

**INSTITUTO
FEDERAL**

Goiás

Instituto Federal de Goiás

Campus Formosa

Análise e Desenvolvimento de Sistemas

<http://www.ifg.edu.br/formosa>

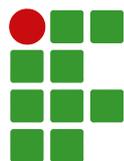
**CARVAVI: SISTEMA *WEB* PARA CONTROLE DE VACINAÇÃO DE PACIENTES
ATENDIDOS EM UNIDADES BÁSICAS DE SAÚDE**

YAMILLA DA SILVA CALIXTO

Trabalho de Conclusão de Curso

Formosa

2023



**INSTITUTO
FEDERAL**

Goiás

Instituto Federal de Goiás

Campus Formosa

Análise e Desenvolvimento de Sistemas

<http://www.ifg.edu.br/formosa>

**CARVAVI: SISTEMA *WEB* PARA CONTROLE DE VACINAÇÃO DE
PACIENTES ATENDIDOS EM UNIDADES BÁSICAS DE SAÚDE**

Yamilla da Silva Calixto

Trabalho de Conclusão de Curso apresentado ao Departamento de Áreas Acadêmicas da Instituto Federal de Goiás campus Formosa, como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Mário Teixeira Lemes

Formosa

2023

Yamilla da Silva Calixto

CARVAVI: Sistema *web* para controle de vacinação de pacientes atendidos em unidades básicas de saúde/ Yamilla da Silva Calixto. – Formosa, 2023-
58 p.; 30 cm.

Orientador Prof. Me. Mário Teixeira Lemes

Trabalho de Conclusão de Curso – Instituto Federal de Goiás, 2023.

1. Plano Semestral de Trabalho Docente 2. IFG 3. Desenvolvimento de software
4. Aplicação *Web* I. Orientador: Prof. Me. Mário Teixeira Lemes. II. Instituto Federal de Goiás. IV. Título: CARVAVI: Sistema *web* para controle de vacinação de pacientes atendidos em unidades básicas de saúde

CDU 02:141:005.7

.pdf" .png" .jpg" .mps" .jpeg" .jbig2" .jb2" .PDF" .PNG" .JPG" .JPEG" .JBIG2" .JB2" .eps"



INSTITUTO FEDERAL
Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CÂMPUS FORMOSA

ATA DA SESSÃO PÚBLICA DE APRESENTAÇÃO E DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

Ao 10º dia do mês de janeiro do ano de dois mil e vinte e três, às 20:30 horas, no Instituto Federal de Goiás, Câmpus Formosa, situado à Rua 64, Setor Expansão Parque Lago da cidade de Formosa, Estado de Goiás, foi realizada a sessão pública de apresentação e defesa do Trabalho de Conclusão de Curso da Graduanda **Yamilla da Silva Calixto** (matrícula 20171070130080) do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, no segundo semestre letivo do ano de dois mil e vinte e dois. A banca foi composta pelos seguintes membros: Profº. Me. Mário Teixeira Lemes (IFG/Formosa), Profª Ma. Iasmini Virginia Oliveira Lima (IFG/Formosa) e Profº Me. Gleidson Caetano da Silveira Pinto (IFG/Formosa). O Trabalho de Conclusão de Curso tem como título "Carvavi: Sistema Web para Controle de Vacinação de Pacientes Atendidos em Unidades Básicas de Saúde", da área de Informática, sob orientação do Profº. Me. Mário Teixeira Lemes. Após apresentação do Trabalho de Conclusão de Curso, tendo sido a autora arguida pela Banca Examinadora, a nota obtida foi 8,0 pontos, sendo, portanto, aprovada com correções.

Encerra-se a presente sessão às 21 horas e 55 minutos. Eu, Profª Ma. Iasmini Virginia Oliveira Lima, dato e assino a presente ata que segue assinada por todos os membros da Banca e pela graduanda.

Profª Ma. Iasmini V. Oliveira Lima

Profº Me. Gleidson C. da Silveira Pinto

Profº. Me. Mário Teixeira Lemes

Yamilla da Silva Calixto
(Graduanda)

Documento assinado eletronicamente por:

- **Mário Teixeira Lemes**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 06/02/2023 16:07:26.
- **Gleidson Caetano da Silveira Pinto**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 10/01/2023 21:52:13.
- **Iasmini Virginia Oliveira Lima**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 10/01/2023 21:51:24.

Este documento foi emitido pelo SUAP em 09/01/2023. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifg.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 362918
Código de Autenticação: a14fd8dfb9



Instituto Federal de Educação, Ciência e Tecnologia de Goiás
Rua 64 esquina com Rua 11, S/Nº, Expansão Parque Lago, Formosa / GO, CEP 73813-816
(61) 3642-9491 (ramal: 9491), (61) 3642-9493 (ramal: 9493)



INSTITUTO FEDERAL

Goiás

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
SISTEMA INTEGRADO DE BIBLIOTECAS

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

- | | |
|--|---|
| <input type="checkbox"/> Tese | <input type="checkbox"/> Artigo Científico |
| <input type="checkbox"/> Dissertação | <input type="checkbox"/> Capítulo de Livro |
| <input type="checkbox"/> Monografia – Especialização | <input type="checkbox"/> Livro |
| <input checked="" type="checkbox"/> TCC - Graduação | <input type="checkbox"/> Trabalho Apresentado em Evento |
| <input type="checkbox"/> Produto Técnico e Educacional - Tipo: _____ | |

Nome Completo do Autor: Yamilla da Silva Calixto

Matrícula: 20171070130080

Título do Trabalho: CARVAVI: Sistemaweb para controle de vacinação de pacientes atendidos em unidades básicas de saúde

Restrições de Acesso ao Documento

Documento confidencial: Não Sim, justifique: _____

Informe a data que poderá ser disponibilizado no ReDi/IFG: ___/___/___

O documento está sujeito a registro de patente? Sim Não

O documento pode vir a ser publicado como livro? Sim Não

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

- i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
- ii. obteve autorização de quaisquer materiais incluídos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
- iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Formosa, GO, 28 / 02 / 2023

Local Data

Yamilla da Silva Calixto

Assinatura do Autor e/ou Detentor dos Direitos Autorais

Agradeço primeiramente a minha mãe Eliene que esteve comigo todos os anos da minha vida. Sempre esteve presente durante minha caminhada escolar, dizendo que sem estudo nunca seria nada e que me apoiou em todas decisões e escolhas até o presente momento.

Agradeço também ao meu namorado Matheus que me apoia na minha vida educacional/-pessoal e que me fez crescer. Ele esteve do meu lado quando eu pensei em desistir e me mostrou que sou capaz, não me deixando desanimar no meio da caminhada. Por isso eu cheguei aqui.

Agradeço aos meus amigos que fiz durante todos esses anos e que dividiram conhecimentos, ideias e sonhos.

Agradeço ao meu orientador professor Mário Teixeira Lemes que me acompanhou durante todos esses anos de pesquisa. Ele me apoiou no tema deste TCC e sempre dividiu seus conhecimentos, com paciência e sabedoria, diante as dificuldades enfrentadas.

Por fim, agradeço todos(as) os(as) professores/as do IFG-*Campus* Formosa que passaram nesses anos da minha graduação e contribuíram com conhecimentos que serão importantes para minha carreira profissional como futura Tecnóloga em Análise e Desenvolvimento de Sistemas.

Resumo

A integração de espaços e a facilidade da locomoção impulsionados pela globalização e modernização favorecem a disseminação de doenças. Neste sentido, como contra-medida à proliferação, a vacinação atualizada de toda população é fator primordial para promoção da vida humana pois possibilita a diminuição de taxas de multiplicação e crescimento das enfermidades. A maioria das vacinas podem ser encontradas nas Unidades Básicas de Saúde que disponibilizam profissionais capacitados para registro e aplicação das doses. A rotina de imunização na rede pública é prejudicada pelo fato da não existência de sistema informatizado para controle da vacinação, desde a manipulação das doses disponíveis até o controle das vacinas que foram efetivamente aplicadas. O objetivo deste trabalho é desenvolver um sistema informatizado para controle do sistema de vacinação de uma Unidade Básica de Saúde, permitindo cadastro de técnicos de enfermagem, unidades, pacientes e vacinas, bem como o registro da vacinação e emissão do cartão de vacina de pacientes. O sistema facilitará o trabalho de registro, aplicação das vacinas e acompanhamento dos profissionais de saúde, além de proporcionar maior conforto aos pacientes que poderão certificar das vacinas tomadas, através da emissão do cartão de vacina.

Palavras-chave: Sistema *Web*, Vacinação, Controle de Vacinas, Cartão de Vacina Virtual.

Abstract

The integration of spaces and the ease of movement driven by globalization and modernization favor the spread of diseases. In this context, as a countermeasure to proliferation, up-to-date vaccination of the entire population is a key factor as it makes it possible to reduce the rates of multiplication and growth of diseases. Most vaccines can be found in Basic Health Units that provide trained professionals to record and apply doses. The immunization routine in Basic Health Units is hampered by the fact that there is no computerized system to control vaccination, from the manipulation of available doses to the control of vaccines that were actually applied. The main of this work is to develop a computerized system to control the vaccination system of a Basic Health Unit, allowing registration of nursing technicians, units, patients and vaccines, as well as the registration of vaccination and issuance of the card patient vaccines. The system will facilitate the work of health professionals, from registration and administration of vaccines to medical follow-up, and will provide greater comfort to patients who will be able to have guarantees of the vaccines taken, through the issuance of the mirror of the virtual vaccine card.

Keywords: Web System, Vaccination, Vaccine Control, Virtual Vaccine Card.

Lista de Figuras

2.1	Funcionamento do Padrão MVC	25
2.2	Representação Gráfica dos Diagramas da UML	27
2.3	Componentes e Usuários de um Banco de Dados	27
2.4	Representação Gráfica da Interação entre <i>Front-End</i> e <i>Back-End</i>	30
4.1	Diagrama de Casos de Uso	37
4.2	Diagrama de Atividades - Administrador	41
4.3	Diagrama de Atividades - Técnico de Enfermagem	41
4.4	Diagrama de Atividades - Paciente	42
4.5	Diagrama de Sequência - Ator Administrador	43
4.6	Diagrama de Sequência - Ator Técnico de Enfermagem	44
4.7	Diagrama de Sequência - Ator Paciente	45
4.8	Diagrama de Classes	46
4.9	Diagrama Entidade-Relacionamento	48
4.10	Área do Administrador	49
4.11	Cadastrar Unidade Básica de Saúde	50
4.12	Unidades Básicas de Saúde Cadastradas	50
4.13	Cadastrar Vacina	51
4.14	Incluir Detalhes das Vacinas Cadastradas	52
4.15	PDF do Cartão de Vacina Virtual	53

Lista de Tabelas

4.1	Documentação do Caso de Uso - Cadastrar Unidade Básica de Saúde.	38
4.2	Documentação do Caso de Uso - Cadastrar Técnico de Enfermagem.	38
4.3	Documentação do Caso de Uso - Efetuar <i>Login</i>	38
4.4	Documentação do Caso de Uso - Cadastrar Vacinas.	39
4.5	Documentação do Caso de Uso - Cadastrar Pacientes.	39
4.6	Documentação do Caso de Uso - Incluir Vacinas	40
4.7	Documentação do Caso de Uso - Gerar o Cartão de Vacina.	40

Lista de Acrônimos

HTML	<i>Hypertext Markup Language</i>	28
CSS	<i>Cascading Style Sheets</i>	29
SQL	<i>Structured Query Language</i>	28
PDF	<i>Portable Document Format</i>	35
PNI	Programa Nacional de Imunizações	21
UML	Linguagem de Modelagem Unificada	22
BD	Banco de Dados	26
IDE	<i>Integrated Development Environment</i>	32
TCC	Trabalho de Conclusão de Curso	23
MVC	<i>Model-View-Controller</i>	25
UP	<i>Unified Process</i>	26
SGBD	Sistema Gerenciador de Banco de Dados	28
UBS	Unidade Básica de Saúde	21
PHP	<i>Hypertext Preprocessor</i>	28

Sumário

1	Introdução	21
1.1	Descrição dos Capítulos	23
2	Referencial Teórico	25
2.1	Arquitetura de <i>Software</i>	25
2.1.1	Padrão <i>Model-View-Controller</i> (MVC)	25
2.2	Processo Unificado	26
2.3	UML	26
2.4	Banco de Dados	26
2.4.1	Banco de Dados Relacional	27
2.4.2	SQL	28
2.4.3	Sistema Gerenciador de Banco de Dados (SGBD)	28
2.4.3.1	MySQL	28
2.5	XAMPP	28
2.6	<i>Hypertext Preprocessor</i> (PHP)	28
2.7	Sistema <i>Web</i>	28
2.7.1	<i>Hypertext Markup Language</i> (HTML)	29
2.7.2	<i>Cascading Style Sheets</i> (CSS)	29
2.7.3	<i>Framework</i>	29
2.7.4	Programação <i>Front-end</i>	29
2.7.4.1	<i>Bootstrap</i>	29
2.7.5	Programação <i>Back-end</i>	29
3	Método	31
3.1	Métodos de Pesquisa	31
3.2	Ferramentas Utilizadas	32
4	CARVAVI: Análise e Documentação	33
4.1	Metodologia de Desenvolvimento	33
4.1.1	Entrevista	33
4.1.2	Requisitos Funcionais	34
4.1.3	Requisitos Não-Funcionais	35
4.1.4	Artefatos da UML	36
4.1.4.1	Diagrama de Casos de Uso	36
4.1.4.2	Documentação dos Casos de Uso	37
4.1.4.3	Diagrama de Atividades	40

