



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

**REGISTRO ELETRÔNICO DE SAÚDE BASEADO EM
SOFTWARE LIVRE E INTEGRADO A SISTEMAS DE
APOIO AO DIAGNÓSTICO**

IGOR RAFAEL SILVA VALENTE

FORTALEZA – CEARÁ
2010



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

DISSERTAÇÃO DE MESTRADO
IGOR RAFAEL SILVA VALENTE

**REGISTRO ELETRÔNICO DE SAÚDE BASEADO EM
SOFTWARE LIVRE E INTEGRADO A SISTEMAS DE
APOIO AO DIAGNÓSTICO**

ORIENTADOR:

PROF. DR. PAULO CÉSAR CORTEZ

CO-ORIENTADOR:

PROF. DR. JOSÉ MARQUES SOARES

*Dissertação de Mestrado apresentada
à Coordenação do Curso de
Pós-graduação em Engenharia de
Teleinformática da Universidade
Federal do Ceará, como parte
dos requisitos para obtenção do
grau de Mestre em Engenharia de
Teleinformática.*

FORTALEZA – CEARÁ

2010

IGOR RAFAEL SILVA VALENTE

**REGISTRO ELETRÔNICO DE SAÚDE BASEADO EM SOFTWARE LIVRE E
INTEGRADO A SISTEMAS DE APOIO AO DIAGNÓSTICO**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia de Teleinformática e aprovada em sua forma final pelo Programa de Pós-graduação em Engenharia de Teleinformática da Universidade Federal do Ceará.

Banca Examinadora:

Prof. Dr. Paulo César Cortez
Orientador

Prof. Dr. José Marques Soares
Co-orientador

Prof. Dr. Giovanni Cordeiro Barroso

Prof. Dr. Luis Roberto de Oliveira

Prof. Dr. Ivan Torres Pisa

Fortaleza, 26 de maio de 2010.

Resumo

A gestão e a utilização da informação e do conhecimento constituem o cerne das atividades profissionais e do processo decisório na prática da saúde. Os registros de saúde, dentre os quais o mais importante é o prontuário do paciente, até recentemente eram representados por documentos em papel mantidos em uma variedade de formatos, conteúdos e locais distintos. A impossibilidade de acessar e integrar dados de pacientes individuais ou de grupos de pacientes registrados em documentos manuais resulta em uma visão fragmentada da evolução dos problemas de saúde de uma comunidade. O Registro Eletrônico de Saúde é a solução para grande parte dos problemas existentes neste cenário. Este trabalho tem como objetivo desenvolver um Registro Eletrônico de Saúde baseado na WEB e software livre, garantindo aspectos como integridade dos dados, padronização, integração com sistemas de apoio ao diagnóstico, inovação e facilidade de uso.

Palavras-chave: registro eletrônico de saúde, prontuário eletrônico do paciente e informática em saúde.

Abstract

The management and use of information and knowledge are the core of professional activities and decision-making in health practice. Health records, among which the most important is the patient's chart until recently were represented by paper documents kept in a variety of formats, content and locations. The inability to access and integrate data from individual patients or groups of patients enrolled in manual documents results in a fragmented view of the evolution of a community's health problems. The Electronic Health Record is the solution to many problems in this scenario. This project aims to develop an Electronic Health Record, web and free software based, ensuring data integrity, standardization, diagnostic support integration, innovation and easy usage.

Keywords: computer-based patient record, electronic medical record systems, electronic patient record, electronic health record and health informatics.

*Dedico este trabalho aos meus pais
José Oliveira Valente e Maria de Fátima Silva Valente
pelo exemplo de amor, honestidade e fé.*

Agradecimentos

A Deus, acima de tudo.

Aos meus amados familiares Oliveira (pai), Fátima (mãe) e Cristiane (irmã) pelo amor, apoio, paciência e dedicação investidos durante toda a minha vida.

À minha amada Natália pelo carinho, amor, compreensão e incentivo para que eu pudesse concluir esta importante etapa da minha vida.

Aos professores e amigos Paulo César Cortez e José Marques Soares pelos ensinamentos, incentivo, confiança e dedicação depositados ao longo deste trabalho.

Ao amigo Daniel Colaço, por possibilitar o início desta jornada.

Aos colegas do Grupo de Interação Homem-Computador (GIHM) e do Grupo de Biomédica, pelas valorosas discussões e apoio durante a realização deste trabalho.

Ao amigo Rodrigo Costa pelas valiosas dicas da ferramenta L^AT_EX.

Ao colega Pedro Pedrosa pela disponibilização do código-fonte utilizado na ferramenta de apoio ao diagnóstico.

Aos professores e funcionários da Universidade Federal do Ceará.

À CAPES pelo suporte financeiro.

O único lugar onde o sucesso vem antes do trabalho é no dicionário.

Albert Einstein.

Sumário

Lista de Figuras	xii
Lista de Tabelas	xiii
Lista de Siglas	xiii
1 Introdução	1
1.1 O Prontuário do Paciente	1
1.2 Histórico	3
1.3 O Prontuário em Papel	6
1.4 A Informática em Saúde e o Prontuário Eletrônico do Paciente	8
1.4.1 Definições do PEP	10
1.4.2 Denominações do PEP	11
1.4.3 Níveis evolutivos do Registro Eletrônico de Saúde	11
1.4.4 Atributos do Registro Eletrônico de Saúde	12
1.5 Desafios no desenvolvimento do Registro Eletrônico de Saúde	15
1.6 Padronização	18
1.6.1 Doenças: a Classificação Internacional de Doenças (CID)	20
1.6.2 Procedimentos: a CBHPM e a Tabela SUS	21
1.6.3 Medicamentos: a Classificação Anatômica-Terapêutica-Clínica	22
1.6.4 Materiais hospitalares: a Tabela SIMPRO	22
1.6.5 Imagens radiológicas: o padrão DICOM	23
1.6.6 Comunicação em saúde: o padrão <i>Health Level Seven</i> (HL-7)	23
1.6.7 Usuários: o Projeto Cartão Nacional de Saúde	24
1.7 Motivação	24
1.8 Trabalho proposto	25
1.8.1 Objetivos gerais	25
1.8.2 Objetivos específicos	26
1.9 Organização do trabalho	26
2 Projeto e implementação do RES	28
2.1 Introdução	28
2.1.1 Visão geral	28

2.1.2	Definição do modelo de processo	29
2.2	Levantamento e análise de requisitos	31
2.2.1	Identificação de requisitos	31
2.2.2	Refinamento de requisitos	31
2.2.3	Documento de requisitos	31
2.3	Projeto de sistema	39
2.3.1	Plano de projeto	39
2.3.2	Modelagem do incremento 01 - autenticação e autorização	41
2.3.3	Modelagem do incremento 02 - prontuário do paciente	45
2.3.4	Arquitetura do RES proposto	49
2.4	Implementação do sistema	50
2.4.1	Sistema operacional	51
2.4.2	Linguagem de programação	51
2.4.3	Ferramenta de modelagem	52
2.4.4	<i>Framework</i>	52
2.4.5	Ambiente de desenvolvimento	53
2.4.6	Servidor de aplicação	53
2.4.7	Banco de dados	54
2.5	Verificação e validação	55
2.5.1	Verificação de requisitos	55
2.5.2	Validação de requisitos	55
2.6	Demonstração e testes	55
2.6.1	Página oficial	55
2.6.2	Versão de demonstração e teste	56
3	As funcionalidades e interfaces do RES	57
3.1	Introdução	57
3.1.1	Visão geral	57
3.1.2	Internacionalização	58
3.2	Autenticação e autorização	59
3.2.1	Interfaces diferenciadas	59
3.2.2	Permissões	61
3.2.3	Categorias	61
3.2.4	Grupos	62
3.3	Gerenciamento de dados	63
3.3.1	Pessoas	63
3.3.2	Especialidades médicas	64
3.3.3	Médicos	64
3.3.4	Queixas gerais	66
3.3.5	Eventos	69
3.3.6	Vacinas	72
3.3.7	Configurações	73
3.4	Fluxo de atendimento	74
3.4.1	Verificação de usuário	74
3.4.2	Abertura da consulta	76

3.4.3	Fila de espera	77
3.4.4	Realização da consulta	77
3.4.5	Solicitações de eventos	78
3.4.6	Resultados de solicitações de eventos	78
3.4.7	Apoio ao diagnóstico	81
3.4.8	Edição de imagens médicas	81
3.4.9	Imunizações	84
3.4.10	Diagnósticos	84
3.4.11	Prescrição médica	85
3.5	Acesso do Paciente	85
3.5.1	Alteração de senha	87
3.5.2	Atualização de dados pessoais	87
3.5.3	Visualização de dados clínicos	87
3.5.4	Central de mensagens	87
4	Contribuições, conclusões e trabalhos futuros	92
4.1	Contribuições	92
4.1.1	Autenticação e autorização	92
4.1.2	Gerenciamento de dados	93
4.1.3	Acesso do paciente	93
4.1.4	Internacionalização	93
4.1.5	Leitura de arquivos no padrão DICOM	94
4.1.6	Apoio ao diagnóstico	94
4.1.7	Editor de imagens médicas	94
4.1.8	Prescrições médicas	95
4.1.9	Componente dinâmico de inclusão de diagnósticos	95
4.1.10	Código-fonte aberto	95
4.1.11	Componentes para auxiliar a usabilidade	96
4.1.12	Engenharia de software e modelagem de sistemas	96
4.2	Conclusões	96
4.3	Trabalhos futuros	97
Apêndice A	Engenharia de Software e Modelagem de Sistemas	99
A.1	Engenharia de Software	99
A.1.1	O que é Engenharia de Software?	100
A.1.2	Processo de software	101
A.1.3	Modelos de processo de software	102
A.1.4	Requisitos de software	110
A.2	Modelagem de Sistemas	113
A.2.1	O que é Modelagem de Sistemas?	114
A.2.2	Princípios da Modelagem de Sistemas	114
A.2.3	Tipos de modelos	116
A.2.4	O que é UML?	116
A.2.5	Como chegamos à UML?	117
A.2.6	Os diagramas	117
A.2.7	Diagrama de casos de uso	118

A.2.8	Diagrama de classes	119
A.2.9	Diagrama de atividade	119
Apêndice B	Modelagem de Casos de Uso	121
B.1	Incremento 01 - autenticação e autorização	121
B.1.1	Identificação e especificação dos casos de uso	121
B.2	Incremento 02 - prontuário do paciente	130
B.2.1	Identificação e especificação dos casos de uso	130
Referências Bibliográficas		153

Lista de Figuras

1.1	Estrutura do POMR	5
1.2	Enquete sobre as desvantagens do prontuário em papel	7
1.3	Enquete sobre as vantagens do registro eletrônico de saúde	14
2.1	Modelo incremental	30
2.2	Cronograma de desenvolvimento do RES proposto	41
2.3	Diagrama de atores do incremento 01	42
2.4	Diagrama de casos de uso do incremento 01	42
2.5	Diagrama de classes de entidades persistentes do incremento 01	43
2.6	Diagrama de atividade da autenticação	44
2.7	Diagrama de atividade da alteração de senha	45
2.8	Diagrama de casos de uso do administrador	46
2.9	Diagrama de casos de uso do recepcionista	46
2.10	Diagrama de casos de uso do médico	47
2.11	Diagrama de casos de uso do profissional de saúde	47
2.12	Diagrama de casos de uso do paciente e do usuário	47
2.13	Diagrama de classes de entidades persistentes do incremento 02	48
2.14	Diagrama de atividades do plugin de apoio ao diagnóstico	49
2.15	Arquitetura do RES proposto	50
3.1	Página de autenticação no idioma português brasileiro	58
3.2	Página de autenticação no idioma inglês	58
3.3	Página inicial do paciente	60
3.4	Página inicial do médico	60
3.5	Página inicial do administrador	60
3.6	Página de pesquisa de permissões	61
3.7	Página de edição de permissões	61
3.8	Página de pesquisa de categorias	62
3.9	Página de edição de categorias	62
3.10	Página de pesquisa de grupos	63
3.11	Página de edição de grupos	63
3.12	Página de pesquisa de pessoas	64
3.13	Página de edição de pessoas	65

3.14	Página de pesquisa de especialidades médicas	66
3.15	Página de edição de especialidades médicas	66
3.16	Página de pesquisa de médicos	67
3.17	Página de edição de médicos	67
3.18	Página de pesquisa de categorias de queixas gerais	67
3.19	Página de edição de categorias de queixas gerais	68
3.20	Página de pesquisa de queixas gerais	68
3.21	Página de edição de queixas gerais	68
3.22	Página de pesquisa de tipos de eventos	69
3.23	Página de edição de tipos de eventos	69
3.24	Página de pesquisa de categorias de eventos	70
3.25	Página de edição de categorias de eventos	70
3.26	Página de pesquisa de eventos	71
3.27	Página de edição de eventos	71
3.28	Página de pesquisa de vacinas	72
3.29	Página de edição de vacinas	72
3.30	Página de pesquisa de configurações	73
3.31	Página de edição de configurações	73
3.32	Página de autenticação do recepcionista	74
3.33	Página de pesquisa de pessoas	74
3.34	Página de cadastro de paciente	75
3.35	Página de edição de paciente - adicionar consulta	76
3.36	Página de cadastro de consulta	76
3.37	Página de fila de espera	77
3.38	Página inicial do médico com consultas em aberto	77
3.39	Página de edição da consulta	78
3.40	Página de cadastro de solicitação de evento	79
3.41	Página de pesquisa de solicitações de evento	79
3.42	Página de edição da solicitação de evento em aberto	80
3.43	Página de edição da solicitação de evento em atendimento	80
3.44	Página de resultado da solicitação de evento	81
3.45	Página de solicitação de auxílio ao diagnóstico	82
3.46	Página de processamento da solicitação de apoio ao diagnóstico	82
3.47	Página de resultado da solicitação de apoio ao diagnóstico	83
3.48	Página de edição de imagens	84
3.49	Página de solicitação de imunização	85
3.50	Página de inclusão de diagnóstico	85
3.51	Página de inclusão de prescrição médica	86
3.52	Página de impressão de prescrição médica	86
3.53	Página de alteração de senha do paciente	87
3.54	Página de edição de dados pessoais	88
3.55	Página de visualização de dados clínicos - consultas	89
3.56	Página de visualização de dados clínicos - solicitações de eventos	89
3.57	Página de visualização de dados clínicos - imunizações	89
3.58	Página de visualização de dados clínicos - diagnósticos	89

3.59	Página de inclusão de mensagem do paciente	90
3.60	Página inicial do médico com alerta de mensagem não lida	90
3.61	Página de resposta de mensagem do médico	90
A.1	Modelo cascata	104
A.2	Modelo evolucionário	104
A.3	Prototipagem	106
A.4	Modelo espiral	107
A.5	Diagramas da UML 2.0	118
A.6	Exemplo de diagrama de caso de uso	119
A.7	Exemplo de diagrama de classes	120
A.8	Exemplo de diagrama de atividades	120

Lista de Tabelas

1.1	Evolução temporal da CID	20
1.2	Estrutura da CID	21
1.3	Exemplos da ATC	22
1.4	Exemplos da Tabela SIMPRO	23
2.1	Prioridades dos requisitos	32
2.2	Requisitos do incremento 01	33
2.3	Requisitos do incremento 02	36
2.4	Riscos gerenciados durante o desenvolvimento do trabalho	40
A.1	Atributos essenciais de um bom software	102
A.2	Etapas do modelo cascata	103
A.3	Requisitos de usuário e de sistema	111
B.1	Especificação dos casos de uso do incremento 01	122
B.2	Especificação dos casos de uso do incremento 02	130

Lista de Siglas

ABNT	Associação Brasileira de Normas Técnicas
AMB	Associação Médica Brasileira
ANSI-HISB	<i>American National Standards Institute - Healthcare Informatics Standards Board</i>
ATC	Classificação Anatômica-Terapêutica-Clínica
CBHPM	Classificação Brasileira Hierarquizada de Procedimentos Médicos
CEN	Comitê Europeu de Normatização
CFM	Conselho Federal de Medicina
CID	Classificação Internacional de Doenças
DICOM	<i>Digital Imaging and Communication Standards Committee</i>
FUST	Fundo de Universalização de Serviços de Telecomunicações
HL-7	<i>Health Level Seven</i>
ICM	Instituto de Ciências Médicas
IOM	<i>Institute of Medicine</i>
ISO	<i>International Standards Organization</i>
JAAS	<i>Java Authentication and Authorization Service</i>
LDAP	<i>Lightweight Directory Access Protocol</i>
OMG	<i>Object Management Group</i>
OMT	<i>Object Modeling Technique</i>
OOSE	<i>Object-Oriented Software Engineering</i>
PEP	Prontuário Eletrônico do Paciente

POMR	Registro Médico Orientado ao Problema
RES	Registro Eletrônico de Saúde
RF	Requisito Funcional
RNF	Requisito Não-funcional
SBIS	Sociedade Brasileira de Informática em Saúde
SBIS-CFM	Parceria entre SBIS e CFM
SGBD	Sistema Gerenciador de Banco de Dados
SQL	<i>Structured Query Language</i>
S-RES	Sistema de Registro Eletrônico de Saúde
SUS	Sistema Único de Saúde
UF	Unidade da Federação
UFC	Universidade Federal do Ceará
UML	<i>Unified Modeling Language</i>
WEB	<i>World Wide Web</i>

Capítulo 1

Introdução

1.1 O Prontuário do Paciente

O registro das informações dos pacientes faz parte do cotidiano de todos os profissionais que trabalham na área da saúde. O agrupamento e arquivamento dessas informações, de forma que possam ser recuperadas posteriormente, é chamado de prontuário do paciente, prontuário médico, ou ainda, registro médico (COSTA, 2001).

O prontuário do paciente é um elemento crucial no atendimento à saúde dos indivíduos, devendo reunir a informação necessária para permitir a continuidade do atendimento prestado. Tendo sido desenvolvido por médicos e enfermeiros para garantir que se lembrassem de forma sistemática dos fatos e eventos clínicos sobre cada indivíduo, permite também que todos os demais profissionais envolvidos no processo de atenção à saúde possam ter as mesmas informações. Desta forma, na instituição em que o paciente está recebendo cuidados, o prontuário do paciente representa o mais importante meio de comunicação entre os membros da equipe de saúde responsável pelo atendimento (SLEE; SLEE; SCHMIDT, 2000).

Adicionalmente, a análise conjunta dos prontuários é capaz de fornecer informações agregadas ou desagregadas sobre pessoas atendidas, tratamentos realizados, formas terapêuticas com resultados positivos, como os pacientes respondem e quanto custa cada forma de tratamento por procedimento ou cumulativamente para todos os procedimentos de um paciente. Estas informações, quando agregadas e sistematizadas, permitem caracterizar o nível de saúde populacional de uma região e viabilizam a construção de modelos e políticas de

atendimento e gestão das organizações de saúde (MASSAD; MARIN; NETO, 2003).

Em termos mais gerais, pode-se afirmar que o sistema de saúde de um país é estabelecido graças ao que se tem documentado em um prontuário, uma vez que dele são extraídas as informações sobre a saúde dos indivíduos que formam uma comunidade e uma nação.

Atualmente, entende-se que o prontuário do paciente tem como funções (MASSAD; MARIN; NETO, 2003):

- apoiar o processo de atenção à saúde, servindo de fonte de informação clínica e administrativa para tomada de decisão e meio de comunicação compartilhado entre todos os profissionais;
- ser o registro legal das ações médicas;
- apoiar a pesquisa através da realização de estudos clínicos, epidemiológicos, avaliação de qualidade, entre outros;
- promover o ensino e gerenciamento dos serviços, fornecendo dados para cobranças e reembolso, autorização das operadoras de planos de saúde, suporte para aspectos organizacionais e gerenciamento de custo.

Ao considerar as informações que devem estar presentes em um prontuário do paciente, é importante destacar que todo e qualquer atendimento em saúde pressupõe o envolvimento e a participação de múltiplos profissionais: médicos, enfermeiros, nutricionistas, psicólogos, fisioterapeutas, entre outros. Além disso, frequentemente as atividades de atendimento ao paciente ocorrem em locais distintos e as informações são obtidas de diferentes fontes, que utilizam diferentes padrões de dados. Tais dados precisam ser organizados de modo a produzir um apoio para a tomada de decisão sobre o tipo de tratamento ao qual o paciente deve ser submetido, orientando todo o processo de atendimento de um indivíduo ou de uma população.

Outro ponto a ser observado é que a informação clínica é muito heterogênea. Os dados referentes ao controle de sinais vitais precisam ser verificados, dependendo de cada caso, em intervalos muito próximos, sendo apresentados em planilhas e gráficos; os resultados de exames laboratoriais são disponibilizados em forma de tabelas; os exames de tomografia computadorizada, radiologia e ultrasonografia apresentam imagens como parte do prontuário do paciente; observações clínicas

podem estar presentes em intervalos regulares ou sob a forma de texto livre, sem qualquer estrutura ou formato; alguns dados de anamnese¹ são frequentemente registrados através de uma lista de checagem; o registro de medicação contém a listagem das prescrições médicas, a checagem da administração fornecida ao paciente pela enfermeira e a reação do paciente ao medicamento; as observações feitas por psicólogos geralmente são registradas em texto livre. Muitos outros exemplos poderiam ainda ser incluídos, confirmando a diversidade dos dados e informações que podem ser utilizadas para viabilizar o cuidado da saúde (MASSAD; MARIN; NETO, 2003).

Diante do exposto, é possível afirmar que manter registros longitudinais que englobem toda a vida dos indivíduos e criar bancos de dados de conhecimento clínico e administrativo que sirvam de base para montar os prontuários dos pacientes é um dos maiores desafios existentes na área de sistemas de informação. Vários países vêm perseguindo este objetivo há muito tempo e ainda não atingiram este estágio de desenvolvimento (NARDON; FURUIE; TACHINARDI, 2000).

Para facilitar a compreensão do estado atual de desenvolvimento do prontuário do paciente, faz-se necessário percorrer a história para entender as suas origens e, desta forma, compreender como se chegou ao prontuário do paciente dos dias atuais. A história e a evolução do prontuário do paciente são discutidos na seção a seguir.

1.2 Histórico

O registro das informações dos pacientes não é uma prática nova. No século 5 A.C. Hipócrates estimulou os médicos a fazerem seus primeiros registros escritos, utilizando-se de dois argumentos principais para convencê-los: refletir de forma exata o curso da doença e indicar suas possíveis causas. Até o início do século XIX, os médicos basearam suas observações e anotações no que ouviam, sentiam e viam. Suas observações eram registradas em papel e em ordem cronológica, estabelecendo assim o chamado Prontuário Orientado pelo Tempo, em uso desde então (BEMMEL; MUSEN, 1997).

Florence Nightingale (1820-1910), precursora da Enfermagem Moderna, já relatava que a documentação das informações relativas aos pacientes é de fundamental importância para a continuidade dos cuidados, principalmente no que

¹Entrevista realizada pelo médico ao seu paciente com a intenção de ser um ponto inicial no diagnóstico de uma doença.

se refere a assistência de enfermagem. É clássica a frase de Nightingale, quando observa a importância dos registros de saúde: “Na tentativa de chegar à verdade tenho buscado informações em todos os locais, mas em raras ocasiões tenho obtido registros hospitalares possíveis de serem usados para comparações. Estes registros poderiam nos mostrar como o dinheiro tem sido usado, o que de bom foi realmente feito com ele...”.

William Mayo, membro do grupo de colegas responsável por formar a Clínica Mayo em Minnesota, Estados Unidos, observou que a maioria dos médicos mantinham o registro de anotações das consultas de todos os pacientes em ordem cronológica e em um documento único. O conjunto de anotações trazia dificuldade para localizar informações específicas sobre um determinado paciente. Por este motivo, em 1907, a Clínica Mayo adota o registro individual das informações de cada paciente. Isto dá origem ao Prontuário Centrado no Paciente e Orientado pelo Tempo.

Em 1920, ainda na Clínica Mayo, houve um movimento para padronizar o conteúdo dos prontuários através da definição de um conjunto mínimo de dados que deveriam ser registrados. Este conjunto mínimo de dados criou a estrutura sistematizada de apresentação da informação médica que caracteriza o prontuário do paciente atualmente. Entretanto, apesar de todos os esforços de padronização, o prontuário ainda contém uma mistura de queixas, resultados de exames, considerações, planos terapêuticos e achados clínicos que muitas vezes são armazenados de forma desordenada, pois nem sempre é fácil obter uma informação clara sobre a evolução do paciente, principalmente nos casos cujo paciente sofre de mais de uma enfermidade.

No final de 1960, Lawrence Weed desenvolveu uma abordagem conhecida como Registro Médico Orientado ao Problema (POMR), que propôs uma nova organização das informações no prontuário. De acordo com esta abordagem, todas as anotações devem estar relacionadas a um problema específico de saúde, criando então uma lista de problemas numa árvore hierárquica na qual cada problema seria um ramo principal (WEED, 1968).

No POMR, a lista de problemas é o ponto central e atua como uma tabela de conteúdo do prontuário. Os problemas são classificados em ativos e passivos. Os ativos identificam problemas que requerem uma ação ou uma doença que ainda persiste sem cura no paciente. Os passivos não necessitam de nenhuma ação

imediate, podendo posteriormente tornarem-se inativos. A estrutura do POMR pode ser visualizada na Figura 1.1.

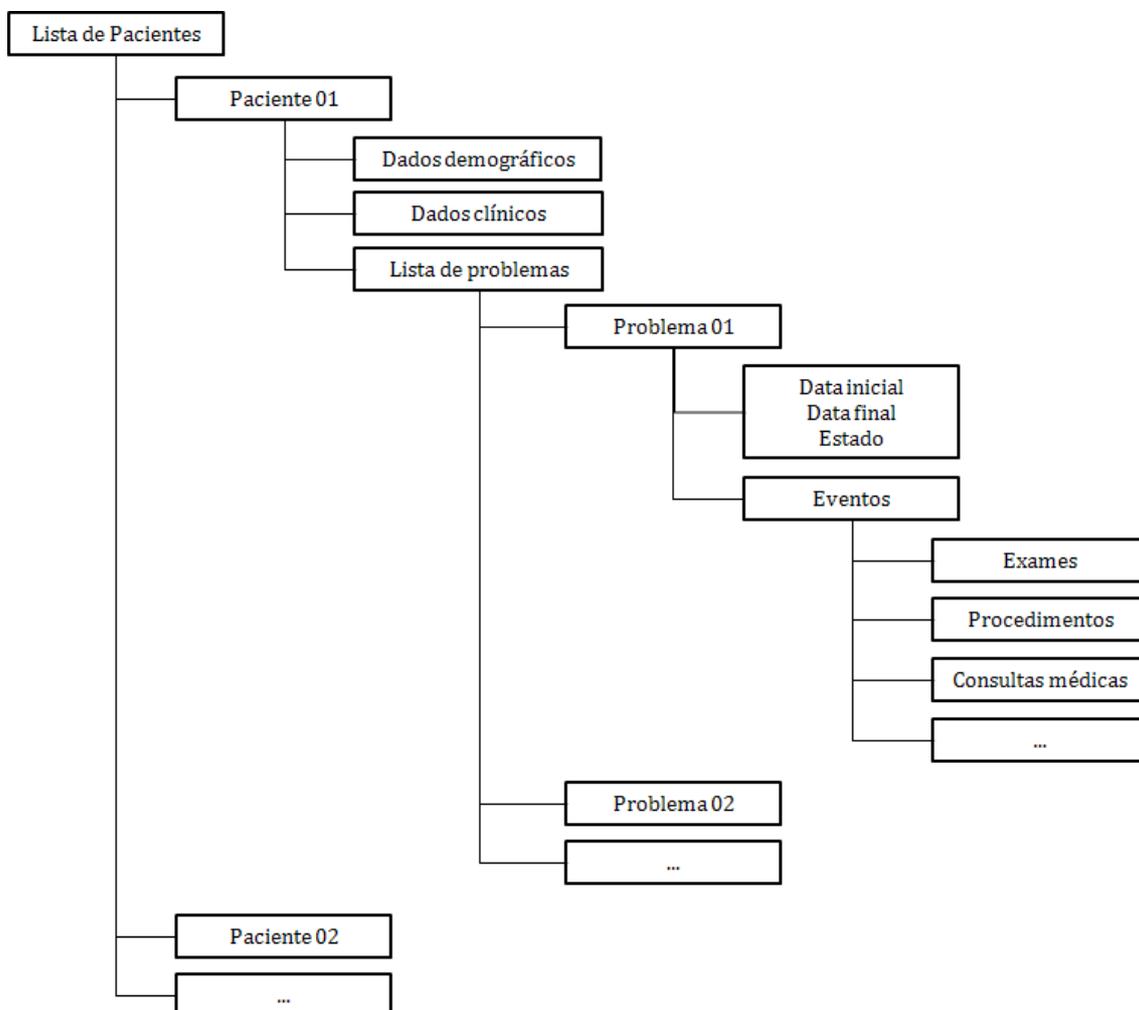


Figura 1.1: estrutura do POMR.

A estrutura do POMR foi muito utilizada em todo o mundo. Vários são os trabalhos que citam o uso da abordagem orientada ao problema como uma estrutura viável (HO *et al.*, 1999). Porém, com o crescente número de sistemas que utilizaram o POMR, as limitações tornaram-se evidentes. Conceitos como consulta médica, episódios de uma mesma doença e subproblemas não são contemplados pela estrutura clássica (SALMON; RAPPAPORT; BAINBRIDGE, 1996). Devido a esta estrutura, a entrada dos dados é lenta, o que constitui outro fator negativo. Isto tem dificultado uma penetração ainda maior do POMR na prática médica diária (MEYERS; MILLER; NAYEMI-RAD, 1998). Com isso, novas estruturas foram gradativamente aparecendo, sanando algumas deficiências da estrutura clássica.

Além das diversas estruturas de armazenamento criadas no decorrer da história, outro ponto merece destaque: o meio de armazenamento. Desde o primeiro registro clínico o papel tem sido utilizado, sendo considerado o principal meio de armazenamento durante muito tempo. Atualmente, mesmo com o desenvolvimento tecnológico dos sistemas de informação, o papel ainda é utilizado em muitas instituições. Várias considerações podem ser feitas sobre o papel como meio de armazenamento, de vantagens a desvantagens.

1.3 O Prontuário em Papel

Desde a época de Hipócrates, o papel tem sido utilizado como meio de armazenamento do prontuário dos pacientes. Esta abordagem possui várias limitações, tanto práticas como lógicas, sendo ineficiente para o armazenamento e organização de grande número de dados de tipos diferentes.

Algumas desvantagens podem ser facilmente percebidas (SABBATINI, 1982; BEMMEL; MUSEN, 1997):

- o prontuário pode estar apenas em um único lugar por vez;
- ilegibilidade;
- ambiguidade;
- perda frequente de informação;
- multiplicidade de pastas;
- dificuldade de pesquisa coletiva;
- falta de padronização;
- dificuldade de acesso;
- fragilidade do papel.

Estas desvantagens tornam-se significativas quando percebe-se que a impossibilidade de acessar e integrar dados de pacientes individuais ou de grupos de pacientes registrados em documentos manuais resulta em uma visão fragmentada da evolução dos problemas de saúde, além da impossibilidade de recuperar a informação agregada dos prontuários de uma comunidade. O resultado deste cenário é a

inexistência de políticas governamentais de saúde direcionadas aos problemas de cada comunidade devido à falta de informações adequadas.

Em contrapartida, podem também ser destacadas algumas vantagens (BEMMEL; MUSEN, 1997):

- algumas pessoas consideram o prontuário em papel mais fácil de transportar;
- maior liberdade para registro das informações;
- facilidade no manuseio;
- não requer treinamento especial;
- nunca fica fora de operação, como os sistemas computacionais.

Por estas razões, é possível considerar que um prontuário em papel bem organizado pode ser melhor que um prontuário informatizado mal estruturado.

Em uma enquete realizada na Maternidade Escola Januário Cicco (Natal, Rio Grande do Norte), os funcionários foram questionados sobre as desvantagens do papel como meio de armazenamento das informações clínicas, na qual a perda de informações e a ilegibilidade foram consideradas as principais falhas (COSTA; MARQUES, 1999). O resultado desta enquete pode ser visualizado na Figura 1.2.

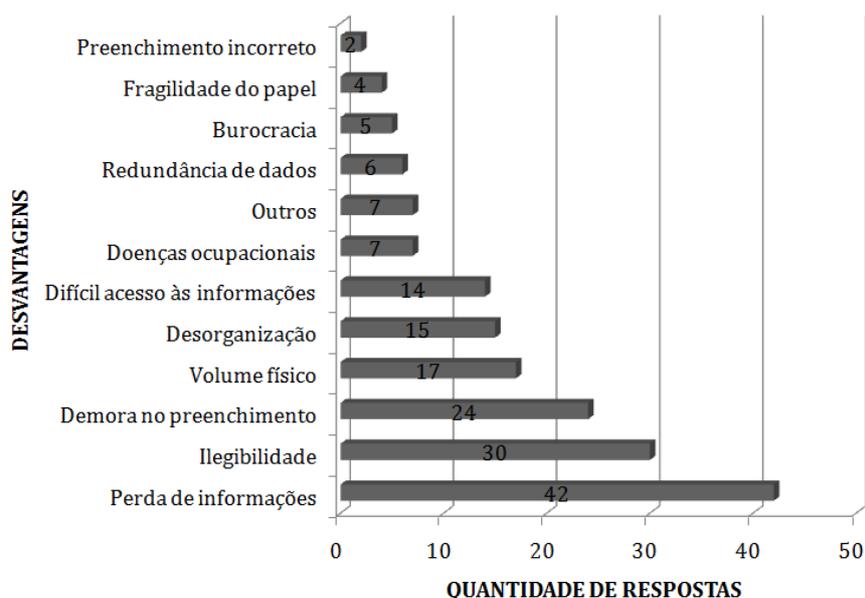


Figura 1.2: resultado da enquete sobre as desvantagens do prontuário em papel, adaptado de Costa (COSTA; MARQUES, 1999).

Nas últimas décadas, principalmente devido aos avanços e disponibilidade de soluções tecnológicas de processamento de dados e dos recursos de telecomunicações, está-se no limiar de uma mudança radical na forma em que são criadas, armazenadas, mantidas e recuperadas as informações de saúde, clínicas e administrativas, referentes a indivíduos e comunidades.

1.4 A Informática em Saúde e o Prontuário Eletrônico do Paciente

Com a evolução da informática e a redução do custo computacional, viu-se um crescimento exponencial de aplicações de informática na área de saúde, constituindo e firmando um novo campo na ciência: a Informática em Saúde. Com essa evolução, os antigos prontuários do paciente baseados em papel estão sendo informatizados, criando-se o conceito de Prontuário Eletrônico do Paciente (PEP) que, além de registrar as informações em meio eletrônico, introduz também novos conceitos que permitem a utilização de novos recursos no registro e recuperação dessas informações (COSTA, 2001).

Grande interesse, investigação e experiências estão ocorrendo em vários países, principalmente na Europa e Estados Unidos. No Brasil, há vários anos existem grupos de pesquisadores dedicados à exploração de três temas centrais relativos ao prontuário: arquitetura, conteúdo e desenvolvimento de software aberto. Recentemente, estas investigações e experiências ganharam impulso graças ao interesse do Ministério da Saúde em estabelecer em âmbito nacional um sistema padronizado de coleta e processamento de dados clínicos e administrativos originados em cada contato com o sistema de atendimento. Isto está concretizado no Projeto Cartão Nacional de Saúde, que traz um conjunto de padrões para identificar o paciente de forma unívoca em todo o país. Além disso, a perspectiva de recursos da área de telecomunicação através do Fundo de Universalização de Serviços de Telecomunicações (FUST) cria a possibilidade de conectar toda a rede assistencial e compartilhar a informação de saúde em benefício do paciente.

Adicionalmente, desde 2002 a Sociedade Brasileira de Informática em Saúde (SBIS) trabalha em parceria com o Conselho Federal de Medicina (CFM). O primeiro resultado desta parceria foi a criação da Resolução N° 1639/2002, que aprovou as “Normas Técnicas para o Uso de Sistemas Informatizados para a Guarda e Manuseio do Prontuário Médico”, dispondo sobre o tempo de guarda dos

prontuários, estabelecendo critérios para certificação dos sistemas de informação e dando outras providências. Posteriormente, esta resolução foi revogada e substituída pela Resolução N° 1821/2007, que aprovou as “Normas Técnicas Concernentes à Digitalização e Uso dos Sistemas Informatizados para a Guarda e Manuseio dos Documentos dos Prontuários dos Pacientes, Autorizando a Eliminação do Papel e a Troca de Informação Identificada em Saúde”.

O segundo resultado desta parceria consistiu da elaboração do Manual de Requisitos de Segurança, Conteúdo e Funcionalidades para Sistemas de Registro Eletrônico em Saúde (S-RES). Desde então, este manual tem sido refinado e atualmente é composto por um conjunto de orientações que dispõem sobre os requisitos que devem ser atendidos pelos sistemas informatizados de prontuário do paciente.

O terceiro resultado disponibilizado nesta parceria se trata de um processo de certificação, no qual é verificado se o sistema proposto atende a todos os requisitos especificados, estando, então, certificado para utilização de acordo com a SBIS e o CFM. Todos estes esforços têm contribuído consideravelmente para o desenvolvimento da área no Brasil.

