

PRONTUÁRIO ELETRÔNICO DO PACIENTE PARA CARDIOLOGIA E ODONTOLOGIA

Trabalho de Conclusão de Curso

Engenharia da Computação

Samuel Luna Martins

Orientador: Prof. Wellington Pinheiro dos Santos



UNIVERSIDADE
DE PERNAMBUCO

SAMUEL LUNA MARTINS

**PRONTUÁRIO ELETRÔNICO DO
PACIENTE PARA CARDIOLOGIA E
ODONTOLOGIA**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

Recife, novembro de 2009.

Agradecimentos

Agradeço antes de tudo a Deus por ter me dado esperança, por te me ajudado em minha carreira e minha vida e por me mostrar o caminho certo.

Aos meus pais que acreditaram no meu potencial e me deram suporte para a conclusão dos meus estudos.

A Wellington, meu professor orientador e amigo, por ter tido paciência e acreditado no meu trabalho.

A todos os meus amigos e colegas que me ajudaram de alguma forma seja com sugestões, críticas ou orientações.

Resumo

A elaboração de prontuários eletrônicos de pacientes envolve um esforço multidisciplinar que engloba desde a computação até as áreas do conhecimento ligadas à saúde. Prontuários eletrônicos visam ser uma solução para uma gestão eficiente de dados clínicos. Este trabalho apresenta uma ferramenta desenvolvida para a gestão de prontuários para as áreas de odontologia e cardiologia: o PEPNET. Foram realizados três estudos de caso de aplicativos disponíveis no mercado de gestão de prontuários. A partir do estudo das ferramentas foram observadas as principais propriedades dos prontuários eletrônicos. A ferramenta desenvolvida neste trabalho, PEPNET, é um aplicativo distribuído, heterogêneo e seguro, condição esta garantida pelo *framework* empregado: *Spring Security*. Para o desenvolvimento foram usados diversos *frameworks* que seguem a arquitetura de *software* Model-View-Controller. Durante o processo de construção da ferramenta, foi adotada a metodologia de desenvolvimento ágil de *software*. O PEPNET se mostrou uma ferramenta com bastante potencial para auxiliar os profissionais de saúde na gestão racional e ética de informações clínicas, acrescentando facilidades no uso tais como a melhora na organização e a integração dos dados clínicos, da legibilidade das informações e sua segurança quanto ao seu armazenamento e acesso.

Abstract

The elaboration of electronic patient records approaches a multidisciplinary effort involving fields of interest from computing to health sciences and administration. Electronic patient records are designed to be efficient solutions to manage clinical data. This work presents a web-based tool developed for electronic patient records management, applied to odontology and cardiology: PEPNET. We performed three cases of study on commercial applications dealing with management of electronic patient records. From this study we extracted the principal properties of these electronic records. PEPNET was developed to be a distributed, heterogeneous and safe web tool. Safety is a condition guaranteed by the use of the framework Spring Security. The other frameworks follow the software architecture Model-View-Controller. To develop this web tool we used the agile development software methodology. PEPNET proved to be potentially an important tool to aid health professionals to rationally and ethically manage clinical data, adding important resources to organize and integrate data, improving issues like information legibility, and store and access security.

Sumário

Resumo	i
Abstract	ii
Sumário	iii
Índice de Figuras	v
Índice de Tabelas	vi
Tabela de Símbolos e Siglas	vii
Capítulo 1 Introdução	8
1.1 Motivação	8
1.2 Objetivos	9
1.3 Organização	9
Capítulo 2 Estado da Arte dos Prontuários Eletrônicos	10
2.1 Prontuário Eletrônico de Paciente	11
2.1.1 Vantagens e Desvantagens do Prontuário Eletrônico	12
2.2 Código de Ética	14
2.3 Padrão de Imagens Médicas	16
2.4 Pesquisa de Mercado	16
2.4.1 Aplicativo 1: Consultório Digital da SBC	16
2.4.2 Aplicativo 2: PEP InCor	18
2.4.3 Aplicativo 3: Dental Office	20
2.5 conclusão	22
Capítulo 3 PEPNET: O que é, características, motivação	24
3.1 Metodologia de Desenvolvimento	24
3.1.1 Processo de Desenvolvimento	24
3.1.2 Análise de Requisitos	25
3.1.3 Análise de Risco	27

3.1.4	Análise e Projeto	28
3.1.5	Implementação	29
3.2	O sistema PEPNET	32
3.2.1	Interface Gráfica	33
3.2.2	Vantagens e Desvantagens	36
3.2.3	Futuro do PEPNET	37
Capítulo 4 Conclusão e Trabalhos Futuros		38
4.1	Contribuições	38
4.2	Trabalhos Futuros	39
Bibliografia		40
Apêndice A Análise de Casos de Uso		43
Apêndice B Especificação de Casos de Uso		46
Apêndice C Diagramas		50
Apêndice D Testes		54

Índice de Figuras

Figura 1.	Desvantagens do Prontuário em Papel. Pesquisa realizada na Maternidade Escola Januário Cicco, Natal/RN [4].....	11
Figura 2.	Vantagens do Prontuário Eletrônico. Pesquisa realizada na Maternidade Escola Januário Cicco, Natal/RN [4].....	13
Figura 3.	Vantagens do Prontuário Eletrônico. Pesquisa realizada na Maternidade Escola Januário Cicco, Natal/RN [4].....	14
Figura 4.	Interface gráfica de usuário do Consultório digital da Sociedade Brasileira de Cardiologia [17].	17
Figura 5.	Interface Gráfica do Prontuário eletrônico do InCor [21].	19
Figura 6.	Interface de usuário do Prontuário eletrônico do Dental Office [22].	21
Figura 7.	Tela de Validação dos usuários do sistema.	34
Figura 8.	Tela de atualização de perfil de um usuário do sistema.....	34
Figura 9.	Tela inicial e principal do sistema.	35
Figura 10.	Tela de Gerenciamento de Pacientes.	35
Figura 11.	Tela de pesquisa do CID.	36
Figura 12.	Diagrama dos Casos de Uso da Gerência de Usuário Administrador..	50
Figura 13.	Diagrama dos Casos de Uso da Gerência da Agenda para Odontologia	50
Figura 14.	Diagrama dos Casos de Uso da Gerência de Pacientes para Odontologia	51
Figura 15.	Diagrama dos Casos de Uso da Gerência de Pacientes para Médico .	52

Índice de Tabelas

Tabela 1.	Comparativo entre os aplicativos analisados.....	22
Tabela 2.	Período das atividades da análise de requisitos.....	25
Tabela 3.	Análise de Riscos para o Escopo Processos	27
Tabela 4.	Análise de Riscos para o Escopo Geral	28
Tabela 5.	Descrição da Classe User	43
Tabela 6.	Descrição da Classe UserServiceImpl	44
Tabela 7.	Descrição da Classe Cid	44
Tabela 8.	Descrição da Classe BaseCidAction	45
Tabela 9.	Descrição da Classe CidserviceImpl	45

Tabela de Símbolos e Siglas

(Dispostos por ordem de aparição no texto)

PEP – Prontuário Eletrônico do Paciente

CFM – Conselho Federal de Medicina

DICOM – Digital Imaging Communications in Medicine

CORBA – Common Object Request Broker Architecture

XML – Extensible Markup Language

HTML – Hypertext Markup Language

InCor – Instituto de Coração do Hospital das Clínicas

MACA – *Middleware* para Autorização e Controle de Acesso

LDAP – Lightweight Directory Access Protocol

CGI – Common Gateway Interface

PACS – Picture Archiving and Communication System

CID – Classificação Internacional de Doenças e Problemas Relacionado à Saúde

XP – eXtreme Programming ou Programação Extrema

MVC – Model-view-Controller (Arquitetura Modelo Visão Controle)

UML – Unified Modeling Language

EJB – Enterprise JavaBeans

SQL – Structured Query Language *ou* Linguagem de Consulta Estruturada

JDBC – Java Database Connectivity

POJOs – Plain Old Java Objects

JSP – Java Server Pages

SGBD – Sistema Gerenciador de Banco de Dados

AJAX – Asynchronous Javascript And XML

Capítulo 1

Introdução

A gestão das informações do paciente é uma prática muito antiga e essencial no acompanhamento clínico. Com a evolução da medicina, modificou-se a forma de armazenamento desses dados bem como quais informações eram mais relevantes a serem registradas.

A semiologia médica [1] estuda através das anamneses, exames clínicos, observações, tabelas, síndromes, entre outros, os sinais e sintomas das doenças humanas com o propósito de se estabelecer um diagnóstico. A medicina sempre buscou o uso de uma gestão eficiente dos dados da semiologia médica. Com o advento da informática, mais precisamente com o surgimento da área “sistemas da informação”, foram desenvolvidos os prontuários eletrônicos do paciente para armazenar os diagnósticos e realizar os acompanhamentos clínicos.

1.1 Motivação

Prontuário eletrônico do paciente é a forma que tem se demonstrado mais eficiente na manipulação dos dados clínicos. A utilização de sistemas eletrônicos para gestão resolve diversos problemas de infra-estrutura e de recuperação das informações dos pacientes.

Há diversas soluções prontas de gestão de prontuários disponíveis na Internet sob licenças proprietária ou livre e são normalmente voltados para um determinado campo da saúde. O estudo e desenvolvimento de um sistema para gestão de prontuários que ofereça suporte a mais de um campo da saúde (por exemplo, odontologia e cardiologia), e que seja acessível via Internet, pode vir a suprir a carência de uma solução que seja mais heterogênea e ao mesmo tempo distribuída.

Foram escolhidas as áreas cardiologia e odontologia, porque inicialmente se teve a um clínica chamada São Mateus localizada na cidade do Crato, Ceará, que possui um problema que é bastante comum a várias clínicas e hospitais no Brasil que é justamente a gestão descentralizada de prontuários. A Clínica São Mateus

trabalha justamente com as áreas cardiologia e odontologia e foi proposto como solução um sistema único de prontuário que pudesse unir a gestão dos dados dos pacientes da clínica em questão.

1.2 Objetivos

O objetivo deste trabalho é estudar sobre a gestão dos prontuários dos pacientes, levando em consideração o código de ética, qual a forma mais eficiente de se administrar os dados clínicos e quais características são imprescindíveis para uma boa gestão.

Para validar este estudo, foi desenvolvido um sistema de gestão de prontuários eletrônicos do paciente para as áreas de odontologia e cardiologia, denominado PEPNET. Este sistema deve estar de acordo com o código de ética de medicina e odontologia e oferecer as funcionalidades básicas e essenciais para a gestão de dados clínicos dos pacientes.

Os resultados obtidos a partir deste trabalho poderão ser usados como base para diversos outros projetos ou estudos mais aprofundados que abordem a administração informatizada clínica e hospitalar dos pacientes.

1.3 Organização

O segundo capítulo deste documento apresenta a visão geral sobre a gestão de prontuários do paciente; o código de ética dos profissionais de saúde, as diferenças, vantagens e desvantagens entre o uso dos prontuários de papeis [2] e dos prontuários eletrônicos; e a análise sobre adoção de alguns sistemas de gestão dos prontuários eletrônicos.

O capítulo 3 apresenta o sistema de gestão de prontuários eletrônicos PEPNET, destacando suas funcionalidades, arquitetura bem como sua metodologia de desenvolvimento.

No capítulo 4, são apresentados os resultados obtidos através do estudo e desenvolvimento do sistema de gestão de prontuários eletrônicos PEPNET e sugestões para possíveis trabalhos futuros que complementarão este projeto.

Capítulo 2

Estado da Arte dos Prontuários

Eletrônicos

A idéia de administrar a informação na área da saúde é bastante antiga. Por volta de 1500 a.C., os papiros [3] já eram usados como base de armazenamento de informações inclusive sobre doenças, tratamentos e acompanhamento de pacientes. Georg Moritz Ebers (1837-1898), um especialista em arqueologia egípcia, liga o uso do papiro como base de armazenamento de dados clínicos a um dos primeiros médicos (ou curandeiros) da humanidade, o egípcio Imhotep, que usava os papiros em diversas finalidades dentre elas na gestão primitiva de informação da saúde. Um dos mais famosos papiros, conhecido como papiro Cirúrgico de Edwin Smith (nome atribuído para homenagear os descobridores), relata diversos tratamentos cirúrgicos como casos de traumatismos cranio-encefálicos, ferimento no pescoço e garganta, na clavícula, no braço, no esterno e costelas, no ombro e na coluna vertebral.

Hoje, mesmo com a evolução tecnológica, o uso apenas do papel como meio de armazenamento de dados ainda persiste em ambientes hospitalares e clínicas de certas regiões do Brasil. O uso somente do papel para a gestão da informação aumenta a lentidão na consulta dos dados, necessita da alocação de grande infraestrutura para o armazenamento dos papeis e abre espaço para ocorrência de erros na manipulação, consulta e registro da informação. Costa e Marques [4] realizaram uma pesquisa em uma instituição hospitalar entre os funcionários e foi constatado, através da Figura 1, que a perda de informações é o problema mais temido pelos entrevistados.

