UNIVERSIDADE FEEVALE

MARCEL DE JESUS SILVA

SISTEMA DE GESTÃO PARA

CLÍNICAS MÉDICAS DE PEQUENO PORTE

Novo Hamburgo

2010

MARCEL DE JESUS SILVA

SISTEMA DE GESTÃO PARA

CLÍNICAS MÉDICAS DE PEQUENO PORTE

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do grau de Bacharel em Sistemas de Informação pela Universidade Feevale

Professor Orientador: Edvar Bergmann Araújo

Novo Hamburgo

2010

RESUMO

Os sistemas de gestão tornaram-se um dos principais componentes dos sistemas de informação das empresas. Levando em consideração a importância de as empresas terem um sistema de gestão da informação e o promissor segmento de serviços, este trabalho apresenta a proposta de criação de um sistema de gestão para clínicas médicas de pequeno porte. O projeto foi realizado seguindo a metodologia de desenvolvimento do processo unificado e a análise de sistemas para clínicas médicas já existentes no mercado. A utilização da linguagem de programação Java e seus *frameworks* JSF, EJB e JPA contribuem para uma arquitetura robusta. Na análise e projeto do sistema foram identificados os requisitos funcionais e os requisitos não funcionais. Foram desenvolvidos os casos de usos necessários, uma matriz de rastreabilidade para determinar quais casos de uso atendem os requisitos e, adicionalmente, foram feitos os diagramas de casos de uso e de classes. Por fim foi desenvolvido o protótipo das telas do sistema. Este software foi planejado com o intuito de oferecer uma redução de tempo no ciclo dos processos, informações mais rápidas sobre as transações da organização, melhoria na gerência financeira, bem como facilitar a conversão do conhecimento tácito sobre o processo em conhecimento explícito.

**Palavras-chave:** ERP. Sistema de gestão. Clínica médica. Sistema de gestão médica.

ABSTRACT

The management systems have become one of the most important components on company’s systems information. Taking into consideration the importance of companies having an information management system and the promising area of services, this paper proposes the creation of a management system for small medical clinics. The project was made following the development methodology of unified process and the analysis of other softwares for medical clinics already on the market. The use of Java development language and its frameworks JSF, EJB and JPA contribute for a robust structure. During the analysis and project of the system the functional and non functional requirements were identified. The necessary use cases were developed, as well as a tracking matrix in order to determine which use cases resolve the requirements. Additionally the diagrams of use cases and classes were made. Finally, the prototype of the system screens was developed. This software was planned with the following objectives: to offer a time reduction on processes cycle, faster information about the company transactions, improvement of financial management and make easier the conversion of tacit knowledge about the process on explicit knowledge.

**Keywords:** ERP. Management system. Medical clinic. Medical system management.

Lista de figuras

[Figura 2.1 - MDMed - Tela de Cadastro de Pacientes 22](#_Toc281252778)

[Figura 2.2 - MDMed - Tela de Agenda 23](#_Toc281252779)

[Figura 2.3 - MDMed - Tela de Ficha Médica 24](#_Toc281252780)

[Figura 2.4 - Gerclim - Tela de Cadastro de Pacientes 26](#_Toc281252781)

[Figura 2.5 - Gerclim - Tela de Agenda 27](#_Toc281252782)

[Figura 2.6 - Gerclim - Tela de Ficha Médica 28](#_Toc281252783)

[Figura 2.7 - ProDoctor - Tela de Cadastro de Pacientes 29](#_Toc281252784)

[Figura 2.8 - ProDoctor - Tela de Agenda 30](#_Toc281252785)

[Figura 2.9 - ProDoctor - Tela de Ficha Médica 31](#_Toc281252786)

[Figura 4.1 - Diagrama de Caso de Uso do Módulo de Segurança 50](#_Toc281252787)

[Figura 4.2 - Diagrama de Caso de Uso do Módulo de Cadastro – Visão da Secretária 50](#_Toc281252788)

[Figura 4.3 - Diagrama de Caso de Uso do Módulo de Cadastro – Visão do Médico 51](#_Toc281252789)

[Figura 4.4 - Diagrama de Caso de Uso do Módulo de Cadastro – Visão do Administrativo 51](#_Toc281252790)

[Figura 4.5 - Diagrama de Caso de Uso do Módulo de Agenda 52](#_Toc281252791)

[Figura 4.6 - Diagrama de Caso de Uso de Prontuário 52](#_Toc281252792)

[Figura 4.7 - Diagrama de Caso de Uso de Estoque 53](#_Toc281252793)

[Figura 4.8 - Diagrama de Caso de Uso Financeiro 54](#_Toc281252794)

[Figura 4.9 - Diagrama das Classes Comuns 54](#_Toc281252795)

[Figura 4.10 - Diagrama de Classes do Módulo de Segurança 55](#_Toc281252796)

[Figura 4.11 - Diagrama de Classes do Módulo de Cadastros 56](#_Toc281252797)

[Figura 4.12 - Diagrama de Classes do Módulo de Agenda 56](#_Toc281252798)

[Figura 4.13 - Diagrama de Classes do Módulo de Prontuário 57](#_Toc281252799)

[Figura 4.14 - Diagrama de Classes do Módulo de Estoque 58](#_Toc281252800)

[Figura 4.15 - Diagrama de Classes do Módulo de Financeiro 59](#_Toc281252801)

[Figura 5.1 - Protótipo da Tela de Cadastro de Pacientes 89](#_Toc281252802)