De modo geral, o princípio básico do PEP baseia-se na integração da informação clínica e administrativa de pacientes individuais. Desta forma, uma vez coletada a informação, é registrada em um determinado formato para fins de armazenamento e esse registro passa a ser fisicamente distribuído entre hospitais, operadoras de planos de saúde, clínicas, laboratórios e demais setores envolvidos, sendo compartilhado entre os profissionais de saúde, de acordo com os direitos de acesso de cada um. Além da integração, um dos requisitos básicos do PEP é a interoperabilidade, que é a habilidade de dois ou mais sistemas computacionais trocarem informações, de modo que a informação trocada possa ser utilizada (KISSINGER; BORCHARDT, 1995).

O PEP é também apresentado como proposta para atender as demandas dos novos modelos de atenção e de gerenciamento dos serviços de saúde. Atualmente, observa-se claramente uma mudança na maneira em que o cuidado é prestado, onde e por quem. A missão dos serviços de saúde dos países está mudando e a tecnologia de informação precisa, conseqüentemente, mudar para continuar atendendo as necessidades de seus usuários. Os modernos sistemas de informação em saúde devem ser construídos de forma a apoiar o processo local de atendimento, sendo, portanto, orientados aos processos, apoiando o trabalho diário e fornecendo

comunicação dentro e fora da instituição, tendo uma estrutura comum. Deve existir um único registro por paciente que atenda às novas demandas de acompanhamento da produção, do custo e da qualidade. Para tanto, alguns pré-requisitos são identificados:

- estrutura padronizada e concordância sobre a terminologia;
- definição de regras claras de comunicação;
- arquivamento;
- segurança;
- privacidade.

Embora sendo apresentado como forte tendência e artigos científicos afirmarem que todos os países, se ainda não têm, haverão de ter no futuro um PEP como modelo para registro de informações clínicas, a maioria dos sistemas em uso infelizmente ainda não é direcionada por tal metodologia de desenvolvimento.

1.4.1 Definições do PEP

Atualmente não há um consenso quando se trata do conceito do termo PEP. O *Institute of Medicine* (IOM) entende que o prontuário eletrônico do paciente é um registro eletrônico que reside em um sistema especificamente projetado para apoiar os usuários, fornecendo acesso a um conjunto completo de dados corretos, alertas, sistemas de apoio ao diagnóstico e outros recursos, como *links* para bases de conhecimento médico.

Por sua vez, o *Computer-based Patient Record Institute* define o prontuário eletrônico ressaltando que um registro computadorizado de paciente é informação mantida eletronicamente sobre o estado de saúde e os cuidados que um indivíduo recebeu durante toda sua vida.

Outras definições também podem ser citadas, porém todas procuram descrever o mesmo conceito apenas com palavras distintas, o que torna inviável considerar cada uma delas ou adotar uma em específico, visto que o conceito de uma complementa o conceito das demais.

1.4.2 Denominações do PEP

Além de conceitos distintos, também há várias denominações para o PEP. O termo Prontuário Eletrônico do Paciente é o mais conhecido e divulgado na língua portuguesa. Porém, atualmente não é mais recomendado por não refletir todas as funcionalidades que podem ser agregadas a um sistema deste tipo. Por este motivo, os termos Registro Eletrônico de Saúde (RES) e Sistema de Registro Eletrônico de Saúde (S-RES) vêm sendo adotados.

Nos países de língua inglesa o RES também é conhecido como: *Computer-Based Patient Record*, *Electronic Medical Record Systems*, *Electronic Patient Record* e *Electronic Health Record*.

Esta diversidade de definições e denominações pode trazer complicações para o entendimento correto das funcionalidades que devem existir em cada caso. Com o objetivo de definir diferenças entre as nomenclaturas, Peter Waegemann propôs a definição de cinco níveis evolutivos para o RES.

1.4.3 Níveis evolutivos do Registro Eletrônico de Saúde

Para definir e enfatizar as etapas do processo evolutivo na criação de um RES e das diferenças entre as denominações, Peter Waegemann, diretor do *Medical Records Institute* nos Estados Unidos, identifica cinco níveis que vão do registro em papel ao registro eletrônico de saúde (WAEGEMANN, 1996).

Primeiro nível - identificado por Registro Médico Automático - o prontuário é armazenado em papel, apesar do fato de que aproximadamente 50% das informações tenham sido geradas por computadores. Desta forma, papel e registro eletrônico coexistem.

Segundo nível - identificado por Sistema de Registro Médico Computadorizado - é muito semelhante ao primeiro nível, exceto pelo fato de que incorpora imagens capturadas via *scanners*. Em geral, esse tipo de sistema é departamentalizado e com pouca integração.

Terceiro nível - identificado por Registro Médico Eletrônico - diferentemente do segundo nível, requer que o sistema esteja implantado na instituição por completo e contenha elementos como integração com sistemas especialistas, como alertas clínicos e programas de educação para o paciente. Neste nível, os requisitos de confidencialidade, segurança e proteção dos dados são atendidos.

Quarto nível - identificado por Sistema de Registro Eletrônico do Paciente - o escopo da informação presente é maior do que o suposto registro médico, ou seja, o registro eletrônico inclui informações que não estão presentes no seu registro médico básico e vão além das paredes da instituição que está atendendo o paciente. Desta forma, este nível requer que a identificação do paciente seja única e feita em nível nacional.

Quinto nível - identificado por Registro Eletrônico de Saúde - inclui uma rede de fornecedores e locais, tendo o paciente como centro. A informação não é baseada somente nas necessidades do serviço de saúde, mas sim na saúde e doenças do indivíduo e da comunidade.

A partir desta classificação, é possível distinguir os RES existentes e identificar, dentre as diversas denominações, em qual nível cada um se encontra. Assim, é possível diminuir a confusão causada pelas várias denominações e planejar o nível evolutivo em que se deseja chegar.

1.4.4 Atributos do Registro Eletrônico de Saúde

O IOM, instituição que tem demonstrado maior estímulo para a criação e implantação do RES, define um grupo de doze atributos que constituem o estado-da-arte na criação, desenvolvimento e implantação do RES (MASSAD; MARIN; NETO, 2003):

1. Oferecer uma lista de problemas que indique os problemas atuais e progressos do paciente. Uma lista de problemas deve denotar o número de ocorrências associadas com o passado e o problema corrente, assim como o estado (ativo, inativo, resolvido, indeterminado, entre outros) atual de cada problema.
2. Ter capacidade de medir o estado funcional e de saúde do paciente. Estas medidas de resultados não têm sido efetivamente tratadas pelos vendedores de sistemas. Em um mercado de saúde crescentemente mais competitivo, é imperativo dar mais atenção a medidas de resultado e de qualidade do cuidado prestado.
3. Permitir a documentação do raciocínio clínico em diagnósticos, conclusões e na seleção de intervenções terapêuticas. Permitir compartilhar o raciocínio clínico com outros profissionais, desenvolver meios automáticos para acompanhar os caminhos no processo de tomada de decisão.

4. Ser um registro longitudinal abrangendo toda a vida do paciente, ligando todos os dados de consultas e atendimentos anteriores.
5. Garantir confidencialidade, privacidade e apoiar os processos de auditoria clínica e administrativa; os desenvolvedores de sistemas precisam suprir os diferentes níveis de segurança para garantir acesso adequado às informações confidenciais do paciente.
6. Oferecer acesso contínuo aos usuários autorizados. Os profissionais de saúde precisam ser capazes de acessar os registros dos pacientes a qualquer momento.
7. Permitir visualização simultânea e personalizada dos dados do paciente pelos profissionais, departamentos e empresas. Esta capacidade melhora a eficiência do trabalho de usuários específicos, permitindo que o dado seja apresentado no formato que é mais usado por estes usuários. A flexibilidade em permitir diferentes e simultâneas visualizações dos dados é uma característica que a maioria dos fabricantes têm dificuldade de atender.
8. Apoiar o acesso a recursos de informação locais e remotos, como bases de dados em texto, correio eletrônico, CD-ROM, entre outros. O acesso às fontes externas deve garantir ao profissional a obtenção da informação necessária para apoiar o cuidado ao paciente.
9. Facilitar a resolução de problemas clínicos fornecendo instrumentos de análise e decisão. Exemplos destes instrumentos são os alertas e os sistemas de apoio ao diagnóstico.
10. Apoiar a entrada de dados diretamente pelo médico através de mecanismos e interfaces simples e diretas.
11. Apoiar os profissionais no gerenciamento e controle de custos para melhoria da qualidade. Esta área não tem sido levada em consideração frequentemente, mas é de grande importância para auxiliar o controle administrativo e financeiro dos sistemas de atenção à saúde, disponibilizando uma margem de competitividade no mercado.
12. Ter flexibilidade para apoiar a incorporação de necessidades existentes e futuras.

McDonald e Barnett (1990) afirmam que todo o potencial de um RES pode ser obtido somente se alguns fatores são observados. Todas as informações sobre os

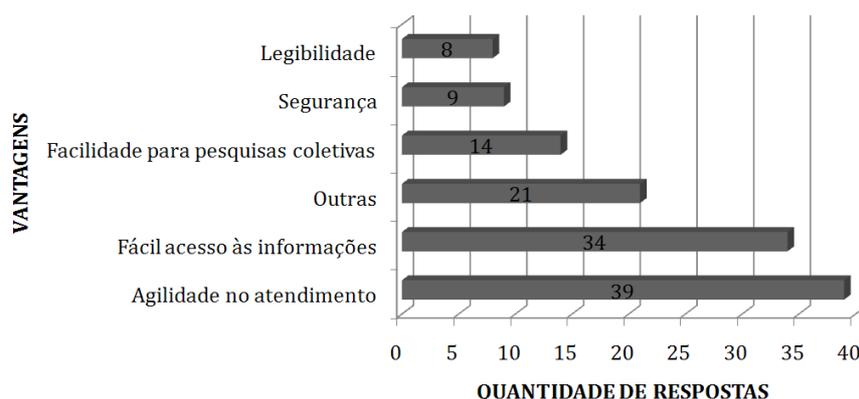


Figura 1.3: resultado da enquete sobre as vantagens do registro eletrônico de saúde, adaptado de Costa (COSTA; MARQUES, 1999).

pacientes devem estar armazenadas. Não se deve, portanto, restringir-se apenas às informações referentes aos pacientes internados. Os dados devem ser armazenados indefinidamente, estando as informações dos últimos anos rapidamente disponíveis e não somente aquelas referentes à última visita. Em vários hospitais, a informação dos últimos cinco anos está imediatamente disponível. Dados em texto livre dificultam a recuperação eficaz das informações e, por este motivo, os dados devem ser armazenados sempre que factível sob forma estruturada e codificados em um vocabulário comum, permitindo a incorporação de sistemas de alerta e de apoio ao diagnóstico. Devem existir em número suficiente e serem distribuídos através de um estudo criterioso de utilização esperada. Ajustes são certamente necessários. A tendência é colocar terminais no “ponto de cuidado”, frequentemente à beira do leito, de forma que os dados possam ser coletados e consultados no local em que os mesmos foram gerados ou utilizados (MASSAD; MARIN; NETO, 2003).

No trabalho desenvolvido por Costa (COSTA; MARQUES, 1999) foi realizada uma enquete para saber a opinião dos funcionários da Maternidade Escola Januário Cicco (Natal, Rio Grande do Norte) a respeito das vantagens de utilizar o RES. O resultado pode ser visualizado na Figura 1.3.

Qualidades de um sistema de sucesso, apontadas por Stetson e Andrew (1996) são (MASSAD; MARIN; NETO, 2003):

- deve ser rápido, permitindo que o usuário realize o acesso e recupere a informação sem dificuldades;
- deve possuir uma interface gráfica familiar ao usuário;

- deve ser flexível, permitindo a personalização do estilo de documentação e facilitando o acesso às informações necessárias para todos os tipos e categorias profissionais;
- deve melhorar o fluxo de trabalho e, conseqüentemente, a eficiência;
- deve permitir que o usuário visualize o sistema como garantia de melhora na documentação.

A partir das vantagens e recomendações de qualidade apresentadas, podem ser identificados alguns desafios a serem enfrentados durante o desenvolvimento de um RES.

1.5 Desafios no desenvolvimento do Registro Eletrônico de Saúde

De uma forma geral, pode-se afirmar que a tecnologia não é o problema para o desenvolvimento do RES. Na verdade, afirmar o contrário seria completamente inadequado. A tecnologia, no nível de desenvolvimento atual, fornece todas as funcionalidades necessárias para desenvolver um RES que atenda a todos os atributos de qualidade mencionados. Porém, o desenvolvimento de qualquer sistema de informação envolve a necessidade de gerenciar riscos. Os riscos podem ser originados tanto a partir da própria equipe de desenvolvimento quanto por fatores externos e o RES, obviamente, não está imune. Alguns riscos e obstáculos que devem ser enfrentados estão descritos a seguir (DICK; STEEN; DETMER, 1997; KOHN, 1999; ANDERSON, 1999).

A falta de entendimento das capacidades e benefícios do RES podem ser fatores limitantes para o seu desenvolvimento. É importante que todos os usuários do sistema e a diretoria da instituição estejam cientes de todos os recursos e benefícios que o RES pode oferecer. Sem o devido entendimento, os usuários podem não vislumbrar todos os recursos que podem usufruir, levando os analistas de sistema a fazer um levantamento de requisitos deficiente. Isto acarreta o desenvolvimento de um sistema ineficiente, incapaz de atender as necessidades reais dos usuários. Por isso, é fundamental que esteja presente na equipe de desenvolvimento, além dos profissionais de saúde na condição de futuros usuários, um profissional experiente em Informática em Saúde.

A falta de padronização provoca a perda ou inviabiliza a utilização de muitos dos recursos que podem ser disponibilizados, como alertas, sistemas de apoio ao diagnóstico, pesquisas clínicas, dentre outros. Além disso, para que os dados sejam armazenados de forma estruturada, o requisito fundamental do RES - a entrada destes dados - deve também ser feita de forma estruturada. Texto livre, embora mais aceito pelos profissionais por ser semelhante aos hábitos de documentação por escrita à mão no prontuário em papel, dificulta sua captura, quando não a inviabiliza.

A construção de sistemas que não valorizam a segurança e confidencialidade dos dados do paciente podem estar fadados ao fracasso e desencadearem processos legais contra a instituição. Além disso, contribuem para criar ou aumentar a falta de confiança dos usuários.

Para realizar o intercâmbio de dados e gerenciamento de recursos, é necessário que sejam adotados padrões de comunicação, leis e regras que regulamentem o processo de transmissão, além de especialistas no desenvolvimento de RES e redes locais, regionais e nacionais.

Estar convencido da necessidade de mudar e aceitar a incorporação de novos recursos não significa comportamento alterado. Sistemas que interferem nos hábitos rotineiros das pessoas, em geral, não são bem aceitos ou demoram algum tempo para serem aceitos, exigindo, portanto, envolvimento e constante treinamento e ensino.

Diante do exposto, pode-se concluir que o desenvolvimento de um RES é um grande desafio, principalmente devido à complexidade intrínseca do domínio em que está inserido. As principais dificuldades estão situadas na natureza organizacional ou relacionadas a forma de trabalho tradicional dos profissionais de saúde. Sistemas integrados pressupõem não somente serviços e organizações integradas, mas principalmente profissionais treinados e integrados. Este aspecto representa, muitas vezes, uma barreira crítica no desenvolvimento e adoção de um RES.

Outros obstáculos que podem ser identificadas para se obter um RES são: falta de planejamento estratégico na implantação do sistema, pouco ou nenhum incentivo interno da organização para atingir a integração clínica, uma vez que a idéia de visualizar o todo para poder tratar uma das partes não é praticada por muitos; autonomia dos hospitais e, principalmente, falta de planejamento da saúde da população (MASSAD; MARIN; NETO, 2003).

Como resultado desta discussão, é importante destacar que antes de iniciar a implantação de um RES em uma determinada instituição, é necessário que alguns requisitos sejam atendidos:

- promover uma mudança comportamental dos profissionais, demonstrando a ineficiência dos registros manuais para lidar com a grande quantidade de informações geradas no sistema de saúde;
- promover uma mudança nos sistemas e adotar sistemas computacionais abertos, que atendam aos requisitos de interoperabilidade;
- utilizar tecnologia moderna, evitando tecnologias obsoletas;
- utilizar normas (padrões) no registro clínico e na transferência das informações em saúde;
- desenvolver e atualizar a legislação, atendendo às necessidades que são impostas pelo desenvolvimento científico e tecnológico com a criação de leis nacionais para estabelecer requisitos mínimos no uso de padrões e melhoria da qualidade da assistência à saúde.

No desenvolvimento e implantação de um RES, algumas ações podem facilitar e direcionar o processo para garantir o êxito. Desta forma, a equipe responsável deve (DICK; STEEN; DETMER, 1997; KOHN, 1999; ANDERSON, 1999):

- identificar e entender os requisitos para o projeto do RES;
- desenvolver, adotar e implantar padrões;
- incluir os usuários no processo de desenvolvimento e implantação;
- pesquisar e conhecer experiências de desenvolvimento;
- demonstrar eficácia e eficiência (custo/benefício);
- reduzir as limitações legais para o uso, elaborando regras e leis que protejam a privacidade dos pacientes;
- preparar a infraestrutura necessária antes de implantar;
- coordenar os recursos e apoio necessários para o desenvolvimento e sua difusão;

- educar e treinar usuários e desenvolvedores;
- garantir soluções de interface adequadas;
- procurar alternativas para redução de custos (tecnologias abertas, projetos colaborativos, entre outros);
- avaliar o processo de implantação do sistema e acompanhar a aceitação do usuário;
- desmistificar questões de segurança e confidencialidade;
- obter o apoio incondicional da diretoria da instituição;
- poder comprovar o aumento da qualidade de atendimento do paciente.

Para concluir, é importante enfatizar que, alheio às dificuldades existentes no desenvolvimento de um RES, a tendência é justamente o crescimento no número de instituições que adotam este tipo de sistema. O investimento necessário é alto, tanto do ponto de vista humano como organizacional. Porém, se o objetivo final é a melhoria no atendimento à saúde da população, este investimento e os recursos necessários estão mais do que justificados.

1.6 Padronização

Desde a invenção da linguagem, símbolos são compartilhados e utilizados para a comunicação inteligível. Atualmente, a informática e o processamento digital da informação mudaram consideravelmente os mecanismos de armazenamento, processamento e transmissão de dados. Entretanto, independente da mídia utilizada, para que a representação e troca de informações possam ocorrer de forma compreensível entre dois ou mais usuários de um sistema de comunicação, são necessárias duas condições básicas: a) definição de um vocabulário comum para representação e registro de conceitos e; b) que a comunicação ocorra segundo um conjunto de regras compartilhadas pelos usuários. Esta visão clara e simples do processo geral de comunicação linguística é importante para compreender o mecanismo de elaboração e aplicação de padrões em saúde (AMARAL, 2003).

O termo padronização pode ser definido como uma atividade para estabelecer provisões de uso contínuo ou repetido, com o objetivo de atingir um grau otimizado

de sistematização e ordenação. Na área da saúde, existem diversas iniciativas para sistematização da informação. Algumas vêm evoluindo há séculos. Em termos de categorização de diagnósticos, a primeira iniciativa foi feita em 1662 por John Graunt. Neste ano, propôs uma tipologia de diagnósticos contendo 60 categorias de doenças. Outros esforços relacionados com a elaboração dos primeiros modelos classificatórios em Medicina estão dispostos a seguir: o *Genera Morborum* de Carolus Linnaeus (1763), o *Nosologia Methodica* de François Bossier de la Croix (1768) e a *Synopsis Nosologica Methodicae* de William Cullen (1785).

Em 1853, no 1º Congresso Mundial de Estatística, quando se pretendeu comparar frequências de doenças em diferentes agrupamentos populacionais, tornou-se evidente a necessidade da uniformização da linguagem médica.

Em 1886, Jacques Bertillon elaborou uma classificação com 161 categorias de diagnósticos. Esta classificação foi apresentada no congresso de Chicago em 1893 e seu sistema foi prontamente adotado por diversos países.

Esta foi a base da primeira versão da Classificação Internacional de Doenças (CID). Atualmente, a CID está na sua décima revisão e tornou-se fundamental para a organização, recuperação e análise de informações em medicina.

Ao longo de um século de evolução, foram criadas várias organizações nacionais e internacionais voltadas exclusivamente para a elaboração de normas padronizadas. A instituição de maior escopo é a *International Standards Organization* (ISO), organização internacional responsável pela padronização e normatização em diversas áreas. Na área de Informática em Saúde, a *American National Standards Institute - Healthcare Informatics Standards Board* (ANSI-HISB) coordena diversas organizações que trabalham com a criação de padrões nos Estados Unidos. Na Europa, existe o Comitê Europeu de Normatização (CEN), que também atua em diversas áreas.

No Brasil, a Associação Brasileira de Normas Técnicas (ABNT) coordena as atividades de padronização. Na área da saúde, o Ministério da Saúde vem liderando iniciativas neste aspecto. Com a criação da Rede Integrada de Informações para Saúde em 1996, foram formados diversos grupos e comissões que estão trabalhando para o estabelecimento de padrões comuns que possibilitem a intercomunicação de sistemas e bases de dados para a efetiva integração da informação em saúde.

Os padrões para registro de dados em saúde foram desenvolvidos com o objetivo

de representar todo tipo de conceito ou fato em medicina. Desta forma, existem padrões para descrever procedimentos, diagnósticos, medicamentos, exames de laboratório, dentre outros.

1.6.1 Doenças: a Classificação Internacional de Doenças (CID)

Uma classificação é um sistema no qual entidades específicas são agrupadas de acordo com critérios de similaridade. As classificações podem sofrer modificações importantes com o tempo, em que diferentes critérios de classificação são utilizados para formular e reformular os grupos. A CID vem sofrendo diversas modificações no decorrer de seus cem anos de existência. Na Tabela 1.1 é apresentada a evolução temporal do número de categorias de diagnósticos da CID (AMARAL, 2003).

Ano	Denominação	Categorias Principais	Total de subcategorias
1853	Farr / d'Espine	139	=
1893	Bertillon	161	=
1900	CID1	179	=
1909	CID2	189	=
1920	CID3	205	=
1929	CID4	214	=
1938	CID5	200	=
1948	CID6	954	=
1955	CID7	965	=
1968	CID8	1.040	=
1968	CID8-H	905	4.334
1975	CID9	1.164	8.173
1979	CID9-CM	1.179	14.473
1993	CID10	1.967	10.468

Tabela 1.1: evolução temporal da CID, adaptado de Amaral (AMARAL, 2003).

A divisão das categorias diagnósticas é feita segundo a localização topográfica da doença. Esta divisão segue a segmentação convencional da medicina em subespecialidades.

A divisão sistêmica dos capítulos é feita considerando-se aspectos anatômicos, morfológicos, etiológicos e funcionais. A estrutura atual da CID é mostrada na Tabela 1.2 (AMARAL, 2003). Do capítulo I ao XVIII existem 1967 diagnósticos gerais. Adicionando-se os capítulos XIX, XX e XXI totalizam-se 2468 diagnósticos. Além disso, cada categoria de diagnóstico geral é subdividida em diagnósticos mais

Capítulo	Sistema	Código
I	Algumas doenças infecciosas e parasitárias	A00-B99
II	Neoplasias [tumores]	C00-D48
III	Doenças do sangue e dos órgãos hematopoéticos e alguns transtornos imunitários	D50-D89
IV	Doenças endócrinas, nutricionais e metabólicas	E00-E90
V	Transtornos mentais e comportamentais	F00-F99
VI	Doenças do sistema nervoso	G00-G99
VII	Doenças do olho e anexos	H00-H59
VIII	Doenças do ouvido e da apófise mastóide	H60-H95
IX	Doenças do aparelho circulatório	I00-I99
X	Doenças do aparelho respiratório	J00-J99
XI	Doenças do aparelho digestivo	K00-K93
XII	Doenças da pele e do tecido subcutâneo	L00-L99
XIII	Doenças do sistema osteomuscular e tecido conjuntivo	M00-M99
XIV	Doenças do aparelho geniturinário	N00-N99
XV	Gravidez, parto e puerpério	O00-O99
XVI	Algumas afecções originadas no período perinatal	P00-P96
XVII	Malformações congênicas e anomalias cromossômicas	Q00-Q99
XVIII	Sintomas, sinais e achados anormais de exames clínicos e de laboratório, não classificados em outra parte	R00-R99
XIX	Lesões, envenenamento e outras conseqüências de causas externas	S00-T98
XX	Causas externas de morbidade e de mortalidade	V01-Y98
XXI	Fatores que influenciam o estado de saúde e o contato com os serviços de saúde	Z00-Z99

Tabela 1.2: estrutura da CID, adaptado de Amaral (AMARAL, 2003).

específicos, cerca de 10.000, que não são descritos neste trabalho.

1.6.2 Procedimentos: a CBHPM e a Tabela SUS

No Brasil, existem duas classificações principais relacionadas à descrição de procedimentos realizados em medicina. A lista de procedimentos utilizada pelo Sistema Único de Saúde (SUS) e a Classificação Brasileira Hierarquizada de Procedimentos Médicos (CBHPM), elaborada pela Associação Médica Brasileira (AMB).

Estas duas Tabelas são muito utilizadas nas instituições de saúde como padrões para registro de procedimentos realizados para fins de reembolso e pagamento de honorários. Genericamente, as instituições públicas utilizam-se da Tabela SUS e

as instituições privadas da Tabela CBHPM. Existe uma certa discussão entre adeptos de uma ou outra classificação, sendo que ambas possuem vantagens e desvantagens. Estas classificações são organizadas em capítulos, que correspondem às especialidades.

1.6.3 Medicamentos: a Classificação Anatômica-Terapêutica-Clínica

Dentre as classificações de medicamentos, a Classificação Anatômica-Terapêutica-Clínica (ATC) merece destaque por ser uma das mais importantes. A ATC, como o próprio nome indica, organiza a informação referente aos medicamentos seguindo hierarquias de: a) sistemas anatômicos; b) uso terapêutico e; c) classe química do medicamento. Atualmente, a ATC está sob administração do Centro Colaborador da Organização Mundial de Saúde localizado em Oslo, Noruega. Na Tabela 1.3 (AMARAL, 2003) é apresentado um exemplo de hierarquias de classes e subclasses de termos descrevendo substâncias químicas ou medicamentos.

Código	Descrição
C	Sistema cardiovascular (1º nível, grupo anatômico principal)
C03	- Diuréticos (2º nível, grupo terapêutico principal)
C03C	- Diuréticos de alto-nível (3º nível subgrupo terapêutico)
C03CA	- Sulfonamidas(4º nível, subgrupo terapêutico químico)
C02CA01	- Furosemida (5º nível, subgrupo para substância química)

Tabela 1.3: exemplos da ATC, adaptado de Amaral (AMARAL, 2003).

1.6.4 Materiais hospitalares: a Tabela SIMPRO

Existem diversas classificações para organizar materiais hospitalares. Uma das classificações mais utilizadas é a Tabela SIMPRO. Esta lista de materiais é organizada segundo grupos de materiais semelhantes, como por exemplo: agulhas, cateteres, sondas, entre outros. É construída com a colaboração de vários fornecedores de produtos hospitalares, que listam e classificam seus produtos de forma sistemática e não repetitiva. Alguns exemplos de materiais hospitalares, segundo a Tabela SIMPRO, são mostrados na Tabela 1.4 (AMARAL, 2003).

Código	Descrição
0767384	Cateter implantofix 270x1,4,40mm seldinger
0016433	Cateter infusão duplo lumen
28000191	Cateter intraport arterial
3953062	Cateter intravenoso 14g jelco 50mm
2800154	Cateter tenckhoff 180 neonatal
0016435	Cateter swan ganz
0012384	Cateter steffens p/dren.79cm 05fr p.oliber curvada

Tabela 1.4: exemplos da Tabela SIMPRO, adaptado de Amaral (AMARAL, 2003).

1.6.5 Imagens radiológicas: o padrão DICOM

O padrão *Digital Imaging and Communication Standards Committee* (DICOM) teve suas origens no ACR/NEMA, formada pela associação entre o Colégio Americano de Radiologistas e da Associação de Manufatura Elétrica Nacional. Este padrão é utilizado por muitos fabricantes de sistemas *Picture Archiving and Communication System* para armazenamento e transmissão de imagens radiológicas.

1.6.6 Comunicação em saúde: o padrão *Health Level Seven* (HL-7)

O Health Level Seven (HL-7) é um protocolo internacional para intercâmbio de dados eletrônicos em todos os ambientes da área de saúde, integrando informações de natureza clínica e administrativa. Sua utilização contribui para o desenvolvimento de sistemas e padrões relacionados com a troca, integração, compartilhamento e recuperação de informação eletrônica em saúde, para apoio da prática médica e administrativa, permitindo maior controle e gerenciamento dos serviços de saúde (HL7, 2010).

O HL-7 é dividido em várias áreas, dentre as quais podem ser citadas: registro do paciente, prescrição médica, prescrição de dietas, observações clínicas, solicitação e resultados de exames. No Brasil, as pesquisas nesta área são chefiadas pelo Dr. Marivan Santiago Abrahão através do Instituto HL-7, organização voluntária, sem fins lucrativos, voltada para o desenvolvimento de padrões e especificações que permitam a troca de informações entre sistemas de informação em saúde. Este instituto é afiliado à *HL7 Organization* e tem sido responsável por impulsionar pesquisas e fornecer capacitação aos profissionais que têm interesse neste padrão.

1.6.7 Usuários: o Projeto Cartão Nacional de Saúde

O Projeto Cartão Nacional de Saúde é um instrumento que possibilita a vinculação dos procedimentos executados no âmbito do SUS ao usuário, ao profissional que os realizou e também à unidade de saúde na qual foram realizados. Para tanto, faz-se necessária a construção de cadastros de usuários, profissionais de saúde e de unidades de saúde. A partir desses cadastros, os usuários do SUS e os profissionais de saúde recebem um número nacional de identificação que permite identificá-los de forma única em todo o país. Esta iniciativa pode contribuir de forma significativa para o desenvolvimento da área de informática em saúde por implementar parte dos requisitos necessários para o desenvolvimento de um registro eletrônico de saúde nacional.

1.7 Motivação

O desenvolvimento de sistemas informatizados com o objetivo de gerenciar informações clínicas e administrativas não é uma prática nova. Entretanto, grande parte das soluções disponíveis no mercado são desenvolvidas através de softwares e ferramentas proprietárias. O *ProDoctor* (PRODOCTOR, 2010), por exemplo, é um software médico para gestão de consultórios, clínicas e hospitais desenvolvido na plataforma proprietária Microsoft. O *Centralx Clinic* (CENTRALXCLINIC, 2010) é um software para gestão de clínicas, ambulatórios e hospitais que funciona na plataforma proprietária Windows.

Esta prática impõe a inclusão de um custo desnecessário, visto que existem ferramentas criadas em software livre que permitem o pleno desenvolvimento e utilização de um RES. O *OpenEMR* (OPENEMR, 2010) é um software de gerenciamento da prática médica bastante popular e totalmente desenvolvido em software livre, porém voltado ao sistema de saúde dos Estados Unidos. Também é notável a quantidade de sistemas que não garantem pré-requisitos básicos, como privacidade e interoperabilidade.

Nos últimos anos têm-se notado um aumento significativo na discussão sobre padronização e desenvolvimento de RES em código aberto. Durante este período, alguns trabalhos foram publicados.

Costa (COSTA, 2001) propõe o desenvolvimento de um PEP baseado na *WEB* e Engenharia de Software, em que pesquisas com pessoas envolvidas no processo de

atenção à saúde orientam o encaminhamento para a criação de um protótipo com interface *WEB*. Porém, não foi possível realizar testes com o protótipo pois o mesmo não estava mais disponível.

Cunha (CUNHA, 2005) expõe um estudo ressaltando a importância do PEP na integração de sistemas de informação em saúde, com foco na importância de sua utilização na melhoria do atendimento ao paciente.

Cancian (CANCIAN, 2006) propõe o desenvolvimento de um PEP baseado em software livre e a estrutura física da rede que dá o suporte à sua utilização. Este sistema foi implantado no Hospital Geral do Exército de Curitiba e sua versão de teste não está disponível.

Um fator que dificulta e muitas vezes impede o levantamento do estado-da-arte de RES desenvolvidos no Brasil é a indisponibilidade de versões de demonstração e teste. Também é possível notar que, até o presente momento, não existe um RES totalmente desenvolvido em software livre que observe os paradigmas da Engenharia de Software e procure utilizar os padrões nacionais e internacionais, incluindo o Manual de Certificação SBIS-CFM. Outro fator que também não é encontrado com frequência é a utilização de sistemas inteligentes de apoio ao diagnóstico integrados ao RES. Todos estes fatores motivaram a proposta de desenvolvimento deste trabalho.

1.8 Trabalho proposto

Diante das limitações das ferramentas analisadas e tomando como base os trabalhos científicos desenvolvidos sobre o tema, notou-se a carência de um RES desenvolvido totalmente em software livre que atingisse os padrões de qualidade necessários. Para atingir o RES esperado, alguns objetivos gerais e específicos devem estar satisfeitos.

1.8.1 Objetivos gerais

Desenvolvimento de um RES que garanta a integridade dos dados, padronização, inovação, utilização de ferramentas livres, interface *WEB*, facilidade de uso e integração com sistemas de apoio ao diagnóstico.

1.8.2 Objetivos específicos

- i. Realizar a revisão bibliográfica do tema;
- ii. Fazer uma revisão crítica dos RES existentes;
- iii. Estudar os padrões de software e comunicação entre equipamentos médicos;
- iv. Avaliar as tecnologias baseadas em software livre e *Internet*;
- v. Definir a arquitetura e modelagem visando a modularidade e integração;
- vi. Estudar usabilidade com foco na adequação ao público-alvo;
- vii. Implementar os módulos básicos de funcionamento;
- viii. Realizar a integração com sistemas de apoio ao diagnóstico;
- ix. Verificar e validar as funcionalidades;
- x. Disponibilizar uma versão de teste aberta na *Internet*;
- xi. Incluir o código-fonte e a documentação em um repositório de software livre.

Este trabalho tem como objetivo desenvolver a arquitetura básica de operação de um RES, incluindo gerenciamento de permissões, usuários, categorias, perfis, consultas, eventos, exames, procedimentos, imunizações, troca de mensagens, emissão automatizada de prescrições médicas, armazenamento de histórico de doenças, queixas gerais, integração com sistemas de apoio ao diagnóstico, dentre outras funcionalidades. Desta forma, o RES desenvolvido neste trabalho possui toda a infraestrutura necessária para a operação básica, permitindo através do seu modelo arquitetural ser facilmente expandido e adaptado de acordo com as necessidades.

1.9 Organização do trabalho

O restante deste trabalho está dividido em três capítulos: projeto e implementação do RES, no qual são apresentadas as etapas desenvolvidas e que permitiram a criação do sistema; as funcionalidades e interfaces do RES, em que são expostos os resultados obtidos através das etapas desenvolvidas e; contribuições, conclusões e trabalhos futuros, em que são sintetizadas as contribuições realizadas neste trabalho, assim como as conclusões obtidas e os trabalhos futuros.

Para complementar o conteúdo, no Apêndice A são definidos os conceitos relacionados à Engenharia de Software e Modelagem de Sistemas necessários para compreender as técnicas utilizadas no Capítulo 2, enquanto no Apêndice B são especificados os casos de uso definidos para o RES proposto.

Capítulo 2

Projeto e implementação do RES

2.1 Introdução

2.1.1 Visão geral

O desenvolvimento de um Registro Eletrônico de Saúde não é um processo trivial. Fontes de informações diversas, divergências de padrões utilizados, resistência por parte dos profissionais envolvidos, dentre outros fatores, contribuem para torná-lo um dos maiores desafios na área de Informática em Saúde.

Entretanto, a Tecnologia da Informação atingiu um patamar no qual são disponibilizadas diversas soluções que podem ser utilizadas para atender todos os requisitos existentes no desenvolvimento de sistemas deste tipo, por mais complexos que sejam. Porém, é um trabalho que demanda grande esforço no sentido de padronizar os dados a serem trocados pelas diversas instituições responsáveis por originar e disponibilizar tais dados.

Como parte do esforço no sentido de estabelecer sistemas que seguem as técnicas e a experiência adquirida ao longo de vários anos de desenvolvimento de software, integradas aos mais recentes padrões de dados formulados na área de saúde, este trabalho propõe o desenvolvimento de um RES baseado em software livre.

Desta forma, com o intuito de esclarecer as técnicas e tecnologias utilizadas no decorrer do seu desenvolvimento, neste Capítulo são apresentadas todas as etapas executadas, desde o planejamento até a disponibilização *on-line* de uma versão de demonstração, iniciando pelo emprego das técnicas de engenharia de software através

da definição do modelo de processo adotado no desenvolvimento do trabalho.

2.1.2 Definição do modelo de processo

Os modelos de processo de software definem um conjunto distinto de atividades, ações, tarefas, marcos e produtos de trabalho que são necessários para desenvolver um software de qualidade. Em outras palavras, estes modelos são uma representação abstrata de um determinado processo de software e descrevem como as atividades são abordadas para se obter um produto de software (PRESSMAN, 2006).

Neste sentido, os modelos de processo permitem organizar a sequência em que as atividades são realizadas durante o desenvolvimento de um software. Existem diversos modelos de processo de software disponíveis, sendo que no desenvolvimento de um determinado software podem ser utilizados um ou mais modelos adequados à realidade deste software. A descrição detalhada dos modelos de processo mais comuns pode ser encontrada no Apêndice A.

