

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO
DIRETORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

THIAGO FERREIRA DE AGUIAR

**SGHV- SISTEMA DE GERENCIAMENTO DE HOSPITAIS
VETERINÁRIOS**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2015

THIAGO FERREIRA DE AGUIAR

**SGHV- SISTEMA DE GERENCIAMENTO DE HOSPITAIS
VETERINÁRIOS**

Trabalho de Conclusão de Curso apresentada
como requisito parcial a obtenção do grau de
Tecnólogo em Análise e Desenvolvimento de
Sistemas de Informação pela Universidade
Tecnológica Federal do Paraná – Campus
Cornélio Procópio

Orientador: Prof. José Antonio Gonçalves

CORNÉLIO PROCÓPIO

2015



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Cornélio Procópio
Diretoria de Graduação e Educação Profissional
Tecnologia em Análise E Desenvolvimento De Sistemas



FOLHA DE APROVAÇÃO

THIAGO FERREIRA DE AGUIAR

SGHV- Sistema de Gerenciamento de Hospitais Veterinários

Trabalho de Conclusão de Curso apresentada como requisito parcial a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas de Informação pela Universidade Tecnológica Federal do Paraná – Campus Cornélio Procópio
Orientador: Prof. José Antonio Gonçalves

Aprovado em 14 de abril de 2015.

Banca Examinadora

Prof. José Antonio Gonçalves - Universidade Tecnológica Federal do Paraná

Prof.^a Adriane Carla Anastácio da Silva - Universidade Tecnológica Federal do Paraná

Prof. Alessandro Silveira Duarte - Universidade Tecnológica Federal do Paraná

AGRADECIMENTOS

Agradeço a Deus, o único digno de louvor e adoração, pois foi Ele que me criou e me colocou na família tão abençoada que tenho.

Até aqui, nessa graduação, Ele guiou meus passos, me deu forças durante toda esta caminhada e, principalmente, neste trabalho de diplomação.

Deus colocou em minha vida meu orientador, Prof. José Antonio Gonçalves, para me orientar, incentivar e apoiar na confecção deste trabalho. O agradeço também, porque houve momentos em que eu pensei em desistir, mas Ele falava em meu ouvido: “-Filho Eu estou contigo, tente mais um pouco.”, e me dava à solução que tanto me angustiava.

RESUMO

DE AGUIAR, Thiago Ferreira. SGHV: Sistema de Gerenciamento de Hospitais Veterinários. Monografia (Graduação em Tecnologia em Análise e Desenvolvimento de Sistemas) – Universidade Tecnológica Federal do Paraná. Cornélio Procópio – PR.

Este trabalho apresenta o desenvolvimento de um sistema *Web* para o gerenciamento de informações de rotina de hospitais veterinários, controlando consultas médicas realizadas, cadastros de clientes, pacientes, funcionários e dos serviços realizados e, ainda, controle do estoque de materiais e medicamentos, procedimentos realizados e, por fim, controle financeiro. Visto o grande mercado voltado a animais no país e a falta de uso de *softwares* de gestão, de sistemas mais abrangentes e de alta disponibilidade, o *software* produzido busca a correta apresentação das informações, ergonomia e disponibilidade de acesso. Para a criação deste sistema foi utilizado o paradigma de Orientação a Objetos, durante o levantamento e análise de requisitos, projeto e modelagem, além da sua implementação. Foi utilizada a linguagem de programação Java, aplicada por meio do *Framework* JSF 2.2 em conjunto com a biblioteca de componentes PrimeFaces 4.0. Empregou-se também a API para persistência JPA e o sistema de gerenciamento de banco de dados PostgreSQL 9.1. Durante o desenvolvimento do projeto verificou-se a importância e dificuldade em se obter um bom planejamento para a construção de *softwares*.

Palavras chaves: gerenciamento veterinário *Web*. Sistema de gerenciamento de hospital veterinário *Web*.

ABSTRACT

DE AGUIAR, Thiago Ferreira. SGHV: Veterinary Hospitals Management System. Monograph (Undergraduate Technology Analysis and Systems Development) - Universidade Tecnológica Federal do Paraná. Cornélio Procópio – PR.

This paper presents the development of a Web system for routine information management in veterinary hospitals, controlling information such as medical consultations, customer records, patients, staff and the services performed and also inventory control of materials and medicines , procedures performed, and finally, financial control. Given the large market oriented animals in the country and the lack of use of management software, more comprehensive systems and high availability, produced software searches the correct presentation of information, ergonomics and availability of access.. For the creation of this system was used Orientation paradigm Object, during lifting in and requirements analysis, design and modeling, as well as its implementation. We used the Java programming language, applied through the JSF Framework 2.2 together with component library PrimeFaces 4.0. It also employed the JPA API for persistence and the database management system PostgreSQL 9.1. During the development of the project there was the importance and difficulty of obtaining good planning to build software.

Keywords: veterinary management Web. Veterinary hospital management system Web

LISTA DE FIGURAS

Figura 1 - <i>Model View Controller</i>	24
Figura 2 - Metodologia Scrum Adaptada.....	26
Figura 3- Casos de Uso Geral.....	30
Figura 4 - Modelo Entidade-Relacionamento (final).....	31
Figura 5 - Diagrama de Classes do Pacote sghv. data.model.....	32
Figura 6 - Diagrama de Navegação	33
Figura 7 - Diagrama de Pacotes.....	35
Figura 8 - Caso de Uso – Manter Cliente.....	37
Figura 9 - Caso de Uso – Manter Paciente.....	37
Figura 10 - Caso de Uso- Manter Funcionário	40
Figura 11 - Caso de Uso – Manter Serviço	41
Figura 12 - Caso de Uso – Manter Consulta	41
Figura 13 - Caso de Uso- Manter Estoque	43
Figura 14 - Caso de Uso – Manter Procedimento.....	44
Figura 15 - Caso de Uso – Manter Usuário.....	46
Figura 16 - Caso de Uso – Manter Caixa.....	47

LISTA DE QUADROS

Quadro 1 - Cronograma Geral	28
Quadro 2 - Cronograma (<i>Sprint</i> 1).....	29
Quadro 3 - Requisitos funcionais	29
Quadro 4 - Cronograma (<i>Sprint</i> 2).....	35
Quadro 5 - Cronograma (<i>Sprint</i> 3).....	39
Quadro 6 - Cronograma (<i>Sprint</i> 4).....	43
Quadro 7 - Cronograma (<i>Sprint</i> 5).....	45
Quadro 8 - Cronograma (<i>Sprint</i> 6).....	46
Quadro 9 - Cronograma (<i>Sprint</i> 7).....	47

LISTA DE SIGLAS

AJAX	<i>Asynchronous Javascript and XML</i>
API	Interface de Programação de Aplicações
CASE	<i>Computer-Aided Software Engineering</i>
CEP	Código de Endereçamento Postal
CPF	Cadastro de Pessoa Física
CLM	Campus Luiz Meneghel (Universidade Estadual do Norte do Paraná)
CRMV	Conselho Regional de Medicina Veterinária
EJB QL	<i>Enterprise JavaBeans Query Language</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	Ambiente Integrado para Desenvolvimento
Java EE	<i>Java Enterprise Edition</i>
Java SE	<i>Java Standard Edition</i>
JPA	API para persistência Java
JPQL	Linguagem de consulta e persistência Java
MVC	Padrão Arquitetural <i>Model View Controller</i>
OMG	<i>Object Management Group</i>
ORM	Mapeamento objeto-relacional
POJOS	<i>Plain Old Java Objects</i>
RG	Registro Geral
SERPRO	Serviço Federal de Processamento de Dados
SGBDOR	Sistema de Gerenciamento de Banco de Dados Objeto Relacional
SGBD	Sistema de Gerenciamento de Banco de Dados
SPC	Serviço de Proteção a Crédito
UML	Linguagem de Modelagem Unificada
UENP	Universidade Estadual do Norte do Paraná
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVOS	13
1.1.1 OBJETIVOS ESPECÍFICOS	13
1.2 JUSTIFICATIVA.....	14
1.3 ORGANIZAÇÃO DO TRABALHO.....	16
2 REVISÃO BIBLIOGRÁFICA	17
2.1 TECNOLOGIAS	17
2.1.1 Java Enterprise Edition 6	17
2.1.2 Java Persistence API	17
2.1.3 Java Server Faces 2.2	18
2.1.4 PrimeFaces 4.0.....	19
2.1.5 PostgreSQL 9.1	19
2.2 FERRAMENTAS.....	20
2.2.1 Visual Paradigm for UML 7.2	20
2.2.2 NetBeans IDE 7.4.....	21
2.2.3 pgAdmin 3	21
2.2.4 XMind.....	21
2.2.5 Microsoft Project 2010	21
2.2.6 Microsoft Word 2010	22
3 PROCESSO E DESENVOLVIMENTO DE SOFTWARE	23
3.1 UML	23
3.2 MVC	24
3.3 SCRUM	25
3.3.1 ARTEFATOS	27
3.4 DESENVOLVIMENTO.....	28
3.4.1 SPRINT 1.....	28
3.4.1.1 Cronograma do o Sprint 1.....	29
3.4.1.2 Requisitos Funcionais.....	29
3.4.1.3 Casos de Uso Geral	30
3.4.1.4 Modelo Entidade-Relacionamento	31
3.4.1.5 Diagrama de Classes.....	32
3.4.1.6 Diagrama de Navegação	33
3.4.1.7 Dificuldades Encontradas Durante o Sprint 1	34
3.4.2 SPRINT 2.....	34
3.4.2.1 Cronograma do Sprint 2	35
3.4.2.2 Digrama de Pacotes	35
3.4.2.3 Caso de Uso - Sprint 2.....	36
3.4.2.3.1 Manter Cliente.....	37
3.4.2.3.2 Manter Paciente.....	37
3.4.2.4 Dificuldades Encontradas Durante o Sprint 2.....	38
3.4.3 SPRINT 3.....	38
3.4.3.1 Cronograma do Sprint 3	39
3.4.3.2 Casos de Uso – Sprint 3.....	40
3.4.3.2.1 Manter Funcionário	40
3.4.3.2.2 Manter Serviços.....	41
3.4.3.2.3 Manter Consulta	41

3.4.3.3 Dificuldades encontradas durante o Sprint 3	42
3.4.4 SPRINT 4.....	42
3.4.4.1 Cronograma do Sprint 4	43
3.4.4.2 Casos de Uso – Sprint 4.....	43
3.4.4.2.1 Manter Estoque	43
3.4.4.2.2 Manter Procedimento	44
3.4.4.3 Dificuldades Encontradas Durante o Sprint 4.....	44
3.4.5 SPRINT 5.....	44
3.4.5.1 Cronograma do Sprint 5	45
3.4.5.2 Dificuldades Encontradas Durante o Sprint 5.....	45
3.4.6 SPRINT 6.....	45
3.4.6.1 Cronograma do Sprint 6	46
3.4.6.2 Caso de Usos.....	46
3.4.6.2.1 Manter Usuário.....	46
3.4.6.2.2 Manter Caixa.....	47
3.4.6.3 Dificuldades encontradas durante o Sprint 6	47
3.4.7 SPRINT 7.....	47
3.4.7.1 Cronograma do Sprint 7	47
3.4.7.2 Dificuldades Encontradas Durante o Sprint 7.....	48
4 CONCLUSÃO	49
4.1 TRABALHOS FUTUROS	49
REFERÊNCIAS	51
APÊNDICE A - ARTEFATOS GERADOS	53

1 INTRODUÇÃO

O Brasil é uma das maiores potências na criação de animais, quer sejam de produção¹ ou companhia². Segundo dados de 2011 do IBGE (Zoetis, 2014), este mercado vem se consolidando e alcançando números cada vez mais representativos, aumentando ainda mais a necessidade de produtos e serviços que o atendam.

Este fato tem-se refletido diretamente na quantidade de médicos veterinários formados, capacitados para atuar nas diversas atividades e nichos de mercado como em fazendas, cooperativas, agropecuárias, hospitais e clínicas veterinárias. Assim, considerando o crescimento deste setor, surge a necessidade de haver no mercado mais opções de abordagens informatizadas que auxiliem os investimentos e as tomadas de decisões gerenciais para o segmento (seja *pet* ou destinado a animais de produção).

Consoante aos fatos supracitados, esta monografia apresenta o desenvolvimento de um sistema de informação, cujo objetivo principal é oferecer o gerenciamento de informações pertinentes para um melhor controle administrativo em hospitais veterinários.

O *software* construído neste trabalho apresenta como diferencial em relação aos existentes no mercado para a área como o QVET (empresa Q-Soft³) pelo fato ser desenvolvido em plataforma *Web*, que possibilita o acesso fora do ambiente hospitalar e tornando o sistema multi-plataforma de sistema operacional e permitindo o acesso através de dispositivos moveis. Também apresenta como diferencial a proximidade com o usuário por considerar as necessidades e características peculiares do local ou usuário.

O conjunto de requisitos utilizados como subsídios básicos para construção deste sistema foi obtido junto à parceria com Hospital Veterinário da Universidade Estadual do Norte do Paraná – Campus Luiz Meneghel (UENP-CLM/Bandeirantes-PR), considerando o interesse dos responsáveis pelo setor, na melhoria da gestão das informações administrativas do ambiente como consultas médicas, fluxo de caixa, pacientes e outros.

¹ Animais de produção: animais criados para o abate e/ou para a obtenção de subprodutos.

² Animais de companhia: animais com o intuito de servir de guarda e/ou prestar companhia ao dono.

³ Informações em: <<http://www.q-soft.net/>> Acesso em: 28/04/2015.

1.1 OBJETIVOS

O objetivo geral deste trabalho é expor e desenvolver um sistema informatizado em ambiente *Web*, visando um gerenciamento mais ágil e preciso das informações de rotina, com base nos subsídios básicos do Hospital Veterinário da UENP-CLM.

1.1.1 OBJETIVOS ESPECÍFICOS

O *software* busca aumentar o grau de controle, agilidade e disponibilização das informações na rotina de um hospital veterinário. O sistema tem por objetivos específicos armazenar e gerenciar informações, como de:

- Clientes (proprietários de animais): dados pessoais como identificação, números de telefones, endereços e seus animais;
- Pacientes (animais): espécie, raça, peso, seu prontuário médico e etc;
- Funcionários: identificação, funcionais, números de telefones e endereços e sua participação nas atividades ambulatoriais e cirúrgicas;
- Agendamentos de consultas com datas, horários, e possíveis envolvidos nas futuras consultas, como paciente, médico(s) veterinário(s) e outros;
- Serviços prestados pelo hospital e, quando necessário, seus valores;
- Procedimentos realizados pelo hospital.

Além destes, o programa também tem outros objetivos específicos:

- Obter a máxima disponibilidade de acesso ao sistema, pois pelo fato de ser construído para a plataforma *Web*, proporciona uma ampla utilização pelos usuários, possibilitando inclusive acessos externos ao hospital pelos médicos veterinários, além de tornar o sistema independente de sistema operacional e permitir o acesso por dispositivos moveis como *smartphones* e *tablets*;
- Manter o controle de acesso (*login*) sob a forma de autenticação de usuário e senha, apresentando apenas as funções pré-determinadas para cada perfil de usuário, atribuído previamente aos funcionários, que define seu acesso à aplicação;
- Administrar o estoque de materiais médicos e medicamentos;
- Gerenciar o fluxo financeiro, controlando informações e mantendo o registro de entradas e saídas.

1.2 JUSTIFICATIVA

Atualmente o Brasil é uma das maiores potências em criação de animais, dentre estes se destaca os animais voltados para companhia, ou ainda as culturas de bovinos, suínos, aves, entre outros. Estes são utilizados tanto para abate quanto para obtenção de seus subprodutos como leite, couro, lã, ovos e etc. Segundo dados do ano de 2011, do Instituto Brasileiro de Geografia e Estatística (IBGE), o país possuía o maior rebanho bovino do mundo com 212,8 milhões de cabeças, era o quinto maior produtor de leite e já liderava as exportações de carne, com aproximadamente 5,37 bilhões de dólares/ano. Em termos de carne suína, possuía o status de quarto maior produtor mundial, com 34,9 milhões de animais abatidos, representando 10% das exportações mundiais. Já em relação às aves a nação ocupava a terceira posição em exportações, a produção ultrapassava os 6 bilhões de cabeças abatidas no ano (2011 apud IBGE, Zoetis, 2014).

Segundo Zoetis (2014 apud Sindicato Nacional da Indústria de Produtos para a Saúde Animal, 2014) a população brasileira de animais de companhia é a segunda maior do mundo, com mais de 53 milhões de indivíduos, movimentando em 2011, cerca de 11 bilhões de reais. Já o mercado de saúde animal, com medicamentos, equipamentos, produtos entre outros, cresce em média 7% ao ano, chegou a movimentar cerca de 3,4 bilhões naquele mesmo ano.

O Brasil possui o maior número de faculdades de medicina veterinária do planeta, são 192 (destas 23 estão no estado do Paraná), colocando no mercado de trabalho mais de 6.000 profissionais anualmente. O país tem o maior número de médicos veterinários do mundo, são aproximadamente 130.000 médicos atuantes, muito destes atuam profissionalmente na área clínica, em instituições de ensino (em hospitais universitários), em clínicas ou, ainda, em hospitais veterinários públicos ou privados (Conselho Federal de Medicina Veterinária, 2014).

Devido ao crescimento exponencial do mercado de criação animal o sistema de saúde veterinário vem sendo sobrecarregado, as clínicas e hospitais veterinários já não acompanham o aumento de demanda. Nestes locais ainda há muito serviço administrativo sendo feito de forma manual, como preenchimento e triagem de documento. Este tipo de controle pode trazer, além de morosidade, outros problemas como a inconsistência de informações.

Os *softwares* de apoio à gestão, existentes para este negócio, ainda são pouco abrangentes, em sua grande maioria não são sistemas *Web*, assim tendo uma disponibilidade

de acesso limitada, além de muitos destes possuírem uma ergonomia deficiente, dificultando o uso diário. Dos sistemas atuais no mercado cita-se três:

- O QVET⁴, *software* da empresa de origem europeia Q-Soft, trata-se de um dos mais famosos *softwares* da área médica veterinária, que possui um alto controle de gestão do estabelecimento porem não possui plataforma Web;
- O sistema OranGest VET⁵, da Magnisoftware, também em plataforma *desktop*, possui uma interface complexa com diversas informações e funções na mesma tela;
- O Doctor Vet⁶ da empresa Xionce, além de não utilizar a plataforma *Web*, apresenta uma interface deficiente com botões sem especificações, não há cadastros de funcionários e médicos veterinários, nem controle de estoque, e ainda a possibilidade de gerar inconsistências de informações como a possibilidade de haver no sistema consultas sem cliente ou médico veterinário associado a esta. Assim se torna um aplicativo muito básico no controle de informações para a área.

O *software* construído e apresentado neste trabalho nasce buscando como diferencial, em relação aos existentes no mercado para a área, o fato ser desenvolvido em plataforma *Web* e, também, a proximidade com o usuário por considerar as necessidades e características peculiares do local em que se encontra. Além disso, procura ter um olhar especial para a correta apresentação das informações e ergonomia do sistema.

O sistema especificado por este trabalho busca mitigar os problemas de inconsistência de informações, morosidade e gerência no atendimento, informatizando e controlando as informações na rotina do Hospital Veterinário da Universidade Estadual do Norte do Paraná - Campus Luiz Meneghel na cidade de Bandeirantes, e procura oferecer melhor ergonomia, além de utilizar a plataforma *Web*, permitindo a utilização deste em diversos dispositivos e sistemas operacionais.

⁴Informações em: <<http://qvvet.com.br/>> Acesso em: 22/12/2014.

⁵Mais informações em: <<http://www.magnisoftware.pt/products/orangest-vet/>> Acesso em: 22/12/2014.

⁶Demais informações em: <<http://www.xionce.com/doctorvet/pt/>> Acesso em: 22/12/2014.

1.3 ORGANIZAÇÃO DO TRABALHO

O trabalho está dividido em forma de capítulos abordando revisões bibliográficas, processos e desenvolvimento de *software* e conclusão.

No Capítulo 2, com o título de Revisão Bibliográfica, são apresentados as plataformas utilizadas, além das tecnologias de desenvolvimento aplicadas, e também as ferramentas empregadas durante todo o projeto. Já no Capítulo 3 (Processo e Desenvolvimento de Software) é apresentado o modelo de processo e arquitetural, além da linguagem de documentação de artefatos, e, ainda, artefatos gerados durante a construção do *software*. Também apresenta o desenvolvimento das atividades, cronograma e dificuldades encontradas, a cada interação do projeto. Por último, no Capítulo 4 denominado Conclusão é exposto a conclusão do trabalho e os trabalhos futuros para dar continuidade no desenvolvimento do sistema.

2 REVISÃO BIBLIOGRÁFICA

O desenvolvimento do projeto se utilizou, em sua grande maioria de plataformas de sistemas operacionais livres, distribuição Linux Ubuntu, ressalva-se que, o Cronograma do projeto foi criado e acompanhado por meio da ferramenta proprietária Microsoft Project 2010, e a confecção deste documento utilizou o editor de texto Microsoft Word 2010. Estas duas ferramentas foram selecionadas, mesmo havendo similares grátis, tendo em vista, a experiência de uso e acesso a estas, e em especial a segunda, visando à possibilidade da perda da formatação do texto e seus componentes, também de poder-se contar com ajuda de seu corretor ortográfico, a popularidade de arquivos gerados e do controle de revisão, todos recursos mais consistentes nesta ferramenta.

O ambiente Linux foi adotado para a parte de modelagem e codificação, devido ao maior desempenho e robustez das ferramentas utilizadas, para o desenvolvimento de aplicações *Web*, comparando-se, anteriormente, em relação ao ambiente Microsoft Windows.

2.1 TECNOLOGIAS

Na construção de uma aplicação são utilizadas diversas tecnologias, a fim de se obter um projeto de acordo com os requisitos apontados na Análise de Requisitos e, também, a utilização de técnicas que promovam a maior manutenibilidade ao sistema. Considerando isto, foram utilizadas as seguintes tecnologias para desenvolvimento da aplicação:

2.1.1 *Java Enterprise Edition 6*

O Java EE é uma plataforma que contém um conjunto de tecnologias para diminuição do custo e complexidade do desenvolvimento de aplicações em várias camadas. É baseado na plataforma *Java Standard Edition*, oferece um conjunto de APIs⁷ para desenvolvimento e execução de aplicações, robustas, escaláveis e confiáveis, além de serem seguras e alocadas no lado do Servidor (Netbeans, 2014).

2.1.2 *Java Persistence API*

Java Persistence API (JPA) é uma especificação padrão da linguagem Java, é um dos principais componentes do Java EE, para mapeamento objeto-relacional (ORM), baseada em

⁷ Interfaces de Programação de Aplicações.

objetos que possuem design simples, que não dependem da herança de interfaces ou classes de *frameworks* externos, denominados POJOS (*Plain Old Java Objects*) para persistir em banco de dados os objetos Java (DEVMEDIA, 2014).

O JPA em sua versão atual 2.0, possui as principais características:

- Simples configuração: configuração de persistência através de anotações no código ou arquivos XML, ou ainda uma combinação destes;
- POJOS persistentes;
- Consultas em JPQL: as consultas em objetos podem ser realizadas por meio da *Java Persistence Query Language* (JPQL), uma linguagem de consulta que é derivada do *Enterprise JavaBeans Query Language* (EJB QL), a qual, na execução do sistema, é transformada para SQL para a comunicação com o SGBD;
- Integração e Testes: nas aplicações atuais, normalmente, há uma grande dificuldade em se aplicar testes em ferramentas que são executadas em um Servidor. Com isso em vista isto o JPA permite ser executado fora do Servidor de aplicação, assim os testes unitários podem ser executados mais facilmente e, também, à redução na possibilidade de perda e inconsistência de dados, visto que, os testes não concorrem com os acessos dos usuários a ferramenta no Servidor.

Segundo CAELUM (2014) o *Java Persistence API* possui várias implementações das quais se pode citar o *Hibernate*, que é a implementação mais utilizada, e o *EclipseLink*, que é a implementação referencial, ou seja, a implementação padrão, e ,também, foi a escolhida para aplicar ao projeto.

2.1.3 Java Server Faces 2.2

Java Server Faces (JSF) é um *framework* que permite criar aplicações Java para *Web* utilizando componentes visuais pré-prontos, dinamizando o desenvolvimento visual da aplicação, visto que não há a preocupação do desenvolvedor em criar componentes de tela utilizando HTML e *JavaScript* (CAELUM, 2014).

O JSF possui a vantagem de ser uma especificação da tecnologia Java EE, isto é, todo servidor de aplicações Java deve conter uma implementação desta especificação (CAELUM, 2014).

2.1.4 PrimeFaces 4.0

PrimeFaces é um repositório de componentes JSF para aplicações *Web*, baseada em AJAX e JSF 2.0, criada pela empresa turca PrimeTek. Possui documentação disponibilizada pelo próprio desenvolvedor, por meio de uma vitrine de componentes e um manual de usuário, disponível para *download*⁸ (PrimeFaces, 2014).