4.1.4.4	Diagrama de Sequência	42
4.1.4.5	Diagrama de Classes	43
4.1.5	Diagrama Entidade-Relacionamento	47
4.2	CARVAVI: Visão Geral	49
4.2.1	Área do Administrador	49
4.2.1.1	Cadastrar Unidade Básica de Saúde	49
4.2.1.2	Cadastrar Técnico de Enfermagem	50
4.2.1.3	Unidades Básicas de Saúde Cadastradas	50
4.2.1.4	Técnicos de Enfermagem Cadastrados	51
4.2.2	Área do Técnico de Enfermagem	51
4.2.2.1	Cadastrar Paciente	51
4.2.2.2	Cadastrar Vacina	51
4.2.2.3	Vacinas Cadastradas	52
4.2.2.4	Pacientes Cadastrados	52
4.2.2.5	Gerar Cartão de Vacina	52
4.2.3	Área do Paciente	53

5 Conclusão **55**

1

Introdução

A globalização é caracterizada pela facilidade de locomoção das pessoas entre diferentes cidades, estados e países. Essa integração do espaço mundial favorece a disseminação de doenças, especialmente de pandemias. As pandemias, tais como a COVID-19, têm impactado o cenário mundial, agravando as taxas de morbidade e mortalidade (?). Segundo (?), diante desse cenário, é urgente a necessidade de capacitação e qualificação de profissionais da área da saúde no enfrentamento das doenças, além de possibilitar uso de sistemas informatizados para devido controle e eficácia do sistema de vacinação.

No Brasil, as vacinas são aplicadas diariamente pela rede pública de saúde. Assim, qualquer pessoa pode dirigir-se a uma Unidade Básica de Saúde (UBS) para manter suas vacinas devidamente atualizadas. As campanhas de vacinação são controladas pelo Programa Nacional de Imunizações (PNI) da Secretaria de Vigilância em Saúde do Ministério da Saúde. Esta entidade tem a responsabilidade de controlar as doenças, através da disponibilização de vacinas (?). As vacinas são ofertadas a comunidade em diferentes periodicidades: algumas estão disponíveis diariamente, enquanto outras são ofertadas anualmente.

De acordo com (?), as primeiras vacinas foram criadas por volta do ano de 1800. Para os autores, as vacinas são produzidas por partes enfraquecidas dos vírus ou de seus derivados. Para (?), a vacina carrega informações sobre a doença para que o corpo receba informações sobre os vírus, produza antígenos e os combata. Assim, quando a pessoa imunizada tiver contato com a doença não a contrai, ou em caso positivo, os sintomas são mais brandos e, em alguns casos, evitam o óbito. Entende-se, dessa forma, que a vacina tem o objetivo de promoção da vida humana e é crucial o fato da manutenção da vacinação atualizada de toda população, a fim de evitar proliferação de doenças.

As vacinas são divididas por etapas na vida do ser humano, no qual a primeira etapa se inicia logo após nascimento, passando pela infância até completar o ciclo imunológico na fase adulta. Na adolescência, considerada um ciclo de reforço, diferentes vacinas são reaplicadas. Segundo (?), as vacinas são especialmente importantes para gestantes e idosos que precisam ter seus sistemas imunológicos reforçados.

A rotina de imunização na rede pública é prejudicada pelo fato do não uso de sistema

informatizado para controle da vacinação, desde a manipulação das doses disponíveis bem como o controle das vacinas que foram efetivamente aplicadas. O espelho do cartão de vacina é um documento que certifica as vacinas que foram ministradas a uma pessoa, contendo informações tais como nome do imunizante, lote de fabricação, tipo da dose e etc. O controle das vacinas nas Unidades Básicas de Saúde e o cartão de vacina são, em sua maioria, produzidos e controlados manualmente usando papel. O trabalho manual é repetitivo e suscetível a erros, podendo ocasionar inconsistência dos dados. Outro problema comum é a perda dos cartões de vacina por pacientes. Em caso de perda do cartão de vacina, o paciente deve procurar uma UBS e novamente realizar a imunização. Este fato dificulta o acompanhamento médico, além de gerar desconforto para o paciente devido ao fato da necessidade de reaplicação de vacinas que o mesmo possa ter esquecido que tenha tomado (?).

?) desenvolveu um sistema gerenciador de vacinação que informa aos paciente as vacinas que devem ser tomadas. O ponto positivo deste trabalho é a possibilidade dos alertas enviados para possibilitar a vacinação. Apesar de disponibilizar a carteira de vacinação dos pacientes de forma *online*, não é possível realizar o *download* do documento para uso *offline*. Já ?) implementou um sistema de controle com intuito de informatizar o processo de vacinação, permitindo que os dados relacionados a imunização dos pacientes estejam em uma base de dados única. Apesar de possibilitar a emissão do cartão de vacinação, o sistema não realiza o controle da quantidade de doses de vacinas disponíveis para aplicação.

Objetivo Geral

Realizar análise, documentação e desenvolvimento de um sistema *Web* para controle do sistema de vacinação dentro de uma Unidade Básica de Saúde, permitindo cadastro de técnicos de enfermagem, unidades básicas de saúde, pacientes e vacinas, bem como possibilitar o registro da vacinação e emissão do cartão de vacinas de pacientes devidamente cadastrados.

Objetivos Específicos

1. Realizar análise e levantamento dos requisitos do sistema de controle de vacinação;
2. Construir diagramas de apoio usando a Linguagem de Modelagem Unificada (UML);
3. Modelar o banco de dados para armazenamento das informações geradas;
4. Implementar o sistema de vacinação utilizando uma linguagem de programação;
5. Construir um repositório público para disponibilização dos códigos;
6. Documentar processos de instalação, inicialização e uso do sistema.

1.1 Descrição dos Capítulos

Este Trabalho de Conclusão de Curso (TCC) está dividido em 4 (quatro) capítulos. Neste Capítulo tem-se uma breve introdução sobre o contexto da importância das vacinas para controle das doenças. São apontados os problemas encontrados nas Unidades Básicas de Saúde devido a não existência de sistemas informatizados e o uso de controle manual de vacinas aplicadas e geração de cartões de vacina. Apresentamos também o objetivo geral e os objetivos específicos.

O Capítulo 2 apresenta uma explicação teórica sobre conceitos importantes que foram usados durante todo o desenvolvimento deste trabalho. Já no Capítulo 3 são apresentados métodos de pesquisa e ferramentas que foram utilizadas para que fosse possível atingir os objetivos.

No Capítulo 4 apresentamos a análise e documentação do sistema de controle de vacinação, batizado de CARVAVI. Neste Capítulo encontram-se os requisitos funcionais e não-funcionais, diagramas da UML, bem como a construção do diagrama entidade-relacionamento. Este Capítulo também apresenta o sistema CARVAVI em diferentes perspectivas: mostramos como o sistema pode ser utilizado por administradores, técnicos de enfermagem e pacientes, bem como apresentamos as principais funcionalidades de cada nicho de usuários. Ademais, mostramos também a documentação para instalação, inicialização e uso. Finalmente, no Capítulo 5 são abordadas a conclusão e a descrição de possíveis trabalhos futuros.

2

Referencial Teórico

2.1 Arquitetura de *Software*

A arquitetura de *software* determina a forma que o sistema irá ser construído. Na arquitetura em camadas, o *software* é dividido por camadas que realizam funções específicas. Já na arquitetura orientada a objetos, os componentes se comunicam e coordenam suas ações através da passagem de mensagens. Os dados e operações são encapsulados. Finalmente, a arquitetura monolítica é aquela construída somente por um módulo (?).

2.1.1 Padrão *Model-View-Controller* (MVC)

O padrão *Model-View-Controller* (MVC) é uma arquitetura de *software* em camadas, no qual são separadas a lógica de negócios da interface do sistema. Cada camada é independente, o que facilita a reutilização de código e a manutenção da aplicação (?).

Na Figura 2.1 pode-se observar a comunicação do padrão MVC. A camada *view* é a interface gráfica que comunica-se com a camada *controller* que por sua vez é responsável por intermediar as requisições solicitadas pela *view* com as respostas enviadas pela camada *model*.

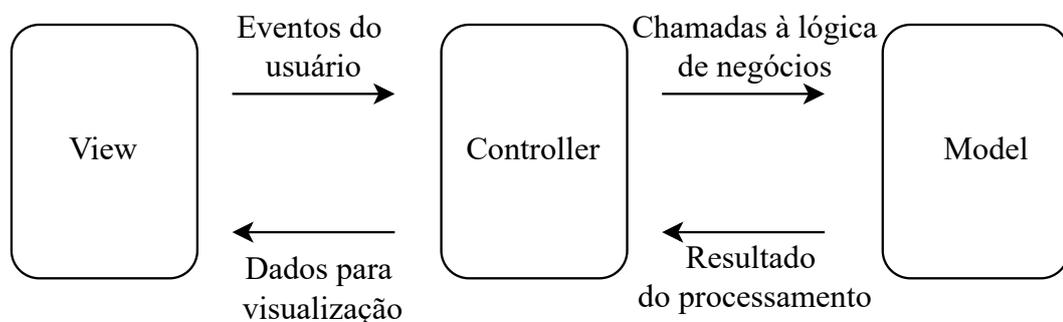


Figura 2.1: Funcionamento do Padrão MVC

Fonte: Autoria própria

2.2 Processo Unificado

É um conjunto de atividades com o intuito de converter os requisitos de usuário para um sistema de *software*. O *Unified Process* (UP) é um processo de repetição de ciclos durante todo o desenvolvimento do sistema. Em cada ciclo, uma versão do sistema está pronta para distribuição. O UP é dividido em 4 (quatro) fases: concepção, elaboração, implementação e transição.

Concepção é a fase de definição do escopo do sistema que está sendo desenvolvido. Elaboração é a etapa de desenvolvimento da definição dos requisitos. Construção onde o sistema é construído. Transição é quando o sistema é entregue para o cliente podendo ter alteração de acordo com as requisições apresentadas pelo cliente.(?).

2.3 UML

Segundo (?), a UML é uma linguagem gráfica de visualização e especificação, facilitando a construção e documentação de artefatos de sistemas complexos. A UML permite desenvolvimento padronizado de *software*, seguindo regras de arquitetura de projetos de sistemas.

A Figura 2.2 mostra alguns dos diagramas da UML que podem ser construídos para facilitar o processo de desenvolvimento de sistemas. O diagrama de classes é o responsável por determinar as classes que estarão presentes no sistema tendo juntamente seus atributos, métodos e como cada classe se relaciona uma com as outras.

O diagrama de casos de uso é o mais utilizado na parte de levantamento de requisitos e de análise. Este diagrama permite determinar o que os atores (usuários) irão fazer dentro do sistema. Já o diagrama de máquina de estados mostra o comportamento de um objeto, com base em passos da transição entre estados.

O diagrama de atividades mostra todo o caminho percorrido até a conclusão de uma determinada atividade. O diagrama de interação determina como os objetos do sistema interagem entre si. O diagrama de estrutura é responsável por determinar toda a estrutura interna de uma classe de sistema.

O diagrama de pacotes é usado para mostrar a representação gráfica da arquitetura das classes. O diagrama de implantação é o que determina as necessidades de *hardware*. O diagrama de componentes é a representação gráfica dos componentes de *software*. Finalmente, o diagrama de objetos é responsável por mostrar os objetos que foram instanciados dentro das classes.