[Figura 5.2 - Protótipo da Tela de Cadastro de Clínicas 90](#_Toc281252803)

[Figura 5.3 - Protótipo da Tela de Agenda 91](#_Toc281252804)

[Figura 5.4 - Protótipo da Tela de Cadastro de Atendimentos 92](#_Toc281252805)

[Figura 5.5 - Protótipo da Tela de Fluxo de Caixa 93](#_Toc281252806)

[Figura 5.6 - Protótipo da Tela de Contas a Pagar e Contas a Receber 94](#_Toc281252807)

Lista de TABELAS

[Tabela 2.1 – Análise de Características 32](#_Toc281252808)

[Tabela 4.1 - Requisitos Funcionais 38](#_Toc281252809)

[Tabela 4.2 - Requisitos Não Funcionais 40](#_Toc281252810)

[Tabela 4.3 - Casos de Uso - Módulo de Segurança 41](#_Toc281252811)

[Tabela 4.4 - Casos de Uso - Módulo de Cadastros 42](#_Toc281252812)

[Tabela 4.5 - Casos de Uso - Módulo de Agenda 43](#_Toc281252813)

[Tabela 4.6 - Casos de Uso - Módulo de Prontuário 44](#_Toc281252814)

[Tabela 4.7 - Casos de Uso - Módulo de Estoque 45](#_Toc281252815)

[Tabela 4.8 - Casos de Uso - Módulo Financeiro 46](#_Toc281252816)

[Tabela 4.9 - Matriz de Rastreabilidade 47](#_Toc281252817)

Lista de ABREVIAturas e siglas

|  |  |
| --- | --- |
| *BI* | *Business Intelligence* |
| *CRM* | *Customer Relationship Management* |
| *DAO* | *Data Access Object* |
| *EJB* | *Enterprise Java Beans* |
| *ERP* | *Enterprise Resource Planning* |
| *JDBC* | *Java Data Base Connectivity* |
| *JPA* | *Java Persistence API* |
| *JSF* | *Java Server Faces* |
| *MVC* | *Model View Controller* |
| PIB | Produto Interno Bruto |
| RF | Requisito Funcional |
| RNF | Requisito Não Funcional |
| *SCM* | *Supply Chain Management* |
| SGM | Sistema de Gestão Médica |
| *SQL* | *Structured Query Language* |
| *UC* | *Use Case* |
| *UML* | *Unified Modeling Language* |
| *W3C* | *World Wide Web Consortium* |
|  |  |
|  |  |

SUMÁRIO

[INTRODUÇÃO 10](#_Toc281252818)

[1. PROCESSO UNIFICADO 13](#_Toc281252819)

[1.1. Introdução ao Processo Unificado 13](#_Toc281252820)

[1.1.1. Dirigido por casos de uso 13](#_Toc281252821)

[1.1.2. Centrado em arquitetura 14](#_Toc281252822)

[1.1.3. Iterativo e incremental 15](#_Toc281252823)

[1.2. As quatro fases do processo unificado 17](#_Toc281252824)

[1.3. Os nove ciclos do processo unificado 18](#_Toc281252825)

[2. SISTEMAS SIMILARES 20](#_Toc281252826)

[2.1. O sistema MDMed 21](#_Toc281252827)

[2.2. O sistema Gerclim 24](#_Toc281252828)

[2.3. O sistema ProDoctor 28](#_Toc281252829)

[2.4. Análise de características dos sistemas similares 31](#_Toc281252830)

[3. TECNOLOGIAS 33](#_Toc281252831)

[3.1. Linguagem Java 33](#_Toc281252832)

[3.2. JSF – *Java Server Faces* 34](#_Toc281252833)

[3.3. EJB – *Enterprise Java Beans* 34](#_Toc281252834)

[3.4. JPA – *Java Persistence API* 35](#_Toc281252835)

[3.5. JDBC e SQL 35](#_Toc281252836)

[3.6. Padrões de projetos: MVC e DAO 36](#_Toc281252837)

[4. ANÁLISE E PROJETO 38](#_Toc281252838)

[4.1. Requisitos Funcionais 38](#_Toc281252839)

[4.2. Requisitos não funcionais 40](#_Toc281252840)

[4.3. Casos de Uso 41](#_Toc281252841)

[4.3.1. Módulo de Segurança 41](#_Toc281252842)

[4.3.2. Módulo de Cadastros 42](#_Toc281252843)

[4.3.3. Módulo de Agenda 43](#_Toc281252844)

[4.3.4. Módulo de Prontuário 44](#_Toc281252845)

[4.3.5. Módulo de Estoque 45](#_Toc281252846)

[4.3.6. Módulo Financeiro 46](#_Toc281252847)

[4.4. Matriz de Rastreabilidade 47](#_Toc281252848)

[4.5. Diagrama de Casos de Uso 48](#_Toc281252849)

[4.5.1. Segurança 49](#_Toc281252850)

[4.5.2. Cadastros 50](#_Toc281252851)

[4.5.3. Agenda 52](#_Toc281252852)

[4.5.4. Prontuário 52](#_Toc281252853)

[4.5.5. Estoque 53](#_Toc281252854)

[4.5.6. Financeiro 53](#_Toc281252855)

[4.6. Diagrama de Classes 54](#_Toc281252856)

[4.6.1. Segurança 55](#_Toc281252857)

[4.6.2. Cadastros 55](#_Toc281252858)

