

Aplicativo Móvel para Controle de Vacinação para Setores Públicos e Privados

Gabriel Toller Marconatto¹, Gustavo Stangherlin Cantarelli¹

¹Curso de Sistemas de Informação – Universidade Franciscana
CEP 97010-032 – Santa Maria – RS – Brasil

gabrielmarconatto17@gmail.com, gus.cant@gmail.com

Abstract. *This work presents the solution to computerize the vaccination card in both the public and private service, so that both the health professional and the patient have access to the vaccination card in a mobile application and to reduce the work of professionals who store the information of each patient's vaccination card. To develop this application, the Feature Drive Development (FDD) methodology was used to develop the construction processes, using the Dart language, the Flutter framework and, as a database management system, PostgreSQL.*

Resumo. *Este trabalho apresenta a solução de computadorizar a carteira de vacinação tanto no serviço público e privado, a fim de tanto o profissional da saúde quanto o paciente ter acesso à carteira de vacinação em um aplicativo móvel e diminuir o trabalho dos profissionais que armazenam as informações da carteira de vacinação de cada paciente. Para conseguir desenvolver este aplicativo, foi utilizada a metodologia Feature Drive Development (FDD) para o desenvolvimento dos processos de construção, com o uso da linguagem Dart, do framework Flutter e, como sistema gerenciador de banco de dados, o PostgreSQL.*

1. Introdução

As vacinas são de grande importância para a sobrevivência da humanidade, dado que evitam milhares de mortes e aumentam a expectativa de vida, sendo um investimento de excelente custo e efetividade [Mizuta *et al.* 2019]. No Brasil, é utilizado o Programa Nacional de Imunizações (PNI) para a imunização na rede pública, sendo este amplamente respeitado pela população e a comunidade científica, dado que, efetivamente, cumpre com sua função. Entretanto, apesar de serem pouco numerosos, os grupos de recusa vacinal podem colocar em risco o sucesso já obtido pelo PNI. Tais grupos induzem atitudes que colocam em risco todos à sua volta, já que epidemias de sarampo, coqueluche e varicela já foram associadas a essas atitudes.

A carteira de vacinação tem como finalidade anotar as vacinas tomadas pelo indivíduo. Graças a ela, pode-se saber quais as últimas imunizações feitas e quais devem ser aplicadas no futuro, sendo, com isso, um ótimo indicador para a valorização pessoal e importância das vacinas [Mizuta *et al.* 2019].

1.1. Justificativa

Atualmente, a logística brasileira de vacinação utiliza-se de papel para o controle da carteira de vacinação, dificultando o acesso aos dados, tanto do profissional de saúde quanto do cidadão.

Verificou-se que já ocorreram implementações do uso de tecnologia nesta área, ao procurar programas para este problema, foi descoberto o programa ConnectSus que atualmente possui foco somente na Covid-19.

Com isso, será proposto o desenvolvimento de uma aplicação móvel com sistema operacional Android para a Instituição Lauduz, a qual está auxiliando na luta contra a SARS-CoV-2. A Lauduz é uma entidade brasileira que tem como objetivo levar cuidados de saúde de forma remota e on-line às pessoas em vulnerabilidade social tanto na forma privada quanto pública.

1.2. Objetivo geral

Este trabalho visa o desenvolvimento de uma aplicação com plataforma Android a fim de substituir o uso do papel no manuseio da carteira de vacinação. Tanto o profissional de saúde possuirá os dados do cidadão de forma rápida, podendo vaciná-lo rapidamente, quanto o cidadão sempre saberá quando deve se vacinar.

1.3. Objetivos específicos

Neste projeto, os objetivos específicos são:

- Estudar e utilizar o *framework* Flutter, assim como sua linguagem Dart;
- Estudar e utilizar o PostgreSQL;
- Estudar e utilizar a metodologia *Feature Driven Development* (FDD) para o trabalho;
- Criar uma interface amigável e simples para o usuário.

2. Referencial Teórico

Nesta seção, serão abordados todos os conceitos e tópicos necessários para o desenvolvimento deste trabalho. Sendo eles, os conceitos da vacinação que englobam o DATASUS e uma integração de banco de dados, além do *framework* Flutter com a linguagem Dart, o Virtual Studio Code como editor de código e o PostgreSQL como gerenciador de banco de dados.

2.1. Tecnologia aplicada à saúde

A popularização dos smartphones tem sido considerada por muitos a revolução tecnológica de maior impacto nos últimos tempos. Essa popularização gerou oportunidades comerciais e sociais em diversas áreas, ou seja, desenvolver soluções computacionais para aplicativos móveis é um meio eficaz de disponibilizar a ferramenta para um público-alvo desejado.

Fazendo um estudo voltado para a área de saúde, conclui-se que os aplicativos móveis ajudaram na tomada de decisão diagnóstica dos profissionais de saúde, facilitaram o armazenamento de informações e o acesso do profissional de saúde e o paciente às condições dele [Tibes 2014].

2.2. Vacinação

A vacina tem um grande impacto na saúde, dado que evita milhões de mortes por ano e aumenta a expectativa de vida. À medida que se aumenta o número de vacinas disponíveis, tanto no setor privado quanto no público, aumenta também a quantidade de pessoas e grupos que se preocupam com a segurança e a necessidade da vacinação.

Saber as últimas vacinas recebidas e dispor da carteira de vacinação são bons indicativos da valorização da proteção pessoal e do controle das imunizações, ou seja, possuir um bom controle da carteira de vacinação são ótimos indícios para a vacinação no país [Mizuta *et al.* 2019].

2.3. DATASUS

Desde que o Sistema Único de Saúde (SUS) foi implementado, foram criadas várias ferramentas para que a saúde pública faça o levantamento de dados dos vários sistemas de informações. O surgimento da principal ferramenta ocorreu com a criação do DATASUS. Sendo o responsável por disponibilizar as informações, seguindo política superior, foi considerado, pela Organização Mundial da Saúde (OMS) e Organização Pan-Americana de Saúde (OPAS), um dos mais completos do mundo.

Com o desenvolvimento e a melhor estruturação dos sistemas, o DATASUS criou produtos e serviços que fornecem dados e ferramentas de trabalho ao Ministério da Saúde (MS). Ele também passou a reorientar seus serviços, fazendo com que empresas e convênios do setor público ou privado possam participar [Lima *et al.* 2015].

Atualmente, no Brasil, criou-se uma ferramenta chamada eSUS APS, que é uma estratégia do Departamento de Saúde da Família e da Secretaria de Atenção Primária à Saúde para estruturar as informações de saúde da Atenção Primária à Saúde (APS) e possibilitar ao cidadão o acesso às informações e uso de portuário eletrônico. Os pacientes podem agendar consultas de forma on-line, vinculando esta agenda pelo aplicativo “Conecte SUS Cidadão”, permitindo com que a população acesse informações pessoais e clínicas disponíveis no banco de dados [Celuppi *et al.* 2021].

O aplicativo Conecte SUS inicial está sendo usado apenas para vacinas referentes ao vírus SARS-CoV-2.

2.4. Flutter

O Flutter é um *framework* construído pela Google para o desenvolvimento móvel multiplataforma. Ele faz com que cada componente nativo seja implementado pelo respectivo *framework* e apresentado ao usuário por um motor de renderização próprio.

O Flutter utiliza a linguagem Dart, que também foi criada pela Google, e é semelhante ao JavaScript [Payne 2019].

2.5. Visual Studio Code

O Visual Studio Code é um editor de código criado em 2015 pela Microsoft para o desenvolvimento de aplicações web. Trata-se de uma ferramenta leve e multiplataforma que atende a vários projetos, como ASP.NET e Node.js, além de ser *open source*,

permitindo que a comunidade utilize seu desenvolvimento para criação de extensões e novas funcionalidades [Dollard 2008].