2.1.5 PostgreSQL 9.1

O PostgreSQL é um SGBDOR (Sistema Gerenciador de Bancos de Dados Objeto Relacional), é um produto *Open Source* disponível para *download*⁹ Possuindo versões para a maioria dos sistemas operacionais, possui recursos como:

- Consultas complexas: consultas no banco de dados que envolvem duas ou mais tabelas, consulta alinhadas, operações de conjuntos, junções e etc.;
- Chaves estrangeiras: é um campo, que aponta para a chave primária de outra tabela, este recurso garante a integridade referencias das associações entre as tabelas;
- Integridade transacional: é a qualidade do SGBD em manter o banco de dados íntegro após as transações. Transações são unidades atômicas de trabalho, que podem ser totalmente aplicadas ou revertidas;
- Controle de concorrência multi-versão: é o controle de acessos simultâneos ao banco de dados. O SGBD não permite que dois ou mais dados sejam alterados ao mesmo tempo evitando a inconsistência de dados;
- Suporte ao modelo híbrido objeto-relacional: embora este recurso não tenha sido utilizado, trata-se da possibilidade de se ter banco de dados orientados a objeto, onde objetos, classes e herança são suportados diretamente nos Esquemas do banco de dados e na linguagem de consulta e a possibilidade de se ter tipos de dados e métodos personalizados;
- Gatilhos: é um tipo especial de procedimento armazenado, que é executado sempre que há uma tentativa de modificar, apagar ou inserir dados em uma tabela que é assistida por ele;

⁸Download disponível em: <<http://primefaces.org/downloads>> Acesso em: 22/12/2014.

⁹Download disponível em: <<http://www.postgresql.org/download/>> Acesso em: 22/12/2014.

- Visão: é uma consulta armazenada que gera uma tabela virtual dinamicamente dos dados em no banco de dados quando o acesso àquela visão é solicitada;
- Interpretação de linguagem procedural em várias linguagens como PL/pgSQL, PL/Python, PL/Java, PL/Perl, para construção e armazenamento de procedimentos construídos pelo usuário;
- Indexação por texto: é a possibilidade de otimizar pesquisas utilizando como índices com dados de texto.

O desenvolvimento do PostgreSQL iniciou-se em 1995 derivando de outro projeto chamado Ingres, hoje já alcançou uma maturidade no mercado. No Brasil atualmente ele está presente na esfera pública como Caixa Econômica Federal, Ministério da Saúde (Datusus – Departamento de Informática do Sistema Único de Saúde), Serviço Federal de Processamento de Dados(Serpro), Banco do Brasil, Companhia de Tecnologia da Informação e Comunicação do Paraná (Celepar), Metrô de São Paulo, Sistema de Vigilância da Amazônia (SIVAM) e etc. (4LINUX, 2014).

2.2 FERRAMENTAS

No desenvolvimento de *software* as ferramentas são de suma importância, elas são o meio para se empregar as tecnologias no projeto. A escolha dos meios certos podem trazer ao projeto melhoramentos em planejamento, agilidade e, até mesmo, compreensibilidade no desenvolvimento da aplicação, como se pode verificar nas descrições das ferramentas a seguir.

2.2.1 Visual Paradigm for UML 7.2

Visual Paradigm for UML é uma ferramenta CASE com várias opções de diagramas com notações da UML 2, também oferece diagramas de requisitos SysML e permite trabalhar o projeto físico por meio ne notações que dão suporte a construção do modelo Entidade-Relacionamento. O *software* oferece suporte a visualização e manipulação do projeto de modelagem, a transformações específicas para códigos-fonte de algumas linguagens de programação e também a técnica de engenharia reversa (Universidade Federal de Viçosa, 2014).

2.2.2 NetBeans IDE 7.4

O NetBeans é um ambiente integrado de desenvolvimento que permite ao usuário construir, compilar, depurar e instalar programas. Possui funcionalidades extensíveis por meio de módulos, isto é, possui um repositório de *plug-ins* e extensões para melhor adequar as necessidades do desenvolvedor (NetBeans,2014). É um produto *Open Source*, disponível gratuitamente para *download*.¹⁰ Com essa IDE é possível explorar todas as particularidades da linguagem Java, por meio de uma interface gráfica limpa e eficiente.

2.2.3 pgAdmin 3

O pgAdmin é uma interface gráfica de administração do SGBD PostgreSQL, com ela é possível explorar todas as funcionalidades deste SGBD. Tem alta disponibilidade de acesso, já que possui versões adequadas à maioria dos sistemas operacionais do mercado, ainda possui código livre, e é desenvolvido por uma comunidade de especialistas em PostgreSQL (pgAdmin, 2014).

2.2.4 XMind

O XMind é um programa para auxiliar na organização de ideias, possibilita criar fluxogramas e mapas mentais para estruturar os trabalhos e futuros empreendimentos (XMind, 2014).

2.2.5 Microsoft Project 2010

O Project é um aplicativo proprietário, desenvolvido pela Microsoft, para gestão de projetos. Com seu uso é possível verificar e reestruturar rapidamente os cronogramas e permite ver como as mudanças afetaram o projeto. Este programa oferece várias ferramentas que ajudam durante e ao longo de todas as fases de planejamento do projeto, permitindo obter resultados efetivos, entre os recursos tem-se o gráfico de Gantt, Rede e Recurso, e, ainda, as Planilhas de Recursos e a Utilização de Recursos e Tarefas (HSoluções em Informática, 2014).

¹⁰Download disponível em: <<https://netbeans.org/downloads/7.4/index.html> Acesso em: 22/12/ 2014.

2.2.6 Microsoft Word 2010

Conhecido editor de textos do pacote *Office* também da empresa Microsoft. Ele dispõe características como:

- Diversos tipos e tamanhos de fontes e símbolos gráficos;
- Auto formatação de textos e documentos;
- Revisor ortográfico incorporado;
- Visualização da imagem real de impressão do documento;
- Controle de revisões recurso que permite controlar as alterações feitas no documento por um ou mais usuário.

3 PROCESSO E DESENVOLVIMENTO DE SOFTWARE

O processo de *software* pode ser definido como uma coleção de atividades, ações, tarefas, produto de trabalho e padrões comportamentos relacionados que são necessários ao desenvolvimento de *software*. Um Padrão de Processo fornece modelos, métodos consistentes, para descrever características importantes do processo de *software*. Um processo de desenvolvimento pode ser definido pela combinação de padrões que satisfaçam as necessidades do projeto (PRESSMAN, 2006).

3.1 UML

A Linguagem de Modelagem Unificada (UML) é uma linguagem padrão para o desenvolvimento de projetos de *softwares*, munida notações gráficas para descrever a estrutura do projeto. (BOOCH; RUMBAUCH; JACOBSON, 2006). Esta é um padrão controlado por um consórcio de empresas chamado OMG (*Object Management Group*), destinada a:

- Visualizar;
- Especificar;
- Construir;
- Documentar os artefatos de um sistema.

Neste projeto foram utilizadas as principais notações da UML, para especificar funcionalidades, estrutura e comportamento do sistema. Os diagramas eleitos para isto foram:

- Diagrama de Casos de Uso: ele representa em alto nível as funcionalidades do sistema e os determinados atores externos;
- Diagrama de Classes: representa uma coleção de classes e seus relacionamentos;
- Diagrama de Pacotes: apresenta a organização dos pacotes da aplicação;
- Diagrama de Sequência: representa uma perspectiva cronológica da troca de mensagens entre os objetos.

3.2 MVC

O *Model View Controller* (MVC) é um modelo arquitetural de *software* que consiste na construção da aplicação em camadas (Figura 1):

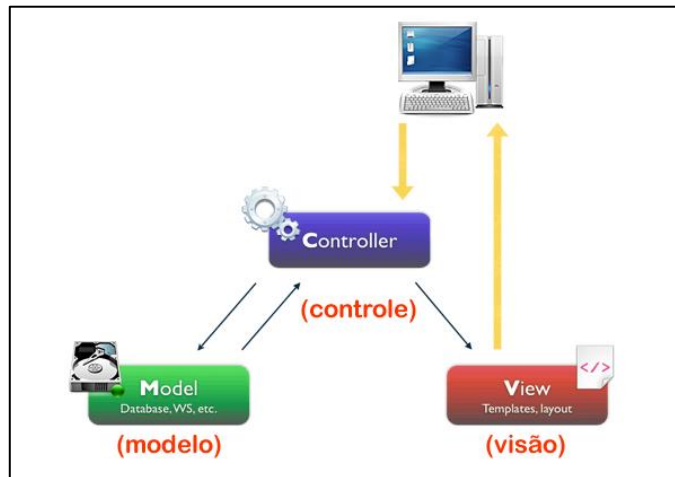


Figura 1 - Model View Controller

Fonte: <http://www.webartz.com.br/especiais/tecnologia-da-informacao/codeigniter-e-mvc-parte-01-mvc-conceito-e-funcionamento/> Acesso em: 02/02/2015

A primeira camada chamada *View* (Visão) ou camada de Apresentação é responsável pela interação com o usuário, isso pode ser desde linhas de comandos ou, até mesmo, uma complexa interface gráfica. Sobre o projeto desenvolvido, nesta camada enquadra-se as páginas xhtml, os *Managed Beans*, conversores, validadores e *Back Beans*, este conjunto de arquivos é responsável pela pura apresentação e coleta dos dados, validação e conversão dos dados em tempo de execução, além de controle e tratamento das ações das páginas e também encaminhamento requisições a camada de controle.

A camada *Controller* (Controle), ou Domínio, é responsável pelas regras de negócio do sistema, ou seja, possui o trabalho que a aplicação deve executar. Esta comunica a camada de Apresentação com a camada de Fonte de Dados. Nesta camada, no projeto, estão presentes as classes controller, onde estão as regras de negócio em sua forma mais pura, promovendo uma maior compreensibilidade e manutenção destas.

A *Model* (Modelo), ou Fonte de Dados, é responsável pela comunicação com outros sistemas que executam serviços para a determinada aplicação, geralmente ela se comunica com um SGBD. Nessa porção encontra-se as classes de entidade de banco de dados do projeto, as quais são responsáveis em transformar dados relacionais em objetos e vice versa.

Martin Fowler (FOWLER, 2006) apresenta a tabela de responsabilidades de cada camada da aplicação, vide Tabela 1.

Tabela 1 – Model View Controller

Camada	Responsabilidade
Apresentação (<i>View</i>)	Fornecimento de serviços, exibição de informações (p.ex. em Windows ou HTML, tratamento de solicitações do usuário -cliques com o mouse, pressionamento de teclas-, requisições HTTP, chamadas em linhas de comandos, API em lote).
Domínio (<i>Controller</i>)	Lógica que define qual é o real propósito do sistema
Fonte de Dados (<i>Model</i>)	Comunicação com os SGBD's, sistemas de mensagens, gerenciadores de transições e outros serviços.

Fonte: Fowler, 2006 – adaptado

3.3 SCRUM

“O Scrum enfatiza o uso de um conjunto de “padrões de processo de *software*” que se mostraram efetivos para projetos com prazos apertados, requisitos mutantes e criticalidade de negócio” (PRESSMAN, 2006).

O Scrum é uma metodologia ágil para pequenas equipes, que tem por algumas características:

- A possibilidade de declarar o produto pronto a qualquer momento do processo;
- O trabalho dividido em partes pequenas e claras, para serem abordadas de forma ordenada para constituir um produto;
- O processo é incremental e adaptável tanto a modificações técnicas quanto de negócios;
- Os testes e documentação são realizados a medida da construção do produto;
- Designação de papéis para todos os envolvidos com o desenvolvimento do produto.

Para a melhor adequação e aproveitamento desse modelo, o mesmo foi adaptado mantendo a essência do Scrum, tal alteração manteve os envolvidos no projeto com papéis determinados:

- Product Owner: futuros usuários do sistema (funcionários do Hospital Veterinário da UENP-CLM);
- Team Scrum e Scrum Master: o desenvolvedor do sistema, assumi os papéis de chefe de equipe e também de equipe de desenvolvimento.

Este modelo de processo ficou dividido em:

- Pendencias (Backlog): lista priorizada de requisitos ou características do projeto (vide Quadro 3);
- *Sprints*: unidades de trabalho necessárias para satisfazer um requisito definido nas pendencias, em um determinado tempo, ou seja, incremento (vide Quadro 1), cada Sprint ficou responsável por um trecho do desenvolvimento do software (vide Quadro 2, Quadro 4 a 9);
- Revisões de Progresso: revisões feitas pelo desenvolvedor sobre o progresso do projeto, as quais acontecem semanalmente;
- *Mostra de Demos*: mostra de incremento de *software* funcional de modo que o usuário possa avaliar. Ao longo do projeto foram apresentadas aos futuros usuários do sistema, as funcionalidades implementadas durante o incremento e também colhidas sugestões e críticas deles.

A Figura 1 demonstra o processo de desenvolvimento escolhido para o projeto.

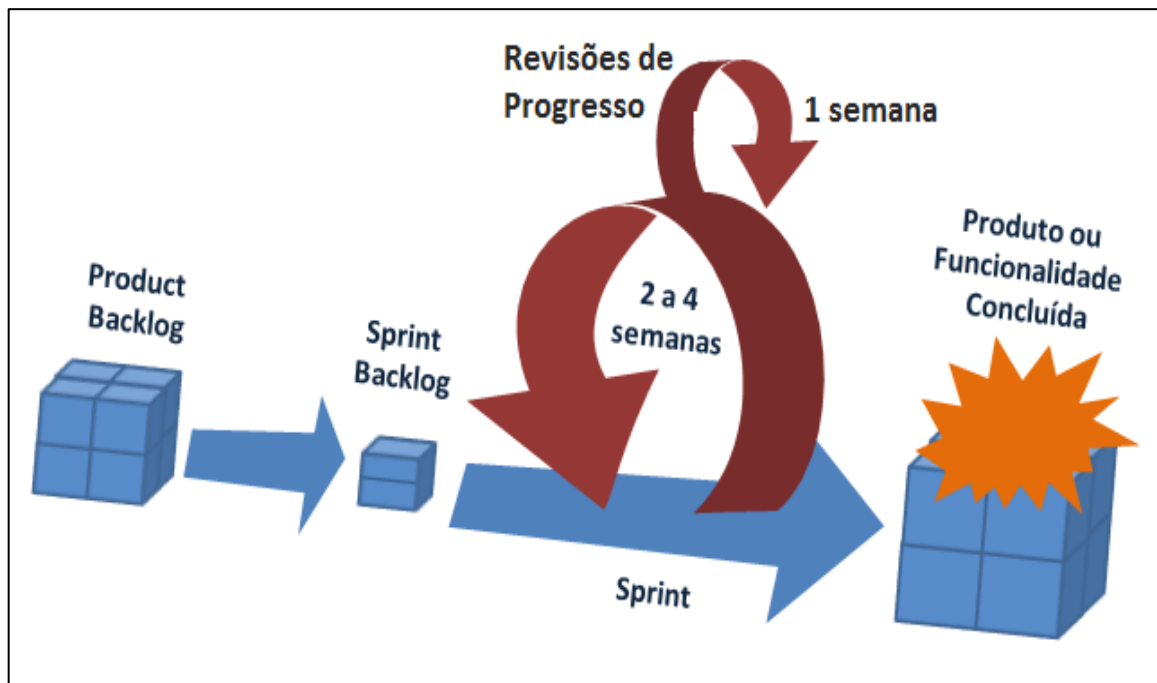


Figura 2- Metodologia Scrum Adaptada

Fonte: Adaptado de <<http://acordocoletivo.org/2014/12/25/o-que-e-scrum-afinal/>> Acesso em: 16/04/2015

3.3.1 ARTEFATOS

O modelo de processo Scrum determina que durante o desenvolvimento do projeto seja gerados artefatos para a melhor documentação do *software*. Este conjunto de artefatos nos permite a melhor percepção do problema e resolução deste, podem ser em forma de tabelas, quadros, diagramas diversos, entre outros. Como principais artefatos deste projeto conta-se com:

- Lista de Requisitos Funcionais: determina o que o sistema será capaz de realizar, ou seja, as possíveis funções do sistema. Através de entrevistas com os futuros usuários, e análise de documentos internos do Hospital Veterinário, pode-se delimitar e especificar os requisitos funcionais do sistema, identificando o grau de prioridade e risco de cada requisito (vide Quadro 3);
- Diagramas de Casos de Uso: representam ações realizadas no sistema pelos agentes externos (usuários), estes se apresentam de forma geral e partindo para a explosão destes casos de uso. Estes foram determinados com base na lista de requisitos funcionais;
- Modelo Entidade-Relacionamento (MER): descreve o modelo de dados de um sistema com alto nível de abstração, é a principal representação gráfica do Modelo Entidade-Relacionamento, tal representação foi modelada com base nos dados especificados e utilizados na rotina do HV/UENP-CLM (vide Figura 4);
- Diagramas de Classes: apresenta as estruturas das classes, com seus atributos e métodos, além de suas associações. Estes diagramas foram divididos pelos pacotes que cada classe se encontra;
- Diagrama de Pacotes: demonstra estruturalmente a forma que esta disposta às classes e código do sistema (vide Figura 7);
- Diagramas de Sequência: exhibe de forma temporal a interação e troca de mensagens nos relacionamentos com cada camada, classe e objetos, estes estão delimitados em digrama de fluxo normal e diagramas de fluxo alternativo.

Os diagramas de Pacotes, Casos de Uso, Classes, o Modelo Entidade-Relacionamento e também a Lista de Requisitos Funcionais, são apresentados e descritos ao decorrer do tópico que trata sobre o desenvolvimento do projeto. Os demais diagramas estão reunidos e dispostos como APÊNDICE A.

3.4 DESENVOLVIMENTO

Para uma melhor gerência do projeto é proposto um modelo de desenvolvimento o qual foi adotado o Scrum e um cronograma de execução estimado, com a descrição das atividades e datas de início e término de cada uma delas assim podendo acompanhar e analisar o andamento do projeto, possibilitando alterações e adequações se esta for necessário.

O projeto contou com um total de 165 dias de desenvolvimento, contando com atividades de planejamento, modelagem, codificação, além de execução de testes e revisões. Tais atividades contaram com o emprego de tecnologias e ferramentas apropriadas às quais já foram descritas.

O desenvolvimento do projeto ocorreu em conformidade com o cronograma apresentado no Quadro 1. O cronograma está dividido em sete *Sprint's* e uma mostra final do projeto ao cliente (HV/UENP-CLM), cada parte, ou incremento, possui atividades as quais são apresentadas em sub cronogramas dispostos ao longo deste tópico.

Nome da tarefa	Duração	Início	Término
+ 1ª Sprint	29 dias	Seg 28/04/14	Qui 05/06/14
+ 2ª Sprint	24 dias	Sex 06/06/14	Qua 09/07/14
+ 3ª Sprint	26 dias	Qui 10/07/14	Qui 14/08/14
+ 4ª Sprint	19 dias	Qui 14/08/14	Ter 09/09/14
+ 5ª Sprint	21 dias	Ter 09/09/14	Ter 07/10/14
+ 6ª Sprint	21 dias	Ter 07/10/14	Ter 04/11/14
+ 7ª Sprint	24 dias	Qua 05/11/14	Seg 08/12/14
Mostra Final ao Cliente	1 dia	Ter 09/12/14	Ter 09/12/14

Quadro 1 - Cronograma Geral

3.4.1 SPRINT 1

Com início em 28/04/2014 e término em 05/06/2014, o Sprint 1 foi responsável pela redefinição dos escopos e alguns artefatos do projeto, de acordo com os apresentados na Proposta de Trabalho de Conclusão de Curso. Por meio de algumas reuniões com os envolvidos no projeto, foram revistos os requisitos funcionais e dados a serem manipulados pelo sistema, assim alterando os Casos de Uso e promovendo a manutenção do Modelo Entidade-Relacionamento do projeto, e estipulando um novo cronograma geral.

3.4.1.1 Cronograma do Sprint 1

O *Sprint 1* do projeto obedeceu o cronograma apresentado no Quadro 2.

Nome da tarefa	Duração	Início	Término
1ª Sprint	29 dias	Seg 28/04/14	Qui 05/06/14
Determinação dos requisitos	8 dias	Seg 28/04/14	Qua 07/05/14
Diagrama de Casos de Uso Geral	2 dias	Qui 08/05/14	Sex 09/05/14
Diagrama Entidade Relacionamento	8 dias	Seg 12/05/14	Qua 21/05/14
Diagrama de Classes	8 dias	Qui 22/05/14	Seg 02/06/14
Diagrama de Navegação	3 dias	Ter 03/06/14	Qui 05/06/14

Quadro 2 - Cronograma (*Sprint 1*)

3.4.1.2 Requisitos Funcionais

Os requisitos funcionais definem o que um sistema será capaz de realizar. Por meio deles são especificadas funções e ações do sistema, uma função ou ação é descrita como um conjunto de entradas e saídas. Também foi atribuído um grau de risco e prioridade de desenvolvimento a cada função. No Quadro 3 podemos ver os requisitos levantados, juntamente com seus graus riscos e prioridade destes.

	Requisito	Risco	Prioridade
R1	Manter registro de Clientes	Alto	Alta
R2	Manter registro de Pacientes	Baixo	Alta
R3	Gerenciar Serviços	Baixo	Alta
R4	Gerenciar Estoque	Baixo	Alta
R5	Gerenciar Consultas	Alto	Alta
R6	Gerenciar Procedimentos	Alto	Alta
R7	Manter Caixa	Alto	Alta
R8	Gerenciar Usuários	Alto	Alta
R9	Manter registro de Funcionários	Alto	Alta

Quadro 3 - Requisitos funcionais

3.4.1.3 Casos de Uso Geral

De posse da Lista de Requisitos Funcionais é gerados o Diagrama de Casos de Uso, o qual a primeiro momento descreve de forma geral as funcionalidades do sistema e a sua interação com os usuários (vide Figura 3). Cada caso de uso é explorado em outros diagramas de Casos de Uso, que serão apresentados mais a frente.

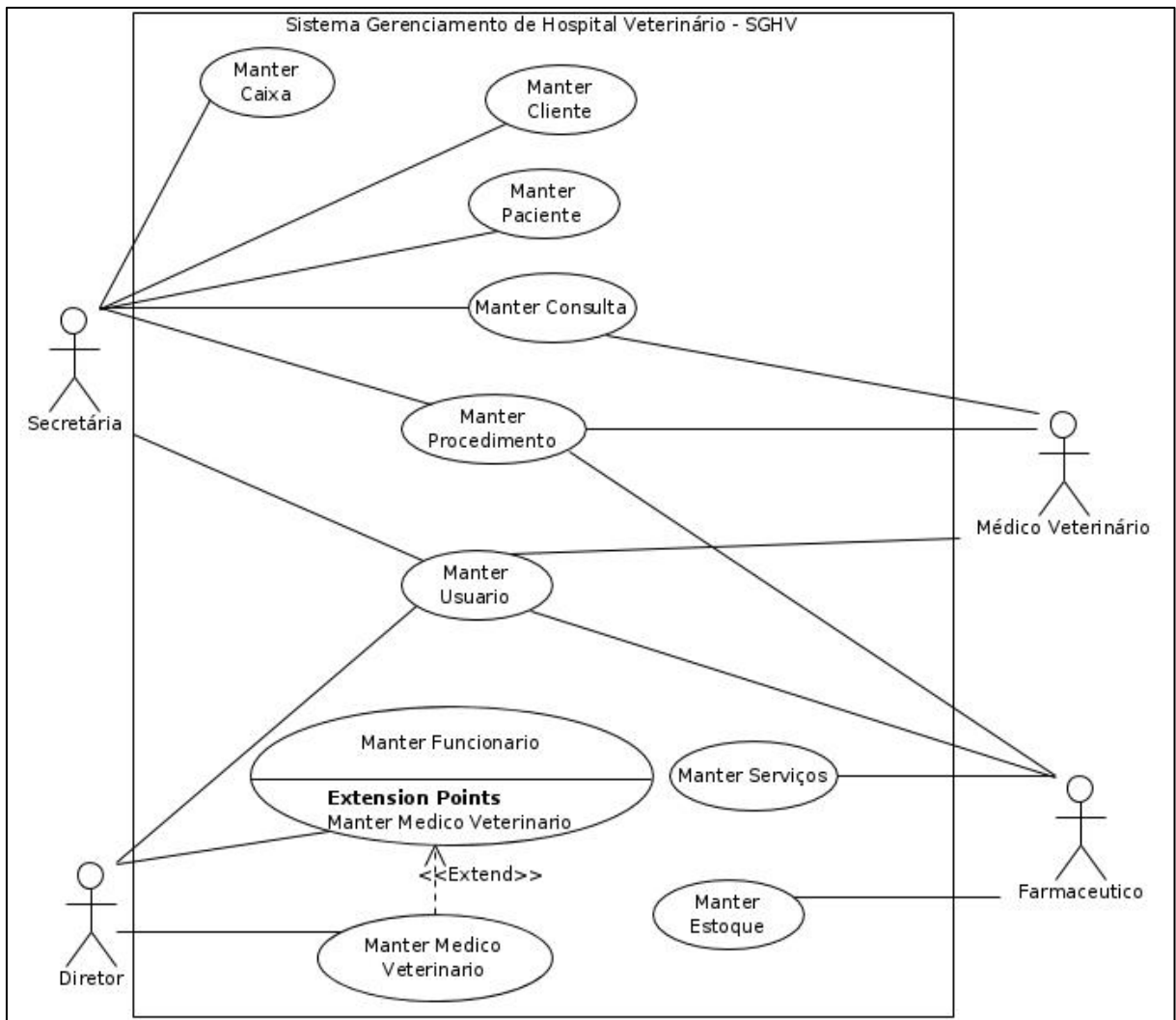


Figura 3- Casos de Uso Geral

3.4.1.4 Modelo Entidade-Relacionamento

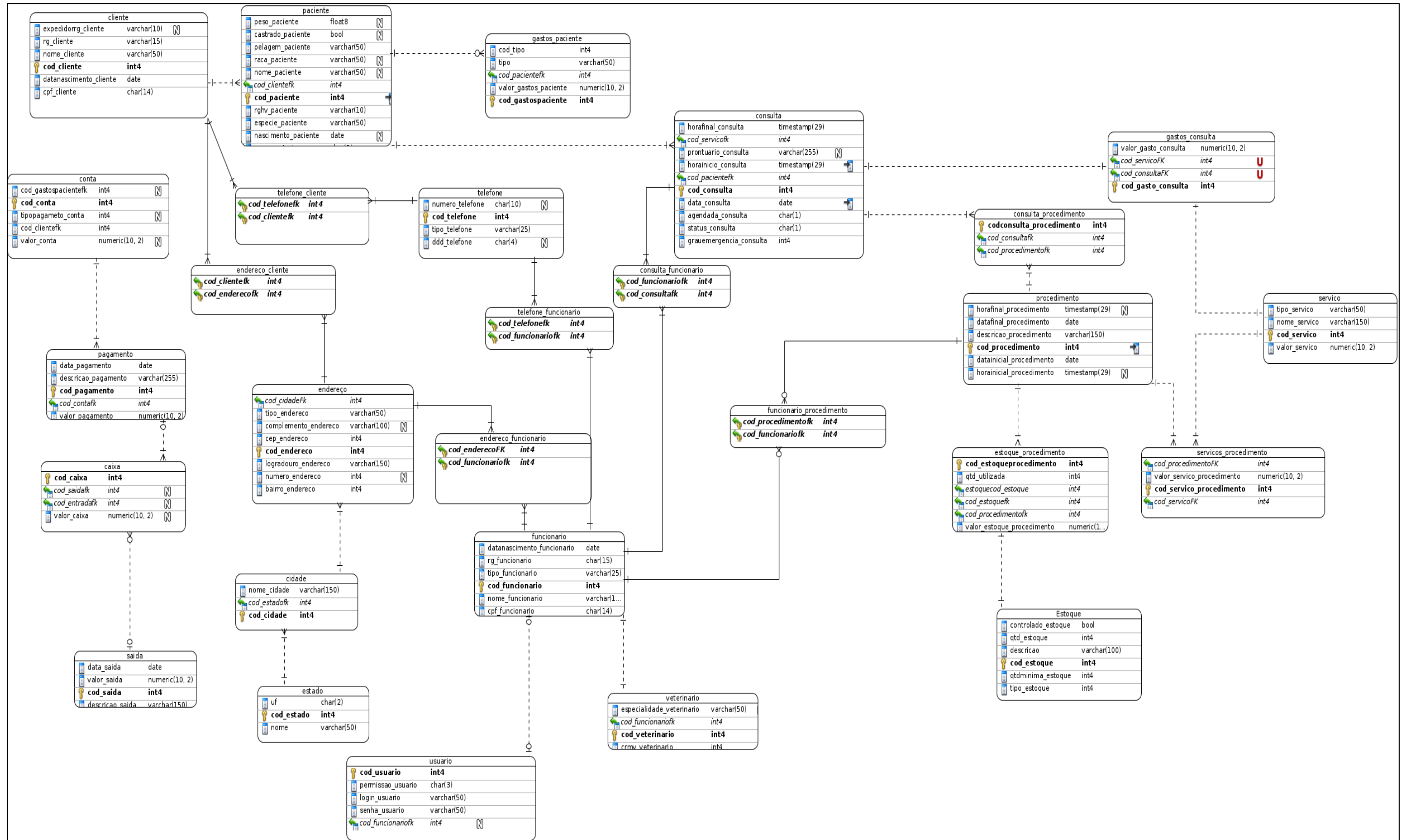


Figura 4 - Modelo Entidade-Relacionamento (final)