2.4 Banco de Dados

Para (?), um Banco de Dados (BD) é uma representação de uma porção de mundo real, denominado mini-mundo, no qual cada alteração que é realizada na representação do mundo real, é necessário a atualização do mini-mundo. Neste sentido, BD é uma coleção de um conjunto lógico e ordenado de dados que possuem significado para o usuário. A Figura 2.3 mostra a representação dos processos de comunicação entre usuários, banco de dados e aplicação. Note

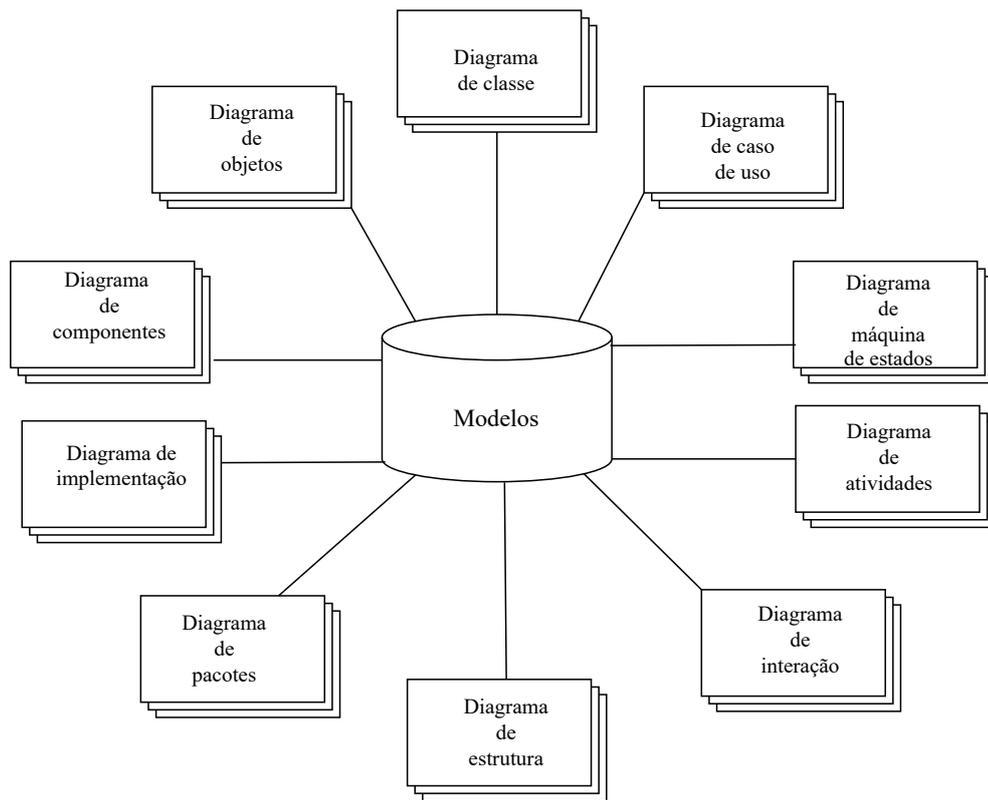


Figura 2.2: Representação Gráfica dos Diagramas da UML

Fonte: Autoria própria

na Figura 2.3 que usuários interagem com os aplicativos de acesso. Os aplicativos, por sua vez, interagem com o Sistema Gerenciador de Banco de Dados (SGBD). Este último recebe e envia informações do banco de dados.

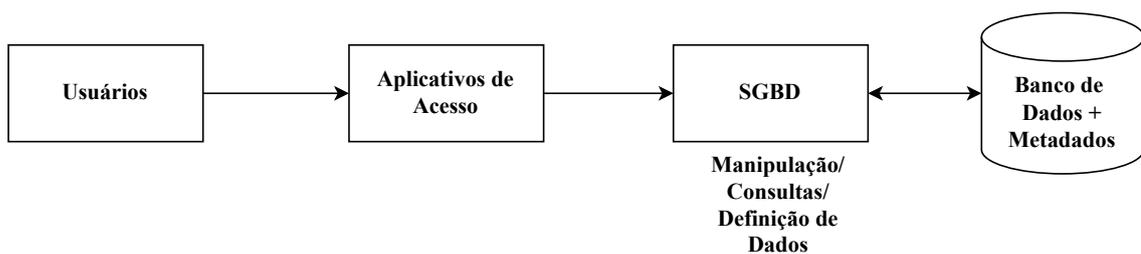


Figura 2.3: Componentes e Usuários de um Banco de Dados

Fonte: Autoria própria

2.4.1 Banco de Dados Relacional

Banco de dados relacional determina a forma que os dados são organizados e relacionados. Os dados são armazenados em tabelas. Cada tabela é estruturada em linhas (entidades) e colunas (os atributos das entidades). Cada tabela contém uma identificação única denominada de chave primária. Para o relacionamento das tabelas é usado uma chave estrangeira. A construção do

banco de dados relacional facilita a visualização e compressão das informações (?).

2.4.2 SQL

Structured Query Language (SQL) é uma linguagem que foi desenvolvida com o intuito de padronizar as ações que são realizadas no banco de dados, permitindo que seja possível realizar várias interações com os dados de um banco de dados relacional. Pode ser utilizada em vários Sistema Gerenciador de Banco de Dados (SGBD). A linguagem SQL possui várias funções que podem ser aplicadas no obter um determinado resultado. Dentre algumas funções, podemos citar: inserir, recuperar, alterar, excluir, dentre outras interações (?).

2.4.3 SGBD

É um conjunto de aplicações que dá a liberdade para o usuário no desenvolvimento e manutenção do bancos de dados. Os SGBDs facilitam a construção, manipulação e o compartilhamentos entre vários usuários ou mesmos entre as aplicações (?).

2.4.3.1 MySQL

É um servidor de SGBD completo, robusto, eficiente, *open source* usado para gerenciamento de aplicações com banco de dados. O MySQL utiliza a linguagem SQL para inserir, acessar e gerenciar conteúdos armazenados (?).

2.5 XAMPP

Para (?), o XAMPP é um *software* que tem como a funcionalidade a presença de servidores de código aberto, tais como o APACHE, MySQL e o *Hypertext Preprocessor* (PHP). Este programa facilita o trabalho de desenvolvimento para programadores *Web*, uma vez que simula um servidor local, facilitando o processo de construção e depuração de códigos criados, ao retirar a necessidade da hospedagem das páginas em servidores na *Internet*.

2.6 PHP

Segundo (?), PHP é uma linguagem de código-aberto para o desenvolvimento *Web* do lado do servidor. Esta linguagem possui interação com *Hypertext Markup Language* (HTML), o que facilita a criação de sistemas. PHP foi criado no ano de 1994 pelo engenheiro de *software* Rasmus Lerdof e foi inspirada nas linguagens C, Perl e Java.

2.7 Sistema Web

De acordo com (?), sistema *Web* é um *software* que é hospedado na *Internet*, não sendo necessário a instalação no computador do usuário. Dessa forma, o mesmo pode ser acessado de

qualquer navegador.

2.7.1 HTML

HTML é uma linguagem de marcação que tem como função auxiliar a criação de páginas e documentos eletrônicos para a *Internet*, trazendo informações para usuários, navegadores e mecanismos de busca. O HTML não é uma linguagem de programação, já que seu foco é marcar as estruturas de linguagens de programação. O HTML é usado no desenvolvimento visual de um sistema *Web* (?).

2.7.2 *Cascading Style Sheets* (CSS)

Para (?), CSS é uma linguagem de estilo que tem a funcionalidade de criação de *layouts* das páginas ou *sites*, permitindo que o desenvolvedor projete páginas personalizadas. CSS é usado para definir as apresentações dos documentos que foram escritos por uma linguagem de marcação, tais como na linguagem HTML.

2.7.3 *Framework*

Framework são estruturas compostas por códigos prontos que facilitam o desenvolvimento de aplicações. Essas estruturas carregam vários códigos pré-prontos personalizáveis, fazendo com que o desenvolvimento se torne mais eficiente. Os *frameworks* são seguros, versáteis e eficientes (?).

2.7.4 Programação *Front-end*

É responsável por toda a interface gráfica e suas interações entre usuários e aplicação. O desenvolvimento *Web front-end* utiliza-se de HTML, CSS e JavaScript (?).

2.7.4.1 *Bootstrap*

Bootstrap é um *framework front-end* de código-fonte aberto e tem como capacidade o fornecimento de estruturas em HTML e CSS pré-programadas, tais como botões, tabelas, listas, entre várias outras possibilidades. O foco é o desenvolvimento de *sites* e aplicações responsivas de forma rápida e simples. O *bootstrap* foi desenvolvido por um grupo de desenvolvedores comandado pela dupla Mark Otto e Jacob Thornton (?).

2.7.5 Programação *Back-end*

Segundo (?), a lógica de programação *back-end* é relacionada com servidores e banco de dados. Neste caso, o objetivo é tratar a manipulação dos dados, tais como inserção, leitura e armazenamento. A Figura 2.4 ilustra uma representação do funcionamento do *back-end*. Note

que a lógica *back-end* é separada da representação gráfica. No *front-end*, o cliente, normalmente um navegador de *Internet*, realiza uma requisição para o *back-end*, que por sua vez, realiza o processamento junto ao banco de dados e retorna informações que são exibidas para o cliente.

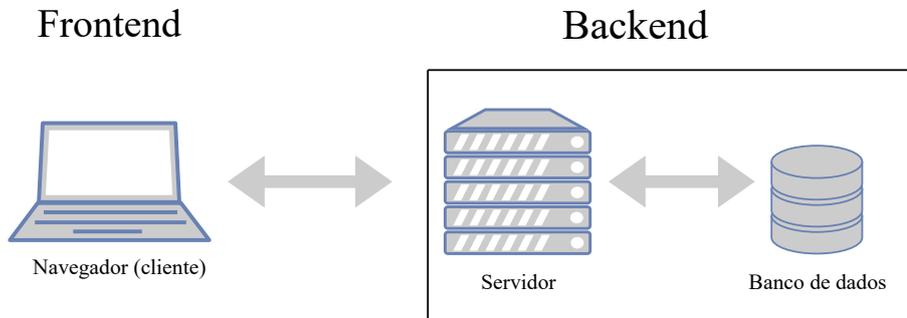


Figura 2.4: Representação Gráfica da Interação entre *Front-End* e *Back-End*

Fonte: Autoria própria

3

Método

3.1 Métodos de Pesquisa

A pesquisa bibliográfica é um método de pesquisa que busca informações teóricas, permitindo com que pesquisadores aprofundem em conceitos do trabalho que está sendo desenvolvido. Este tipo de pesquisa permite que o pesquisador investigue conclusões apontadas por outros autores (?). Este TCC utilizou-se da pesquisa bibliográfica pois foi realizada pesquisa em artefatos acadêmicos, tais como livros, TCCs e artigos, para aprofundamento dos conhecimentos necessários para condução deste trabalho. As pesquisas que compõem o acervo bibliográfico foram encontradas em bibliotecas eletrônicas, tais como "Google Acadêmico" e *sites* de busca.

A metodologia de desenvolvimento utilizada neste TCC é baseada no UP. O processo unificado é composto por um conjunto de atividades que transforma requisitos do usuário em requisitos de sistema de *software*. Segundo (?), UP é baseado em componentes e consiste na repetição de ciclos durante as fases de concepção, elaboração, construção (implementação) e transição. Na fase de concepção e elaboração do UP foi realizado a extração e definição dos requisitos do sistema, através da técnica de entrevista. Ainda durante essas fases, foram elaborados e descritos os casos de uso, diagrama de classes, atividade e de sequência. Na fase de construção, os módulos executáveis do sistema foram implementados. A fase de transição não foi executada.

A arquitetura de *software* escolhida para o desenvolvimento do sistema *Web* foi a MVC, no qual a aplicação é separada em 3 (três) camadas. Por ser uma arquitetura de fácil manutenção, onde aceita atualização em suas camadas e de rápido desenvolvimento sendo também uma arquitetura que constrói sistemas mais leves.

O sistema CARVAVI segue este método dentro do seu código desenvolvido onde ele é separado as suas camadas sendo *model*, *view* e *controller*, onde foi construído independentes porém tem suas conexões que pode ser visualizado no *Github*¹.

¹<https://github.com/Yamilla/carvavi>

3.2 Ferramentas Utilizadas

Para a criação dos diagramas de UML presentes no Capítulo 4 foi utilizado o *software Draw.io*². Este programa é um editor gráfico *online* que permite a construção de desenhos, gráficos, fluxogramas da UML, diagramas, entre outros. Sua escolha foi pelo fato de ser gratuito e *online*, onde não há necessidade de realizar o *download* de um *software*.

Para o desenvolvimento dos códigos usamos a *Integrated Development Environment* (IDE) Visual Studio³, na versão 1.72.2. O motivo de sua escolha foi por ser uma IDE de *software* livre e também por ter várias ferramentas em si que ajudam no desenvolvimento dos códigos. Para possibilitar o teste das páginas *Web* foi utilizado o servidor XAMPP⁴, em sua versão 8.1.10-1. Sua escolha foi pelo motivo do XAMPP criar um servidor na própria máquina agilizando o teste das páginas. Para armazenamento e manipulação dos dados foi utilizado SGBD *phpMyAdmin*, na versão 5.2.0. Foi escolhido por ser gratuito e ter uma interface mais simplificada para o uso. Para desenvolvimento *front-end* foi utilizado o *Bootstrap* na versão 5.0. O *bootstrap* foi escolhido por já ter vários *templates* prontos e também permitindo a personalização de acordo com a necessidade. Já para o desenvolvimento *back-end* foi utilizada a linguagem de programação PHP, na versão 8.1.10. O PHP foi escolhido por ser uma linguagem muito utilizada com uma facilidade de aprendizado além de ser gratuita.

Para realizar armazenamento, versionamento e disponibilização do código utilizamos o *Github*⁵. Foi usado o *github* pela sua facilidade para a disponibilização dos códigos do sistema. Todas as ferramentas utilizadas foram usadas no sistema operacional *Windows 11*, na versão 22H2.

