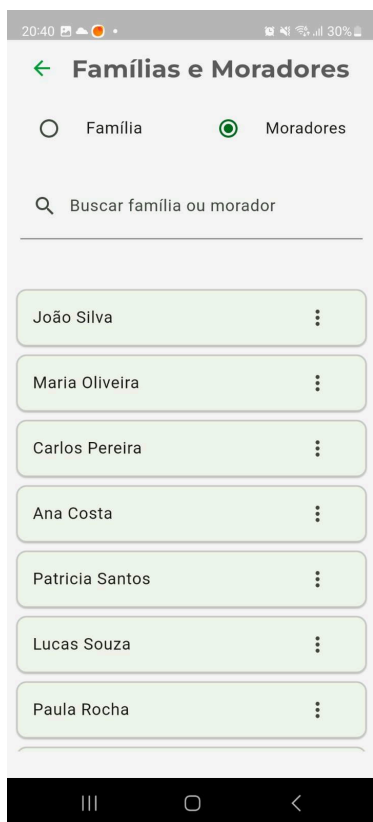
Figura 11 - Tela de Notícias.



Fonte: autoria própria.

O RF3 e o RF4 foram atendidos na implementação das telas apresentadas nas Figuras 9 e 10, respectivamente, onde o aplicativo exibe ao usuário as famílias e moradores cadastrados no posto, permitindo também ao usuário filtrar a partir do modo de visualização escolhido por ele.

Figura 12 - Tela de Listagem de Moradores.



Fonte: autoria própria.

Figura 13 - Tela de Listagem de Famílias.



Fonte: autoria própria.

Os RFs 5 e 6 foram atendidos pela implementação da tela exibida na Figura 11, onde o aplicativo exibe inicialmente ao usuário todas as visitas cadastradas por ele e permite também um filtro por períodos, informando uma data inicial e uma data final para a filtragem.

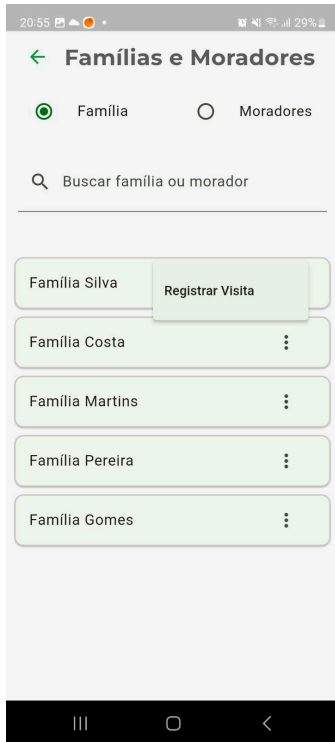
Figura 14 - Tela de Visitas.



Fonte: autoria própria.

O RF 7 foi atendido através da tela exibida na Figura 12, a partir de um menu, exibido apenas no elemento na listagem de famílias, que ao ser pressionado exibirá a opção de registrar visita, onde o usuário ao clicar verá uma caixa de diálogo para confirmação do registro de visita, como exibido na Figura 13 e ao clicar em registrar irá computar a visita na família indicada. Baseado nesse comportamento o RF 8 foi atendido, no modo de visualização de moradores, como mostrado nas Figuras 14 e 15, onde exibe a opção de registrar caso e a confirmação de registro a partir do caso escolhido pelo usuário.

Figura 15 - Menu de Registro de Visita.



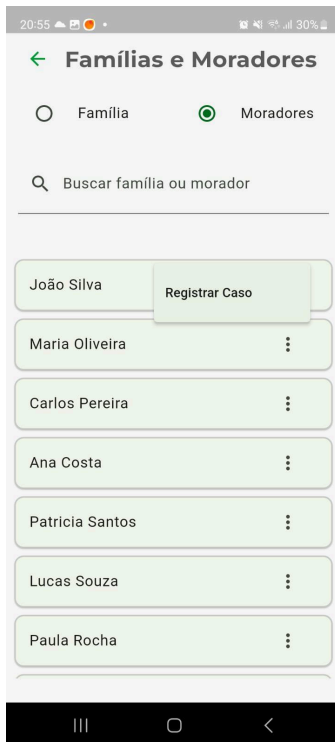
Fonte: autoria própria.

Figura 16 - Confirmar Registro de visita.