A escolha do modelo de processo a ser utilizado durante o desenvolvimento de um software é uma decisão de extrema relevância, da qual dependem diretamente o bom andamento e sucesso final do desenvolvimento. Uma escolha inadequada pode trazer impactos negativos graves, demandando aumento de custo, atrasos e até mesmo insucesso no desenvolvimento. Para que esta situação não ocorra, a natureza do software a ser desenvolvido deve ser analisada em detalhes, assim como devem ser observados fatores como conhecimento dos requisitos, probabilidade de mudanças, nível de complexidade, público-alvo, dentre outros.

Para o RES proposto, baseado na natureza complexa do software, na probabilidade de mudanças, no desconhecimento de parte dos requisitos do software no início do desenvolvimento, a opção que melhor se adapta ao cenário existente é o modelo de processo conhecido como Desenvolvimento Incremental. O fluxo de atividades deste modelo de processo pode ser visualizado na Figura 2.1.

Neste modelo de processo, as atividades de levantamento e análise de requisitos, projeto, implementação, verificação e validação são intercaladas em ciclos. Inicialmente é realizado o levantamento e agrupamento de requisitos de acordo com o grau de importância, sendo cada agrupamento chamado de incremento. Cada incremento define um grupo de funcionalidades que devem ser desenvolvidas, sendo os requisitos mais importantes incluídos nos primeiros incrementos. Baseado no desenvolvimento de um incremento e de acordo com os resultados gerados, novos

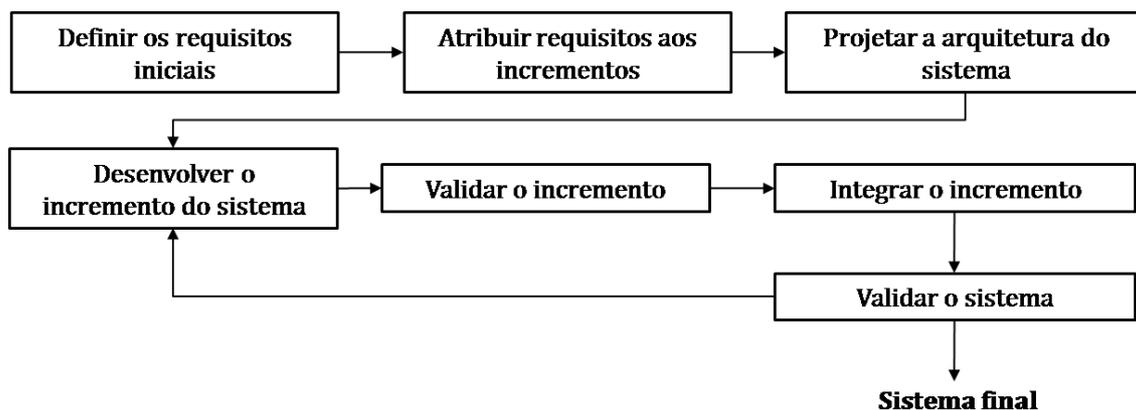


Figura 2.1: modelo incremental, adaptado de Sommerville (SOMMERVILLE, 2007).

requisitos podem ser identificados e planejados.

Partindo da análise dos requisitos definidos, dois incrementos iniciais são identificados:

- 01 - autenticação e autorização;
- 02 - prontuário do paciente.

O primeiro incremento tem como objetivo principal criar a infraestrutura necessária de gerenciamento de usuários, autenticação, autorização, permissões e grupos para permitir a utilização dos diferentes perfis de usuário existentes no contexto do software.

O segundo incremento tem como objetivo principal fornecer as funcionalidades básicas de operação de um sistema de registro eletrônico de saúde, seguindo as recomendações das instituições de padronização nacionais e internacionais, além de incluir ferramentas de apoio ao diagnóstico.

Outros incrementos foram identificados mas não foram incluídos para desenvolvimento no escopo deste trabalho, podendo serem encontrados na Seção 4.3.

A seguir é descrita a etapa de levantamento e análise de requisitos, na qual são discutidos os meios utilizados para obter os requisitos de cada um dos incrementos definidos, assim como é feita a definição de cada requisito identificado.

2.2 Levantamento e análise de requisitos

A etapa de levantamento e análise de requisitos tem como objetivo principal definir o grupo de requisitos funcionais e não funcionais do software que vai ser desenvolvido. Este processo não possui um formato ideal para todos os tipos de software, devendo ser adaptado de acordo com cada situação. Nas seções a seguir são descritas as etapas que geralmente estão presentes em processos deste tipo.

2.2.1 Identificação de requisitos

Em geral, o processo de levantamento e análise de requisitos é iniciado através de reuniões com os clientes ou através de pesquisa e levantamento bibliográfico, nos quais são obtidos os requisitos de usuário. Este tipo de requisito geralmente é descrito em linguagem natural e composto por declarações gerais de quais serviços o software deve disponibilizar. Esta etapa é conhecida como identificação de requisitos.

Para o RES proposto, esta etapa é composta por levantamento bibliográfico e análise crítica dos sistemas de registro eletrônico de saúde existentes no mercado nacional e internacional. Durante a execução desta etapa, o manual SBIS-CFM mostrou ter grande valia ao disponibilizar a especificação de requisitos que devem ser considerados ao desenvolver sistemas deste tipo.

2.2.2 Refinamento de requisitos

O principal problema dos requisitos de usuário é que são muito gerais e não definem em detalhes as características que devem ser observadas no desenvolvimento de cada uma das funcionalidades. Entretanto, servem de base para a elaboração dos requisitos de sistema, que definem detalhadamente as funções, serviços e as restrições operacionais do sistema. Esta etapa é conhecida como refinamento de requisitos e tem como resultado a elaboração do documento de requisitos.

2.2.3 Documento de requisitos

O documento de requisitos tem como propósito coletar, analisar e definir as necessidades do software e as características de alto nível que o mesmo deve prover, focando nos requisitos técnicos identificados e no motivo destes existirem. Desta forma, torna-se o documento oficial utilizado na análise, projeto, implementação, verificação e validação do software. Por este motivo, sua elaboração

deve ser considerada uma das atividades mais importantes de todo o processo de desenvolvimento.

Para o RES proposto, baseado no modelo de processo utilizado, os requisitos estão agrupados de acordo com o incremento ao qual pertencem, conforme apresentado nas seções a seguir.

Prioridades de requisitos

Para estabelecer a prioridade dos requisitos, são adotadas as denominações “essencial”, “importante” e “desejável”. A prioridade dos requisitos é utilizada no gerenciamento do escopo das etapas do projeto e na definição das prioridades durante o desenvolvimento. A descrição de cada prioridade é apresentada na Tabela 2.1.

Prioridade	Descrição
Essencial	Requisito sem o qual o software não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, devendo ser desenvolvidos nas primeiras etapas de implementação.
Importante	Requisito sem o qual o software entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implantados o mais rápido possível, mas, se não forem, parte do sistema pode ser implantada mesmo assim.
Desejável	Requisito que não compromete as funcionalidades básicas do software, isto é, o software pode funcionar de forma satisfatória sem o mesmo. Requisitos desejáveis são requisitos que podem ser implantados por último, sem comprometer o funcionamento do software.

Tabela 2.1: prioridades dos requisitos.

Requisitos do incremento 01 - autenticação e autorização

Esta seção descreve os requisitos funcionais (RF) e não-funcionais (RNF) referentes ao incremento 01. Cada requisito está identificado por uma chave única, composta pelo código que representa o tipo de requisito e um número sequencial.

Tabela 2.2: requisitos do incremento 01.

Requisito	Descrição	Prioridade
[RF001] O usuário	Os usuários dos RES compõem todos os profissionais de saúde envolvidos no processo e também os pacientes. Em outras palavras, o paciente também deve ser um usuário do RES que possua permissões de acesso às suas informações.	Essencial
[RF002] Categorias de usuário	Devem existir as seguintes categorias de usuário (no mínimo): paciente, <i>root</i> , administrador, médico, enfermeiro e recepcionista. A criação posterior de categorias deve estar disponível.	Essencial
[RF003] Método de autenticação	Utilizar, no mínimo, o método de usuário, senha e verificador <i>captcha</i> .	Essencial
[RF004] Proteção dos parâmetros de autenticação	Todos os dados ou parâmetros utilizados no processo de autenticação de usuário devem ser armazenados de forma protegida. Por exemplo, armazenar somente o código <i>hash</i> da senha do usuário e garantir que o local de armazenamento possua restrições de acesso.	Essencial
[RF005] Segurança de senhas	Utilizar os seguintes controles de segurança: qualidade da senha (verificar a qualidade da senha no momento de sua definição pelo usuário, obrigando a utilização de, no mínimo, 8 caracteres dos quais, no mínimo, 1 caractere deve ser não numérico), periodicidade de troca de senhas (o RES deve ter a funcionalidade de obrigar a troca de senhas pelos usuários, em um período máximo configurável) e armazenar a senha de forma codificada (como exemplo, armazenar somente o código <i>hash</i>).	Essencial
[RF006] Controle de tentativas de autenticação	O RES deve ter mecanismos para bloquear o usuário após um número máximo configurável de tentativas inválidas de autenticação.	Importante
[RF007] Encerramento por inatividade	A sessão de usuário deve ser encerrada após um período configurável de inatividade, invalidando o parâmetro de controle da sessão como, por exemplo, o <i>cookie</i> .	Importante

Continua na próxima página

Tabela 2.2 – continuação da página anterior.

Requisito	Descrição	Prioridade
[RF008] Alteração de senha	Todo usuário deve ter a opção de alterar sua senha no momento que julgar oportuno.	Essencial
[RF009] Alteração de dados pessoais	Todo usuário deve ter a opção de alterar seus dados pessoais quando necessário. As informações devem incluir: nome completo, naturalidade, nacionalidade, e-mail, telefones, religião, nome do pai, nome da mãe, CPF, RG, data de nascimento, data de criação do cadastro, sexo, tipo sanguíneo, endereços, nome de usuário, senha (apenas o <i>hash</i>), próxima data de alteração de senha, data do último acesso, número de tentativas de autenticação com erro, grupos e categorias das quais faz parte. Algumas destas informações podem ser modificadas apenas por um administrador (grupos e categorias).	Essencial
[RF010] Endereços	Os usuários devem possuir a opção de cadastrar mais de um endereço no RES.	Essencial
[RF011] Funcionalidades disponíveis de acordo com as permissões	As funcionalidades devem estar disponíveis somente para os usuários que têm permissões para acessá-las. Por este motivo, deve haver um sistema de gerência de permissões que permita associar um usuário às suas permissões, sendo as funcionalidades equivalentes apresentadas na interface principal do usuário após sua autenticação.	Essencial
[RF012] Permissões	O administrador deve possuir uma interface que permita pesquisar, criar, alterar e excluir permissões do sistema. Estas permissões devem definir o componente, método e descrição a que estão associadas. Estas permissões, por si sós, não são responsáveis pela associação com os usuários, ou seja, são um mero cadastro.	Essencial

Continua na próxima página

Tabela 2.2 – continuação da página anterior.

Requisito	Descrição	Prioridade
[RF013] Categorias de permissões	O administrador deve possuir uma interface que permita pesquisar, criar, alterar e excluir categorias de permissões. Estas categorias devem possuir um nome e as permissões relacionadas. Desta forma, quando um usuário for associado a uma categoria, automaticamente herdará todas as permissões da mesma. Por exemplo: administrador, paciente e médico são categorias que possuem permissões distintas.	Essencial
[RF014] Grupos	O administrador deve possuir uma interface que permita pesquisar, criar, alterar e excluir grupos. Estes grupos devem ser mantidos em conjunto com as categorias de permissões, mas não estão relacionados diretamente a estas. Seu principal objetivo é a mera classificação em grupos que não estejam relacionados com categorias, ou seja, são macro classificações.	Essencial
[RF015] Gerência de usuários	Deve existir uma interface que permita pesquisar, criar, alterar e desativar usuários.	Essencial
[RF016] Interface	A interface padrão de acesso ao sistema é a <i>WEB</i> , devendo ser compatível com os navegadores mais utilizados (Internet Explorer, Mozilla Firefox, Google Chrome, Opera, dentre outros). Deve permitir ao paciente ter acesso a partir de sua própria residência.	Essencial
[RNF001] Segurança contra roubo de sessão do usuário	A sessão de comunicação deve possuir controles de segurança a fim de não permitir o roubo da sessão do usuário. Nota: O roubo de sessão pode ser possível inclusive em sessões protegidas (ex. SSL/TLS). Por exemplo, se o controle de sessão for realizado através de <i>cookie</i> na URL, em determinadas situações, a URL da sessão de um usuário pode ser obtida e utilizada por outro usuário, personificando o usuário anterior.	Importante
[RNF002] Independência de plataforma	O software deve ser desenvolvido utilizando uma linguagem livre e multi-plataforma, garantindo a compatibilidade com a maior quantidade de plataformas disponíveis.	Essencial

Requisitos do incremento 02 - prontuário do paciente

Esta seção descreve os requisitos funcionais (RF) e não-funcionais (RNF) referentes ao incremento 02. Cada requisito está identificado por uma chave única, composta pelo código que representa o tipo de requisito e um número sequencial.

Tabela 2.3: requisitos do incremento 02.

Requisito	Descrição	Prioridade
[RF017] Médicos	Deve existir uma interface que permita identificar os usuários que são médicos, incluindo para cada um deles o conselho a que estão associados, UF e número de registro.	Essencial
[RF018] Especialidades médicas	Deve existir um cadastro de especialidades médicas, de acordo com o qual o médico deve estar associado. Cada especialidade deve conter um nome, sociedade e site oficial. Deve seguir a Resolução CFM N° 1.763/05.	Essencial
[RF019] Queixas gerais	Deve existir um cadastro de queixas gerais com a lista de queixas gerais disponíveis. Pesquisa, criação, alteração e exclusão devem estar disponíveis. As queixas gerais devem permitir ser categorizados e devem conter código, nome e descrição.	Essencial
[RF020] Vacinas	Deve existir um cadastro de vacinas com a lista de vacinas disponíveis. Pesquisa, criação, alteração e exclusão devem estar disponíveis.	Essencial
[RF021] Eventos	Deve existir um cadastro de eventos com a lista de eventos disponíveis. Pesquisa, criação, alteração e exclusão devem estar disponíveis. Cada evento deve possuir a possibilidade de obter resultados em texto, imagem ou arquivo binário. Sua categorização deve ser feita através de tipos de eventos e categorias de eventos.	Essencial
[RF022] Prescrições	Deve existir um cadastro de prescrições associado ao prontuário do paciente. O médico deve conseguir visualizar as prescrições anteriores do paciente, como também criar novas prescrições. Depois de finalizada, uma prescrição não pode ser alterada.	Essencial

Continua na próxima página

Tabela 2.3 – continuação da página anterior.

Requisito	Descrição	Prioridade
[RF023] Doenças	Deve existir um cadastro de doenças com a lista de doenças existentes. Deve seguir a Classificação Internacional de Doenças (CID).	Essencial
[RF024] Imunizações	Deve ser possível cadastrar, alterar e excluir as imunizações de uma pessoa. A imunização deve estar relacionada com uma pessoa e uma vacina, contendo também a data provável e a data efetiva da aplicação.	Essencial
[RF025] Triagem do paciente	O RES deve possuir uma categoria de usuário responsável pela triagem do paciente. Este deve verificar a existência de cadastro, providenciando-o caso ainda não exista. Então, de acordo com as queixas gerais apresentadas pelo paciente, deve ser criada uma nova consulta com o médico adequado. É importante ressaltar que esta ação deve ser realizada por profissional capacitado, pois já faz parte da anamnese do paciente. Além disso, devem ser desenvolvidas ferramentas que permitam a identificação de registros duplicados.	Essencial
[RF026] Lista de espera	Deve existir uma busca de consultas abertas (ainda não atendidas), classificadas em ordem de chegada e agrupadas por médico, de forma que a ordem de chamada das consultas possa ser estabelecida. A partir do momento que o médico inicia a consulta, seu estado deve ser modificado automaticamente para que não apareça mais na lista de espera. Devem ser verificados os estados possíveis da consulta, criando-se um atributo para representá-los.	Essencial
[RF027] Consulta	Na tela da consulta, o médico deve ser capaz de visualizar as queixas gerais que motivaram a consulta, solicitar exames, procedimentos, aplicação de medicamentos e gerar prescrições médicas, além de obter as informações do prontuário do paciente. Cada consulta deve possuir um diagnóstico associado, informando a doença que o paciente possui.	Essencial

Continua na próxima página

Tabela 2.3 – continuação da página anterior.

Requisito	Descrição	Prioridade
[RF028] Dados do prontuário	O prontuário do paciente deve reunir todas as informações que podem auxiliar no diagnóstico do problema que levou o paciente ao atendimento. Deve conter, no mínimo: dados pessoais, endereços, profissão, doenças anteriores, histórico familiar, hábitos de vida, aspectos sócio-econômicos, alergias, imunizações e medicamentos em uso. O médico também deve poder acessar informações sobre qualquer exame, procedimento, medicação ou atividades afins que estejam disponíveis no RES.	Essencial
[RF029] Execução de eventos	Cada evento solicitado pelo médico deverá ser registrado através do prontuário do paciente. Antes de realizar o evento, o paciente terá seu prontuário acessado pelo profissional de saúde que verificará os exames e procedimentos necessários para, então, realizá-los. Os resultados devem ser incluídos nas respostas dos seus respectivos eventos e, conseqüentemente, no prontuário do paciente.	Essencial
[RF030] Mensagens	Todos os usuários devem conseguir comunicar-se com qualquer outro usuário do RES através de uma central de mensagens. Quando existir uma nova mensagem para um determinado usuário, deve ser emitido um alerta na página principal do prontuário, permitindo que o mesmo visualize a nova mensagem e efetue a resposta.	Essencial
[RF031] Apoio ao diagnóstico	O RES deve disponibilizar um sistema de <i>plugins</i> capaz de integrar sistemas de apoio ao diagnóstico, obtendo as informações através da leitura de arquivos no padrão DICOM e exibindo os resultados na interface padrão do software.	Importante
[RNF003] Usabilidade	As interfaces devem possuir usabilidade adequada de acordo com o público-alvo de usuários em questão. Deve-se observar a rotina dos profissionais de saúde com o objetivo de orientar o direcionamento do fluxo de atividades dentro do RES.	Essencial

Continua na próxima página

Tabela 2.3 – continuação da página anterior.

Requisito	Descrição	Prioridade
[RNF004] Padronização	Observar os padrões internacionais e nacionais no cadastro das entidades do sistema.	Essencial

2.3 Projeto de sistema

Nesta seção são descritas as etapas desenvolvidas durante a fase de planejamento e projeto do RES proposto, fornecendo as informações necessárias para gerenciar o processo de implementação do software. Assim, o risco de atraso na entrega ou até mesmo insucesso no desenvolvimento podem ser minimizados.

2.3.1 Plano de projeto

O plano de projeto é responsável por estabelecer os recursos disponíveis para o projeto, assim como a sua estrutura analítica e um cronograma para realização do trabalho. Através deste documento, é possível definir com detalhes os objetivos e marcos esperados no decorrer da execução do projeto, permitindo o seu acompanhamento e gerenciamento. Em geral, definem-se elementos como organização do trabalho, análise de riscos, requisitos de *hardware* e cronograma de projeto, dentre outros.

Com o objetivo de monitorar e gerenciar o desenvolvimento do RES proposto, as seções a seguir descrevem a execução dos elementos citados anteriormente.

Análise de riscos

Um risco pode ser qualquer fator que altere o fluxo de desenvolvimento normal de um projeto de software. A sua avaliação e monitoramento, assim como a tomada de ações preventivas, são de grande importância pois permitem evitar ou pelo menos minimizar o efeito que um risco efetivado pode causar.

Para o RES proposto, alguns riscos são identificados no início do planejamento, cuja descrição, probabilidade e efeito são mostrados na Tabela 2.4.

Risco	Probabilidade	Efeito
Desistência de algum membro da equipe.	Baixa	Catastrófico
Membros-chave estão doentes em épocas críticas do projeto.	Moderada	Sério
Componentes do software que deveriam ser reutilizados contêm defeitos que limitam suas funcionalidades.	Moderada	Sério
Mudanças nos requisitos que requerem grandes alterações no projeto são propostas.	Moderada	Sério
A base de dados utilizada na aplicação não consegue processar o número de transações por segundo esperado.	Moderada	Sério
O tempo necessário para desenvolver o software é subestimado.	Alta	Sério
A complexidade do software é subestimada.	Alta	Catastrófico
A curva de aprendizado das tecnologias utilizadas é maior do que a esperada.	Moderada	Sério

Tabela 2.4: riscos gerenciados durante o desenvolvimento do trabalho.

Recursos de *hardware*

Os recursos de *hardware* disponíveis para o desenvolvimento do RES proposto foram fornecidos pelo Laboratório de Engenharia de Sistemas de Computação e por membros da equipe de desenvolvimento. A seguir são descritos os equipamentos utilizados:

- computador *desktop* IBM Think Centre M52 (Pentium D 2.6GHz, 2GB RAM e 80GB HD);
- computador *desktop* Ibyte (Pentium D 3Ghz, 2GB RAM e 80GB HD);
- computador portátil Compaq Presario V6210BR (AMD Sempron 3500+, 1GB RAM e 60GB HD).

Cronograma de desenvolvimento

A definição e elaboração de um cronograma de desenvolvimento é um pré-requisito para obter sucesso no desenvolvimento de todo e qualquer software. A partir do cronograma podem ser definidos prazos e marcos de entrega, de forma que o processo de acompanhamento possa ser realizado. Além disso, os membros da equipe de desenvolvimento conhecem as atividades que devem ser desenvolvidas, assim como suas respectivas datas de início e finalização. O cronograma de desenvolvimento do RES proposto pode ser visualizado na Figura 2.2.

WBS	Name	Start	Finish	Work	Assigned to
1	▣ Fase I	jan 5	mar 19	60d	Igor
1.1	Revisão bibliográfica	jan 5	fev 10	30d	Igor
1.2	Avaliação crítica dos RES existentes	fev 10	mar 19	30d	Igor
2	▣ Fase II	mar 19	mai 14	80d	Igor
2.1	Levantamento e análise de requisitos	mar 19	abr 24	30d	Igor
2.2	Estudo de padrões de software e comunicação entre equipamentos médicos	abr 27	mai 14	25d	Igor
2.3	Estudo e avaliação de tecnologias baseadas em software livre e Internet	mar 19	abr 23	25d	Igor
3	▣ Fase III	mai 14	nov 3	164d	Igor
3.1	Elaboração de projeto e modelagem	mai 14	jun 19	25d	Igor
3.2	Implementação (Incremento 01)	mai 14	jul 15	50d	Igor
3.3	Verificação e validação (Incremento 01)	jul 15	jul 23	7d	Igor
3.4	Implementação (Incremento 02)	jul 24	out 26	75d	Igor
3.5	Verificação e validação (Incremento 02)	out 26	nov 3	7d	Igor
4	▣ Fase IV	jan 5	fev 1	316d	Igor
4.1	Escrita da dissertação	jan 5	fev 1	316d	Igor

Figura 2.2: cronograma de desenvolvimento do RES proposto.

2.3.2 Modelagem do incremento 01 - autenticação e autorização

Nas seções a seguir são apresentadas as técnicas de modelagem de sistemas utilizadas durante o desenvolvimento do incremento 01.

Modelagem de casos de uso

A modelagem de casos de uso permite entender conceitualmente o conjunto de funções que o sistema deve executar para atender os requisitos especificados, capturando e modelando os requisitos funcionais em uma linguagem muito mais acessível. É composta por um conjunto de diagramas e tabelas que definem cada caso de uso identificado nos diagramas.

Os diagramas de casos de uso referentes ao incremento 01 podem ser visualizados nas Figuras 2.3 e 2.4. As tabelas de especificação dos casos de uso identificados estão descritas no Apêndice B.

Modelagem de diagrama de classes

O diagrama de classes é o mais utilizado e mais importante da *Unified Modeling Language* (UML), pois permite visualizar as classes que compõem o sistema e como se relacionam entre si. É composto por dois elementos básicos: classe e relacionamento. As classes definem os métodos e atributos, enquanto os relacionamentos definem a quantidade de objetos que participam de cada relacionamento. Existem vários tipos de relacionamento: associação, generalização,

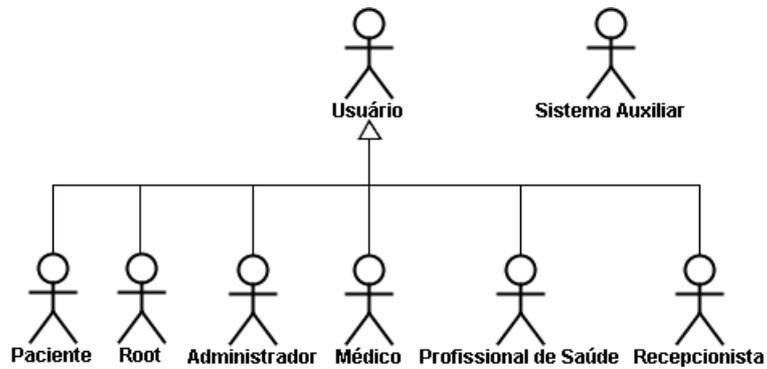


Figura 2.3: diagrama de atores identificados no incremento 01.

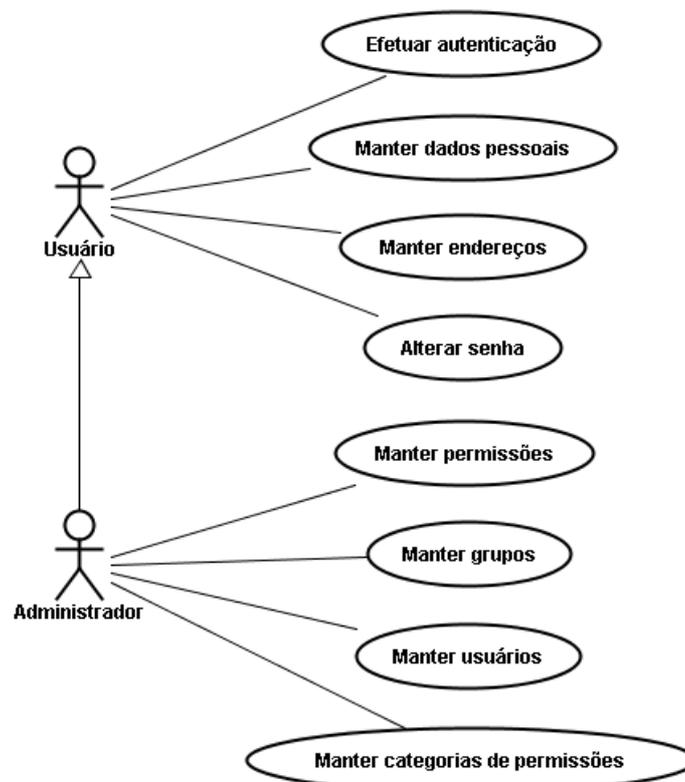


Figura 2.4: diagrama de casos de uso identificados no incremento 01.

dependência e realização. Cada relacionamento representa um conceito distinto da programação orientada a objetos. O relacionamento de generalização, por exemplo, é utilizado para representar o conceito de herança.

As classes de entidades desenvolvidas no incremento 01 podem ser visualizadas na Figura 2.5. Não foram incluídos os atributos e métodos, pois o objetivo no momento é apenas apresentar a estrutura geral e os relacionamentos entre as classes.

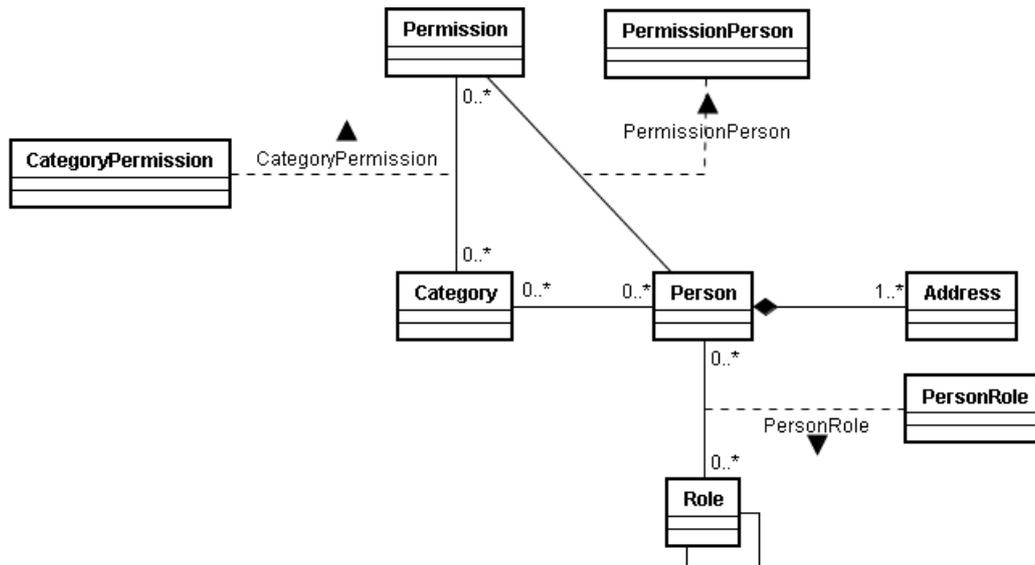


Figura 2.5: diagrama de classes de entidades persistentes identificadas no incremento 01.

Modelagem de diagrama de atividade

O diagrama de atividade é o modelo com maior ênfase no nível de algoritmo da *Unified Modeling Language* (UML), preocupando-se em descrever os passos a serem percorridos para a conclusão de um método ou algoritmo específico. Por esta razão, é utilizado para descrever com detalhes as atividades cujos passos de execução necessitem de uma descrição mais detalhada. É composto pelos elementos a seguir: estado inicial, estado final, transições, estado de ação, ponto de decisão, barra de sincronização, estado de subatividade, fluxo de objetos e raias de natação.

Para o incremento 01, foram elaborados dois diagramas de atividade. O primeiro diagrama, mostrado na Figura 2.6, representa os passos de execução necessários para efetuar a autenticação. O segundo diagrama, apresentado na Figura 2.7, representa os passos de execução necessários para alterar a senha.

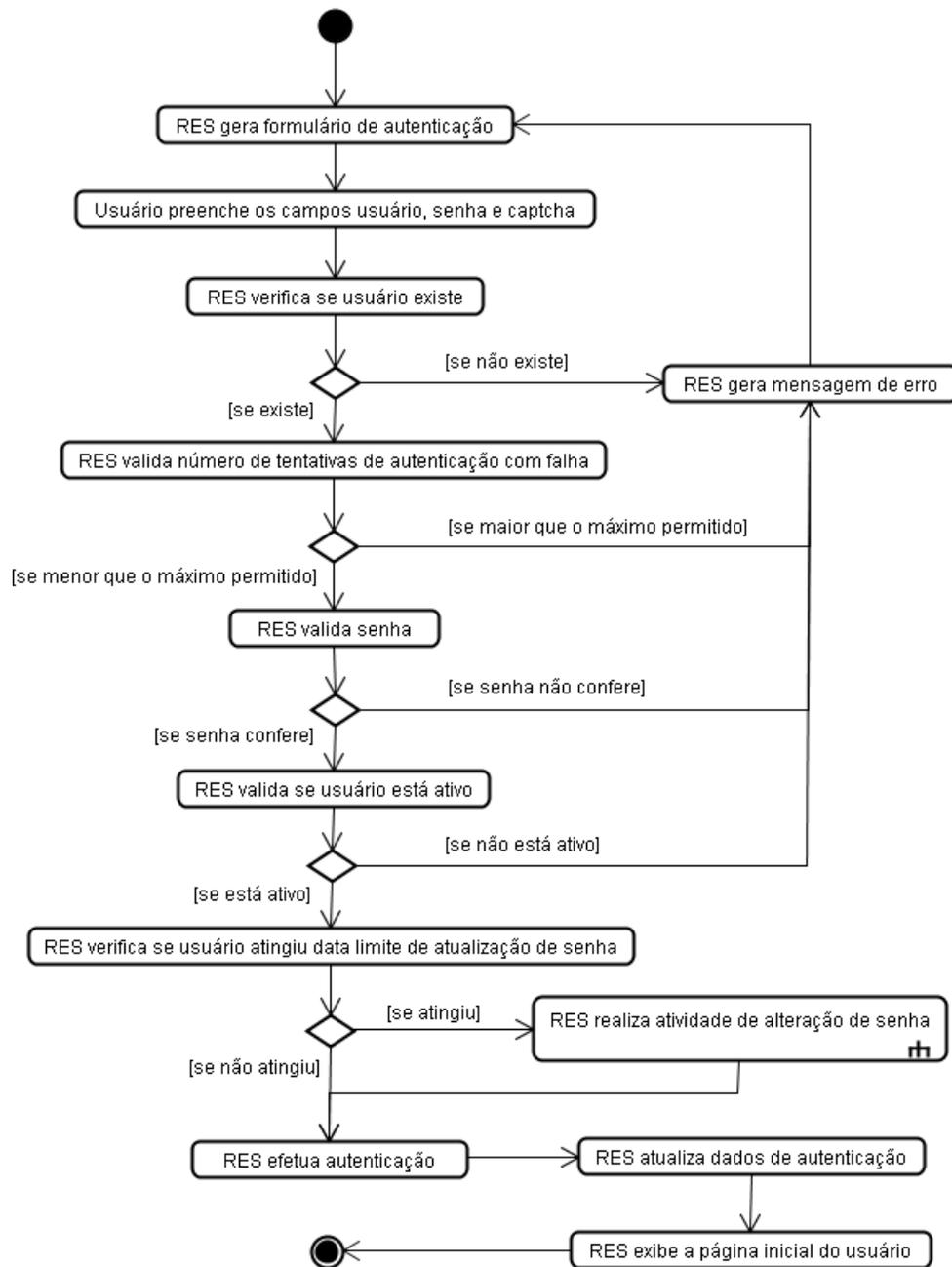


Figura 2.6: diagrama de atividade da autenticação.

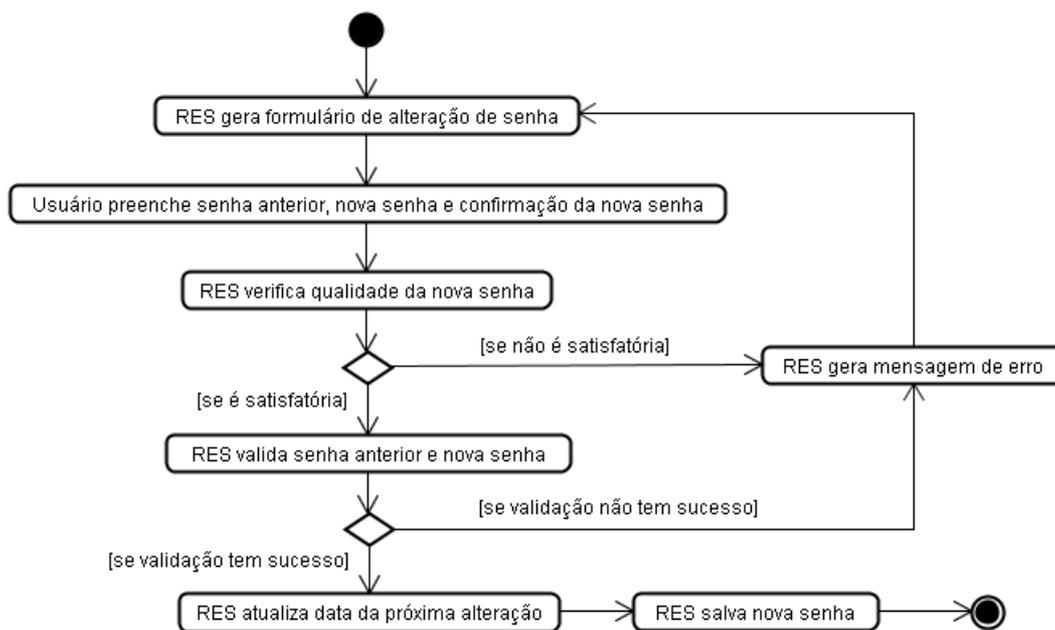


Figura 2.7: diagrama de atividade da alteração de senha.

2.3.3 Modelagem do incremento 02 - prontuário do paciente

Nas seções a seguir são apresentadas as técnicas de modelagem de sistemas utilizadas durante o desenvolvimento do incremento 02.

Modelagem de casos de uso

Neste incremento foram definidos os casos de uso de cada um dos atores identificados anteriormente. Os diagramas de caso de uso são mostrados nas Figuras 2.8 a 2.12 e as suas respectivas Tabelas de Especificação podem ser encontradas no Apêndice B.

Modelagem de diagrama de classes

O diagrama de classes referente ao incremento 02 é mostrado na Figura 2.13. Os atributos e métodos das classes não foram incluídos por se tratar apenas da visão estrutural do ambiente e seus relacionamentos.

Modelagem de diagrama de atividade

Para o incremento 02, apenas um diagrama de atividade é criado, sendo mostrado na Figura 2.14, que representa os passos necessários para executar os procedimentos de auxílio ao diagnóstico.