²<https://app.diagrams.net/>

³<https://visualstudio.microsoft.com/pt-br/downloads/>

⁴https://www.apachefriends.org/pt_br/index.html

⁵<https://github.com/Yamilla/carvavi>

4

CARVAVI: Análise e Documentação

4.1 Metodologia de Desenvolvimento

O desenvolvimento do sistema iniciou-se com o levantamento de requisitos. A seguir, para aprofundamento dos requisitos elencados, diversos diagramas da UML foram modelados. Os artefatos de *software* construídos foram:

- Entrevista;
- Requisitos Funcionais;
- Requisitos Não-Funcionais;
- Diagrama de Casos de Uso;
- Documentação dos Casos de Uso;
- Diagrama de Atividades;
- Diagrama de Sequência;
- Diagrama de Classes;
- Diagrama Entidade-Relacionamento.

4.1.1 Entrevista

Para entender como é o funcionamento de uma sala de vacinação foi realizada uma entrevista com duas técnicas de enfermagem que trabalham em uma UBS. As respostas da entrevista, presentes no Apêndice A, mostram o funcionamento dos processos de vacinação. As perguntas abrangem a rotina dos profissionais e os processos utilizados para organização das vacinas, bem como para controle das doses e registro de aplicação. Com base nas respostas, alguns requisitos de sistema foram definidos. A seguir, mostramos os requisitos funcionais do sistema CARVAVI.

4.1.2 Requisitos Funcionais

Os requisitos foram classificados em 3 (três) categorias distintas: *essencial*, *importante* ou *desejável*. Um requisito *essencial* é aquele a qual o sistema não funciona sem sua existência. Portanto, requisitos essenciais são necessários, e devem ser implementados para o sistema minimamente funcionar.

Já o requisito classificado como *importante* não compromete o funcionamento do sistema caso não seja devidamente implementado. Porém, no caso da não implementação dos requisitos importantes, algumas funcionalidades deixam de ser oferecidas aos usuários do sistema. Finalmente, os requisitos desejáveis são aqueles que possuem menor importância quando comparados aos requisitos essenciais e importantes, não afetando o sistema caso não (ou nunca) sejam implementados. A seguir, mostramos os requisitos funcionais, abreviados pela sigla RF, do sistema de controle de vacinação e suas devidas classificações.

- **[RF001] Cadastrar Unidade Básica de Saúde.**

- **Classificação:** Essencial Importante Desejável

- **Descrição:** O sistema deve permitir que possa ser feito o cadastro de unidade básica de saúde.

- **[RF002] Cadastrar Técnico de Enfermagem.**

- **Classificação:** Essencial Importante Desejável

- **Descrição:** O sistema deve permitir que possa ser feito o cadastro de técnico de enfermagem.

- **[RF003] Cadastrar Paciente.**

- **Classificação:** Essencial Importante Desejável

- **Descrição:** O sistema deve permitir que possa ser feito o cadastro de um paciente.

- **[RF004] Cadastrar Vacina.**

- **Classificação:** Essencial Importante Desejável

- **Descrição:** O sistema deve permitir que seja realizado o cadastro de vacina com nome, laboratório, data de fabricação, data de validade e lote.

- **[RF005] Incluir Vacinas.**

- **Classificação:** Essencial Importante Desejável

- **Descrição:** O sistema deve permitir que possa ser incluídas vacinas aos pacientes cadastrados.

- **[RF006] Gerar Cartão de Vacina.**
 - **Classificação:** Essencial Importante Desejável
 - **Descrição:** O sistema deve permitir que o usuário gere um arquivo *Portable Document Format* (PDF) do cartão de vacinas.

- **[RF007] Cadastrar Campanhas de Vacinação**
 - **Classificação:** Essencial Importante Desejável
 - **Descrição:** O sistema deve permitir possa ser feito o cadastro campanhas de vacinação que estão sendo ofertadas na unidade básica de saúde.

- **[RF008] Alertar Campanhas de Vacinação**
 - **Classificação:** Essencial Importante Desejável
 - **Descrição:** O sistema deve enviar um alerta aos pacientes que está tendo uma campanha de vacinação.

- **[RF009] Notificar as Doses da Vacinas**
 - **Classificação:** Essencial Importante Desejável
 - **Descrição:** O sistema deve notificar aos pacientes que tem doses de vacinas para serem tomadas.

- **[RF010] Mostrar as Unidades Básicas de Saúde Próximas ao Endereço do Paciente**
 - **Classificação:** Essencial Importante Desejável
 - **Descrição:** O sistema deve mostrar ao paciente as unidades básicas de saúde mais próxima do seu endereço.

- **[RF011] Alertar Vacinas com Validade Próxima**
 - **Classificação:** Essencial Importante Desejável
 - **Descrição:** O sistema deve informar que tem vacinas perto da data de validade.

4.1.3 Requisitos Não-Funcionais

Os requisitos não-funcionais, abreviados pela sigla NF, e também pela sequência numérica, da mesma forma que os requisitos funcionais são apresentados. Dessa forma, segue requisitos não-funcionais do sistema de controle de vacinação:

- **[NF001] Segurança:** O sistema deverá usar criptografia para guardar dados sensíveis (senha) dos usuários.

- **[NF002] Portabilidade:** O sistema deverá rodar em qualquer plataforma com acesso à *Internet*.

- **[NF003] Implementação:** O sistema deverá ser implementado na linguagem de programação PHP.

- **[NF004] Interoperabilidade:** O sistema deverá se comunicar com o banco de dados MySQL.

4.1.4 Artefatos da UML

Foi desenvolvido um conjunto de artefatos da UML para modelagem do sistema CARVAVI. Nesta Seção apresentaremos os artefatos construídos.

4.1.4.1 Diagrama de Casos de Uso

O diagrama de casos de uso mostra as funcionalidades que cada usuário pode realizar dentro do sistema. Cada usuário (ator) é representado por um boneco e as interações possíveis são representadas por balões. Foi construído 1 (um) diagrama de casos de uso para representar os atores administrador, técnico de enfermagem e paciente. O ator administrador é capaz de efetuar *login*, cadastrar UBS e cadastrar técnico de enfermagem, como pode ser observado na Figura 4.1.

O ator técnico de enfermagem é capaz de efetuar *login*, cadastrar vacinas e pacientes, incluir vacinas, isto é, realizar a vacinação de um determinado paciente, e gerar o cartão de vacina. Note na Figura 4.1 o ator técnico de enfermagem.

Finalmente, ator paciente, apenas a geração do cartão de vacinas é permitida, após devido autenticação, como pode ser observada na Figura 4.1.

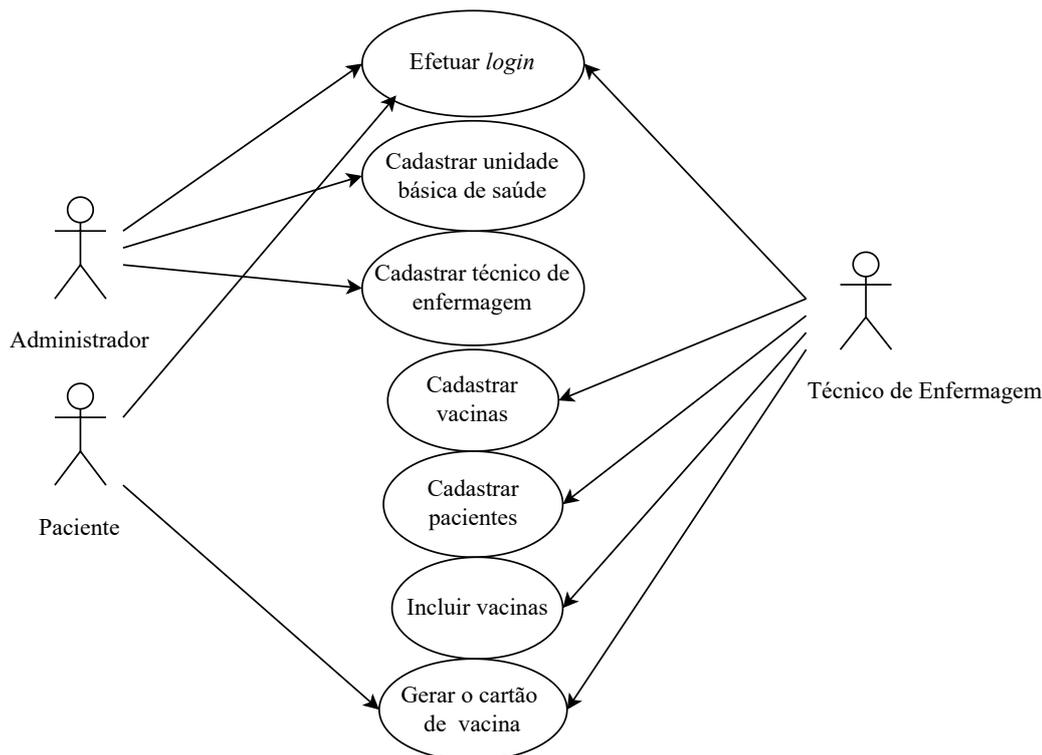


Figura 4.1: Diagrama de Casos de Uso

4.1.4.2 Documentação dos Casos de Uso

Esta subseção mostra a especificação (documentação) completa e abstrata dos casos de uso descritos na Figura 4.1. O objetivo da documentação dos casos de uso é mostrar detalhes adicionais relacionados, tais como atores, pré e pós condições, restrições e ações realizadas pelo usuário ou sistema.

A Tabela 4.1 mostra o caso de uso relacionado ao cadastro de UBS. Note na Tabela que o ator principal deste caso de uso é o usuário administrador. A pré-condição se remete à necessidade do administrador estar cadastrado e autenticado no sistema. Em relação a pós-condição, destaca-se a permissão de uso da UBS devidamente cadastrada. É importante notar que a UBS cadastrada poderá ser usada no momento que for realizado o cadastro de um técnico de enfermagem. Basicamente, o ator precisa preencher os dados da UBS e o sistema irá salvar as informações junto ao BD.

A Tabela 4.2 mostra a especificação do caso de uso cadastrar técnico de enfermagem. Neste caso, o administrador é o ator principal. Note que, para cadastrar um técnico de enfermagem, é necessário a satisfação da pré-condição relacionada a existência de UBS previamente cadastrada. Após o preenchimento dos dados do técnico de enfermagem pelo ator administrador, juntamente com o vínculo do usuário com a UBS, o sistema irá salvar essas informações.

O caso de uso efetuar *login* é mostrado na Tabela 4.3. Neste caso, os atores principais são todos disponíveis no sistema CARVAVI, isto é, administrador, técnico de enfermagem e paciente. Para efetuar *login*, os usuários, independente do seu tipo, devem estar previamente cadastrados.

Tabela 4.1: Documentação do Caso de Uso - Cadastrar Unidade Básica de Saúde.

Nome do caso de uso	Cadastrar unidade básica de saúde
Ator principal	Administrador
Resumo	Esse caso de uso descreve como a unidade básica de saúde será cadastrada
Pré-condições	O Administrador deve estar autenticado no sistema
Pós-condições	Ser permitido usar a unidade básica de saúde no momento que é realizado o cadastro de técnico de enfermagem
Ações do Ator	Ações do sistema
	1. Abrir o painel de formulário de cadastro
2. Preencher os dados	
	3. Salvar as informações
Restrições	Não é permitido o cadastro com campos nulos.

Tabela 4.2: Documentação do Caso de Uso - Cadastrar Técnico de Enfermagem.

Nome do caso de uso	Cadastrar técnico de enfermagem
Ator principal	Administrador
Resumo	Esse caso de uso descreve como o técnico de enfermagem será cadastrado no sistema
Pré-condições	Necessária existência de unidade básica de saúde no sistema
Pós-condições	Liberação de uso do sistema para o técnico de enfermagem.
Ações do Ator	Ações do sistema
	1. Abrir o painel de formulário de cadastro
2. Preencher os dados	
	3. Salvar as informações
Restrições	Não é permitido cadastrar técnico de enfermagem se não houver unidade básica de saúde no sistema.

Dessa forma, são verificados os dados de usuário e senha junto ao banco de dados para que o *login* seja permitido. A Tabela 4.4 especifica o caso de uso cadastrar vacinas. Este caso de uso

Tabela 4.3: Documentação do Caso de Uso - Efetuar *Login*.

Nome do caso de uso	Efetuar <i>login</i>
Ator principal	Administrador/Técnico de Enfermagem/ Paciente
Resumo	Esse caso de uso descreve como será efetuado o <i>login</i> no sistema.
Pré-condições	O usuário deve estar cadastrado no sistema.
Pós-condições	Será liberado o uso do sistema.
Ações do Ator	Ações do sistema
1. Entrar no sistema com o usuário e senha	
	2. Verifique se este usuário está cadastrado.
	3. Abrir o sistema para o usuário.
Restrições	Os dados de usuário e senha devem estar corretos.

pode ser realizado apenas pelo usuário técnico de enfermagem que precisa estar previamente cadastrado e autenticado no sistema. Não é permitido a omissão dos dados sensíveis das vacinas, tais como nome, laboratório, lote, data de validade, data de fabricação e quantidade disponível.