3.4.1.6 Diagrama de Navegação

O Diagrama de Navegação apresenta as páginas pertencentes ao sistema e os possíveis caminhos para ter acesso a uma determinada página. A Figura 6 apresenta a navegabilidade do sistema.

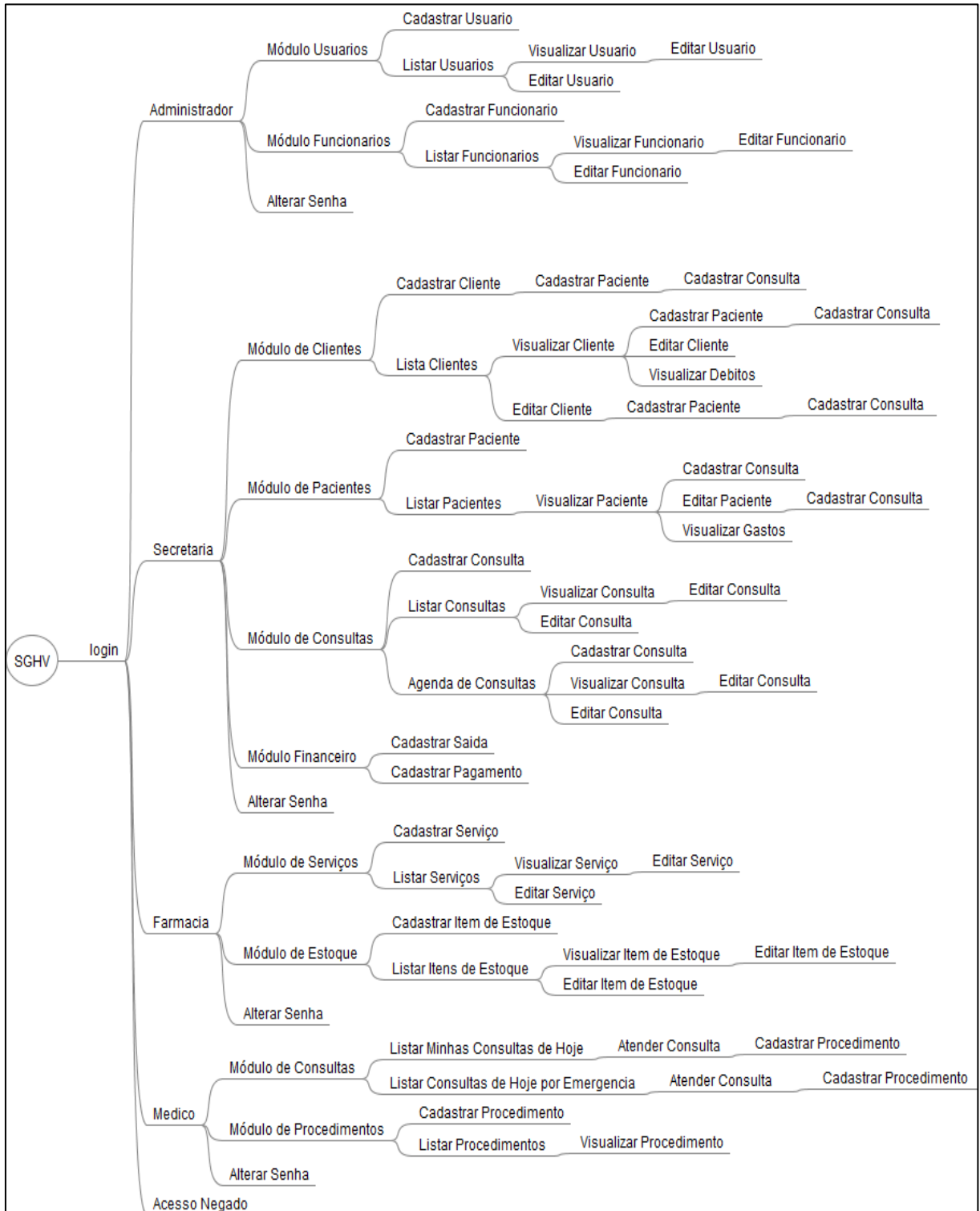


Figura 6 - Diagrama de Navegação

3.4.1.7 Dificuldades Encontradas Durante o Sprint 1

Como dificuldades durante este *Sprint* pode-se citar o levantamento de requisitos junto ao usuário, se dada por falta de experiência em técnicas de abordagem, para conseguir distinguir o que o usuário diz que necessita, do que ele precisa que o sistema controle e os dados necessários para isto.

3.4.2 SPRINT 2

Durante o Sprint 2 foi criado a base de dados inicial no SGBD com base no Modelo Entidade-Relacionamento, definido no período anterior, também foi criado a estrutura inicial do projeto na IDE compreendendo a estrutura de pacotes, a importação de bibliotecas, o padrão inicial de interface e, ainda, a conexão com a base de dados. Durante este incremento iniciou-se também a codificação através dos módulos de Paciente e Cliente.

No Módulo Cliente foram desenvolvidas as funcionalidades:

- Cadastrar Cliente: cadastro com dados pessoais de identificação como nome, CPF, RG e data de nascimento, além de cadastro de telefones com tipo, código de área e número, e também o cadastro de endereços com o tipo, logradouro, CEP, cidade, entre outros;
- Listar Clientes: listagem de todos os clientes em forma de tabela, com os dados principais destes e campos pesquisáveis com a opção de editá-los ou visualizá-los em outra tela.

No Módulo Paciente foram implementadas as funções:

- Cadastrar Paciente: cadastro com os dados de identificação do animal como nome, cor, peso, seu proprietário, entre outros;
- Listar Pacientes: listagem de todos os animais atendidos pelo hospital, em forma de tabela exibindo seus principais dados, também opção de editá-los ou visualizá-los em outra tela e, ainda, pesquisa em determinados campos.
- Visualizar Paciente: apresentação de todos os dados do paciente, seguido da opção de editá-lo;
- Editar Paciente: opção que permite alterar dados dos pacientes como peso, se é castrado ou não, encontra-se em óbito, entre outros.

3.4.2.1 Cronograma do Sprint 2

O *Sprint2* apresentado por meio do Quadro 4, teve o seguinte cronograma:

Nome da tarefa	Duração	Início	Término
2ª Sprint	24 dias	Sex 06/06/14	Qua 09/07/14
Criação da Base de Dados	2 dias	Seg 09/06/14	Ter 10/06/14
Criação do Projeto na IDE	2 dias	Qua 11/06/14	Qui 12/06/14
Diagramas de Casos de Uso	2 dias	Sex 13/06/14	Seg 16/06/14
Manter Cliente	1 dia	Sex 13/06/14	Sex 13/06/14
Manter Paciente	1 dia	Seg 16/06/14	Seg 16/06/14
Codificação	15 dias	Qua 18/06/14	Ter 08/07/14
Modulo de Cliente	9 dias	Qua 18/06/14	Seg 30/06/14
Cadastrar Cliente	5 dias	Ter 17/06/14	Seg 23/06/14
Listar Clientes	1 dia	Ter 24/06/14	Ter 24/06/14
Visualizar Cliente(tela)	1 dia	Qua 25/06/14	Qua 25/06/14
Editar Cliente(tela)	1 dia	Qui 26/06/14	Qui 26/06/14
Modulo de Paciente	5 dias	Ter 01/07/14	Seg 07/07/14
Cadastrar Paciente	2 dias	Ter 01/07/14	Qua 02/07/14
Listar Pacientes	1 dia	Qui 03/07/14	Qui 03/07/14
Visualizar Paciente	1 dia	Sex 04/07/14	Sex 04/07/14
Editar Paciente	1 dia	Seg 07/07/14	Seg 07/07/14

Quadro 4 - Cronograma (Sprint 2)

3.4.2.2 Digrama de Pacotes

Para a melhor organização dos arquivos de código, o projeto seguiu a seguinte estruturação de pacotes, promovendo um desenvolvimento mais ágil e aumentando a manutenibilidade deste (vide Figura 7).

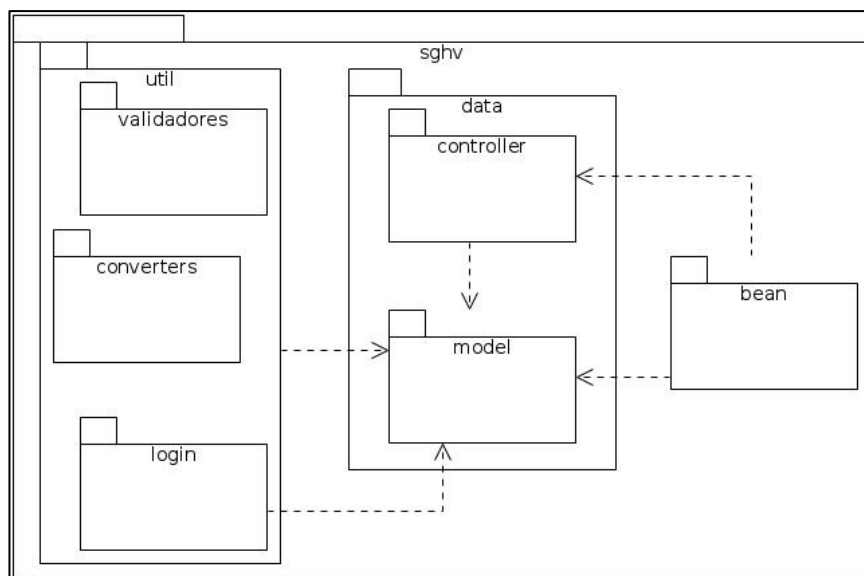


Figura 7 – Diagrama de Pacotes

Cada pacote do sistema reúne arquivos de código com funcionalidades em comum, assim descritos:

- *sghv*: possui os pacotes *bean*, *útil* e *data*(vide Figura 7);
- *bean*: se encontram os *managed beans* que são os controladores das telas do sistema. Os *managed beans* se comunicam com as classes controladoras do pacote *controller*;
- *data*: possui os pacotes *model* e *controller*;
- *model*: estão presentes as classes de entidade de banco de dados, utilizadas pelo JPA para a transformação de dados relacionais, do banco de dados, em objetos na aplicação;
- *controller*: neste pacote se encontram as classes *controller* do *software*, nestas então as regras de negocio de forma pura, para uma melhor manutenibilidade, e compreensão da execução destas por parte do desenvolvedor a estas regras, as classes *controller* se comunicam com as classes de entidade de banco de dados presentes no pacote *model*;
- *util*: são encontrados os pacotes de validadores, *converters* e *login*, além de uma gama de *Back Beans* utilizados para pequenas funções de tela em tempo de execução como alinhamento de combos;
- *validadores*: neste pacote estão os validadores de dados usados em tempo de execução, como validação de CPF e etc.;
- *converters*: nele estão os conversores do sistema , utilizados para transformar um determinado dado de um tipo em outro, para a apresentação nas telas do sistema;
- *login*: possui os filtros para o controle de acesso dos usuários ao sistema. O acesso do usuário se dá por meio de autenticação (*login*) de usuário e senha, cada usuário tem um perfil pré-determinado e este dá acesso a determinadas funções e telas, e este acesso é filtrado por estes filtro presentes neste pacote.

3.4.2.3 Caso de Uso - Sprint 2

Neste *Sprint* foram obedecidos os casos de usos Manter Paciente, e Cliente, apresentados a seguir:

3.4.2.4 Dificuldades Encontradas Durante o Sprint 2

Neste *Sprint* encontrou-se a dificuldade em distinguir quais as diferenças e aplicações de um *Managed Bean* para uma classe *controller*. Chegando ao resultado que um *Managed Bean* é um controlador de tela com um escopo definido e uma classe *controller* é onde são aplicadas as regras de negócio do sistema, é o local onde os dados da camada de modelo são manipulados por um acionamento do controlador de tela através de uma requisição.

Também se permitiu o refinamento do escopo para um *Managed Bean* que controlaria uma tela com inserção de dados em um ou mais relacionamentos do tipo N : N.

3.4.3 SPRINT 3

O *Sprint* 3 foi responsável pela criação dos módulos de Funcionário, Serviço e Consulta.

No Módulo Funcionário foram desenvolvidas as funcionalidades:

- Cadastrar Funcionário: cadastro com dados de identificação como nome, CPF, RG e data de nascimento, além de cadastro de telefones com tipo, código de área e número, além do cadastro de endereços com o tipo, logradouro e demais dados, e também o cargo que exerce, e se este for um médico veterinário o seu número no CRMV;
- Listar Funcionários: listagem de todos os clientes em tabela com os dados principais destes, e campos pesquisáveis e ainda a opção de visualizá-los ou editá-los a diante.

No Módulo de Serviços implementou-se:

- Cadastrar Serviço: cadastro de serviço com o tipo, a descrição e seu referido valor;
- Listar Serviços: listagem de todos os serviços em forma de tabela com todos os dados destes, campos com permissão de pesquisa e as opções de editar ou visualizar em outra tela;
- Visualizar Serviço: visualização dos dados do serviço, seguido da opção de editá-lo;
- Editar Serviço: função que possibilita alterar o valor do serviço.

No Módulo de Consultas foram construídas as funções:

- Agendar Consulta: cadastro de consulta com um campo de texto para prontuário, funcionários envolvidos, serviço e paciente;
- Listar Consultas: listagem dos principais dados das consultas em tabela e opção de pesquisa por campos, e ainda editá-las ou visualizá-las em outra tela;
- Visualizar Consulta: visualização dos dados da Consulta.

3.4.3.1 Cronograma do Sprint 3

O cronograma apresentado no Quadro 5 descreve como foi o procedimento do *Sprint 3*.

Nome da tarefa	Duração	Início	Término
3ª Sprint	26 dias	Qui 10/07/14	Qui 14/08/14
Diagramas de Casos de Uso	3 dias	Qui 10/07/14	Seg 14/07/14
Manter Funcionario	1 dia	Qui 10/07/14	Qui 10/07/14
Manter Serviço	1 dia	Sex 11/07/14	Sex 11/07/14
Manter Consulta	1 dia	Seg 14/07/14	Seg 14/07/14
Codificação	23 dias	Ter 15/07/14	Qui 14/08/14
Modulo Funcionario	10 dias	Ter 15/07/14	Seg 28/07/14
Cadastrar Funcionario	5 dias	Ter 15/07/14	Seg 21/07/14
Listar Funcionarios	1 dia	Ter 22/07/14	Ter 22/07/14
Visualizar Funcionario(tela)	2 dias	Qua 23/07/14	Qui 24/07/14
Editar Funcionario(tela)	2 dias	Sex 25/07/14	Seg 28/07/14
Modulo Serviço	5 dias	Ter 29/07/14	Seg 04/08/14
Cadastrar Serviço	1 dia	Ter 29/07/14	Ter 29/07/14
Listar Serviços	1 dia	Qua 30/07/14	Qua 30/07/14
Visualizar Serviço	1 dia	Qui 31/07/14	Qui 31/07/14
Editar Serviço	1 dia	Seg 04/08/14	Seg 04/08/14
Modulo Consulta	8 dias	Ter 05/08/14	Qui 14/08/14
Cadastrar Consulta	4 dias	Ter 05/08/14	Sex 08/08/14
Listar Consultas	1 dia	Seg 11/08/14	Seg 11/08/14
Visualizar Consulta(tela)	2 dias	Ter 12/08/14	Qua 13/08/14

Quadro 5 - Cronograma (Sprint 3)

3.4.3.2 Casos de Uso – Sprint 3

Neste período foram explorados os casos de uso de Manter Funcionário, Manter Serviço e Manter Consulta.

3.4.3.2.1 Manter Funcionário

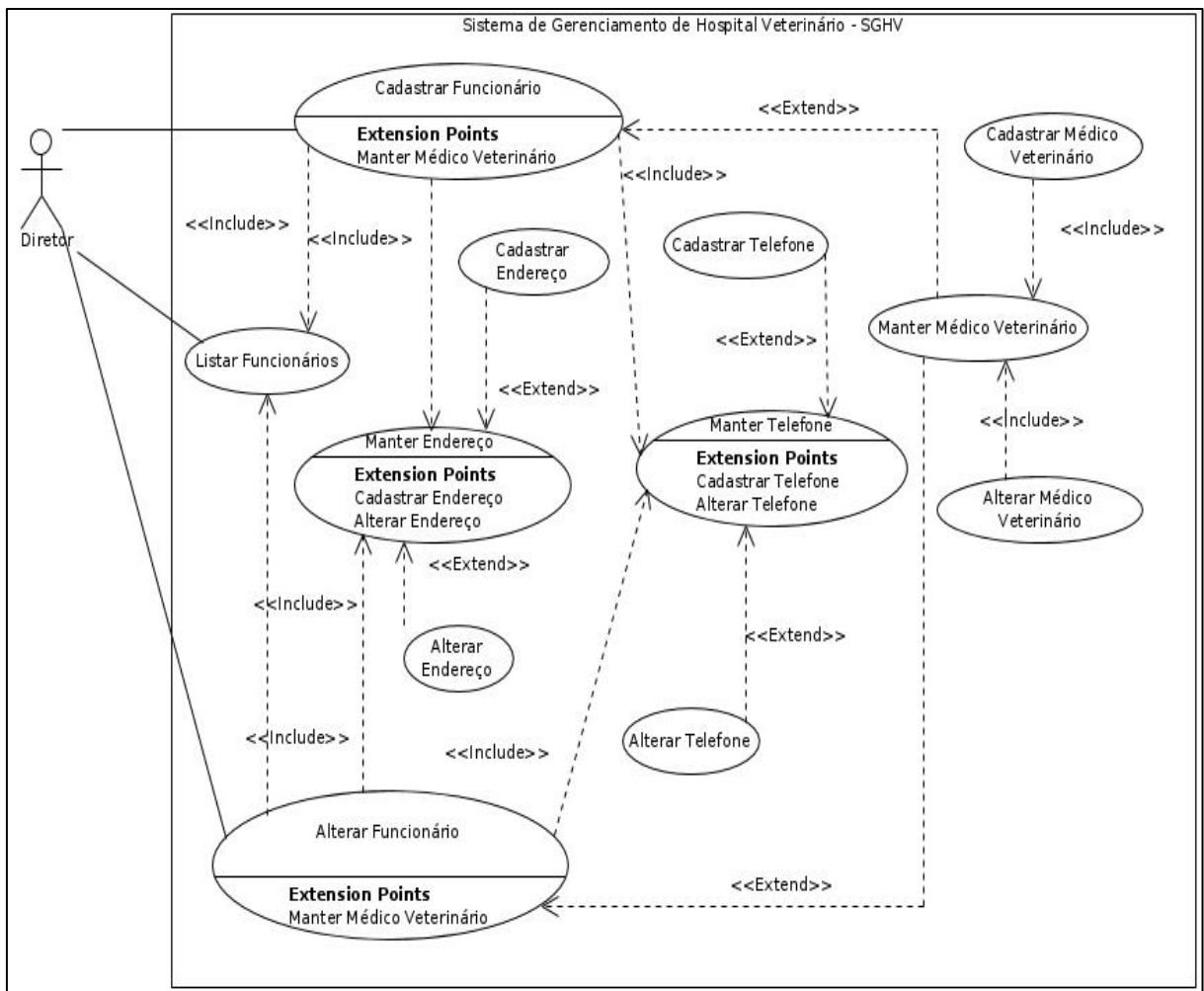


Figura 10 – Caso de Uso- Manter Funcionário

3.4.3.2 Manter Serviço

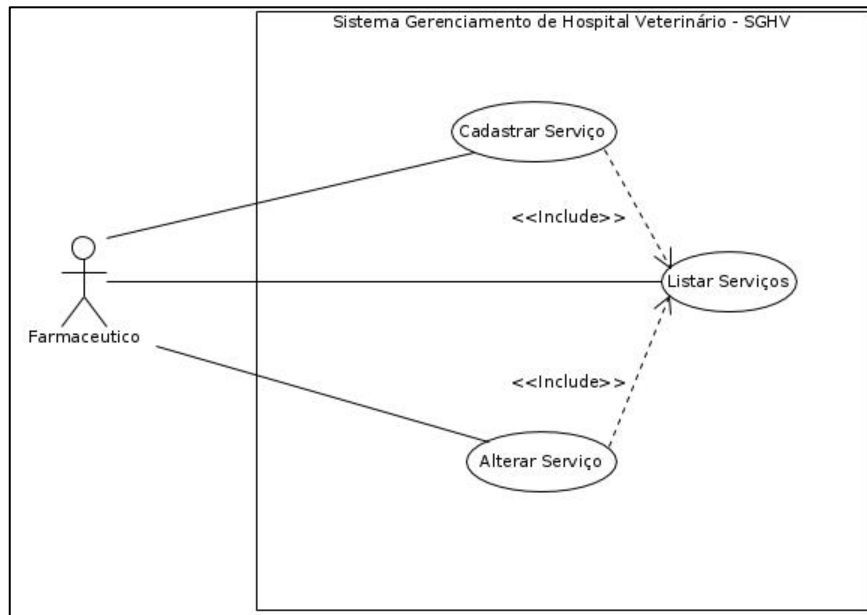


Figura 11 – Caso de Uso – Manter Serviço

3.4.3.2.3 Manter Consulta

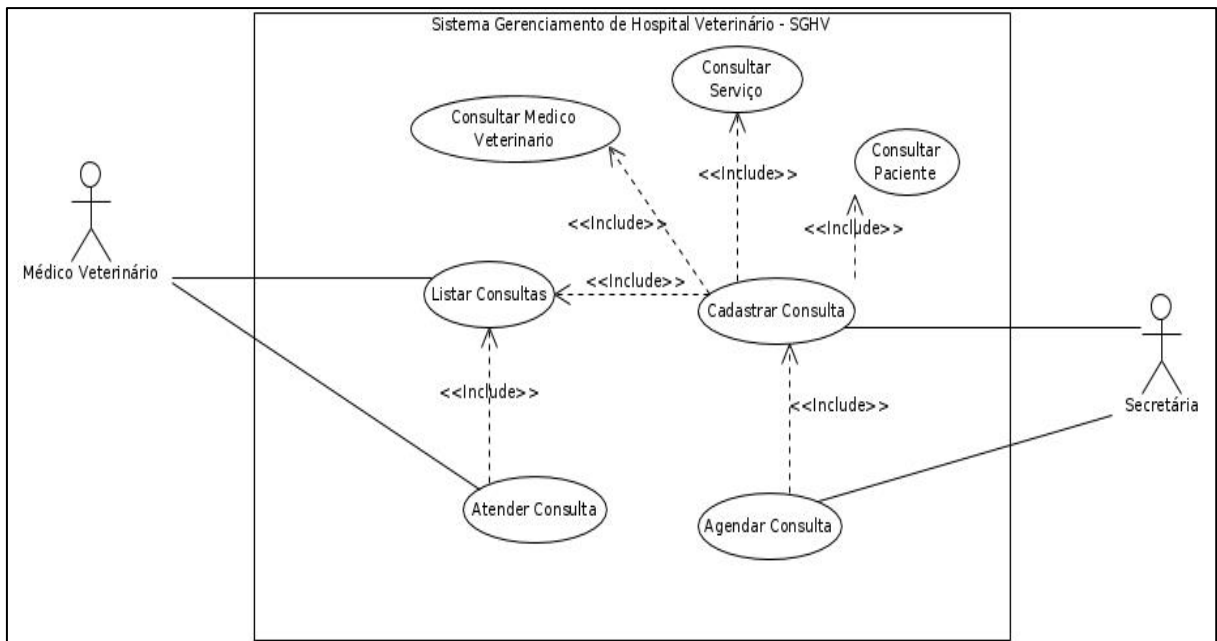


Figura 12- Caso de Uso – Manter Consulta

3.4.3.3 Dificuldades encontradas durante o Sprint 3

Apresentou-se a dificuldade com os escopos de visão (*viewScoped*) dos *Managed Beans* e a passagem de dados de um para outro em um redirecionamento de página. Tal contratempo requereu um maior esforço, o qual acabou não solucionado, ocasionando a determinação de mais um *Sprint* para a solução deste e outros problemas semelhantes.