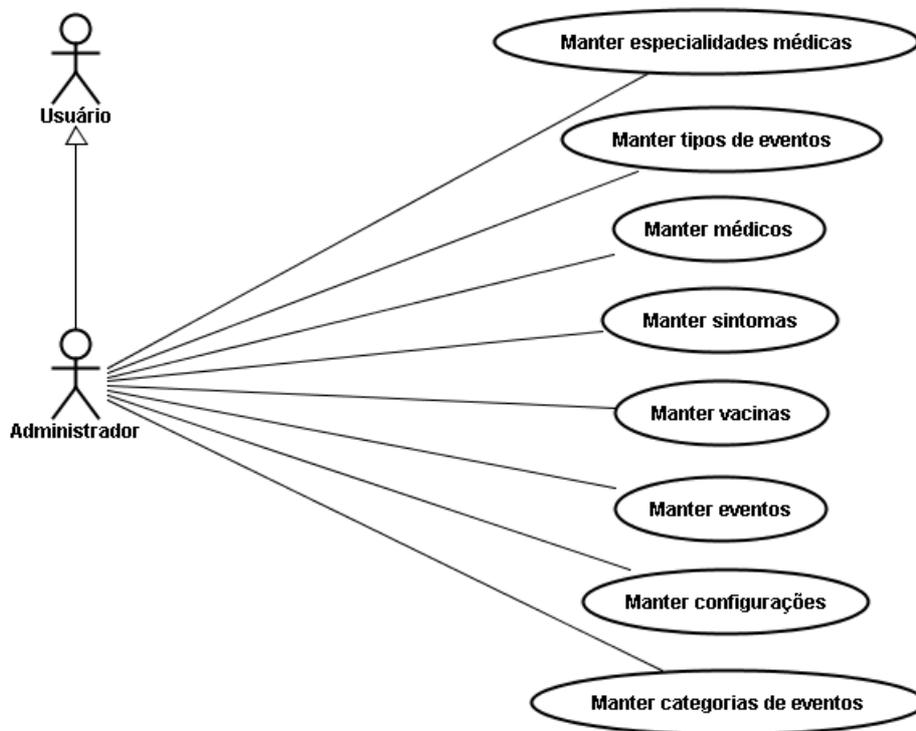


Figura 2.8: diagrama de casos de uso do administrador.

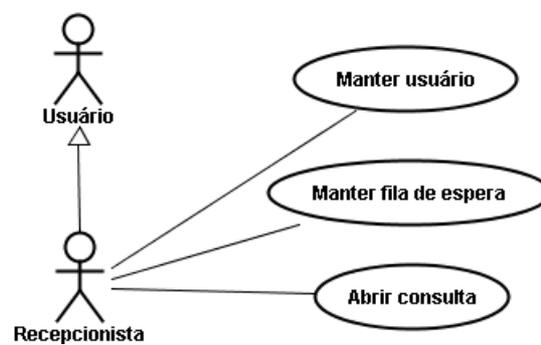


Figura 2.9: diagrama de casos de uso do recepcionista.

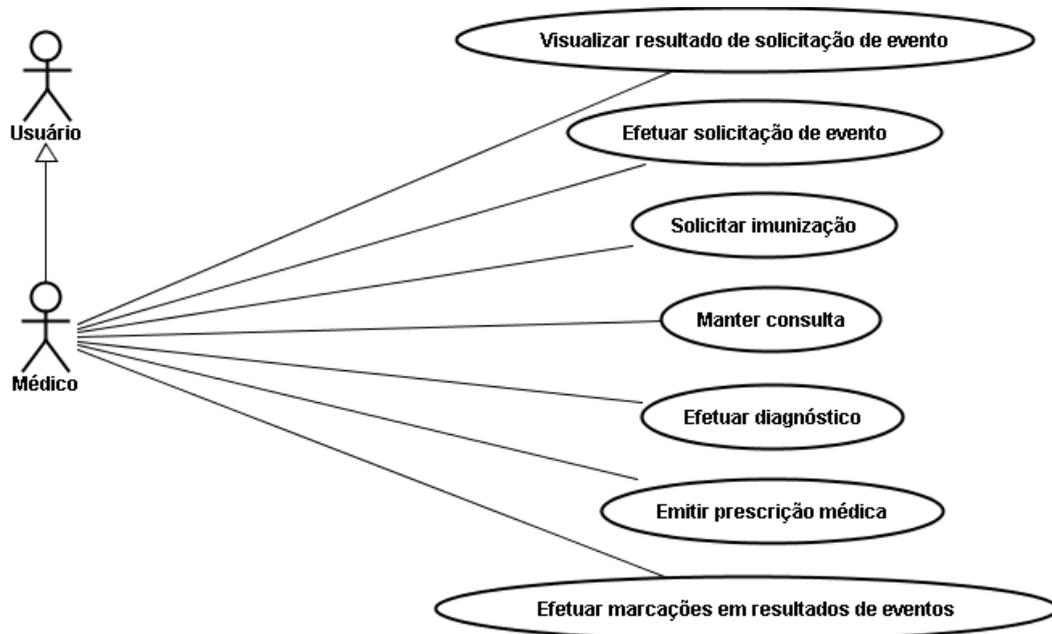


Figura 2.10: diagrama de casos de uso do médico.

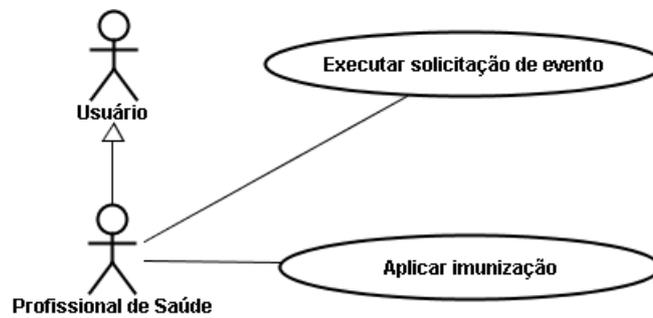


Figura 2.11: diagrama de casos de uso do profissional de saúde.

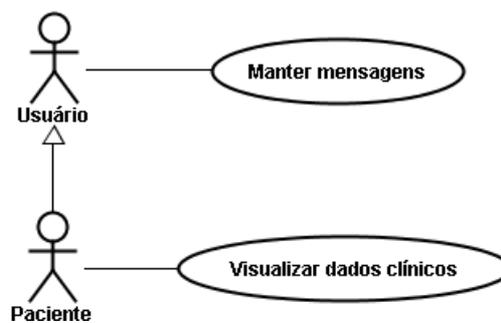


Figura 2.12: diagrama de casos de uso do paciente e do usuário.

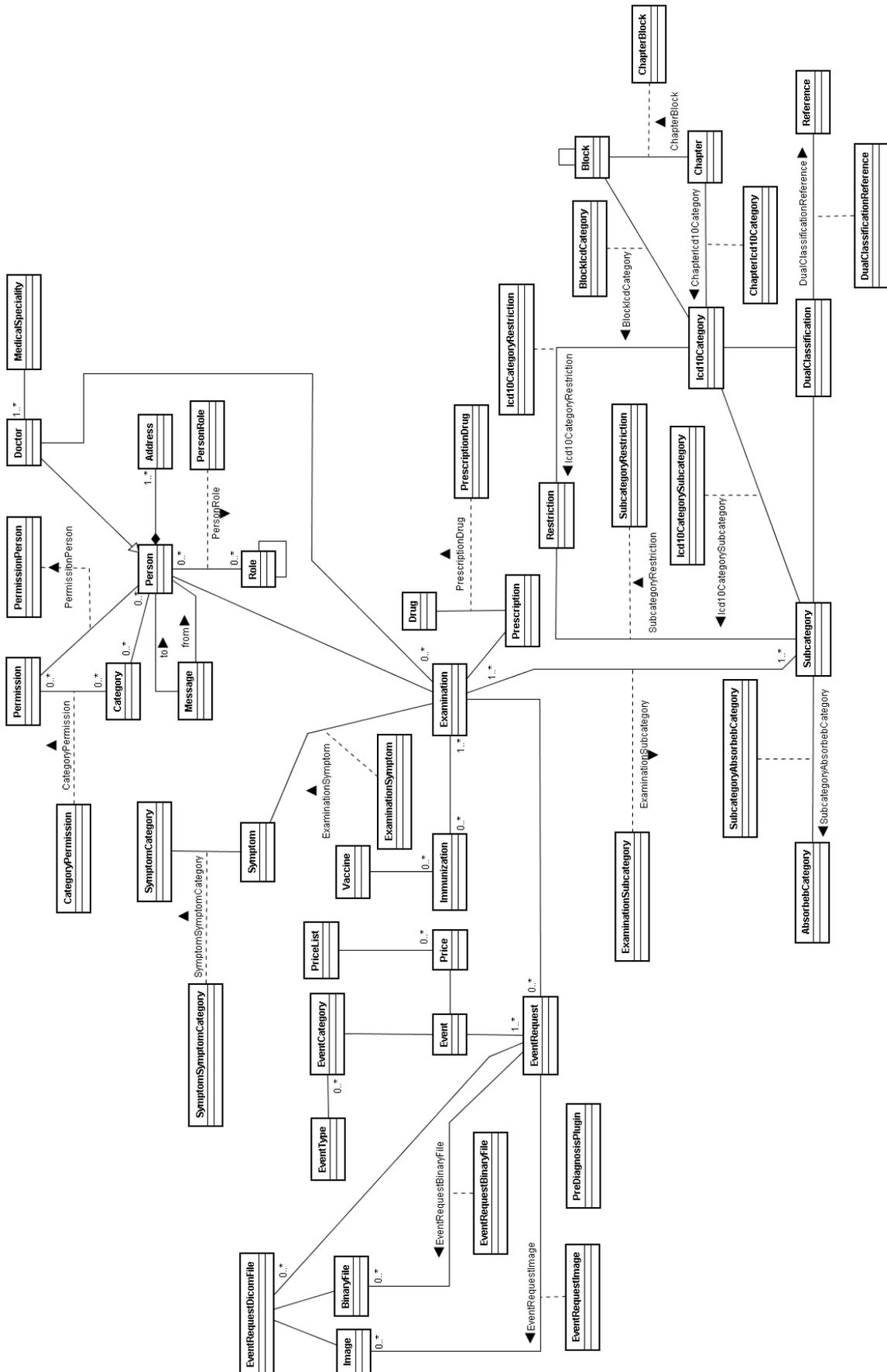


Figura 2.13: diagrama de classes de entidades persistentes identificadas no incremento 02.

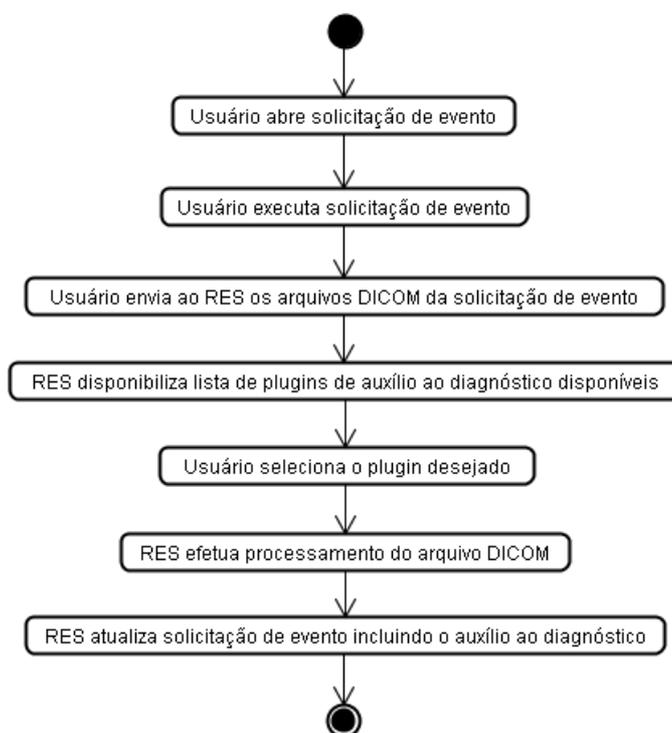


Figura 2.14: diagrama de atividades do *plugin* de apoio ao diagnóstico.

2.3.4 Arquitetura do RES proposto

Como é possível observar na Figura 2.15, o RES proposto apresenta uma estrutura multicamadas, conforme especificado no padrão *Model-View-Controller*. Este padrão realiza a separação entre a camada de visualização e a camada de dados. Assim, alterações efetuadas na camada de visualização não afetam a manipulação de dados, desde que as mesmas fontes de dados tenham sido preservadas. Da mesma forma, alterações nos dados não implicam em alterações na camada de visualização. Este padrão é uma técnica consagrada e que se adapta perfeitamente aos requisitos do RES proposto, motivo pelo qual é adotado.

Além da estrutura multicamadas, a arquitetura do RES proposto apresenta um mecanismo de comunicação com a *Internet* e com sistemas auxiliares. A comunicação com a *Internet* permite que qualquer dispositivo que possua um navegador compatível com as tecnologias utilizadas tenha acesso aos recursos disponíveis, enquanto a integração com sistemas auxiliares permite a execução de *plugins* de apoio ao diagnóstico e troca de informações diversas. A arquitetura utilizada permite fácil evolução do software desenvolvido, fazendo com que este se adapte às mais diversas mudanças no cenário das instituições de saúde.

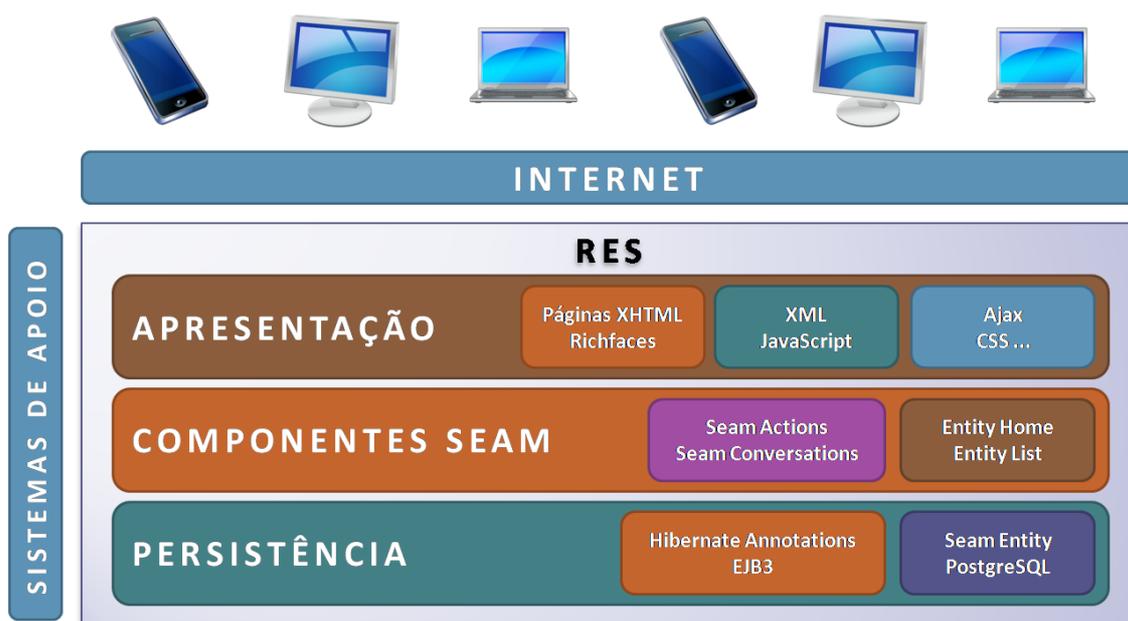


Figura 2.15: arquitetura do RES proposto.

2.4 Implementação do sistema

A modelagem permite a compreensão do grupo de funcionalidades que devem estar disponíveis no software antes de iniciar o desenvolvimento e, assim, fornece o embasamento conceitual necessário para a escolha das tecnologias necessárias para atender os requisitos especificados.

A escolha equivocada de uma determinada tecnologia ou ferramenta pode causar grande impacto negativo, gerando atrasos na entrega e crescimento do custo de desenvolvimento. Por esta razão, este processo necessita de grande estudo e pesquisa com profissionais da área, fóruns e listas de discussão especializadas, além do estudo detalhado das funcionalidades de cada ferramenta em seus respectivos *websites* oficiais.

Baseado nos requisitos especificados e na natureza do software proposto, os pontos-chave observados durante o estudo das tecnologias são: segurança, estabilidade, compatibilidade e produtividade. Por se tratar de um sistema em que as informações só devem estar acessíveis a quem é de direito, o cuidado com a segurança é o primeiro item a ser avaliado. A estabilidade vem em seguida, pois um sistema para este fim deve ser necessariamente estável para estar disponível 24 horas por dia, 7 dias por semana.

A compatibilidade é o próximo item, pois um dos pré-requisitos do RES proposto é ser multiplataforma. A produtividade é, sem dúvida, um item também muito importante. A velocidade e a facilidade de incluir novas funcionalidades e manter as funcionalidades já existentes é muito importante para a continuidade do RES e para a satisfação dos usuários finais, que podem ter as necessidades atendidas mais rapidamente.

Com o objetivo de compreender a necessidade de cada tecnologia e justificar a sua escolha, as próximas seções descrevem cada tipo de tecnologia utilizada no desenvolvimento do RES proposto.

2.4.1 Sistema operacional

Um dos requisitos do RES proposto é a utilização de tecnologias livres em todo o processo, incluindo não apenas o seu período de desenvolvimento, mas também todo o período de utilização.

Por este motivo, o sistema operacional utilizado para executar o software também deve seguir esta mesma orientação. O sistema operacional utilizado para hospedar e executar os serviços desenvolvidos é o *Ubuntu* (UBUNTU, 2009), sistema operacional baseado no *Debian* (DEBIAN, 2009), reconhecidamente de fácil utilização até mesmo para usuários com pouca experiência em sistemas operacionais baseados em *GNU/Linux*. Sua escolha é motivada por atender os requisitos de hospedagem do servidor e também pela facilidade de utilização nos clientes.

2.4.2 Linguagem de programação

Um requisito-chave do RES proposto é ser livre e multiplataforma. Para tanto, faz-se necessário que o software seja desenvolvido utilizando uma linguagem de programação livre e portátil, permitindo a sua execução nos mais diversos ambientes, sejam livres ou até mesmo proprietários. Portanto, entende-se que para satisfazer este requisito sem alterações de código-fonte específicas para cada plataforma, a linguagem deve ser interpretada e não compilada.

Do grupo de linguagens interpretadas existentes, uma tem se destacado no cenário nacional e internacional: a linguagem Java (JAVA, 2009). Facilidade de desenvolvimento, eficiência, robustez, orientação a objetos, comunidade ativa, livre e multiplataforma são algumas das características desta linguagem, além de um grande número de *frameworks* disponíveis para os mais diversos tipos de aplicação.

Baseado nestas considerações e na experiência anterior já existente no time de desenvolvimento deste software, a linguagem Java é a escolhida como linguagem base de desenvolvimento.

2.4.3 Ferramenta de modelagem

A modelagem é uma das etapas mais importantes de todo e qualquer processo de desenvolvimento de software. Nesta fase, são coletadas e especificadas as funcionalidades que devem ser disponibilizadas ao final do desenvolvimento.

Existem diversas ferramentas de modelagem de sistemas baseadas na linguagem de modelagem padrão, a *Unified Modeling Language* (UML). Dentre estas, uma ferramenta tem sido utilizada frequentemente pelas empresas e universidades por possuir características como portabilidade e uma versão gratuita. Esta ferramenta é conhecida por *Astah Community* (ASTAH, 2010) e disponibiliza todas as funcionalidades necessárias para a modelagem do RES proposto: diagramas de caso de uso, diagramas de classes e diagramas de atividade.

2.4.4 *Framework*

Um *framework* de software consiste em um conjunto de códigos-fonte desenvolvidos em uma linguagem de programação específica, que disponibilizam um grupo de funcionalidades genéricas e comuns a vários tipos de projetos. Funcionalidades como autenticação, autorização, gerenciamento de transações, dentre outras, são funcionalidades comuns e utilizadas em vários tipos de projetos de software. Por esta razão, a partir do momento que estas funcionalidades estão disponíveis através da utilização de um *framework*, a sua adoção se torna mais interessante do que o desenvolvimento de todas as funcionalidades necessárias. Esta prática permite o reuso de software e contribui para a manutenção dos custos de desenvolvimento na menor faixa possível.

O RES proposto tem como requisito o funcionamento em interface *WEB*. Por este motivo, faz-se necessário pesquisar o estado-da-arte dos *frameworks* desenvolvidos na linguagem de programação Java que disponibilizam funcionalidades para a interface *WEB*.

Através de consultas a especialistas no assunto em fóruns de discussão especializados e listas de e-mail, foi possível reunir um grupo de *frameworks* que poderiam ser utilizados. Baseado nos aspectos de avaliação definidos anteriormente,

o *framework JBoss Seam* (SEAM, 2009) mostrou-se o mais adequado por ser moderno, aderente aos padrões de desenvolvimento Java, facilmente adaptável e por disponibilizar várias funcionalidades necessárias ao RES proposto. Várias tecnologias de ponta estão incorporadas na utilização deste *framework*, como a especificação *EJB3* (EJB3, 2009), *Ajax*, as bibliotecas *Hibernate* (HIBERNATE, 2009), *Hibernate Annotations* (ANNOTATIONS, 2009) e *Richfaces* (RICHFACES, 2009), dentre outras.

Estas tecnologias são utilizadas na persistência de dados e no desenvolvimento de componentes de interação com o usuário através da *WEB*, permitindo uma experiência rica e facilitando a usabilidade do usuário final.

2.4.5 Ambiente de desenvolvimento

Existem dois ambientes de desenvolvimento que são largamente utilizados no desenvolvimento de software através da tecnologia Java: o *NetBeans* (NETBEANS, 2009) e o *Eclipse* (ECLIPSE, 2009). Ambos são compatíveis com o *framework* escolhido, porém o ambiente de desenvolvimento *Eclipse* possui um *plugin* denominado *JBossTools* (JBOSSTOOLS, 2009), utilizado para desenvolver projetos com o *framework JBoss Seam*.

Este *plugin* disponibiliza interfaces que permitem a criação e configuração de projetos utilizando o *framework JBoss Seam* de uma forma simples e intuitiva, contribuindo de forma significativa para o aumento da produtividade. Por esta razão, o ambiente de desenvolvimento *Eclipse* é o escolhido.

2.4.6 Servidor de aplicação

Um servidor de aplicação é um software que disponibiliza um ambiente para a instalação e execução de aplicações. Seu objetivo é disponibilizar uma plataforma que abstraia do desenvolvedor de software algumas das complexidades de um sistema computacional.

No desenvolvimento de aplicações comerciais, por exemplo, o foco dos desenvolvedores deve ser a resolução de problemas relacionados ao negócio da empresa, e não questões de infraestrutura da aplicação. O servidor de aplicações responde à algumas questões comuns a todas as aplicações, como segurança, garantia de disponibilidade, balanceamento de carga e tratamento de exceções.

Baseado na natureza do RES proposto e nas tecnologias utilizadas no seu

desenvolvimento, a adoção de um servidor de aplicação traz benefícios que contribuem para a produtividade e estabilidade do software. Dentre os servidores de aplicação compatíveis com o *framework* escolhido, o *JBoss AS* (JBASSAS, 2009) mostrou-se a melhor escolha por sua fácil integração e estabilidade reconhecida, além de ser desenvolvido pelo mesmo fornecedor do *framework* adotado, o que garante a total compatibilidade. Além destes fatores, a experiência anterior do time de desenvolvimento com esta tecnologia contribuiu para a escolha.

2.4.7 Banco de dados

O banco de dados é a estrutura responsável por armazenar todas as informações que não podem ser perdidas após a finalização de uma sessão ou desligamento do sistema. Atualmente, não é comum o armazenamento através de arquivos e protocolos próprios, a não ser em casos específicos. O cenário mais comum nos dias de hoje é a utilização de um Sistema Gerenciador de Banco de Dados (SGBD), responsável por gerenciar os dados armazenados e disponibilizar uma interface de consulta, alteração, criação e exclusão.

O *Mysql* (MYSQL, 2009) e o *PostgreSQL* (POSTGRESQL, 2009) são exemplos de SGBDs gratuitos que utilizam o *Structured Query Language* (SQL) como linguagem padrão de acesso aos dados. Porém, no atual estágio de desenvolvimento de software e com a utilização de tecnologias de mapeamento entre classes e entidades de bancos de dados, não é mais necessário escrever sentenças SQL para realizar as operações citadas.

A especificação *EJB3* e a tecnologia *Hibernate Annotations*, disponíveis no framework *JBoss Seam*, permitem realizar o mapeamento das classes que representam entidades e que devem ter seus dados armazenados no banco de dados através da inclusão de anotações no código-fonte. Esta nova abordagem facilita o desenvolvimento de software e diminui o custo de implementação e evolução do software, visto que a alteração de atributos nas classes é automaticamente convertida para o banco de dados.

Além disso, estas tecnologias incluem uma camada de transparência que permite a utilização de vários SGBDs sem a necessidade de alteração do código-fonte, ou seja, a substituição de um SGBD por outro pode ser realizada através de uma mera configuração, não sendo necessário alterar nenhum código-fonte.

Para o RES proposto, foram realizados testes com os SGBDs *Mysql* e

PostgreSQL, tendo como resultado o funcionamento pleno e adequado em ambos. Desta forma, pode-se afirmar que o RES proposto faz uso do estado-da-arte na camada de persistência de dados, podendo ser facilmente portado para qualquer SGBD compatível com a especificação *EJB3* e a tecnologia *Hibernate Annotations*.

2.5 Verificação e validação

2.5.1 Verificação de requisitos

A verificação de requisitos compreende a atividade de análise dos requisitos e as funcionalidades desenvolvidas, ou seja, é um processo interno no qual é verificado se as funcionalidades desenvolvidas estão de acordo com a especificação dos requisitos.

No RES proposto, esta atividade é desempenhada pela própria equipe responsável pelo desenvolvimento, tendo sido realizada durante todas as etapas de produção do software.

2.5.2 Validação de requisitos

A validação de requisitos é um processo externo que compreende a apresentação das funcionalidades desenvolvidas para o cliente, no qual este informa se as funcionalidades foram desenvolvidas conforme sua expectativa.

No RES proposto, esta atividade é desempenhada pelo orientador e co-orientador deste trabalho, tendo sido realizada ao final do desenvolvimento de cada incremento.

2.6 Demonstração e testes

2.6.1 Página oficial

Com o objetivo de disponibilizar todas as informações necessárias sobre o RES proposto e criar um canal de comunicação oficial com os colaboradores, uma conta foi criada no portal de hospedagem de softwares livres mais conhecido em todo o mundo, o *Source Forge*.

Através deste canal, várias operações podem ser realizadas:

- visualização das últimas informações a respeito do desenvolvimento;
- comunicação direta com a equipe de desenvolvimento;

- *download* de código-fonte e documentação;
- fóruns de discussão;
- lista de e-mails, dentre outras.

O endereço oficial do RES proposto no *Source Forge* é <https://sourceforge.net/projects/res-ufc>.

2.6.2 Versão de demonstração e teste

Uma das grandes dificuldades encontradas durante a fase de avaliação dos sistemas de registro eletrônico de saúde existentes foi a falta de versões *on-line* de demonstração e teste. Para que o RES proposto não seja incluído nesta lista, está disponível uma versão de demonstração e teste que pode ser acessada por qualquer dispositivo conectado a Internet através do endereço <http://res.lesc.ufc.br/demo>.

Os comentários, sugestões e críticas podem ser incluídos diretamente na página oficial, disponível no *Source Forge*.

Capítulo 3

As funcionalidades e interfaces do RES

3.1 Introdução

3.1.1 Visão geral

Neste Capítulo são apresentados os resultados obtidos ao final do desenvolvimento do RES proposto. Para tanto, a abordagem utilizada consiste da captura de um conjunto de imagens obtidas durante o seu funcionamento, com o objetivo de demonstrar seus componentes e também todas as funcionalidades que são consideradas inovações em relação aos sistemas disponíveis atualmente.

Inicialmente, são apresentadas as funcionalidades de internacionalização, autenticação e autorização, responsáveis por garantir a confidencialidade e segurança das informações através de uma estrutura completa de gerenciamento de permissões. Em seguida, são apresentadas as funcionalidades referentes ao gerenciamento de dados, responsáveis por manter as informações que servem de base para o funcionamento do RES proposto. Posteriormente, é apresentado um fluxo de atendimento completo, iniciado pela autenticação do usuário responsável pela recepção do paciente e encerrado com a identificação e registro do diagnóstico médico, incluindo todas as funcionalidades desenvolvidas para as etapas intermediárias deste processo. Para finalizar, são apresentadas as funcionalidades disponíveis na interface de acesso do paciente, de forma que este possa atualizar suas informações pessoais e visualizar suas informações clínicas, além de estabelecer

um canal de comunicação com os usuários do RES proposto.

3.1.2 Internacionalização

O termo internacionalização, em informática, corresponde à execução de processos de desenvolvimento e adaptação de um software para a língua e cultura de um país. A camada de apresentação do RES proposto é projetada para permitir a internacionalização, de forma que os textos apresentados ao usuário possam ser traduzidos para vários idiomas. O processo de inclusão de um novo idioma pode ser realizado facilmente, sendo compreendido pela criação de um novo arquivo contendo a tradução dos elementos-chave para o novo idioma. Esta funcionalidade permite que estrangeiros que fizeram uso dos serviços de saúde no Brasil possam ter acesso às informações armazenadas nos seus idiomas nativos.

A utilização desta funcionalidade pode ser visualizada nas Figuras 3.1 e 3.2, que contêm a página de autenticação do RES proposto nos dois idiomas disponíveis nesta versão: português brasileiro e inglês.

HUWC.UFC [Entrar](#)

Autenticação

Nome de Usuário

Senha

13 + 33 =

[Entrar](#)

português (Brasil) [Alterar Idioma](#)

Figura 3.1: página de autenticação no idioma português brasileiro.

HUWC.UFC [Login](#)

Authentication

Username

Password

24 + 27 =

[Login](#)

English [Change Language](#)

Figura 3.2: página de autenticação no idioma inglês.

3.2 Autenticação e autorização

O módulo de autenticação e autorização desenvolvido para o RES proposto é baseado em permissões, categorias de permissões e usuários. Uma permissão define o direito de utilizar uma funcionalidade de um determinado componente, enquanto uma categoria de permissões define um grupo de permissões que fazem parte de um mesmo contexto de usuário. Assim, ao invés de atribuir permissões específicas para um determinado usuário, faz-se necessário apenas informar as categorias que o usuário está associado, sendo a importação de suas permissões realizada automaticamente. Além disso, a alteração das permissões que compõem uma categoria de permissões é automaticamente convertida para o usuário, garantindo que os direitos de acesso e utilização estejam sempre atualizados.

Desta forma, qualquer acesso ao RES proposto implica em verificar se o usuário em questão está autenticado corretamente e possui permissões suficientes para acessar o recurso desejado. Desta forma, garante-se a segurança e o sigilo das informações de todos os usuários.

Para facilitar a compreensão da capacidade disponibilizada por este módulo, as seções a seguir descrevem como as permissões podem ser gerenciadas e utilizadas para disponibilizar interfaces diferenciadas.

3.2.1 Interfaces diferenciadas

Bloquear um usuário que faz uma tentativa de acesso a uma funcionalidade que não possui permissão é uma técnica que garante a segurança, mas não a usabilidade. Disponibilizar todas as opções de acesso e efetuar o bloqueio quando necessário não é uma técnica benéfica para a experiência de utilização do usuário. Assim, faz-se necessário que sejam apresentadas somente as funcionalidades que o usuário tem direito de acesso.

O RES proposto não apenas disponibiliza esta funcionalidade, como também permite realizar a personalização da interface *WEB* de acordo com a categoria de permissões que o usuário está associado. Através desta funcionalidade, pode-se ter um ganho de produtividade, pois tentativas de acesso a funcionalidades não permitidas não são realizadas por erros de interface, além de que o usuário pode ser facilmente identificado através da utilização de padrões de cores diferenciados. Estas funcionalidades podem ser visualizadas nas Figuras 3.3 a 3.5.

Seja bem-vindo(a), paciente3!
 Você tem mensagens não lidas (1). [Visualizar](#)

Perfil do paciente

Imunizações

Vacina ↕	Data Provável ↕	Data Efetiva ↕	Ação
VACINA INFLUENZA (ATENUADA)	01/01/2010		Visualizar

português (Brasil) [Alterar Idioma](#)

Figura 3.3: página inicial do paciente.

Seja bem-vindo(a), medico3!

Perfil do médico

Últimas consultas

Pessoa ↕	CPF ↕	Solicitado ↕	Situação ↕	Ação
[Redacted]	[Redacted]	26/01/2010	EM CURSO	Visualizar Editar

Perfil do paciente

Imunizações

Não há nenhuma imunização cadastrada para esta pessoa.

português (Brasil) [Alterar Idioma](#)

Figura 3.4: página inicial do médico.

Seja bem-vindo(a), root!

Perfil do paciente

Imunizações

Não há nenhuma imunização cadastrada para esta pessoa.

português (Brasil) [Alterar Idioma](#)

Figura 3.5: página inicial do administrador.

3.2.2 Permissões

As páginas de pesquisa e edição de permissões são mostradas nas Figuras 3.6 e 3.7. Através destas funcionalidades, o administrador pode realizar as atividades de pesquisa, cadastro, alteração e exclusão de permissões.

Permissões - Filtro de Pesquisa

Código

Alvo

Ação

Descrição

Pesquisa Reiniciar Cadastrar Permissão

Resultados da Pesquisa (101)

Código ↕	Alvo ↕	Ação ↕	Descrição ↕	Ação
1	configHome	create	Configuração: criar	Visualizar Editar
2	configHome	update	Configuração: atualizar	Visualizar Editar
3	configHome	delete	Configuração: excluir	Visualizar Editar
4	configList	search	Configuração: pesquisar	Visualizar Editar
5	configList	viewForm	Configuração: pesquisar (ver formulário)	Visualizar Editar

[Próxima Página >](#) [Última Página >>](#)

Figura 3.6: página de pesquisa de permissões.

Editar Permissão

Alvo *

Ação *

Descrição *

* Campos Obrigatórios

Salvar Alterações Excluir Cancelar Editar

Figura 3.7: página de edição de permissões.

3.2.3 Categorias

As páginas de pesquisa e edição de categorias de permissões podem ser visualizadas nas Figuras 3.8 e 3.9. Através destas funcionalidades, o administrador pode realizar as atividades de pesquisa, cadastro, alteração e exclusão de categorias de permissões.



Figura 3.8: página de pesquisa de categorias.



Figura 3.9: página de edição de categorias.

3.2.4 Grupos

O RES proposto também possui a opção de realizar a classificação dos usuários em grupos, embora esta não cause nenhuma implicação no gerenciamento de permissões. Em outras palavras, a organização dos usuários em grupos é meramente classificatória, não influenciando no gerenciamento de permissões. Esta classificação permite organizar os usuários em grupos de maior abrangência, de acordo com a necessidade da instituição de saúde.

Um aspecto importante desta classificação é que um grupo pode fazer parte de outro grupo, ou seja, podem haver subgrupos que são incluídos em grupos mais abrangentes. As Figuras 3.10 e 3.11 permitem visualizar as páginas de pesquisa e edição de grupos, nas quais podem ser realizadas as atividades de pesquisa, cadastro, alteração e exclusão.



Figura 3.10: página de pesquisa de grupos.



Figura 3.11: página de edição de grupos.

3.3 Gerenciamento de dados

Para permitir a utilização de um Registro Eletrônico de Saúde de forma adequada, faz-se necessário que exista um banco de dados de pessoas, especialidades médicas, médicos, queixas gerais, eventos, vacinas e configurações, dentre outros. No RES proposto, todas estas informações são gerenciáveis e podem ser alteradas, excluídas ou incluídas sempre que necessário, conforme pode ser visualizado nas seções a seguir. Esta é uma flexibilidade que garante a manutenção do RES proposto.

3.3.1 Pessoas

No RES proposto, todos os usuários têm domínio sobre os seus dados pessoais, podendo realizar alterações sempre que necessário. Porém, existe uma permissão especial para que um usuário possa alterar os dados pessoais de outros usuários.

Esta permissão disponibiliza a ativação ou desativação de um usuário, como também a alteração de grupos e categorias de permissões a que este usuário pertence. As Figuras 3.12 e 3.13 permitem visualizar tais funcionalidades.

Pessoas - Filtro de Pesquisa

Nome Completo

RG

CPF

Ativo

Pesquisa Reiniciar Cadastrar Pessoa

Resultados da Pesquisa (10)

Código	Nome Completo	Nome do Pai	Nome da Mãe	Ação
1			NOME DA MÃE	Visualizar Editar
2			NOME DA MÃE	Visualizar Editar
3			NOME DA MÃE	Visualizar Editar
4			NOME DA MÃE	Visualizar Editar
5			NOME DA MÃE	Visualizar Editar

[Próxima Página >](#) [Última Página >>](#)

Figura 3.12: página de pesquisa de pessoas.

3.3.2 Especialidades médicas

Cada médico deve estar associado a uma especialidade médica. Para tanto, deve existir um cadastro de especialidades médicas que permita realizar a associação com os médicos cadastrados no RES proposto. As Figuras 3.14 e 3.15 mostram as páginas de pesquisa e edição de especialidades médicas, nas quais o administrador pode realizar as atividades de pesquisa, cadastro, alteração e exclusão.