Tabela 4.4: Documentação do Caso de Uso - Cadastrar Vacinas.

Nome do caso de uso	Cadastrar vacinas
Ator principal	Técnico de Enfermagem
Resumo	Esse caso de uso descreve como será o cadastro das vacinas.
Pré-condições	O técnico de enfermagem deve estar autenticado no sistema.
Pós-condições	Permitir adicionar descrição na vacina
Ações do Ator	Ações do sistema
	1. Abrir o painel de formulário de cadastro de vacinas
2. Preencher o formulário	
	3. Verificar se formulário foi preenchido como deve
	4. Salvar os dados.
Restrições	Não é permitido o cadastro com campos nulos.

A Tabela 4.5 mostra a documentação do caso de uso cadastrar pacientes. Os pacientes do sistema CARVAVI são cadastrados exclusivamente por técnicos de enfermagem que devem estar devidamente autenticados. Como acontece na especificação do caso de uso de cadastrar vacinas (Tabela 4.4), não é permitido a inserção de dados vazios.

Tabela 4.5: Documentação do Caso de Uso - Cadastrar Pacientes.

Nome do caso de uso	Cadastrar pacientes
Ator principal	Técnico de Enfermagem
Resumo	Este caso de uso de descreve como será o cadastro dos pacientes.
Pré-condições	O técnico de enfermagem deve ser estar autenticado no sistema.
Pós-condições	Será permitido incluir vacinas
Ações do Ator	Ações do sistema
	1. Abrir o painel de formulário de cadastro
2. Preencher o formulário	
	3. Verifique se as informações são as solicitadas
	5. Salvar as informações
Restrições	Não é permitido o cadastro com campos nulos.

A Tabela 4.6 mostra a especificação do caso de uso incluir vacinas. Este caso de uso possui como ator principal o técnico de enfermagem. Neste caso, este caso de uso descreve como serão cadastradas as vacinas serão aplicadas aos pacientes. Para registrar uma vacina para um determinado paciente é necessário previamente realizar o cadastro do paciente por um técnico de enfermagem. As vacinas são aplicadas de acordo com seguinte fluxo: primeiro, abre-se o menu pacientes. Ao escolher o paciente que será aplicado a vacina, deve-se clicar em vacinar. Neste ponto, o ator técnico de enfermagem deve incluir as vacinas que serão aplicadas. As restrições relacionadas a este caso de uso relacionam-se com a existência de paciente previamente cadastro e a disponibilidade da dose de vacina a ser aplicada.

A Tabela 4.7 mostra a especificação do caso de uso gerar o cartão de vacina. Neste caso

Tabela 4.6: Documentação do Caso de Uso - Incluir Vacinas

Nome do caso de uso	Incluir vacinas
Ator principal	Técnico de Enfermagem
Resumo	Esse caso de uso descreve como registrar a aplicação de uma vacina a um paciente.
Pré-condições	O técnico de enfermagem deve ser feito o cadastro do paciente no sistema.
Pós-condições	Será permitido gerar um arquivo PDF do cartão de vacina
Ações do Ator	Ações do sistema
	1. Abrir o menu paciente
2. Escolher o paciente que será vacinado	
3. Clicar em vacinar.	
4. Incluir vacinas que serão aplicadas	
	5. Salvar os dados.
Restrições	O paciente deve estar cadastrado e deve haver disponibilidade de vacinas a serem aplicadas.

de uso, os atores responsáveis são o técnico de enfermagem e o paciente. Caso o paciente não tenha vacinas aplicadas, o cartão de vacinas é gerado apenas com o cabeçalho principal (dados do paciente). Não há restrições deste caso de uso. Como pós-condição, será possível visualizar o arquivo do cartão de vacinas no formato PDF.

Tabela 4.7: Documentação do Caso de Uso - Gerar o Cartão de Vacina.

Nome do caso de uso	Gerar o cartão de vacina
Ator principal	Técnico de Enfermagem/Paciente
Resumo	Esse caso de uso descreve como gerar o cartão vacina.
Pré-condições	O técnico de enfermagem e o paciente devem estar cadastrados no sistema.
Pós-condições	O cartão de vacina do paciente poderá ser visualizado.
Ações do Ator	Ações do sistema
	1. Abrir o campo de cartão de vacina
2. Clicar em gerar o cartão de vacina	
	3. Finalizar uma sessão
Restrições	Não há

4.1.4.3 Diagrama de Atividades

O diagrama de atividades ilustra o processo de negócio ou fluxo de trabalho entre usuários e sistema. O usuário administrador primeiramente deve efetuar o *login* para que seja possível a execução de outras atividades inerentes ao seu perfil. Após *login*, o administrador pode cadastrar técnico de enfermagem ou cadastrar UBS. A partir do *login* do usuário administrador também é possível atualizar os dados cadastrados, bem como realizar sua exclusão, conforme Figura 4.2.

Em relação ao usuário técnico de enfermagem, nota-se que após o *login*, é possível cadastrar vacinas ou pacientes. É possível alterar dados relacionados aos pacientes, bem como

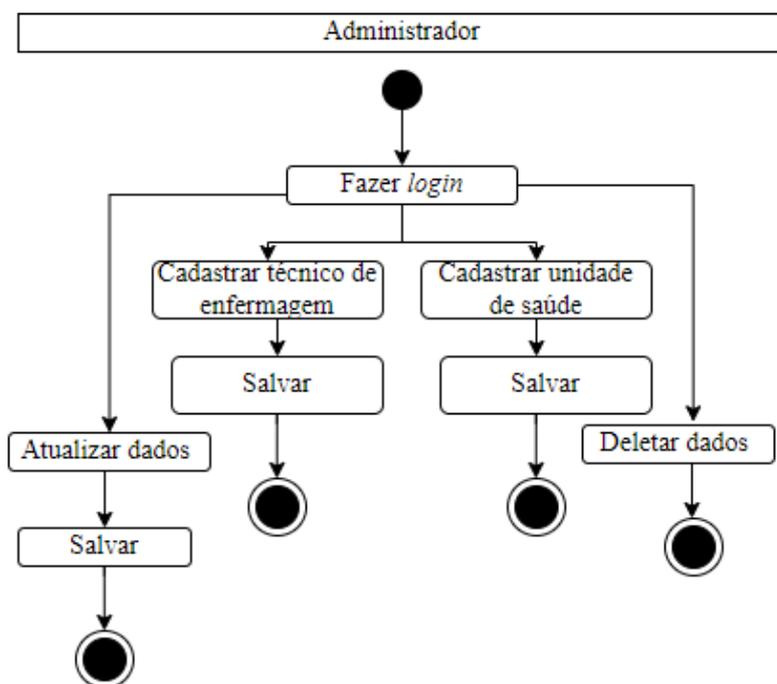


Figura 4.2: Diagrama de Atividades - Administrador

realizar sua exclusão. Ademais, o usuário técnico de enfermagem pode imprimir o cartão de vacina de um determinado paciente. O diagrama de atividades do usuário técnico de enfermagem pode ser observado na Figura 4.3.

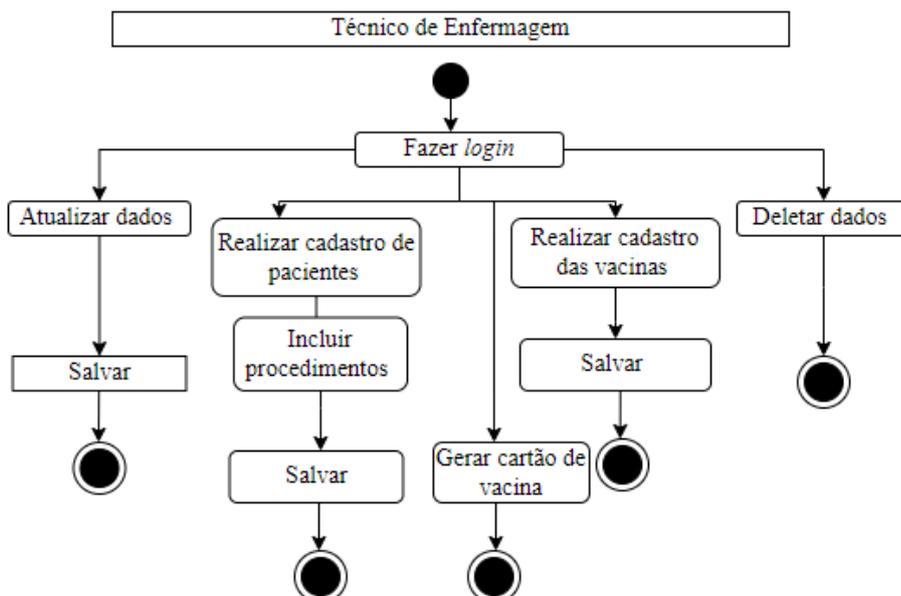


Figura 4.3: Diagrama de Atividades - Técnico de Enfermagem

O usuário do tipo paciente possui funções mínimas no contexto do sistema de controle de vacinação. Para este usuário, após devido *login* é permitido apenas a visualização de dados pessoais e emissão do cartão de vacina em formato PDF, como pode ser observado na Figura 4.4.

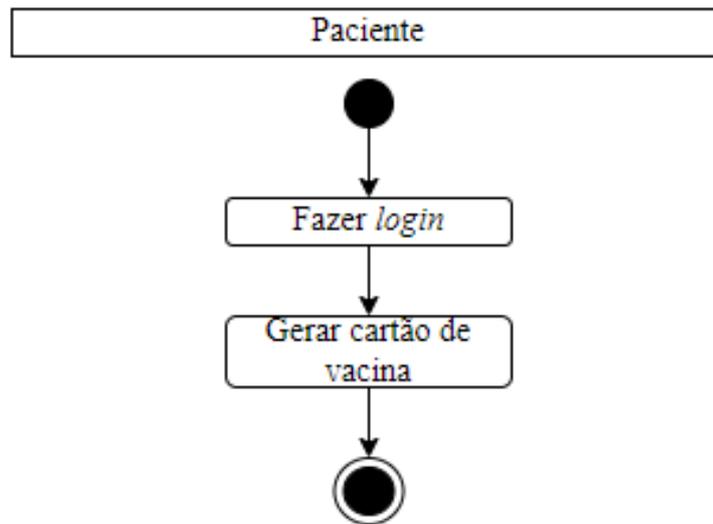


Figura 4.4: Diagrama de Atividades - Paciente

4.1.4.4 Diagrama de Sequência

O diagrama de sequência é uma solução dinâmica de modelagem da UML que mostra troca de mensagens do sistema, com base na interação entre processos e componentes, antes do final da linha da vida dos mesmos. Neste sentido, e considerando os 3 (três) principais atores do sistema de vacinação: administrador, técnico de enfermagem e paciente, foram construídos diagramas independentes por autor que modelam a interação de troca de informações entre ator, *site* e base de dados.

Como é mostrado na Figura 4.5, no diagrama de sequência relacionado ao ator administrador, a requisição inicial é relacionada ao *login*, por meio de usuário e senha previamente cadastrados. Após, é disparada a função *validarLoginadmin()* para a base de dados para autenticação do usuário. Quando o administrador realiza o cadastro de uma unidade e de um técnico de enfermagem, a função de *insert()* é invocada para inserção dos dados correspondentes retornando a mensagem de cadastro realizado com sucesso. O usuário administrador pode visualizar unidades cadastradas. Essa ação inicializa a função *getAllDataTable()* que retorna todas unidades cadastradas, após consulta na base de dados. O administrador pode atualizar UBSs cadastradas ou realizar sua exclusão. Essas ações disparam as funções *update()* e *delete()*, respectivamente. De forma análoga, a atualização dos técnicos de enfermagem cadastrados, bem como a exclusão dessa informação também é permitida.