3.4.4 SPRINT 4

Neste período foram construídos os módulos de Estoque e Procedimento. Como funcionalidades contempladas no Módulo Estoque, temos:

- Cadastrar Item de Estoque: cadastro de um item com categoria, descrição, valor unitário, se este é controlado, além de a sua quantidade mínima e quantidade presente no estoque;
- Listar Estoque: listagem de todos os itens deste em forma de tabela com todos os dados principais destes em campos pesquisáveis e opções que permite de editá-los ou visualizá-los em outra tela;
- Visualizar Item de Estoque: visualização dos dados do item seguido da opção de editá-lo;
- Editar Item de Estoque: função que possibilita alterar os dados do item como valor unitário, quantidade atual e também a quantidade mínima em estoque.

Já no Módulo Procedimento foram feitas:

- Cadastrar Procedimento: cadastro de procedimento com a sua descrição, funcionários envolvidos, serviços realizados, consultas relacionadas, ou ainda os itens de estoque utilizados;
- Listar Procedimentos: listagem dos procedimentos em forma de tabela com todos os dados principais destes e campos pesquisáveis, e ainda as opção de visualiza-los ou edita-los;
- Visualizar Procedimento: visualização dos dados do procedimento;

3.4.4.1 Cronograma do Sprint 4

No quadro abaixo podemos ver como foi elaborado o *Sprint 4* do projeto:

Nome da tarefa	Duração	Início	Término
4ª Sprint	19 dias	Qui 14/08/14	Ter 09/09/14
Diagramas de Casos de Uso	2 dias	Sex 15/08/14	Seg 18/08/14
Manter Estoque	1 dia	Sex 15/08/14	Sex 15/08/14
Manter Procedimento	1 dia	Seg 18/08/14	Seg 18/08/14
Codificação	16 dias	Ter 19/08/14	Ter 09/09/14
Modulo de Estoque	7 dias	Ter 19/08/14	Qua 27/08/14
Cadastrar Item	2 dias	Ter 19/08/14	Qua 20/08/14
Listar Itens	1 dia	Qui 21/08/14	Qui 21/08/14
Visualizar Item	2 dias	Sex 22/08/14	Seg 25/08/14
Editar Item	2 dias	Ter 26/08/14	Qua 27/08/14
Modulo de Procedimento	10 dias	Qua 27/08/14	Ter 09/09/14
Cadastrar Procedimento	4 dias	Qui 28/08/14	Ter 02/09/14
Listar Procedimentos	2 dias	Qua 03/09/14	Qui 04/09/14
Visualizar Procedimento(tela)	2 dias	Seg 08/09/14	Ter 09/09/14

Quadro 6 - Cronograma (*Sprint 4*)

3.4.4.2 Casos de Uso – Sprint 4

Os casos de uso de Manter Estoque e Manter Procedimento foram implementados nesta etapa do projeto.

3.4.4.2.1 Manter Estoque

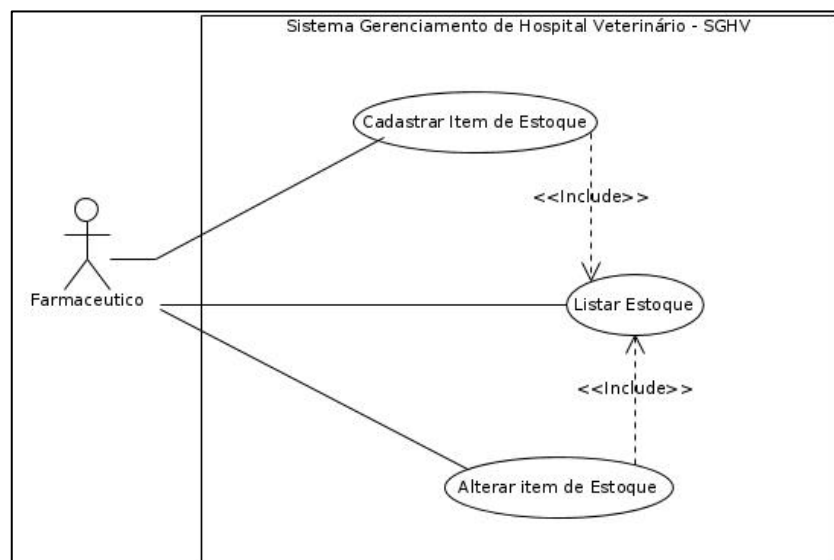


Figura 13 - Caso de Uso- Manter Estoque

3.4.4.2 Manter Procedimento

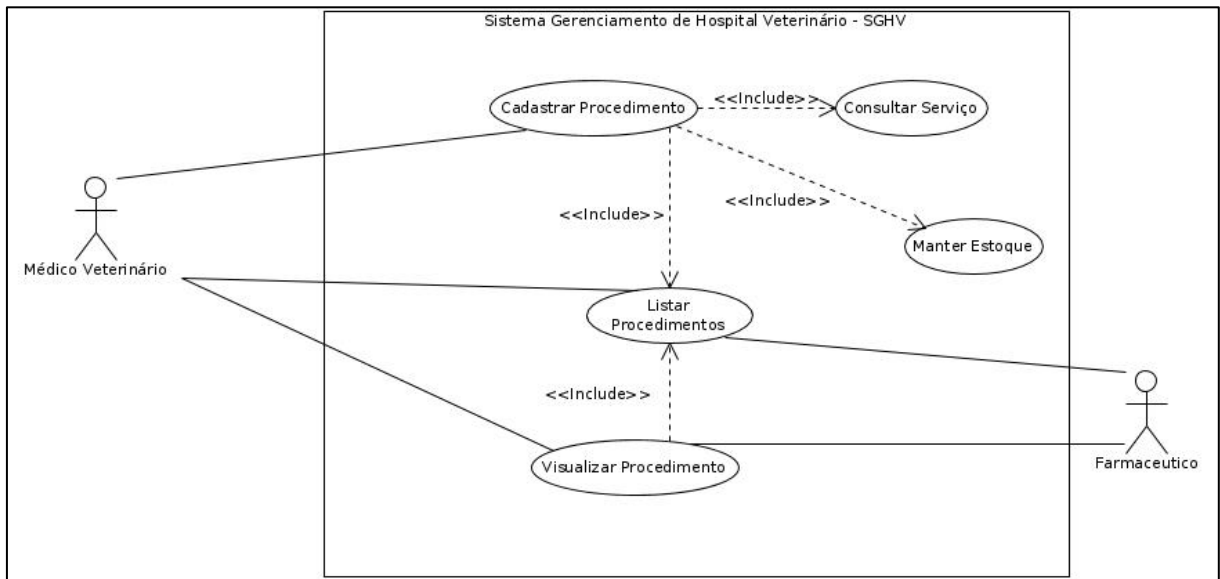


Figura 14 - Caso de Uso – Manter Procedimento

3.4.4.3 Dificuldades Encontradas Durante o Sprint 4

Foi encontrada uma contrariedade com os tipos de “*cascade*” do JPA, que é uma propriedade que determina como se dará as alterações de dados em relacionamentos $N : N$ e $1 : N$, em relação ao procedimento, visto que várias soluções encontradas por meio de pesquisa na *Web*, não se enquadravam com o que o sistema necessitava.

3.4.5 SPRINT 5

Neste incremento foram solucionados os problemas apresentados nos módulos de Cliente, Funcionário, Consulta e Procedimento, nas telas de editar e visualizar, onde em redirecionamentos de páginas, os quais os dados passados de uma tela para outra acabavam se perdendo em tempo de execução. Os controladores das referidas telas possuíam o escopo de visão (*viewScoped*), devido a relacionamentos tipo $N : N$, trabalhados nestas telas, estes controladores possuíam tal escopo, para que os relacionamentos aconteçam de forma transparente, fácil e visual pelo usuário.

Neste período houve também a construção da função de editar Procedimento, permitindo inserir ou retirar envolvidos e serviços, além de inserir mais itens de estoque utilizados. Também foi feita uma revisão em toda a documentação e alguns acertos em gerais no código do projeto.

3.4.5.1 Cronograma do Sprint 5

O *Sprint 5* ocorreu conforme o cronograma apresentado no Quadro 7.

Nome da tarefa	Duração	Início	Término
5ª Sprint	21 dias	Ter 09/09/14	Ter 07/10/14
Revisão da Documentação	5 dias	Qua 10/09/14	Ter 16/09/14
Codificação	16 dias	Ter 16/09/14	Ter 07/10/14
Visualizar Cliente(mecanismo)	2 dias	Qua 17/09/14	Qui 18/09/14
Visualizar Funcionario(mecanismo)	1 dia	Sex 19/09/14	Sex 19/09/14
Visualizar Consulta(mecanismo)	1 dia	Seg 22/09/14	Seg 22/09/14
Visualizar Procedimento(mecanismo)	2 dias	Ter 23/09/14	Qua 24/09/14
Editar Cliente(mecanismo)	2 dias	Qui 25/09/14	Sex 26/09/14
Editar Funcionario(mecanismo)	1 dia	Seg 29/09/14	Seg 29/09/14
Editar Procedimento	2 dias	Ter 30/09/14	Qua 01/10/14
Acertos gerais	4 dias	Qui 02/10/14	Ter 07/10/14

Quadro 7 - Cronograma (*Sprint 5*)

3.4.5.2 Dificuldades Encontradas Durante o Sprint 5

Houve uma incompreensão dos problemas apresentados e as diferenças dos escopos de visão (*viewScoped*) e de requisição (*requestScoped*). Também foi encontrada a dificuldade de achar a solução mais adequada e segura, sem que a mesma incorpore-se mais bibliotecas ou alterasse os padrões do projeto, e colocar esta solução em perfeito funcionamento em todos os locais necessitados.

3.4.6 SPRINT 6

Neste *Sprint* foram criados os Módulos de Caixa e Usuário, neste foi feita toda a implementação do controle de acesso para cada perfil de usuário.

No Módulo de Usuário foram construídas as funções:

- Cadastrar Usuário: atribuição de um usuário com *login*, senha e permissão (perfil de acesso) a um funcionário;
- Listar Usuários: listagem de todos os usuários em forma de tabela, com a opção de edita-los em outra tela;
- Editar Usuário: função que permite alterar a permissão (perfil de acesso) do usuário;
- Alterar Senha: opção que permite alterar a senha do usuário.

No Modulo de Caixa as foram as funcionalidades:

- Inserir pagamento: cadastro de entrada financeira com valor, descrição e data.
- Inserir saída: cadastro de saída financeira com valor, descrição e data.
- Listar caixa: listagem de todas as entradas e saídas financeiras.

3.4.6.1 Cronograma do Sprint 6

O Cronograma do *Sprint 6* é apresentado no Quadro 8:

Nome da tarefa	Duração	Início	Término
6ª Sprint	21 dias	Ter 07/10/14	Ter 04/11/14
Diagrama de Casos de Uso	2 dias	Qua 08/10/14	Qui 09/10/14
Manter Usuário	1 dia	Qua 08/10/14	Qua 08/10/14
Manter Caixa	1 dia	Qui 09/10/14	Qui 09/10/14
Codificação	18 dias	Sex 10/10/14	Ter 04/11/14
Modulo de Usuario	5 dias	Sex 10/10/14	Qui 16/10/14
Cadastrar Usuario	1 dia	Sex 10/10/14	Sex 10/10/14
Listar Usuarios	1 dia	Seg 13/10/14	Seg 13/10/14
Visualizar Usuario	1 dia	Ter 14/10/14	Ter 14/10/14
Editar Usuario	1 dia	Qua 15/10/14	Qua 15/10/14
Alterar Senha	1 dia	Qui 16/10/14	Qui 16/10/14
Criação do mecanismo de login	2 dias	Sex 17/10/14	Seg 20/10/14
Preparação das telas p/ cada perfil	6 dias	Ter 21/10/14	Ter 28/10/14
Administrador	1 dia	Ter 21/10/14	Ter 21/10/14
Secretaria	2 dias	Qua 22/10/14	Qui 23/10/14
Farmacia	1 dia	Sex 24/10/14	Sex 24/10/14
Médico Veterinário	2 dias	Seg 27/10/14	Ter 28/10/14
Modulo de Caixa	5 dias	Qua 29/10/14	Ter 04/11/14
Listar Entradas e Saldas	1 dia	Qua 29/10/14	Qua 29/10/14
Cadastrar Saída	2 dias	Qui 30/10/14	Sex 31/10/14
Cadastrar Entrada(Pagamento)	2 dias	Seg 03/11/14	Ter 04/11/14

Quadro 8 - Cronograma (*Sprint 6*)

3.4.6.2 Caso de Usos

Nesta etapa foram respeitados os casos de uso de Manter Caixa e Manter Usuário.

3.4.6.2.1 Manter Usuário

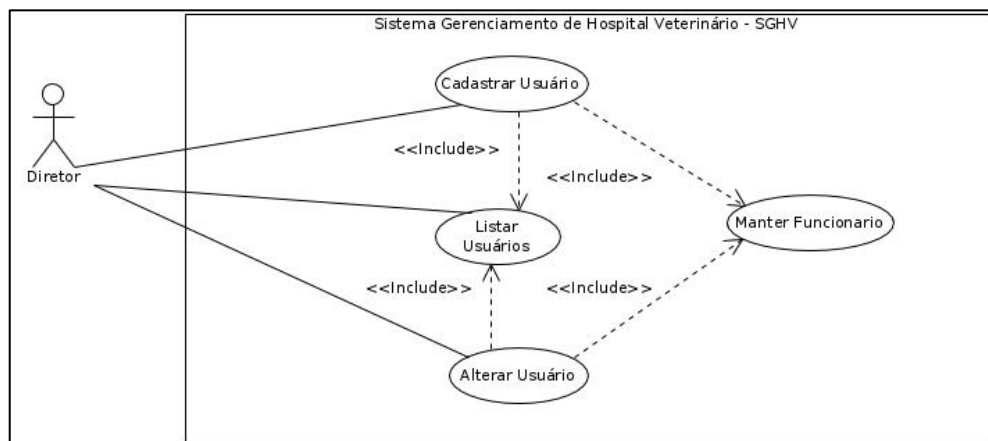


Figura 15 - Caso de Uso – Manter Usuário

3.4.6.2 Manter Caixa

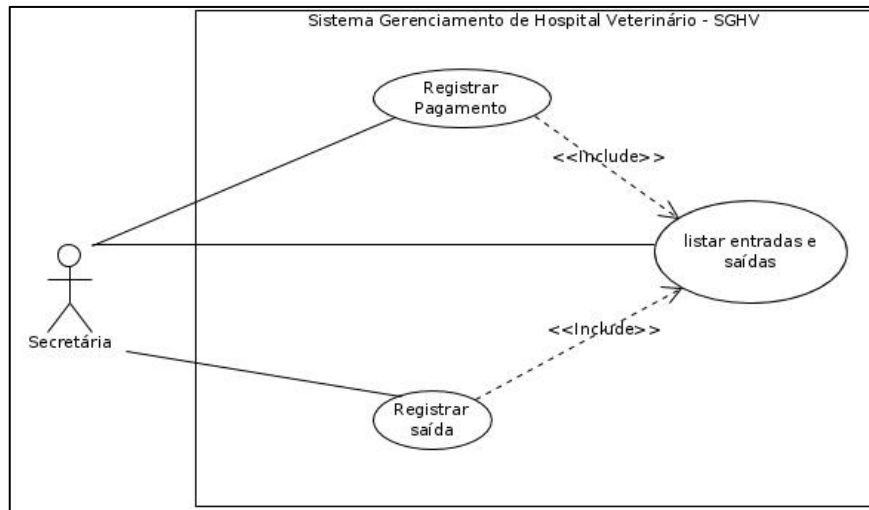


Figura 16 - Caso de Uso – Manter Caixa

3.4.6.3 Dificuldades encontradas durante o Sprint 6

O maior obstáculo encontrado neste incremento foi escolher do melhor método de controle de perfil de acesso do sistema, ao final foi escolhido o controle por meio de filtros, devido à integridade deste método.

3.4.7 SPRINT 7

Nesta etapa foi abordada a melhoria nas apresentações das informações ao usuário, ou seja, apresentar informações de forma inteligíveis e não apenas só dados armazenados. Fez-se uma revisão criteriosa em todas as telas da aplicação visando melhoria da usabilidade, ergonomia e simplicidade na apresentação das informações.

3.4.7.1 Cronograma do Sprint 7

O desenvolvimento do *Sprint 7* ocorreu conforme o Quadro 9.

Nome da tarefa	Duração	Início	Término
7ª Sprint	24 dias	Qua 05/11/14	Seg 08/12/14
Revisão de Telas	24 dias	Qua 05/11/14	Seg 08/12/14
Revisão de Usabilidade	24 dias	Qua 05/11/14	Seg 08/12/14
Revisão do código	24 dias	Qua 05/11/14	Seg 08/12/14
Mostra ao Cliente	24 dias	Qua 05/11/14	Seg 08/12/14
Revisão com o Cliente	24 dias	Qua 05/11/14	Seg 08/12/14
Mostra Final ao Cliente	1 dia	Ter 09/12/14	Ter 09/12/14

Quadro 9 - Cronograma (Sprint 7)

3.4.7.2 Dificuldades Encontradas Durante o Sprint 7

Foi encontrada a falta de compreensão e entendimento em discernir o que é dado armazenado de informação e quando os dados armazenados se torna informação, pois é isso que define um bom sistema de informação. Também foi encontrada a dificuldade em definir qual é o melhor momento, caminho para o acesso e apresentação de uma informação que meu usuário necessita.

4 CONCLUSÃO

Este trabalho teve o intuito de descrever o desenvolvimento, objetivos, materiais e métodos, além de apresentar justificativas para a construção de um sistema *Web* para o gerenciamento de informações de rotina em hospitais veterinários. Visto que o mercado de atuação do *software* apresenta nos últimos anos um crescimento e valores cada vez mais expressivos, onde as aplicações atuantes neste mercado possuem deficiências, de abrangência de informações ou de disponibilidade de acesso.

O SGHV- Sistema de Gerenciamento de Hospitais Veterinários é um sistema que permite o controle de informações pertinentes à rotina diária destes ambientes, buscando uma correta apresentação das informações, sempre propondo caminhos alternativos para estas e almejando uma melhor ergonomia na aplicação.

No decorrer do desenvolvimento do projeto foram encontradas várias dificuldades no âmbito acadêmico, mais especificamente de compreensão de tecnologias e técnicas envolvidas no projeto, tecnologias que são relativamente novas e complexas como Java Server Faces e SGBD PostgreSQL, tais dificuldades se transformaram em força e muito conhecimento.

Com o uso da UML e do modelo de processo Scrum, que propõem um planejamento para a construção de um sistema computacional, pôde-se conscientizar que a parte mais complexa e mais suscetível a erro é o planejamento, visto que a exímia eficiência neste critério, provem do autoconhecimento e prática profissional de desenvolvimento de *software*.

Durante este trabalho verificou-se a possibilidade de continuidade deste, para o aumento de sua abrangência.

4.1 TRABALHOS FUTUROS

No desenrolar do desenvolvimento do projeto verificou-se que algumas funcionalidades e melhorias poderiam ser contempladas neste num futuro próximo. Entre os seguimentos a serem implantados no sistema tem-se:

- Criação de relatórios: como primeiro passo na continuidade do projeto será a geração de outros relatórios pelo sistema, como termos de ciência do proprietário do animal do uso dos serviços do hospital veterinário, lista de débitos do cliente, lista de entradas e saídas, e etc.;

- Melhoria no prontuário: cadastro com dados específicos, desmembrando o atual da consulta, tendo uma tabela no banco de dados separada com relacionamento de N: N entre as tabelas Consultas e Procedimentos;
- Criação do controle de cargos de um funcionário: função que possibilitará um funcionário possuir mais de um cargo ao mesmo tempo, assim a administração conseguirá suprir a falta de outro funcionário, alcançando uma maior flexibilidade ao sistema e o rastreamento dos cargos de um funcionário de forma cronológica;
- Criação dos modelos de exame: cadastro com dados específicos de cada exame e possibilidade do arquivamento de dados multimídia como vídeo e imagens, muito importantes na determinação de um diagnóstico;
- Acesso do cliente (proprietário do animal) ao sistema: permitir, de um modo seguro, o acesso às informações de seu(s) animal(ais), consultas e demais procedimentos realizados.

REFERÊNCIAS

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. Elsevier Brasil, 2006.

CAELUM. **Java para Desenvolvimento Web**. Disponível em: <<http://www.caelum.com.br/apostila-java-web/uma-introducao-pratica-ao-jpa-com-hibernate/>>. Acesso em: 22 dez. 2014.

_____. **Lab. Java com Testes, JSF e Design Patterns**. Disponível em: <<http://www.caelum.com.br/apostila-java-testes-jsf-web-services-design-patterns/introducao-ao-jsf-e-primefaces/#7-2-caracteristicas-do-jsf>>. Acesso em: 22 dez. 2014.

CFMV - **Conselho Federal de Medicina Veterinária**. Disponível em: <http://www.cfmv.org.br/portal/ensino_frm_exibir_opcao.php>. Acesso em: 19 dez. 2014.

DEVMEDIA. **Introdução à JPA - Java Persistence API**. Disponível em: <<http://www.devmedia.com.br/introducao-a-jpa-java-persistence-api/28173>> Acesso em: 22 dez.2014.

FOWLER, Martin. **Padrões de arquitetura de aplicações corporativas**. Bookman, 2006.

HSoluções em Informática. **Microsoft Project 2007**. Disponível em: <<http://www.wcorpsa.com/arquivos/LIVROS-APOSTILAS/Apostila%20Project%202007.pdf>>. Acesso em : 22 dez. 2014.

NetBeans, **Trilha do Aprendizado do Java EE e Java Web**. Disponível em: <https://netbeans.org/kb/trails/java-ee_pt_BR.html>. Acesso em: 22 dez. 2014.

_____, **NetBeansIDE**. Disponível em: < <https://www.netbeans.org/>>. Acesso em: 22 dez. 2014.

PRESSMAN, Roger. **Engenharia de Software**. 6.ed.São Paulo: Mcgraw-hill /Tecmedd, 2006.

pgAdmin. Disponível em:< <http://www.pgadmin.org/>> .Acesso em 22 dez. 2014.

PostgreSQL.. Disponível em: < <http://www.postgresql.org/>>. Acesso em: 22 dez. 2014.

PrimeFaces .Disponível em:< <http://primefaces.org/> >.Acesso em:22 dez. 2014.

4Linux. **O QUE É POSTGRESSQL?**.Disponível em:< <http://www.4linux.com.br/o-que-e-postgresql>>.Acesso em:22 dez. 2014.

Zoetis Brasil. **Mercado de Saúde Animal**. Disponível em:
<<http://www.zoetis.com.br/Mercado-de-Saude-Animal>>. Acesso em: 19 dez. 2014.

Universidade Federal de Santa Catarina. **Microsoft Word - Comandos básicos**. Disponível em: <<http://www.inf.ufsc.br/~barreto/cca/tratexto/word.html>>.Acesso em:22 dez. 2014

Universidade Federal de Viçosa. **Tutorial Visual Paradigm for UML**. Disponível em:
<http://www.dpi.ufv.br/projetos/geoprofile/tutoriais/Visual_Paradigm_for_UML_Tutorial.pdf
>. Acesso em: 22 dez. 2014.