3.3.3 Médicos

Para efetuar o cadastro de um médico no RES proposto, necessita-se inicialmente que este seja cadastrado como um usuário padrão, de forma que possa ser reconhecido também como um paciente. Em seguida, o administrador deve incluir a categoria de permissões referente aos médicos no seu cadastro, para que este receba todas as permissões necessárias para realizar o atendimento dos seus pacientes. Para finalizar, deve ser criado um cadastro de médico utilizando o usuário já cadastrado, de forma que possam ser incluídos os dados referentes à prática da medicina, como número de cadastro no conselho ao qual está associado, assim como o próprio nome

Autenticado como: ROOT (root)

Editar Pessoa

Dados Pessoais | Endereços | Consultas | Solicitações de Eventos | Imunizações | Diagnósticos

Nome Completo *

Nome de Usuário *

Nome do Pai

Nome da Mãe *

CHS da Mãe

Sexo *

Data de Nascimento *

E-mail

Naturalidade

Data de Naturalização

Registro de Naturalização

Data de Entrada no Brasil

Estado de Nascimento *

Nacionalidade

RG

Emissor do RG

CPF

Estudante *

Telefone Residencial

Telefone Celular

Telefone Comercial

Telefone de Contato

Nome da Pessoa de Contato

Religião

Tipo Sanguíneo

Escolaridade *

Situação Familiar *

Raça *

Ativar

Grupos

Todos os grupos

Nome
GRUPO 01
GRUPO 02
GRUPO 03
GRUPO 04
GRUPO 05

Grupos

Nome

Categorias

Todas as categorias

Nome
ADMINISTRADOR
MÉDICO
RECEPCIONISTA
ENFERMEIRO

Categorias

Nome
PACIENTE
ROOT

* Campos Obrigatórios

Figura 3.13: página de edição de pessoas.

Especialidades Médicas - Filtro de Pesquisa

Nome

Pesquisa Reiniciar Cadastrar Especialidade Médica

Resultados da Pesquisa (53)

Nome	Sociedade	Site	Ação
ACUPUNTURA			Visualizar Editar
ALERGIA E IMUNOLOGIA			Visualizar Editar
ANESTESIOLOGIA			Visualizar Editar
ANGIOLOGIA			Visualizar Editar
CANCEROLOGIA			Visualizar Editar

[Próxima Página >](#) [Última Página >>](#)

Figura 3.14: página de pesquisa de especialidades médicas.

Editar Especialidade Médica

Nome *

Sociedade

Site

* Campos Obrigatórios

Salvar Alterações Excluir Cancelar Editar

Figura 3.15: página de edição de especialidades médicas.

do conselho e a unidade federativa. As Figuras 3.16 e 3.17 apresentam as páginas de pesquisa e edição de médicos, nas quais podem ser realizadas as atividades de pesquisa, cadastro, alteração e exclusão pelo administrador.

3.3.4 Queixas gerais

Ao abrir uma consulta, o profissional de saúde ou recepcionista deve informar as queixas gerais apresentadas pelo paciente. Para realizar esta atividade e permitir a elaboração de estatísticas posteriores, deve existir um cadastro de queixas gerais, além de uma funcionalidade que permita categorizá-las. As Figuras 3.18 a 3.21 mostram a alteração, exclusão e cadastro de queixas gerais e categorias de queixas gerais.

[HOME/UFES](#)
[Sua Conta](#)
[Pessoa](#)
[Saúde](#)
[Segurança](#)
[Configurações](#)
Autenticado como: ROOT (root) [Sair](#)

Médicos - Filtro de Pesquisa

Registro
UF
Tipo

[Pesquisa](#)
[Reiniciar](#)
[Cadastrar Médico](#)

Resultados da Pesquisa (3)

Registro ↕	Tipo ↕	UF ↕	Especialidade Médica ↕	Ação
██████	CRM	ESPÍRITO SANTO	CIRURGIA CARDIOVASCULAR	Visualizar Editar
██████	CRM	CEARÁ	CIRURGIA VASCULAR	Visualizar Editar
██████	CRM	DISTRITO FEDERAL	CLÍNICA MÉDICA	Visualizar Editar

Figura 3.16: página de pesquisa de médicos.

[HOME/UFES](#)
[Sua Conta](#)
[Pessoa](#)
[Saúde](#)
[Segurança](#)
[Configurações](#)
Autenticado como: ROOT (root) [Sair](#)

Editar Médico

Pessoa *

Especialidade Médica *

Registro *

Tipo *

UF *

* Campos Obrigatórios

[Salvar Alterações](#)
[Excluir](#)
[Cancelar Editar](#)

Figura 3.17: página de edição de médicos.

[HOME/UFES](#)
[Sua Conta](#)
[Pessoa](#)
[Saúde](#)
[Segurança](#)
[Configurações](#)
Autenticado como: ROOT (root) [Sair](#)

Categorias de Queixa Geral - Filtro de Pesquisa

Nome
Descrição

[Pesquisa](#)
[Reiniciar](#)
[Cadastrar Categoria de Queixa Geral](#)

Resultados da Pesquisa (8)

Nome ↕	Descrição ↕	Ação
SISTEMAS CIRCULATÓRIO E RESPIRATÓRIO		Visualizar Editar
SISTEMA DIGESTIVO E ABDÔMEN		Visualizar Editar
PELE E TECIDO SUBCUTÂNEO		Visualizar Editar
SISTEMAS NERVOSO E MUSCULOESQUELÉTICO		Visualizar Editar
SINAIS E SINTOMAS GERAIS		Visualizar Editar

Figura 3.18: página de pesquisa de categorias de queixas gerais.

Figura 3.19: página de edição de categorias de queixas gerais.

Nome	Descrição	Ação
TAQUICARDIA		Visualizar Editar
BRADICARDIA		Visualizar Editar
PALPITAÇÃO		Visualizar Editar
SOPRO CARDÍACO		Visualizar Editar
SANGRAMENTO NASAL		Visualizar Editar

Figura 3.20: página de pesquisa de queixas gerais.

Figura 3.21: página de edição de queixas gerais.

3.3.5 Eventos

Ao realizar o atendimento de um paciente, o médico pode solicitar a execução de diversas atividades, como procedimentos e exames. No RES proposto, estas atividades são definidas como eventos e podem ser mantidas através de um cadastro de eventos, tipos de eventos e categorias de eventos. Esta hierarquia de classificação permite a subclassificação e organização de acordo com as necessidades da instituição de saúde.

Tipos de eventos

As páginas de pesquisa e edição de tipos de eventos são mostradas nas Figuras 3.22 e 3.23. Através destas funcionalidades, o administrador pode realizar as atividades de pesquisa, cadastro, alteração e exclusão de tipos de eventos.

Tipos de Eventos - Filtro de Pesquisa

Nome

Descrição

Pesquisa Reiniciar Criar Tipo de Evento

Resultados da Pesquisa (1)

Nome ↕	Descrição ↕	Ação
EXAME		Visualizar Editar

Figura 3.22: página de pesquisa de tipos de eventos.

Editar Tipo de Evento

Nome *

Descrição

* Campos Obrigatórios

Salvar Alterações Excluir Cancelar Editar

Figura 3.23: página de edição de tipos de eventos.

Categorias de eventos

As páginas de pesquisa e edição de categorias de eventos podem ser visualizadas nas Figuras 3.24 e 3.25. Através destas funcionalidades, o administrador pode realizar as atividades de pesquisa, cadastro, alteração e exclusão de categorias de eventos.

Categorias de Eventos - Filtro de Pesquisa

Nome

Descrição

Pesquisa Reiniciar Cadastrar Categoria de Evento

Resultados da Pesquisa (8)

Nome ↕	Descrição ↕	Ação
EXAMES GERAIS		Visualizar Editar
ELETROENCEFALOGRAFIA		Visualizar Editar
ENDOSCOPIA		Visualizar Editar
EXAMES FÍSICOS		Visualizar Editar
EXAMES DE FEZES		Visualizar Editar

[Próxima Página >](#) [Última Página >>](#)

Figura 3.24: página de pesquisa de categorias de eventos.

Editar Categoria de Evento

Tipo de Evento * EXAME ▾

Nome * EXAMES GERAIS

Descrição

* Campos Obrigatórios

Salvar Alterações Excluir Cancelar Editar

Figura 3.25: página de edição de categorias de eventos.

Eventos

As páginas de pesquisa e edição de eventos são mostradas nas Figuras 3.26 e 3.27. Através destas funcionalidades, o administrador pode realizar as atividades de pesquisa, cadastro, alteração e exclusão de eventos.

Nome	Descrição	Ação
ABDOMINOSCOPIA		Visualizar Editar
ABREUGRAFIA		Visualizar Editar
ANTILOGRAMA		Visualizar Editar
ANUSCOPIA		Visualizar Editar
ANÁLISE MOLECULAR PARA A SÍNDROME DO X FRÁGIL		Visualizar Editar

Figura 3.26: página de pesquisa de eventos.

* Campos Obrigatórios

Figura 3.27: página de edição de eventos.

3.3.6 Vacinas

Para permitir a solicitação de imunizações para um paciente, faz-se necessário que exista um cadastro das vacinas disponíveis para aplicação. As Figuras 3.28 e 3.29 mostram as páginas de pesquisa e edição de eventos, nas quais podem ser realizadas as atividades de pesquisa, cadastro, alteração e exclusão, respectivamente.

Vacinas - Filtro de Pesquisa

Nome

Descrição

Pesquisa Reiniciar Cadastrar Vacina

Resultados da Pesquisa (65)

Nome ↕	Descrição ↕	Ação
SORO ANTIARACNÍDICO (LOXOSCELES E PHONEUTRIA) E ANTIESCORPIÔNICO		Visualizar Editar
SORO ANTIBOTRÓPICO (PENTAVALENTE)		Visualizar Editar
SORO ANTIBOTRÓPICO (PENTAVALENTE) E ANTICROTÁLICO		Visualizar Editar
SORO ANTIBOTRÓPICO (PENTAVALENTE), ANTICROTÁLICO E ANTILAQUÉTICO		Visualizar Editar
SORO ANTIBOTRÓPICO (PENTAVALENTE) E ANTILAQUÉTICO		Visualizar Editar

[Próxima Página >](#) [Última Página >>](#)

Figura 3.28: página de pesquisa de vacinas.

Editar Vacina

Nome *

Descrição

* Campos Obrigatórios

Salvar Alterações Excluir Cancelar Editar

Figura 3.29: página de edição de vacinas.

3.3.7 Configurações

Algumas funcionalidades do RES proposto necessitam de configurações para funcionar apropriadamente. Para facilitar o gerenciamento destas configurações, é criado um cadastro que permite realizar as atividades de pesquisa, cadastro, alteração e exclusão, conforme mostrado nas Figuras 3.30 e 3.31.



The screenshot shows a web application interface with a navigation bar at the top containing links for 'MIMC MEC', 'Sua Conta', 'Pessoa', 'Saúde', 'Segurança', 'Configurações', and 'Autenticado como: ROOT (root)'. Below the navigation bar is a search form titled 'Configurações - Filtro de Pesquisa' with two input fields: 'Nome' and 'Valor'. Below the search form are three buttons: 'Pesquisa', 'Reiniciar', and 'Cadastrar Configuração'. Below the buttons is a table titled 'Resultados da Pesquisa (5)' with three columns: 'Nome', 'Valor', and 'Ação'. The table contains five rows of configuration data.

Nome	Valor	Ação
maxLoginFailed	3	Visualizar Editar
passwordUpdateIntervalDays	30	Visualizar Editar
patientCategoryId	1	Visualizar Editar
serverIp	127.0.0.1	Visualizar Editar
serverPort	8080	Visualizar Editar

Figura 3.30: página de pesquisa de configurações.



The screenshot shows a web application interface with a navigation bar at the top containing links for 'MIMC MEC', 'Sua Conta', 'Pessoa', 'Saúde', 'Segurança', 'Configurações', and 'Autenticado como: ROOT (root)'. Below the navigation bar is a form titled 'Editar Configuração' with two input fields: 'Nome' and 'Valor'. The 'Nome' field contains the text 'maxLoginFailed' and the 'Valor' field contains the text '3'. Below the form is a note: '* Campos Obrigatórios'. Below the note are three buttons: 'Salvar Alterações', 'Excluir', and 'Cancelar Editar'.

Figura 3.31: página de edição de configurações.

3.4 Fluxo de atendimento

Esta seção tem como objetivo demonstrar todos os passos de um fluxo de atendimento no RES proposto, iniciando pela autenticação do usuário responsável pela recepção, seguido pelo cadastro de pacientes e abertura de consulta.

3.4.1 Verificação de usuário

O formulário de autenticação utilizado pelo recepcionista é mostrado na Figura 3.32. Ao receber um paciente, o recepcionista é responsável por verificar se este está cadastrado no RES, conforme pode ser visualizado na Figura 3.33.

HUWC.UFC [Entrar](#)

Autenticação

Nome de Usuário

Senha

7 + 26 =

[Entrar](#)

Figura 3.32: página de autenticação do recepcionista.

HUWC.UFC Sua Conta Pessoa Saúde Autenticado como: (recepcionista1) [Sair](#)

Pessoas - Filtro de Pesquisa

Nome Completo

RG

CPF

Ativo

[Pesquisa](#) [Reiniciar](#) [Cadastrar Pessoa](#)

Resultados da Pesquisa (0)

Não há resultados para sua consulta.

Figura 3.33: página de pesquisa de pessoas.

Caso o paciente possua cadastro, o recepcionista pode partir para abertura da consulta. Caso não possua, o recepcionista deve efetuar o cadastro do novo paciente, conforme mostrado na Figura 3.34.

HUWC_UFC Sua Conta Pessoa Saúde Autenticado como: (recepcionista) Sair

Adicionar Pessoa

Dados Pessoais

Senha **15312030**

Nome Completo *

Nome de Usuário * paciente4

Nome do Pai

Nome da Mãe *

CNIS da Mãe

Sexo * FEMININO

Data de Nascimento * 09/08/1950

E-mail

Naturalidade

Data de Naturalização

Registro de Naturalização

Data de Entrada no Brasil

Estado de Nascimento * MATO GROSSO

Nacionalidade BRASILEIRO

RG

Emissor do RG

CPF

Estudante * Não

Telefone Residencial

Telefone Celular

Telefone Comercial

Telefone de Contato

Nome da Pessoa de Contato

Religião

Tipo Sanguíneo U

Escolaridade * NÃO SABE LER/ESCREVER

Situação Familiar * CONVIVE COM COMPANHEIRA(O) E FILHO(S)

Raça * BRANCA

Ativar Sim

Grupos

Todos os grupos

Nome
GRUPO 01
GRUPO 02
GRUPO 03
GRUPO 04
GRUPO 05

» Copiar Todos

» Copiar

« Excluir

« Excluir Todos

Grupos

Nome

⌵ Topo

⬅ Cima

⬇ Baixo

⌵ Rodapé

Categorias

Todas as categorias

Nome
ROOT
ADMINISTRADOR
MÉDICO
RECEPCIONISTA
ENFERMEIRO

» Copiar Todos

» Copiar

« Excluir

« Excluir Todos

Categorias

Nome
PACIENTE

⌵ Topo

⬅ Cima

⬇ Baixo

⌵ Rodapé

* Campos Obrigatórios

Cadastrar Cancelar Adicionar

Figura 3.34: página de cadastro de paciente.

3.4.2 Abertura da consulta

Com o paciente cadastrado, o recepcionista pode proceder a abertura da consulta. Para tanto, o recepcionista efetua a solicitação de abertura da consulta, conforme apresentado na Figura 3.35.

HUWC.UFC Sua Conta Pessoa Saúde Autenticado como: (recepcionista1) Sair

Editar Pessoa

Dados Pessoais Endereços Consultas Solicitações de Eventos Imunizações Diagnósticos

Não há nenhuma consulta associada a este médico.

Adicionar Consulta

Figura 3.35: página de edição de paciente - adicionar consulta.

Em seguida, comunica-se com o paciente para obter as informações necessárias para preencher o formulário de abertura da consulta, conforme mostrado na Figura 3.36.

HUWC.UFC Sua Conta Pessoa Saúde Autenticado como: (recepcionista1) Sair

Adicionar Consulta

Médico * CRM- -DISTRITO FEDERAL

Situação * ABERTA

Solicitado * 29/1/2010 22:18

Processado

Em Espera

Executado

Queixas Gerais

Todos as queixas gerais

Nome
DOENÇA CARDÍACA
STOKES
TAQUICARDIA
TENESMO VESICAL
TETANIA
TONTURA
TREMOR
URTIQARIA

Queixas Gerais

Nome
ESCARRO
TOSSE

Copiar Todos Copiar Excluir Excluir Todos Topo Cima Baixo Rodapé

Observações

Cadastrar Cancelar Adicionar

Figura 3.36: página de cadastro de consulta.

3.4.3 Fila de espera

O recepcionista também é responsável por manter a fila de espera do atendimento. Assim, à medida que o médico está disponível para atendimento, o recepcionista seleciona o paciente que deve ser atendido pela ordem de chegada, conforme pode ser visualizado na Figura 3.37. Em seguida, abre a consulta e altera o seu estado para que a mesma fique em “atendimento”.

Consultas - Filtro de Pesquisa

Situação: ABERTA

Pesquisa Reiniciar

Resultados da Pesquisa (2)

Pessoa	Médico	Sintomas	Solicitado	Situação	Ação
		▪ TOSSE	26/1/2010 22:29	ABERTA	Visualizar Editar
		▪ ESCARRO ▪ TOSSE	29/1/2010 22:18	ABERTA	Visualizar Editar

Figura 3.37: página de fila de espera.

3.4.4 Realização da consulta

A partir da sua página de acesso, o médico pode visualizar a lista de pacientes que estão aguardando atendimento. Esta lista inclui as últimas consultas que foram realizadas e as consultas que ainda estão em aberto, conforme apresentado na Figura 3.38.

Perfil do médico

Últimas consultas

Pessoa	CPF	Solicitado	Situação	Ação
		29/01/2010	EM CURSO	Visualizar Editar
		26/01/2010	ABERTA	Visualizar Editar

Figura 3.38: página inicial do médico com consultas em aberto.

A partir do momento que o paciente chega ao consultório, o médico pode visualizar a consulta aberta, incluindo a queixa geral que motivou sua vinda,

conforme mostrado na Figura 3.39.

Editar Consulta

Médico * [CRM-DISTRITO FEDERAL]

Situação * EM CURSO

Solicitado * 29/1/2010 22:18

Processado 29/1/2010 22:23

Em Espera

Executado

Queixas Gerais

Todos as queixas gerais

Nome
ABDÔMEN AGUDO
AGEUSIA
AGNOSIA
ALEXIA
AMNÉSIA ANTERÓGRADA
AMNÉSIA RETRÓGRADA
ANOREXIA/POLIDIPSIA/POLIFAGIA

Queixas Gerais

Nome
ESCARRO
TOSSE

Observações

Solicitações de Eventos: Imunizações | Diagnóstico | Prescrição Médica

Não há nenhuma solicitação de evento cadastrada para esta consulta.

Adicionar Solicitação de Evento

Salvar Alterações | Cancelar Editar | Aguardando | Pronto

Figura 3.39: página de edição da consulta.

3.4.5 Solicitações de eventos

Após realizar a consulta com o paciente, o médico pode optar por solicitar a realização de algum evento, como um exame, por exemplo. Para tanto, a partir da página referente à consulta, o médico pode efetuar o cadastro de uma nova solicitação de evento, conforme mostrado na Figura 3.40. Neste momento, o médico pode modificar o estado da consulta para “aguardando” e encaminhar o paciente para que a solicitação de evento seja realizada.

3.4.6 Resultados de solicitações de eventos

Nas instituições de saúde existem profissionais dedicados à execução dos mais diversos tipos de eventos, desde exames simples a tomografias computadorizadas. No RES proposto, todas as solicitações de eventos realizadas pelos médicos podem

Adicionar Solicitação de Evento

Evento * TOMOGRAFIA COMPUTADORIZADA

Requisitado * 29/1/2010 22:25

Executado

Estado * ABERTO

* Campos Obrigatórios

Cadastrar Cancelar Adicionar

Pessoa

Nome Completo	Nome de Usuário	Nome do Pai	Nome da Mãe
	paciente4		

Figura 3.40: página de cadastro de solicitação de evento.

ser visualizadas por estes profissionais, de forma que possam executá-las. Isto é apresentado na Figura 3.41.

Solicitações de Eventos - Filtro de Pesquisa

Estado * ABERTO

Pesquisa Reiniciar

Resultados da Pesquisa (1)

Pessoa	Tipo de Evento	Categoria de Evento	Evento	CPF	Ação
	EXAME	REFLEXOS			Visualizar Editar

Figura 3.41: página de pesquisa de solicitações de evento.

No momento em que o profissional de saúde identifica uma solicitação de evento de sua responsabilidade, o RES disponibiliza a opção de visualizá-la, assim como apresentado na Figura 3.42.

Em seguida, o profissional altera o estado da solicitação para “atendimento”, permitindo a inclusão de resultados em texto, imagem, arquivos binários e arquivos no padrão DICOM, conforme pode ser visualizado na Figura 3.43.

Para finalizar, o profissional responsável altera o estado da solicitação de evento para “realizado”. Neste momento, o médico que efetuou a solicitação de evento pode verificar o seu resultado, conforme visto na Figura 3.44.

HUWC.UFC Sua Conta Pessoa Saúde Autenticado como: (profisaude1) Sair

Editar Solicitação de Evento

Evento * TOMOGRAFIA COMPUTADORIZADA

Requisitado * 29/1/2010 22:25

Executado

Estado * ABERTO

* Campos Obrigatórios

Salvar Alterações Em processamento Cancelar Solicitação de Evento Cancelar Editar

Figura 3.42: página de edição da solicitação de evento em aberto.

HUWC.UFC Sua Conta Pessoa Saúde Autenticado como: (profisaude1) Sair

Editar Solicitação de Evento

Evento * TOMOGRAFIA COMPUTADORIZADA

Requisitado * 29/1/2010 22:25

Executado 29/1/2010 23:59

Estado * PROCESSAMENTO

Resultado Textual:

Imagens: Escolher Arquivos

Arquivos Binários: Escolher Arquivos

Arquivos DICOM: Escolher Arquivos Limpar Todos

dpoc1_supina_cpap0_apice1.dcm
Done Limpar

* Campos Obrigatórios

Salvar Alterações Pronto Cancelar Editar

Pessoa Imagens Arquivos Binários Arquivos DICOM

Nome Completo	Nome de Usuário	Nome do Pai	Nome da Mãe
	paciente4		

Figura 3.43: página de edição da solicitação de evento em atendimento.

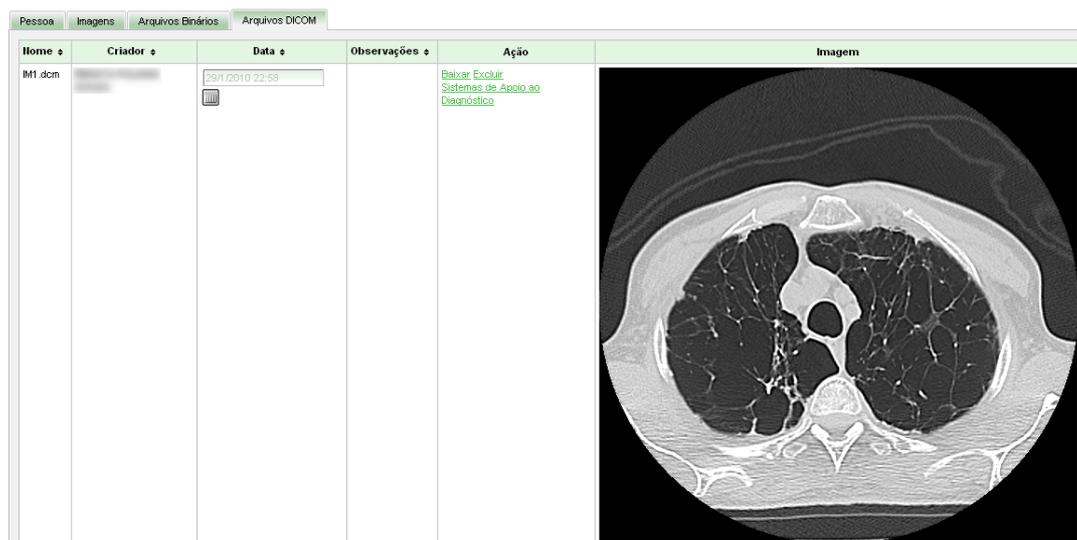


Figura 3.44: página de resultado da solicitação de evento.

3.4.7 Apoio ao diagnóstico

O RES proposto disponibiliza uma infraestrutura de *plugins* para inclusão de sistemas de apoio ao diagnóstico médico. Para esta versão, foi realizada a integração com um sistema de análise de enfisema pulmonar desenvolvido por Filho (FILHO, 2010), que analisa o percentual de enfisema em cada um dos pulmões a partir de um arquivo DICOM obtido de tomografia computadorizada do tórax. A solicitação de execução de um sistema de apoio ao diagnóstico pode ser visualizada na Figura 3.45.

Como a resposta do sistema de apoio ao diagnóstico não é imediata, o RES proposto libera a interface do usuário para utilização de outras funcionalidades enquanto o processamento é efetuado, conforme mensagem mostrada na Figura 3.46. Quando o processamento é finalizado, a resposta pode ser obtida na lista de arquivos DICOM referentes à solicitação de evento, conforme mensagem que pode ser visualizada na Figura 3.47.

3.4.8 Edição de imagens médicas

Outra funcionalidade que o RES proposto disponibiliza aos médicos é a edição de imagens, com o objetivo de satisfazer a necessidade de efetuar marcações nas imagens de exames para indicar os fatores que conduzem ao diagnóstico. Através do editor de imagens, o médico pode desenhar retângulos, elipses, retas e texto, assim como modificar a cor do desenho, conforme pode ser visualizado na Figura 3.48.

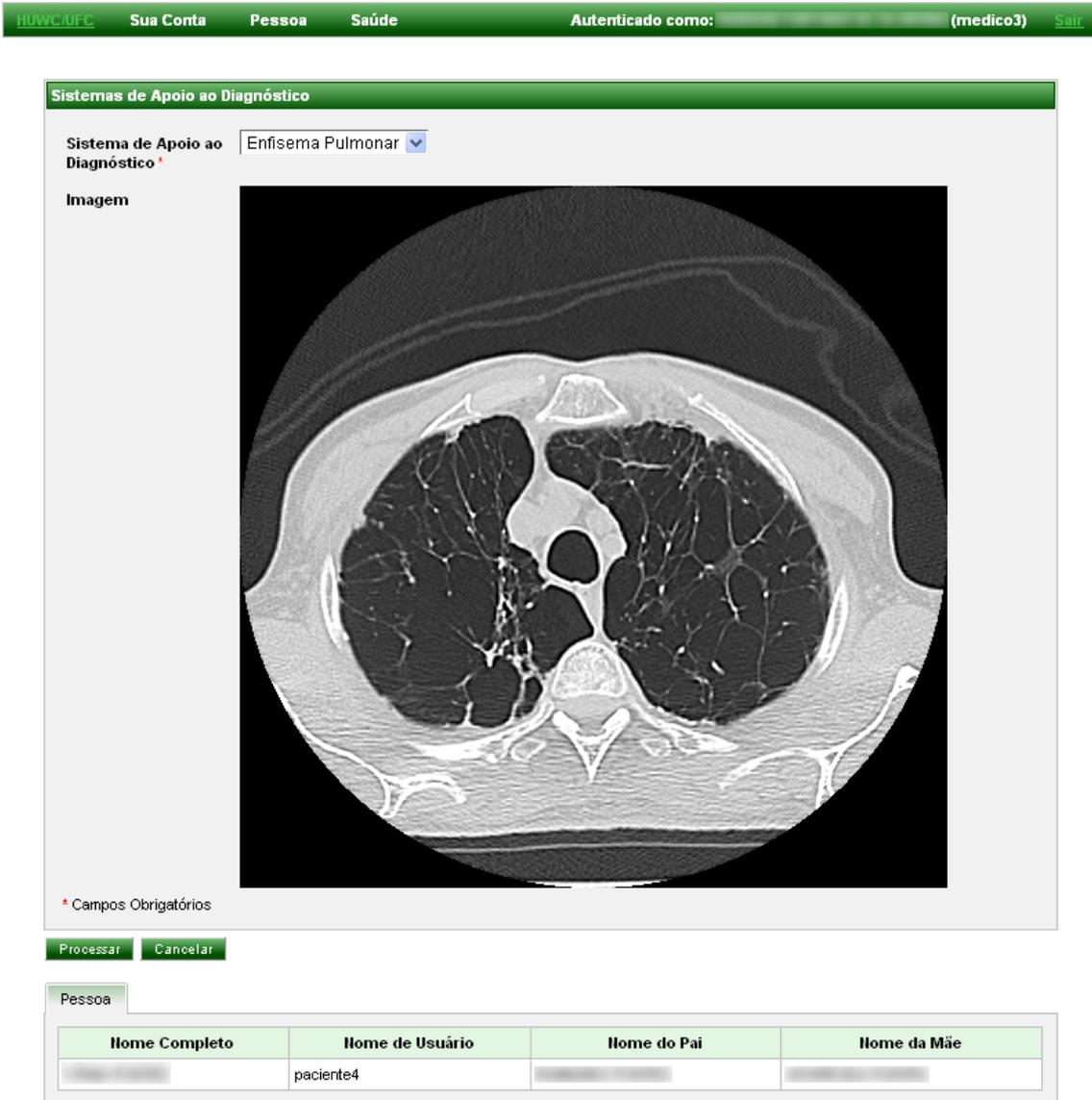


Figura 3.45: página de solicitação de auxílio ao diagnóstico.



Figura 3.46: página de processamento da solicitação de apoio ao diagnóstico.

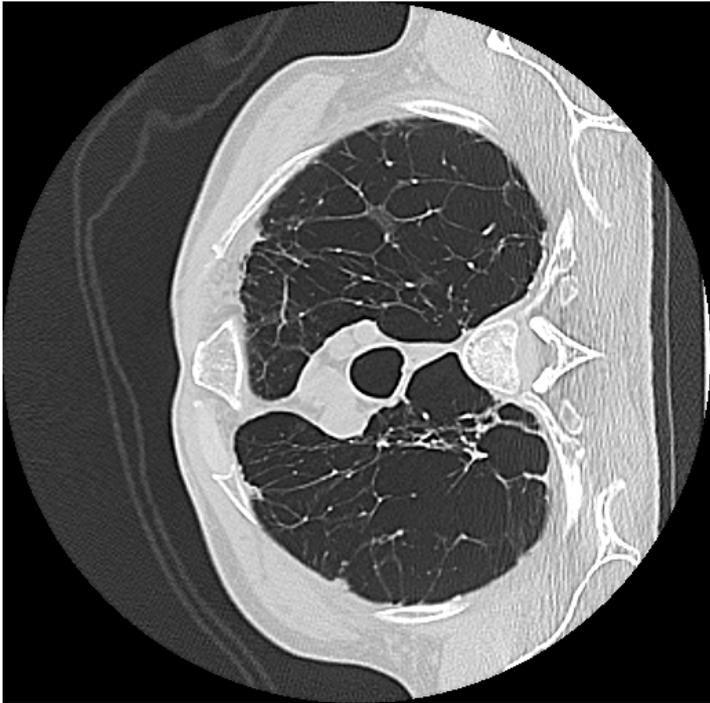
Pessoa			
Imagens			
Arquivos Binários			
Arquivos DICOM			
Nome ↕	lm1.dcm	Criador ↕	[Redacted]
Data ↕	29/11/2010 22:58 [Calendar icon]	Observações ↕	Percentual de enfisema no PE: 68.0 Percentual de enfisema no PD: 64.0 Regiao segmentada no PE: 277773 Regiao segmentada no PD: 29634
Ação	Baixar Excluir Sistemas de Arquivo ao Diagnóstico	Imagem	

Figura 3.47: página de resultado da solicitação de apoio ao diagnóstico.

No instante em que o médico termina a edição, uma nova imagem é gerada com as marcações e, desta forma, a imagem original é mantida intacta.

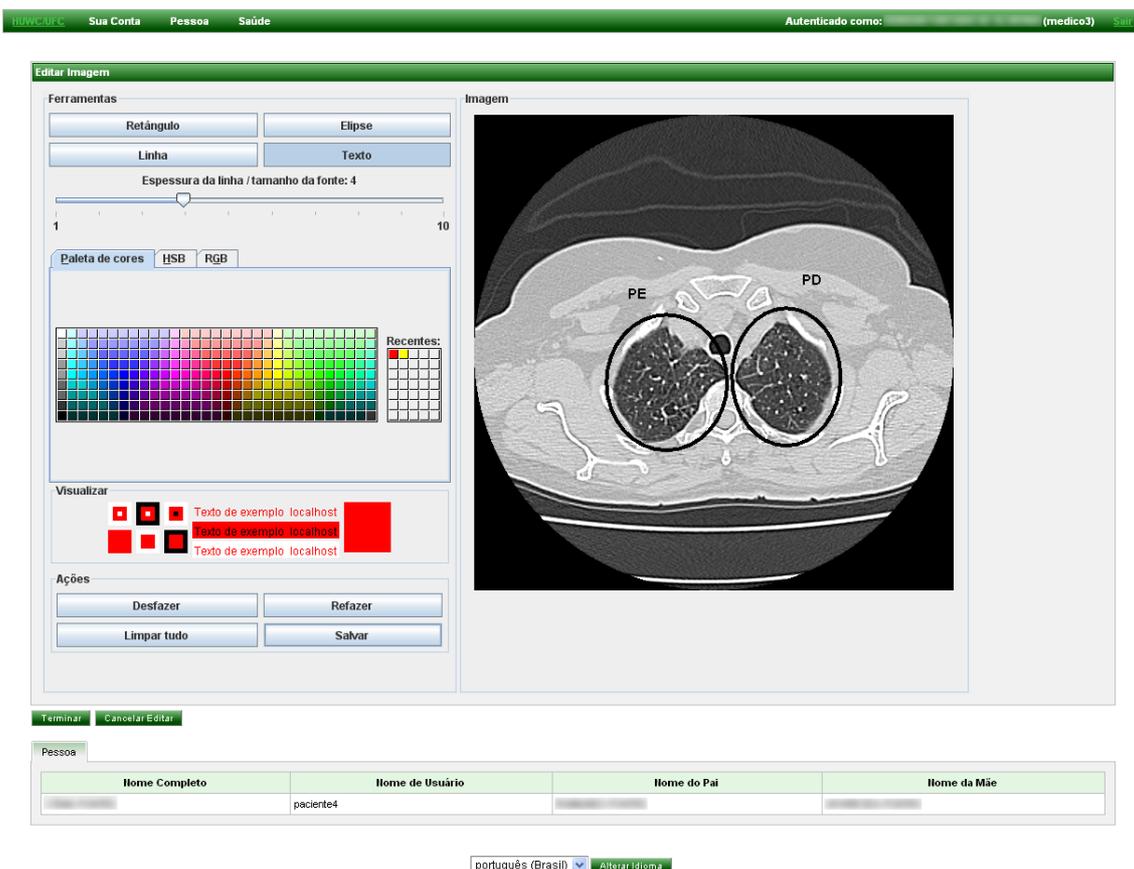


Figura 3.48: página de edição de imagens.

3.4.9 Imunizações

Durante a consulta, o médico pode solicitar uma imunização para o paciente, conforme pode ser visualizado na Figura 3.49. Esta imunização permanece como forma de alerta na página inicial do usuário até que seja realizada.

3.4.10 Diagnósticos

O RES proposto disponibiliza um componente de inclusão de diagnósticos dinâmico, responsável por buscar as doenças na CID e trazer ao médico uma lista dos diagnósticos disponíveis para o trecho de texto fornecido. O objetivo desta interface é disponibilizar um componente de busca automatizado, dado que a quantidade de registros na CID é alta o suficiente para dificultar a busca por meios tradicionais. A inclusão de um diagnóstico é mostrada na Figura 3.50.

Nome Completo	Nome de Usuário	Nome do Pai	Nome da Mãe
	paciente4		

Figura 3.49: página de solicitação de imunização.

Nome Completo	Nome de Usuário	Nome do Pai	Nome da Mãe
	paciente4		

Figura 3.50: página de inclusão de diagnóstico.

3.4.11 Prescrição médica

O RES proposto disponibiliza uma interface de criação e impressão em arquivo no formato *PDF* das prescrições médicas originadas em uma consulta. O cadastro de uma prescrição médica pode ser visualizado na Figura 3.51 e o arquivo no formato *PDF* na Figura 3.52, podendo ser personalizado para o formulário padrão da instituição de saúde.

3.5 Acesso do Paciente

No RES proposto, o usuário é dono de suas informações e, por este motivo, tem o direito de acessá-las independente da categoria de usuário ao qual está associado. Por esta razão, a interface padrão de acesso utilizada é a *WEB*, garantindo que qualquer usuário conectado à Internet e com um navegador compatível com as tecnologias utilizadas possa efetuar o acesso. Desta forma, atividades como alteração de senha,

The screenshot shows a web application interface with a green header bar. The header contains the text 'HUVCF', 'Sua Conta', 'Pessoa', 'Saúde', 'Autenticado como: [nome] (medico3)', and a 'Sair' button. Below the header is a form titled 'Adicionar Prescrição Médica'. The form has a label 'Prescrição *' and a text input field containing 'SEKI - 1 FRASCO' and '5ML 6/6H'. Below the input field is a note '* Campos Obrigatórios'. At the bottom of the form are two buttons: 'Cadastrar' and 'Cancelar Adicionar'.