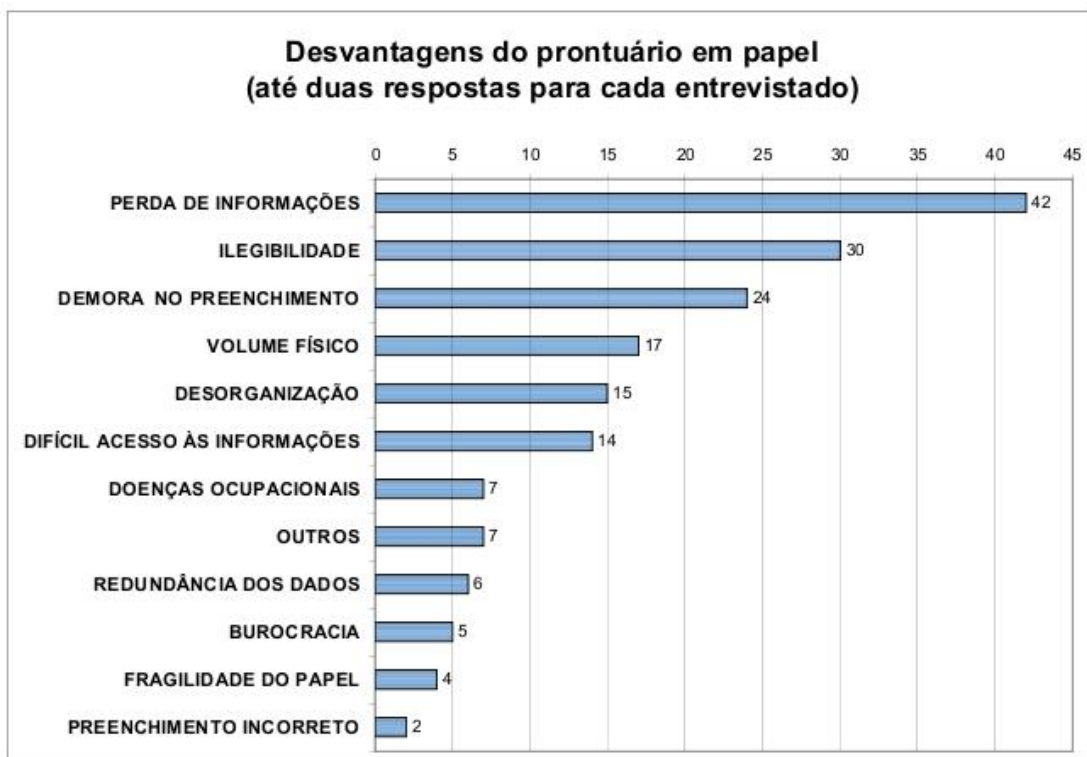


Figura 1. Desvantagens do Prontuário em Papel. Pesquisa realizada na Maternidade Escola Januário Cicco, Natal/RN [4].

Automatizar esse processo de administração de informações de pacientes através do uso da tecnologia faz com que os hospitais possam gerir de uma forma melhor os seus recursos financeiros a fim de aperfeiçoar o atendimento do paciente e saber do que realmente eles necessitam. Além de garantir a segurança contra perdas, extravios e vazamentos de informações privadas.

2.1 Prontuário Eletrônico de Paciente

Prontuário Eletrônico de Paciente (PEP) consiste de um documento único com caráter legal, sigiloso e científico, constituído de informações dos pacientes que contém os dados pessoais, endereço, histórico das consultas, exames, observações clínicas importantes, imagens e todos os dados necessários para o acompanhamento clínico [5]. Principais características inerentes ao PEP:

- É um sistema distribuído - se considerarmos que uma pessoa recebe que ao longo da vida atendimento em diversas instituições de saúde e que cada uma

armazena parte de informações clínicas do indivíduo, então se pode dizer que o PEP é conceitualmente distribuído.

- É um sistema heterogêneo - como cada instituição de saúde possui um sistema próprio de armazenamento das informações clínicas e com suas próprias características (*hardware*, sistema operacional, etc.), então se pode dizer que o PEP é conceitualmente heterogêneo.

Unificar toda essa base de informação não é uma tarefa nada fácil. Ultimamente, têm surgido tecnologias que podem auxiliar a comunicação dos diversos sistemas PEP como XML [6], que nada mais é que um padrão de representação de documentos. Outra tecnologia muito discutida e defendida para a padronização dos sistemas de saúde é o CORBA [7]. O CORBA é usado para criar um *Middleware Médico* que nada mais é que uma camada de *software* que disponibiliza as aplicações distribuídas em um pacote de serviços básicos para sistemas de saúde, tais como identificação e cadastro de pacientes, acesso às imagens médicas, etc.

O principal usuário do PEP é o profissional de saúde. Muitos destes profissionais ainda não estão totalmente familiarizados com computadores ou não possuem total habilidade em digitação. Para garantir que o sistema seja utilizado da forma correta e plenamente adotado nos ambientes clínicos e hospitalares, deve-se ouvir as observações feitas por eles. Essa interação é crucial para a implantação de um prontuário eletrônico funcional e com boa usabilidade.

2.1.1 Vantagens e Desvantagens do Prontuário Eletrônico

São diversas as vantagens do uso do prontuário eletrônico em relação ao prontuário de papel, dentre elas se destaca:

- O acesso remoto e simultâneo;
- Legibilidade;
- Segurança dos dados;
- Confidencialidade dos dados do paciente;
- Flexibilidade do layout dos dados;
- Integração com outros sistemas de informação;

- Construção de diversos tipos de relatórios;
- Atualização dos dados.

Continuando a análise da pesquisa feita por Costa e Marques em 1999 [4], a agilidade no atendimento é sem dúvida a maior vantagem no uso do Prontuário Eletrônico, como é explanado na Figura 2.

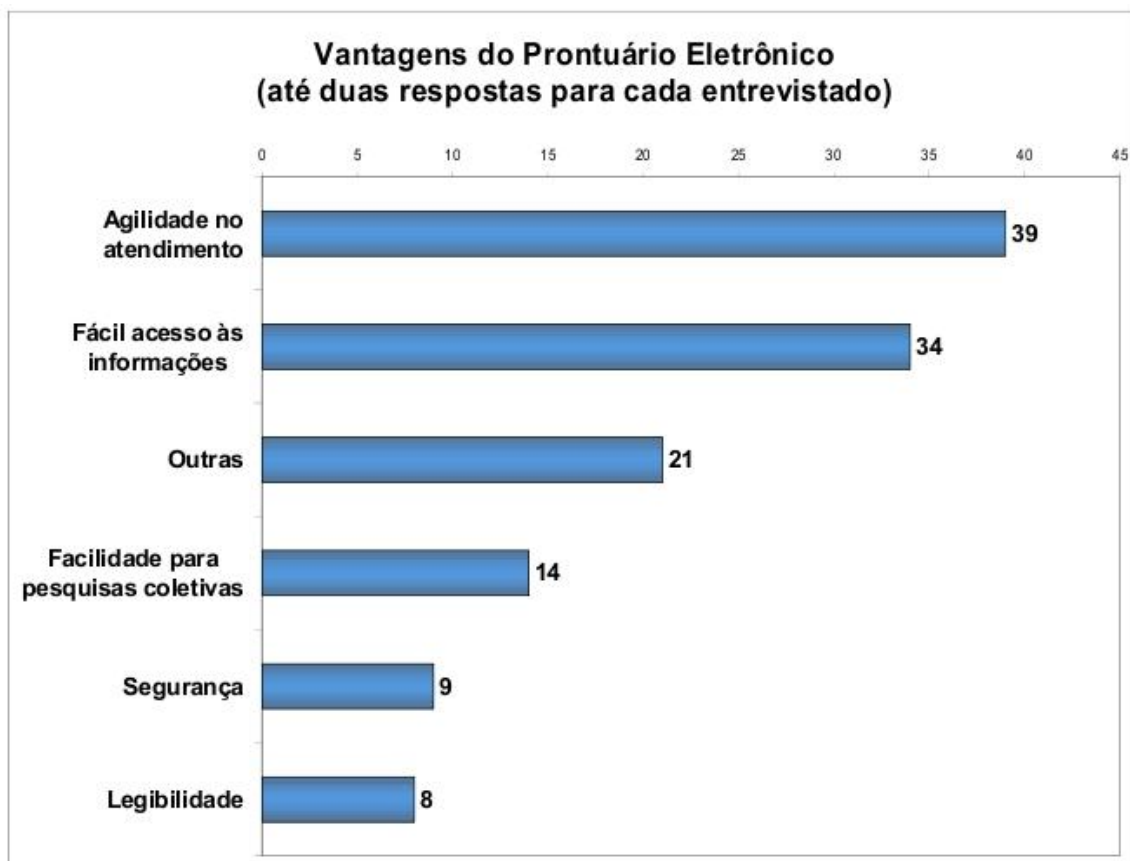


Figura 2. Vantagens do Prontuário Eletrônico. Pesquisa realizada na Maternidade Escola Januário Cicco, Natal/RN [4].

Mas para garantir as vantagens citadas anteriormente, deve-se assegurar que: os dados devem ser armazenados indefinidamente, de forma estruturada, codificados num vocabulário comum que permita o apoio à pesquisa. Adicionalmente, deve existir um número suficiente de terminais de acesso ao sistema, distribuídos em todos os locais de atendimento da instituição com todas as informações sobre os pacientes disponíveis.

Foram levantadas também por Costa e Marques [4] as possíveis desvantagens no uso do Prontuário Eletrônico, como mostra a Figura 3, e o principal temor dos funcionários é a necessidade de capacitação para o uso do sistema. Ainda foi citada a possibilidade de o sistema vir a ficar temporariamente indisponível o que poderia resultar na paralisação do atendimento ou na volta do uso do papel para o armazenamento dos dados dos pacientes.

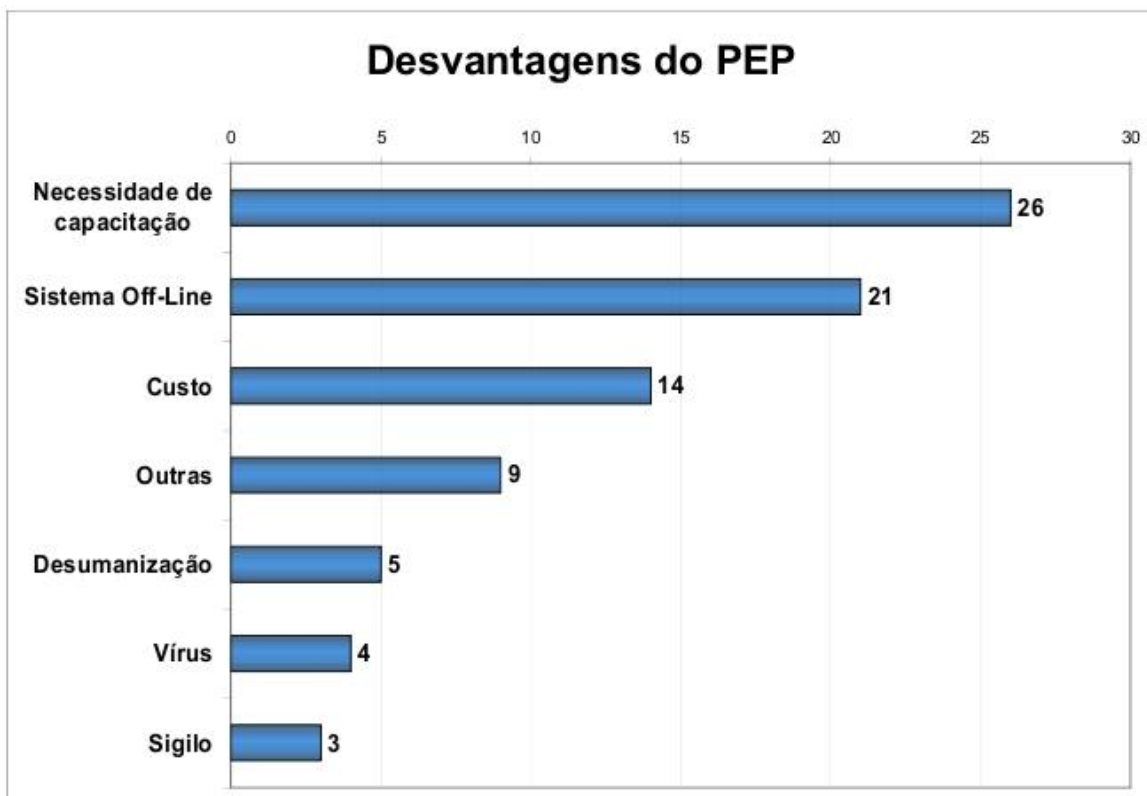


Figura 3. Vantagens do Prontuário Eletrônico. Pesquisa realizada na Maternidade Escola Januário Cicco, Natal/RN [4].

2.2 Código de Ética

Está presente no Código de Ética dos profissionais da saúde o dever da gestão e segurança dos dados clínicos e a disponibilização dessas informações aos pacientes como prova para finalidade jurídica ou pericial.