XMind. Disponível em: < <https://www.xmind.net/>>. Acesso em:22 dez. 2014

APÊNDICE A - ARTEFATOS GERADOS

CRONOGRAMA

Nome da tarefa	Duração	Início	Término
+ 1ª Sprint	29 dias	Seg 28/04/14	Qui 05/06/14
+ 2ª Sprint	24 dias	Sex 06/06/14	Qua 09/07/14
+ 3ª Sprint	26 dias	Qui 10/07/14	Qui 14/08/14
+ 4ª Sprint	19 dias	Qui 14/08/14	Ter 09/09/14
+ 5ª Sprint	21 dias	Ter 09/09/14	Ter 07/10/14
+ 6ª Sprint	21 dias	Ter 07/10/14	Ter 04/11/14
+ 7ª Sprint	24 dias	Qua 05/11/14	Seg 08/12/14
Mostra Final ao Cliente	1 dia	Ter 09/12/14	Ter 09/12/14

CRONOGRAMA-SPRINT 1

Nome da tarefa	Duração	Início	Término	Predecessoras
- 1ª Sprint	29 dias	Seg 28/04/14	Qui 05/06/14	
Determinação dos requisitos	8 dias	Seg 28/04/14	Qua 07/05/14	
Diagrama de Casos de Uso Geral	2 dias	Qui 08/05/14	Sex 09/05/14	2
Diagrama Entidade Relacionamento	8 dias	Seg 12/05/14	Qua 21/05/14	3
Diagrama de Classes	8 dias	Qui 22/05/14	Seg 02/06/14	4
Diagrama de Navegação	3 dias	Ter 03/06/14	Qui 05/06/14	5

CRONOGRAMA-SPRINT 2

Nome da tarefa	Duração	Início	Término	Predecessoras
- 2ª Sprint	24 dias	Sex 06/06/14	Qua 09/07/14	1
Criação da Base de Dados	2 dias	Seg 09/06/14	Ter 10/06/14	6
Criação do Projeto na IDE	2 dias	Qua 11/06/14	Qui 12/06/14	8
- Diagramas de Casos de Uso	2 dias	Sex 13/06/14	Seg 16/06/14	9
Manter Cliente	1 dia	Sex 13/06/14	Sex 13/06/14	9
Manter Paciente	1 dia	Seg 16/06/14	Seg 16/06/14	11
- Codificação	15 dias	Qua 18/06/14	Ter 08/07/14	10
- Modulo de Cliente	9 dias	Qua 18/06/14	Seg 30/06/14	10
Cadastrar Cliente	5 dias	Ter 17/06/14	Seg 23/06/14	10
Listar Clientes	1 dia	Ter 24/06/14	Ter 24/06/14	15
Visualizar Cliente(tela)	1 dia	Qua 25/06/14	Qua 25/06/14	16
Editar Cliente(tela)	1 dia	Qui 26/06/14	Qui 26/06/14	17
- Modulo de Paciente	5 dias	Ter 01/07/14	Seg 07/07/14	14
Cadastrar Paciente	2 dias	Ter 01/07/14	Qua 02/07/14	14
Listar Pacientes	1 dia	Qui 03/07/14	Qui 03/07/14	20
Visualizar Paciente	1 dia	Sex 04/07/14	Sex 04/07/14	21
Editar Paciente	1 dia	Seg 07/07/14	Seg 07/07/14	22

CRONOGRAMA-SPRINT 3

Nome da tarefa	Duração	Início	Término	Predecessoras
3ª Sprint	26 dias	Qui 10/07/14	Qui 14/08/14	7
Diagramas de Casos de Uso	3 dias	Qui 10/07/14	Seg 14/07/14	7
Manter Funcionario	1 dia	Qui 10/07/14	Qui 10/07/14	7
Manter Serviços	1 dia	Sex 11/07/14	Sex 11/07/14	26
Manter Consultas	1 dia	Seg 14/07/14	Seg 14/07/14	27
Codificação	23 dias	Ter 15/07/14	Qui 14/08/14	25
Modulo Funcionarios	10 dias	Ter 15/07/14	Seg 28/07/14	25
Cadastrar Funcionario	5 dias	Ter 15/07/14	Seg 21/07/14	25
Listar Funcionarios	1 dia	Ter 22/07/14	Ter 22/07/14	31
Visualizar Funcionario(tela)	2 dias	Qua 23/07/14	Qui 24/07/14	32
Editar Funcionario(tela)	2 dias	Sex 25/07/14	Seg 28/07/14	33
Modulo Serviços	5 dias	Ter 29/07/14	Seg 04/08/14	30
Cadastrar Serviço	1 dia	Ter 29/07/14	Ter 29/07/14	30
Listar Serviços	1 dia	Qua 30/07/14	Qua 30/07/14	36
Visualizar Serviços	1 dia	Qui 31/07/14	Qui 31/07/14	37
Editar Serviço	1 dia	Seg 04/08/14	Seg 04/08/14	38
Modulo Consultas	8 dias	Ter 05/08/14	Qui 14/08/14	35
Cadastrar Consulta	4 dias	Ter 05/08/14	Sex 08/08/14	35
Listar Consultas	1 dia	Seg 11/08/14	Seg 11/08/14	41
Visualizar Consulta(tela)	2 dias	Ter 12/08/14	Qua 13/08/14	42

CRONOGRAMA-SPRINT 4

Nome da tarefa	Duração	Início	Término	Predecessoras
4ª Sprint	19 dias	Qui 14/08/14	Ter 09/09/14	24
Diagramas de Casos de Uso	2 dias	Sex 15/08/14	Seg 18/08/14	24
Manter Estoque	1 dia	Sex 15/08/14	Sex 15/08/14	24
Manter Procedimento	1 dia	Seg 18/08/14	Seg 18/08/14	46
Codificação	16 dias	Ter 19/08/14	Ter 09/09/14	45
Modulo de Estoque	7 dias	Ter 19/08/14	Qua 27/08/14	45
Cadastrar Item	2 dias	Ter 19/08/14	Qua 20/08/14	45
Listar Itens	1 dia	Qui 21/08/14	Qui 21/08/14	50
Visualizar Item	2 dias	Sex 22/08/14	Seg 25/08/14	51
Editar Item	2 dias	Ter 26/08/14	Qua 27/08/14	52
Modulo de Procedimentos	10 dias	Qua 27/08/14	Ter 09/09/14	49
Cadastrar Procedimento	4 dias	Qui 28/08/14	Ter 02/09/14	49
Listar Procedimentos	2 dias	Qua 03/09/14	Qui 04/09/14	55
Visualizar Procedimento(tela)	2 dias	Seg 08/09/14	Ter 09/09/14	56

CRONOGRAMA-SPRINT 5

Nome da tarefa	Duração	Início	Término	Predecessoras
5ª Sprint	21 dias	Ter 09/09/14	Ter 07/10/14	44
Revisão da Documentação	5 dias	Qua 10/09/14	Ter 16/09/14	44
Codificação	16 dias	Ter 16/09/14	Ter 07/10/14	59
Visualizar Cliente(mecanismo)	2 dias	Qua 17/09/14	Qui 18/09/14	59
Visualizar Funcionario(mecanismo)	1 dia	Sex 19/09/14	Sex 19/09/14	61
Visualizar Consulta(mecanismo)	1 dia	Seg 22/09/14	Seg 22/09/14	62
Visualizar Procedimento(mecanismo)	2 dias	Ter 23/09/14	Qua 24/09/14	63
Editar Cliente(mecanismo)	2 dias	Qui 25/09/14	Sex 26/09/14	64
Editar Funcionario(mecanismo)	1 dia	Seg 29/09/14	Seg 29/09/14	65
Editar Procedimento	2 dias	Ter 30/09/14	Qua 01/10/14	66
Acertos gerais	4 dias	Qui 02/10/14	Ter 07/10/14	67

CRONOGRAMA-SPRINT 6

Nome da tarefa	Duração	Início	Término	Predecessoras
6ª Sprint	21 dias	Ter 07/10/14	Ter 04/11/14	58
Diagrama de Casos de Uso	2 dias	Qua 08/10/14	Qui 09/10/14	58
Manter Usuário	1 dia	Qua 08/10/14	Qua 08/10/14	58
Manter Caixa	1 dia	Qui 09/10/14	Qui 09/10/14	71
Codificação	18 dias	Sex 10/10/14	Ter 04/11/14	70
Modulo de Usuario	5 dias	Sex 10/10/14	Qui 16/10/14	70
Cadastrar Usuario	1 dia	Sex 10/10/14	Sex 10/10/14	70
Listar Usuarios	1 dia	Seg 13/10/14	Seg 13/10/14	75
Visualizar Usuario	1 dia	Ter 14/10/14	Ter 14/10/14	76
Editar Usuario	1 dia	Qua 15/10/14	Qua 15/10/14	77
Alterar Senha	1 dia	Qui 16/10/14	Qui 16/10/14	78
Criação do mecanismo de login	2 dias	Sex 17/10/14	Seg 20/10/14	79
Preparação das telas p/ cada perfil	6 dias	Ter 21/10/14	Ter 28/10/14	80
Administrador	1 dia	Ter 21/10/14	Ter 21/10/14	74
Secretaria	2 dias	Qua 22/10/14	Qui 23/10/14	82
Farmacia	1 dia	Sex 24/10/14	Sex 24/10/14	83
Médico Veterinário	2 dias	Seg 27/10/14	Ter 28/10/14	84
Modulo de Caixa	5 dias	Qua 29/10/14	Ter 04/11/14	81
Listar Entradas e Saidas	1 dia	Qua 29/10/14	Qua 29/10/14	81
Cadastrar Saida	2 dias	Qui 30/10/14	Sex 31/10/14	87
Cadastrar Entrada(Pagamento)	2 dias	Seg 03/11/14	Ter 04/11/14	88

CRONOGRAMA-SPRINT 7

Nome da tarefa	Duração	Início	Término	Predecessoras
7ª Sprint	24 dias	Qua 05/11/14	Seg 08/12/14	69
Revisão de Telas	24 dias	Qua 05/11/14	Seg 08/12/14	69
Revisão de Usabilidade	24 dias	Qua 05/11/14	Seg 08/12/14	69
Revisão do código	24 dias	Qua 05/11/14	Seg 08/12/14	69
Mostra ao Cliente	24 dias	Qua 05/11/14	Seg 08/12/14	69
Revisão com o Cliente	24 dias	Qua 05/11/14	Seg 08/12/14	69
Mostra Final ao Cliente	1 dia	Ter 09/12/14	Ter 09/12/14	90

LISTA DE REQUISITOS FUNCIONAIS

	Requisito	Risco	Prioridade
R1	Manter registro de Clientes	Alto	Alta
R2	Manter registro de Pacientes	Baixo	Alta
R3	Gerenciar Serviços	Baixo	Alta
R4	Gerenciar Estoque	Baixo	Alta
R5	Gerenciar Consultas	Alto	Alta
R6	Gerenciar Procedimentos	Alto	Alta
R7	Manter Caixa	Alto	Alta
R8	Gerenciar Usuários	Alto	Alta
R9	Manter registro de Funcionários	Alto	Alta