[4.6.3. Agenda 56](#_Toc281252859)

[4.6.4. Prontuário 57](#_Toc281252860)

[4.6.5. Estoque 57](#_Toc281252861)

[4.6.6. Financeiro 58](#_Toc281252862)

[4.7. Detalhamento dos casos de uso 59](#_Toc281252863)

[4.7.1. Cadastros 59](#_Toc281252864)

[4.7.2. Agenda 68](#_Toc281252865)

[4.7.3. Prontuário 72](#_Toc281252866)

[4.7.4. Estoque 77](#_Toc281252867)

[4.7.5. Financeiro 81](#_Toc281252868)

[5. PROTÓTIPO DO SISTEMA 89](#_Toc281252869)

[5.1. Cadastro de Pacientes 89](#_Toc281252870)

[5.2. Cadastro de Clínicas 90](#_Toc281252871)

[5.3. Agenda 91](#_Toc281252872)

[5.4. Cadastro de Atendimento 91](#_Toc281252873)

[5.5. Fluxo de Caixa 92](#_Toc281252874)

[5.6. Contas a Pagar e Contas a Receber 93](#_Toc281252875)

[CONCLUSÃO 95](#_Toc281252876)

[REFERÊNCIAS 97](#_Toc281252877)

INTRODUÇÃO

Para garantir não apenas vantagens no mercado, mas sua sobrevivência nele, as empresas precisam estar atentas a todos os fatores ambientais, organizacionais e tecnológicos que as cercam. Com o acesso rápido e fácil às informações, os consumidores estão cada vez mais exigentes e o mercado está em constante mudança. As organizações precisam integrar a grande quantidade de informações que impactam em seu negócio para poder acessá-la de forma rápida e tomar decisões assertivas no menor tempo possível. Souza e Saccol (2008) acreditam que os sistemas de informações são recursos cada vez mais críticos para que as empresas garantam a sobrevivência de seus empreendimentos.

Os sistemas de ERP (*Enterprise Resource Planning*) tornaram-se um dos principais componentes dos sistemas de informação de empresas. Isto porque permitem a integração de dados de todos os processos de negócios da empresa, fornecendo uma visão ampla da empresa aos gestores e permitindo que tenham informações precisas e atualizadas para basear a tomada de decisões.

Entre os sistemas de informação disponíveis hoje no mercado, os sistemas integrados de gestão, conhecidos como *Enterprise Resource Planning* (ERP), vêm recebendo continuamente a atenção de profissionais e acadêmicos da área de Administração. A proposta destes sistemas é a de unir e disponibilizar informações para a organização toda. (SOUZA e SACCOL, 2008 p. 173)

Davenport (2002) defende que alguns dos benefícios alcançados pelas empresas através do uso de sistemas de gestão empresarial são:

1 - Redução de tempo do ciclo dos processos;

2 - Informações mais rápidas sobre as transações da organização;

3 - Melhoria na gerência financeira;

4 - Conversão do conhecimento tácito sobre o processo em conhecimento explícito.

Padilha e Marins (2005 apud DAVENPORT 1998) apresentam as funcionalidades dos sistemas ERP separando-as em funções internas (*back-office*), composta por recursos humanos, manufatura e finanças, e funções externas (*front-office*), composta por vendas e serviços, além da tecnologia e do chamado Gerenciamento da Cadeia de Suprimentos- SCM (*Supply Chain Management*).

Padilha e Marins (2005) destacam que, dentre as principais tendências e novidades incorporadas pelos principais fornecedores de ERP estão:

* **Foco nas Empresas de Pequeno e Médio Porte (*Small/Middle Market*)**: atualmente, especialmente no Brasil, o principal alvo das produtoras de sistemas ERP é o chamado "*small/middle market*", composto por empresas de pequeno e médio porte;
* **Internet**: uma grande tendência entre os fornecedores de ERP é a gradual incorporação de módulos que possam ser operacionalizados via Internet, permitindo a prática do comércio e outras práticas empresariais, por meio eletrônico (*e-business*);
* ***Business Intelligence (BI)***: é um termo genérico para aplicações, plataformas, ferramentas e tecnologias que suportam o processo de exploração de dados de negócio e análise de suas correlações e tendências. Aplicações de BI conferem às empresas meios para coletar e preparar dados com o objetivo de facilitar a geração de relatórios, análises e tomada de decisão;
* ***Supply Chain Management -* SCM**: ou gerenciamento da cadeia de suprimentos, é o nome do recurso que permite a integração de uma empresa com as demais organizações envolvidas no processo produtivo (clientes e fornecedores), buscando otimizar o funcionamento como um todo, com reduções de custos e ganhos de produtividade e qualidade;
* **CRM (*Customer Relashionship Management*)**: ou gerenciamento das relações com o cliente, está assumindo um papel muito importante nos departamentos de marketing, que também utilizam a expressão marketing de relacionamento para os conceitos apoiados por esta nova ferramenta.

Dentre os setores que estão implementando sistemas de gestão para ganhar competitividade em seus mercados, está o setor de serviços. Os negócios com serviços estão crescendo no mundo todo, sendo que, em muitos países, já há uma predominância deste setor nos resultados da economia. Este crescimento teve início nos países desenvolvidos e “em 1996 o setor de serviços representava 79% de todos os empregos e pelo menos 76% do PIB dos Estados Unidos” (VALARIE e BITNER, 2008, p. 31).