O Código de Ética Odontológica [8], no capítulo III, dos Deveres fundamentais, cita que:

- *Art. 5 – Constituem deveres fundamentais dos profissionais e entidades de Odontologia:*
 - VIII. elaborar e manter atualizados os prontuários de pacientes, conservando-os em arquivos próprios.*
 - XIII. resguardar sempre a privacidade do paciente.*
 - XVI. garantir ao paciente ou seu responsável legal, acesso a seu prontuário, sempre que for expressamente solicitado podendo conceder cópia do documento, mediante recibo de entrega.*

O código de Ética Médica [9] cita que:

- *É vedado ao Médico: Art. 69 - Deixar de elaborar prontuário médico para cada paciente.*
- *É vedado ao Médico: Art. 70 - Negar ao paciente acesso a seu prontuário médico, ficha clínica ou similar, bem como deixar de dar explicações necessárias à sua compreensão, salvo quando ocasionar riscos para o paciente ou para terceiros.*
- *É vedado ao Médico: Art. 71 - Deixar de fornecer laudo médico ao paciente, quando do encaminhamento ou transferência para fins de continuidade do tratamento, ou na alta, se solicitado.*
- *Por fim, o Art. 11 - O médico deve manter sigilo quanto às informações confidenciais de que tiver conhecimento no desempenho de suas funções. O Mesmo se aplica ao trabalho em empresas, exceto nos casos em que seu silêncio prejudique ou ponha em risco a saúde do trabalhador ou da comunidade.*

Isso reflete diretamente como devem ser geridos os prontuários médicos. A segurança é um dos fatores que mais pesa de forma positiva para a migração dos prontuários de papeis para documentos digitais.

Em 2002, através das Resoluções 1.638 [10] e 1.639 [11] (CONSELHO FEDERAL DE MEDICINA, 2002), o PEP foi reconhecido como uma forma legalizada de armazenamento de informações clínicas de pacientes. O Conselho Federal de Medicina (CFM) reconhece também a assinatura eletrônica como legítima para os documentos de internação, alta e prescrição.

2.3 Padrão de Imagens Médicas

A utilização de Imagens Médicas [12] [13] [14] é essencial para avaliação médica de tecidos e funções orgânicas do corpo humano e normalmente acompanha os prontuários de pacientes como item de embasamento do diagnóstico escrito pelo profissional de saúde. Embora ainda existam muitas imagens médicas feitas a partir dos raios-X (radiografia), com a invenção da Tomografia Computadorizada em 1972, o uso da imagem digitalizada vem se tornando uma constante na área de saúde. Com a adoção do padrão DICOM (*Digital Imaging and Communication*) [3] pela sociedade de Radiologia da América do Norte e por diversos fabricantes de equipamentos médicos, direcionou o mercado mundial a migrar para o uso de aparelhos médicos com suporte à interface DICOM. DICOM, de acordo o dicionário Babylon [15] consiste em: "um conjunto de normas para tratamento e transmissão de informação médica (imagens médicas) num formato eletrônico, estruturando um protocolo. Foi criado, com a finalidade de padronizar a formatação das imagens diagnósticas como tomografia, Ressonâncias Magnéticas, Radiografias, Ultrassonografias, etc.(...)". A principal vantagem na adoção do padrão DICOM é a possibilidade de se trocar imagens médicas e informações entre equipamentos de diagnósticos geradores de imagens, computadores e hospitais.

2.4 Pesquisa de Mercado

Foram analisados três soluções disponíveis na Internet que oferecem o serviço de gerenciamento de prontuários eletrônico de pacientes, sob os aspectos relacionados ao suporte, licença do aplicativo, usabilidade, robustez, confiança, segurança, manutenção, preço e tempo de vida. Dois dos aplicativos eram voltados exclusivamente para área médica e um terceiro, para odontologia.

2.4.1 Aplicativo 1: Consultório Digital da SBC

Consultório digital [16] [17] é um aplicativo criado pela Sociedade Brasileira de Cardiologia que, apesar de não se propor a ser um Prontuário Eletrônico, ele atende a vários requisitos no gerenciamento de informações clínicas. Foi desenvolvido para

ser utilizado em computadores pessoais que utilizam a plataforma Microsoft Windows e oferece suporte para o gerenciamento informações sobre os pacientes, exames, imagens médicas, convênios, medicamentos além de oferecer uma agenda de controle de consultas. Como recurso extra, pode-se exportar dados para dispositivos móveis como Palmtops ou para Internet. Uma característica importante desse sistema é a boa usabilidade como se pode observar através da Figura 4, onde demonstra uma interface de usuário simples e com poucos cliques, consegue chegar a uma área desejada do aplicativo.

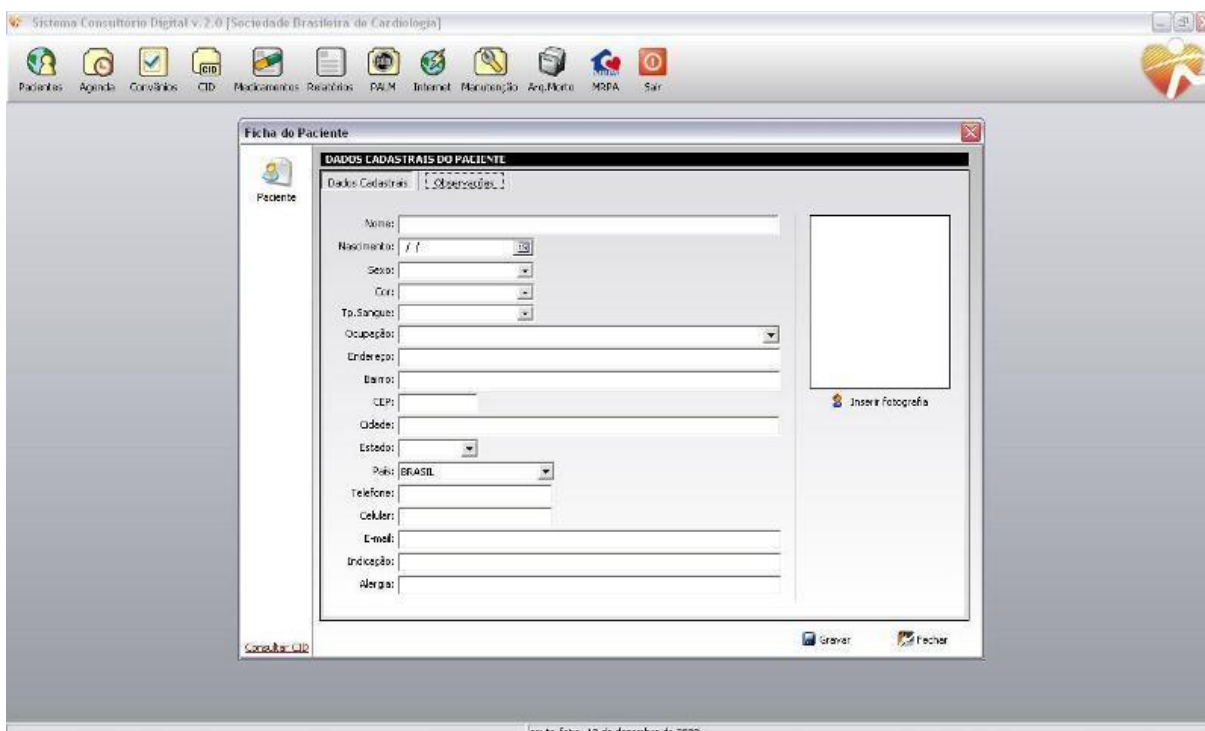


Figura 4. Interface gráfica de usuário do Consultório digital da Sociedade Brasileira de Cardiologia [17].

Vantagens

- Fácil de instalar, não precisa ser um especialista em informática para implantar e configurar o sistema;
- Possui uma boa usabilidade;
- Possui suporte ao usuário via site de Internet, Email e telefone;
- Todo ano é lançada uma nova versão do aplicativo com melhorias;

- É gratuito para os associados à Sociedade Brasileira de Cardiologia;
- Gera gráfico a partir dos dados armazenados;
- Gera laudos personalizados.

Desvantagens

- Não atende a todos os requisitos de um Prontuário eletrônico, como por exemplo abordar a mais de uma área de saúde;
- Criado para funcionar em um computador pessoal, ele não possui uma interface de acesso direta pela internet, impossibilitando o acesso remoto 24 horas ao sistema;
- Não há suporte ao padrão DICOM;
- Fraco gerenciamento de imagens médicas;
- Feito somente para a plataforma proprietária Microsoft Windows;
- Não adaptável a outras necessidades que não seja cardiologia;
- Licença proprietária, disponível gratuitamente apenas aos associados à Sociedade Brasileira de Cardiologia;
- Utiliza um sistema de banco de dados de padrão proprietário e não criptografado;
- Não é um sistema heterogêneo e nem distribuído.

2.4.2 Aplicativo 2: PEP InCor

O Instituto do Coração (InCor) [14], para suprir as necessidades de integração de informações hospitalares, implementou um conjunto de sistemas para gerenciar Imagens Médicas e as informações administrativas e clínica hospitalares. Esse pacote de sistemas foi denominado de Prontuário eletrônico de Pacientes (PEP). O InCor faz parte do Hospital das Clínicas da universidade de São Paulo. Além dele, há mais cinco outras instituições que compõem o Hospital das Clínicas. Cada unidade é capaz de gerar uma quantidade expressiva de informações que muitas vezes são dados que estão duplicados nas outras instituições. Ao integrar essas informações, o

Hospital das Clínicas cortou o desperdício, agilizou o processo de gerência e possibilitou o acesso em rede e compartilhado das informações.

O sistema desenvolvido consiste de um prontuário eletrônico completo e funcional que pode gerenciar exames laboratoriais, imagens DICOM (estáticas, dinâmicas e 3D), laudos, documentos e mesmo sinais vitais de tempo real. Os aplicativos desenvolvidos estão disponíveis sob licença gratuita e são de código-fonte aberto. O PEP possui a característica de ser independente de plataforma, faz uso de um sistema a parte de autenticação de código-fonte livre baseado no OPENLDAP [18] chamado MACA [19], de um sistema de prontuário chamado MiniWebPEP, base de dados PostgreSQL [20] e base de imagens através do sistema PACS. A arquitetura do PEP InCor pode ser dividida em camadas onde a parte superior pertence ao sistema web que trabalha em conjunto com o servidor Web e o servidor de autenticação CORBA. A camada mais inferior é onde está a toda a base de dados estruturada contendo o conjunto de prontuários, das imagens DICOM, documentos digitalizados e dos laudos XML. Na Figura 5 é exibido uma tela do sistema de prontuário do InCor.

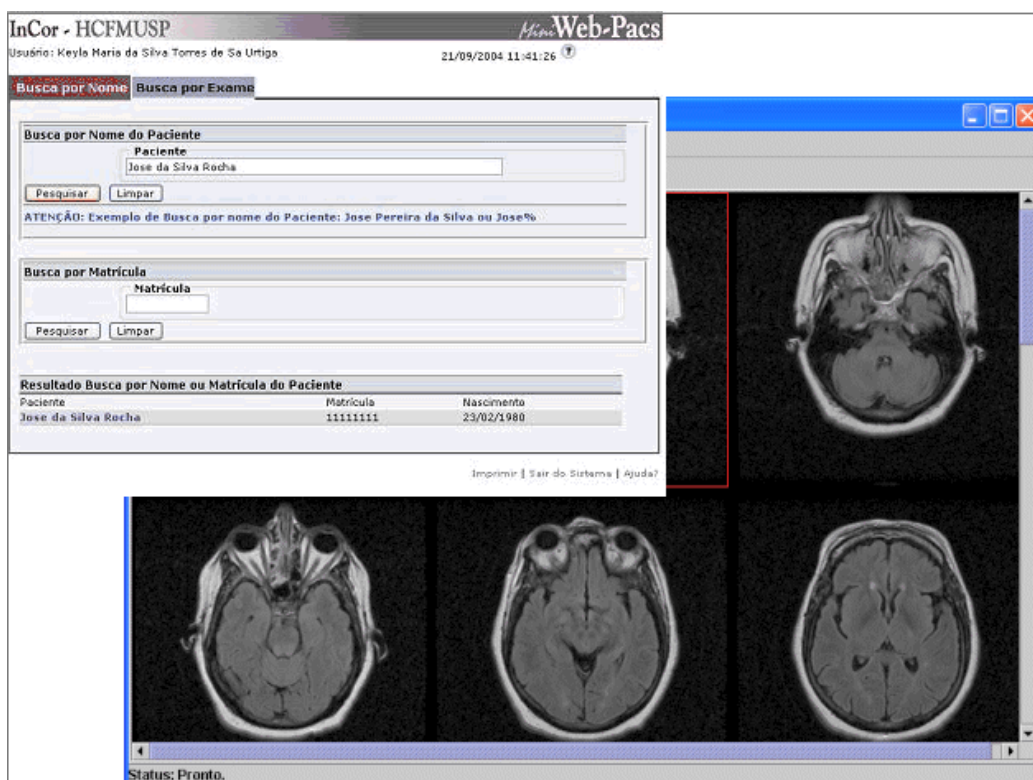


Figura 5. Interface Gráfica do Prontuário eletrônico do InCor [21].

Vantagens

- Código fonte aberto;
- Criado para ser heterogêneo e distribuído.
- Criado para ser acessado remotamente através de um navegador de Internet e assim estar disponível 24 horas;
- Já testado com sucesso pelo Hospital das Clínicas de São Paulo;
- É Gratuito e disponível na Internet;
- Alto nível de segurança, o projeto foi implementado seguindo padrões de segurança.

Desvantagens

- Projeto sem novas atualizações;
- Fraco suporte oferecido. Há um site de suporte com informações desatualizadas e Email para contato;
- Difícil de instalar. É necessário um especialista em informática para implantar e configurar o sistema em um servidor.