DIAGRAMA DE CASOS DE USO-MANTER CLIENTE

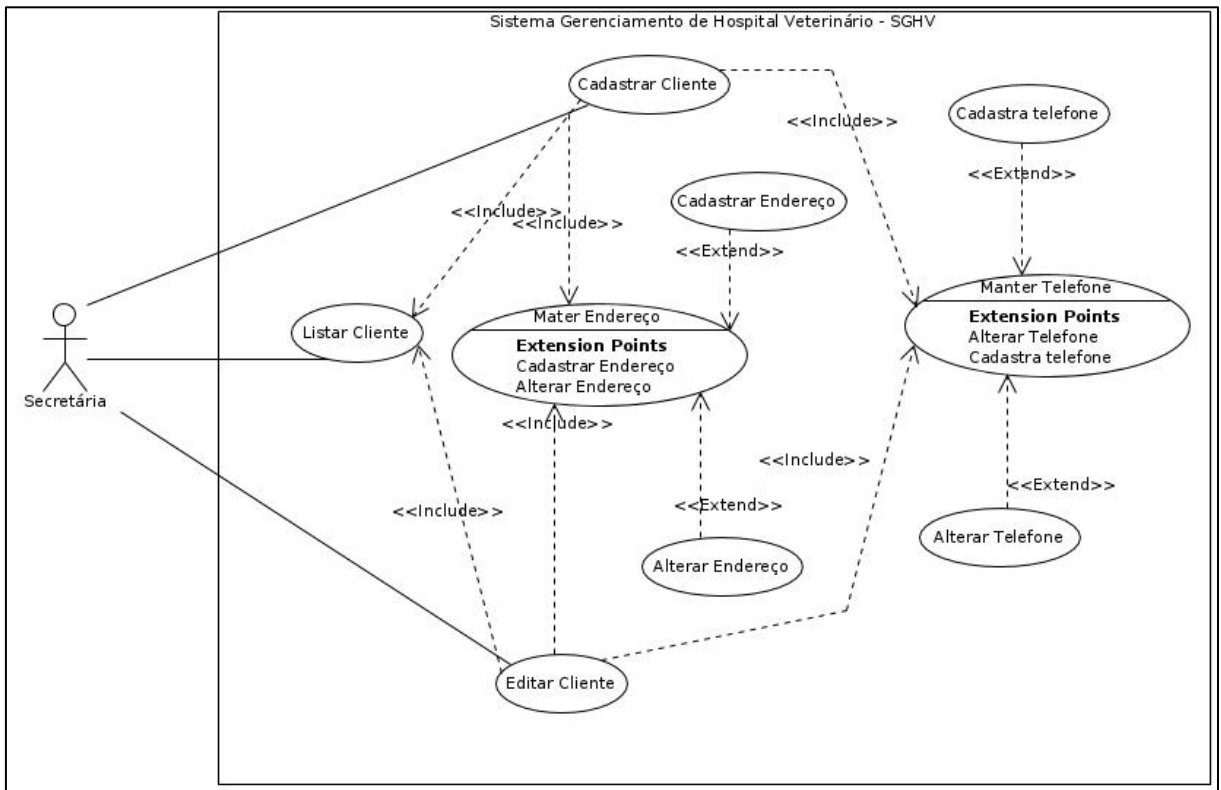


DIAGRAMA DE CASOS DE USO-MANTER CAIXA

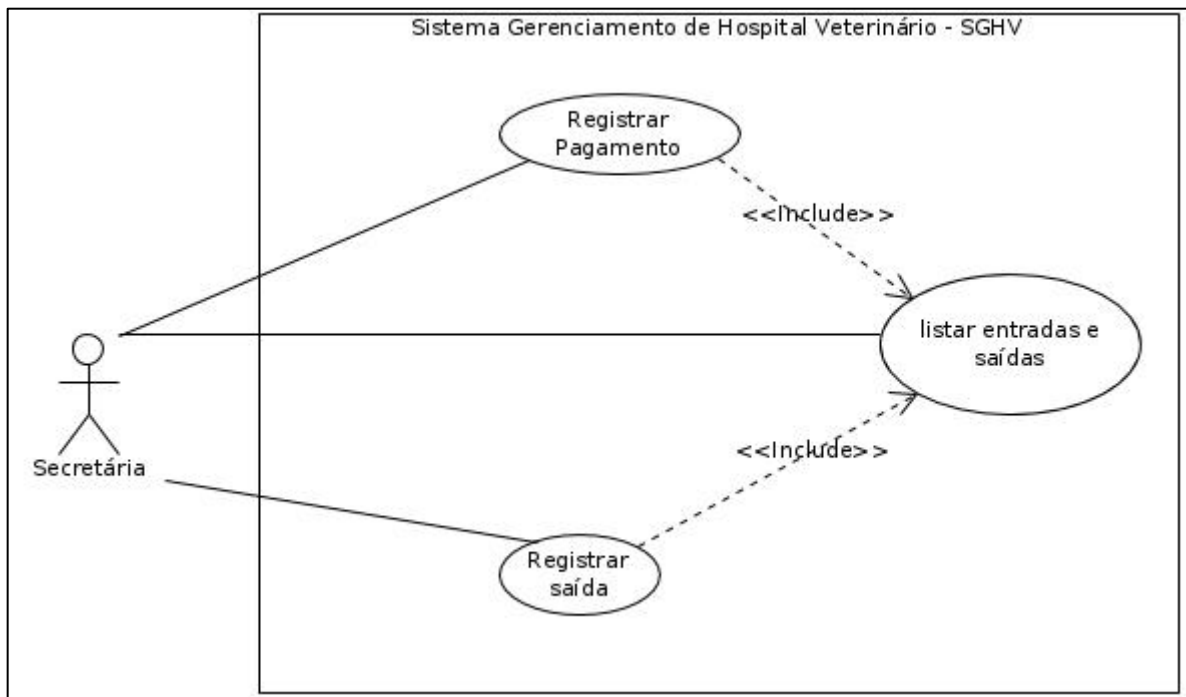


DIAGRAMA DE CASOS DE USO-MANTER CONSULTA

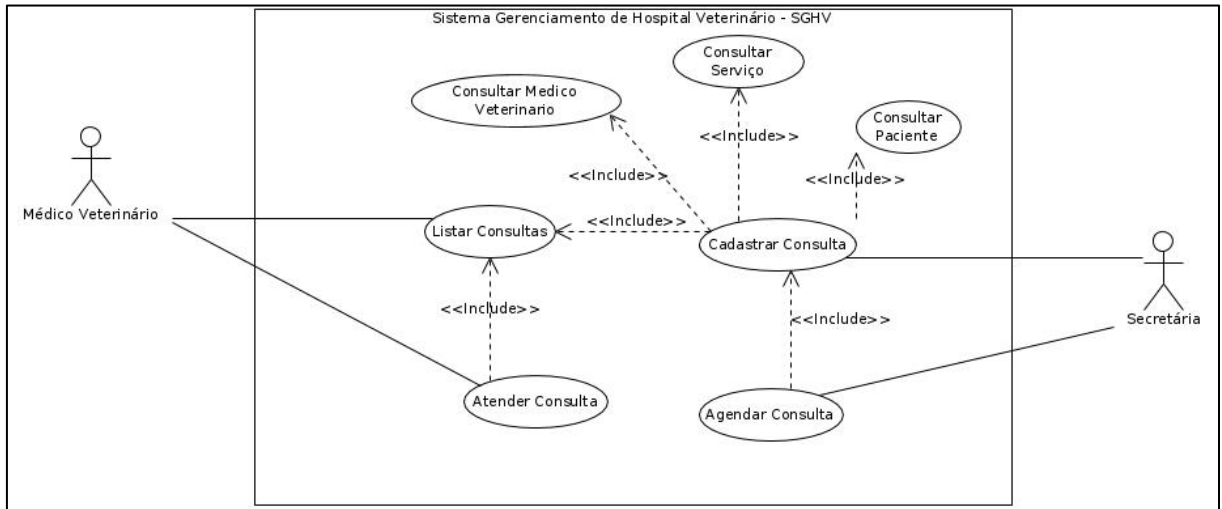


DIAGRAMA DE CASOS DE USO-MANTER ESTOQUE

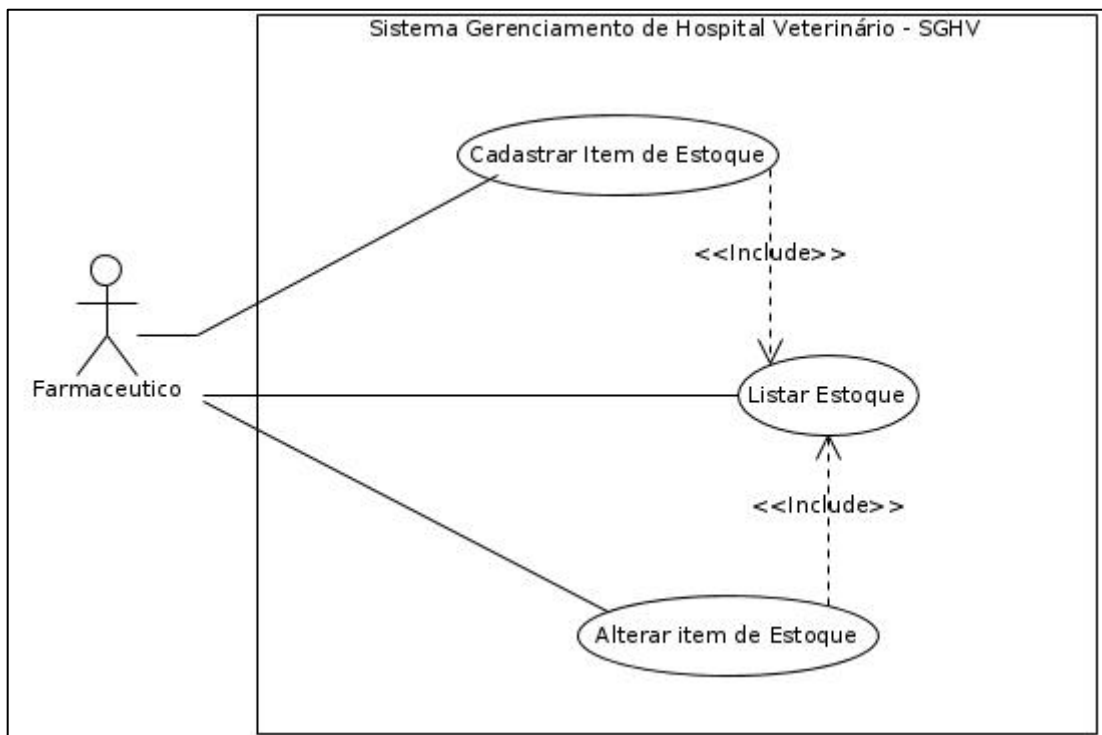


DIAGRAMA DE CASOS DE USO-MANTER FUNCIONARIO

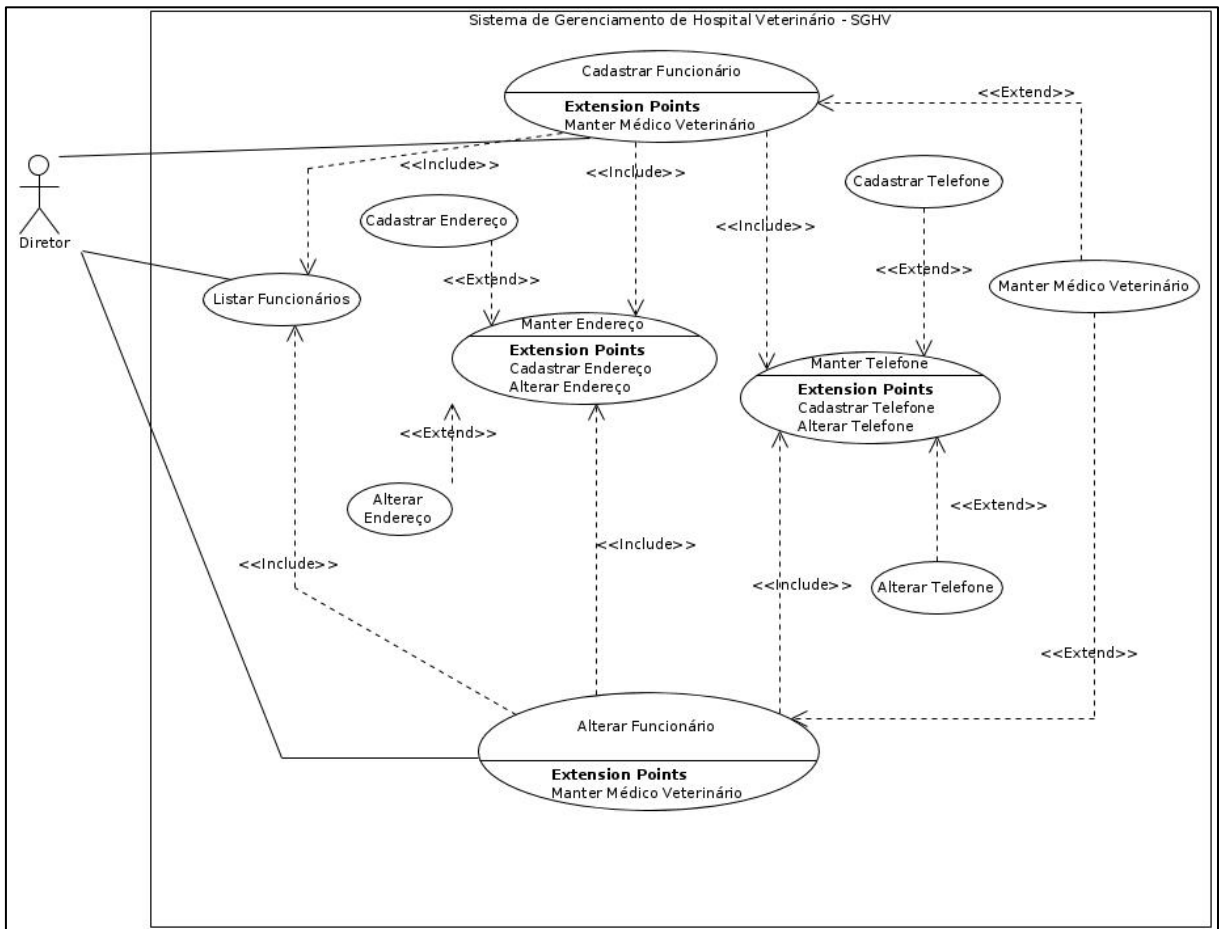


DIAGRAMA DE CASOS DE USO-MANTER PROCEDIMENTO

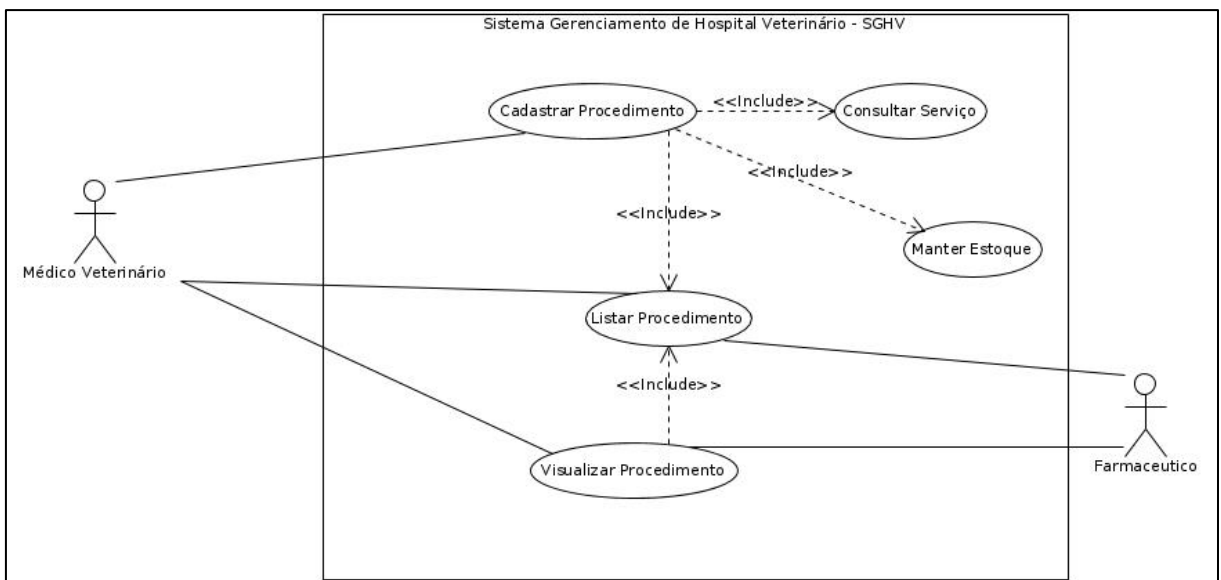


DIAGRAMA DE CASOS DE USO- MANTER SERVIÇO

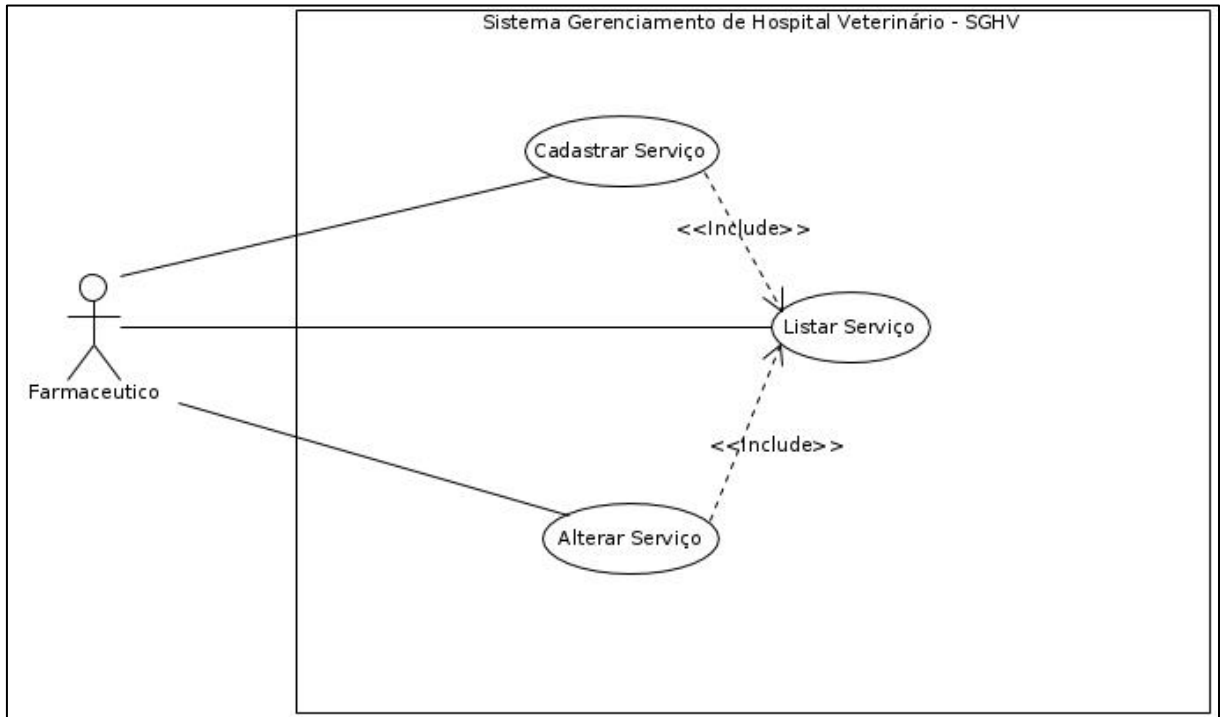


DIAGRAMA DE CASOS DE USO-MANTER USUARIO

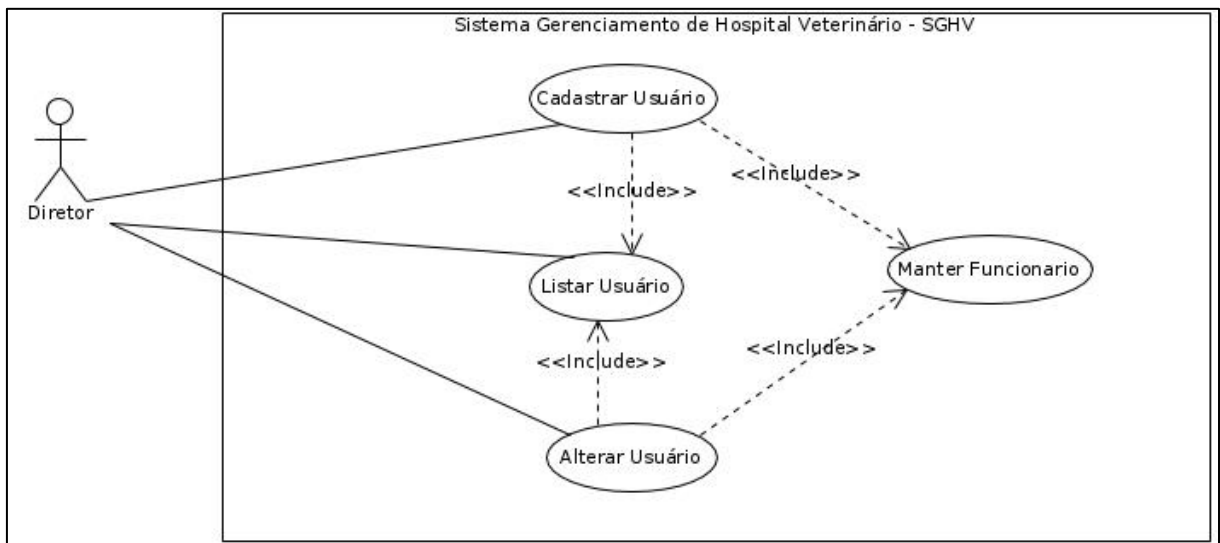


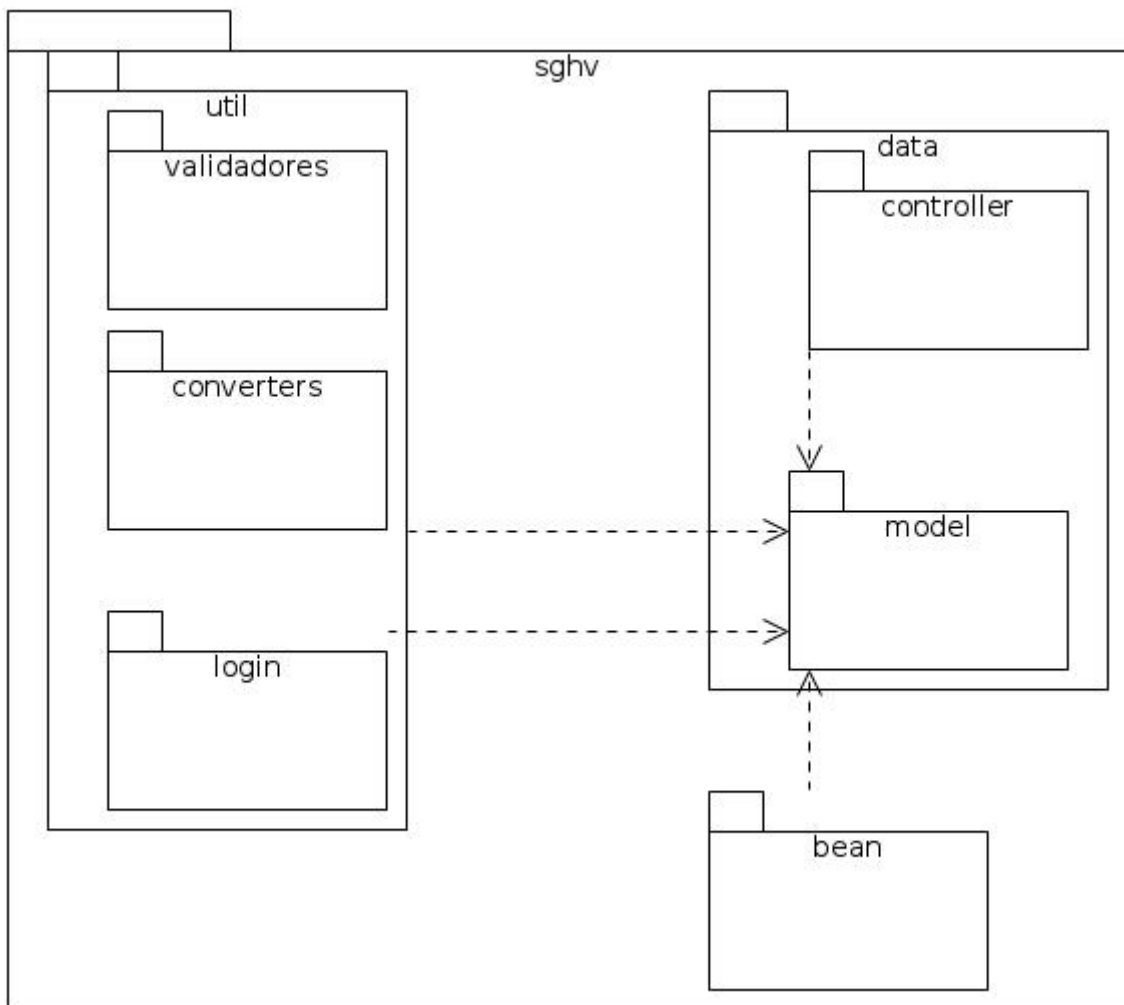
DIAGRAMA DE PACOTES

DIAGRAMA DE CLASSES DO PACOTE SGHV.BEAN

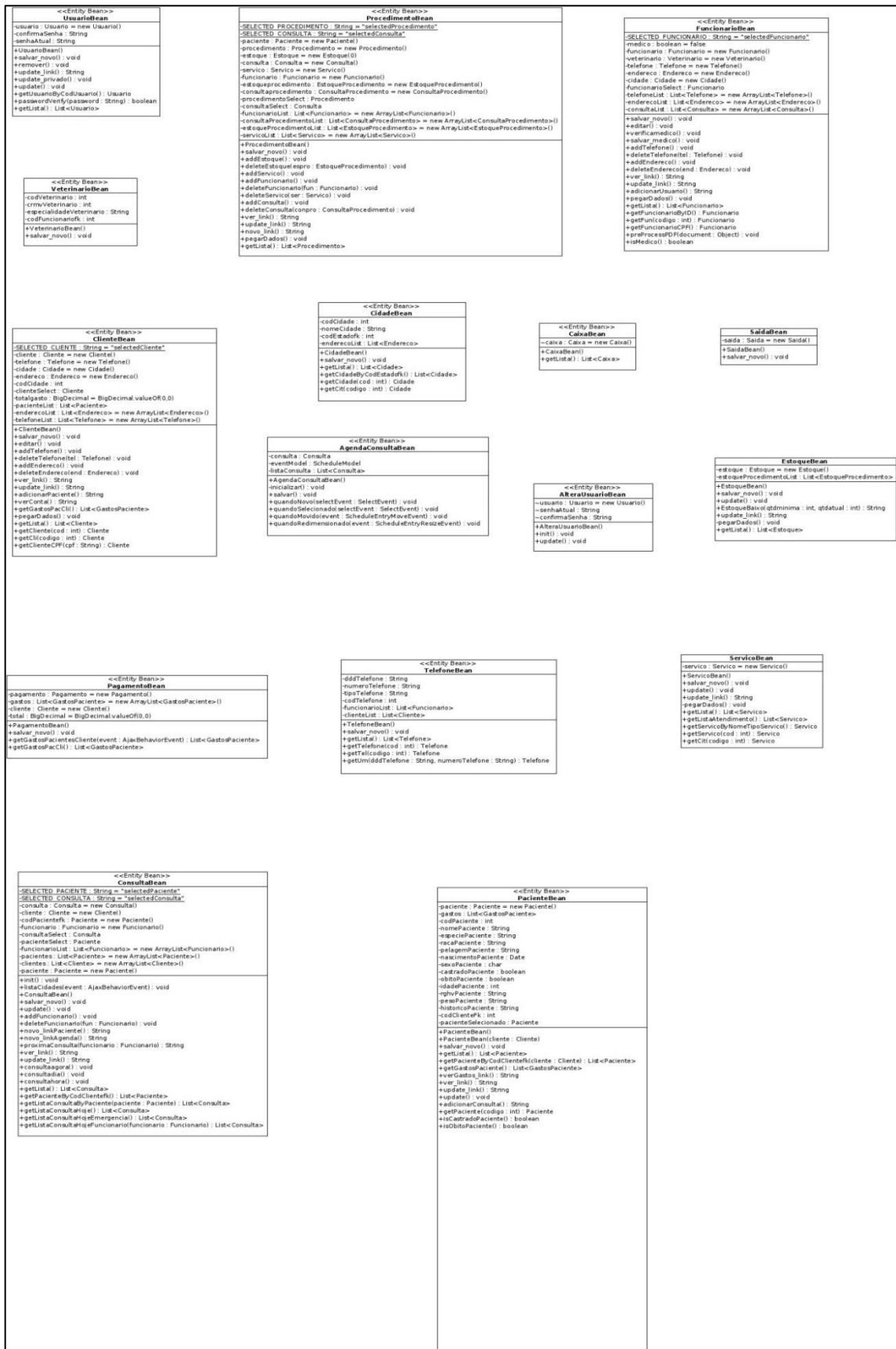


DIAGRAMA DE CLASSES DO PACOTE SGHV.DATA.CONTROLLER



DIAGRAMA DE CLASSES DO PACOTE SGHV.UTIL.CONVERTER

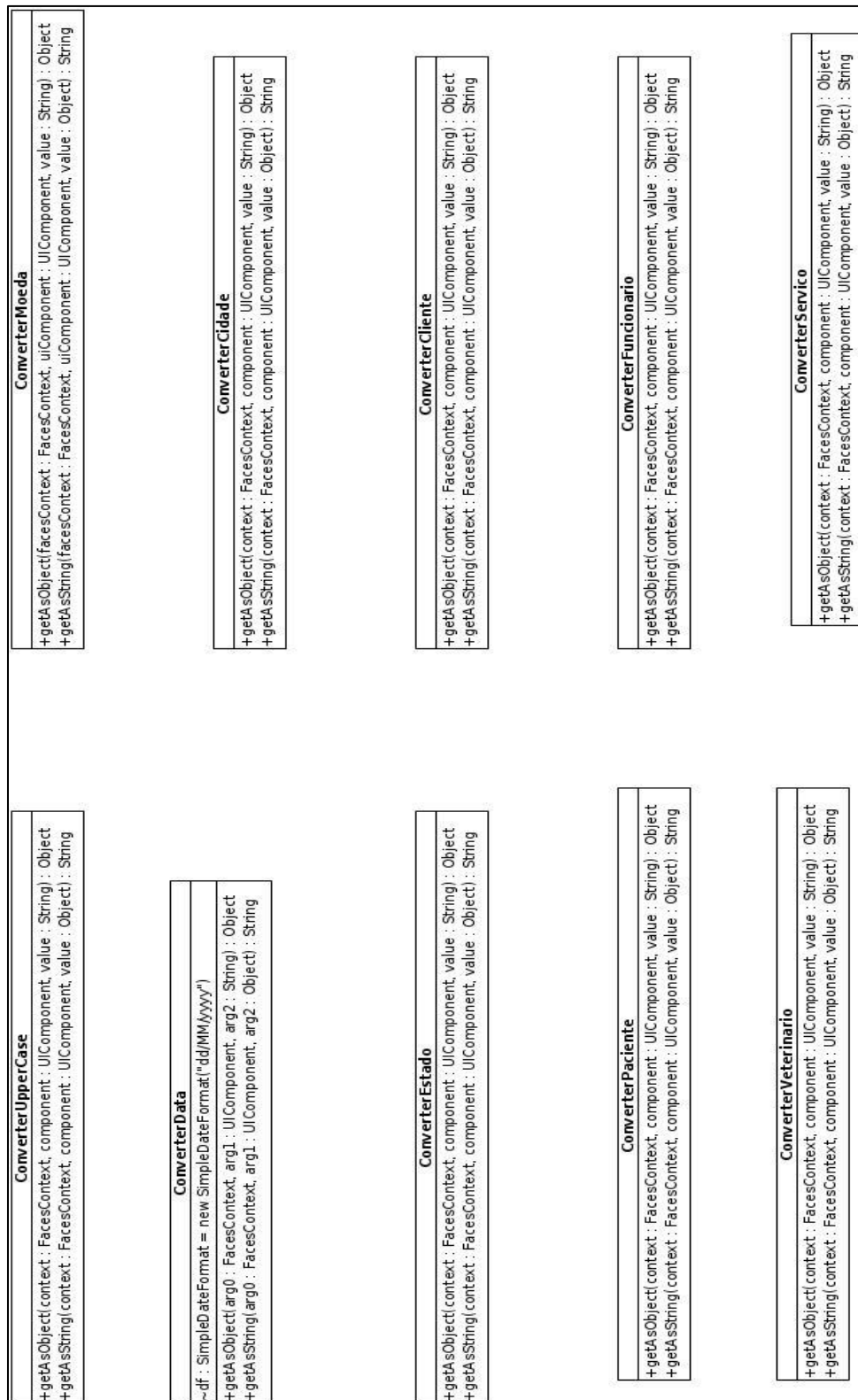


DIAGRAMA DE CLASSES DO PACOTE SGHV.UTIL.LOGIN

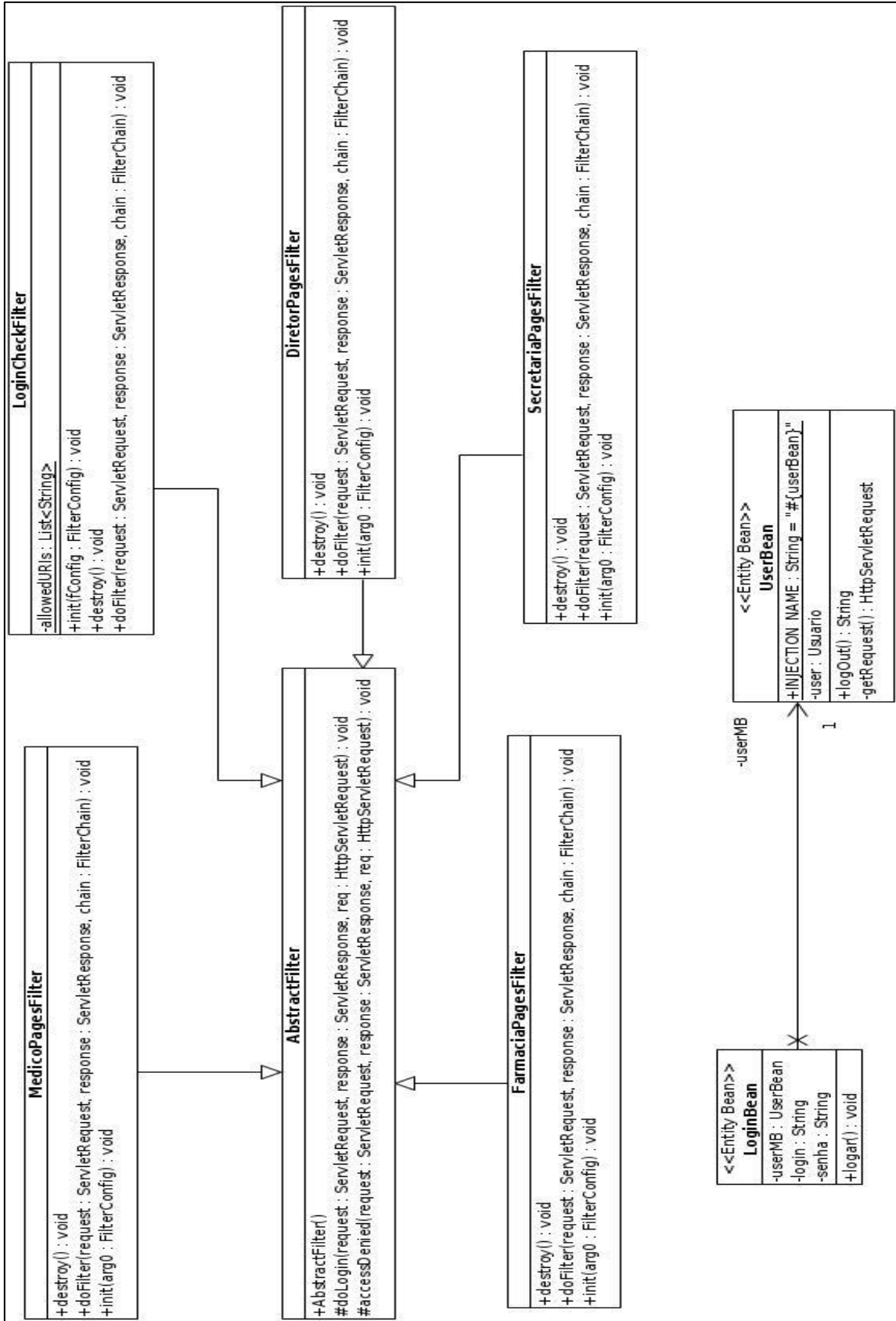


DIAGRAMA DE CLASSES DO PACOTE SGHV.UTIL.VALIDADORES