As empresas que oferecem serviços médicos também estão buscando através de sua informatização melhorar seus processos e oferecer serviços de maior qualidade a seus clientes.

Levando em consideração a importância de as empresas terem um sistema de gestão da informação e o promissor segmento de serviços, este trabalho terá como objetivo estruturar o desenvolvimento de um sistema de gestão para clínicas médicas de pequeno porte. Objetiva-se responder assim ao seguinte problema: como desenvolver um sistema de gestão empresarial para clínicas médicas de pequeno porte permitindo, assim, que reduza significativamente a quantidade de papéis arquivados e manuseados e facilite o acesso às informações necessárias para a eficaz gestão desta clínica.

Este trabalho propõe contribuir com a informatização das clínicas da área de saúde da região do Vale do Sinos. A proposta é oferecer para estas empresas um sistema que atenda às suas necessidades específicas, não apenas de controle, mas também que gere informações com base nestes dados permitindo a tomada de decisões estratégicas por parte dos gestores.

Resumindo, o objetivo principal é projetar um sistema de gestão para clínicas médicas de pequeno porte, disponibilizando de forma simples e direta o acesso às informações sobre o gerenciamento de cadastros, agendamentos, prontuários eletrônicos, custos, estoque e relatórios, facilitando a gestão da clínica, a rapidez no atendimento de seus clientes e maior segurança das informações.

Outros objetivos que podem ser destacados são as comparações deste sistema com sistemas similares, a descrição das necessidades de clínicas médicas e a análise e projeto deste sistema de forma detalhada.

O trabalho está organizado em cinco capítulos. O primeiro capítulo apresenta o processo unificado e o porquê de sua utilização. O segundo descreve os sistemas similares ao proposto, destacando suas características. O terceiro capítulo apresenta as tecnologias utilizadas, incluindo a linguagem de programação, bem como seus frameworks e os padrões de projetos aplicados. O quarto capítulo aborda a análise de requisitos do sistema, descreve os requisitos funcionais e não funcionais, os casos de uso e seus diagramas divididos em módulos. O quinto e último capítulo demonstra o protótipo do sistema através de suas principais telas.

1. PROCESSO UNIFICADO

A engenharia de software pode ser definida como um conjunto de disciplinas que incluem a especificação, o desenvolvimento, o gerenciamento e a evolução dos sistemas de software. Seu objetivo é um maior rigor no desenvolvimento de softwares. (AUDY e PRIKLADNICKI, 2007, p.10).

Pressman (2004) define a Engenharia de software como dividida em quatro camadas. A camada base é o foco na qualidade, seguida de processos, métodos e ferramentas. As ferramentas proporcionam apoio automatizado ao método que, por sua vez, consiste em “como fazer”, ou seja, quais são os passos que devem ser seguidos para a construção de um software de alta qualidade. O processo é a camada mais importante da engenharia de software, pois ela faz a intermediação entre as ferramentas e os métodos, além de possibilitar um desenvolvimento racional do software.

Um processo de software é um conjunto de atividades e resultados associados que geram um produto de software. Um processo define a seqüência em que os métodos serão aplicados, como os produtos serão entregues, os controles que ajudam a assegurar a qualidade e a coordenar as mudanças, e os marcos de referencia que possibilitam aos gerentes de software avaliar o progresso do desenvolvimento. (AUDY e PRIKLADNICKI, 2007, p. 11)

Neste capítulo será abordado um detalhamento do processo unificado, sendo esta a metodologia escolhida para o desenvolvimento do software em questão no presente trabalho.

## Introdução ao Processo Unificado

Processo unificado significa criar um sistema de software através de um conjunto de requisitos do cliente utilizando um conjunto de atividades. Processo unificado também pode ser definido como “uma estrutura genérica de processo que pode ser customizado adicionando-se ou removendo-se atividades com base nas necessidades específicas e nos recursos disponíveis para um projeto” (SCOTT, 2003, p.19). O processo unificado é caracterizado por ser dirigido por casos de uso, centrado em arquitetura e interativo e incremental. A seguir estas três características serão abordadas de maneira mais detalhada.

### Dirigido por casos de uso

Casos de uso são seqüências de ações, onde o próprio sistema ou atores (pessoas) as executam, gerando resultados de valor para um ou mais atores. Um dos principais atributos do processo unificado é a utilização de casos de uso no seu desenvolvimento. “A expressão *dirigido por casos de uso* refere-se ao fato de se utilizar os casos de uso para dirigir todo o trabalho de desenvolvimento, desde a captação inicial e negociação de requisitos até a aceitação do código” (SCOTT, 2003, p.21).

Segundo o autor, casos de uso são importantes para encontrar requisitos, elaborar a análise, projeto e implementação, pelas seguintes razões:

* São demonstrados sob a ótica dos usuários;
* São expressos de maneira simples, são intuitivos e estão na linguagem do cliente;
* Melhoram consideravelmente o entendimento dos requisitos reais do sistema;
* Atingem um alto grau de rastreamento de requisitos que resultam do desenvolvimento posterior;
* Simplificam a decomposição dos requisitos que permitem alocar trabalho a subequipes o que ajuda na gerência do projeto.

### Centrado em arquitetura

No contexto da computação, arquitetura possui diferentes significados, dependendo de quem a define. Uma definição proposta é a seguinte:

Arquitetura é a organização fundamental do sistema como um todo. Entre os aspectos de uma arquitetura, estão incluídos elementos estáticos, elementos dinâmicos o modo como estes elementos trabalham juntos e o estilo arquitetônico total que guia a organização do sistema. A arquitetura também se refere a questões como desempenho, escalabilidade, reuso e restrições econômicas e tecnológicas. (SCOTT, 2003, p.21).