A Figura 4.6 mostra o diagrama de sequência desenvolvido para o ator Técnico de Enfermagem. Por questões de simplicidade, focamos na explicação do diagrama nas ações que diferem o usuário Técnico de Enfermagem para o usuário Administrador. Neste sentido, ao realizar o cadastro ou a descrição de uma vacina, a função *insert()* é invocada para que os dados sejam adicionados a base de dados. Primeiramente é adicionado um nome da vacina e, em seguida, é solicitado dados adicionais. Ao visualizar as vacinas cadastradas, *visualizarVacina()* é responsável por consultar os dados no banco e retornar a consulta. O ator Técnico de enfermagem

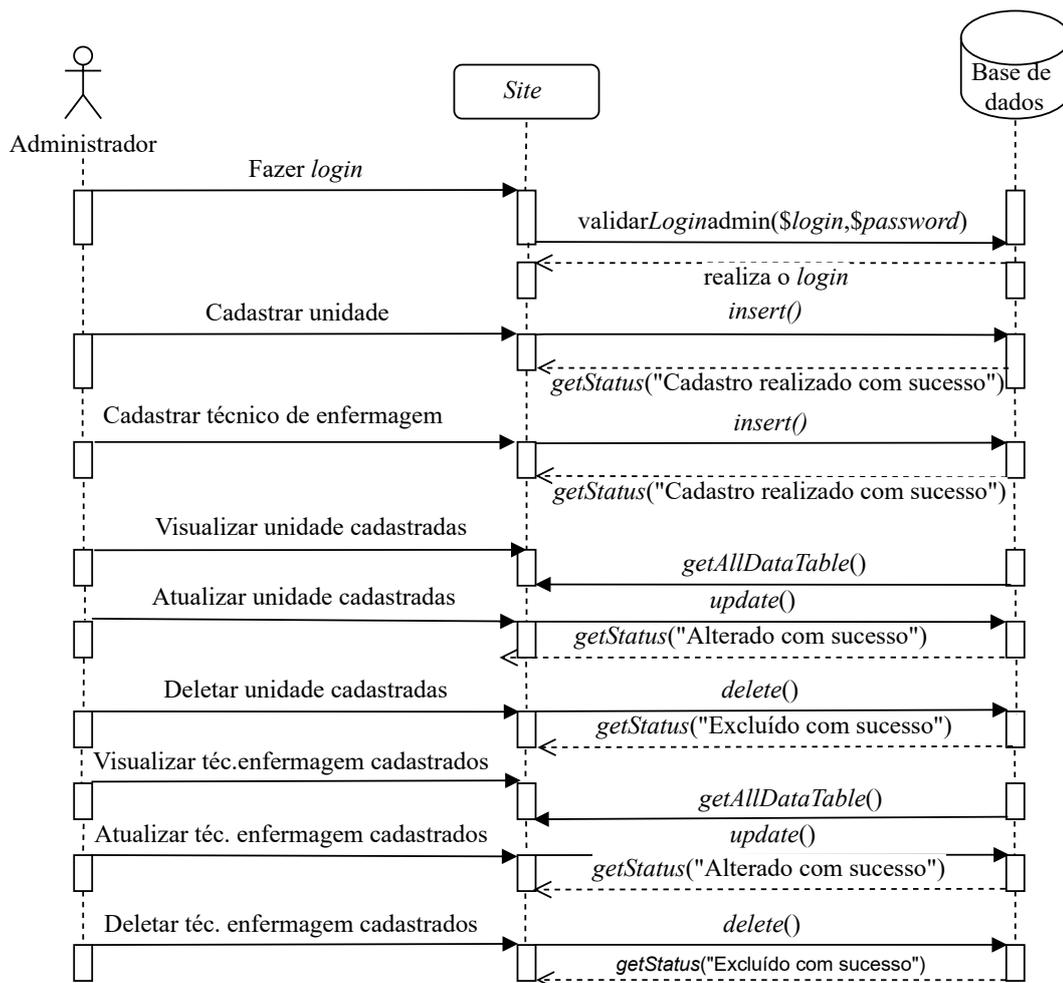


Figura 4.5: Diagrama de Sequência - Ator Administrador

pode cadastrar pacientes para que sejam vacinados. Após visualizar os pacientes cadastrados é possível vaciná-los. Ao vacinar pacientes cadastrados, a função *vacinasAplicadas()* é invocada para que seja realizado o controle das vacinas aplicadas ao paciente bem como a quantidade de doses disponíveis na UBS. Finalmente, o ator Técnico de Enfermagem pode gerar cartão de vacina. Essa funcionalidade dispara as função *getById()* e *cartaoVacina()* responsáveis por gerar o cartão de vacina do paciente com a descrição de dados pessoais e vacinas recebidas. Para o ator Paciente, após *login*, é possível ver dados pessoais e gerar cartão de vacina. Essas ações disparam a função de *validarLoginPaciente()* e as funções *getById()* e *cartaoVacina()*, respectivamente. Veja na Figura 4.7 o diagrama de sequências para o ator Paciente.

4.1.4.5 Diagrama de Classes

Diagrama de classes é a representação gráfica das classes com seus atributos, métodos e relacionamentos. No desenvolvimento deste artefato de *software*, cada tabela é dividida em 3 (três) partes. A primeira parte apresenta o nome da classe. Já a segunda parte mostra seus atributos, isto é, as propriedades semelhantes dos objetos que a instanciam. Finalmente na

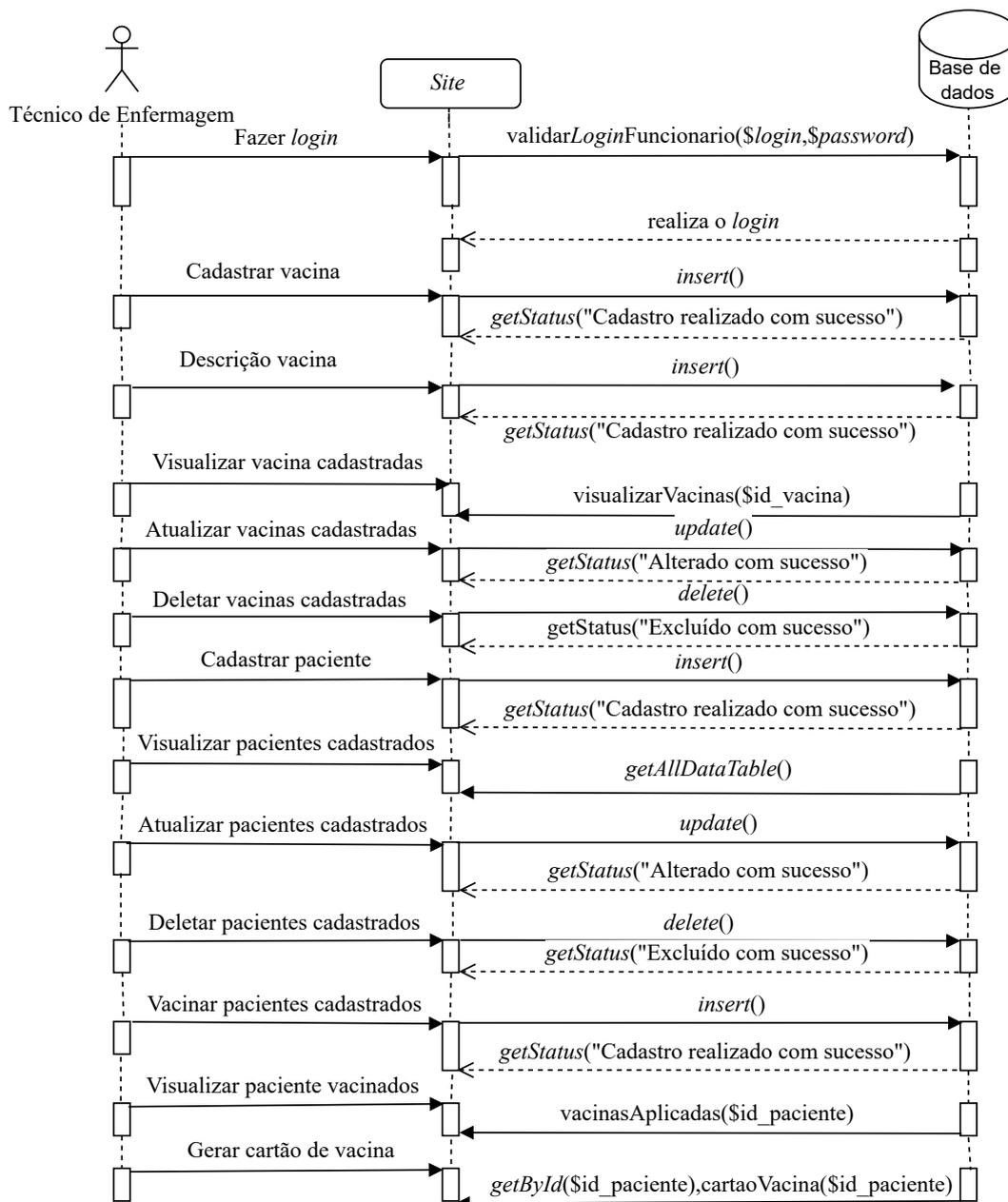


Figura 4.6: Diagrama de Sequência - Ator Técnico de Enfermagem

terceira parte apresentam-se as operações (métodos) que essas classes poderão realizar. As classes do sistema de vacinação CARVAVI relacionam-se entre si com diferentes cardinalidades.

Como é mostrado na Figura 4.8, o diagrama é composto por 7 (sete) classes, a citar: Administrador, Funcionario, Paciente, Vacina, Descricao_Vacina, Unidade_de_Saude e Vacinas_Aplicadas. A classe Administrador possui os atributos *id*, *email*, *usuário* e *senha*. O atributo *id* é do tipo inteiro, enquanto o restante dos atributos são do tipo *string*. Os métodos disponíveis na classe Administrador são: *login()*, *cadastrar()*, *deletar()*, *atualizar()* e *consultar()*. A classe Administrador relaciona-se com as classes Funcionario e Unidade_de_Saude com a cardinalidade 1-n, ou seja, 1 para muitos.

A classe Funcionario possui os atributos *id*, *nome*, *cpf*, *rg*, *orgao_expedidor*, *telefone*,

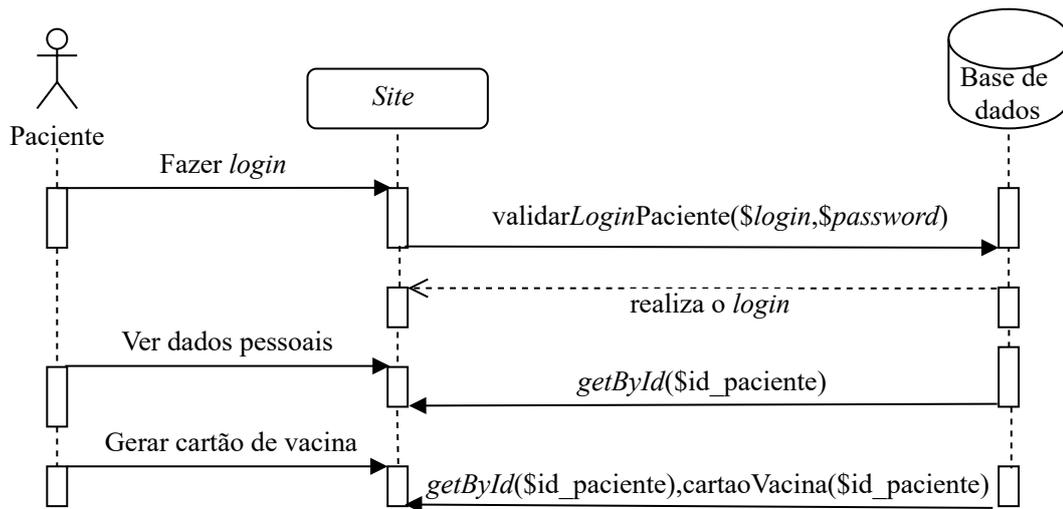


Figura 4.7: Diagrama de Sequência - Ator Paciente

data_nascimento, rua, numero, bairro, cidade, estado, cep, cargo, email, usuario e senha. O atributo *id* é do tipo inteiro e os outros atributos desta classe são do tipo *string*. Os métodos presentes são: *login()*, *cadastrar()*, *deletar()*, *atualizar()* e *consultar()*. A classe *Funcionario* relaciona-se com a classe *Paciente* com a cardinalidade n-n, ou seja, muitos para muitos. Já a classe *Unidade_de_Saude* possui os atributos *id, nome, rua, numero, bairro, cidade, estado, cep e numero*. O atributo *id* é do tipo inteiro e outros são do tipo *string*. Os métodos dentro dessa classe são: *cadastrar()*, *deletar()*, *atualizar()* e *consultar()*. A classe *Unidade_de_Saude* se relaciona com a classe *Funcionario* e sua cardinalidade é n-n, ou seja, muitos para muitos.

A classe *Paciente* possui os mesmos atributos da classe *Funcionario*, exceto *cargo*. Adicionalmente, esta classe possui o atributo *sus* que identifica o número do cartão do paciente. Os métodos dessa classe são: *login()*, *consultar()*, *cadastrar()*, *atualizar()*, *deletar()*. A classe *Paciente* se relaciona com a classe *Vacinas_Aplicadas* com a cardinalidade n-n, ou seja, muitos para muitos. Já a classe *Vacina* possui os atributos *id e nome*. O atributo *id* é do tipo inteiro e *nome* do tipo *string*. Os métodos presentes nessa classe são: *consultar()* e *cadastrar()*. A classe *Vacina* se relaciona com as classes *Descricao_Vacina* e *Vacinas_Aplicadas* com a cardinalidade é n-n, ou seja, muitos para muitos. Já a classe *Descricao_Vacina* possui os atributos *id lote, quantidade, data_fabricacao, data_validade, laboratorio*. Os métodos presentes nessa classe são: *cadastrar()*, *deletar()*, *atualizar()* e *consultar()*. Finalmente, a classe *Vacinas_Aplicadas* possui os atributos *id, nome, lote, data_aplicacao, dose e laboratorio*. Os métodos são idênticos a classe *Descricao_Vacina*.