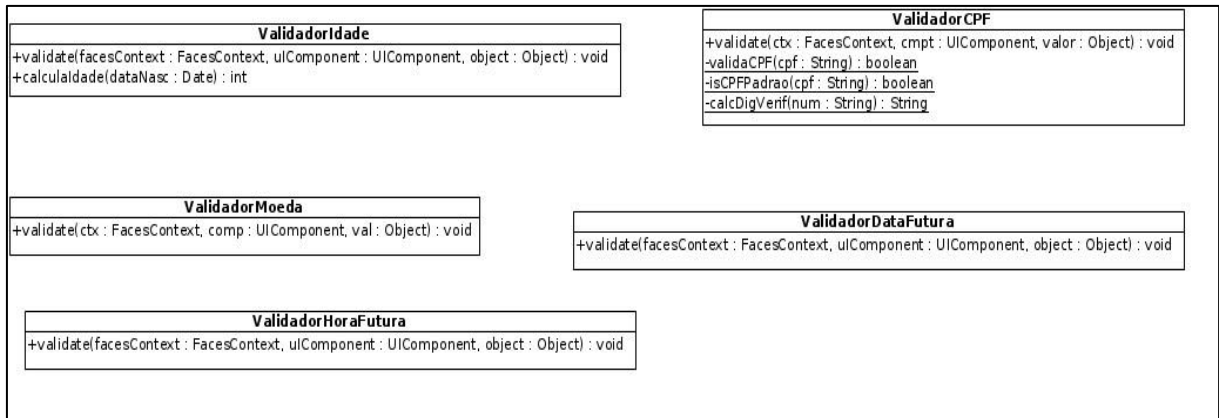


DIAGRAMA DE CLASSES DO PACOTE SGHV.UTIL

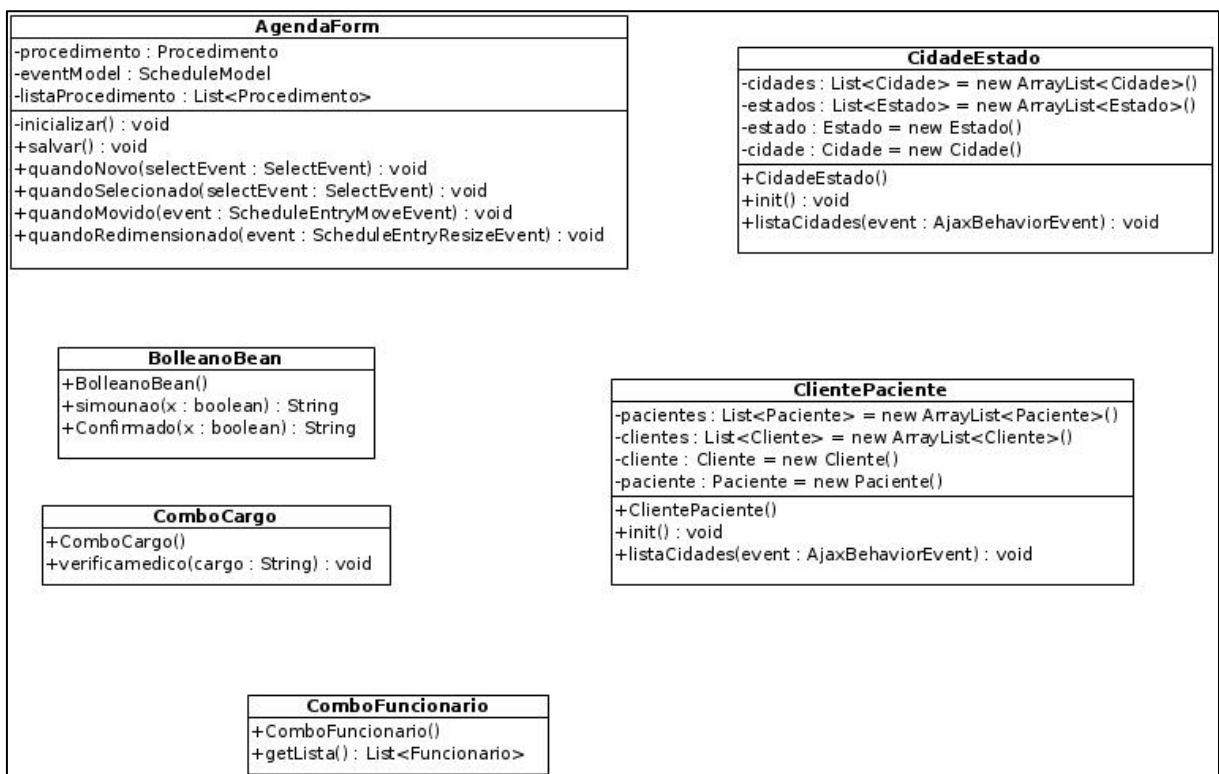


DIAGRAMA DE NAVEGAÇÃO

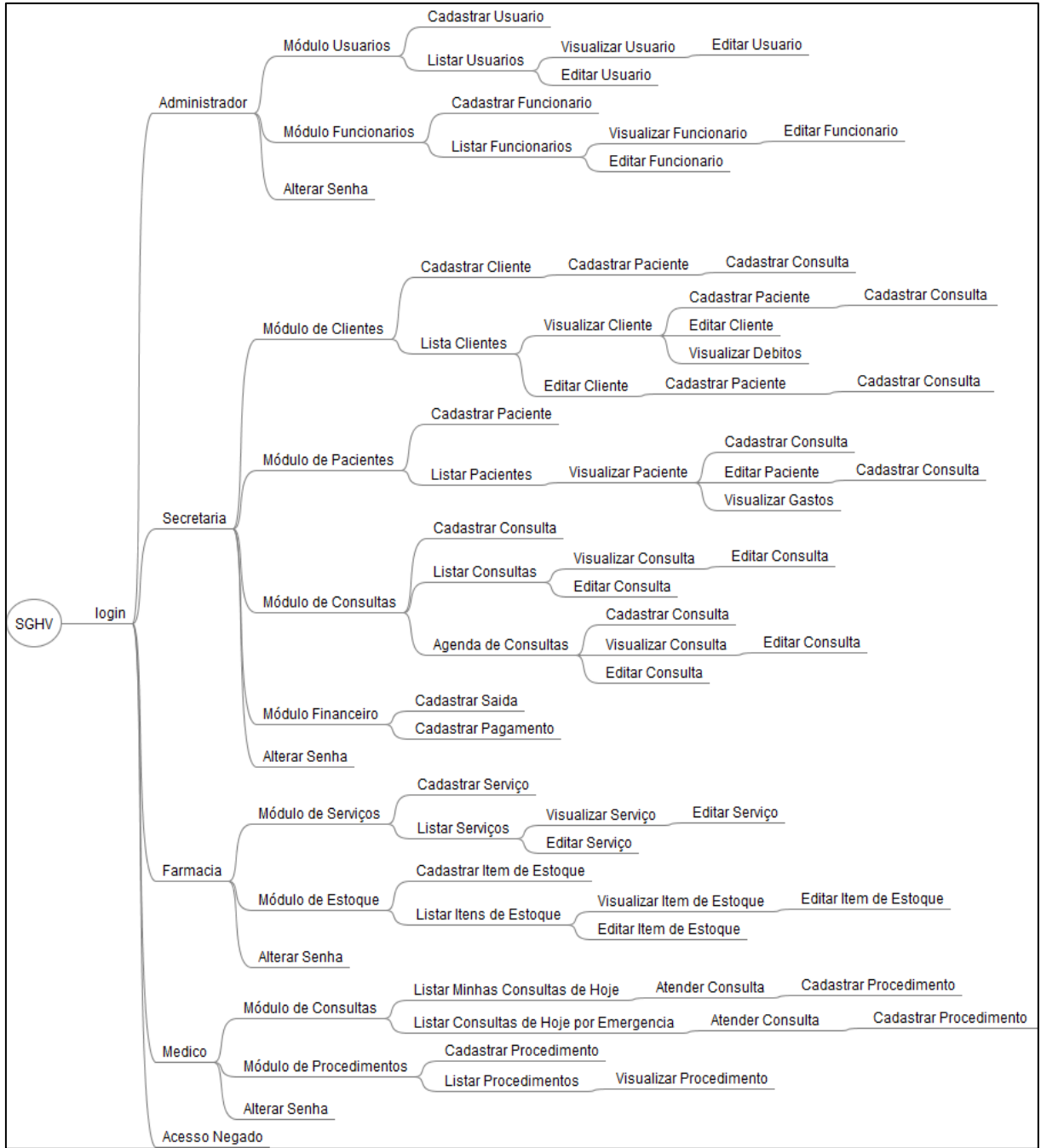


DIAGRAMA DE SEQUENCIA- CADASTRAR CLIENTE

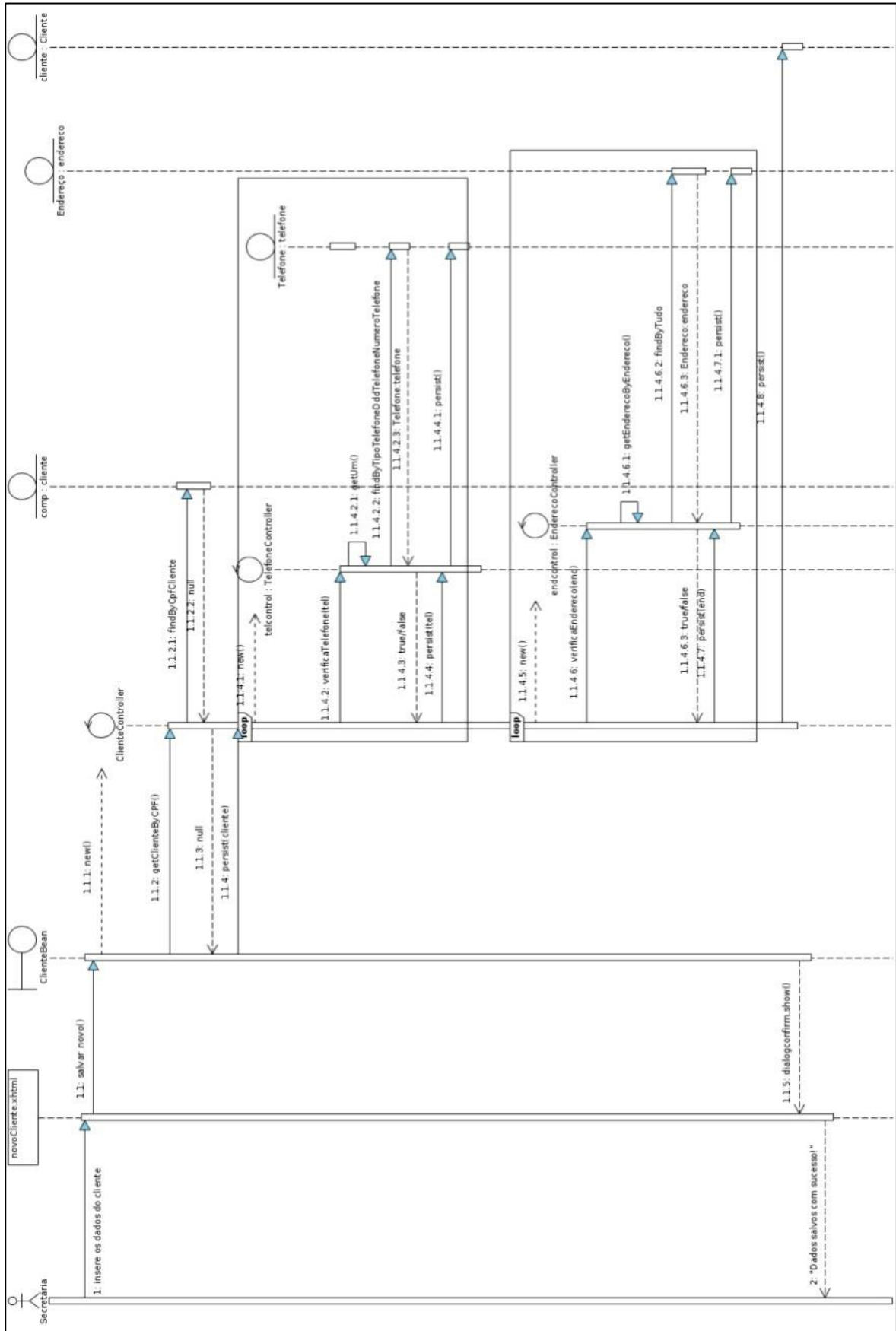


DIAGRAMA DE SEQUENCIA- CADASTRAR CLIENTE ALTERNATIVO 1

OCORRÊNCIA - Cliente já cadastrado.

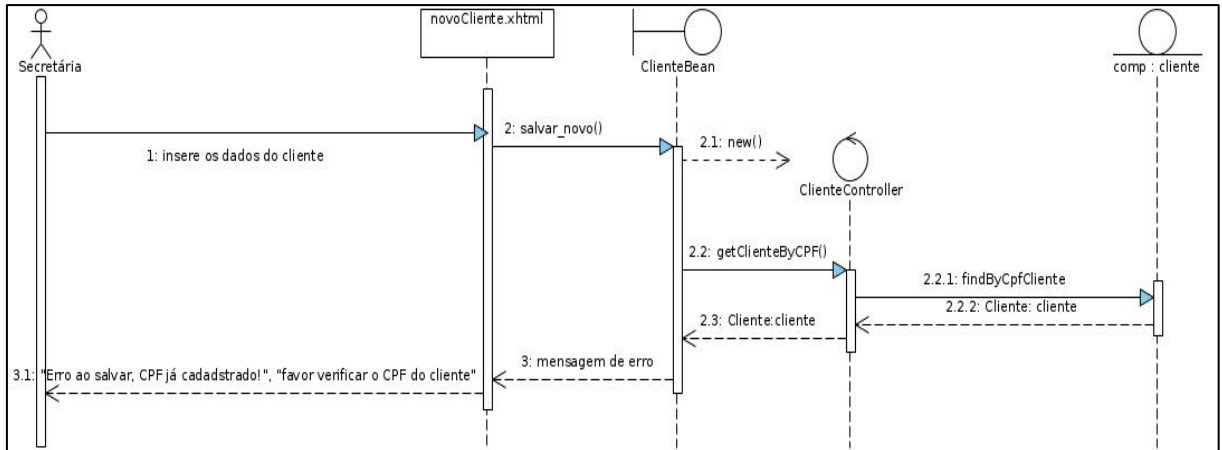


DIAGRAMA DE SEQUENCIA- CADASTRAR CLIENTE ALTERNATIVO 2

OCORRÊNCIA – erro ao persistir no banco de dados um telefone do cliente.

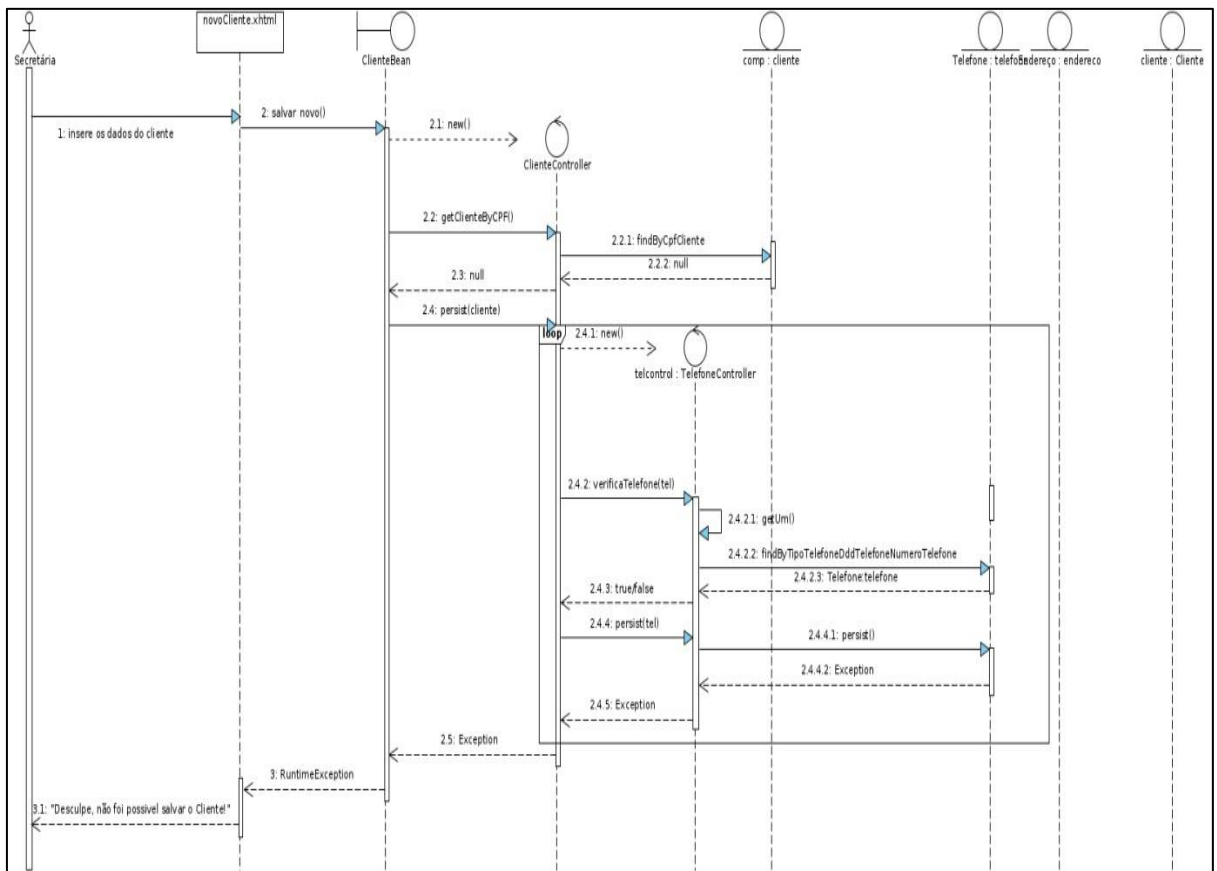


DIAGRAMA DE SEQUENCIA- CADASTRAR CLIENTE ALTERNATIVO 3

OCORRÊNCIA – erro ao persistir no banco de dados um endereço do cliente.

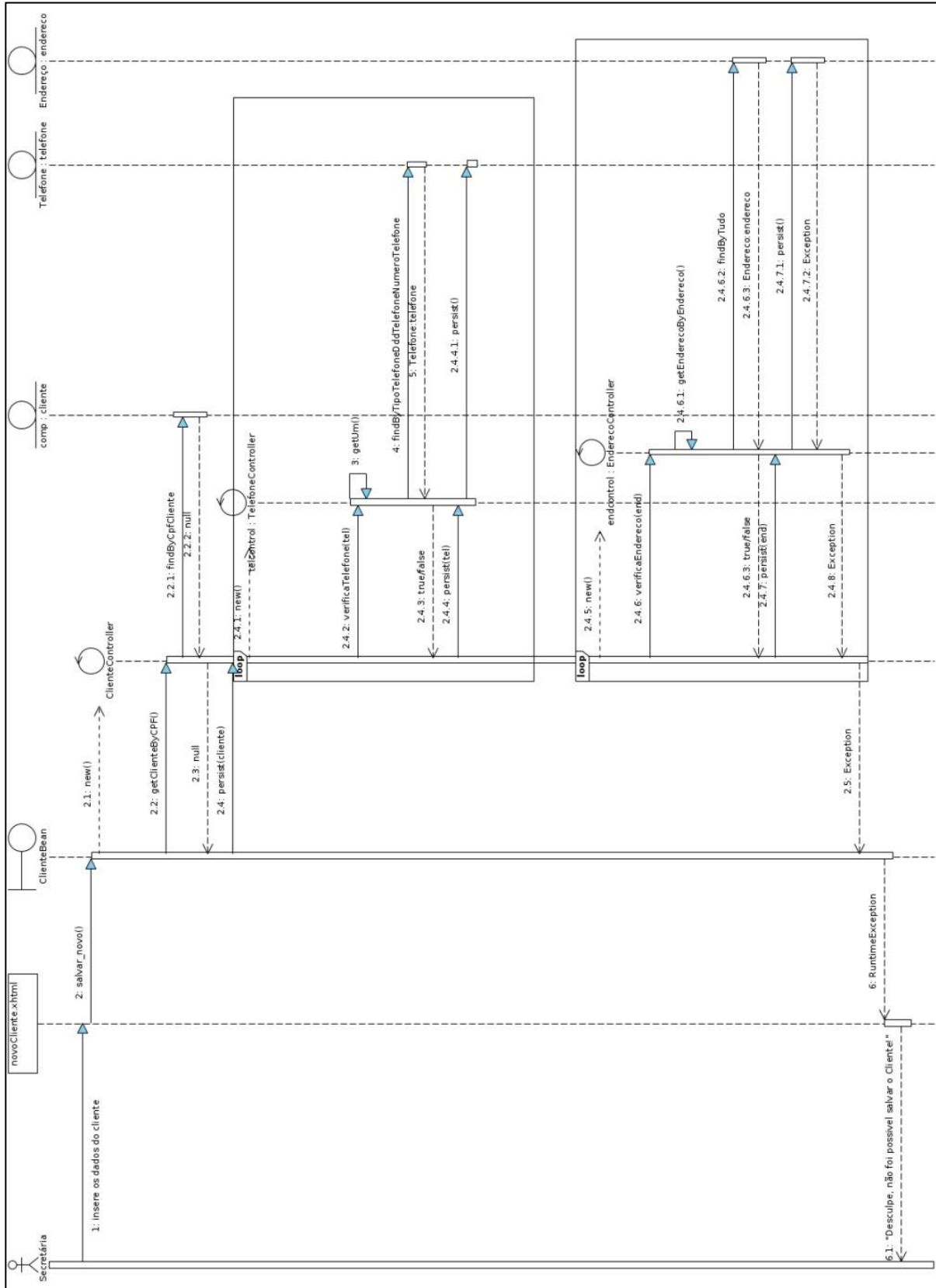


DIAGRAMA DE SEQUENCIA- CADASTRAR CLIENTE ALTERNATIVO 4

OCORRÊNCIA – erro ao persistir o cliente.

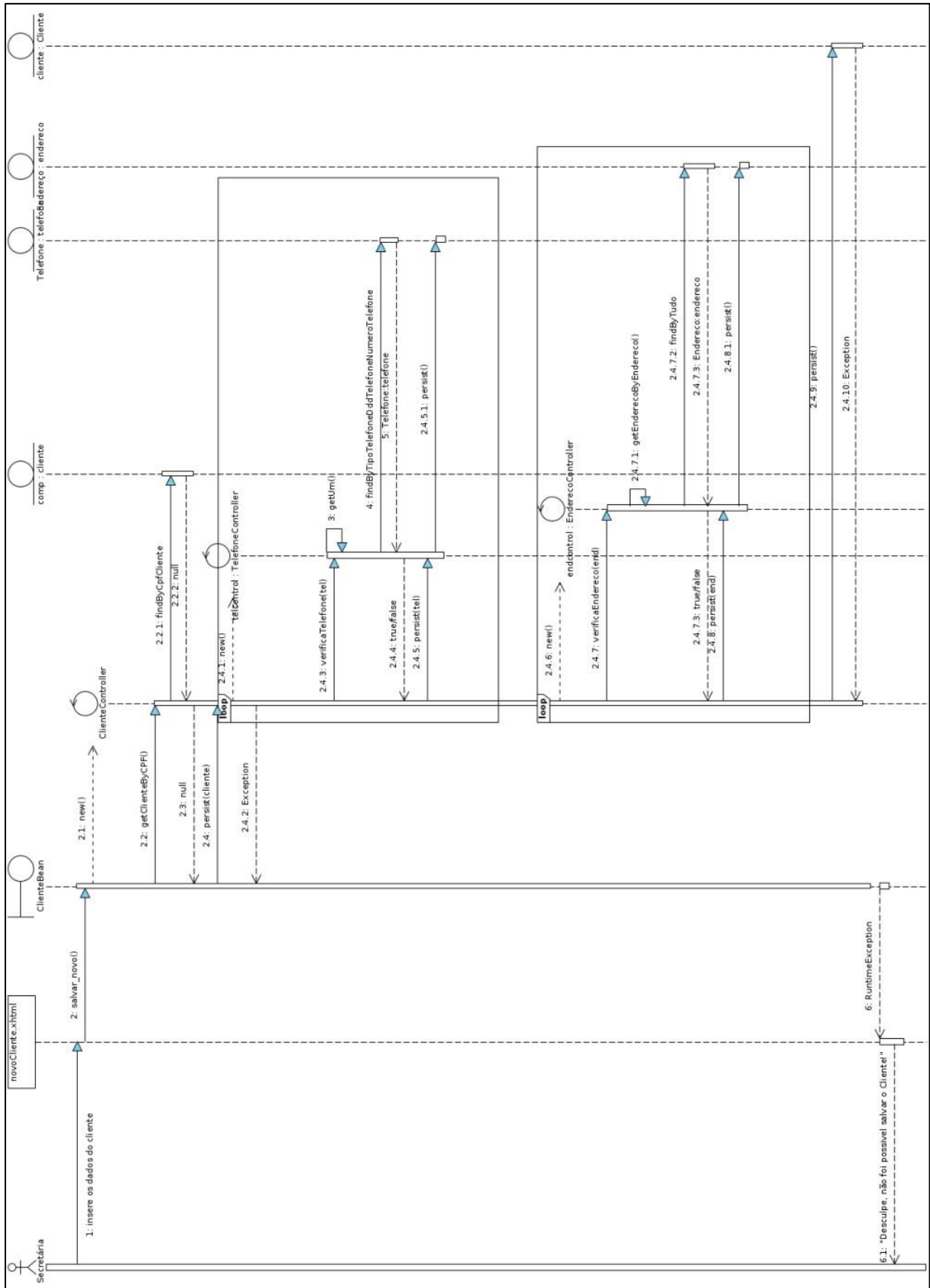


DIAGRAMA DE SEQUENCIA- CADASTRAR PACIENTE

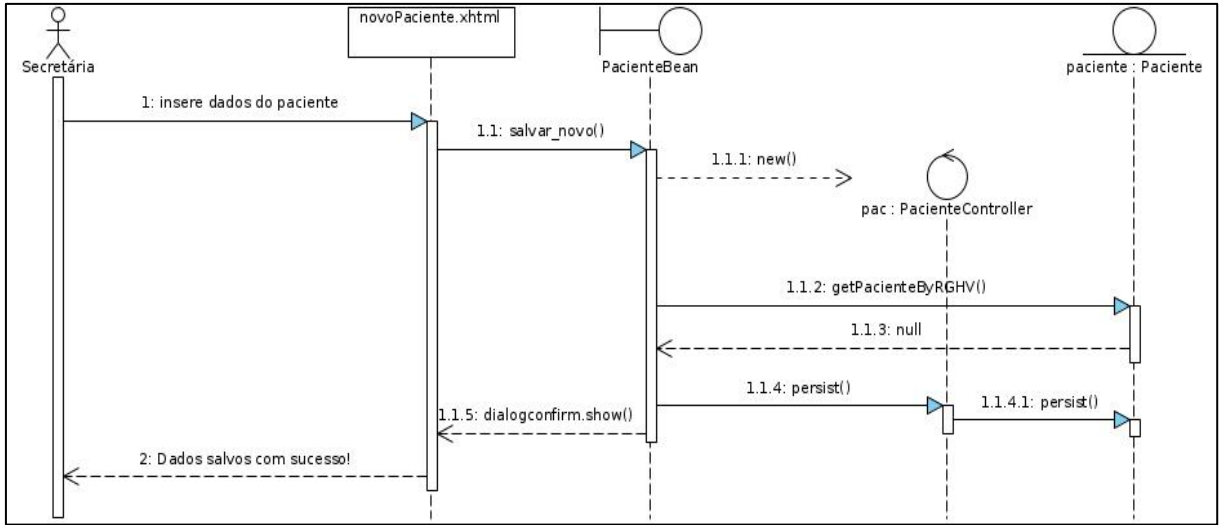


DIAGRAMA DE SEQUENCIA- CADASTRAR PACIENTE ALTERNATIVO 1

OCORRÊNCIA – paciente já cadastrado.