2.4.3 Aplicativo 3: Dental Office

Dental Office [22] é um aplicativo proprietário, pago, que está no mercado desde 1993 voltado exclusivamente para profissionais de odontologia. Ele possui uma seção de administração clínica e de prontuários de pacientes. Atualmente, há duas versões deste aplicativo: uma para funcionar no *desktop* e outra disponível como um sistema *web* de gerência de consultórios. Na Figura 6, temos uma exemplificação da interface de usuário do sistema.

Vantagens

- Possui suporte via o site de Internet, Email e telefone;
- Projeto sempre em desenvolvimento e atualização.

- Há duas versões, uma para computadores pessoais e outra que é um serviço online de prontuário disponibilizado no próprio servidor do fabricante;
- Possui uma boa usabilidade;
- A versão para computadores pessoais possui uma fácil instalação e configuração sem necessitar de um profissional especializado em informática;

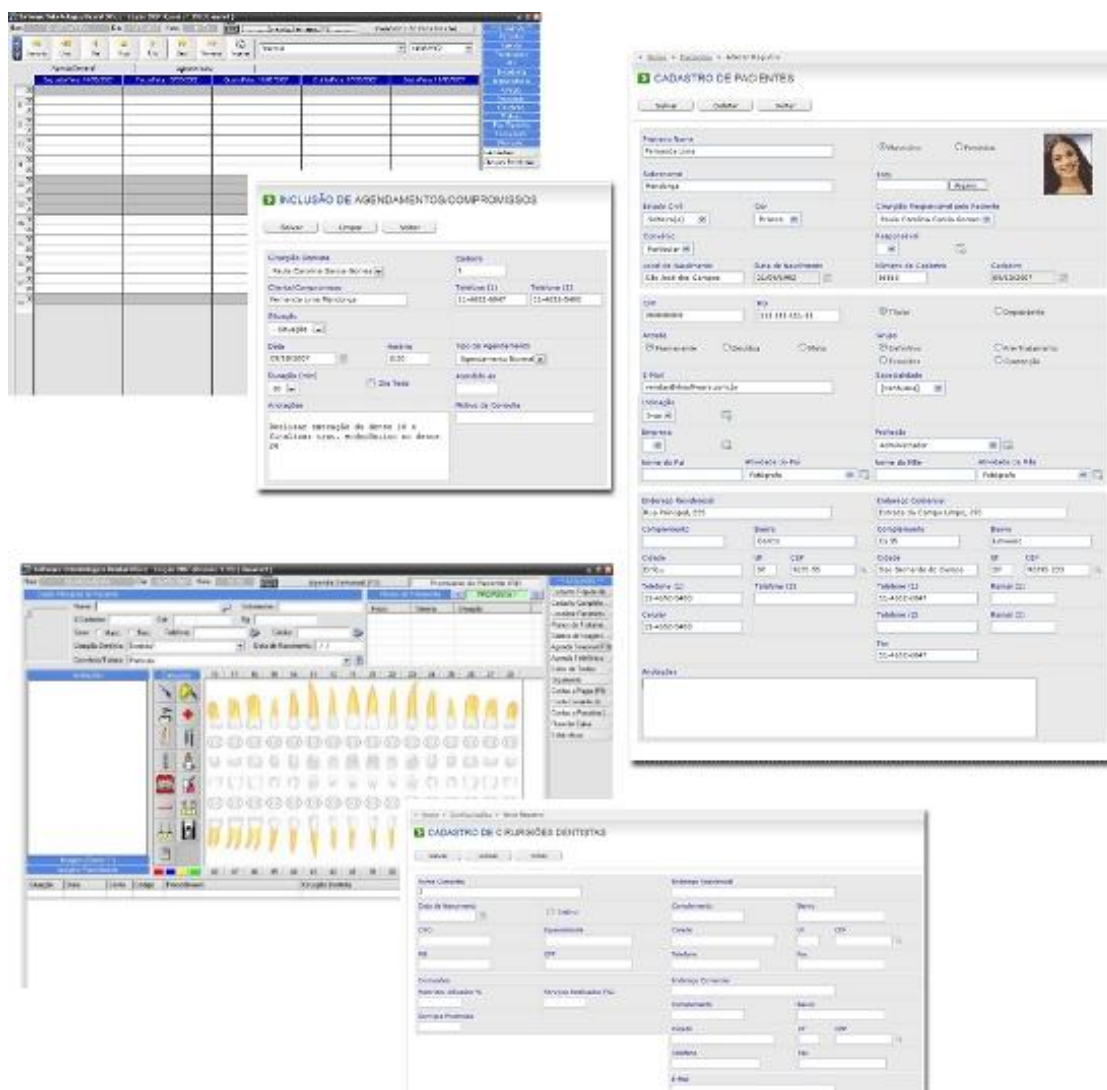


Figura 6. Interface de usuário do Prontuário eletrônico do Dental Office [22].

Desvantagens

- É um sistema que atende exclusivamente à necessidade da área odontológica;

- Licença proprietária, o sistema tem preço único para a versão para computadores pessoais e uma mensalidade para a versão *online*;
- A versão para computador pessoal é feita somente para a plataforma proprietária Microsoft Windows;
- Não adaptável possíveis às novas necessidades do usuário;
- A versão para computador pessoal não possui as características de ser heterogêneo e nem distribuído.

2.5 conclusão

O gerenciamento de dados clínicos é essencial para uma boa administração hospitalar e um bom acompanhamento do quadro clínico dos pacientes. O uso da tecnologia no armazenamento e manipulação dos dados traz inúmeras vantagens quanto à segurança, agilidade e qualidade no atendimento de pacientes. Grandes instituições, como o InCor (Instituto de Coração do Hospital das Clínicas), vivenciou na prática as melhorias alcançadas pela adoção do PEP como padrão de prontuários médicos. A tendência é que mais e mais hospitais e clínicas sigam a mesma trajetória de sucesso do InCor e de muitas instituições que já usam os prontuários eletrônicos. A Tabela 1 mostra um resumo comparativo entre as três soluções estudadas levando-se em conta os critérios mais importantes em um sistema de prontuários eletrônicos.

Tabela 1. Comparativo entre os aplicativos analisados.

Critérios	Aplicativo 1	Aplicativo 2	Aplicativo 3
Suporte ao cliente	Bom.	Frac.	Bom.
Suporte a imagens Médicas	Frac.	Bom.	Frac.
Portabilidade	Fraca.	Boa.	Fraca.
Usabilidade	Boa.	Boa.	Boa.
Manutenção	Fácil.	Difícil.	Fácil.

Preço	É pago.	É gratuito.	É pago.
Distribuído	Não.	Sim.	Depende da versão.
Heterogêneo	Não.	Sim.	Não.
Seguro	Não.	Sim.	Sim.

Para que um sistema de gestão de dados clínicos seja considerado realmente um sistema de gerenciamento de prontuário eletrônico do paciente, deve-se possuir basicamente três características: ser distribuído, heterogêneo e seguro. Através do comparativo realizado entre os três aplicativos na Tabela 1, pode-se perceber que somente a solução apresentada pelo InCor é de fato um prontuário eletrônico do paciente.

Capítulo 3

PEPNET: O que é, características, motivação

3.1 Metodologia de Desenvolvimento

O processo de desenvolvimento se guiou através da metodologia ágil de desenvolvimento [23], mas também fez uso de UML. Foi priorizado a implementação de um sistema funcional em detrimento de uma elaboração de documentos detalhados. Neste processo as interações são mais importantes que os processos e ferramentas e a capacidade de resposta às mudanças foi considerada mais importante que uma obediência a um planejamento rígido [24].

3.1.1 Processo de Desenvolvimento

Para o projeto do sistema PEPNET, foi adotada uma combinação da metodologia ágil XP (extreme programming) e a metodologia RUP (Rational Unified Process), mais especificamente, no uso da linguagem UML para descrição de casos de uso. Dentre as técnicas adotadas para o processo de desenvolvimento, deve-se destacar:

- Participação do cliente através de entrevistas iniciais e reuniões periódicas;
- Elaboração dos artefatos do XP (tarefas visuais, iterações, quadro de histórias, etc.);
- Descrição de funcionalidades por meio de casos de uso.

Importante ressaltar que essa forma de processo combinando metodologias foi escolhida porque estava de acordo com a realidade do projeto PEPNET em que a equipe é composta apenas de uma pessoa que é responsável por todas as fases do projeto (planejamento, análise, implementação e testes). É também uma tendência hoje em dia em selecionar quais técnicas são as mais adequadas das diversas

metodologias existentes, para uma gestão eficiente de projetos com curto prazo de tempo para entrega.

3.1.2 Análise de Requisitos

A idéia de elaboração de um sistema de prontuários eletrônicos do paciente surgiu da necessidade da Clínica São Mateus em implantar inicialmente um sistema de consultório digital unindo a gestão de pacientes das áreas de cardiologia e odontologia dos agentes de saúde dessa instituição. Notou-se que os profissionais de saúde da clínica precisavam organizar melhor a gestão dos prontuários dos pacientes. O levantamento de requisitos [5] foi feito através de reuniões com um médico e um dentista, na obtenção de modelos de prontuários de papéis (que estão disponíveis no apêndice A) e na análise apresentada no capítulo 2. A Tabela 2 mostra o período e as etapas da atividade de análise de requisitos. Foi necessário um mês para que todos os requisitos fossem levantados e então seguir para a fase de criação dos casos de uso.

Tabela 2. Período das atividades da análise de requisitos

Atividade	Mês
Primeiras reuniões com os médicos e dentistas da clínica.	Julho/2009
Criação do primeiro protótipo em papel a ser discutido com os profissionais de saúde.	Julho/2009
Revisão do primeiro protótipo e geração do 2º protótipo em papel.	Julho/2009

Quanto aos requisitos não-funcionais, foram levantados:

- Usabilidade, o sistema terá como usuários profissionais de saúde que não possuem total domínio quanto aos assuntos específicos da área de informática;
- Distribuição, o acesso do sistema deve ser realizado pela Internet;

- Segurança, o acesso deve acontecer apenas através de funcionários cadastrados no sistema.

Os casos de uso [5] (que estão disponíveis no apêndice B) especificam os comportamentos de um sistema ou parte dele e consiste da descrição conjunto de seqüência de ações e variantes realizadas pelo sistema para produzir um resultado observável do valor de um ator.

Os casos de uso foram definidos a partir das seguintes funcionalidades:

- Validação do usuário do sistema através de um nome de usuário e senha;
- Gerenciar administrador, relacionado com cadastro, consulta e remoção de administrador no sistema;
- Gerenciar funcionários, relacionado com cadastro, consulta e remoção de funcionários no sistema;
- Gerenciar agenda, relacionado a cadastro, pesquisa e remoção de consultas de pacientes;
- Gerenciar pacientes, relacionado com cadastro, consulta e remoção de pacientes no sistema;
- Gerenciar anamnese dos pacientes, relacionado com edição e atualização do documento de anamnese de cada paciente cadastrado no sistema;
- Gerenciar odontograma dos pacientes, relacionado com edição e atualização do documento de odontograma de cada paciente de odontologia cadastrado no sistema;
- Gerenciar medicamentos, relacionado com cadastro, consulta e remoção de informação de medicamentos no sistema;
- Gerenciar laboratórios, relacionado com cadastro, consulta e remoção de informação de laboratórios no sistema;
- Consulta ao CID (Classificação Internacional de Doenças e Problemas Relacionado à Saúde);
- Visualizar e gerar relatórios a partir de dados dos pacientes no sistema;

- Gerenciar o banco de dados, relacionado a realizar cópia de segurança e restauração do banco de dados do sistema.

No apêndice C encontram-se os diagramas de caso de uso do sistema PEPNET.

3.1.3 Análise de Risco

Foram levantados os riscos [5] do desenvolvimento do sistema PEPNET e as devidas ações de prevenção para se evitar problemas graves de gerência ao longo do ciclo de vida do projeto. Os riscos foram divididos de acordo com o escopo do problema, sob os aspectos de segurança, capacitação em relação ao usuário e gestão, análise e implementação em relação ao processo de desenvolvimento do projeto. As análises estão apresentadas na Tabela 3 e na Tabela 4.

Tabela 3. Análise de Riscos para o Escopo Processos

Análise de Riscos para o Escopo Humano		
Tipo	Descrição	Ação Preventiva
Segurança	Os usuários do sistema podem utilizar o sistema de forma inadequada compartilhando com terceiros o nome de usuário e senha o que prejudica a confidencialidade das informações dos pacientes.	Orientar o cliente a estabelecer uma política educativa na instituição de como usar o sistema ressaltando a importância do código de ética médico e odontológico.
Capacitação	A falta de capacitação dos usuários pode fazer com que o sistema não seja utilizado da forma adequada ou até abandonado em troca do uso de papel na gerência dos prontuários.	Orientar o cliente a estabelecer uma política capacitação dos funcionários que irão utilizar o sistema.

Tabela 4. Análise de Riscos para o Escopo Geral

Análise de Riscos para o Escopo Geral		
Tipo	Descrição	Ação Preventiva
Gestão	A pouca experiência em gestão de projetos pode ocasionar em decisões erradas durante o ciclo de vida do projeto.	Estudo aprofundado dos conceitos de gerência de projetos e metodologias de desenvolvimento.
Análise	Análise mal feita pode acarretar no desvio do objetivo final do projeto.	Estudo através de livros e artigos sobre elaboração de análises de requisitos.
Implementação	O aprendizado da tecnologia usada para o desenvolvimento do projeto ocorre na fase de implementação o que pode dificultar a entrega do projeto no prazo estabelecido.	Estudo através de livros e artigos sobre as tecnologias utilizadas.