2.6. PostgreSQL

O PostgreSQL é um sistema de gerenciamento de bancos de dados objeto-relacional de uso geral, que foi projetado para funcionar em plataformas semelhantes ao Unix, mas também foi projetado para ser portátil. Sendo de código aberto e bastante estável, ele é recomendado pelo seu baixo custo de desenvolvimento em comparação a outros sistemas de gerenciamento de banco de dados [Milani 2008].

3. Trabalhos Relacionados

Os três trabalhos apresentados na sequência demonstram o desenvolvimento de um software sobre a carteira de vacinação.

3.1. Carteira de vacinação visual

O trabalho realizado por Muratt *et al.* (2016) tem como principal foco o desenvolvimento de uma plataforma web que gera um cartão inteligente de vacinação, auxiliando no acesso às informações e integridade dos dados.

O trabalho utiliza, no desenvolvimento, a linguagem Java, JavaScript, HTML e CSS em conjunto com o *framework* Hibernate, e, como sistema de gerenciamento de banco de dados, foi utilizado o PostgreSQL.

Com a substituição da carteira de vacinação manual para virtual, notou-se o acesso ágil na busca e no fornecimento de informações consistentes.

3.2. O controle das consultas odontológicas dos bebês por meio da carteira de vacina: avaliação de um programa-piloto desenvolvido na Estratégia Saúde da Família em Ponta Grossa

O trabalho produzido por Stocco e Baldani (2011) utiliza um programa-piloto para controlar a frequência às consultas odontológicas dos bebês, a fim de monitorar sua efetividade neste uso.

Após uma extensa análise, utilizando o programa para saber se a cárie nos bebês é influenciada pelo retorno odontológico, concluiu-se que ela influencia. Ou seja, o acompanhamento odontológico via carteira de vacinação é uma solução viável e de baixo custo, mas somente ocorrendo com o comprometimento profissional da comunidade.

3.3. Desenvolvimento de um software para controle nominal do calendário de vacinação pela rede pública de saúde

O trabalho concluído por Lantin, Shama e Bez (2020) focou na criação de um software com uma interface minimalista utilizando Excel como banco de dados. Foi feito um software para controle nominal do calendário vacinal da população para o setor público.

Por questões sociais, culturais e más condições de saúde, muitas pessoas não se vacinam. Esses fatores mostram que o uso de tecnologia para resolver esse problema é bastante promissor, visto que, com ela, é possível enviar lembrete ao usuário, fazendo com que haja maior retorno na vacinação da população.

Na Figura 1, apresenta-se a interface do software desenvolvido por Lantin, Shama e Bez.

CONTROLE DE VACINAÇÃO

Cadastro do Paciente | Vacinas A - H | Vacinas M - T | Ajuda | Desenvolvedor

NOME DO PACIENTE (sem acento)

DATA NASCIMENTO IDADE

ENDEREÇO N°

WHATSAPP (xx) - 912345678 TEL. FIXO (xx) - 12345678

E-MAIL

NOME DA MÃE (sem acento)

AG. SAÚDE RESPONSÁVEL

ÁREA DA USF MICRO ÁREA

Mudou de bairro

PROCURAR PACIENTE SALVAR LIMPAR

ATRASADOS ESTATÍSTICAS SAIR

Figura 1. Interface do software de Lantin, Shama e Bez

3.4. Conclusão sobre os trabalhos relacionados

Os trabalhos relacionados anteriormente possuem a finalidade de construir um software com foco na vacinação ou analisar um programa.

O trabalho desenvolvido por Muratt *et al.* (2016) se propôs a desenvolver um sistema web para substituir a carteira de vacinação. O projeto é semelhante com este devido ao objetivo dele, além de utilizar o mesmo SGBD. Sua interface é moderna e será uma boa referência para realizar a métrica deste projeto.

Já Stocco e Baldani (2011) focaram em um software pronto e utilizam-se deste para analisar os dados obtidos ao comparar a frequência ao retorno dos pais que utilizaram o método tradicional e os que utilizaram o software. Esse projeto foi importante para saber os resultados que um acadêmico da área da saúde obteve ao utilizar uma tecnologia semelhante com o presente projeto.

E Lantin, Shama e Bez (2020) criaram um software para controle do calendário vacinal da população. Esse projeto apresenta informações rasas sobre o seu uso de tecnologia, mas o objetivo é parecido com o presente projeto e seu foco no setor público chama a atenção, visto que ele foca em um valor mais social e cultural. Sua interface legada é um problema para o usuário, principalmente se considerar seu ano de desenvolvimento, considerando que este projeto utiliza uma ferramenta moderna sendo ela o Flutter, a interface moderna será um diferencial.

Os três trabalhos são valiosos para a ideia atual deste projeto, ao qual visa ajudar a população utilizando-se de sistemas simples.

4. Metodologia

Assim como o Scrum faz parte dos processos ágeis originais, o FDD “*Feature Driven Development*” é um deles.

Foi desenvolvido por Peter Coad e Jeff De Luca e publicado em 1999 no livro “*Java Modeling in Color with UML*”. Todavia, uma versão completa da metodologia foi publicada apenas em 2002 após sua utilização em uma agência do *United Overseas Bank* em Singapura [Reis 2016].

Ele incorpora boas práticas de desenvolvimento já reconhecidas pela indústria de software. Práticas estas que são orientadas a funcionalidades, o qual é um conceito de valor do ponto de vista do cliente. O FDD tem como foco entregar uma parte de software tangível e funcional para o cliente em espaços de tempo regulares [Reis 2016].

As práticas do FDD não são extremamente rígidas, sendo recomendadas dependendo do ambiente de desenvolvimento. No entanto, existe um conjunto de práticas fundamentais que definem o FDD. A Figura 2 apresenta os processos e seu ciclo de construção.

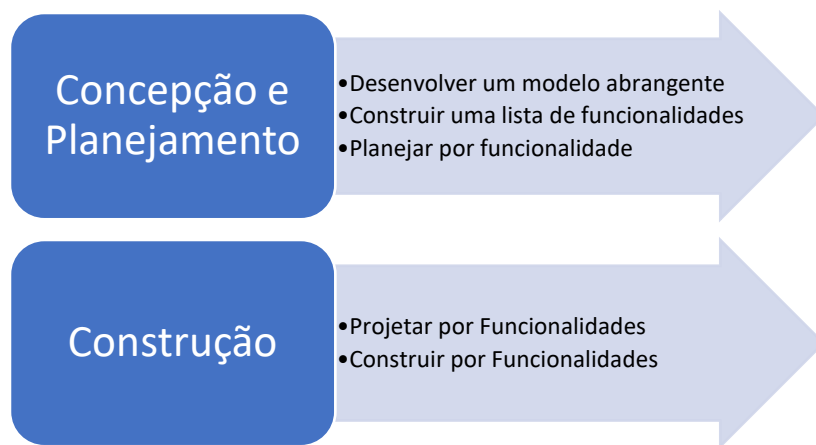


Figura 2. Representação do ciclo do FDD

4.1. Desenvolver modelo abrangente

Esta é a fase que abrange todo o escopo do sistema e deve ser realizada por membros do domínio de negócio juntamente com os desenvolvedores. Após ser realizado um estudo sobre o escopo de sistema, foca-se em um estudo de melhor detalhamento sobre o domínio do negócio para cada área a ser modelada [Reis 2016].

Na Figura 3, é apresentado o Diagrama de Domínio inicial do projeto que representa uma visão de alto nível do projeto, mostrando seus conceitos e dependências.