Figura 3.51: página de inclusão de prescrição médica.

The screenshot shows a print page for a medical prescription. At the top is a blue banner with the logo of Hospital Universitário Walter Cantídio and the text 'UNIVERSIDADE FEDERAL DO CEARÁ'. Below the banner is the title 'Prescrição Médica'. The page contains the following information: 'EM: 30 Jan 2010 03:47:12 GMT', 'AO PACIENTE: [nome]', a barcode with '(11)' below it, a box containing the prescription text 'SEKI - 1 FRASCO' and '5ML 6/6H', a signature line with a blurred name, 'CLÍNICA MÉDICA', and 'CRM [número] DISTRITO FEDERAL'. At the bottom is the address: 'Rua Capitão Francisco Pedro, 1290 - Rodolfo Teófilo - 60430-370 - (85) 3366.8167' and 'Fortaleza - Ceará'.

Figura 3.52: página de impressão de prescrição médica.

atualização de dados pessoais e visualização de dados clínicos podem ser realizadas.

3.5.1 Alteração de senha

Por medida de segurança, todo usuário do RES proposto deve atualizar sua senha em um período de tempo configurável, conforme pode ser visualizado na Figura 3.53.

HUWC.UFC Sua Conta Pessoa Autenticado como: (paciente4) Sair

❗ Erro ao alterar senha: a senha informada é muito simples. Mínimo de 8 caracteres, sendo pelo menos um numérico.

Alterar Senha

Senha Anterior *

Nova Senha *

Confirmação da Nova Senha *

Salvar Alterações

Figura 3.53: página de alteração de senha do paciente.

3.5.2 Atualização de dados pessoais

Sempre que for necessário, os usuários podem alterar os seus dados pessoais como nome, endereço, dentre outros. A utilização desta funcionalidade é mostrada na Figura 3.54.

3.5.3 Visualização de dados clínicos

Assim como os usuários podem visualizar seus dados pessoais, também podem visualizar os seus dados clínicos, como consultas, eventos, imunizações, diagnósticos e prescrições médicas, dentre outros. As Figuras 3.55 a 3.58 permitem visualizar a utilização destas funcionalidades.

3.5.4 Central de mensagens

Outra funcionalidade que o RES proposto disponibiliza aos seus usuários é a central de mensagens. A partir desta, os usuários podem se comunicar com qualquer outro usuário cadastrado, sejam pacientes, médicos, recepcionistas ou administradores, assim como saber se a mensagem enviada já foi lida ou não. As Figuras 3.59 a 3.61 demonstram a comunicação de um paciente com um médico.

HUWC/UFC Sua Conta Pessoa Autenticado como: (paciente4) Sair

Editar Pessoa

Dados Pessoais Endereços Consultas Solicitações de Eventos Inunizações Diagnósticos

Nome Completo *

Nome de Usuário *

Nome do Pai

Nome da Mãe *

CNS da Mãe

Sexo *

Data de Nascimento *

E-mail

Naturalidade

Data de Naturalização

Registro de Naturalização

Data de Entrada no Brasil

Estado de Nascimento *

Nacionalidade

RG

Emissor do RG

CPF

Estudante *

Telefone Residencial

Telefone Celular

Telefone Comercial

Telefone de Contato

Nome da Pessoa de Contato

Religião

Tipo Sanguíneo

Escolaridade *

Situação Familiar *

Raça *

Ativar

Grupos

Todos os grupos		Grupos	
Nome		Nome	
GRUPO 01			
GRUPO 02			
GRUPO 03			
GRUPO 04			
GRUPO 05			

Categorias

Todas as categorias		Categorias	
Nome		Nome	
ROOT		PACIENTE	
ADMINISTRADOR			
MÉDICO			
RECEPCIONISTA			
ENFERMEIRO			

* Campos Obrigatórios

Figura 3.54: página de edição de dados pessoais.

HUWC.UFC Sua Conta Pessoa Autenticado como: (paciente4) Sair

Editar Pessoa

Dados Pessoais Endereços Consultas Solicitações de Eventos Imunizações Diagnósticos

Médico	Especialidade Médica	Sintomas	Solicitado	Situação	Ação
	CLÍNICA MÉDICA	<ul style="list-style-type: none"> ESCARRO TOSSE 	29/1/2010 22:18	REALIZADA	Visualizar

português (Brasil) Alterar Idioma

Figura 3.55: página de visualização de dados clínicos - consultas.

HUWC.UFC Sua Conta Pessoa Autenticado como: (paciente4) Sair

Editar Pessoa

Dados Pessoais Endereços Consultas Solicitações de Eventos Imunizações Diagnósticos

Evento	Requisitado	Executado	Estado	Ação
TOMOGRAFIA COMPUTADORIZADA	29/01/2010	29/01/2010	REALIZADO	Visualizar

português (Brasil) Alterar Idioma

Figura 3.56: página de visualização de dados clínicos - solicitações de eventos.

HUWC.UFC Sua Conta Pessoa Autenticado como: (paciente4) Sair

Editar Pessoa

Dados Pessoais Endereços Consultas Solicitações de Eventos Imunizações Diagnósticos

Não há nenhuma imunização cadastrada para esta pessoa.

português (Brasil) Alterar Idioma

Figura 3.57: página de visualização de dados clínicos - imunizações.

HUWC.UFC Sua Conta Pessoa Autenticado como: (paciente4) Sair

Editar Pessoa

Dados Pessoais Endereços Consultas Solicitações de Eventos Imunizações Diagnósticos

Código	Nome
J431	Enfisema panlobular

português (Brasil) Alterar Idioma

Figura 3.58: página de visualização de dados clínicos - diagnósticos.

[HUWC/UFC](#) Sua Conta Pessoa Autenticado como: [nome] (paciente4) [Sair](#)

Adicionar Mensagem

De [nome]
Categoria * MÉDICO
Para * [nome]
Assunto * Medicação
Texto *

Olá Dr. [nome]

A minha tosse voltou. Posso tomar o mesmo medicamento?

Atenciosamente,
[nome]

Figura 3.59: página de inclusão de mensagem do paciente.

[HUWC/UFC](#) Sua Conta Pessoa Saúde Autenticado como: [nome] (medico3) [Sair](#)

Você tem mensagens não lidas (1). [Visualizar](#)

Perfil do médico

Últimas consultas

Pessoa ↕	CPF ↕	Solicitado ↕	Situação ↕	Ação
[nome]		29/01/2010	REALIZADA	Visualizar Editar
[nome]		26/01/2010	ABERTA	Visualizar Editar

Figura 3.60: página inicial do médico com alerta de mensagem não lida.

[HUWC/UFC](#) Sua Conta Pessoa Saúde Autenticado como: [nome] (medico3) [Sair](#)

Responder Mensagem

De [nome]
Para * [nome]
Assunto * Re: Medicação
Texto *

Ainda não.

Volte ao meu consultório para que eu possa avaliar o seu estado atual.

Atenciosamente,
[nome]

Figura 3.61: página de resposta de mensagem do médico.

As funcionalidades apresentadas permitem conhecer os resultados obtidos no desenvolvimento do RES proposto, de forma que os objetivos especificados possam ser verificados e validados. Todas as funcionalidades desenvolvidas têm como objetivo principal facilitar o trabalho diário dos profissionais de saúde, tornando-se uma ferramenta de apoio ao atendimento de qualidade. Entretanto, o desenvolvimento de um RES faz parte de um processo longo e, como tal, necessita de tempo e de uma equipe de profissionais envolvidos. Por esta razão, algumas funcionalidades não estão disponíveis na primeira versão do RES proposto. Estas funcionalidades tornaram-se trabalhos futuros, sendo enumeradas no Capítulo a seguir, juntamente com as conclusões e contribuições deste trabalho.

Contribuições, conclusões e trabalhos futuros

4.1 Contribuições

Desde as etapas iniciais de desenvolvimento do RES proposto, o foco utilizado para guiar as principais escolhas é a disponibilização de recursos que contribuam para a execução das atividades do dia-a-dia dos profissionais envolvidos no processo de atenção à saúde. Partindo deste objetivo, este trabalho disponibiliza um grupo de contribuições obtidas através do seu desenvolvimento, sintetizadas a seguir.

4.1.1 Autenticação e autorização

O desenvolvimento de um módulo de autenticação e autorização completamente configurável, que permite o gerenciamento de permissões, categorias de permissões e usuários de acordo com a estrutura administrativa da instituição de saúde, disponibiliza uma funcionalidade essencial para a segurança e sigilo das informações de todos os usuários. Através deste recurso, a instituição de saúde pode definir quantos perfis de usuário forem necessários, selecionando exatamente quais componentes e funções podem ser utilizados por cada um destes perfis. Além disso, o padrão utilizado para armazenamento das permissões, baseado no *Java Authentication and Authorization Service* (JAAS) (JAAS, 2009), garante a possibilidade de armazenamento através de registros em diretórios *Lightweight Directory Access Protocol* (LDAP) (LDAP, 2009), de forma que possam ser compartilhados com outros sistemas baseados nesta mesma tecnologia.

4.1.2 Gerenciamento de dados

Além das permissões e categorias de permissões, todo o conjunto de informações responsáveis por disponibilizar os recursos necessários para utilização do RES proposto são completamente gerenciáveis. Desta forma, o cadastro de especialidades médicas, médicos, grupos, eventos, tipos e categorias de eventos, queixas gerais e categorias de queixas gerais, vacinas, usuários, dentre outros, podem ser facilmente mantidos de acordo com o interesse da instituição de saúde. Isto significa que sempre que necessário podem ser incluídas novas informações, como também modificadas ou excluídas, desde que não possuam vínculo com outras entidades em uso.

4.1.3 Acesso do paciente

Outro recurso disponível no RES proposto é o acesso do paciente. Entretanto, diferentemente de sistemas de registro médico individuais, como o *Google Health* (GOOGLEHEALTH, 2010), é mantido não apenas pelo paciente, mas por todos os profissionais envolvidos no processo de atenção à saúde. Assim, ao realizar seu cadastro no RES proposto, todos os passos realizados dentro da instituição de saúde são registrados e podem ser visualizados pelo paciente. Para citar um exemplo, se o paciente perder a prescrição fornecida pelo médico, pode facilmente acessar a consulta realizada e visualizar as dosagens dos medicamentos prescritas pelo médico. Da mesma forma, esta atividade pode ser utilizada para visualizar resultados de exames, lista de imunizações pendentes, dentre outras atividades. Além disso, através da central de mensagens o paciente pode comunicar-se com o médico que lhe atendeu, assim como com qualquer outro usuário cadastrado. Como a comunicação é realizada de forma interna, pode-se garantir a privacidade das informações trocadas. Adicionalmente, a ferramenta foi desenvolvida para permitir que o remetente saiba se o destinatário leu a mensagem. Desta forma, esta ferramenta constitui-se como um importante meio de integração entre o paciente e a instituição de saúde, permitindo que o paciente deixe de lado o papel tradicionalmente passivo e possa atuar como membro ativo do processo.

4.1.4 Internacionalização

Apesar do foco de trabalho utilizado durante o desenvolvimento do RES proposto estar direcionado totalmente ao paciente brasileiro, nosso país possui uma vocação inegável para o turismo e, por este motivo, recebe uma quantidade sempre crescente

de turistas em seu território. Logicamente, em alguma situação uma parte destes turistas necessita de algum tipo de atendimento médico e são encaminhados para as nossas instituições de saúde. Com o objetivo de atender este público, o RES proposto possui suporte para inclusão de múltiplos idiomas, de acordo com a necessidade da instituição de saúde. Na primeira versão, a interface possui suporte para os idiomas português brasileiro e inglês.

4.1.5 Leitura de arquivos no padrão DICOM

Outro recurso integrado ao RES proposto é a leitura de arquivos no padrão DICOM. Através da utilização deste recurso, as solicitações de eventos que possuem arquivos DICOM como resultado podem ser inseridas no prontuário do paciente, sendo automaticamente interpretadas para que o médico possa visualizar informações importantes para o diagnóstico. Esta prática viabiliza a utilização destes arquivos em outras atividades, como sistemas de apoio ao diagnóstico.

4.1.6 Apoio ao diagnóstico

Um dos recursos mais importantes do RES proposto, responsável por incluir uma camada de inteligência, trata-se da integração com sistemas de apoio ao diagnóstico. Para atingir este objetivo, foi desenvolvida uma infraestrutura de componentes de apoio ao diagnóstico, responsável por permitir a inclusão e utilização de componentes que realizam conexões com sistemas externos através da Internet, utilizando protocolos de comunicação bem definidos. Através deste recurso, os médicos podem realizar consultas a estes sistemas para obter apoio no diagnóstico realizado. A primeira versão do RES proposto possui integração com um sistema de análise de tomografias computadorizadas de tórax, capaz de indicar o percentual de enfisema nos pulmões. A integração com este tipo de sistema é de grande relevância para a prática médica, sendo considerada um dos grandes avanços na área da informática em saúde nos últimos anos.

4.1.7 Editor de imagens médicas

Com o objetivo de registrar a linha de raciocínio utilizada durante a análise de imagens médicas e elaboração de diagnósticos, o RES proposto disponibiliza um componente de edição de imagens médicas integrado, que disponibiliza a opção de incluir marcadores que indiquem pontos que levaram a um determinado diagnóstico, ao invés da descrição dos fatos utilizando linguagem natural. Através deste recurso,

os médicos podem incluir elipses, retângulos, retas e texto nas imagens dos exames médicos de seus pacientes, assim como possuem recursos para reverter erros e salvar novas imagens, preservando sempre a imagem original. A utilização deste recurso facilita a documentação do diagnóstico, uma vez que os detalhes percebidos nas imagens podem ser facilmente marcados e observados por outros médicos posteriormente, contribuindo para a difusão do raciocínio utilizado.

4.1.8 Prescrições médicas

Outro recurso desenvolvido trata da elaboração de prescrições médicas. Através deste recurso, o médico pode descrever os medicamentos que o paciente deve fazer uso diretamente no RES proposto, deixando por sua conta a formatação e impressão em documento padronizado. Este documento já inclui todas as informações necessárias, como identificação da instituição de saúde, do médico e do paciente, inclusive gerando código de barras para facilitar a identificação deste.

4.1.9 Componente dinâmico de inclusão de diagnósticos

Com o objetivo de padronizar o registro de diagnósticos médicos visando a elaboração de políticas preventivas, o RES proposto faz uso do Cadastro Internacional de Doenças (CID) na sua décima versão. Entretanto, dada a quantidade de doenças cadastradas, a utilização de sistemas tradicionais de pesquisa não é suficiente para buscar os registros de forma eficiente. Para resolver este problema, o RES proposto faz uso de um componente dinâmico de busca, em que à medida que o médico informa parte do nome da doença, a lista de doenças é analisada e automaticamente refinada, de forma que sejam apresentados dinamicamente os resultados que respeitam este critério de pesquisa. Desta forma, a definição de diagnósticos se torna muito mais simples e rápida.

4.1.10 Código-fonte aberto

Apesar do estado atual de desenvolvimento dos mais diversos tipos de softwares livres, ainda é bastante comum a utilização de softwares proprietários durante o desenvolvimento de sistemas de saúde. Para contribuir com a modificação deste cenário, o RES proposto foi desenvolvido utilizando exclusivamente software livre. Com isso, o custo relativo ao software proprietário é completamente removido, diminuindo o investimento necessário para sua implantação. Além disso, todo o código-fonte desenvolvido está disponível para a comunidade de informática em

saúde através do portal *Source Forge*. Desta forma, os conhecimentos adquiridos podem ser difundidos e reutilizados em outros projetos livres. Além disso, todas as tecnologias livres utilizadas durante o desenvolvimento foram frutos de extensa pesquisa para garantir a utilização do estado-da-arte, contribuindo para inovação e vanguarda da área da informática em saúde.

4.1.11 Componentes para auxiliar a usabilidade

O RES proposto possui um público-alvo diferenciado e, por este motivo, deve estar preparado para atuar adequadamente em cada caso. Durante todas as etapas de execução do trabalho, foram observados aspectos de usabilidade do sistema. Isto é demonstrado através do desenvolvimento de componentes que contribuem para a sua melhor utilização, como a busca dinâmica de doenças e o editor de imagens médicas. Outros fatores pontuais também foram incluídos com o objetivo de contribuir para o trabalho dos usuários, como componentes para inclusão de datas, menus e abas dinâmicas, dentre muitos outros componentes que melhoram a experiência de utilização do sistema.

4.1.12 Engenharia de software e modelagem de sistemas

Para finalizar, todo o desenvolvimento foi executado utilizando técnicas de engenharia de software e modelagem de sistemas, material que pode ser utilizado como fonte de pesquisa e adaptado para utilização em sistemas semelhantes. Além disso, sua utilização permitiu o controle das etapas de desenvolvimento, contribuindo significativamente para a entrega do RES proposto dentro das expectativas.

4.2 Conclusões

Baseado em todas as informações apresentadas neste trabalho, é possível concluir que o desenvolvimento de um registro eletrônico de saúde pode ser considerado um grande desafio para qualquer equipe de arquitetos de software ou engenheiros de software. O envolvimento de diversos padrões, a preocupação com a segurança e inviolabilidade dos dados, dentre muitos outros aspectos, são componentes que formam um dos softwares mais difíceis e complexos de se projetar.

Uma das maiores dificuldades encontradas durante o desenvolvimento foi a indisponibilidade de documentação referente a estes padrões, assim como a sua implementação. Também foram encontradas dificuldades para ter acesso à

informação médica especializada, inclusive para fazer a avaliação dos resultados produzidos.

Entretanto, se a implantação de um sistema deste tipo dá suporte à melhoria do atendimento aos pacientes, seu planejamento e desenvolvimento estão mais do que justificados.

Também é possível concluir que as tecnologias baseadas em software livre, desde os sistemas gerenciadores de banco de dados às bibliotecas de interface com o usuário, não apenas atendem aos requisitos de um registro eletrônico de saúde, mas em alguns casos apresentam-se como opções mais viáveis em relação às soluções proprietárias disponíveis.

Para finalizar, o entendimento de que o desenvolvimento de um registro eletrônico de saúde faz parte de um processo complexo, em que a tecnologia é o menor dos problemas, é essencial para compreender que nenhum registro eletrônico de saúde terá sucesso se os profissionais de saúde que atuam no processo não contribuem para tanto. Porém, sua utilização efetiva não depende somente da postura dos profissionais. Se a tecnologia disponibilizar funcionalidades que contribuam de forma efetiva nas atividades do dia-a-dia destes profissionais, todos irão contribuir para sua implantação.

4.3 Trabalhos futuros

Baseado nos resultados obtidos durante o desenvolvimento do RES proposto, novos trabalhos e pesquisas relacionados à informática em saúde podem ser desenvolvidos. Os resultados apresentados demonstram a capacidade das técnicas utilizadas, assim como a viabilidade do uso de software livre. Partindo desse ponto, alguns trabalhos futuros são observados e definidos:

- desenvolvimento de uma rotina automatizada de testes;
- desenvolvimento de um componente para inclusão de medicamentos;
- integração com a Classificação SIMPRO e Tabela SUS;
- integração com sistema de folha de pagamento;
- planejamento e desenvolvimento de incrementos de comunicação, auditoria, contribuição e documentação;

-
- implantação do RES proposto no Instituto de Ciências Médicas (ICM), instituição vinculada à Universidade Federal do Ceará (UFC).

Engenharia de Software e Modelagem de Sistemas

Desde a criação do primeiro software até os dias atuais têm sido desenvolvidas diversas técnicas que permitem controlar o conjunto de passos necessários para se obter um produto de software. Estas técnicas permitem estimar o custo, tempo de desenvolvimento e garantir determinados níveis de qualidade. A engenharia de software e a modelagem de sistemas contribuem diretamente para este conjunto de técnicas, sendo responsáveis por criar um ambiente propício para o desenvolvimento de sistemas complexos e de grande porte. Este Apêndice aborda os conceitos principais destes dois temas, introduzindo a fundamentação teórica necessária para compreender a sua aplicação no desenvolvimento deste trabalho.

A.1 Engenharia de Software

Atualmente, todos os países do mundo dependem de sistemas complexos baseados em computadores. O gerenciamento operacional, financeiro, educacional, de saúde, entre outros, é cada vez mais realizado por sistemas computacionais. Por esta razão, produzir e evoluir os sistemas dentro de custos adequados é essencial para o funcionamento da economia nacional e internacional (SOMMERVILLE, 2007).

Entretanto, o processo que compreende o desenvolvimento e a evolução de sistemas não é simples. Sua estrutura é composta por um grupo de atividades que são desenvolvidas em paralelo e que implicam em gerenciamento constante, além da necessidade de se prever custos, tempo de desenvolvimento e riscos. Estas atividades

dizem respeito ao desenvolvimento de *hardware*, projetos de políticas, processos e implantação do sistema.

O software, parte integrante do sistema descrito, vêm tomando um papel cada vez mais relevante nas últimas décadas. Este fato se deve principalmente pela diminuição do custo de desenvolvimento de software, além da facilidade na alteração dos requisitos, que ainda é um grande problema no desenvolvimento de sistemas que envolvem *hardware*.

É comum a associação do termo software aos programas de computador. No entanto, esta visão é muito restritiva e não representa corretamente o seu conceito. O software não é apenas o programa, mas inclui todos os dados de documentação e configuração necessários para que o programa funcione corretamente. Portanto, um software é composto de um conjunto de programas, arquivos de configuração, documentação do sistema, documentação do usuário e páginas na Internet que informam os usuários das notícias recentes sobre o software (SOMMERVILLE, 2007).

A.1.1 O que é Engenharia de Software?

Fritz Bauer (BAUER, 1972) foi o pioneiro ao definir a engenharia de software como a criação e utilização de sólidos princípios de engenharia a fim de obter softwares econômicos que sejam confiáveis e que trabalhem eficientemente em máquinas reais.

Como é possível observar, esta definição diz pouco sobre os aspectos técnicos da qualidade de software, não inclui diretamente a necessidade de satisfação do cliente ou pontualidade, entre outros aspectos. Desta forma, ao longo da evolução do software vários autores têm desenvolvido definições pessoais do termo engenharia de software.

Segundo Sommerville (SOMMERVILLE, 2007), a engenharia de software é um ramo da engenharia relacionado com todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a sua evolução, depois que este entrar em operação. Este conceito é mais atual e engloba de modo mais claro a real função da engenharia de software.

O software é abstrato e intangível. Não é limitado por materiais ou controlado por leis da física ou processos de fabricação. De certa forma, estas características simplificam a engenharia de software, pois não existem limitações físicas no seu potencial. Contudo, a falta de restrições naturais significa que o software pode

facilmente se tornar extremamente complexo e, portanto, muito difícil de ser compreendido.

As primeiras discussões a respeito da engenharia de software iniciaram-se em 1968, em um momento em que o desenvolvimento de software estava em crise. Esta crise resultava diretamente da introdução do novo *hardware* de computador baseado em circuitos integrados. Seu poder fez das aplicações de computador, até então consideradas não realizáveis, propostas viáveis. O software era ordens de grandeza maior e mais complexo que sistemas anteriores.

A experiência inicial mostrou que o desenvolvimento informal de software não é suficiente. Projetos importantes apresentavam, algumas vezes, anos de atraso. O software, cujo custo superava as previsões, não era confiável, era difícil de manter e seu desempenho era insatisfatório. Os custos de *hardware* estavam caindo, enquanto os custos de software aumentavam rapidamente. Neste cenário, novas técnicas e métodos foram necessários para controlar a complexidade inerente aos grandes sistemas de software.

Estas técnicas tornaram-se parte da engenharia de software e são amplamente utilizadas hoje em dia. No entanto, assim como aumentou a habilidade de produzir software, cresceu também a necessidade por sistemas de software mais complexos. Novas tecnologias resultantes da convergência de computadores e sistemas de comunicação impuseram novos desafios aos engenheiros de software.

Atualmente, é consenso que não existe uma abordagem única e ideal para todo e qualquer desenvolvimento de software. No entanto, noções fundamentais de processo e de organização de sistemas constituem a base de todas as técnicas que compõem a essência da engenharia de software. Estes conceitos são abordados nas seções a seguir.

A.1.2 Processo de software

Segundo Sommerville (SOMMERVILLE, 2007), um processo de software trata-se do conjunto de atividades e resultados associados que produzem um produto de software. Existem quatro atividades fundamentais, que são comuns a todos os processos de software: especificação, projeto e implementação, validação e evolução.

Na fase de especificação, as funcionalidades do software e as restrições de operação são definidas. Na fase de projeto e implementação, é desenvolvido o

software que atenda às especificações. Na fase de validação, o software é validado para garantir que possui as funcionalidades que o cliente deseja. Na fase de evolução, o software é modificado para atender às demandas mutáveis do cliente.

Estas atividades têm como foco o desenvolvimento de um software de qualidade e dentro dos custos esperados. Algumas características que descrevem um software de qualidade podem ser visualizadas na Tabela A.1.

Característica	Descrição
Facilidade de manutenção	O software deve ser escrito de modo que possa evoluir para atender as necessidades de mudança dos clientes. É um atributo fundamental, pois a mudança de software é uma consequência inevitável de um ambiente de negócios em constante mutação.
Confiança	O nível de confiança do software tem uma série de características, incluindo confiabilidade, proteção e segurança. Um software confiável não deve causar danos físicos ou econômicos no caso de falha.
Eficiência	O software não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. Portanto, a eficiência inclui tempo de resposta, tempo de processamento, utilização de memória, dentre outros.
Usabilidade	O software deve ser usável, sem esforço excessivo, pelo tipo de usuário para o qual foi projetado. Isso significa que o software deve apresentar interface e documentação adequadas.

Tabela A.1: atributos essenciais de um bom software, adaptado de Sommerville (SOMMERVILLE, 2007).

A.1.3 Modelos de processo de software

Os modelos de processo de software definem um conjunto distinto de atividades, ações, tarefas, marcos e produtos de trabalho que são necessários para desenvolver um software de qualidade. Em outras palavras, são uma representação abstrata de um determinado processo de software e descrevem como as atividades são abordadas para se obter um produto de software (PRESSMAN, 2006).

Desta forma, os modelos de processo permitem organizar a sequência em que as atividades são realizadas durante o desenvolvimento de software. Existem diversos modelos de processo de software disponíveis, sendo que o desenvolvimento de um determinado software pode fazer uso de um único modelo ou de um misto de modelos adequados à realidade deste software. A seguir são apresentados alguns modelos conhecidos por modelos genéricos, pois servem de base para muitos outros modelos.

O modelo cascata

O modelo cascata ou modelo clássico, como também é chamado, é o primeiro modelo de processo de desenvolvimento de software conhecido. É originado dos processos gerais de engenharia de sistemas, no qual existe um conjunto de fases que são executadas sequencialmente. As principais etapas deste modelo são descritas na Tabela A.2.

Etapa	Descrição
Análise e definição de requisitos	Os serviços, restrições e objetivos do sistema são definidos por meio de consulta aos usuários do sistema. Portanto, são definidos detalhadamente e servem como uma especificação de sistema.
Projeto de sistema e software	O processo de projeto de sistema divide os requisitos em sistemas de <i>hardware</i> ou de software, estabelecendo uma arquitetura geral do sistema.
Implementação e teste de unidade	Durante este estágio, o projeto de software é realizado como um conjunto de programas ou unidades de programa. O teste unitário envolve a verificação de que cada unidade atende à sua especificação.
Integração e teste de sistema	As unidades individuais de programa ou os programas são integrados e testados como um sistema completo para garantir que os requisitos de software foram atendidos. Após os testes, o sistema de software é liberado para o cliente.
Operação e manutenção	Geralmente esta é a fase mais longa do ciclo de vida do software, pois envolve a correção de erros não detectados em ambiente de desenvolvimento, aprimoramento da implementação das unidades do sistema e a ampliação dos serviços de sistema à medida que novos requisitos são identificados.

Tabela A.2: etapas do modelo cascata, adaptado de Sommerville (SOMMERVILLE, 2007).

As vantagens deste modelo consistem na documentação produzida em cada fase e sua aderência a outros modelos de processo de engenharia. Seu maior problema é a divisão inflexível do projeto em estágios distintos. Esta estrutura dificulta a mudança dos requisitos do sistema, pois a sua alteração implica na execução de todas as etapas que precedem a atual e regerar toda a documentação e material produzido até o momento, de forma que os mesmos contemplem as modificações realizadas. Além disso, o cliente inicia a utilização do sistema apenas no final do processo. O fluxo de trabalho utilizado no modelo cascata pode ser visualizado na Figura A.1.

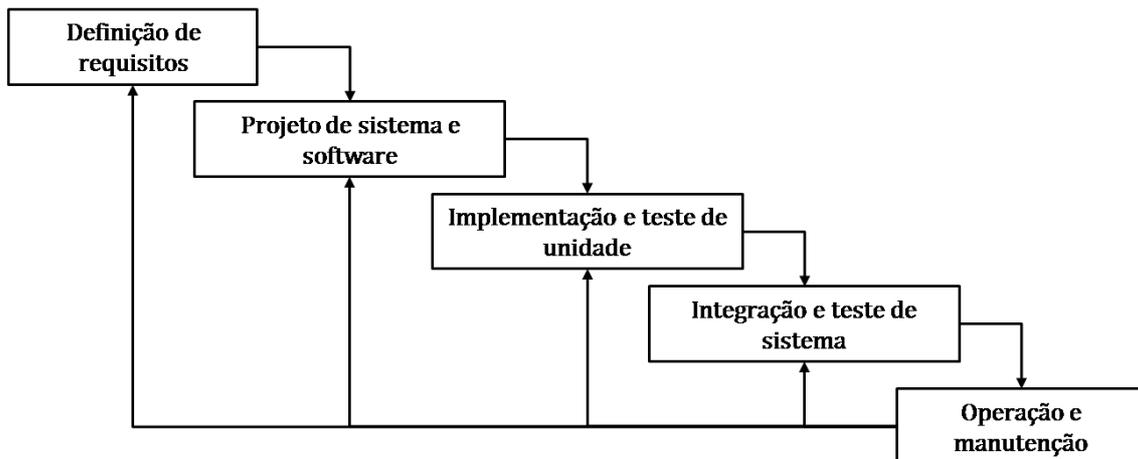


Figura A.1: modelo cascata.

Portanto, o modelo cascata deve ser utilizado somente quando os requisitos do software são muito bem compreendidos e há pouca probabilidade de mudanças radicais durante o desenvolvimento.

Desenvolvimento evolucionário

O desenvolvimento evolucionário engloba os modelos que baseiam-se na idéia de desenvolvimento de uma implementação inicial, expondo os resultados aos comentários do cliente e refinando esse resultado por meio de várias versões até que seja desenvolvido um sistema adequado. A estrutura de um modelo evolucionário pode ser visualizada na Figura A.2.

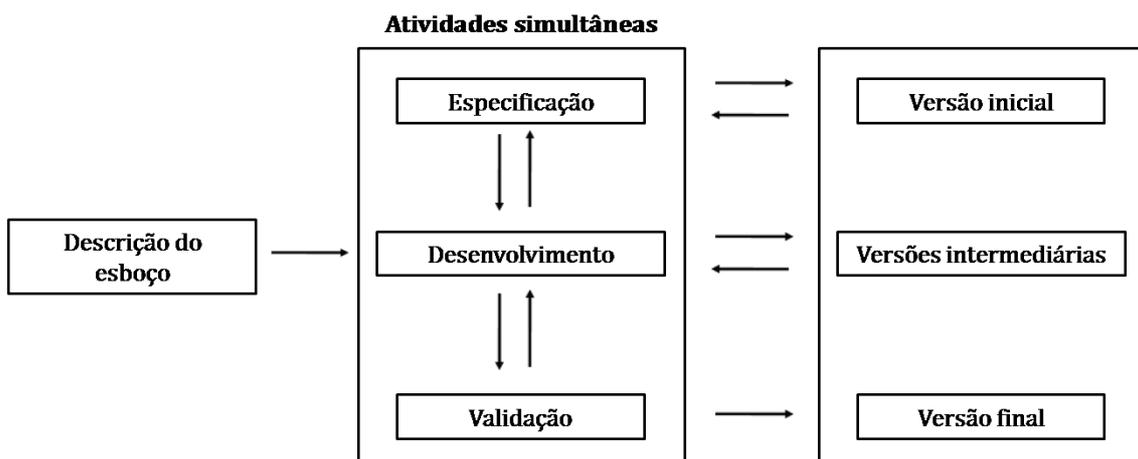


Figura A.2: modelo evolucionário.

Esta abordagem é geralmente mais eficaz do que a abordagem em cascata

na produção de sistemas que atendam às necessidades imediatas do cliente. A vantagem de um processo de software baseado na abordagem evolucionária é que a especificação pode ser desenvolvida à medida que os usuários compreendem melhor o seu problema.

Entretanto, do ponto de vista da engenharia e do gerenciamento, esta abordagem possui dois problemas principais. O primeiro trata que o processo não é visível. Os gerentes precisam de produtos regulares para medir o processo. Se os sistemas são desenvolvidos rapidamente, não é viável economicamente produzir documentos que reflitam cada versão do sistema. O segundo trata do frequente problema na estruturação do software. A mudança contínua tende a corromper a estrutura do software. Desta forma, a incorporação de mudanças torna-se cada vez mais difícil e onerosa.

Dentre os modelos evolucionários, alguns merecem destaque: prototipagem e modelo espiral.

O modelo de prototipagem é utilizado geralmente em dois casos: quando o cliente define o software apenas em linhas gerais, mas não identifica detalhadamente os requisitos de entrada, saída e processamento, ou quando o engenheiro de software está inseguro da eficiência de um algoritmo, da adaptabilidade de um sistema operacional ou da forma que a interação entre homem e máquina deve ocorrer.

Em ambos os casos, o engenheiro de software faz uso de protótipos desenvolvidos rapidamente com as partes já compreendidas do sistema e entrega-os ao cliente para avaliação. De acordo com o *feedback* do cliente, os requisitos são refinados de forma a compreender os objetivos do software por completo. Os protótipos são, então, descartados. Sua principal vantagem está relacionada com a utilização do sistema por parte do usuário desde as etapas iniciais do desenvolvimento. Porém, esta vantagem torna-se uma desvantagem quando o cliente imagina que já possui a funcionalidade completamente desenvolvida, enquanto possui apenas um protótipo funcionando de forma precária e sem interligação com as outras partes do sistema. O fluxo de desenvolvimento do modelo prototipagem pode ser visualizado na Figura A.3.

O modelo espiral foi proposto originalmente por Boehm (BOEHM, 1988). Ao invés de representar o processo de software como uma sequência de atividades com algum retorno entre uma atividade e outra, o processo é representado como uma espiral. Cada volta na espiral representa uma fase do processo de software. Desta forma,

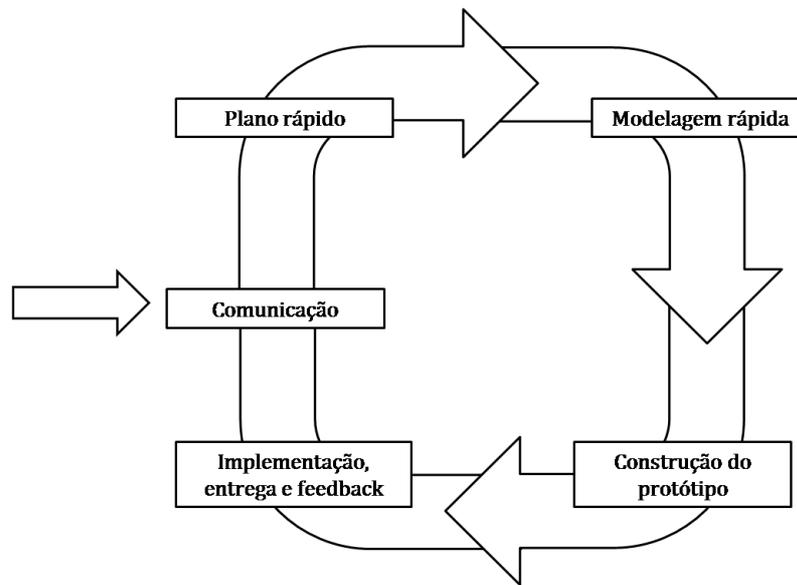


Figura A.3: prototipagem, adaptado de Pressmann (PRESSMAN, 2006).

a volta mais interna pode estar relacionada à viabilidade do sistema; a próxima à definição de requisitos; a próxima ao projeto de sistema e assim por diante.

A principal diferença entre o modelo espiral e os outros modelos de processo de software é a inclusão do gerenciamento de riscos. Informalmente, risco significa algo que pode dar errado. Os riscos podem causar problemas no desenvolvimento do projeto, tal como ultrapassar o cronograma e os custos. Por este motivo, a minimização dos riscos é uma atividade de gerenciamento de projeto muito importante. O fluxo de desenvolvimento do modelo espiral pode ser visualizado na Figura A.4.

Desenvolvimento incremental

A mudança é inevitável em todos os projetos de grande porte. Os requisitos do sistema mudam à medida que a empresa responde a pressões externas e internas. As prioridades de gerenciamento mudam. Quando novas tecnologias se tornam disponíveis, projetos e implementações mudam. Isto significa que o processo de software não é de execução única, mas ao contrário, as atividades são repetidas regularmente à medida que o sistema é retrabalhado (SOMMERVILLE, 2007).