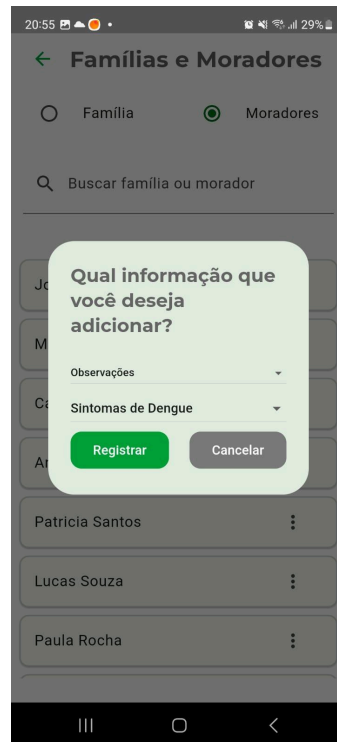
Fonte: autoria própria.

Figura 17 - Registro de Caso.



Fonte: autoria própria.

Figura 18 - Confirmar Registro de Caso.

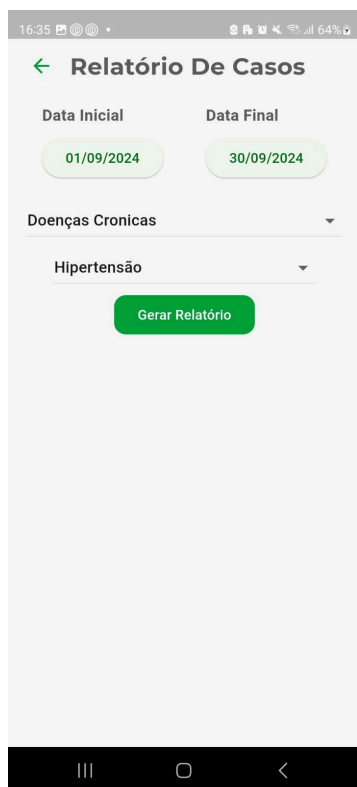


Fonte: autoria própria.

O RF9 foi implementado a partir de duas telas, representadas nas Figuras 16 e 17. A primeira é onde o usuário seleciona os tópicos para filtragem, como data inicial, data final e caso de condição de saúde. Ao clicar no botão de gerar relatório, é solicitado ao banco de dados as latitudes e longitudes que correspondam às características passadas no filtro, e utilizando a biblioteca do Flutter, flutter_map_heatmap⁹, é montado um mapa de calor indicando assim os locais e níveis de incidência dos casos localizados no período informado, no caso apresentado pelas Figuras foram utilizados 70 casos de condições clínicas, registrados pela funcionalidade de registro de casos.

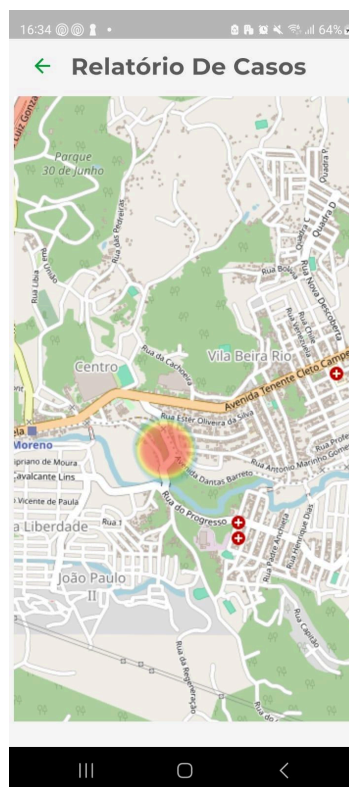
Este RF foi baseado em duas necessidades levantadas nas respostas do formulário, a necessidade de exibição de um mapa que mostrasse a região do posto e a necessidade de dados para planejamentos estratégicos de combate a casos de condições de saúde na região, como hipertensão, diabetes, obesidade e desnutrição.

Figura 19 - Filtro de Casos.



Fonte: autoria própria.

Figura 20 - Mapa de Calor dos casos.



Fonte: autoria própria.

⁹ https://pub.dev/packages/flutter_map_heatmap

4.5 Validação do público-alvo

O aplicativo foi apresentado a uma agente comunitária de saúde para observar a utilização e identificar pontos de melhoria. Durante a apresentação o usuário relatou que o aplicativo atendia ao seu objetivo, onde o relatório com mapa de calor seria de grande ajuda para localizar pontos importantes na região para planejamentos estratégicos que são discutidos em reuniões periódicas na UBS. Porém, alguns fluxos como os de registros de visita e de casos não ficaram tão claros e fáceis de localizar inicialmente e foi sugerida alguma forma de comunicação, como um link para o WhatsApp do morador, no aplicativo.