Os autores Bass, Clements, Kazman e Wesley (1998 apud Scott, 2003), falam que para o processo unificado, uma das principais preocupações da equipe de projeto deve ser a arquitetura do sistema, já que ela é o alicerce fundamental sobre o qual ele se erguerá. Além disso, assim como os casos de uso, a arquitetura deve orientar a exploração de todos os aspectos do sistema.

Para Scott (2003), existe algumas razões-chave da importância da arquitetura para o processo unificado, seguem elas:

* **Entendendo a visão global -** por mais queas ferramentas de desenvolvimento estejam evoluindo, não há uma perspectiva próxima de estas ferramentas alcançarem de forma satisfatória os níveis de complexidade gerados com sistemas distribuídos. Em conseqüência disto se torna difícil para quase todas as pessoas entender o sistema de uma maneira mais significativa. A definição da arquitetura objetiva acima de tudo entender o sistema em construção. Sendo assim a modelagem e legibilidade dos diagramas UML (*Unified Modeling Language*) junto com os textos de apoio serão de vital importância para melhorar o entendimento da “visão global” do sistema;
* **Organizando o esforço de desenvolvimento -** usar padrões arquitetônicos ajuda a dar forma ao esforço de desenvolvimento em vários níveis. Como exemplos de padrões arquitetônicos bem conhecidos, pode-se citar “Cliente/Servidor” e “*Model-View-Controller”* (MVC) que também é chamado de “três camadas”. É possível aumentar em muito as chances de comunicação das subequipes e agregar valor ao esforço destas ao utilizar efetivamente este aspecto da arquitetura;
* **Facilitando as possibilidades de reuso -** a idéia de componentes reutilizáveis com um mínimo relativo de customização é um dos princípios fundamentais do desenvolvimento baseado em componentes. Uma arquitetura bem construída disponibiliza uma estrutura sólida na qual os componentes podem trabalhar uns com os outros de forma simples e elegante, facilitando para as equipes de construção o possível reuso de componentes. “A linha de base é que quanto menos tempo for necessário à construção de novos componentes, mais tempo haverá para entender os problemas dos clientes e modelar as soluções”;
* **Facilitando a evolução do sistema -** com a constante e rápida evolução da internet, os modelos de negócio costumam mudar com maior freqüência do que antes. Ter uma arquitetura sólida é um ótimo apoio para executar estas mudanças, e se construída de modo que as mudanças em uma determinada parte do sistema não afetem as demais, também contribuirá para a evolução efetiva e eficiente do sistema;
* **Dirigindo os casos de uso -** se por um lado os casos de uso direcionam a arquitetura de um sistema, por outro a arquitetura guia a seleção e utilização de casos de uso. As decisões dos arquitetos possuem uma forte influência na escolha dos casos de uso que agregarão valor a arquitetura, o que ajuda a dar forma ao conteúdo daqueles casos de uso e à natureza do trabalho envolvido no desenvolvimento deles.

### Iterativo e incremental

Scott (2003) descreve que “iteração é um miniprojeto que resulta em uma versão do sistema. Supõe-se que essa versão ofereça uma melhora incremental sobre a anterior.” Ele ainda fala de como iterações e incrementos se encaixam no contexto de processo completo. Os itens a seguir demonstram as vantagens do desenvolvimento iterativo e incremental:

* **Progresso lógico para uma arquitetura robusta -** o processo demonstra a forma de a equipe trabalhar, propõe uma base sólida para a arquitetura durante as iterações iniciais. Após é proposta uma visão completa da arquitetura, que influencia a maior parte do desenvolvimento realizado em dada iteração. Usar o modo iterativo e incremental permite ainda no início do processo, fazer as mudanças necessárias a um custo consideravelmente menor;
* **Lidando com mudanças contínuas nos requisitos -** processos que se baseiam no desenvolvimento em cascata enfrentam o que hoje parece ser um problema inevitável, os requisitos costumam ser instáveis e os clientes não conseguem visualizar o sistema. O processo unificado tem o objetivo de quebrar o desenvolvimento do sistema em unidades, nas quais sua construção é uma versão funcional de uma parte do sistema completo. Utilizar um conjunto limitado de casos de uso e apresentar protótipos favorece a negociação de requisitos de modo progressivo, reduzindo assim o risco de tentar especificar todos os requisitos no início do projeto;
* **Maior Flexibilidade para mudar o plano -** sendo a iteração um mini projeto, os riscos associados com o projeto devem ser tratados assim que a equipe cria uma iteração do sistema. No decorrer das iterações é possível realizar os ajustes necessários em uma pequena escala e propagá-los por todo o projeto. “O objetivo é isolar os problemas dentro das iterações e lidar com eles em uma escala relativamente pequena, em vez de permitir que se alastrem”;
* **Integração contínua -** cada incremento cria uma funcionalidade aperfeiçoada e uma combinação de novas características para o sistema. Isto permite que seja medido o progresso do projeto em relação aos objetivos específicos. Com a utilização de componentes integrados é possível separar os problemas que podem ser provocados no sistema e resolvê-los de maneira a não afetar a integridade do mesmo, possibilitando decisões drásticas como recomeçar um incremento sobre novos alicerces;
* **Entendimento precoce -** como cada atividade da equipe tem uma definição simples, “o processo é projetado para permitir o aprendizado e treinamento contínuos”. Errar e experimentar neste processo não causa um grande impacto, pois os erros serão isolados;

À medida que o trabalho prossegue, é possível nivelar o entendimento sobre o que se está tentando construir e os riscos associados, o que desenvolve uma capacidade de trabalho, que, por sua vez, permite melhorias contínuas no modo de lidar com as tarefas (SCOTT, 2003, p.25).