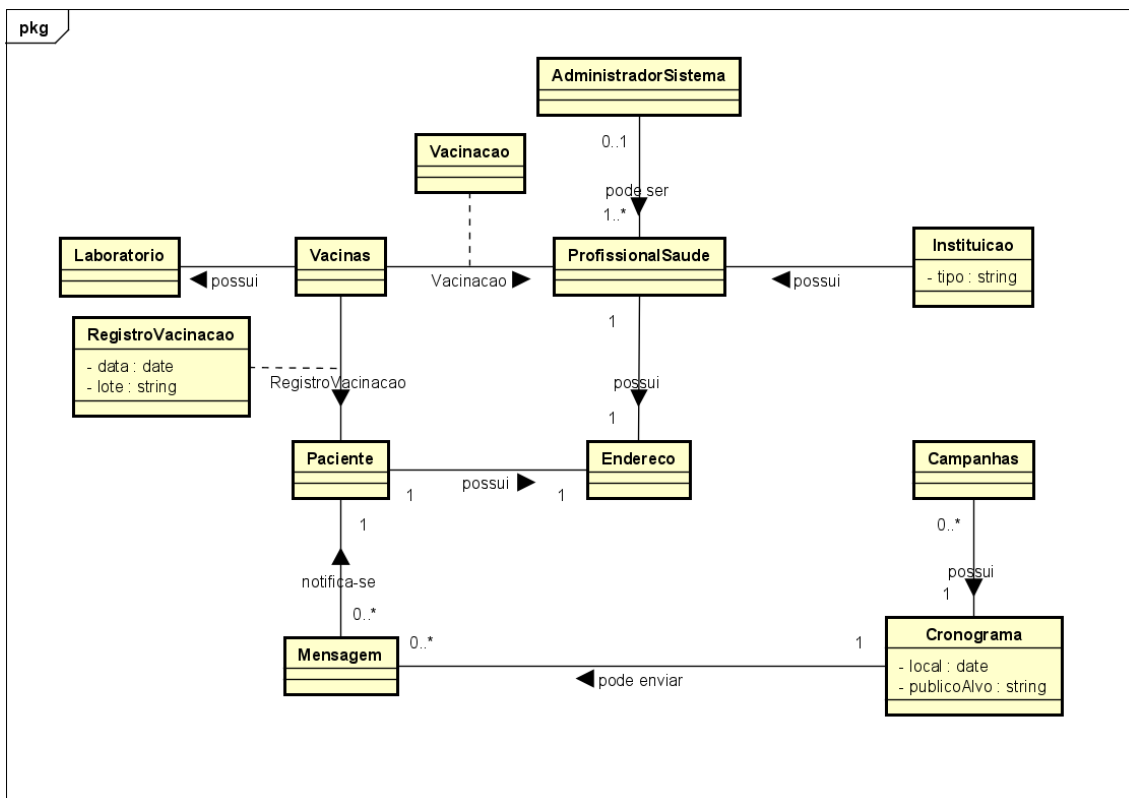


Figura 3. Diagrama de Domínio

4.2 Detalhar por Funcionalidade

É a segunda fase do FDD, que busca reconhecer as diversas funcionalidades que satisfazem os requisitos do sistema [Reis 2016]. Na Tabela 1, são apresentados os requisitos funcionais (RF) e não funcionais do sistema (RNF).

Tabela 1. Requisitos Funcionais e Não Funcionais do Sistema

RF.1 Gerenciar Paciente
Descrição: O aplicativo deve permitir as funções de cadastrar, consultar e alterar os pacientes se o tipo de usuário for um profissional da saúde.
Complexidade: baixa.
RF.1.1 Validar Paciente
Descrição: O sistema deverá validar o cadastro do paciente de acordo com as informações recebidas (CPF, idade, condição especial, entre outros).
Complexidade: média.
RF.2 Gerenciar Profissional da Saúde
Descrição: O aplicativo deve permitir as funções de cadastrar, consultar e alterar os profissionais de saúde se o tipo de usuário for um administrador.
Complexidade: baixa.
RF.2.1 Validar Profissional da Saúde
Descrição: O sistema deverá validar o cadastro do usuário de acordo com as informações recebidas (CPF, idade, entre outros).
Complexidade: média.
RF.3 Gerenciar Administrador

Descrição: O aplicativo deve permitir as funções de cadastrar, consultar, alterar e excluir os administradores de acordo com o tipo de usuário.
Complexidade: baixa.
RF.4 Gerenciar Vacina
Descrição: O aplicativo deve permitir as funções de inserir, consultar, alterar e excluir as vacinas se o tipo de usuário for um profissional da saúde.
Complexidade: média.
RF.5 Gerenciar Laboratório
Descrição: O aplicativo deve permitir as funções de inserir, consultar, alterar e excluir os laboratórios onde foram criadas as vacinas se o tipo de usuário for um profissional da saúde.
Complexidade: baixa.
RF.6 Gerenciar Vacinas Aplicadas
Descrição: O aplicativo deve permitir as funções de consultar, alterar e excluir as vacinas que já foram aplicadas se o tipo de usuário for um profissional da saúde.
Complexidade: média.
RF.6.1 Gerenciar Faixa Etária das Vacinas
Descrição: O aplicativo deve permitir as funções de alterar as vacinas de acordo com as doses, a faixa etária do usuário, e da condição especial do usuário se o tipo de usuário for um profissional da saúde.
Complexidade: alta.
RF.7 Gerenciar Campanha
Descrição: O aplicativo deve permitir as funções de inserir, consultar, alterar e excluir as campanhas se o tipo de usuário for um profissional da saúde.
Complexidade: média.
RF.7.1 Gerenciar Cronograma da Campanha
Descrição: O aplicativo deve permitir inserir, consultar, alterar e excluir o cronograma e o público-alvo da campanha.
Complexidade: baixa.
RF.8 Gerenciar Mensagem
Descrição: O aplicativo deve informar ao paciente informações como a data, local e público-alvo da campanha e quais vacinas ele precisa tomar.
Complexidade: média.
Requisitos Não Funcionais
RNF.1 A aplicação será desenvolvida para a plataforma Android.
RNF.2 Para o desenvolvimento será utilizado Flutter e linguagem Dart.
RNF.3 Os dados do software serão armazenados na nuvem, através do banco de dados PostgreSQL.

4.3 Detalhamento por Funcionalidade

Esta fase tem como objetivo mostrar o fluxo de execução do sistema, quais as atividades que podem ser realizadas e como funciona o seu tratamento [Reis 2016].

A Figura 4 apresenta o Diagrama de Atividades que mostra as ações que um profissional da saúde pode realizar dentro do sistema.

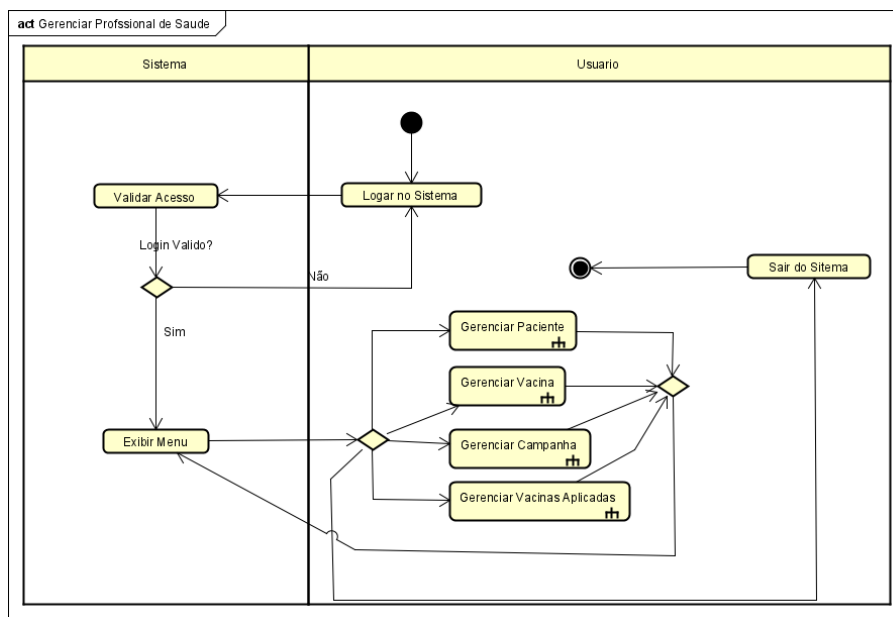


Figura 4. Diagrama de Atividades do Gerenciar Profissional de Saúde

Já Figura 5 apresenta o Diagrama de Atividades que demonstra as ações que um profissional da saúde pode realizar dentro do sistema para o gerenciamento das vacinas.