5. Conclusão

A relevância das unidades básicas de saúde, nas regiões beneficiadas por elas, é inegável e influencia toda a rede de saúde pública. Como porta de entrada das pessoas com o SUS, as unidades básicas utilizam um contato mais próximo dos residentes como forma de obter e divulgar informações importantes de prevenção e promoção de saúde, além de oferecer serviços de saúde pública, tratamentos médicos e encaminhamentos para casos mais especializados.

Por isso o aplicativo MinhaUBS surgiu com a proposta de reduzir a distância entre agentes comunitários, de saúde e de endemias, e a população através do desenvolvimento de um aplicativo. A partir da aplicação do design thinking e do modelo de processo de desenvolvimento cascata, foi possível identificar as principais necessidades do público-alvo, transformar essas necessidades em funcionalidades, prototipar as funcionalidades, desenvolver um aplicativo e validar com um usuário real.

Embora haja outros trabalhos desenvolvidos anteriormente com foco na melhoria da saúde pública a partir das unidades básicas de saúde, como o desenvolvido por Coelho (2023), esse tem como foco inicial o agente comunitário. A partir da disponibilização de ferramentas para melhoria dos planejamentos e ações dos agentes comunitários, podemos pensar em como novas ferramentas podem ser

desenvolvidas para residentes, enquanto eles são beneficiados indiretamente pelas ferramentas atuais.

O aplicativo desenvolvido pode ser visto como um ponto de partida para desenvolvimentos futuros na área, ampliando e melhorando a rede de saúde pública a partir da tecnologia. Para isso é necessária uma quantidade maior de usuários reais validando o sistema, pois um ponto de limitação deste trabalho foi a validação de apenas um agente comunitário, será possível obter um conhecimento mais amplo e preciso de como a ferramenta pode ser melhorada.

5.1 Trabalhos Futuros

Pretende-se aumentar o público de validação do aplicativo, como forma de encontrar pontos de melhoria dentro da aplicação tornando a solução mais robusta. Corrigir os pontos identificados na validação com o usuário ao adicionar um link para o WhatsApp do morador e identificar formas de melhorar a usabilidade dos fluxos.

Além disso, como perspectiva futura para esse trabalho, pretende-se aplicar novamente as ferramentas utilizadas com moradores atendidos pelo sistema público de saúde. O objetivo dessa aplicação é identificar as necessidades dos moradores em relação às unidades básicas de saúde, permitindo assim a elaboração de novas funcionalidades para um outro tipo de usuário, o morador.

Referências:

BIANCHI, Evaldo. Sistema de envio e recebimento de mensagens para plataforma android utilizando spring boot e google cloud messaging. Pato Branco 2015 Universidade Tecnológica Federal do Paraná

BRASIL. Ministério da Saúde. Manual do e-SUS Atenção Primária à Saúde. Brasília: Ministério da Saúde, 2021. Disponível em: https://saps-ms.github.io/Manual-eSUS_APS/. Acesso em: 30 jul. 2023.

BROWN, Tim. Design Thinking: uma metodologia poderosa para decretar o fim das velhas ideias. [S.l.]: Alta Books, 2020.

CEEN - Centro de Estudos de Enfermagem e Nutrição. Agente Comunitário de Saúde. 2021. Disponível em: <https://www.ceen.com.br/agente-comunitario-de-saude>. Acesso em: 22 set. 2024.

CELUPPI, Ianka Cristina; PRADO, Ricardo Luiz Camargo; HAMMES, Jades Fernando; DALMARCO, Eduardo Monguilhott; WAZLAWICK, Raul Sidnei. Aplicativo e-SUS Território: Ferramenta para a gestão do território na Atenção Primária à Saúde do Brasil. In: WORKSHOP DE FERRAMENTAS E APLICAÇÕES - SIMPÓSIO BRASILEIRO DE SISTEMAS MULTIMÍDIA E WEB (WEBMEDIA), 28. , 2022, Curitiba. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2022. p. 111-114. ISSN 2596-1683. DOI: https://doi.org/10.5753/webmedia_estendido.2022.226542. Acesso em: 23 set. 2023.

COELHO, Luciano Figueiredo. Tem No Posto De Saúde? Aplicativo Para Promover O Acesso à Informação E Uso Racional Dos Medicamentos No Município De Tijucas/SC. 2023.

CONASS - Conselho Nacional de Secretários de Saúde. Atenção primária é capaz de resolver 85% das demandas de saúde. 2019. Disponível em: <https://www.conass.org.br/atencao-primaria-e-capaz-de-resolver-85-das-demandas-de-saude/>. Acesso em: 22 set. 2024.

COSTA, E. C. Unidade Básica de Saúde: o ponto de partida para uma vida mais saudável. 2023. Disponível em: <https://saudelab.com/unidade-basica-de-saude/>. Acesso em: 25 set. 2024.