O modelo cascata requer que os clientes de um sistema se comprometam com um conjunto de requisitos antes do início do projeto e o projetista se comprometa com estratégias específicas de projeto antes da implementação. Mudanças nos requisitos

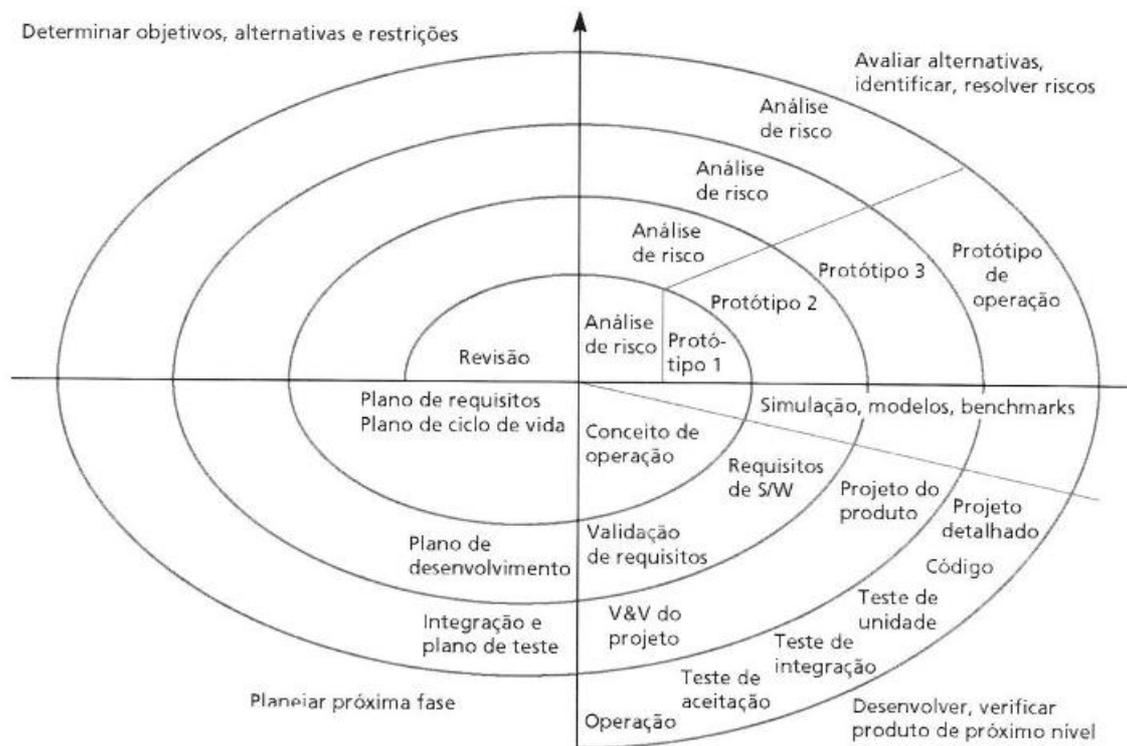


Figura A.4: modelo espiral, adaptado de Boehm (BOEHM, 1988).

requerem retrabalho na definição de requisitos, do projeto e da implementação. Contudo, a separação e a implementação do projeto devem levar a sistemas bem documentados sujeitos à mudanças. Por outro lado, uma abordagem evolucionária para o desenvolvimento permite que as decisões de requisitos e projeto sejam postergadas, mas também pode levar frequentemente a um software mal estruturado e difícil de compreender e manter.

O desenvolvimento incremental é uma abordagem intermediária que combina as vantagens destes dois modelos. Em um processo de desenvolvimento incremental, o cliente identifica, em linhas gerais, os serviços a serem fornecidos pelo sistema. Neste momento são definidos os requisitos mais importantes e menos importantes. Assim, um número de incrementos é definido, com cada incremento fornecendo um subconjunto de funcionalidades do sistema. A alocação de serviços aos incrementos depende da prioridade do serviço, com os serviços de alta prioridade sendo entregues primeiro.

Após a identificação dos incrementos do sistema, os requisitos dos serviços a serem entregues no primeiro incremento são definidos detalhadamente e desenvolvidos. Durante o desenvolvimento, pode ser realizada a análise dos próximos

requisitos para os incrementos posteriores, mas não são aceitas mudanças de requisitos para o incremento atual.

Depois que um incremento é concluído e entregue, o cliente pode colocá-lo em operação. Isto significa que desde o início do processo o cliente tem contato com as partes do sistema já desenvolvidas. Desta forma, podem experimentá-las e obter melhores esclarecimentos dos requisitos dos próximos incrementos.

O processo de desenvolvimento incremental possui uma série de vantagens:

1. Os clientes não precisam esperar até a entrega do sistema inteiro para se beneficiarem dele. O primeiro incremento satisfaz os requisitos mais críticos e, desta forma, é possível utilizar o software imediatamente.
2. Os clientes podem utilizar os incrementos iniciais como protótipos e ganhar experiência, obtendo informações sobre os requisitos dos incrementos posteriores do sistema.
3. Existe um risco menor de falha geral do projeto. Embora possam ser encontrados problemas em alguns incrementos, é provável que alguns sejam entregues com sucesso aos clientes.
4. Como os serviços de prioridade mais alta são entregues primeiro e os incrementos posteriores são integrados, é inevitável que os serviços mais importantes do sistema recebam mais testes. Isto significa que os clientes têm menor probabilidade de encontrar falhas de software nas partes mais importantes do sistema.

No entanto, alguns problemas relativos ao mapeamento dos incrementos e as funcionalidades que cada um contém podem ser encontrados. Entretanto, pelas vantagens apresentadas e por se adequar perfeitamente ao cenário existente, este modelo é o escolhido para o desenvolvimento deste trabalho.

Desenvolvimento baseado em componentes

O reuso ocorre em grande parte dos projetos de software. Na maioria dos casos, isto ocorre quando as pessoas que trabalham no projeto conhecem códigos-fonte similares aos necessários. Estas pessoas procuram por esses produtos, os modificam e os incorporam ao sistema. Na abordagem evolucionária, por exemplo, o reuso

é geralmente essencial para o desenvolvimento rápido do sistema (SOMMERVILLE, 2007).

Este reuso informal ocorre independentemente do modelo de processo. No entanto, uma abordagem para desenvolvimento de software baseada no reuso, chamada de desenvolvimento baseado em componentes, tem emergido e se tornado cada vez mais utilizada.

O desenvolvimento baseado em componentes depende de uma grande base de componentes de software reusáveis e algum *framework* de integração. Algumas vezes, estes componentes são sistemas comerciais independentes que podem fornecer uma funcionalidade específica, como a formatação de texto ou um cálculo numérico. Os estágios que compõem este modelo de processo são descritos a seguir.

1. Análise de componentes: partindo de uma especificação de requisitos, é realizada uma busca pelos componentes que implementam as funcionalidades necessárias. Geralmente, não existe uma correspondência exata e os componentes que podem ser utilizados fornecem apenas parte da funcionalidade necessária.
2. Modificação de requisitos: durante este estágio, os requisitos são analisados usando as informações sobre os componentes encontrados. Em seguida, são modificados para refletir os componentes disponíveis. Quando as modificações não são possíveis, a atividade de análise de componentes pode ser novamente realizada para procurar soluções alternativas.
3. Projeto de sistema com reuso: neste estágio o *framework* do sistema é projetado ou um *framework* existente é reusado. Os projetistas levam em consideração os componentes reusados, organizando o *framework* para o contexto desejado. Pode ser necessário projetar algum novo software caso os componentes reusáveis não estejam disponíveis.
4. Desenvolvimento e integração: o software que não pode ser adquirido externamente é desenvolvido e integrado ao novo sistema. A integração, neste modelo, pode ser parte do processo de desenvolvimento, em vez de ser uma atividade separada.

Este modelo tem a vantagem de diminuir a quantidade de software a ser desenvolvido, reduzindo os riscos e custos. Isto geralmente leva a uma entrega mais

rápida. Entretanto, algumas vezes compromissos com os requisitos são inevitáveis e isto pode levar a um sistema que não atenda às reais necessidades dos usuários. Além disso, algum controle sobre a evolução do sistema é perdido se novas versões dos componentes reusáveis não estiverem sob controle da organização que as utiliza.

A.1.4 Requisitos de software

Os requisitos de um sistema são descrições dos serviços oferecidos pelo sistema e as suas restrições operacionais. Esses requisitos refletem as necessidades dos clientes de um sistema que ajuda a resolver algum problema. Apesar de simples, o conceito do termo requisito não é usado pela indústria de software de maneira consistente. Em alguns casos, um requisito é simplesmente uma declaração abstrata de alto nível de um serviço que o sistema deve oferecer ou uma restrição do sistema. No outro extremo, é uma definição formal e detalhada de uma função do sistema (SOMMERVILLE, 2007).

Por este motivo, os requisitos podem ser divididos em dois grupos principais: requisitos de usuário e requisitos de sistema. Os requisitos de usuário são declarações em linguagem natural de quais serviços são esperados e as restrições sob as quais deve operar. Os requisitos de sistema definem detalhadamente as funções, serviços e as restrições operacionais do sistema. O documento de requisitos de sistema deve ser preciso, devendo definir exatamente as funcionalidades desejadas, podendo fazer parte do contrato entre o comprador do sistema e os desenvolvedores de software.

Os diferentes níveis são úteis porque comunicam informações sobre o sistema a diferentes tipos de leitores. Os leitores dos requisitos de usuário geralmente não estão preocupados com o modo como o sistema é implementado e podem ser gerentes que não estão interessados nos recursos detalhados do sistema. Os leitores dos requisitos de sistema necessitam conhecer mais precisamente o que o sistema faz, pois estão preocupados com o modo como o sistema apóia os processos de negócio ou porque estão envolvidos na implementação do sistema. Um exemplo dos dois tipos de requisitos pode ser visualizado na Tabela A.3.

A classificação de requisitos de usuário e de sistema é muito geral. Por este motivo, os requisitos de sistema de software são frequentemente classificados em requisitos funcionais, não funcionais e de domínio.

Tipo de requisito	Requisitos
Usuário	1. O LIBSYS deve manter o acompanhamento de todos os dados exigidos pelas agências de licenciamento de direitos autorais no Reino Unido e em outros lugares.
Sistema	1. Ao solicitar um documento ao LIBSYS, deve ser apresentado ao solicitante um formulário que registra os detalhes do usuário e da solicitação realizada. 2. Os formulários de solicitação do LIBSYS devem ser armazenados no sistema durante cinco anos, a partir da data da solicitação. 3. Todos os formulários do LIBSYS devem ser indexados por usuário, nome do material solicitado e fornecedor da solicitação. 4. O LIBSYS deve manter um registro de todas as solicitações feitas ao sistema. 5. Para materiais aos quais se aplicam os direitos de empréstimo dos autores, os detalhes do empréstimo devem ser enviados mensalmente às agências de licenciamento de direitos autorais que se registraram no LIBSYS.

Tabela A.3: requisitos de usuário e de sistema, adaptado de Sommerville (SOMMERVILLE, 2007).

Requisitos funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer. Estes requisitos dependem do tipo do software que está sendo desenvolvido, dos usuários a que o software se destina e da abordagem geral considerada pela organização ao redigir os requisitos. Os requisitos funcionais descrevem a função do sistema detalhadamente, suas entradas e saídas, exceções, entre outras.

Requisitos não funcionais

Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema. Podem estar relacionados às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e espaço de armazenamento. Como alternativa, podem definir restrições, como capacidade dos dispositivos de entrada e saída e as representações de dados usadas nas interfaces do sistema. Em outras palavras, pode-se afirmar que os requisitos não funcionais restringem os requisitos funcionais.

Requisitos de domínio

Os requisitos de domínio são derivados do domínio de aplicação do sistema, em vez das necessidades específicas dos usuários do sistema. Geralmente incluem

uma terminologia específica de domínio ou fazem referência a conceitos do domínio. Podem ser novos requisitos funcionais em si mesmos, podem restringir os requisitos funcionais existentes ou estabelecer como cálculos específicos devem ser realizados. Como esses requisitos são especializados, os engenheiros de software frequentemente encontram dificuldades em compreender como estão relacionados a outros requisitos do sistema.

Documento de requisitos

O documento de requisitos de software, também chamado de especificação de requisitos de software, é a declaração oficial do que os desenvolvedores de sistema devem implementar. Deve incluir os requisitos de usuário e uma especificação detalhada dos requisitos de sistema.

O nível de detalhamento a ser incluído em um documento de requisitos depende do tipo de sistema que está sendo desenvolvido e do modelo de processo utilizado. Quando o sistema for desenvolvido por um fornecedor externo, as especificações devem ser precisas e muito detalhadas. Quando houver maior flexibilidade nos requisitos e quando um processo de desenvolvimento interno e iterativo for utilizado, o documento de requisitos pode ser menos detalhado e qualquer ambiguidade é resolvida durante o desenvolvimento do sistema.

Um dos padrões mais conhecidos de documento de requisitos é o IEEE/ANSI 830-1998 (IEEE, 1998) e pode ser visualizado a seguir:

1 Introdução

- 1 Propósito do documento de requisitos
- 2 Escopo do produto
- 3 Definições, acrônimos e abreviaturas
- 4 Referências
- 5 Visão geral do restante do documento

2 Descrição geral

- 1 Perspectiva do produto
- 2 Funções do produto
- 3 Características dos usuários

4 Restrições gerais

5 Suposições e dependências

3 Requisitos específicos

Abrangem os requisitos funcionais, não funcionais e de domínio. Esta é a parte mais substancial do documento, mas devido à grande variação na prática organizacional, não é apropriado definir uma estrutura padrão para esta seção. Os requisitos podem documentar interfaces externas, descrever a funcionalidade e o desempenho do sistema, especificar os requisitos lógicos de banco de dados, restrições de projeto, propriedades emergentes do sistema e características de qualidade.

4 Apêndices

5 Índice

Embora o padrão IEEE/ANSI 830-1998 não seja o ideal, contém boas recomendações de como redigir requisitos e evitar problemas. Normalmente, este padrão é muito geral para funcionar sem modificações em todos os projetos de uma organização, configurando-se como um *framework* que pode ser estendido para incluir as necessidades específicas do software. Não existe um modelo ou norma que defina um documento de requisitos ideal, pois as necessidades variam muito dependendo do software que está sendo desenvolvido. A tarefa de adaptação e customização dos modelos existentes faz parte da atividade de engenharia de software e é necessária para permitir um entendimento completo dos requisitos do sistema.

A.2 Modelagem de Sistemas

Para entender corretamente o conceito de modelagem de sistemas, faz-se necessário compreender que o principal produto de uma equipe de desenvolvimento de software não são documentos bonitos, reuniões sofisticadas ou linhas de código elegantes. O principal produto é um bom software capaz de satisfazer às necessidades de seus usuários e seus respectivos negócios. Tudo mais é secundário (BOOCH; RUMBAUGH; JACOBSON, 2005).

Entretanto, para atingir o produto descrito é necessário que ocorram reuniões para interagir com os usuários e definir os requisitos corretamente. Para que o

software permita alterações futuras com baixo custo, será necessário criar uma arquitetura que permita tal tarefa. Para desenvolver o software de forma rápida, com o mínimo de desperdício e de retrabalho, é necessário dispor das pessoas certas, das ferramentas adequadas e do enfoque correto. Para realizar todas estas tarefas de maneira previsível e consistente, com uma avaliação dos custos reais do sistema, é necessário um processo seguro de desenvolvimento que possa ser adaptado às novas necessidades do software e da tecnologia utilizada. Todas estas atividades fazem parte da modelagem de sistemas.

A.2.1 O que é Modelagem de Sistemas?

Segundo Booch (BOOCH; RUMBAUGH; JACOBSON, 2005), a modelagem é a parte central de todas as atividades que levam à implantação de um bom software. Constitui-se de uma técnica de engenharia aprovada e bem aceita na qual são desenvolvidos modelos para auxiliar o entendimento do produto final. Estes modelos fornecem uma descrição do sistema muitas vezes não passível de reconhecimento sem análise prévia, permitindo que os próprios analistas e clientes visualizem o produto final antes da sua entrega.

Os modelos podem compreender planos detalhados, assim como planos mais gerais. Um bom modelo inclui componentes que têm ampla repercussão e omite os componentes menores que não são relevantes em um determinado nível de abstração. Todos os sistemas podem ser descritos sob diferentes aspectos, com a utilização de modelos distintos e cada modelo se torna, então, uma abstração semanticamente específica do sistema.

Os modelos podem ser estruturais, dando ênfase à organização do sistema, ou podem ser comportamentais, dando ênfase à dinâmica do sistema. De forma sucinta, é possível afirmar que os modelos são criados para compreender melhor o sistema que está sendo desenvolvido.

A.2.2 Princípios da Modelagem de Sistemas

Segundo Booch (BOOCH; RUMBAUGH; JACOBSON, 2005), a utilização da modelagem em todas as disciplinas de engenharia possui uma história longa e rica. A experiência obtida durante este período sugere que toda modelagem deve seguir quatro princípios básicos, descritos a seguir.

1 - A escolha dos modelos a serem criados tem profunda influência sobre a

maneira como um determinado problema é atacado e como uma solução é definida.

O primeiro princípio afirma que os modelos devem ser escolhidos corretamente, pois os modelos corretos iluminam de modo brilhante os problemas de desenvolvimento mais complicados, proporcionando conclusões que simplesmente não são possíveis de outra maneira. Por outro lado, a escolha de modelos inadequados pode causar confusão, desviando a atenção para questões irrelevantes.

2 - Cada modelo poderá ser expresso em diferentes níveis de precisão.

Dado que existem diversos tipos de pessoas envolvidas durante o desenvolvimento de um software complexo, cada uma com um nível específico de conhecimento, é justo que existam modelos que traduzam a estrutura e o comportamento do sistema em uma linguagem passível de entendimento por todos os envolvidos. Em qualquer situação, os melhores tipos de modelos serão aqueles que permitem a escolha do grau de detalhamento, dependendo de quem esteja fazendo a visualização.

3 - Os melhores modelos estão relacionados à realidade.

O modelo físico de um prédio que não responde da mesma forma que os materiais reais tem um valor limitado. O modelo matemático de um avião em que são consideradas apenas condições de vôo ideais e fabricação perfeita, poderá ocultar características potencialmente fatais do avião real. A questão a ser abordada é que todos os modelos simplificam a realidade, mas o segredo está em ter certeza que esta simplificação não oculta detalhes importantes, que venham a gerar problemas.

4 - Nenhum modelo único é suficiente. Qualquer sistema não trivial será melhor investigado por meio de um pequeno conjunto de modelos quase independentes com vários pontos de vista.

No desenvolvimento de um software orientado a objetos, é necessário recorrer a diversas visões de modelos que representem o software nos mais diversos aspectos, desde a estrutura principal até o modelo de armazenamento das informações em um banco de dados. Desta forma, é possível concluir que não existe um único modelo que apresente as informações da melhor forma possível e para todos os tipos de interesses, sendo mais interessante possuir um grupo de modelos que apresentem estas diferentes visões.

A.2.3 Tipos de modelos

Cada tipo de modelo é organizado de modo diferente e possui o seu próprio foco. Para o domínio de desenvolvimento de software, existem várias maneiras de se definir um modelo. As duas maneiras mais comuns são provenientes da perspectiva de um algoritmo ou da perspectiva orientada a objetos.

A visão tradicional do desenvolvimento de software adota a perspectiva de um algoritmo. Nesta visão, o principal bloco de construção do software é o procedimento ou a função. Esta perspectiva conduz os desenvolvedores a voltar seu foco de atenção para questões referentes ao controle e a decomposição de algoritmos maiores em outros menores.

A visão contemporânea no desenvolvimento de software adota uma perspectiva orientada a objetos. Nesta visão, o principal bloco de construção de todos os sistemas de software é o objeto ou a classe. Esta abordagem tem se mostrado valorosa para construção de sistemas em todos os tipos de domínios de problemas, abrangendo todos os graus de tamanho e de complexidade. Além disso, muitas linguagens, sistemas operacionais e ferramentas de desenvolvimento são, de alguma forma, orientados a objetos. Este fato fortalece a visão de mundo em termos de objetos (BOOCH; RUMBAUGH; JACOBSON, 2005).

A.2.4 O que é UML?

A *Unified Modeling Language* (UML) é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de orientação a objetos. Esta linguagem tornou-se um padrão de modelagem de software nos últimos anos, sendo adotada internacionalmente pela engenharia de software (GUEDES, 2008).

É importante salientar que a UML não é uma linguagem de programação e sim uma linguagem de modelagem, cujo objetivo é auxiliar os engenheiros de software a definirem as características do software, tais como seus requisitos, comportamento, estrutura lógica, dinâmica de processos e necessidades específicas em relação ao equipamento sobre o qual o sistema deverá ser implantado. Tais características são definidas por meio da UML antes do software começar a ser implementado.

A.2.5 Como chegamos à UML?

A UML surgiu da união de três métodos de modelagem: o método de Booch, o método *Object Modeling Technique* (OMT) de Jacobson e o método *Object-Oriented Software Engineering* (OOSE) de Rumbaugh. Estas técnicas eram, até meados da década de 1990, os métodos de modelagem orientada a objetos mais populares entre os profissionais da área de desenvolvimento de software. A união destes métodos contou com amplo apoio da Rational Software, que a incentivou e a financiou (BOOCH; RUMBAUGH; JACOBSON, 2005).

O esforço inicial do projeto se deu com a união do método de Booch ao OMT de Jacobson, que resultou no lançamento do Método Unificado, no final de 1995. Em seguida, Rumbaugh juntou-se a Booch e Jacobson na Rational Software e seu método OOSE foi incorporado à nova metodologia. Estas três técnicas, juntas, deram origem à primeira versão da UML em 1996 (GUEDES, 2008).

Após o lançamento da primeira versão da UML, muitas empresas atuantes na área de modelagem e desenvolvimento de software passaram a contribuir com o projeto, fornecendo sugestões para melhorar e ampliar a linguagem. Finalmente, em 1997, a UML foi adotada pelo *Object Management Group* (OMG) como uma linguagem padrão de modelagem. Até os dias de hoje, a UML tem sofrido diversas modificações e encontra-se na versão 2.0 (GUEDES, 2008).

A.2.6 Os diagramas

A UML é composta por diversos diagramas. O principal objetivo é fornecer múltiplas visões do sistema a ser modelado, analisando-o e modelando-o sob diversos aspectos, procurando desta forma atingir a completude da modelagem e permitir que cada diagrama complemente a visão de outros (GUEDES, 2008). A lista de diagramas disponíveis na UML 2.0 pode ser visualizada na Figura A.5.

Para o desenvolvimento deste trabalho, apenas alguns dos diagramas disponíveis na UML são utilizados. Suas escolhas foram determinadas pela relevância do diagrama no contexto em que a aplicação está inserida e no objetivo de manter a modelagem mais clara possível. As seções a seguir descrevem os diagramas que são utilizados na modelagem deste trabalho.

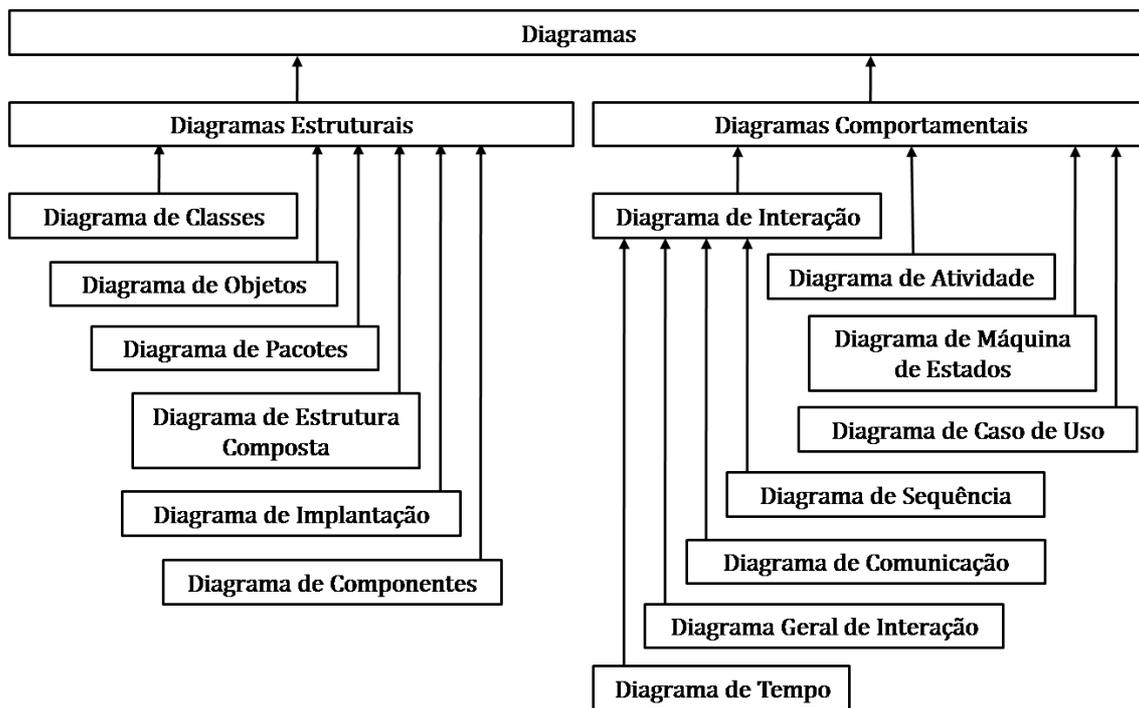


Figura A.5: diagramas da UML 2.0, adaptado de Guedes (GUEDES, 2008).

A.2.7 Diagrama de casos de uso

O diagrama de casos de uso permite entender conceitualmente o conjunto de funções que o sistema deve executar para atender os requisitos especificados, capturando e modelando os requisitos funcionais em uma linguagem muito mais acessível ao público em geral. Em outras palavras, o sistema é descrito sob a visão do usuário. Um exemplo de diagrama de casos de uso pode ser visualizado na Figura A.6.

Cada diagrama possui três elementos básicos: ator (homem-palito), caso de uso (elipse) e relacionamento (reta). Os atores representam as entidades externas que se relacionam com o sistema. Os casos de uso definem funcionalidades que os atores podem solicitar para o sistema. Os relacionamentos fazem a ligação dos atores com os casos de uso e entre casos de uso. Para complementar o diagrama, deve ser desenvolvida uma tabela para cada caso de uso, de forma que possam ser identificados os atores envolvidos, pré-condições, pós-condições, fluxo principal de ações do caso de uso e fluxos alternativos, caso alguma exceção que bloqueie o fluxo principal aconteça.

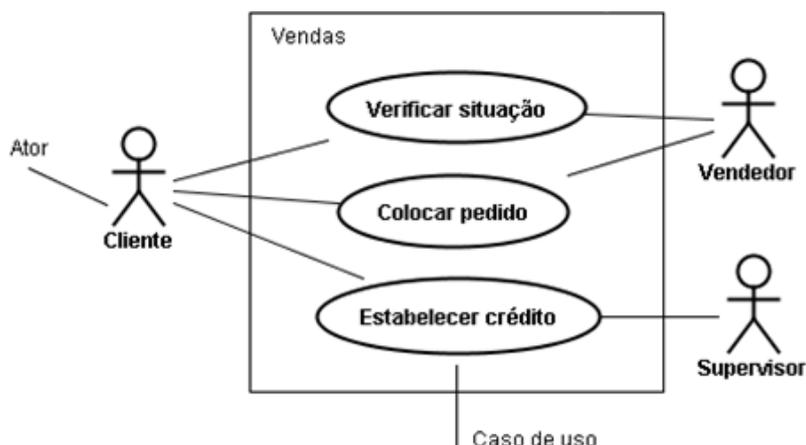


Figura A.6: exemplo de diagrama de caso de uso.

A.2.8 Diagrama de classes

O diagrama de classes é o mais utilizado e mais importante da UML, pois permite visualizar as classes que compõem o sistema e como se relacionam entre si. É composto por dois elementos básicos: classe e relacionamento. As classes definem os métodos e atributos, enquanto os relacionamentos definem a quantidade de instâncias de cada classe que participam de cada relacionamento. Existem vários tipos de relacionamento: associação, generalização, dependência e realização. Um exemplo de diagrama de classes pode ser visualizado na Figura A.7.

A.2.9 Diagrama de atividade

O diagrama de atividade é o diagrama com maior ênfase no nível de algoritmo da UML, preocupando-se em descrever os passos a serem percorridos para a conclusão de um método ou algoritmo específico. É composto pelos elementos a seguir: estado inicial, estado final, transições, estado de ação, ponto de decisão, barra de sincronização, estado de subatividade, fluxo de objetos e raias de natação. Um exemplo de diagrama de atividade pode ser visualizado na Figura A.8.

Para finalizar, todas as informações apresentadas neste Apêndice têm como objetivo disponibilizar a fundamentação teórica necessária para compreender as técnicas utilizadas para planejar e modelar o desenvolvimento deste trabalho.

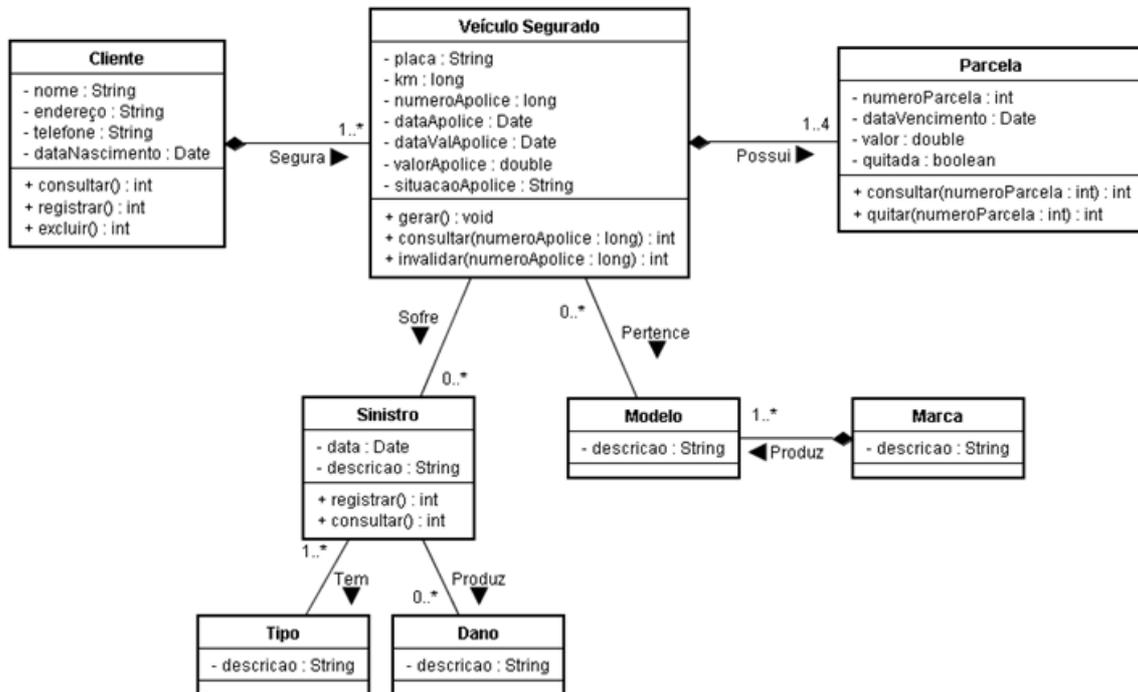


Figura A.7: exemplo de diagrama de classes.

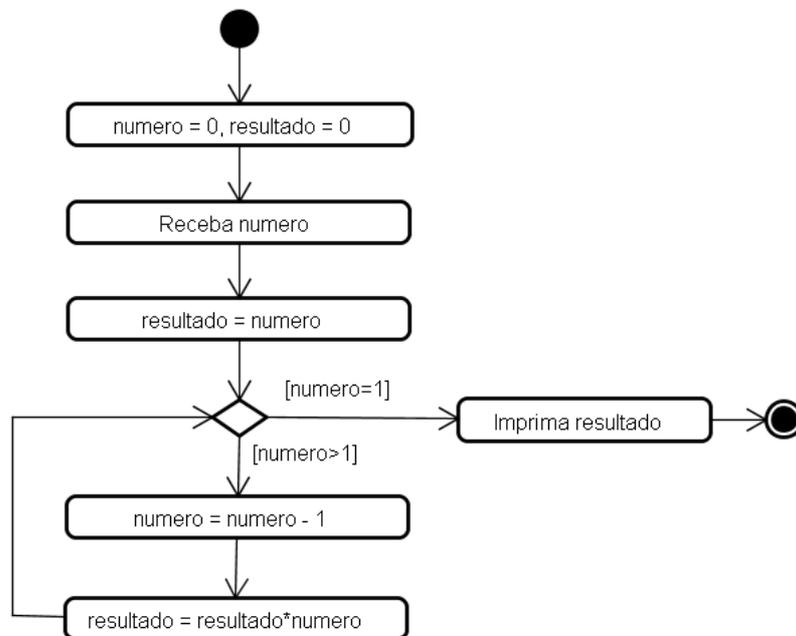


Figura A.8: exemplo de diagrama de atividades.

Apêndice **B**

Modelagem de Casos de Uso

Este apêndice tem como objetivo especificar e detalhar os casos de uso identificados no RES proposto. Para cada caso de uso identificado, é criada uma tabela descritiva formada pelos campos a seguir.

1. Atores: lista de atores que participam do caso de uso.
2. Descrição: síntese das ações realizadas no caso de uso.
3. Pré-condições: lista de condições que devem estar satisfeitas antes da execução do caso de uso.
4. Pós-condições: estado esperado ao final da execução do caso de uso.
5. Fluxo principal: caminho natural de execução do caso de uso.
6. Fluxos alternativos: caminhos alternativos de execução do caso de uso.
7. Fluxos de exceção: caminhos executados quando ocorrem exceções nos fluxos anteriores.

Assim como a especificação de requisitos, o detalhamento dos casos de uso está agrupado de acordo com o incremento em que estão inseridos, conforme pode ser visualizado nas seções a seguir.

B.1 Incremento 01 - autenticação e autorização

B.1.1 Identificação e especificação dos casos de uso

Tabela B.1: especificação dos casos de uso do incremento 01.

Caso de uso: [UC001] Efetuar autenticação	
Atores:	Usuário.
Descrição:	O usuário deve realizar sua autenticação antes de executar qualquer ação no RES.
Pré-condições:	O usuário deve estar previamente cadastrado.
Pós-condições:	Usuário autenticado e permissões carregadas.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de autenticação. 2. Usuário preenche os campos usuário, senha e <i>captcha</i>. 3. RES verifica se usuário existe. 4. RES valida número de tentativas de autenticação com falha. 5. RES valida senha. 6. RES valida se usuário está ativo. 7. RES verifica se usuário atingiu data limite de atualização de senha. 8. RES efetua a autenticação. 9. RES atualiza dados de autenticação. 10. RES exibe página inicial do usuário.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. Usuário atingiu data limite de atualização de senha (7): <ol style="list-style-type: none"> a. O usuário é redirecionado para atualizar sua senha (UC002).
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação (3, 4, 5, 6): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao usuário.
Caso de uso: [UC002] Alterar senha	
Atores:	Usuário.
Descrição:	O usuário pode alterar sua senha quando for conveniente ou quando a data limite de atualização de senha for atingida.
Pré-condições:	O usuário deve estar autenticado no RES.
Pós-condições:	Senha atualizada.

Continua na próxima página

Tabela B.1 – continuação da página anterior

Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de alteração de senha. 2. Usuário preenche senha anterior, nova senha e confirmação de nova senha. 3. RES verifica qualidade da nova senha. 4. RES valida senha anterior e nova senha. 5. RES atualiza data da próxima alteração. 6. RES salva nova senha.
Fluxos alternativos:	Nenhum fluxo alternativo definido.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação (3, 4): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao usuário.
Caso de uso: [UC003] Manter dados pessoais	
Atores:	Usuário.
Descrição:	O usuário pode visualizar e alterar os seus dados pessoais.
Pré-condições:	O usuário deve estar autenticado no RES.
Pós-condições:	Os dados do usuário estão atualizados.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de dados pessoais do usuário. 2. Usuário visualiza seus dados pessoais. 3. Usuário altera as informações desejadas. 4. RES valida as informações inseridas. 5. RES atualiza os dados alterados.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. Os dados já se encontram atualizados (2): <ol style="list-style-type: none"> a. Caso o usuário não encontre dados para atualizar, o caso de uso é encerrado.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação (4): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao usuário.

Continua na próxima página

Tabela B.1 – continuação da página anterior

Caso de uso: [UC004] Manter endereços	
Atores:	Usuário.
Descrição:	Um usuário pode possuir vários endereços. Cadastro, alteração e exclusão devem estar disponíveis.
Pré-condições:	O usuário deve estar autenticado no RES.
Pós-condições:	Os endereços do usuário estão atualizados.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de endereços do usuário. 2. Usuário visualiza seus endereços. 3. Usuário pode alterar, excluir ou criar um novo endereço. 4. Após a realização de uma destas ações, o caso de uso é encerrado.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. Os dados já se encontram atualizados (2): <ol style="list-style-type: none"> a. Caso o usuário não encontre dados para atualizar, o caso de uso é encerrado. 2. Usuário seleciona um endereço para alterar (3): <ol style="list-style-type: none"> a. RES gera tela de alteração. b. Usuário preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza o endereço. 3. Usuário seleciona um endereço para excluir (3): <ol style="list-style-type: none"> a. RES gera tela de exclusão. b. Usuário confirma a exclusão. c. RES exclui o endereço. 4. Usuário deseja cadastrar um novo endereço (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Usuário informa os dados necessários. c. RES valida as informações. d. RES cadastra o novo endereço.