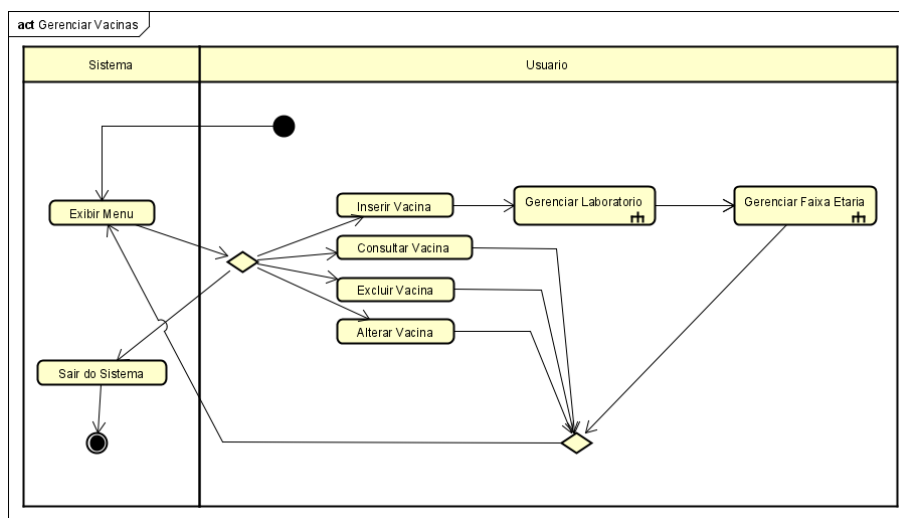


Figura 5. Diagrama de Atividades do Gerenciar Vacina

4.4 Construir por funcionalidade

Nesta etapa, foi desenvolvido o aplicativo, onde foram elaboradas as codificações utilizando as tecnologias propostas. Foram selecionados alguns métodos mais relevantes para a criação desta aplicação.

Para realizar a conexão com o banco PostgreSQL, foi utilizada a função *initDatabase*, mostrada na Figura 6.

```

class DatabaseHelper {
    // make this a singleton class
    DatabaseHelper._privateConstructor();
    static final DatabaseHelper instance = DatabaseHelper._privateConstructor();

    //instanciando o objeto para ele realizar a conexão com o banco
    static PostgreSQLConnection _database;
    Future<PostgreSQLConnection> get database async {
        _database = await initDatabase();
        return _database;
    }

    // Realizando a conexão com o banco
    Future<PostgreSQLConnection> initDatabase() async {
        var conn = PostgreSQLConnection('pgsql.guscant.kinghost.net', 5432, 'vacinadb',
        | | username: 'guscant3', password: 'tfggabriel');
        await conn.open();
        return conn;
    }
}

```

Figura 6. Trecho do código que realiza a conexão com o banco

Foi criada uma variável *instance* do tipo *DatabaseHelper* que inicializa a função *database* que retorna a conexão com o banco de dados. Para isso, é necessário inserir o IP e a porta, também o nome do banco junto ao nome de usuário e a senha. Na Figura 7, é mostrado um exemplo de uma função que utiliza o PostgreSQL.

```

Future<String> getLaboratoriaPorId(int laboratorioId) async {
    PostgreSQLConnection db = await instance.database;
    var results = await db.query('SELECT NOME FROM LABORATORIO WHERE LAB_ID = ' +
    | | laboratorioId.toString());
    String resultado = "";
    for (var row in results) {
        resultado = row[0].toString();
    }
    await db.close();
    return resultado;
}

```

Figura 7. Trecho do código que utiliza o select do banco

A função *instance.database* faz a conexão com o banco de dados e, através da função *query*, é possível utilizar a linguagem do SGBD. Esta função tem a possibilidade de retornar valores que são usados na aplicação.

```

? ListView.builder(
  itemCount: _foundUsers.length,
  // front-end
  itemBuilder: (context, index) => Card(
    key: ValueKey(_foundUsers[index]["id"]),
    color: Color.fromARGB(255, 130, 205, 255),
    elevation: 4,
    margin: EdgeInsets.symmetric(vertical: 10),
    child: ListTile(
      //continuação trabr
      onTap: () {
        Navigator.push(
          context,
          new MaterialPageRoute(
            builder: (context) =>
              new PacienteInformacaoProMedico(
                pacienteId: int.parse(
                  _foundUsers[index]["id"])); //
          ),
        leading: Icon(
          Icons.account_circle,
          color: Colors.black,
        ), // Icon
        title: Text(_foundUsers[index]['name']),
        subtitle: Text('${_foundUsers[index]["email"]}'),
      ), // ListTile

```

Figura 8. Trecho do código para criação de fichas de usuário

A função representada na Figura 8, se trata da mesma linguagem que nas Figuras 7 e 6, mas está é utilizado o framework Flutter o que causa esta formatação diferente das demais, esta função cria um menu para o profissional da saúde ter uma visualização dos pacientes e escolher um deles para a inserção da vacina, o parâmetro denominado *_foundUsers* apresenta uma lista com todos os pacientes e, caso o médico procure pelo nome de um paciente específico, a lista será alterada.

5. Resultados

O desenvolvimento da aplicação abrangeu a visualização da carteira de vacinação em um celular para o usuário e para o profissional da saúde a inserção de vacinas, visando o benefício deles. Para a validação da aplicação, foram feitos testes com um emulador de dispositivo chamado Pixel 2 utilizando o Android 11.0, disponível no Android Studio.

Foram selecionadas as interfaces de maior importância para o projeto, sendo a maioria visível para os Profissionais de Saúde e outras para os pacientes.