DART. 2021. Dart overview. Disponível em: <https://dart.dev/overview>. Acesso em: 22 set. 2024.

DEVELOPERS, Android. 2018. Conheça o Android Studio. Disponível em: <https://developer.android.com/studio/intro?hl=pt-br>. Acesso em: 22 set. 2024.

Diário Gaúcho. Postos de saúde da Capital usam WhatsApp para facilitar a comunicação com usuários. 2021. Disponível em: <http://diariogaucho.clicrbs.com.br/rs/dia-a-dia/noticia/2021/10/postos-de-saude-da-capital-usam-whatsapp-para-facilitar-a-comunicacao-com-usuarios-21268455.html>. Acesso em: 22 set. 2024.

DORIGHELLO, Renato Sellaro et al. CONSTRUÇÃO DE APLICATIVO PARA AUXÍLIO NO ENFRENTAMENTO À PANDEMIA DA COVID-19. Conhecimento Interativo, v. 16, n. 1, 2022.

EBAC. 2023. O que é Figma e como usar?. Disponível em : <https://ebaonline.com.br/blog/o-que-e-figma-e-como-usar>. Acesso em: 22 set. 2024.

IDEO. IDEO Design Thinking. 2022. Disponível em: <https://designthinking.ideo.com/>. Acesso em: 18/09/2024.

MARTIN, R. C. Arquitetura Limpa: O guia do artesão para estrutura e design de software. Alta Books, 2019.

MERRIAM, S. B. Qualitative research: A guide to design and implementation. Chichester, England: John Wiley & Sons, 2009.

MICROSOFT. 2018. Visual Studio Code. Disponível em: <https://visualstudio.microsoft.com/pt-br/#vscode-section>. Acesso em: 22 set. 2024.

MICROSOFT. 2023. O que é o Java Spring Boot? Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-java-spring-boot>. Acesso em: 22 set. 2024.

MICROSOFT. 2024. Introdução ao Visual Studio Code. Disponível em: <https://learn.microsoft.com/pt-br/training/modules/introduction-to-visual-studio-code>. Acesso em: 12 out. 2024.

Ministério da Saúde. Meu SUS Digital. 2023. Disponível em: <https://www.gov.br/pt-br/servicos/acessar-a-plataforma-movel-de-servicos-digitais-do-ministerio-da-saude>. Acesso em: 11 out. 2024.

Ministério da Saúde. O Agente Comunitário de Saúde no controle da dengue. 2009. Disponível em: https://bvsmis.saude.gov.br/bvs/publicacoes/agente_comunitario_saude_dengue.pdf. Acesso em: 22 set. 2024.

Ministério da Saúde. SUS. 2021. Disponível em: <https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/s/sus>. Acesso em: 22 set. 2024.

OLIVEIRA, Klebson Alberto De. Concepção de um aplicativo móvel para o gerenciamento de vacinas. 2021.

PINHEIRO, Fagner. Flutter: Widgets Stateful e Stateless. 2020. Disponível em: <https://www.treinaweb.com.br/blog/flutter-widgets-stateful-e-stateless>. Acesso em: 12 out. 2024.

PRESSMAN, R.; MAXIM, B. Software Engineering: A Practitioner's Approach. 9. ed. Columbus, OH, USA: McGraw-Hill Education, 2019.

MELO, Rubens de. Flutter para iniciantes. 2019. Disponível em: <https://flutterparainiciantes.com.br/widgets/>. Acesso em 12 out. 2024.

SAMPAIO, Juliana. Spring Boot: como começar. 2021. Disponível em: <https://zup.com.br/blog/spring-boot>. Acesso em: 12 out. 2024.

SOMMERVILLE, Ian. Engenharia de Software. 9.ed. São Paulo: Pearson Prentice Hall, 2011.

STRAUSS, Anselm; CORBIN, Juliet. Pesquisa qualitativa : técnicas e procedimentos para o desenvolvimento de teoria fundamentada. 2.ed. Artmed, 2008.

SOUZA, F. M. DE L. C. et al. Desenvolvimento de aplicativo móvel para o acompanhamento pré-natal e validação de conteúdo. Acta Paulista de Enfermagem, v. 35, 2022.

UNA-SUS. Maior sistema público de saúde do mundo, SUS completa 31 anos. 2021. Disponível em: <https://www.unasus.gov.br/noticia/maior-sistema-publico-de-saude-do-mundo-sus-completa-31-anos>. Acesso em: 22 set. 2024.

VALENTE, Marco Tulio. Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade. Editora: Independente, 2020.