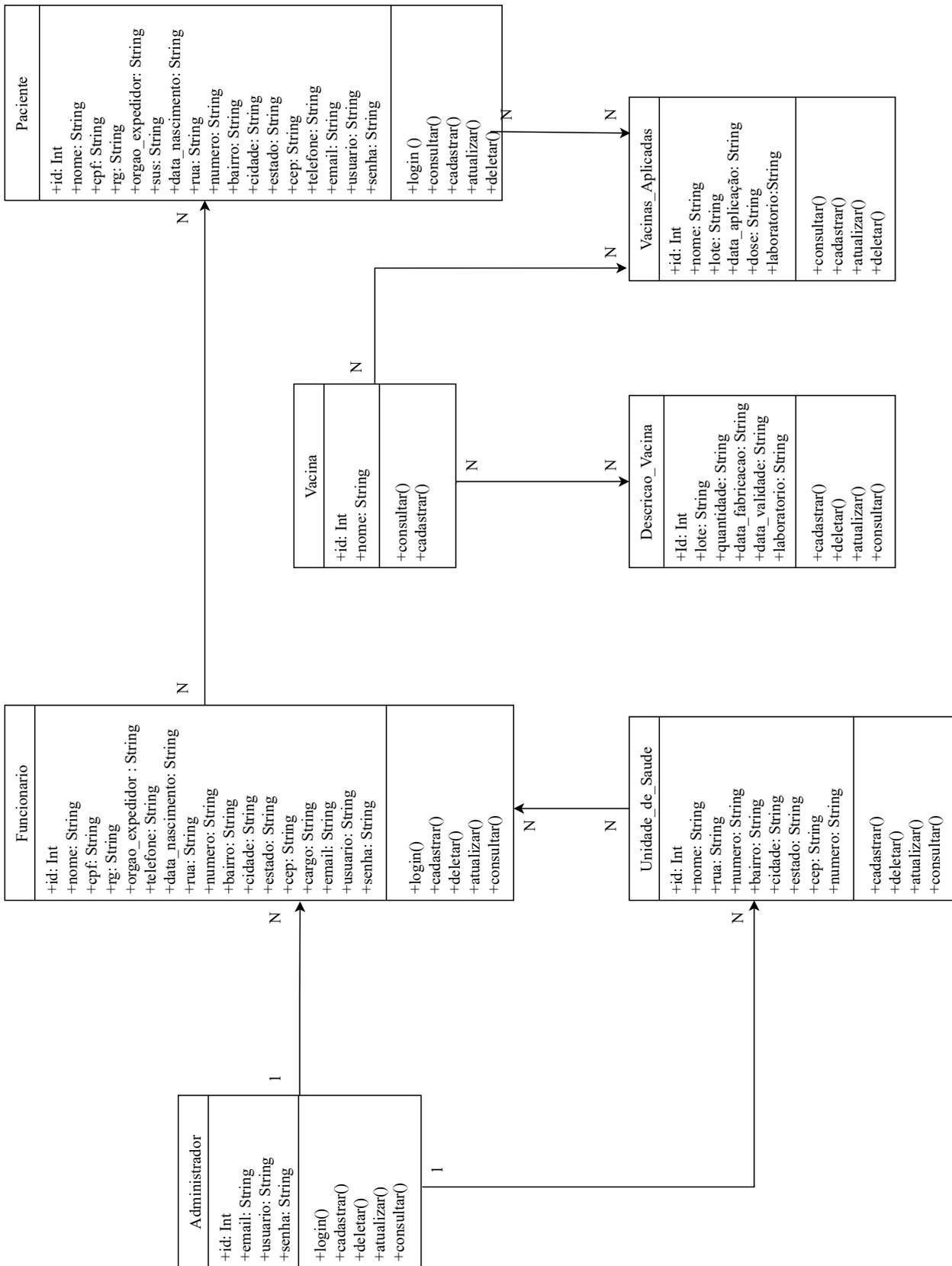


Figura 4.8: Diagrama de Classes

4.1.5 Diagrama Entidade-Relacionamento

O diagrama entidade-relacionamento é construído por conjuntos gráficos marcados pela presença de entidades, relacionamentos com suas cardinalidades e seus atributos. O intuito do diagrama entidade-relacionamento é apresentar um esquema visual do banco de dados. Pode observar na subseção 4.1.4.5 encontra-se o diagrama de classe, onde é um complemento do diagrama entidade-relacionamento. Como pode ser observado na Figura 4.9, o diagrama entidade-relacionamento é constituído por 7 (sete) entidades, a citar: administrador, funcionario, unidade_de_saude, paciente, vacina, descricao_vacina e vacinas_aplicadas.

A entidade administrador contém o atributo chave *id_administrador* e os outros atributos *email*, *usuario* e *senha*. A entidade tem o relacionamento cadastra com a entidade unidade_de_saude e a entidade funcionario com cardinalidade um-para-muitos. Já a entidade funcionario contém o atributo chave *id_funcionario* e os outros atributos *nome*, *cpf*, *rg*, *orgao_expedidor*, *telefone*, *data_nascimento*, *rua*, *numero*, *bairro*, *cidade*, *estado*, *cep*, *cargo*, *email*, *usuario* e *senha*. A entidade funcionario tem o relacionamento trabalha com a entidade unidade_de_saude com cardinalidade um-para-muitos. Ademais, funcionario relaciona-se com as entidades paciente, vacinas_aplicadas, descricao_vacina e vacina com cardinalidades um-para-muitos.

A entidade unidade_de_saude possui o atributo chave *id_unidade* e os outros atributos *nome*, *rua*, *numero*, *bairro*, *cidade*, *estado*, *cep*, *numero_unidade*. A entidade vacina possui o atributo chave *id_vacina* e o atributo *nome*. A entidade descricao_vacina possui o atributo chave *id_descricao_vacina* e os outros atributos *laboratorio*, *lote*, *quantidade*, *data_fabricacao* e *data_validade*. A entidade paciente possui atributo chave *id_paciente* e os outros atributos *nome_paciente*, *cpf*, *rg*, *orgao_expedidor*, *sus*, *telefone*, *nascimento*, *rua*, *numero*, *bairro*, *cidade*, *estado*, *cep*, *email*, *usuario* e *senha*. A entidade vacinas_aplicadas possui o atributo chave *id_vacinas_aplicadas*, *nome*, *lote*, *data_aplicacao*, *dose* e *laboratorio*.

De acordo com os relacionamentos ilustrados na Figura 4.9, a entidade administrador é capaz de cadastrar uma ou mais unidades básicas de saúde e um ou mais funcionários. Cada unidade básica de saúde pode ser cadastrada somente uma vez pela entidade administrador. Cada funcionário pode trabalhar em uma ou mais unidades básicas de saúde. Simultaneamente, o funcionário pode cadastrar uma ou mais vacinas e um ou mais pacientes. Os pacientes, por sua vez, podem ser cadastrados somente por um único funcionário. Cada vacina possui uma ou mais descrições. Porém, cada descrição é referente a uma única vacina. A entidade funcionário também é capaz de cadastrar vacinas em um ou mais pacientes, que por sua vez, possuem uma ou mais vacinas registradas.

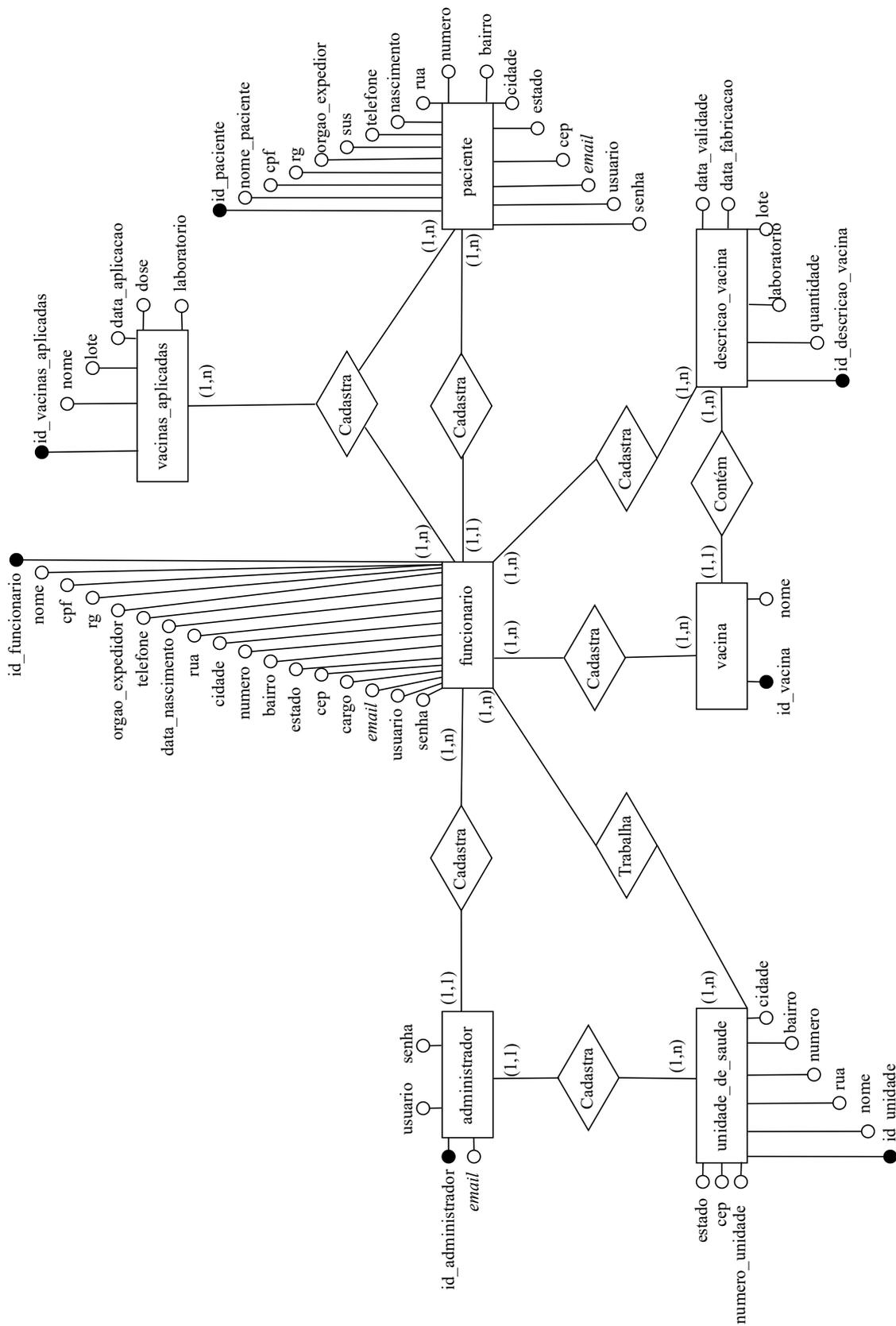


Figura 4.9: Diagrama Entidade-Relacionamento

4.2 CARVAVI: Visão Geral

CARVAVI é um sistema *Web* para controle do sistema de vacinação a ser utilizado em uma UBS, e que permite cadastro de unidades básicas de saúde, técnicos de enfermagem, pacientes e vacinas, bem como possibilita o registro da vacinação e emissão do cartão de vacinas de pacientes devidamente cadastrados. O sistema, desenvolvido seguindo o padrão MVC, pode ser compreendido através da perspectiva de 3 (três) atores distintos: administrador, técnico de enfermagem e paciente. Nesta Seção, documentamos principais pontos relacionados as diferentes áreas do sistema CARVAVI, bem como mostramos algumas das funcionalidades do sistema desenvolvido. Para mais informações, recomendamos visita ao repositório público no *GitHub*¹.

4.2.1 Área do Administrador

O usuário administrador é previamente cadastrado com usuário e senha no banco de dados. A senha padrão 123 deve-se ser imediatamente trocada após o primeiro *login* do usuário administrador. Dessa forma, após devida autenticação do usuário administrador, é possível: (i) cadastrar técnico de enfermagem, (ii) cadastrar UBS, (iii) visualizar funcionários cadastrados e (iv) visualizar unidades básicas de saúde cadastradas, podendo visualizar na Figura 4.5, onde as 3 (três) telas iniciais seguem o mesmo padrão. A seguir, mostramos como é realizado o cadastro de uma UBS.