3.1.4 Análise e Projeto

MVC (Model-View-Controller) [25] [23] é uma arquitetura (padrão ou modelo) popular para desenho de aplicações *Web* complexas e modulares. O padrão isola a lógica de negócio da entrada e da apresentação e permite o desenvolvimento, teste e manutenção das partes de forma independente. Na camada de modelo (*model*), estão as classes que modelam os dados e a lógica de negócio. Em geral, podem ser representadas pelas tecnologias *beans* e EJBs. A camada de controle (*controller*) normalmente é utilizada para filtro, redirecionamento de páginas e validações de dados. Normalmente, podem ser implementadas através das tecnologias *servlets*, *beans* ou EJBs. Por fim, na camada de visão (*view*) estão as páginas estáticas e dinâmicas responsáveis por exibir informações na tela aos usuários.

Testes de unidade [23] são muito importantes para o desenvolvimento de projeto pois garantem a validação lógica das classes e das regras de negócio. O *framework* Struts facilita o teste de classes de ação em um ambiente isolado. Uma

chave para facilitar a execução de testes é assegurar que qualquer objeto com o qual a ação interaja seja injetado na classe através de um *setter*. No apêndice D encontra-se disponível algumas classes de testes utilizadas no sistema PEPNET.

3.1.5 Implementação

Nesta seção serão apresentadas as tecnologias utilizadas no desenvolvimento do sistema PEPNET: o gerenciador de projetos Maven2, os *frameworks* Struts, Hibernate, Spring e SiteMesh, a biblioteca Acegi, JSP.

Maven2

Maven2 [23] [26] é um gerenciador de projetos e consiste de uma ferramenta de linha de comando usada para construir, testar, emitir relatório e empacotar projetos.

Maven provê facilidades para que auxiliem o desenvolvimento de projetos, como:

- Estrutura padronizada de diretório: cada projeto que faz uso do Maven terá a mesma estrutura de diretório;
- Arquitetura *plug-in*: cada função de Maven é efetuada por um *plug-in*, seja compilar ou utilizar o site. Quando o projeto é testado ou utilizado pela primeira vez, o Maven gerencia o *download* dos *plug-ins* necessários para a execução do projeto e os armazena em um repositório comum;
- Gerenciamento de dependência: quando dependências forem descritas no arquivo de configuração Maven, elas serão acessadas de um repositório local ou baixadas do repositório local durante o processo de construção;
- Gerenciamento de escopo: o pacote final de distribuição contém apenas os elementos exigidos. Os códigos de teste e as dependências que não são necessários ao pacote final são deixados de fora;
- Arquétipos: o *plug-in* de arquétipos permite que os desenvolvedores criem um *template default* de implementação para uma categoria de projeto. Este é então usado para criar um novo projeto rapidamente sem a necessidade de criar a estrutura comum de diretório, criar os arquivos de configuração e codificar as classes *default* e testes a partir do zero.

Framework Apache Struts

Apache Struts [23] [27] é um *framework* de aplicativos *web* para o desenvolvimento de programas em Java para *Web*. Esse *framework* faz uso estende da API do Java *Servlets* para estimular o uso por parte dos desenvolvedores da arquitetura Model-View-Controller (MVC).

Framework Hibernate

Hibernate [23] [28] é um *framework* usado para mapear objetos-relacionais escritos em Java. Ele facilita o mapeamento de atributos entre uma base de dados relacionais e o modelo objeto de uma aplicação. Tem por objetivo diminuir a complexidade da interface de comunicação entre o aplicativo Java e o banco de dados relacional através da transformação das classes em Java para tabelas de dados. O Hibernate cria os códigos SQL e livra os desenvolvedores do trabalho manual da conversão dos dados para o padrão do banco de dados. A grande vantagem é a portabilidade do aplicativo desenvolvido porque fica independente de um banco de dados específico.

Framework Spring

Spring [23] [29] é um *framework* de código-fonte livre que provê diversas funcionalidades usadas para desenvolver aplicativos Java. Com o Spring, pode-se configurar os componentes da aplicação e gerenciar o tempo de vida dos objetos. O Spring fornece ferramentas para se trabalhar com SGBD relacionais usando o JDBC e ferramentas de mapeamento de objetos relacionais. Abaixo, algumas das funcionalidades do Spring:

- O módulo Spring Core possui as principais funcionalidades do Spring, dentre elas, uma implementação responsável em remover a programação de *Singletons* e permitindo o baixo acoplamento entre a configuração e a especificação de dependências, de sua lógica de programação;
- O módulo Spring DAO fornece uma camada de abstração para JDBC, eliminando grande parte da codificação necessária para interagir com um banco de dados.
- O módulo ORM fornece uma integração do Spring com outros *frameworks* para persistência de objetos como Hibernate.

- O módulo AOP fornece uma implementação de orientação a aspectos;
- O módulo Spring Web fornece funcionalidades específicas para o projeto *Web* como carregamento de arquivos e suporte para utilização de inversão de controle;
- O módulo Spring MVC fornece uma implementação de *framework Web*, semelhante ao Struts.

O Spring possui uma arquitetura baseada em interfaces e POJOs (Plain Old Java Objects), oferecendo aos POJOs características como mecanismos de segurança e controle de transações. Também facilita testes unitários e surge como uma alternativa à complexidade existente no uso de EJBs.

Framework Acegi

O *framework* Acegi [23] [30] (também conhecido como Spring Security) fornece ferramentas de segurança, como autenticação segura através de criptografia da senha e lista de controle de acesso, para ser usado em conjunto com o *framework* Spring. Esta biblioteca tem por função evitar que o desenvolvedor necessite elaborar e implementar toda a infra-estrutura de segurança do sistema *web*. Foi criado originalmente para a comunidade Spring e faz uso intenso das bibliotecas do *framework* Spring em sua implementação.

A integração com o Acegi fornece um controle de alto nível sobre a configuração e autorização de usuários que usam o aplicativo *web*. Usuários podem ser criados pela lógica de seu aplicativo e os mesmos serviços de negócio podem ser usados para verificar usuários para autenticação, o que evita a aplicação de qualquer *servlet* externo.

Framework SiteMesh

SiteMesh [23] [31] é um *framework* de integração e de *layout* de páginas de Internet e é usado para auxiliar a criação de grandes sites que possui muitas páginas e necessitam de consistência na aparência. O SiteMesh intercepta requisições para qualquer página *Web*, estática ou dinâmica, faz um *parsing* no conteúdo e gera a página final.

Java Server Pages e o Servidor Apache Tomcat

JSP [25] [32] (Java Server Pages) é uma tecnologia para desenvolvimento de páginas de *Internet* de conteúdo dinâmico. Por ser baseada em Java, possui a característica de ser independente da plataforma. Atualmente, há dois famosos servidores Java que suportam a tecnologia JSP: o Apache Tomcat [33] (padrão em hospedagem de páginas JSP) e o Sun Glassfish. Foi adotado para o projeto PEPNET o servidor Tomcat por fornecer uma compatibilidade maior com o *framework* Struts que também pertence à empresa Apache e por ser o padrão na *Web*.

3.2 O sistema PEPNET

O PEPNET foi desenvolvido como resposta ao problema de gerenciamento inadequado e descentralizado da clínica São Mateus. O PEPNET foi criado a fim de substituir dois aplicativos utilizados na clínica, um para cardiologia e outro para odontologia, que não estão de acordo com o código de ética médico e odontológico pois não são de fato prontuários eletrônico. Por não serem prontuários eletrônicos, há a necessidade de se utilizar os prontuários de papeis (que é uma forma legalizada) na gestão dos dados clínicos. O PEPNET vem para diminuir os gastos nessa gerência redundante e unificar a gestão dos prontuários da clínica São Mateus.

O sistema PEPNET possui as seguintes funcionalidades desenvolvidas:

- Validação de usuário – através do *framework* Spring Security;
- Consulta ao CID – funcionalidade que é útil para a elaboração de atestados;
- Gerenciamento dos usuários – criação, atualização de perfis e remoção de usuários;
- Gerenciamento de pacientes usuários – criação, atualização de perfis e remoção de pacientes;
- Gerenciamento de anamnese usuários – atualização de anamnese;
- Gerenciamento de convênios usuários – criação, atualização de perfis e remoção de planos de saúde;

- Gerenciamento de Medicamentos usuários – criação, atualização de perfis e remoção de medicamentos;

3.2.1 Interface Gráfica

A construção da interface gráfica foi feita através do uso de *templates*, do *framework* SiteMesh, de HTML e JSP. Para cada funcionalidade descrita na seção 3.1.2, **Erro! Fonte de referência não encontrada.** há uma tela do sistema correspondente. Abaixo há uma descrição de algumas telas do sistema:

Análise da Tela de Efetuar Validação do Usuário

A tela exibida na Figura 7 Tela de Validação dos usuários do sistema corresponde à validação do usuário e é realizada fornecendo um email válido no campo Nome e uma senha de seis a doze dígitos no campo senha. Todas as demais telas do sistema (com exceção da tela inicial) dependem da validação do usuário para serem acessíveis.

Análise da Tela de Editar Perfil do Usuário

A tela exibida na Figura 8 corresponde à funcionalidade de atualização dos dados do usuário. Há um campo para o nome, senha, email e retrato.

Análise da Tela Principal do Sistema

A tela da Figura 9 corresponde à página principal e inicial do sistema. Nota-se que abaixo do menu horizontal da tela há um espaço com imagens e textos, este contudo inserido é apenas para exemplificar a possibilidade futura de personalização do espaço do sistema com notícias RSS, *banners* e assim por diante.

Análise da Tela de Gerenciamento dos Pacientes

A tela conforme a Figura 10 é responsável por oferecer as quatro funcionalidades básicas no gerenciamento de um paciente: cadastrar, pesquisar, editar e excluir.

Análise da Tela de Consulta do CID

O CID é composto por mais de 12 mil código de doenças. A tela da Figura 11 mostra justamente a funcionalidade de pesquisa do código do CID além de listar as doenças associada a cada código.



Figura 7. Tela de Validação dos usuários do sistema.

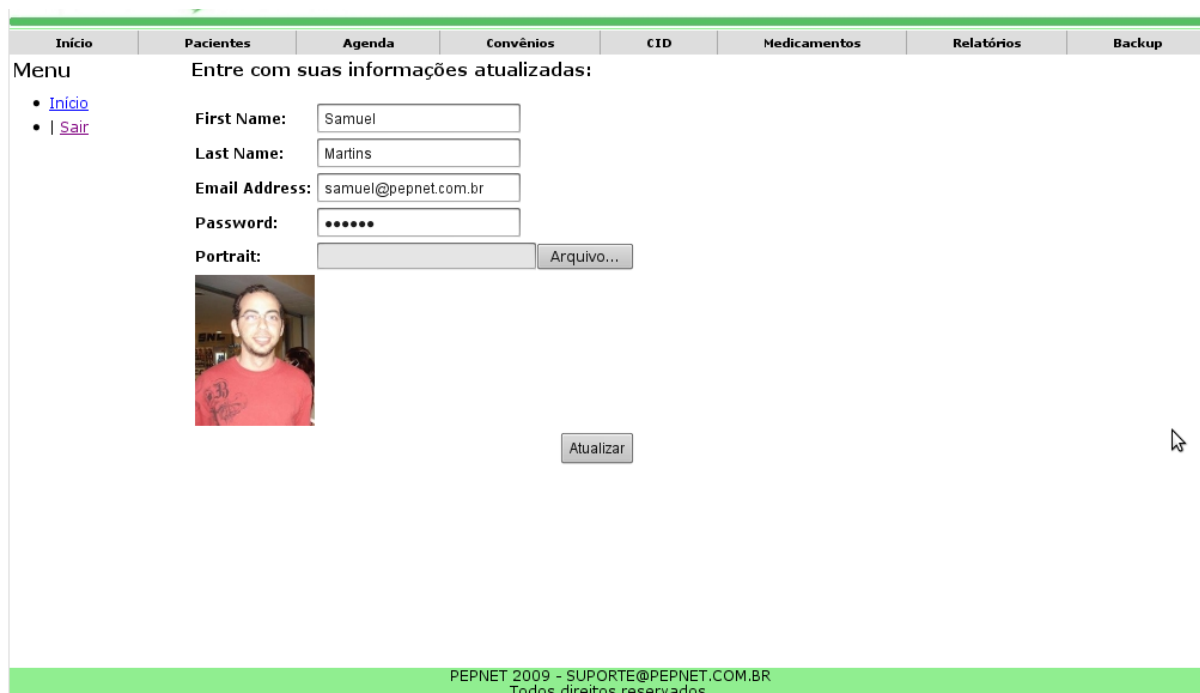


Figura 8. Tela de atualização de perfil de um usuário do sistema.