* **Foco contínuo sobre riscos -** a mais importante vantagem do desenvolvimento iterativo e incremental talvez seja a possibilidade de no início do projeto concentrar os esforços nos riscos mais críticos. A equipe organiza as iterações de acordo com as análises de riscos de uma forma contínua, tendo como objetivo minimizá-los durante cada iteração, fazendo com que cada vez as iterações possuam menos riscos e que estes tenham menor importância do que os anteriores.

## As quatro fases do processo unificado

Booch, Rumbaugh e Jacobson (2000) afirmam que o processo unificado é composto por quatro fases, que são detalhadas a seguir. Segundo os autores, “uma fase é o período de tempo entre dois importantes marcos de progresso do processo em que um conjunto de objetivos bem definidos é alcançado”. As principais fases do Processo Unificado são:

1. **Concepção -** nesta fase o caso de negócio é estabelecido e o escopo delimitado para o projeto. O caso de negócio deve incluir critérios de sucesso, avaliação de riscos, definição de recursos necessários e um plano definindo os principais marcos de progresso. Ao fim desta fase se define a viabilidade de se prosseguir com o projeto em plena escala;
2. **Elaboração -** são definidos o plano de projeto e a arquitetura. Os objetivos desta fase são “análise do domínio de problema, o estabelecimento da fundação de uma arquitetura sólida, o desenvolvimento do plano do projeto e a eliminação dos elementos de mais alto risco do projeto”. (BOOCH, RUMBAUGH e JACOBSON: 2000, p.445). Como é necessária uma compreensão de todo o sistema para definir a arquitetura, se faz necessário o detalhamento da maioria de seus requisitos. Ao fim, se decide se deve-se prosseguir com a construção;
3. **Construção -** consiste no desenvolvimento do sistema. Nesta fase é desenvolvido um sistema completo, passando pela descrição dos requisitos restantes e de critérios de aceitação, pelo desenvolvimento em si, pela implantação e, finalmente, pelos testes do software. Ao fim se analisa se o software, o ambiente e os usuários estão prontos para se tornarem operacionais;
4. **Transição -** quando o sistema é entregue aos usuários finais. Após o início da utilização do software por seus usuários, podem surgir necessidades de desenvolvimentos adicionais. Por isso, segundo os autores, esta fase normalmente se inicia com uma versão beta do sistema que, posteriormente, é substituída pelo sistema de produção. No final desta fase se decide se os objetivos de ciclo de vida do projeto foram alcançados e se determina se outro ciclo de desenvolvimento deve ser iniciado.

A passagem pelas quatro principais fases é chamada um ciclo de desenvolvimento e resulta na geração de um software. O primeiro passo das quatro fases é chamado o ciclo inicial de desenvolvimento. A menos que a vida do produto seja interrompida, um produto existente evoluirá para sua próxima geração pela repetição da mesma seqüência de fases de concepção, elaboração, construção e transição. Isso é a evolução do sistema, de modo que os ciclos de desenvolvimento posteriores ao ciclo inicial são seus ciclos de evolução. (BOOCH, RUMBAUGH e JACOBSON: 2000, p.445)

## Os nove ciclos do processo unificado

De acordo com Audy e Prikladnicki (2007), existem nove *workflows* no processo unificado, que representam uma divisão das atividades em grupos lógicos e estão detalhados a seguir:

1. **Modelagem de negócio -** tem por objetivo entender as estruturas, a dinâmica e os problemas da organização. Assim é possível definir quais necessidades do cliente o sistema deverá suprir. Alguns documentos associados são os modelos de casos de uso, modelos de objeto do negócio, glossário de negócio e documento de arquitetura de negócios;
2. **Requisitos -** definição de o que o sistema deve fazer e por quê. Assim os clientes permitem que os desenvolvedores tenham um melhor entendimento dos requisitos do sistema. São definidos ainda o escopo do sistema e as bases para estimativas de tempo e custo de desenvolvimento;
3. **Análise e projeto -** tem como objetivo descrever a especificação de como implementar o projeto com base nos seus requisitos. Adicionalmente, visa definir uma arquitetura robusta para o sistema e adaptar o projeto para o ambiente de implementação;
4. **Implementação -** é o desenvolvimento do software no ambiente de implementação definido. É quando o código é organizado em forma de subsistemas de implementação organizados em camadas; as classes e objetos em termos de componentes são implementadas; os componentes são testados como unidades e os resultados produzidos são integrados em um sistema executável;
5. **Teste -** o seu objetivo é garantir que os resultados obtidos são aqueles solicitados pelo cliente. É necessário verificar neste *workflow* as interações e a integração entre os componentes, além de identificar possíveis defeitos. Ao fim deste ciclo deve ser certificado que o sistema está pronto para uso;
6. **Implantação –** é quando o software é disponibilizado para os usuários. São realizadas atividades de instalação, distribuição, treinamento de usuários e teste em seu ambiente final de operação;
7. **Gerência de configuração e alteração -** visa o controle das modificações e a manutenção da integridade dos artefatos do projeto. São realizadas atividades de identificação dos itens de configuração, restrição e auditoria das alterações;
8. **Gerência de projeto -** tem como objetivo descrever várias estratégias para o trabalho com um processo iterativo. Alguns artefatos envolvidos são o plano de desenvolvimento de software, caso de negócio, plano de iteração, registro de revisões e plano de garantia de qualidade;
9. **Ambiente -** tem como objetivo fornecer processos e ferramentas para aquisição, instalação e configuração de ferramentas, configuração e melhoria de processo, assim como serviços técnicos de suporte.