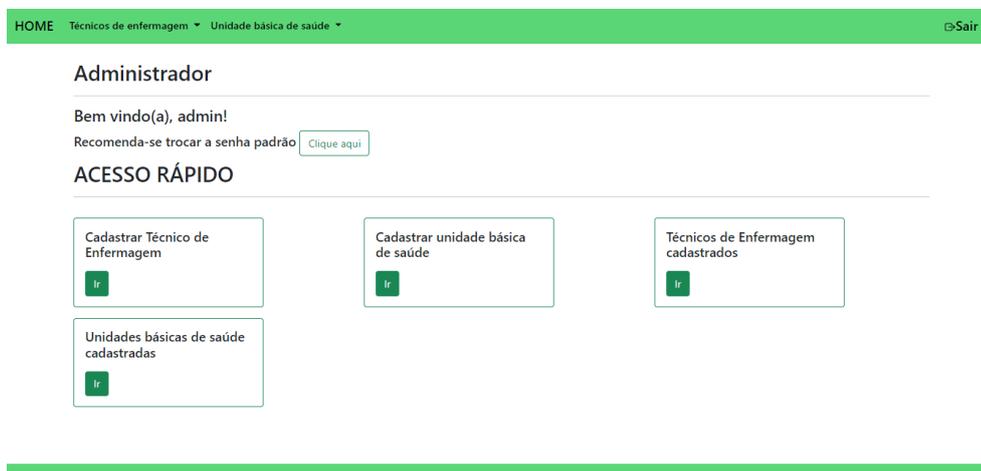


Figura 4.10: Área do Administrador

4.2.1.1 Cadastrar Unidade Básica de Saúde

O administrador deve clicar no botão *ir* que está no campo cadastrar UBS. Neste ponto, será aberta uma tela com um formulário a ser preenchido com dados da unidade. No final do formulário existem 2 (dois) botões: um para salvar e outro para cancelar. Importante salientar

¹<https://github.com/Yamilla/carvavi.git>

que os dados preenchidos somente serão salvos se todos os campos estiverem não-nulos. A Figura 4.11 mostra o processo de cadastro de UBS no sistema CARVAVI.

Cadastro de unidade básica de saúde

Nome:

Rua:

Número:

Bairro:

Cidade:

Estado:

CEP:

Número da unidade:

Figura 4.11: Cadastrar Unidade Básica de Saúde

4.2.1.2 Cadastrar Técnico de Enfermagem

O processo de cadastro de um técnico de enfermagem é semelhante ao processo descrito na Subseção 4.2.1.1. Excepcionalmente, no cadastro de técnico de enfermagem é necessário cadastrar usuário e senha para posterior acesso do mesmo.

4.2.1.3 Unidades Básicas de Saúde Cadastradas

Após clicar no botão ir, uma tela com as unidades básicas de saúde cadastradas será apresentada. A partir desta tela é possível alterar ou excluir unidades básicas de saúde, como pode ser observado na Figura 4.12. A exclusão da UBS somente é permitida se não houver pendência/registro de uso desta informação em outra área do sistema CARVAVI, isto é, se um determinado técnico de enfermagem estiver associado a esta UBS, a exclusão não será permitida.

Unidades básicas de saúde cadastradas

Nome	Rua	Número	Bairro	Cidade	Estado	CEP	Número da unidade	ALTERAR	EXCLUIR
unidade básica de saúde	00	00	teste	Formosa	Goiás	00000	0	<input type="button" value="Alterar"/>	<input type="button" value="Excluir"/>

Figura 4.12: Unidades Básicas de Saúde Cadastradas

4.2.1.4 Técnicos de Enfermagem Cadastrados

O processo de visualização dos técnicos de enfermagem cadastrados é semelhante ao procedimento presente na Subseção 4.2.1.3. Após clicar no botão ir que está no campo técnicos de enfermagem cadastrados, uma tela exibirá com todos técnicos de enfermagem que poderão ser alterados e/ou excluídos. Da mesma forma como descrito anteriormente, um técnico de enfermagem só poderá ser excluído se não estiver sendo usado por outra parte do sistema CARVAVI.

4.2.2 Área do Técnico de Enfermagem

A área do técnico de enfermagem é a área do sistema que possui mais funcionalidades pelo fato do sistema ser focado para uso do técnico de enfermagem que trabalha numa UBS. Nesta área o usuário poderá realizar o gerenciamento de pacientes e vacinas presentes na unidade. Também é possível registrar o processo de vacinação, bem como gerar o PDF do cartão de vacina dos pacientes previamente cadastrados.

4.2.2.1 Cadastrar Paciente

O processo de cadastro de um paciente é semelhante ao processo de cadastro de técnicos de enfermagem descrito na Subseção 4.2.1.2. Após clicar no botão ir, é possível preencher dados dos pacientes. Pacientes também possuirão usuário e senha para emissão do cartão de vacinas virtual. A seguir, é mostrado como uma vacina pode ser cadastrada no sistema CARVAVI.

4.2.2.2 Cadastrar Vacina

Após clicar no botão ir no campo de cadastrar vacina, o nome da vacina será exigido. Primeiramente, como pode ser observado na Figura 4.13, é necessário apenas cadastrar o nome da vacina. A seguir mostramos como é realizado o processo de inclusão de doses disponíveis de



Figura 4.13: Cadastrar Vacina

vacinas previamente cadastradas.

4.2.2.3 Vacinas Cadastradas

Após clicar no botão ir no menu vacinas cadastradas, será exibido uma lista de vacinas cadastradas. O botão incluir dados da vacina permite a inserção de detalhes adicionais das vacinas, como pode ser observado na Figura 4.14. É possível incluir lote, quantidade, data de fabricação, data de validade e laboratório. A seguir mostramos como é realizado o processo de vacinação de um determinado paciente.

HOME Pacientes Vacinas Sair

Descrição de vacinas

Lote:
Lote

Quantidade:
Quantidade

Data de fabricação:
dd/mm/aaaa

Data de validade:
dd/mm/aaaa

Laboratório:
Laboratório

Cadastrar Cancelar

Figura 4.14: Incluir Detalhes das Vacinas Cadastradas

4.2.2.4 Pacientes Cadastrados

Ao clicar no botão ir no menu pacientes será listado pacientes previamente cadastrados. Os pacientes poderão ser alterados ou excluídos com o processo semelhante ao descrito nas seções anteriores. Ademais, é possível registrar a vacinação de um paciente através do botão vacinar. Ao clicar no botão vacinar, dados da vacina serão requisitados. A vacinação será registrada no momento que os dados forem salvos.

4.2.2.5 Gerar Cartão de Vacina

Ao clicar em ir no menu cartão de vacina, os pacientes serão listados. É possível visualizar vacinas e gerar o cartão de vacinas de cada um dos pacientes. O botão de gerar o cartão de vacina executará uma chamada a biblioteca FPDF para geração em formato PDF do cartão de vacina. Os dados recuperados do banco de dados para emissão do cartão de vacina virtual são: (i) dados pessoais, (ii) vacinas aplicadas, (iii) UBS responsável pela aplicação, (iv) nome do profissional de enfermagem responsável pela aplicação e (v) data da vacinação. A Figura 4.15 mostra como o cartão de vacina é gerado para visualização e impressão.

CARTÃO DE VACINA VIRTUAL

DADOS PESSOAIS SUS: 0000.0000.0000.00

Nome: Joana Silva
 CPF: 000.000.000-00 RG: 000000 Orgão expedidor: SSP
 Endereço: 0 0 teste
 Cidade: Formosa Estado: Goiás CEP: 73802-120
 Data de nascimento: 1999-01-08 Telefone: (00) 0 0000-0000
 E-mail: myllacalixto13@gmail.com

VACINAS

NOME	LOTE	DOSE	LABORATÓRIO	DATA	UBS	ASS
Dengue	1449	1	flocruz	2022-10-26	Vila Carolina	Maria Pereira da Silva

Figura 4.15: PDF do Cartão de Vacina Virtual

4.2.3 Área do Paciente

O paciente acessa o sistema CARVAVI através de *login* e senha previamente cadastrados por um técnico de enfermagem. Após autenticação, é possível listar os dados pessoais. Neste momento, será possível clicar no botão gerar cartão de vacina. A geração do cartão de vacina pelo paciente permite que o mesmo visualize informações pessoais e de vacinação, semelhante ao processo descrito na Subseção 4.2.2.5.

5

Conclusão

A vacinação é um processo importante na vida do ser humano por ser a principal forma de prevenção de doenças. As vacinas são principalmente encontradas gratuitamente em unidades básicas de saúde, onde a população pode manter seu cartão de vacinas atualizados. Um problemática enfrentada dentro das salas de vacinas é que todos dados produzidos são armazenados em papéis, através do controle manual. Este fato possibilita erros, propositais ou não, e perda de dados sensíveis. Outro ponto que merece destaque é a perda dos próprios cartões de vacinas pelos pacientes.

Neste TCC foi desenvolvido um sistema *Web* para o controle de vacinação que tem a possibilidade de ser usado por 3 (três) nichos de usuários distintos. Utilizamos a técnica de entrevista como base na elucidação dos requisitos de *software* que foram classificados com diferentes propriedades: essencial, importante e desejável. Diferentes diagramas da UML foram projetados a fim de melhor entendimento do sistema. Ademais, mostramos as funcionalidades presentes no sistema de vacinação desenvolvido. Primeiramente, apresentamos uma visão geral do *software*. Em seguida, para cada um dos possíveis usuários do sistema, mostramos suas principais funcionalidades com destaque em características que julgamos mais importantes. Finalmente, mostramos também o repositório público construído no *GitHub*, bem como documentação para instalação, inicialização e uso.

Como trabalhos futuros pretende-se realizar implementação de recursos adicionais, tais como manipulação e envio de alertas de informações relacionadas as campanhas de vacinação. Outro recurso interessante é a implementação de envio de notificações relacionados as doses pendentes (vacinas em atraso), doses com prazo de validade próximo e a recomendação de unidades básicas de saúde de acordo com a localização geográfica do paciente.

Apêndice A

1. Como é a rotina dentro da sala de vacina?

- **Resposta da Pessoa 1:** Deixar a sala limpa , caixas térmicas e geladeira limpa e organizada, colocar os imunobiológicos nas caixas térmicas somente os necessários para o uso e evitando abrir a geladeira durante o dia , verificar o termômetro no horário correto.
- **Resposta da Pessoa 2:** Manter a sala arrumada , caixas térmicas e geladeira sempre muito bem limpa e organizada, fazer o manuseio dos imunobiológicos as caixas térmicas colocando o necessário a ser utilizado durante o período evitando abrir a geladeira mais de duas vezes ao dia, monitorar as leituras adequadas e nos horário corretos os termômetro.

2. Quais são os dados que devem ser coletados no momento da chegada das doses das vacinas na Unidade Básica de Saúde?

- **Resposta da Pessoa 1:** Quais os imunobiológicos, quantidade, lotes, laboratórios.
- **Resposta da Pessoa 2:** Quais os imunobiológicos, quantidade, lotes, laboratórios lembrando sempre de fazer rodízio de doses já existente na geladeira colocando sempre a frente para que seja usada primeiro.

3. Como é realizado o processo de cadastro do paciente que irá receber a vacina?

- **Resposta da Pessoa 1:** É realizada a ficha espelho.
- **Resposta da Pessoa 2:** É feita a ficha espelho.

4. Como é feita a organização das vacinas dentro da Unidade Básica das Saúde?

- **Resposta da Pessoa 1:** Armazenada em uma geladeira, separada por tipos das vacinas.
- **Resposta da Pessoa 2:** Armazenada em um refrigerador de uso exclusivo sendo que na grade superior armazena-se as vacinas virais e na inferior as bacterianas.

5. O que é feito com as informações que são coletadas no final da vacinação?

- **Resposta da Pessoa 1:** Elas ficam guardadas na unidade e são enviadas para o núcleo de vigilância epidemiológica.

- **Resposta da Pessoa 2:** Todas as informações além de ficar nas unidade também são enviadas ao núcleo de vigilância epidemiológica.
6. **Quais são as informações que são geradas entre os pacientes e os profissionais de saúde dentro da sala de vacina ?**
- **Resposta da Pessoa 1:** São solicitados todos os dados do paciente para fazer a ficha de vacinação, e são passadas orientações sobre reações da vacina.
 - **Resposta da Pessoa 2:** Além de todos os dados do paciente para preenchimento da ficha de vacinação, são orientados quanto às reações adversas que podem ser causadas pela vacina, cuidados e a importância de se manter o cartão de vacinas atualizado.
7. **Qual é o método informativo que é usado para informar para o paciente sobre quais vacinas estão presentes na Unidade Básica de Saúde?**
- **Resposta da Pessoa 1:** Pelos agentes comunitários de saúde, campanhas, Tv, rádio, internet.
 - **Resposta da Pessoa 2:** Através dos agentes comunitários de saúde, redes sociais quanto às vacinas de campanha além dos agentes comunitários de saúde, redes sociais também temos rádios e tv.
8. **Quais são as etapas de atendimento ao paciente realizadas dentro da sala de vacina?**
- **Resposta da Pessoa 1:** É realizada a triagem do paciente, orientações e aplicação.
 - **Resposta da Pessoa 2:** Triagem, orientações e aplicação.
9. **Contém alguma forma de espelho do cartão de vacina dos pacientes ? Se sim, como ele é confeccionado e armazenado ?**
- **Resposta da Pessoa 1:** Sim. Através da ficha de vacinação, na própria ficha.
 - **Resposta da Pessoa 2:** Sim, através da ficha de vacinação (Cartão espelho).
10. **Como são armazenadas todas as informações que são coletadas?**
- **Resposta da Pessoa 1:** No arquivo que fica na sala por ordem alfabética.
 - **Resposta da Pessoa 2:** Pasta arquivo por ordem alfabética e no computador.