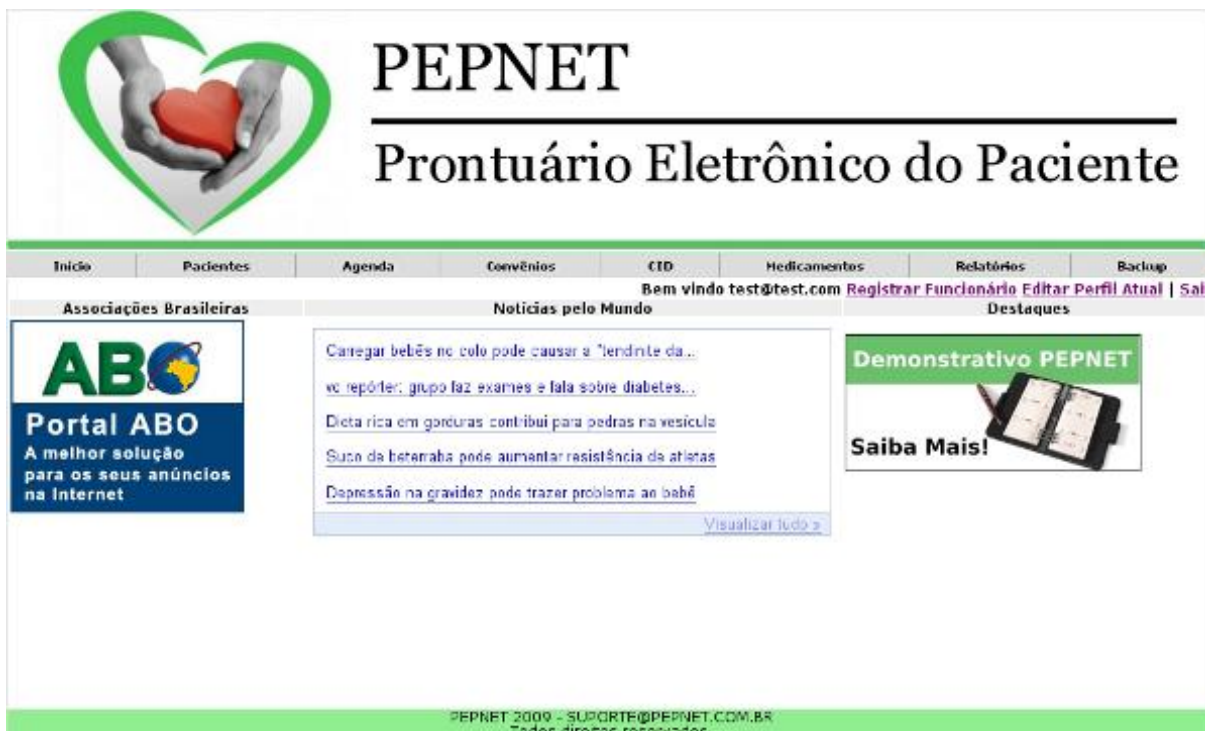
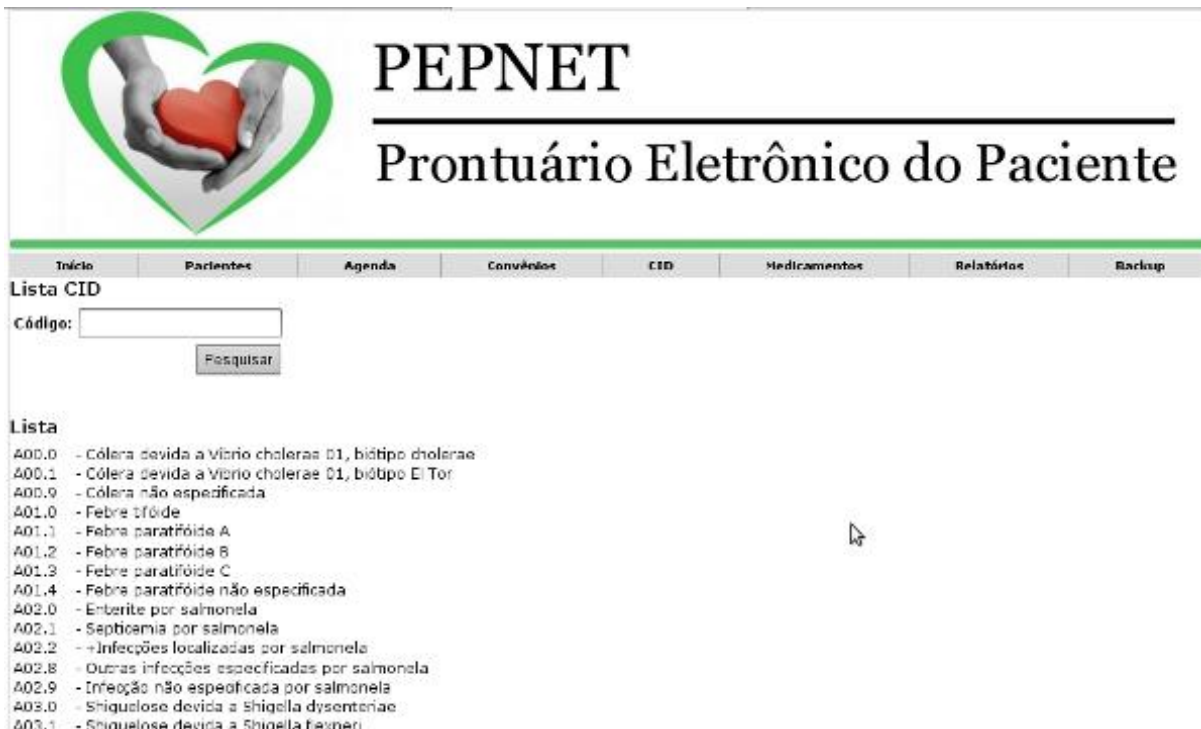


Figura 9. Tela inicial e principal do sistema.



Figura 10. Tela de Gerenciamento de Pacientes.



PEPNET
Prontuário Eletrônico do Paciente

Início Pacientes Agenda Convênios CID Medicamentos Relatórios Backup

Lista CID

Código:

Pesquisar

Lista

- A00.0 - Cólera devida a *Vibrio cholerae* 01, biotipo cholerae
- A00.1 - Cólera devida a *Vibrio cholerae* 01, biotipo El Tor
- A00.9 - Cólera não especificada
- A01.0 - Febre tifóide
- A01.1 - Febre paratífóide A
- A01.2 - Febre paratífóide B
- A01.3 - Febre paratífóide C
- A01.4 - Febre paratífóide não especificada
- A02.0 - Enterite por salmonela
- A02.1 - Septicemia por salmonela
- A02.2 - Infecções localizadas por salmonela
- A02.8 - Outras infecções especificadas por salmonela
- A02.9 - Infecção não especificada por salmonela
- A03.0 - Shigelose devida a *Shigella dysenteriae*
- A03.1 - Shigelose devida a *Shigella flexneri*

Figura 11. Tela de pesquisa do CID.

3.2.2 Vantagens e Desvantagens

O sistema PEPNET é um sistema que de fato pode ser chamado de prontuário eletrônico do paciente porque possui características de ser heterogêneo, distribuído e seguro. O sistema é heterogêneo porque aborda mais de uma área de saúde, cardiologia e odontologia; é distribuído porque o seu acesso é distribuído, o usuário pode acessar o aplicativo de onde estiver bastando possuir um ponto de acesso a Internet; e é seguro porque a tecnologia utilizada para construção do aplicativo, o *framework Spring Security*, garante essa qualidade ao gerenciar a validação de usuário e proteger contra técnicas comuns de ataques de *hackers*. Essas três características apresentadas são justamente as vantagens do sistema PEPNET.

Além das vantagens citadas anteriormente, o sistema PEPNET possui as seguintes qualidades:

- É gratuito;
- De fácil instalação;
- Possui uma boa usabilidade;
- É portátil a diversos sistemas operacionais.

A desvantagem do sistema é que ele ainda não possui todas as funcionalidades implementadas e desejadas em um aplicativo de gestão clínicas, como:

- Não há suporte a imagens médicas;
- Não há suporte ao cliente;
- Não há um gerenciador de relatórios;
- Não realiza backup dos dados armazenados;
- Não possui um gerenciador de agenda.

3.2.3 Futuro do PEPNET

Até o final deste trabalho, a construção do sistema PEPNET não ficou totalmente concluída o que impossibilitou de ser implantada na clínica São Mateus. Não há intenção de vender o sistema para clínica, a idéia é que o PEPNET se torne uma solução gratuita e que seja adotada alguma licença de software livre para o aplicativo. PEPNET vai ser tornar produto que estará disponível para download gratuitamente na Internet sob o endereço www.pepnet.com.br que já foi registrado com sucesso. As funcionalidades que não foram desenvolvidas até o momento da finalização deste trabalho foram deixadas para trabalhos futuros.

Capítulo 4

Conclusão e Trabalhos Futuros

4.1 Contribuições

O PEP mostrou-se como uma área de estudo multidisciplinar bastante interessante em pesquisas acadêmicas como também na área de venda de *softwares* para um mercado ainda carente de um sistema unificador de prontuários do paciente. O PEP surge para atuar como uma ferramenta de melhoria da gestão da saúde, no auxílio dos profissionais a trabalharem conforme o código de ética e no melhor atendimento aos pacientes. É um campo ainda em expansão e com bastante potencial para estudos futuros.

O sistema PEPNET foi desenvolvido com um pacote de funcionalidades básicas e essenciais para a gestão de prontuários dos pacientes. Por ser um sistema acessível via Internet, os profissionais de saúde contam com a possibilidade de acessar os dados clínicos de qualquer terminal conectado a rede. A adoção do PEPNET como um novo modelo de gestão de dados clínicos está de acordo com uma das necessidades mais importantes da atualidade: uma gestão em prol da preservação da natureza e contra o desperdício através do uso do sistema eletrônico ao invés dos prontuários em papel.

Assim, o gerenciamento de dados clínicos informatizado traz inúmeros benefícios na administração clínica e hospitalar: segurança, agilidade e qualidade no atendimento de pacientes, integração dos dados, legibilidade, entre outros. Grandes instituições, como a InCor, já tiveram a experiência de reformular a gestão dos dados clínicos e vivenciaram na prática as melhorias alcançadas pela adoção do PEP como padrão de prontuários médicos. O PEPNET surge como uma alternativa a ser analisada para se atingir metas de uma boa administração dos pacientes.

4.2 Trabalhos Futuros

Por ser um campo vasto, não foi possível abordar neste projeto todas as características que fazem parte de um prontuário do paciente. Então foi levantando uma lista de estudo e funcionalidades que podem complementar este trabalho:

- Criação de um gerenciador ou adoção de um *framework* que gerencie imagens médicas que suporte o padrão de imagens DICOM;
- Criação de um gerenciador ou adoção de um *framework* que gerencie cópia de segurança e restauração da base dados do sistema;
- Criação de um gerenciador ou adoção de um *framework* que gerencie relatórios;
- Criação de um gerenciador ou adoção de um *framework* que gerencie eventos da agenda médica;
- Criação de um gerenciador ou adoção de um *framework* que gerencie notícias internas e externas com suporte a RSS;
- Melhoria da interface gráfica com a utilização da tecnologia AJAX;
- Melhoria o sistema de validação de usuários adicionando a funcionalidade de gestão de níveis de acesso;
- Pesquisa de outras áreas de saúde que necessitem de um sistema PEP, mas se encontram carentes de um aplicativo que as atenda.

Bibliografia

- [1] TRIANA, A. **A semiologia biomédica e seus limites: desvendando o caminho entre o sutil e o evidente**. UNICAMP. Campinas. 1999.
- [2] MARTINS, S. **Relatório técnico sobre prontuário do paciente para odontologia e medicina**. Universidade de Pernambuco. Recife, p. 27. 2009.
- [3] OSLER, W. **The Evolution Of Modern Medicine: A Series Of Lectures Delivered At Yale University In April 1913**. New York: Kaplan, 2009. p. 15-18,32.
- [4] COSTA, C.; MARQUES, A. **Implementação de um Prontuário Eletrônico do Paciente na Maternidade Escola Januário Cicco: Um Primeiro Passo**, Rio de Janeiro, 1999.
- [5] COSTA, C. G. A. **Desenvolvimento e Avaliação Tecnológica de um Sistema de Prontuário Eletrônico do Paciente, Baseado nos Paradigmas da World Wide Web e da Engenharia de Software**, Universidade Estadual de Campinas, Campinas, 2001.
- [6] W3C. **Extensible Markup Language (XML)**. Disponível em: <<http://www.w3.org/XML/>>. Acesso em: 18 dez. 2009.
- [7] OMG. **Welcome To The OMG's CORBA Website**. Disponível em: <<http://www.corba.org/>>. Acesso em: 18 dez. 2009.
- [8] BRASIL. **Resolução CFO-42/2003**, Rio de Janeiro, 20 Maio 2003.
- [9] BRASIL. **Resolução CFM nº 1.246/88**, Rio de Janeiro, p. 23, 08 Setembro 2009.
- [10] BRASIL. **Resolução CFM nº 1.638/2002**. Disponível em: <http://www.portalmedico.org.br/resolucoes/cfm/2002/1638_2002.htm>. Acesso em: 18 dez. 2009.
- [11] BRASIL. **Resolução CFM nº 1.639/2002**. Disponível em: <http://www.portalmedico.org.br/resolucoes/cfm/2002/1639_2002.htm>. Acesso em: 18 dez. 2009.