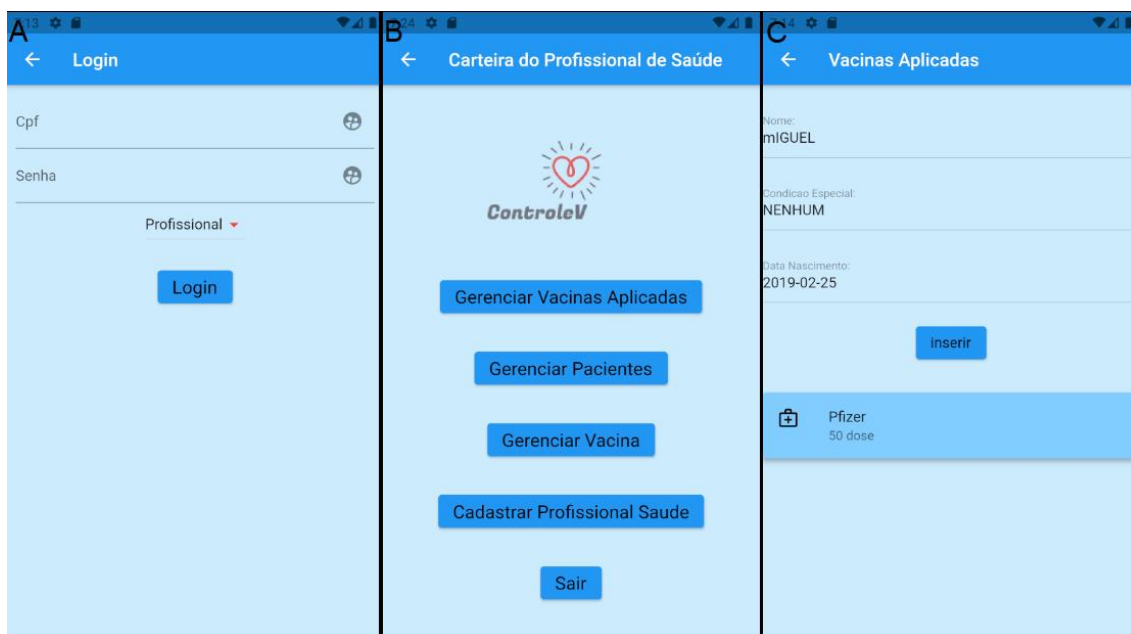


Figura 9. Interfaces de Login, menu principal e vacinas aplicadas

Na Figura 9A, é apresentada a interface da aplicação para a realização do login. Existem dois tipos de usuários, sendo eles “Profissional” ou “Paciente”, que podem ser escolhidos pela *ComboBox*. Caso o login seja efetuado com sucesso, o usuário do tipo “Profissional” irá para a interface mostrada na Figura 9B.

A Figura 9B apresenta o menu do profissional de saúde que foi logado com sucesso. Caso o profissional de saúde seja um administrador, ele poderá cadastrar outros profissionais de saúde; todavia, caso não seja, o botão de cadastrar estará desabilitado em uma coloração cinza. O cadastro do paciente é realizado pelo profissional da saúde, bem como a inserção de novas vacinas.

O profissional da saúde seleciona a opção “Gerenciar Vacinas Aplicadas” e escolhe o paciente pelo nome. Após, ele será direcionado para interface mostrada na Figura 9C, onde se apresentam algumas informações do paciente e a possibilidade de inserir as vacinas aplicadas naquele paciente, e a possibilidade de alterar ou excluir algumas das vacinas.

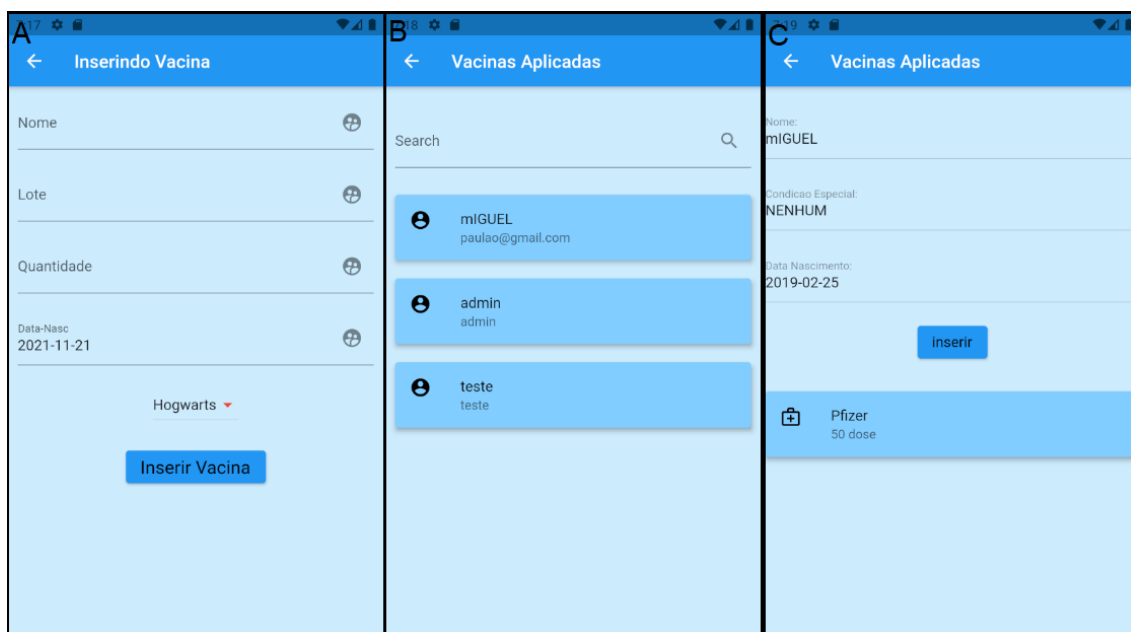


Figura 10. Interfaces de inserção de vacina, escolha do paciente nas vacinas aplicadas e a interface do paciente.

A Figura 10A apresenta a visualização quando o profissional de saúde seleciona a opção de “Gerenciar Vacina”. Por meio dessa interface, será possível inserir uma nova vacina que estará disponível para ser utilizada ao inserir uma vacina aplicada no paciente, como demonstrado na Figura 9C.

Na Figura 10B, é apresentada a interface que contém a função de escolher o paciente visando a aplicação da vacina.

Ao acessar o sistema como paciente, o usuário poderá olhar todas as suas vacinas inseridas pelo profissional de saúde e poderá inserir novas vacinas de sua escolha. Entretanto, ressaltando que o usuário não poderá remover ou alterar qualquer vacina inserida pelo profissional da saúde, somente conseguirá fazê-lo com as vacinas inseridas por ele.

6. Conclusão

Este trabalho mostrou o desenvolvimento e implementação de uma aplicação móvel no uso da carteira de vacinação, tanto no setor privado quanto no público, para o indivíduo verificar suas vacinas ou o profissional da saúde inserir as vacinas dos pacientes.

Através das pesquisas relacionadas, foram localizadas algumas aplicações para essas finalidades, porém não foi encontrada alguma para os setores públicos e privados e que possua uma interface moderna e seja comum aos dois setores. O trabalho correlato realizado por Muratt *et al.* (2016) mostrou um sistema que serviu como inspiração pela sua interface e tecnologias utilizadas, principalmente a conexão com o mesmo SGBD PostgreSQL.

A implementação desta aplicação digitalizou as informações oriundas da carteira de vacinação, tanto para o paciente quanto para o profissional de saúde, visando um melhor atendimento na área da saúde para o bem da comunidade. No futuro, pretende-se demonstrar esta aplicação para os profissionais da saúde que trabalham na Lauduz, o

recebendo seu retorno para saber se o software correspondeu às expectativas para sua validação.

Para o desenvolvimento, optou-se pelo uso de boas práticas da metodologia ágil FDD, dado que esta suporta mudanças em qualquer etapa do projeto. Foram inclusos os principais diagramas do FDD: domínio e classes apenas desta metodologia, representando os processos, arquiteturas e requisitos do aplicativo.

Para trabalhos futuros, é sugerido que seja implementado o gerenciamento de campanha, que se refere às campanhas utilizadas pelo país para combater determinado vírus tendo público-alvo ou não. Sugere-se, também, a realização de uma visualização estatística para saber os números de usuários que forem vacinados em uma determinada região. E a utilização da criptografia Hash-based Message Authentication Code (HMAC) que é uma combinação de uma chave secreta junto à função hash, podendo ser usada com outra função criptográfica, combinando com uma chave secreta previamente compartilhada que envia multimídia criptografada, oferecendo confiabilidade e integridade [Hércules 2014].