Com base no que foi exposto neste capítulo é possível afirmar que a utilização de uma metodologia como o Processo Unificado é de suma importância para o bom desempenho de qualquer projeto de software. Utilizar o Processo Unificado tem como meta assegurar que um sistema possua qualidade no seu desenvolvimento, que supra as necessidades de seus usuários e tenha uma agenda previsível (AUDY e PRIKLADNICKI: 2007, p.22).

1. SISTEMAS SIMILARES

De acordo com Pfleeger (2004), as pessoas devem ser capazes de avaliar os produtos e o modo como os produzem. Utilizando técnicas de avaliação consegue-se medir os principais aspectos de produtos, processos e recursos, e utilizam-se essas informações para determinar se estão satisfazendo os objetivos de produtividade, desempenho, qualidade e outros atributos desejáveis. É possível classificar uma técnica de avaliação em uma das quatro categorias a seguir:

* Análise de características;
* Pesquisa de opinião;
* Estudo de caso;
* Experimento formal.

No presente trabalho foram realizadas análises em três sistemas similares. Esta análise teve o intuito de verificar quais as principais características destes sistemas. Os sistemas analisados seguiram a técnica de avaliação de análise de características. Segundo Pfleeger (2004), esta técnica é utilizada para atribuir um valor e classificar os atributos de vários produtos, podendo assim escolher qual ferramenta comprar ou que método utilizar.

O autor define alguns passos para realizar a análise de características da ferramenta. Primeiramente relacionar os cinco atributos principais que se deseja que a ferramenta possua, em seguida, identifica-se três ferramentas (sistemas) possíveis que, aparentemente atendam aos objetivos e atribui-se valores aos critérios de cada uma delas, definido 1 (não satisfaz) até 5 (satisfaz completamente). Após são examinadas as pontuações, criando uma pontuação total com base na importância de cada critério, multiplica-se então a importância com o valor atribuído a cada critério e somam-se os valores obtidos.

Os cinco principais atributos definidos neste trabalho foram identificados através de entrevistas com o gestor e médico da clínica de quiropraxia Dr. Everett Langhans e são:

* Boa interface com o usuário / Simplicidade;
* Gestão Financeira;
* Serviços de implantação, treinamentos e suporte;
* Integração entre os módulos;
* Plataforma Web.

Os três sistemas identificados como similares ao que está sendo proposto são:

* MDMed - http://www.mdmed.com.br;
* Gerclim Web - http://www.gerclim.com.br;
* ProDoctor - http://www.prodoctor.net;

Foi feito um levantamento de informações de cada um destes softwares através de pesquisa de dados secundários, no site das empresas desenvolvedoras e em fóruns de usuários dos mesmos. Os resultados obtidos neste levantamento estão detalhados a seguir.

## O sistema MDMed

O primeiro software similar analisado foi o MDMed, desenvolvido por uma empresa localizada no estado de Santa Catarina e que possui distribuidores em outros estados do Brasil. O software começou a ser comercializado em 1998 e por isso a empresa parece possuir boa experiência de mercado e seu aplicativo suporta diversas necessidades dos clientes em potencial.

A interface do sistema MDMed possui várias imagens e opções de ajuda e busca, facilitando assim o entendimento do usuário quanto às funcionalidades do sistema. No entanto, não é um sistema muito simples de se utilizar, já que não é customizável segundo as necessidades de cada cliente e, por isso, acaba tendo muitas opções de funcionalidades. Dentre estas funcionalidades, diversas acabam não sendo utilizadas por alguns clientes, o que torna a utilização um pouco confusa. Além disto, algumas telas e relatórios possuem muitas informações, dificultando a busca pela informação exata que o usuário necessita.

Outro ponto muito importante a salientar, é que o sistema MDMed não possui uma gestão financeira satisfatória, segundo análise feita junto a clínica de quiropraxia. Existe uma parte do sistema que atende o faturamento da clínica que o utiliza, mas tal faturamento é muito voltado aos convênios médicos e acaba pecando na gestão eficiente da clínica.

Na sua região de atuação a empresa oferece serviços de implementação, treinamento de usuários e suporte. Cada um destes serviços é vendido separadamente à licença de uso e os usuários realmente podem optar por não comprar tais serviços, já que a empresa oferece junto com as licenças, manuais e tutoriais explicando os passos para que os próprios usuários façam a instalação e se eduquem quanto à utilização de cada um dos módulos.

Os módulos do sistema parecem estar razoavelmente integrados entre si. Não há opção de rodar este sistema através de uma plataforma web, é necessário que o usuário instale e rode o programa em cada um dos computadores. É possível, no entanto, fazer algumas transferências de dados via internet.

A figura 2.1 demonstra a tela de cadastro de pacientes. O intuito de expor esta tela é demonstrar a interface amigável e também a variedade de campos disponíveis.