- [12] FIGUEREDO, J. C. B. **Sistema Compacto para Armazenamento e Distribuição de Imagens Médicas em Ambientes Clínico-Hospitalares**, São Paulo, 2004.
- [13] URTIGA, K. S. **Desenho de Interface para Sistema de Armazenamento e Distribuição de Imagens Médicas em Ambientes Clínico-Hospitalares**, São Paulo, 2004.
- [14] FURUIE, S. S.; GUTIERREZ, M. A. **Prontuário Eletrônico de Pacientes: integrando informações clínicas e imagens médicas**. Revista brasileira de engenharia biomédica, São Paulo, v. 19, p. 125 - 137, 01 Dezembro 2003. ISSN 1517-3151.
- [15] BABYLON. **DICOM | Portuguese | Dictionary & Translation by Babylon**. Disponível em: <<http://www.babylon.com/definition/DICOM/Portuguese>>. Acesso em: 14 Setembro 2009.
- [16] OLIVEIRA, C. C. **Consultório Digital – Guia do Usuário**. 2ª Edição. ed. [S.l.]: Sociedade Brasileira de Cardiologia, 2009.
- [17] SBC. **Sociedade Brasileira de Cardiologia - Consultório Digital**. Disponível em: <<http://consultorio.cardiol.br/>>. Acesso em: 14 Setembro 2009.
- [18] OPENLDAP FOUNDATION. **OpenLDAP - Main Page**. Disponível em: <<http://www.openldap.org/>>. Acesso em: 14 Setembro 2009.
- [19] MOTTA, G. H. M. B.; FURUIE, S. S. **MACA: Uma Ferramenta de Autorização e Controle de Acesso para o Prontuário Eletrônico de Pacientes**, Natal, 2002.
- [20] POSTGRESQL. **PostgreSQL: The world's most advanced open source database**. Disponível em: <<http://www.postgresql.org/>>. Acesso em: 14 Setembro 2009.
- [21] ALMEIDA, C. A. P. **PRONTUÁRIO ODONTOLÓGICO – Uma orientação para o cumprimento da exigência contida no inciso VIII do art. 5º do Código de Ética Odontológica**, Rio de Janeiro, 2002.
- [22] RH! **Dental Office - Software Odontológico /Software para Odontologia**. Disponível em: <<http://www.dentaloffice.com.br/introducao.html>>. Acesso em: 14 set. 2009.

- [23] ROUGHLEY, I. **Practical Apache Struts 2 – Web 2.0 Projects**. 1ª Edição. ed. Rio de Janeiro: Ciência Moderna, 2008.
- [24] BECK, K. **Manifesto for Agile Software Development**. Disponível em: <<http://agilemanifesto.org/>>. Acesso em: 14 set. 2009.
- [25] FRANKLINT, K. **Java EE 5 - Guia Prático – Servlets, JavaBeans**. 2ª Edição. ed. São Paulo: Érica, 2007.
- [26] APACHE. **Maven - Welcome to Apache Maven**. Disponível em: <<http://maven.apache.org/>>. Acesso em: 14 set. 2009.
- [27] APACHE. **Apache Struts - Welcome**. Disponível em: <<http://struts.apache.org/>>. Acesso em: 14 set. 2009.
- [28] RED HAT. **hibernate.org - Hibernate**. Disponível em: <<https://www.hibernate.org/>>. Acesso em: 14 set. 2009.
- [29] SPRINGSOURCE. **SpringSource.org**. Disponível em: <<http://www.springsource.org/>>. Acesso em: 14 set. 2009.
- [30] SPRINGSOURCE. **Spring Security**, 14 set. 2009. Disponível em: <<http://static.springsource.org/spring-security/site/index.html>>.
- [31] OPENSYPHONY. **SiteMesh - SiteMesh Overview**. Disponível em: <<http://www.opensymphony.com/sitemesh/>>. Acesso em: 14 set. 2009.
- [32] SUN MICROSYSTEMS. **JavaServer Pages Technology**. Disponível em: <<http://java.sun.com/products/jsp/>>. Acesso em: 14 set. 2009.
- [33] APACHE. **Apache Tomcat**, 14 set. 2009. Disponível em: <<http://tomcat.apache.org/>>.
- [34] BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML – Guia Do Usuário**. 2ª Edição. ed. Rio de Janeiro: Campus, 2005.

Apêndice A

Análise de Casos de Uso

Análise de alguns Casos de Uso importantes do sistema PEPNET:

1 Validar Usuário

1.1 Classe User

Descrição: classe que contém as informações cadastrais de um usuário no sistema.

Tabela 5. Descrição da Classe User

Atributos	Descrição	Tipo
Email	<i>Login</i> de acesso do sistema.	string
Senha	Senha de acesso ao sistema.	string
Nome	Nome do usuário.	string
sobrenome	Sobrenome do usuário.	string
Retrato	Retrato do usuário.	string

1.2 Classe AcegilInterceptor

Descrição: classe que intercepta uma ação do sistema e verifica se há algum usuário validado no momento.

1.3 Classe BaseUserAction

Descrição: classe que cuida da ação de criar um objeto Usuário com todas as configurações passadas pela tela de registrar ou atualizar perfil de usuário.

1.4 Classe UserAction

Descrição: semelhante à BaseUserAction.

1.5 Classe UpdateUserAction

Descrição: classe responsável por passar o objeto usuário em questão para a classe UserServiceImpl.

1.6 Classe UserServiceImpl

Descrição: classe responsável por gravar o objeto usuário em questão no banco de dados utilizando o *framework* Hibernate.

Tabela 6. Descrição da Classe UserServiceImpl

Atributos	Descrição	Tipo
emf	instância utilizada pelo aplicativo para obter a entidade de gerenciamento de aplicativo para a camada de dados.	EntityManagerFactory

2 Consultar CID

2.1 Classe Cid

Descrição: classe que contém as informações de um registro CID no sistema.

Tabela 7. Descrição da Classe Cid

Atributos	Descrição	Tipo
código	Código de acesso ao CID.	string
comentário	Descrição da doença referente ao código do CID.	string

2.2 Classe BaseCidAction

Descrição: classe que cuida da ação de criar um objeto Cid com todas as configurações passados pela tela de pesquisa de CID.

Tabela 8. Descrição da Classe BaseCidAction

Atributos	Descrição	Tipo
cid	Objeto CID.	Cid
service	Objeto da classe CidService.	CidService
resultado	Lista de objetos Cid	List<Cid>

2.3 Classe CidAction

Descrição: semelhante à BaseUserAction.

2.4 Classe FindCidAction

Descrição: classe responsável por chamar o método de pesquisa passando como argumento o código do objeto CID criado pelo BaseCidAction a ser pesquisado.

2.5 Classe ListarCidAction

Descrição: semelhante à FindCidAction.

2.6 Classe CidServiceImpl

Descrição: classe responsável por buscar o objeto cid em questão no banco de dados utilizando o *framework* Hibernate.

Tabela 9. Descrição da Classe CidServiceImpl

Atributos	Descrição	Tipo
emf	instância utilizada pelo aplicativo para obter a entidade de gerenciamento de aplicativo para a camada de dados.	EntityManagerFactory

Apêndice B

Especificação de Casos de Uso

Especificação de alguns Casos de Uso importantes do sistema PEPNET:

1 Validar Usuário

1.1 **Descrição:** autoriza a entrada do usuário no sistema

1.2 Fluxo Básico de Eventos:

- Na página de *login*, o usuário (médico ou dentista) insere seu email e senha. Pressiona o botão de entrar
- O usuário passa para página principal do sistema.

1.3 Fluxo Alternativo de Eventos:

- O usuário não está cadastrado no sistema
- O usuário insere um email ou senha errada. Uma mensagem de erro é exibida.

1.4 **Pré-condições:** não há.

1.5 **Pós-condições:** o usuário está autorizado pelo sistema ou uma mensagem de erro é exibida.

2 Atualizar dados do usuário

2.1 **Descrição:** altera email, nome, sobrenome, senha e foto do usuário.

2.2 Fluxo Básico de Eventos:

- Na página de atualização de perfil, o usuário (médico ou dentista) altera seu nome, sobrenome, *e-mail*, senha e foto. Pressiona o botão de atualizar
- O usuário passa para página principal do sistema.

2.3 Fluxo Alternativo de Eventos:

- O usuário não está mais no determinado momento validado pelo sistema

- O usuário insere um email ou senha inválidos. Uma mensagem de erro é exibida.

2.4 Pré-condições: não há.

2.5 Pós-condições: o usuário atualiza as informações do perfil no sistema ou uma mensagem de erro é exibida.

3 Registrar Usuário

3.1 Descrição: cadastra um usuário no sistema.

3.2 Fluxo Básico de Eventos:

- Na página principal do sistema, o usuário validado clica em registrar usuário.
- O usuário cadastra um novo usuário informando os campos de nome, sobrenome, email, senha e foto. Pressiona o botão de cadastrar.
- O usuário passa para página principal do sistema.

3.3 Fluxo Alternativo de Eventos:

- O usuário não está mais no determinado momento validado pelo sistema
- O usuário insere um *e-mail* ou senha inválidos. Uma mensagem de erro é exibida.

3.4 Pré-condições:

- O usuário que irá cadastrar o segundo usuário precisa estar validado no sistema.

3.5 Pós-condições:

- O usuário cadastrar o usuário no sistema ou uma mensagem de erro é exibida.

4 Registrar Paciente

4.1 Descrição: cadastra um paciente no sistema.

4.2 Fluxo Básico de Eventos:

- Na página principal do sistema, o usuário validado clica em paciente no menu horizontal do sistema.

- O usuário cadastra um novo paciente. Pressiona o botão de cadastrar.
- O usuário passa para página de gerenciar paciente do sistema.

4.3 Fluxo Alternativo de Eventos:

- O usuário não está mais no determinado momento validado pelo sistema
- O usuário insere um dados inválidos na ficha de cadastro do paciente. Uma mensagem de erro é exibida.

4.4 Pré-condições:

- O usuário que irá cadastrar um paciente precisa estar validado no sistema.

4.5 Pós-condições:

- O usuário cadastrar um paciente no sistema ou uma mensagem de erro é exibida.

5 Consultar CID

5.1 Descrição: Pesquisa o código e a informação correlata de uma doença no CID do sistema.

5.2 Fluxo Básico de Eventos:

- Na página principal do sistema, o usuário validado clica em CID no menu horizontal do sistema.
- O usuário insere um código a ser pesquisado. Pressiona o botão de pesquisar.
- A descrição e código da doença pesquisada são exibidas na tela.

5.3 Fluxo Alternativo de Eventos:

- O usuário não está mais no determinado momento validado pelo sistema
- O usuário insere um dados inválidos na ficha de cadastro do paciente. Uma mensagem de erro é exibida.
- O usuário insere uma doença a ser pesquisada. Pressiona o botão de pesquisar.
- A descrição e código da doença pesquisada são exibidas na tela.

5.4 Pré-condições:

- O usuário que irá consultar o CID precisa estar validado no sistema.

5.5 Pós-condições:

- O usuário consultar um CID no sistema ou uma mensagem de erro é exibida.