Referências

- Celuppi, I. C., Lima, G. D. S., Rossi, E., Wazlawick, R. S., & Dalmarco, E. M. (2021). Uma análise sobre o desenvolvimento de tecnologias digitais em saúde para o enfrentamento da COVID-19 no Brasil e no mundo. In *Cadernos de Saúde Pública*, v. 37, n. 3, p. 1-12.
- Dollard, K. (2008) “Code Generation in Microsoft”. NET. Apress.
- Hércules, L. A. L. (2014). Impacto no desempenho em aplicações de tempo real utilizando criptografia.
- Lantin, P. J. S., Shama, S. D. F. M. S., & Bez, M. R. (2020). Desenvolvimento de um software para controle nominal do calendário de vacinação pela rede pública de saúde. In *Journal of Health Informatics*, v. 12, n. 3, p. 71-76.
- Lima, A. C., Januário, M. C., Lima, P. T., & de Moura, W. (2015). DATASUS: o uso dos Sistemas de Informação na Saúde Pública. In *Revista FATEC Zona Sul*, v. 1, n. 3, p. 16-31.
- Milani, A. (2008). PostgreSQL-Guia do Programador. Novatec Editora.
- Mizuta, A. H., Succi, G. D. M., Montalli, V. A. M., & Succi, R. C. D. M. (2019). Percepções acerca da importância das vacinas e da recusa vacinal numa escola de medicina. In *Revista Paulista de Pediatria*, v. 37, n. 1, p. 34-40.
- Muratt, P., Alves, R. S., Oliveira, W. W., & Almeida, L. F. D. (2016) Carteira de vacinação virtual. In *Anais do XV Congresso Brasileiro de Informática em Saúde*, SBC, Goiânia.
- Payne, R. (2019) “Hello Flutter”. In: *Beginning App Development with Flutter: Create Cross-Platform Mobile Apps*, Edited by Rap Payne, Apress, Berkeley, CA. p. 3-8.
- Reis, L. (2016) “Metodologia ágil FDD”, <https://leandromtr.com/metodologia-agil-fdd/>, Abril.
- Stocco, G., & Baldani, M. H. (2011). O controle das consultas odontológicas dos bebês por meio da carteira de vacina: avaliação de um programa-piloto desenvolvido na

Estratégia Saúde da Família em Ponta Grossa (PR, Brasil)”. In *Ciência & Saúde Coletiva*, v. 16, n. 4, p. 2311-2321.

Tibes, C. M. D. S., Dias, J. D., & Zem-Mascarenhas, S. H. (2014). Aplicativos móveis desenvolvidos para a área da saúde no Brasil: revisão integrativa da literatura. *Revista Mineira de Enfermagem*, 18(2), 471-486.

Apêndice A. Tabela de Estimativa de Tempo

Tabela 2. Planejamento por funcionalidade por estimativo de tempo

Ordem	Funcionalidade	Tempo estimado
1	RF.1 Gerenciar Paciente	10 dias
2	RF.2 Gerenciar Profissional da Saúde	10 dias
3	RF.3 Gerenciar Administrador	10 dias
4	RF.4 Gerenciar Vacina	20 dias
5	RF.5 Gerenciar Laboratório	10 dias
6	RF.6 Gerenciar Vacinas Aplicadas	30 dias
7	RF.7 Gerenciar Campanha	20 dias
8	RF.8 Gerenciar Mensagem	10 dias

Apêndice B. Diagrama de Caso de Uso

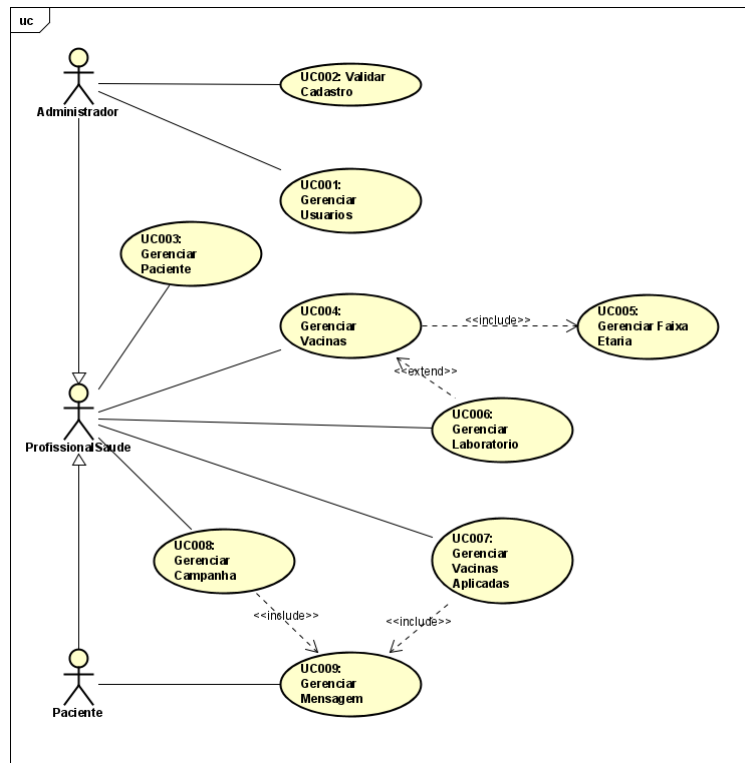


Figura 11. Diagrama de Caso de Uso

A Figura 11 representa a aplicação das funcionalidades do sistema utilizando o Diagrama de Caso de Uso.

Apêndice C. Descritivo de Caso de Uso

Tabela 3. Descritivo de Caso de Uso – Gerenciar Vacinas

Identificação	CDU01
Caso de Uso	Gerenciar Vacinas
Atores Principais	Profissional da Saúde (PA)
Atores Secundários	N/A
Pré-Condição	O ator deve estar logado no sistema
Pós-Condição	N/A
Fluxo Principal	<p>P01 – O Sistema apresenta a interface inicial do sistema;</p> <p>P02 – O ator seleciona o menu Gerenciar Vacinas;</p> <p>P03 – O Sistema apresenta a interface de Gerenciar Vacinas;</p> <p>P04 - O PA seleciona a opção de inserir Vacina [A01], consultar Vacina [A02], excluir Vacina [A03] e alterar Vacina [A04];</p> <p>P05 – O Sistema grava os dados no Banco de Dados;</p> <p>P06 – Fim do caso de uso;</p>
Fluxo Alternativo	<p>[A01] – Inserir Vacina:</p> <p>A01.1 – O Sistema exibe a listagem das vacinas;</p> <p>A01.2 – O PA preenche os dados para inserção [UC006];</p> <p>A01.3 – O PA seleciona a faixa etária [UC005];</p> <p>A01.4 – O ator clica no botão Salvar;</p> <p>A01.5 – O Sistema Grava os dados no Banco de Dados.</p> <p>A01.6 – O Sistema retorna para P03;</p> <p>[A02] – Consultar Vacina:</p> <p>A02.1 – O Sistema exibe a listagem das vacinas;</p> <p>A02.2 – O PA preenche os dados para consulta;</p> <p>A02.3 – O Sistema exibir os dados pesquisados [E02].</p> <p>A02.4 – O Sistema retorna para P03;</p> <p>[A03] – Excluir Vacina:</p> <p>A03.1 – O Sistema exibe a listagem das vacinas;</p> <p>A03.2 – O PA seleciona a Vacina a ser excluída;</p> <p>A03.3 – O ator clica no botão Excluir;</p> <p>A03.4 – O sistema valida os dados [E01];</p> <p>A03.5 – O Sistema Grava os dados no Banco de Dados.</p> <p>A03.6 – O Sistema retorna para P03;</p> <p>[A04] – Alterar Vacina:</p> <p>A04.1 – O Sistema exibe a listagem das vacinas;</p> <p>A04.2 – O PA verifica e altera os dados necessários [UC006];</p> <p>A01.3 – O PA seleciona a faixa etária [UC005] se necessário;</p> <p>A04.4 – O PA clica no botão Alterar;</p> <p>A04.5 – O Sistema valida os dados [E01];</p> <p>A04.6 – O Sistema Grava os dados no Banco de Dados.</p> <p>A04.6 – O Sistema retorna para P03;</p>
Fluxo de Exceção	[E01] – Dados inválidos:

	<p>E01.1 – O Sistema exibe alerta mostrando os dados inválidos;</p> <p>E01.2 – O Sistema retorna para o fluxo correspondente;</p> <p>[E02] – Registros não encontrados:</p> <p>E02.1 – O Sistema exibe a interface de listagem sem registros.</p>
Casos de Usos Incluídos	[UC005]
Casos de Usos Estendidos	[UC006]
Observação	N/A

Tabela 4. Descritivo de Caso de Uso – Gerenciar Cronograma da Campanha

Identificação	CDU01
Caso de Uso	Gerenciar Cronograma da Campanha
Atores Principais	Profissional da Saúde (PS)
Atores Secundários	N/A
Pré-Condição	O ator deve estar logado no sistema
Pós-Condição	N/A
Fluxo Principal	<p>P01 – O Sistema apresenta a interface inicial do sistema;</p> <p>P02 – O PS seleciona o menu Gerenciar Campanha;</p> <p>P03 – O Sistema apresenta a interface de Gerenciar Campanha;</p> <p>P04 – O PS seleciona o menu Gerenciar Cronograma da Campanha;</p> <p>P05 – O PS seleciona a opção de inserir Cronograma [A01], consultar Cronograma [A02], excluir Cronograma [A03], alterar Cronograma[A04];</p> <p>P05 – O Sistema grava os dados no Banco de Dados;</p> <p>P06 – O sistema envia mensagens para o público cadastrado [UC009]</p> <p>P07 – Fim do caso de uso;</p>
Fluxo Alternativo	<p>[A01] – Inserir Cronograma:</p> <p>A01.1 – O Sistema exibe a listagem dos cronogramas;</p> <p>A01.2 – O PS preenche os dados para inserção;</p> <p>A01.3 – O ator clica no botão Salvar;</p> <p>A01.4 – O Sistema Grava os dados no Banco de Dados.</p> <p>A01.5 – O Sistema retorna para A05;</p> <p>[A02] – Consultar Cronograma:</p> <p>A02.1 – O Sistema exibe a listagem dos cronogramas;</p> <p>A02.2 – O PS preenche os dados para consulta;</p> <p>A02.3 – O Sistema exibir os dados pesquisados[E02].</p> <p>A02.4 – O Sistema retorna para P03;</p> <p>[A03] – Excluir Cronograma:</p> <p>A03.1 – O Sistema exibe a listagem dos cronogramas;</p> <p>A03.2 – O PS seleciona o Cronograma a ser excluída;</p> <p>A03.3 – O PS clica no botão Excluir;</p> <p>A03.4 – O Sistema valida os dados [E01];</p>

	<p>A03.5 – O Sistema Grava os dados no Banco de Dados. A03.6 – O Sistema retorna para A06;</p> <p>[A04] – Alterar Cronograma: A04.1 – O Sistema exibe a listagem dos cronogramas; A04.2 – O PS verifica e altera os dados necessários; A04.3 – O PS clica no botão Alterar; A04.4 – O Sistema valida os dados [E01]; A04.5 – O Sistema Grava os dados no Banco de Dados. A04.6 – O Sistema retorna para A07;</p> <p>[A05] – Inserir Público-alvo: A05.1 – O Sistema exibe a listagem do público-alvo; A05.2 – O PS preenche os dados para inserção; A05.3 – O PS clica no botão Salvar; A05.4 – O Sistema Grava os dados no Banco de Dados. A05.5 – O Sistema retorna para P03;</p> <p>[A06] – Excluir Público-alvo: A06.1 – O Sistema exibe a listagem do público-alvo; A06.2 – O PS seleciona o Cronograma a ser excluída; A06.3 – O PS clica no botão Excluir; A06.4 – O Sistema valida os dados [E01]; A06.5 – O Sistema Grava os dados no Banco de Dados. A06.6 – O Sistema retorna para P03;</p> <p>[A07] – Alterar Público-alvo: A07.1 – O Sistema exibe a listagem do público-alvo; A07.2 – O PS verifica e altera os dados necessários; A07.3 – O PS clica no botão Alterar; A07.4 – O Sistema valida os dados [E01]; A07.5 – O Sistema Grava os dados no Banco de Dados. A07.6 – O Sistema retorna para P03;</p>
Fluxo de Exceção	<p>[E01] – Dados inválidos: E01.1 – O Sistema exibe alerta mostrando os dados inválidos; E01.2 – O Sistema retorna para o fluxo correspondente;</p> <p>[E02] – Registros não encontrados: E02.1 – O Sistema exibe a interface de listagem sem registros.</p>
Casos de Usos Incluídos	[UC009]
Observação	N/A