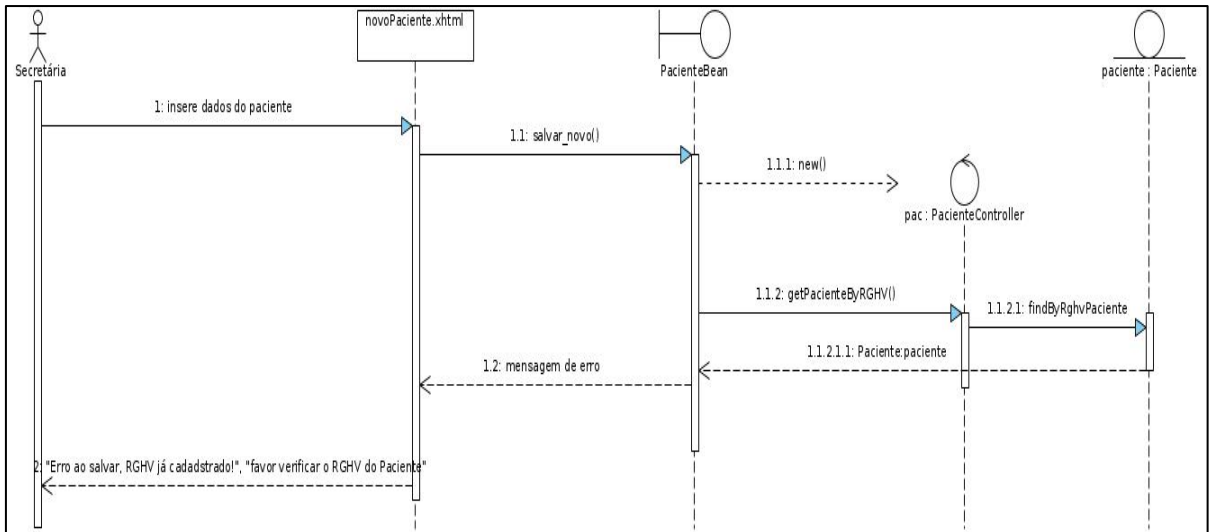


DIAGRAMA DE SEQUENCIA- CADASTRAR PACIENTE ALTERNATIVO 2

OCORRÊNCIA – erro ao persistir o paciente.

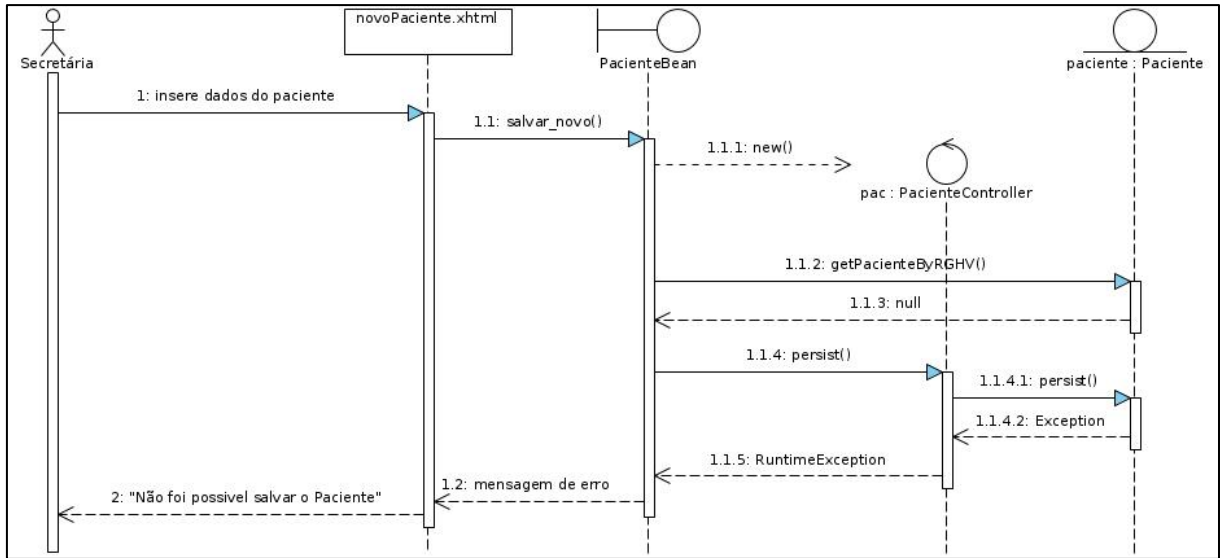


DIAGRAMA DE SEQUENCIA- CADASTRAR CONSULTA

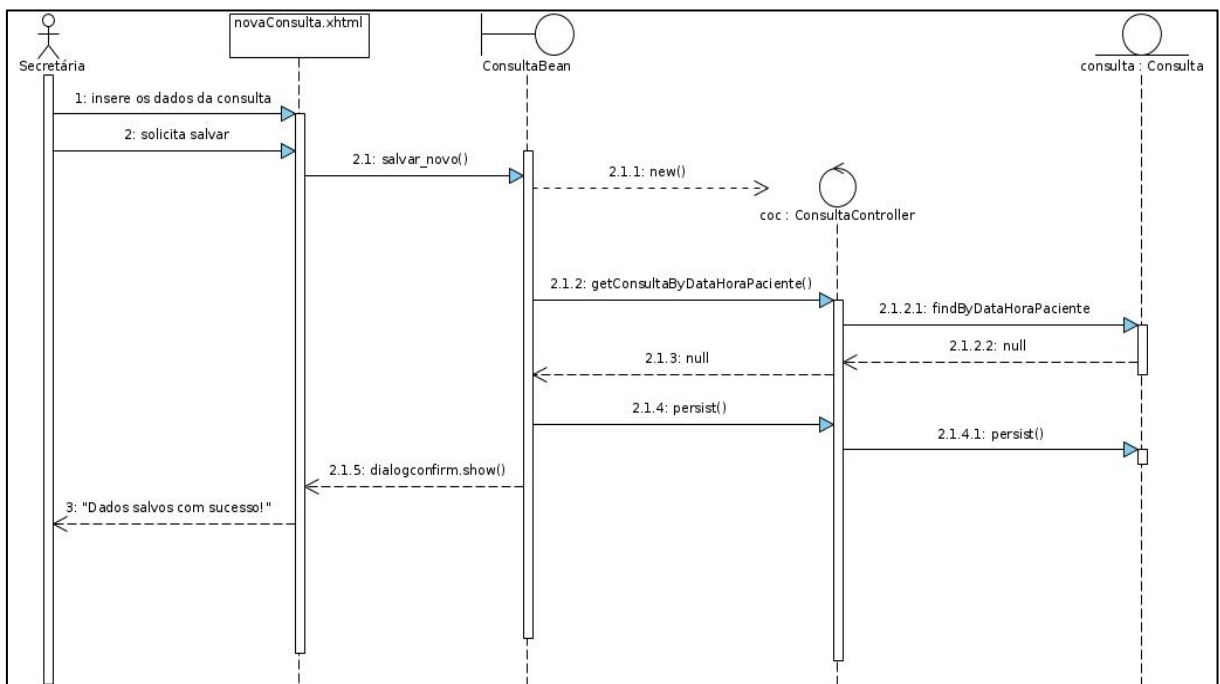


DIAGRAMA DE SEQUENCIA- CADASTRAR CONSULTA ALTERNATIVO 1

OCORRÊNCIA – consulta já cadastrada.

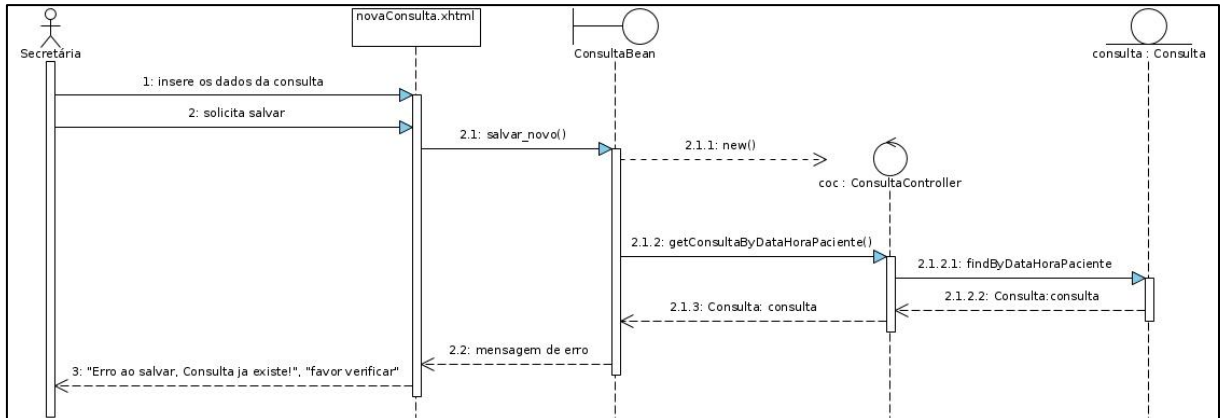


DIAGRAMA DE SEQUENCIA- CADASTRAR CONSULTA ALTERNATIVO 2

OCORRÊNCIA – erro ao persistir a consulta.

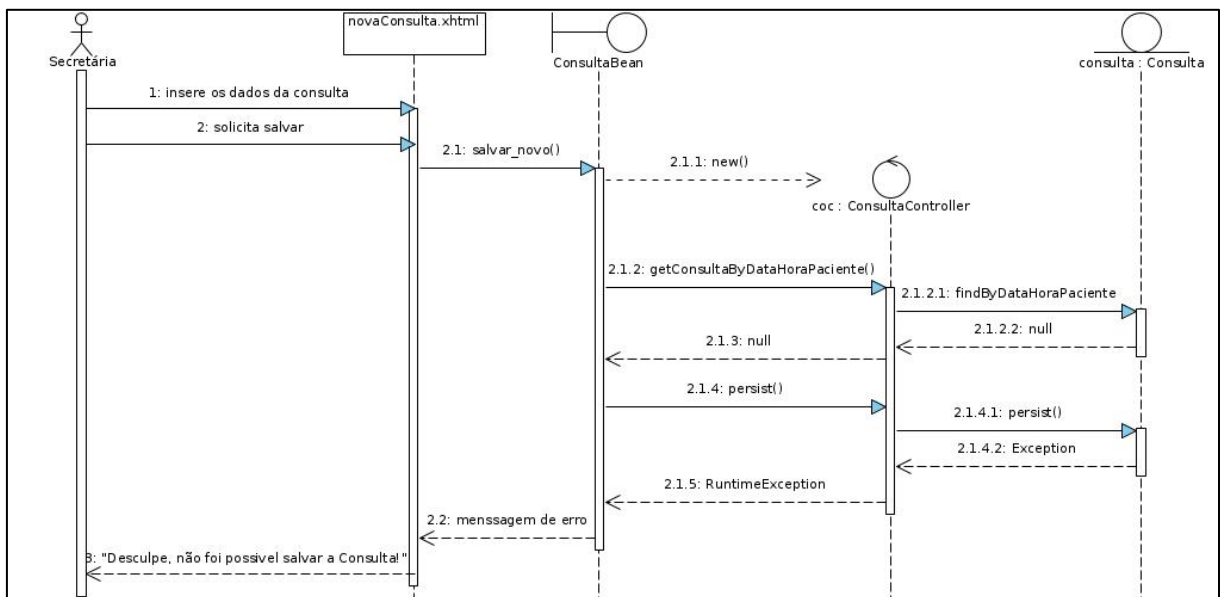


DIAGRAMA DE SEQUENCIA- CADASTRAR PROCEDIMENTO

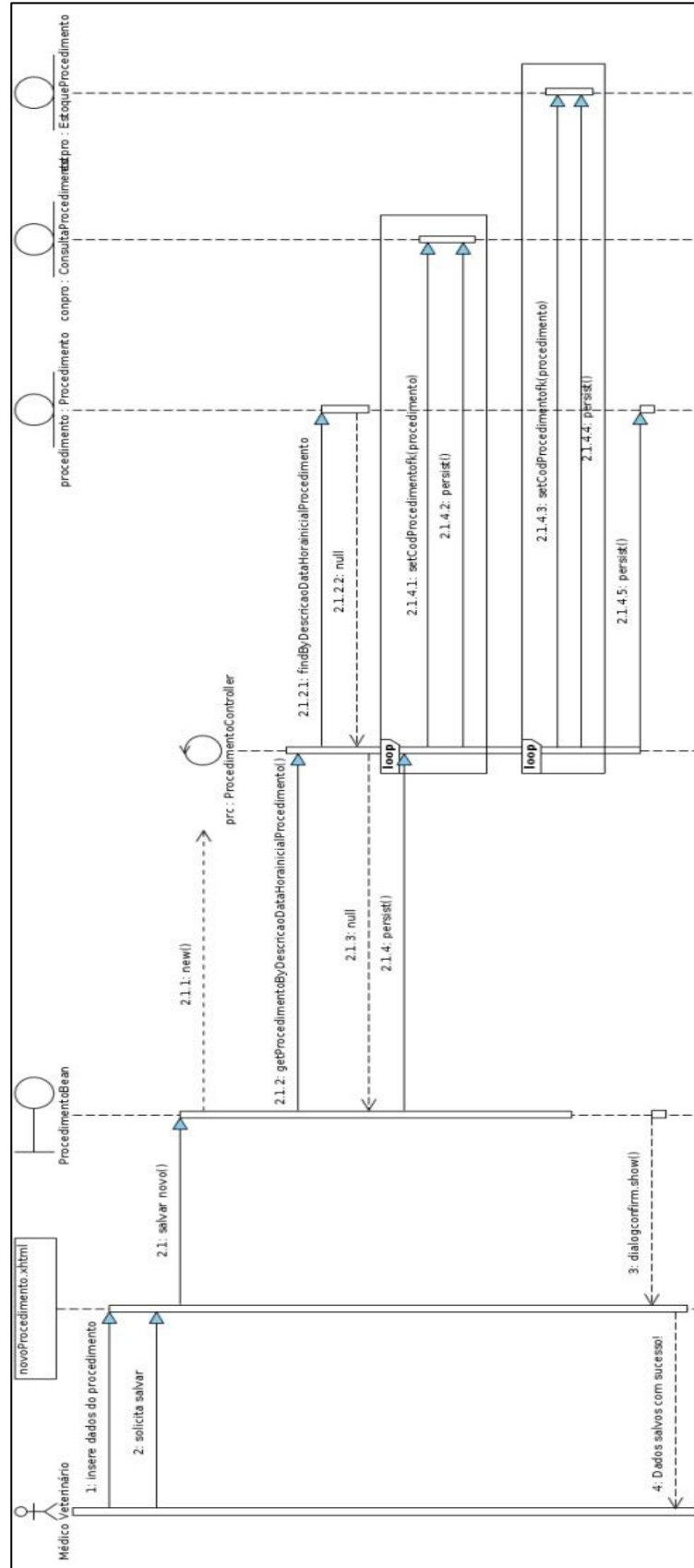


DIAGRAMA DE SEQUENCIA- CADASTRAR PROCEDIMENTO ALTERNATIVO 1

OCORRÊNCIA – procedimento já cadastrado.

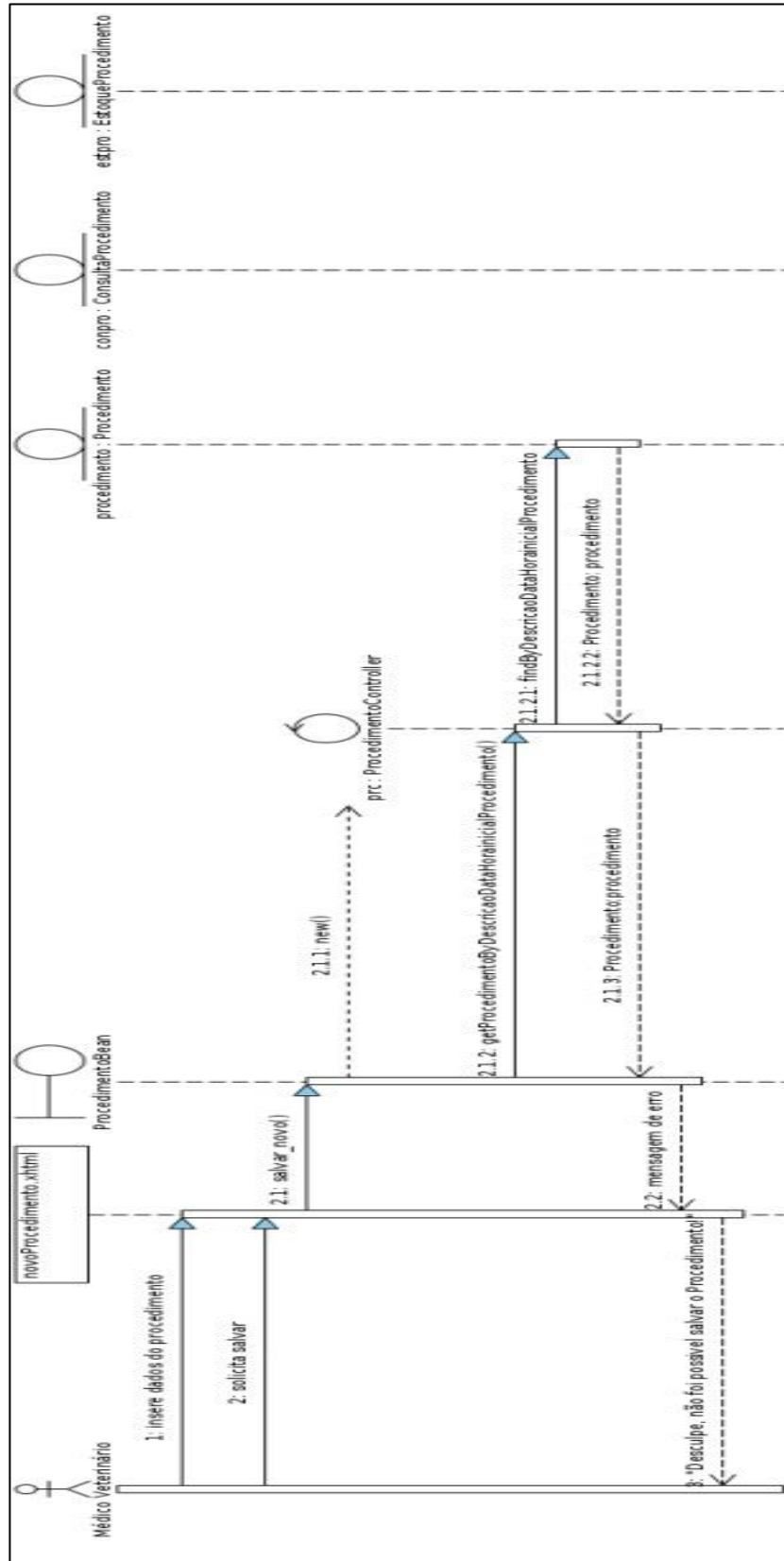


DIAGRAMA DE SEQUENCIA- CADASTRAR PROCEDIMENTO ALTERNATIVO 2

OCORRÊNCIA – erro ao persistir uma consulta pertencente a um procedimento.

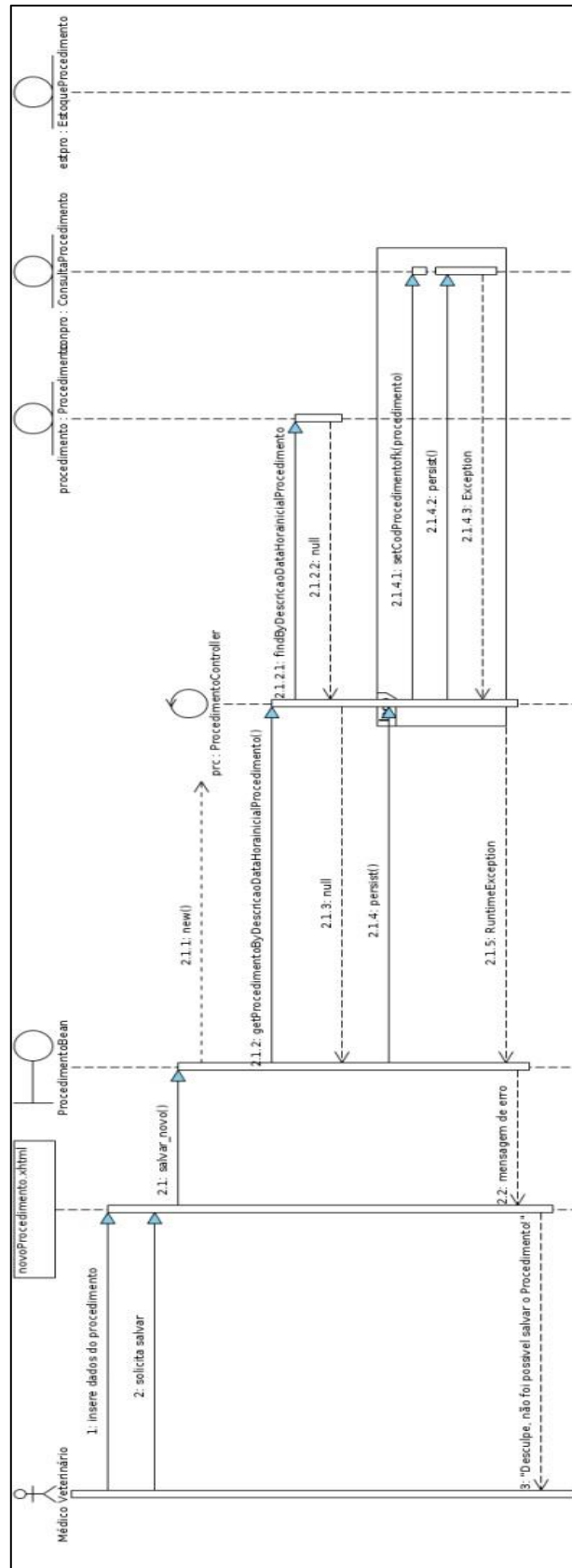


DIAGRAMA DE SEQUENCIA- CADASTRAR PROCEDIMENTO ALTERNATIVO 3

OCORRÊNCIA – erro ao persistir um material utilizado no procedimento.

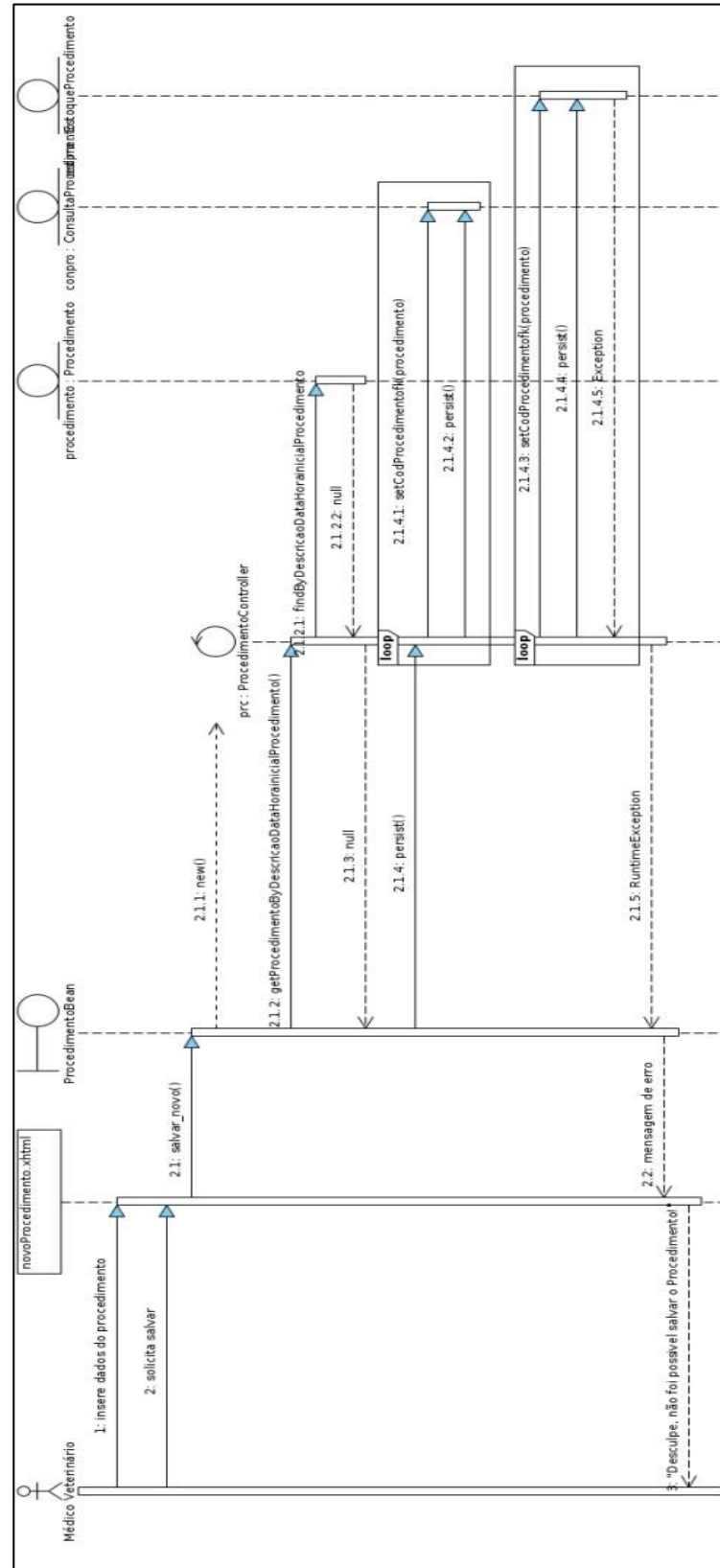


DIAGRAMA DE SEQUENCIA- CADASTRAR PROCEDIMENTO ALTERNATIVO 4

OCORRÊNCIA – erro ao persistir o procedimento.