Continua na próxima página

Tabela B.1 – continuação da página anterior

Fluxos de exceção:	<ol style="list-style-type: none">1. Erro na validação - fluxos alternativos (2.c, 4.c):<ol style="list-style-type: none">a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema.b. A mensagem é exibida ao usuário.
Caso de uso: [UC005] Manter permissões	
Atores:	Administrador.
Descrição:	O administrador pode criar, alterar e excluir permissões.
Pré-condições:	O administrador deve estar autenticado no RES.
Pós-condições:	As permissões estão atualizadas.
Fluxo principal:	<ol style="list-style-type: none">1. RES gera tela de permissões.2. Administrador visualiza permissões.3. Administrador pode alterar, excluir ou criar uma nova permissão.4. Após a realização de uma destas ações, o caso de uso é encerrado.

Continua na próxima página

Tabela B.1 – continuação da página anterior

Fluxos alternativos:	<ol style="list-style-type: none"> 1. As permissões já se encontram atualizadas (2): <ol style="list-style-type: none"> a. Caso o administrador não encontre permissões para atualizar, o caso de uso é encerrado. 2. Administrador seleciona uma permissão para alterar (3): <ol style="list-style-type: none"> a. RES gera tela de alteração. b. Administrador preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza a permissão. 3. Administrador seleciona uma permissão para excluir (3): <ol style="list-style-type: none"> a. RES gera tela de exclusão. b. Administrador confirma a exclusão. c. RES exclui a permissão e atualiza os relacionamentos com as categorias de permissões e administradores. 4. Administrador deseja cadastrar uma nova permissão (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Administrador informa os dados necessários. c. RES valida as informações. d. RES cadastra a nova permissão.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação - fluxos alternativos (2.c, 4.c): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao administrador.
Caso de uso: [UC006] Manter categorias de permissões	
Atores:	Administrador e usuário.
Descrição:	O administrador pode criar, alterar e excluir categorias de permissões. Os usuários podem estar associados a uma ou mais categorias de permissões.
Pré-condições:	O administrador deve estar autenticado no RES.
Pós-condições:	As categorias de permissões estão atualizadas.

Continua na próxima página

Tabela B.1 – continuação da página anterior

Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de categorias de permissões. 2. Administrador visualiza categorias de permissões. 3. Administrador pode alterar, excluir ou criar uma nova categoria de permissão. 4. Após a realização de uma destas ações, o caso de uso é encerrado.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. As categorias de permissões já se encontram atualizadas (2): <ol style="list-style-type: none"> a. Caso o administrador não encontre categorias de permissões para atualizar, o caso de uso é encerrado. 2. Administrador seleciona uma categoria de permissão para alterar (3): <ol style="list-style-type: none"> a. RES gera tela de alteração. b. Administrador preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza a categoria de permissão. 3. Administrador seleciona uma categoria de permissão para excluir (3): <ol style="list-style-type: none"> a. RES gera tela de exclusão. b. Administrador confirma a exclusão. c. RES exclui a categoria de permissão e atualiza os relacionamentos vinculados a esta. 4. Administrador deseja cadastrar uma nova categoria de permissão (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Administrador informa os dados necessários. c. RES valida as informações. d. RES cadastra a nova categoria de permissão.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação - fluxos alternativos (2.c, 4.c): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao administrador.
Caso de uso: [UC007] Manter grupos	
Atores:	Administrador.

Continua na próxima página

Tabela B.1 – continuação da página anterior

Descrição:	O administrador pode criar, alterar e excluir grupos. Os usuários podem estar associadas a um ou mais grupos.
Pré-condições:	O administrador deve estar autenticado no RES.
Pós-condições:	Os grupos estão atualizados.
Fluxo principal:	<ol style="list-style-type: none">1. RES gera tela de grupos.2. Administrador visualiza grupos.3. Administrador pode alterar, excluir ou criar um novo grupo.4. Após a realização de uma destas ações, o caso de uso é encerrado.
Fluxos alternativos:	<ol style="list-style-type: none">1. Os dados já se encontram atualizados (2):<ol style="list-style-type: none">a. Caso o administrador não encontre dados para atualizar, o caso de uso é encerrado.2. Administrador seleciona um grupo para alterar (3):<ol style="list-style-type: none">a. RES gera tela de alteração.b. Administrador preenche os campos que deseja alterar.c. RES valida os dados fornecidos.d. RES atualiza o grupo.3. Administrador seleciona um grupo para excluir (3):<ol style="list-style-type: none">a. RES gera tela de exclusão.b. Administrador confirma a exclusão.c. RES exclui o grupo.4. Administrador deseja cadastrar um novo grupo (3):<ol style="list-style-type: none">a. RES gera tela de cadastro.b. Administrador informa os dados necessários.c. RES valida as informações.d. RES cadastra o novo grupo.

Continua na próxima página

Tabela B.1 – continuação da página anterior

Fluxos de exceção:	<ol style="list-style-type: none">1. Erro na validação - fluxos alternativos (2.c, 4.c):<ol style="list-style-type: none">a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema.b. A mensagem é exibida ao administrador.
Caso de uso: [UC008] Manter usuários	
Atores:	Administrador e usuário.
Descrição:	O administrador pode criar, alterar e desabilitar usuários.
Pré-condições:	O administrador deve estar autenticado no RES.
Pós-condições:	Os usuários estão atualizados.
Fluxo principal:	<ol style="list-style-type: none">1. RES gera tela de usuários.2. Administrador visualiza usuários.3. Administrador pode alterar, excluir ou criar um novo usuário.4. Após a realização de uma destas ações, o caso de uso é encerrado.

Continua na próxima página

Tabela B.1 – continuação da página anterior

Fluxos alternativos:	<ol style="list-style-type: none"> 1. Os dados já se encontram atualizados (2): <ol style="list-style-type: none"> a. Caso o administrador não encontre dados para atualizar, o caso de uso é encerrado. 2. Administrador seleciona um usuário para alterar (3): <ol style="list-style-type: none"> a. RES gera tela de alteração. b. Administrador preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza o usuário. 3. Administrador seleciona um usuário para desabilitar (3): <ol style="list-style-type: none"> a. RES gera tela do usuário. b. Administrador confirma a desabilitação. c. RES desabilita o usuário. 4. Administrador deseja cadastrar um novo usuário (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Administrador informa os dados necessários. c. RES valida as informações. d. RES cadastra o novo usuário.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação - fluxos alternativos (2.c, 4.c): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao administrador.

B.2 Incremento 02 - prontuário do paciente

B.2.1 Identificação e especificação dos casos de uso

Tabela B.2: especificação dos casos de uso do incremento 02.

Caso de uso: [UC009] Manter médicos	
Atores:	Administrador e médico.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Descrição:	O administrador pode criar, alterar e excluir médicos.
Pré-condições:	O administrador deve estar autenticado no RES.
Pós-condições:	Os médicos estão atualizados.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de médicos. 2. Administrador visualiza médicos. 3. Administrador pode alterar, excluir ou criar um novo médico. 4. Após a realização de uma destas ações, o caso de uso é encerrado.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. Os dados já se encontram atualizados (2): <ol style="list-style-type: none"> a. Caso o administrador não encontre dados para atualizar, o caso de uso é encerrado. 2. Administrador seleciona um médico para alterar (3): <ol style="list-style-type: none"> a. RES gera tela de alteração. b. Administrador preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza o médico. 3. Administrador seleciona um médico para excluir (3): <ol style="list-style-type: none"> a. RES gera tela do médico. b. Administrador confirma a exclusão. c. RES exclui o médico. 4. Administrador deseja cadastrar um novo médico (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Administrador informa os dados necessários. c. RES valida as informações. d. RES cadastra o novo médico.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação - fluxos alternativos (2.c, 4.c): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao administrador.
Caso de uso: [UC010] Manter especialidades médicas	

Continua na próxima página

Tabela B.2 – continuação da página anterior

Atores:	Administrador e médico.
Descrição:	O administrador pode criar, alterar e excluir especialidades médicas.
Pré-condições:	O administrador deve estar autenticado no RES.
Pós-condições:	As especialidades médicas estão atualizadas.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de especialidades médicas. 2. Administrador visualiza especialidades médicas. 3. Administrador pode alterar, excluir ou criar uma nova especialidade médica. 4. Após a realização de uma destas ações, o caso de uso é encerrado.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. As especialidades médicas já se encontram atualizadas (2): <ol style="list-style-type: none"> a. Caso o administrador não encontre especialidades médicas para atualizar, o caso de uso é encerrado. 2. Administrador seleciona uma especialidade médica para alterar (3): <ol style="list-style-type: none"> a. RES gera tela de alteração. b. Administrador preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza a especialidade médica. 3. Administrador seleciona uma especialidade médica para excluir (3): <ol style="list-style-type: none"> a. RES gera tela de exclusão. b. Administrador confirma a exclusão. c. RES exclui a especialidade médica e atualiza os relacionamentos associados. 4. Administrador deseja cadastrar uma nova especialidade médica (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Administrador informa os dados necessários. c. RES valida as informações. d. RES cadastra a nova especialidade médica.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação - fluxos alternativos (2.c, 4.c): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao administrador.
Caso de uso: [UC011] Manter queixas gerais	
Atores:	Administrador.
Descrição:	O administrador pode criar, alterar e excluir queixas gerais.
Pré-condições:	O administrador deve estar autenticado no RES.
Pós-condições:	As queixas gerais estão atualizados.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de queixas gerais. 2. Administrador visualiza queixas gerais. 3. Administrador pode alterar, excluir ou criar uma nova queixa geral. 4. Após a realização de uma destas ações, o caso de uso é encerrado.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Fluxos alternativos:	<ol style="list-style-type: none"> 1. Os dados já se encontram atualizados (2): <ol style="list-style-type: none"> a. Caso o administrador não encontre dados para atualizar, o caso de uso é encerrado. 2. Administrador seleciona uma queixa geral para alterar (3): <ol style="list-style-type: none"> a. RES gera tela de alteração. b. Administrador preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza a queixa geral. 3. Administrador seleciona uma queixa geral para excluir (3): <ol style="list-style-type: none"> a. RES gera tela da queixa geral. b. Administrador confirma a exclusão. c. RES exclui a queixa geral. 4. Administrador deseja cadastrar uma nova queixa geral (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Administrador informa os dados necessários. c. RES valida as informações. d. RES cadastra a nova queixa geral.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação - fluxos alternativos (2.c, 4.c): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao administrador.
Caso de uso: [UC012] Manter vacinas	
Atores:	Administrador.
Descrição:	O administrador pode criar, alterar e excluir vacinas.
Pré-condições:	O administrador deve estar autenticado no RES.
Pós-condições:	As vacinas estão atualizadas.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de vacinas. 2. Administrador visualiza vacinas. 3. Administrador pode alterar, excluir ou criar uma nova vacina. 4. Após a realização de uma destas ações, o caso de uso é encerrado.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. As vacinas já se encontram atualizadas (2): <ol style="list-style-type: none"> a. Caso o administrador não encontre vacinas para atualizar, o caso de uso é encerrado. 2. Administrador seleciona uma vacina para alterar (3): <ol style="list-style-type: none"> a. RES gera tela de alteração. b. Administrador preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza a vacina. 3. Administrador seleciona uma vacina para excluir (3): <ol style="list-style-type: none"> a. RES gera tela de exclusão. b. Administrador confirma a exclusão. c. RES exclui a vacina e atualiza os relacionamentos associados. 4. Administrador deseja cadastrar uma nova vacina (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Administrador informa os dados necessários. c. RES valida as informações. d. RES cadastra a nova vacina.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação - fluxos alternativos (2.c, 4.c): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao administrador.
Caso de uso: [UC013] Manter tipos de eventos	
Atores:	Administrador.
Descrição:	O administrador pode criar, alterar e excluir tipos de eventos.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Pré-condições:	O administrador deve estar autenticado no RES.
Pós-condições:	Os tipos de eventos estão atualizados.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de tipos de eventos. 2. Administrador visualiza tipos de eventos. 3. Administrador pode alterar, excluir ou criar um novo tipo de evento. 4. Após a realização de uma destas ações, o caso de uso é encerrado.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. Os dados já se encontram atualizados (2): <ol style="list-style-type: none"> a. Caso o administrador não encontre dados para atualizar, o caso de uso é encerrado. 2. Administrador seleciona um tipo de evento para alterar (3): <ol style="list-style-type: none"> a. RES gera tela de alteração. b. Administrador preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza o tipo de evento. 3. Administrador seleciona um tipo de evento para excluir (3): <ol style="list-style-type: none"> a. RES gera tela do tipo de evento. b. Administrador confirma a exclusão. c. RES exclui o tipo de evento. 4. Administrador deseja cadastrar um novo tipo de evento (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Administrador informa os dados necessários. c. RES valida as informações. d. RES cadastra o novo tipo de evento.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação - fluxos alternativos (2.c, 4.c): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao administrador.
Caso de uso: [UC014] Manter categorias de eventos	

Continua na próxima página

Tabela B.2 – continuação da página anterior

Atores:	Administrador.
Descrição:	O administrador pode criar, alterar e excluir categorias de eventos.
Pré-condições:	O administrador deve estar autenticado no RES.
Pós-condições:	As categorias de eventos estão atualizadas.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de categorias de eventos. 2. Administrador visualiza categorias de eventos. 3. Administrador pode alterar, excluir ou criar uma nova categoria de evento. 4. Após a realização de uma destas ações, o caso de uso é encerrado.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. As categorias de eventos já se encontram atualizadas (2): <ol style="list-style-type: none"> a. Caso o administrador não encontre categorias de eventos para atualizar, o caso de uso é encerrado. 2. Administrador seleciona uma categoria de evento para alterar (3): <ol style="list-style-type: none"> a. RES gera tela de alteração. b. Administrador preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza a categoria de evento. 3. Administrador seleciona uma categoria de evento para excluir (3): <ol style="list-style-type: none"> a. RES gera tela de exclusão. b. Administrador confirma a exclusão. c. RES exclui a categoria de evento e atualiza os relacionamentos associados. 4. Administrador deseja cadastrar uma nova categoria de evento (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Administrador informa os dados necessários. c. RES valida as informações. d. RES cadastra a nova categoria de evento.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Fluxos de exceção:	<ol style="list-style-type: none">1. Erro na validação - fluxos alternativos (2.c, 4.c):<ol style="list-style-type: none">a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema.b. A mensagem é exibida ao administrador.
Caso de uso: [UC015] Manter eventos	
Atores:	Administrador.
Descrição:	O administrador pode criar, alterar e excluir eventos.
Pré-condições:	O administrador deve estar autenticado no RES.
Pós-condições:	Os eventos estão atualizados.
Fluxo principal:	<ol style="list-style-type: none">1. RES gera tela de eventos.2. Administrador visualiza eventos.3. Administrador pode alterar, excluir ou criar um novo evento.4. Após a realização de uma destas ações, o caso de uso é encerrado.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Fluxos alternativos:	<ol style="list-style-type: none"> 1. Os dados já se encontram atualizados (2): <ol style="list-style-type: none"> a. Caso o administrador não encontre dados para atualizar, o caso de uso é encerrado. 2. Administrador seleciona um evento para alterar (3): <ol style="list-style-type: none"> a. RES gera tela de alteração. b. Administrador preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza o evento. 3. Administrador seleciona um evento para excluir (3): <ol style="list-style-type: none"> a. RES gera tela do evento. b. Administrador confirma a exclusão. c. RES exclui o evento. 4. Administrador deseja cadastrar um novo evento (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Administrador informa os dados necessários. c. RES valida as informações. d. RES cadastra o novo evento.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação - fluxos alternativos (2.c, 4.c): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao administrador.
Caso de uso: [UC016] Manter configurações	
Atores:	Administrador.
Descrição:	O administrador pode criar, alterar e excluir configurações.
Pré-condições:	O administrador deve estar autenticado no RES.
Pós-condições:	As configurações estão atualizadas.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de configurações. 2. Administrador visualiza configurações. 3. Administrador pode alterar, excluir ou criar uma nova configuração. 4. Após a realização de uma destas ações, o caso de uso é encerrado.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. As configurações já se encontram atualizadas (2): <ol style="list-style-type: none"> a. Caso o administrador não encontre configurações para atualizar, o caso de uso é encerrado. 2. Administrador seleciona uma configuração para alterar (3): <ol style="list-style-type: none"> a. RES gera tela de alteração. b. Administrador preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza a configuração. 3. Administrador seleciona uma configuração para excluir (3): <ol style="list-style-type: none"> a. RES gera tela de exclusão. b. Administrador confirma a exclusão. c. RES exclui a configuração e atualiza os relacionamentos vinculados a esta. 4. Administrador deseja cadastrar uma nova configuração (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Administrador informa os dados necessários. c. RES valida as informações. d. RES cadastra a nova configuração.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação - fluxos alternativos (2.c, 4.c): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao administrador.
Caso de uso: [UC017] Manter usuário	
Atores:	Recepcionista e usuário.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Descrição:	O recepcionista é responsável por buscar, atualizar e criar novos usuários.
Pré-condições:	O recepcionista deve estar autenticado no RES.
Pós-condições:	O usuário está atualizado ou um novo usuário foi criado.
Fluxo principal:	<ol style="list-style-type: none"> 1. Recepcionista busca usuário no RES. 2. Recepcionista pode alterar ou criar um novo usuário. 3. Recepcionista abre uma nova consulta para o usuário (UC018).
Fluxos alternativos:	<ol style="list-style-type: none"> 1. As informações do usuário já se encontram atualizadas (2): <ol style="list-style-type: none"> a. Se o recepcionista não encontrar informações do usuário para atualizar, o caso de uso segue para o passo 3. 2. Usuário encontrado (2): <ol style="list-style-type: none"> a. RES gera tela do usuário. b. Recepcionista preenche os campos que deseja alterar. c. RES valida os dados fornecidos. d. RES atualiza o usuário. 3. Usuário não encontrado (3): <ol style="list-style-type: none"> a. RES gera tela de cadastro. b. Recepcionista preenche os campos. c. RES valida os dados fornecidos. d. RES cadastra o usuário.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação - fluxos alternativos (1.c, 2.c): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao administrador.
Caso de uso: [UC018] Abrir consulta	
Atores:	Recepcionista e usuário.
Descrição:	De acordo com as queixas gerais apresentados pelo usuário, o recepcionista escolhe o especialista e abre a consulta.
Pré-condições:	O recepcionista deve estar autenticado e o usuário estar cadastrado no RES.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Pós-condições:	Uma nova consulta foi aberta.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela de abertura de consulta. 2. Recepcionista preenche os campos necessários. 3. RES valida as informações fornecidas. 4. RES abre uma nova consulta.
Fluxos alternativos:	Nenhum fluxo alternativo definido.
Fluxos de exceção:	<ol style="list-style-type: none"> 1. Erro na validação (3): <ol style="list-style-type: none"> a. Se a validação ou verificação falhar, o RES cria uma mensagem de erro descrevendo o problema. b. A mensagem é exibida ao recepcionista.
Caso de uso: [UC019] Manter fila de espera	
Atores:	Recepcionista e usuário.
Descrição:	O recepcionista organiza a ordem de chamada para consulta de acordo o especialista escolhido e a ordem de chegada.
Pré-condições:	O recepcionista deve estar autenticado no RES.
Pós-condições:	O usuário é encaminhado para o consultório médico.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela da fila de espera. 2. Recepcionista organiza a ordem de chamada a partir do especialista escolhido e a ordem de chegada. 3. Recepcionista seleciona o próximo usuário a ser atendido. 4. Recepcionista atualiza o estado da consulta e faz o encaminhamento do usuário para o consultório médico. 5. RES remove a consulta da fila de espera.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. Usuário desistiu da consulta (3): <ol style="list-style-type: none"> a. Recepcionista cancela a consulta e o caso de uso é encerrado.
Fluxos de exceção:	Nenhum fluxo de exceção definido.
Caso de uso: [UC020] Manter consulta	
Atores:	Médico e paciente.
Descrição:	O médico atualiza os estados das consultas dos seus pacientes.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Pré-condições:	O médico deve estar autenticado no RES.
Pós-condições:	As consultas estão atualizadas.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela da consulta. 2. Médico seleciona o estado da consulta à medida que as etapas vão sendo executadas (aberta/processamento/finalizada). 3. RES atualiza o estado da consulta.
Fluxos alternativos:	Nenhum fluxo alternativo definido.
Fluxos de exceção:	Nenhum fluxo de exceção definido.
Caso de uso: [UC021] Solicitar imunização	
Atores:	Médico e paciente.
Descrição:	O médico registra uma solicitação de imunização na consulta.
Pré-condições:	O médico deve estar autenticado no RES.
Pós-condições:	Uma nova solicitação de imunização é cadastrada.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela da consulta. 2. Médico efetua solicitação de imunização. 3. RES gera tela de cadastro de imunização. 4. Médico seleciona a vacina e a data provável de imunização. 5. RES salva a nova solicitação de imunização.
Fluxos alternativos:	Nenhum fluxo alternativo definido.
Fluxos de exceção:	Nenhum fluxo de exceção definido.
Caso de uso: [UC022] Efetuar solicitação de evento	
Atores:	Médico e paciente.
Descrição:	O médico registra uma solicitação de evento na consulta. Eventos podem ser procedimentos, exames, dentre outros.
Pré-condições:	O médico deve estar autenticado no RES.
Pós-condições:	Uma nova solicitação de evento é cadastrada.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela da consulta. 2. Médico efetua solicitação de evento. 3. RES gera tela de cadastro de evento. 4. Médico seleciona o evento desejado. 5. RES salva a nova solicitação de evento.
Fluxos alternativos:	Nenhum fluxo alternativo definido.
Fluxos de exceção:	Nenhum fluxo de exceção definido.
Caso de uso: [UC023] Visualizar resultado de solicitação de evento	
Atores:	Médico e paciente.
Descrição:	O médico visualiza o resultado de uma solicitação de evento realizada por ele ou por outro médico.
Pré-condições:	O médico deve estar autenticado no RES.
Pós-condições:	As informações de interesse do médico podem ser obtidas.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera tela da consulta. 2. Médico seleciona a lista de solicitações de evento da consulta. 3. Médico seleciona a solicitação de evento desejada. 4. RES gera tela da solicitação de evento. 5. Médico visualiza o resultado. 6. Médico pode editar imagens que tenham sido anexadas. 7. Médico finaliza a visualização.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. Médico decide editar imagens anexadas à solicitação de evento (3): <ol style="list-style-type: none"> a. RES carrega o editor de imagens (UC024) e o caso de uso é encerrado.
Fluxos de exceção:	Nenhum fluxo de exceção definido.
Caso de uso: [UC024] Efetuar marcações em resultados de eventos	
Atores:	Médico e paciente.
Descrição:	O médico seleciona uma imagem e pode incluir retângulos, elipses, retas e texto em diversas cores e tamanhos com o objetivo de especificar a linha de raciocínio que originou o diagnóstico.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Pré-condições:	O médico deve estar autenticado no RES.
Pós-condições:	Uma nova imagem com as marcações é anexada ao resultado do evento, preservando a original.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES carrega o editor de imagens. 2. Médico pode selecionar os elementos retângulo, elipse, reta e texto. 3. Médico pode alterar a cor. 4. Médico pode alterar a espessura da linha e o tamanho da fonte. 5. Médico pode desfazer e refazer alterações, como também limpar todas as alterações. 6. Médico desenha os elementos conforme desejado. 7. RES salva uma nova imagem, anexando-a ao evento.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. Médico desiste de alterar a imagem (1): <ol style="list-style-type: none"> a. Médico seleciona a opção cancelar e o caso de uso é encerrado.
Fluxos de exceção:	Nenhum fluxo de exceção definido.
Caso de uso: [UC025] Efetuar diagnóstico	
Atores:	Médico e paciente.
Descrição:	O médico registra o diagnóstico na consulta, de acordo com a CID.
Pré-condições:	O médico deve estar autenticado no RES e a consulta não pode estar finalizada.
Pós-condições:	O diagnóstico foi registrado na consulta.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera a tela da consulta. 2. Médico seleciona a lista de diagnósticos. 3. Médico escreve o diagnóstico (ou parte dele). 4. RES pesquisa automaticamente o diagnóstico na lista de doenças e exibe uma lista para o médico selecionar. 5. Médico seleciona o diagnóstico e solicita a inclusão. 6. RES salva o diagnóstico na consulta.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Fluxos alternativos:	<ol style="list-style-type: none"> 1. Médico inclui o diagnóstico errado (1): <ol style="list-style-type: none"> a. Médico seleciona o diagnóstico e solicita a sua exclusão. b. Se a consulta não estiver finalizada, o RES exclui o diagnóstico, senão não exclui. c. RES salva as alterações.
Fluxos de exceção:	Nenhum fluxo de exceção definido.
Caso de uso: [UC026] Emitir prescrição médica	
Atores:	Médico e paciente.
Descrição:	O médico registra a prescrição médica na consulta, gerando um arquivo para impressão em modelo padronizado e com código de barras que permite a identificação do paciente.
Pré-condições:	O médico deve estar autenticado no RES e a consulta não pode estar finalizada.
Pós-condições:	A prescrição médica é cadastrada.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera a tela da consulta. 2. Médico seleciona a lista de prescrições médicas. 3. Médico solicita a criação de uma nova prescrição. 4. RES gera a tela da prescrição. 5. Médico preenche as informações solicitadas e solicita a inclusão. 6. RES salva a nova prescrição e gera um arquivo PDF padronizado.
Fluxos alternativos:	Nenhum fluxo alternativo definido.
Fluxos de exceção:	Nenhum fluxo de exceção definido.
Caso de uso: [UC027] Executar solicitação de evento	
Atores:	Profissional de saúde e paciente.
Descrição:	O profissional de saúde verifica as solicitações de evento agendadas e faz as suas execuções, registrando os resultados e alterando os estados de cada uma em seguida.
Pré-condições:	O profissional de saúde deve estar autenticado no RES.
Pós-condições:	O estado da solicitação de evento é alterado e os resultados incluídos.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera a tela de solicitações de evento. 2. Profissional de saúde seleciona a solicitação a executar. 3. Profissional de saúde executa a solicitação e coleta os resultados. 4. Profissional de saúde registra o resultado na solicitação e faz a finalização. 5. RES salva o resultado e altera o estado da solicitação de evento.
Fluxos alternativos:	Nenhum fluxo alternativo definido.
Fluxos de exceção:	Nenhum fluxo de exceção definido.
Caso de uso: [UC028] Aplicar imunização	
Atores:	Profissional de saúde e paciente.
Descrição:	O profissional de saúde verifica as solicitações de imunização agendadas e faz as suas execuções, atualizando a data efetiva de cada solicitação em seguida.
Pré-condições:	O profissional de saúde deve estar autenticado no RES.
Pós-condições:	A data efetiva da solicitação é inserida.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera a tela de solicitações de imunização. 2. Profissional de saúde seleciona a solicitação a executar. 3. Profissional de saúde executa a solicitação. 4. Profissional de saúde registra a data efetiva da aplicação. 5. RES salva as informações.
Fluxos alternativos:	Nenhum fluxo alternativo definido.
Fluxos de exceção:	Nenhum fluxo de exceção definido.
Caso de uso: [UC029] Visualizar dados clínicos	
Atores:	Paciente.
Descrição:	O paciente pode visualizar todos os dados referentes à consultas, eventos, imunizações, diagnósticos e prescrições médicas relacionadas ao seu cadastro.
Pré-condições:	O paciente deve estar autenticado no RES.
Pós-condições:	O paciente mantém-se informado sobre seus dados clínicos.

Continua na próxima página

Tabela B.2 – continuação da página anterior

Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera a tela do paciente. 2. Paciente pode visualizar todas as suas informações clínicas e imprimir o que for desejado.
Fluxos alternativos:	Nenhum fluxo alternativo definido.
Fluxos de exceção:	Nenhum fluxo de exceção definido.
Caso de uso: [UC030] Manter mensagens	
Atores:	Usuário.
Descrição:	Através da central de mensagens, os usuários podem trocar mensagens com qualquer outro usuário cadastrado no RES, incluindo os médicos.
Pré-condições:	O usuário deve estar autenticado no RES.
Pós-condições:	O usuário envia ou lê mensagens.
Fluxo principal:	<ol style="list-style-type: none"> 1. RES gera a tela mensagens do paciente. 2. Usuário pode visualizar todas as mensagens recebidas/enviadas ou criar uma nova. 3. Após a realização de uma destas ações, o caso de uso é encerrado.
Fluxos alternativos:	<ol style="list-style-type: none"> 1. Usuário deseja visualizar mensagem recebida/enviada (2): <ol style="list-style-type: none"> a. Usuário seleciona a mensagem desejada. b. RES gera a tela da mensagem. c. Usuário visualiza seu conteúdo (alterando seu estado automaticamente) e pode solicitar o envio de resposta, indo para o fluxo alternativo 2. 2. Usuário deseja criar/responder mensagem (2): <ol style="list-style-type: none"> a. RES gera a tela da mensagem. b. Usuário preenche as informações e solicita o envio. c. RES envia a mensagem.
Fluxos de exceção:	Nenhum fluxo de exceção definido.

Referências Bibliográficas

AMARAL, M. B. do. Padrões de registro e transmissão de dados em saúde. In: *O Prontuário do Paciente na Assistência, Informação e Conhecimento Médico*. São Paulo, Brasil: Faculdade de Medicina da Universidade de São Paulo, 2003.

ANDERSON, J. D. *Increasing the Acceptance of Clinical Information Systems*. United States of America: MD Computing, 1999.

ANNOTATIONS. 2009. Disponível em: <<https://annotations.hibernate.org>>. Acesso em: 24 de junho de 2009.

ASTAH. 2010. Disponível em: <<http://astah.change-vision.com/en/product%-/astah-community.html>>. Acesso em: 22 de março de 2010.

BAUER, F. L. *Software Engineering - Information Processing*. Amsterdam, Netherlands: North Holland Publishing, 1972.

BEMMEL, J. V.; MUSEN, M. A. (Ed.). *Handbook of Medical Informatics*. Bohn Stafleu Van Loghum, Houten, the Netherlands: Springer, 1997.

BOEHM, B. W. A spiral model of software development and enhancement. *IEEE computer*, n. 21(5), p. 61–72, 1988.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. Second. Rio de Janeiro, Brasil: Elsevier, 2005.

CANCIAN, P. S. *Proposta de um Sistema de Prontuário Eletrônico do Paciente*. Dissertação (Dissertação de Mestrado) — Universidade Tecnológica Federal do Paraná, Curitiba, Brasil, 2006.

CENTRALXCLINIC. 2010. Disponível em: <<http://www.centralxclinic.com.br>>. Acesso em: 20 de janeiro de 2010.

COSTA, C.; MARQUES, A. Implementação de um prontuário eletrônico do paciente na maternidade escola januário cicco: Um primeiro passo. In: *PEP'99*. Rio de Janeiro, Brasil: Sociedade Brasileira de Informática em Saúde, 1999.

COSTA, C. G. A. da. *Desenvolvimento e Avaliação Tecnológica de um Sistema de Prontuário Eletrônico do Paciente, Baseado nos Paradigmas da World Wide Web e da Engenharia de Software*. Dissertação (Dissertação de Mestrado) — Universidade Estadual de Campinas, São Paulo, Brasil, 2001.

CUNHA, F. J. A. P. *A Gestão da Informação nos Hospitais - Importância do Prontuário Eletrônico na Integração de Sistemas de Informação em Saúde*. Dissertação (Dissertação de Mestrado) — Universidade Federal da Bahia, Salvador, Brasil, 2005.

DEBIAN. 2009. Disponível em: <<http://www.debian.org>>. Acesso em: 26 de março de 2009.

DICK, R. S.; STEEN, E. B.; DETMER, D. E. (Ed.). *The Computer Based Patient Record - An essential Technology for Health Care*. Washington, United States of America: National Academy Press, 1997. (Committee on Improving the Patient Record - Institute of Medicine).

ECLIPSE. 2009. Disponível em: <<http://www.eclipse.org>>. Acesso em: 28 de junho de 2009.

EJB3. 2009. Disponível em: <<http://java.sun.com/products/ejb>>. Acesso em: 22 de junho de 2009.

FILHO, P. P. R. *MCA Crisp: novo método de segmentação dos pulmões em imagens de TC baseado no Método de Contorno Ativo*. Dissertação (Dissertação de Mestrado) — Universidade Federal do Ceará, Fortaleza, Brasil, 2010.

GOOGLEHEALTH. 2010. Disponível em: <<https://www.google.com/health>>. Acesso em: 29 de março de 2010.

GUEDES, G. T. A. *UML: uma abordagem prática*. Third. São Paulo, Brasil: Novatec Editora, 2008.

HIBERNATE. 2009. Disponível em: <<https://www.hibernate.org>>. Acesso em: 24 de junho de 2009.

HL7. 2010. Disponível em: <<http://www.hl7brazil.org>>. Acesso em: 20 de janeiro de 2010.

HO, L. M.; MCGHEE, S. M.; HEDLEY, A. J.; LEONG, J. C. The application of a computerized problem-oriented medical record system and its impact on patient care. *International Journal of Medical Informatics*, n. 55, p. 47–59, 1999.

IEEE. Ieee recommended practice for software requirements specifications. In: *IEEE software engineering standards collection*. New York, USA: IEEE Computer Society Press, 1998.

JAAS. 2009. Disponível em: <<http://java.sun.com/javase/6/docs/technotes/guides/security/jaas/JAASRefGuide.html>>. Acesso em: 22 de junho de 2009.

JAVA. 2009. Disponível em: <<http://java.sun.com>>. Acesso em: 01 de abril de 2009.

JBOSSAS. 2009. Disponível em: <<http://www.jboss.org/jbossas>>. Acesso em: 05 de julho de 2009.

JBOSSTOOLS. 2009. Disponível em: <<http://www.jboss.org/tools>>. Acesso em: 28 de junho de 2009.

KISSINGER, K.; BORCHARDT, S. (Ed.). *Information Technology for Integrated Health System: positioning for the future*. New York, United States of America: John Wiley, 1995.

KOHN, D. *Electronic Health Records: Changing the Vision*. Philadelphia, United States of America: W. B. Saunders Company, 1999.

LDAP. 2009. Disponível em: <<http://pt.wikipedia.org/wiki/LDAP>>. Acesso em: 25 de junho de 2009.

MASSAD, E.; MARIN, H. de F.; NETO, R. S. de A. Prontuário eletrônico do paciente: definições e conceitos. In: *O Prontuário do Paciente na Assistência, Informação e Conhecimento Médico*. São Paulo, Brasil: Faculdade de Medicina da Universidade de São Paulo, 2003.

MEYERS, K. C.; MILLER, H. J.; NAYEMI-RAD, F. Problem focused knowledge navigation: implementing the problem focused medical record and the o-heap note. In: *Proceedings of the AMIA Fall Symposium*. Orlando, USA: Hanley and Belfus, Inc, 1998. p. 325–9.

MYSQL. 2009. Disponível em: <<http://www.mysql.com>>. Acesso em: 10 de julho de 2009.

NARDON, F. B.; FURUIE, S.; TACHINARDI, U. Novas tecnologias para construção do prontuário eletrônico do paciente. In: *Anais do Congresso da Sociedade Brasileira de Informática*. São Paulo, Brasil: Sociedade Brasileira de Informática em Saúde, 2000.

NETBEANS. 2009. Disponível em: <<http://www.netbeans.org>>. Acesso em: 28 de junho de 2009.

OPENEMR. 2010. Disponível em: <<http://www.oemr.org>>. Acesso em: 20 de janeiro de 2010.

POSTGRESQL. 2009. Disponível em: <<http://www.postgresql.org>>. Acesso em: 10 de julho de 2009.

PRESSMAN, R. S. *Engenharia de Software*. Sixth. São Paulo, Brasil: McGraw-Hill, 2006.

PRODOCTOR. 2010. Disponível em: <<http://www.prodoctor.net>>. Acesso em: 20 de janeiro de 2010.

RICHFACES. 2009. Disponível em: <<http://www.jboss.org/richfaces>>. Acesso em: 26 de junho de 2009.

SABBATINI, R. M. E. *Introdução à microinformática para usuário em saúde*. São Paulo, Brasil: Academia de Ciências de São Paulo, 1982.

SALMON, P.; RAPPAPORT, A.; BAINBRIDGE, M. Taking the problem oriented medical record forward. In: *Proceedings of the AMIA Fall Symposium*. Washington, USA: Hanley and Belfus, Inc, 1996. p. 463–7.

SEAM. 2009. Disponível em: <<http://seamframework.org/>>. Acesso em: 22 de junho de 2009.

SLEE, V. N.; SLEE, D. A.; SCHMIDT, H. J. *The Endangered Medical Record: Ensuring Its Integrity in the Age of Informatics*. Saint Paul, Minnessota, United States of America: Tringa Press, 2000.

SOMMERVILLE, I. *Engenharia de Software*. Eighth. São Paulo, Brasil: Pearson Addison-Wesley, 2007.

UBUNTU. 2009. Disponível em: <<http://www.ubuntu.com>>. Acesso em: 26 de março de 2009.

WAEAGEMANN, C. P. The five levels of electronic health records. *M.D. computing: computers in medical practice (MD Comput)*, v. 13, 1996.

WEED, L. Medical record that guide and teach. *The New England Journal of Medicine*, v. 278, p. 593–600, 1968.