Apêndice C

Diagramas

1. Diagramas de Casos de Uso [34]

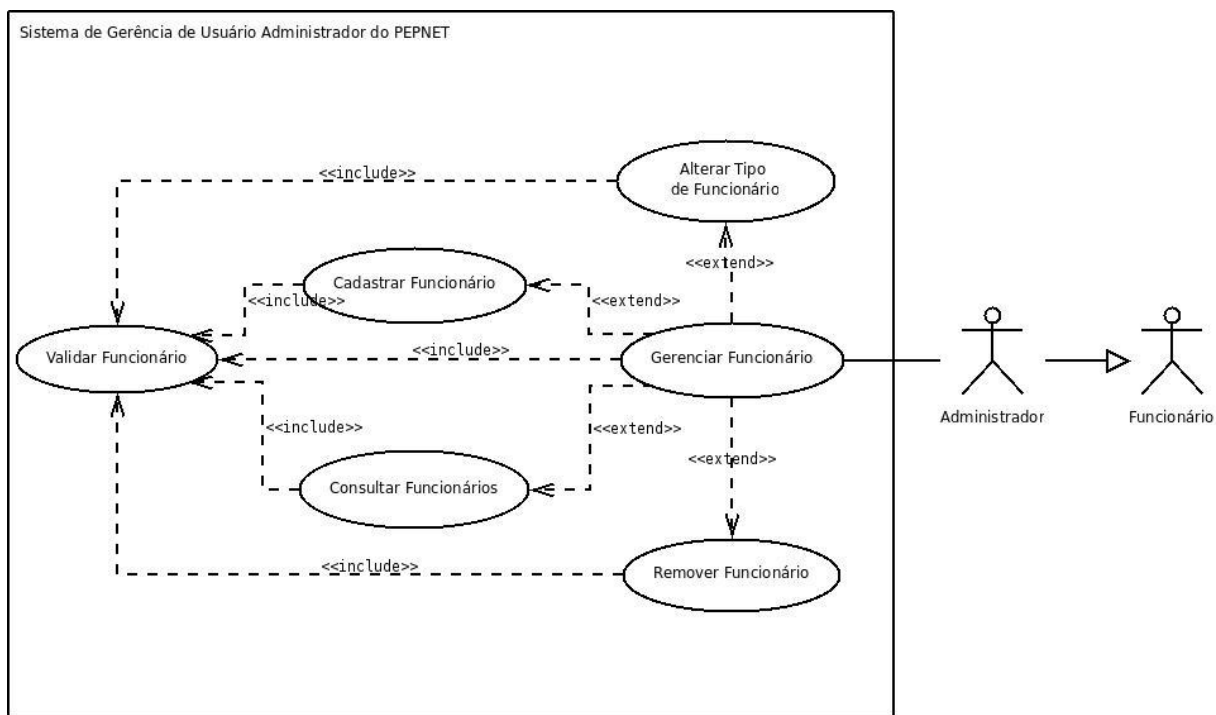


Figura 12. Diagrama dos Casos de Uso da Gerência de Usuário Administrador

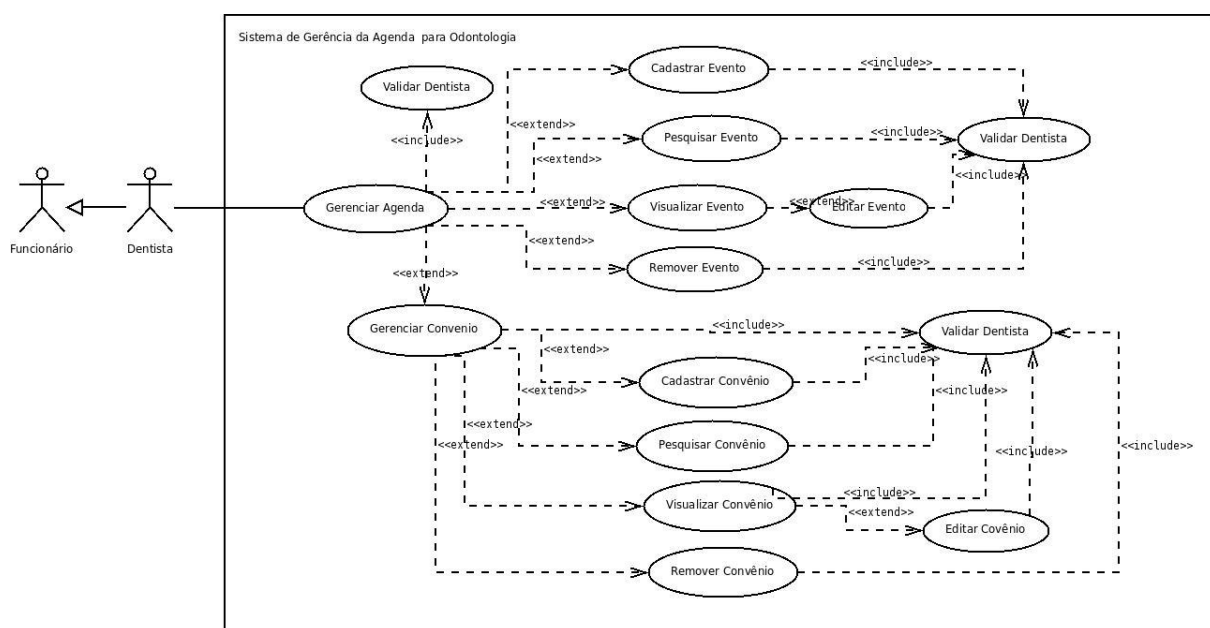


Figura 13. Diagrama dos Casos de Uso da Gerência da Agenda para Odontologia

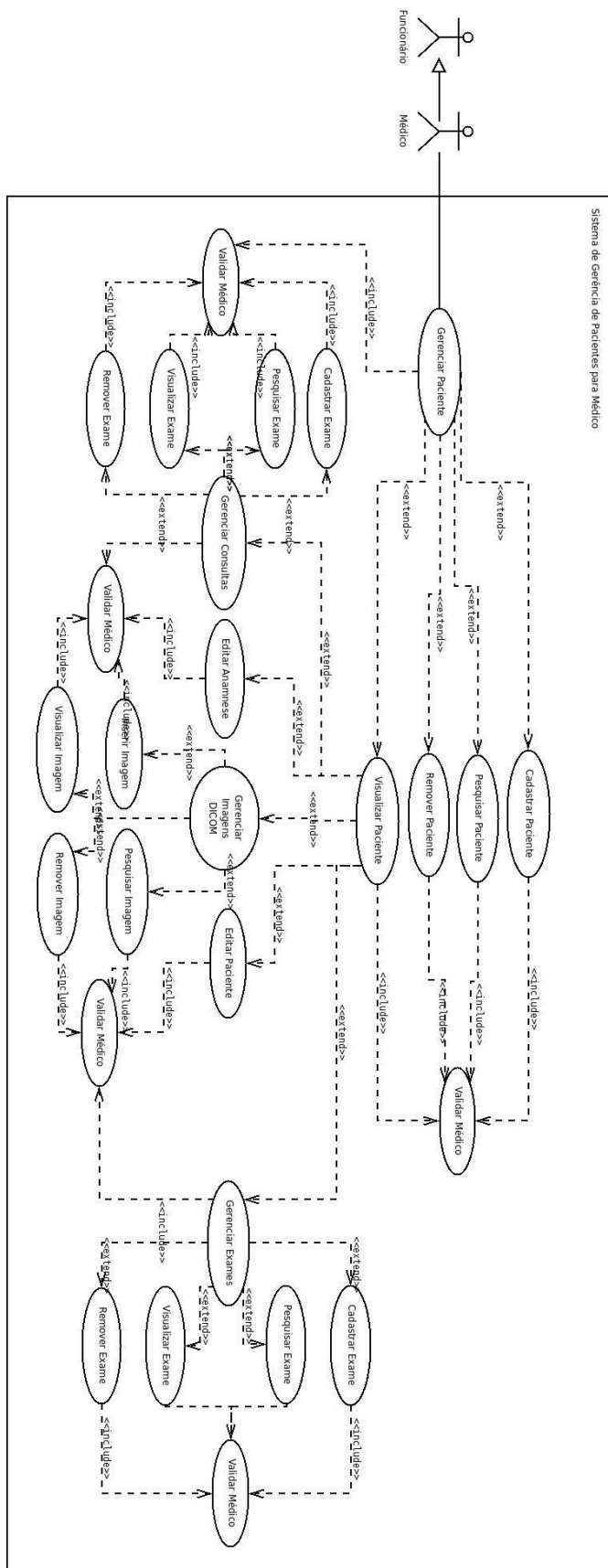
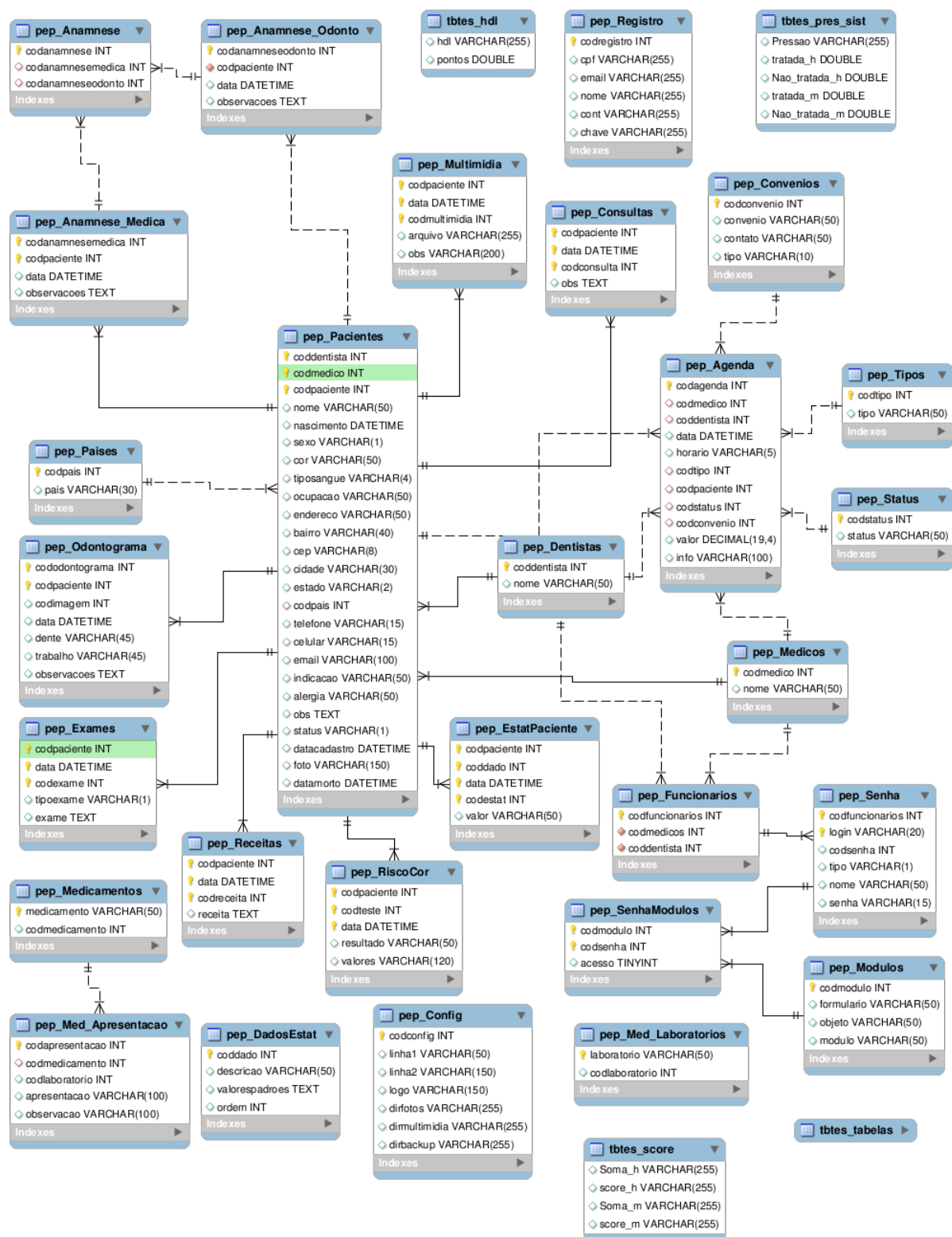


Figura 15. Diagrama dos Casos de Uso da Gerência de Pacientes para Médico

Diagrama do Modelo do Banco de Dados Relacional



Apêndice D

Testes

1. Teste de persistência

```
package br.com.pepnet.s2.domain;

import junit.framework.TestCase;
import javax.persistence.EntityTransaction;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

public class PersistenceBaseClass extends TestCase {
    private EntityManagerFactory emf;
    protected EntityManager entityMgr;
    protected EntityTransaction tx;
    public PersistenceBaseClass() {
        super();
        emf = Persistence.createEntityManagerFactory("slunart_s2pepnet");
    }
    protected void setUp() throws Exception {
        super.setUp();
        entityMgr = emf.createEntityManager();
        tx = entityMgr.getTransaction();
        tx.begin();
    }
    protected void tearDown() throws Exception {
        super.tearDown();
        tx.rollback();
    }
}
```

2. Teste do Domínio User

```
package br.com.pepnet.s2.domain;

public class UserTestCase extends PersistenceBaseClass {
    public void testCreateFind() throws Exception {
        User u = new User();
        u.setEmail("samuel@pepnet.com.br");
        u.setFirstName("Samuel");
        u.setLastName("Martins");
        u.setPassword("43218765");
        entityMgr.persist(u);
        entityMgr.flush();
        User test = entityMgr.find(User.class, u.getEmail());
        assertNotNull(test);
        assertEquals(u.getEmail(), test.getEmail());
        assertEquals(u.getFirstName(), test.getFirstName());
        assertEquals(u.getLastName(), test.getLastName());
        assertEquals(u.getPassword(), test.getPassword());
    }
}
```

3. Teste de Validação do Usuário através do *framework* Acegi

```
package br.com.pepnet.s2.util;

import junit.framework.TestCase;
import br.com.pepnet.s2.services.PermissionedUser;
import br.com.pepnet.s2.domain.User;
import com.opensymphony.xwork2.Action;
import com.opensymphony.xwork2.mock.MockActionInvocation;
import org.acegisecurity.context.SecurityContextHolder;
import org.acegisecurity.context.SecurityContextImpl;
import org.acegisecurity.Authentication;
import org.acegisecurity.GrantedAuthority;
import org.acegisecurity.providers.TestingAuthenticationToken;

public class AcegiInterceptorTestCase extends TestCase {

    public void testIntercept() throws Exception {
        TestAction action = new TestAction();
        AcegiInterceptor interceptor = new AcegiInterceptor();
        MockActionInvocation ai = new MockActionInvocation();
        ai.setAction(action);
        SecurityContextImpl sc = new SecurityContextImpl();
        Authentication auth =
            new TestingAuthenticationToken(
                new PermissionedUser(new User()), "password", new
GrantedAuthority[] {} );
        sc.setAuthentication( auth );
        SecurityContextHolder.setContext(sc);
        assertNull(action.getUser());
        interceptor.intercept(ai);
        assertNotNull(action.getUser());
        assertEquals(auth.getPrincipal(), action.getUser());
    }

    class TestAction {
        private PermissionedUser user;

        public PermissionedUser getUser() {
            return user;
        }

        @AcegiPrincipal
        public void setUser(PermissionedUser user) {
            this.user = user;
        }

        public String execute() {
            return Action.SUCCESS;
        }
    }
}
```