Apêndice D. Diagrama de Classe

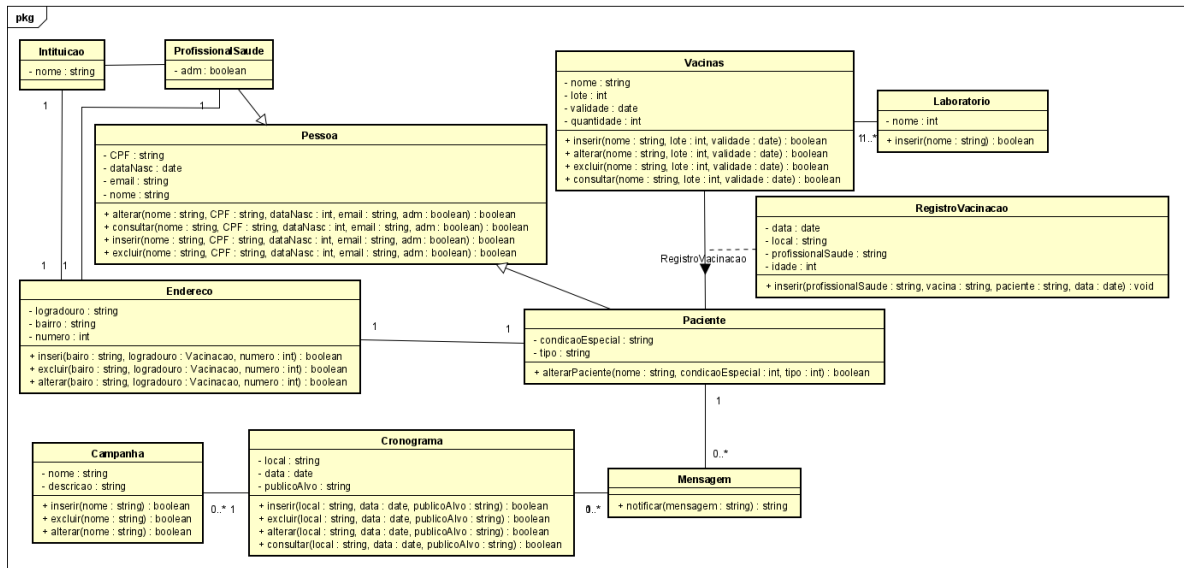


Figura 12. Diagrama de Classe

Apêndice E. Diagrama de Entidade e Relacionamento

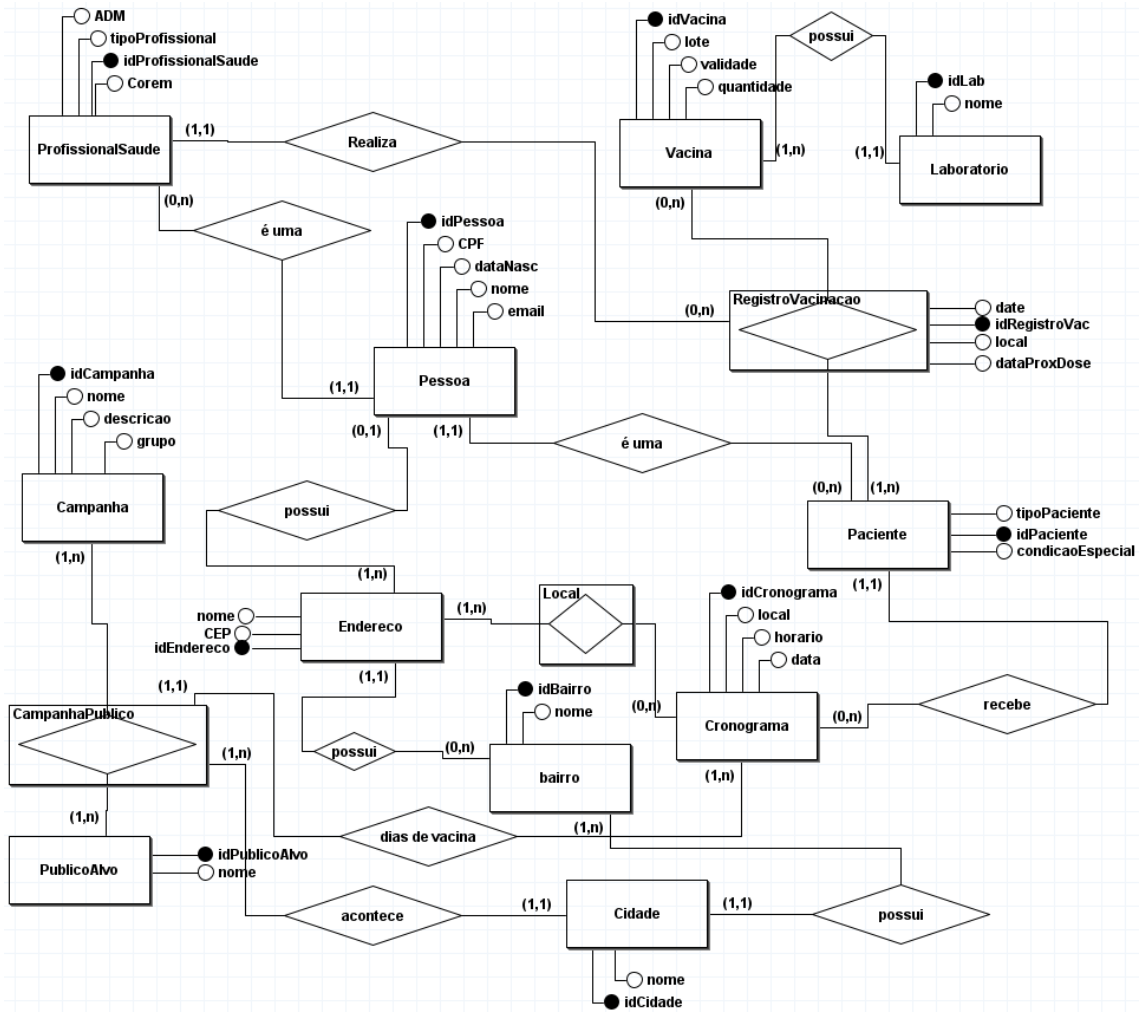


Figura 13. Diagrama Entidade Relacionamento

Apêndice F. Protótipos de Interface

Na Figura 14, são mostradas três telas de protótipos da interface para ilustrar como o aplicativo será desenvolvido. A Figura 14A é a interface do login, onde o usuário poderá adentrar ao sistema como um profissional da saúde ou paciente. A Figura 14B apresenta a interface do usuário, quando ele for um profissional da saúde, onde ele tem acesso às funções descritas nos botões abaixo. Enquanto a Figura 14C é a demonstração ao usuário, quando ele é um paciente, uma interface do software que mostra suas vacinas e suas informações.

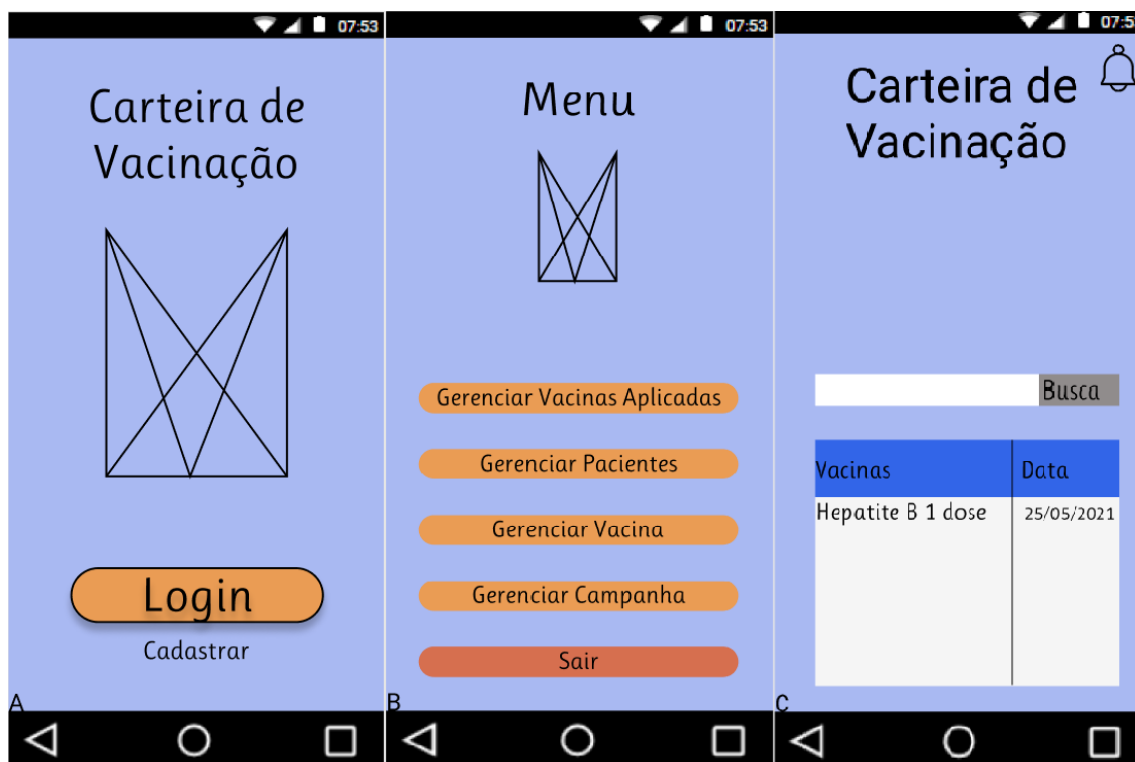


Figura 14. Protótipos de interface para o menu principal do login, paciente e profissional da saúde

Na Figura 15, são apresentadas algumas interfaces de como o aplicativo será desenvolvido, sendo todas elas disponibilizadas para o usuário quando ele for um profissional da saúde. A Figura 15A é a interface para o usuário manusear as vacinas. Enquanto a Figura 15B é a interface de alterar ou visualizar as vacinas que foram aplicadas nos pacientes. Na Figura 15C, são apresentadas as informações do paciente que foram selecionadas pelo usuário, e suas vacinas já aplicadas, com a possibilidade de inserir, alterar, excluir ou pesquisar as vacinas do paciente.



Figura 15. Protótipos de interface de controle de vacina utilizado pelo profissional da saúde