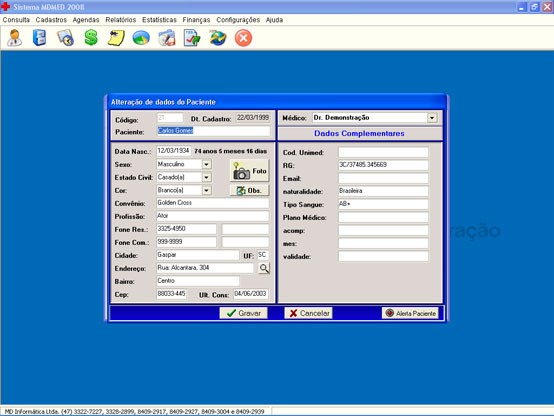


Figura 2.1 - MDMed - Tela de Cadastro de Pacientes

Fonte: http://www.mdmed.com.br

Na figura 2.2 é possível visualizar a tela de agenda, que está sendo exposta com o objetivo de demonstrar a falta de simplicidade do sistema. Também é visível a impossibilidade de visualizar mais de um médico ao mesmo tempo na agenda.

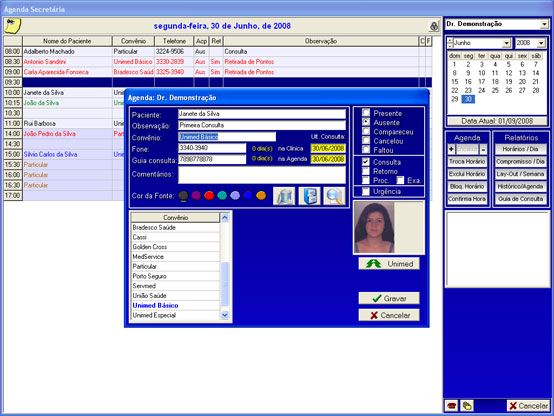


Figura 2.2 - MDMed - Tela de Agenda

Fonte: http://www.mdmed.com.br

A partir da figura 2.3, identifica-se a falta de padronização no armazenamento dos dados da ficha médica do paciente, o que pode ocasionar em redundância de informações, bem como, na dificuldade de pesquisá-las.

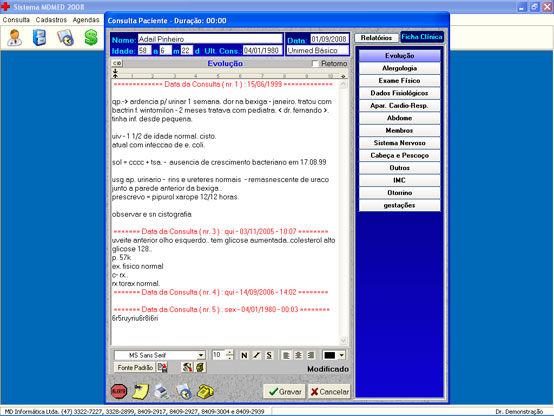


Figura 2.3 - MDMed - Tela de Ficha Médica

Fonte: http://www.mdmed.com.br

A partir das imagens do sistema MDMed e da análise do mesmo pode-se visualizar na seção 2.4 a técnica de análise de características comparando o MDMed com os demais sistemas avaliados.

## O sistema Gerclim

O segundo sistema similar verificado foi o Gerclim. Este sistema foi desenvolvido pela empresa MedSystem, de Minas Gerais que foi introduzido no mercado em 1983. O sistema foi desenvolvido utilizando a linguagem de programação C++ e usa banco de dados Microsoft SQL Server.

Segundo o médico da clínica de quiropraxia, Dr. Everett Langhans, o Gerclim tem uma interface agradável, porém não é muito auto-explicativa. A empresa não oferece serviço de adaptação do software segundo as solicitações de cada cliente. Isso diminui a simplicidade de uso do sistema, uma vez que existem funções que estão disponíveis mas não são usadas pelo cliente. No entanto, algo que ameniza um pouco este aspecto é que o menu do sistema ajusta as opções de funções da esquerda para a direita segundo a freqüência de demanda de cada uma destas funções.

Além da falta de adaptação do sistema segundo as necessidades do cliente, outro fator onde o sistema peca é na gestão financeira da clínica. A utilização de um controle financeiro que não é específico do sistema, acaba não atendendo as necessidades específicas das clínicas que o utilizam e ocasiona uma má gestão ou a não utilização da ferramenta.

A implantação do software deve ser feita pelo próprio usuário, assim como o *upgrade* para novas versões. A empresa oferece serviços de treinamento de usuários que pode ser feito remota ou presencialmente. Através do site ou de um número de telefone, é oferecido suporte ao sistema durante 24 horas por dia / 7 dias por semana. A MedSystem possui um banco de sugestões em seu web site que analisa para desenvolver novas versões do sistema, que são disponibilizadas aos clientes que estão sob manutenção. O Gerclim oferece ainda o serviço de aluguel de *Data Center*, para que os usuários possam armazenar *backups*, fazer backups remotos e banco de dados de clientes.

As funções do sistema são bem integradas, sendo que uma atualização implica na atualização automática das funções a ela relacionadas. Assim como o MDMed, o Gerclim roda localmente e possui conectividade web, no entanto o usuário não tem a opção de rodar este sistema pela internet.

Assim como acontece com o sistema MDMed, o Gerclim possui uma interface amigável, a tela de cadastro de pacientes demonstra esta situação (figura 2.4).

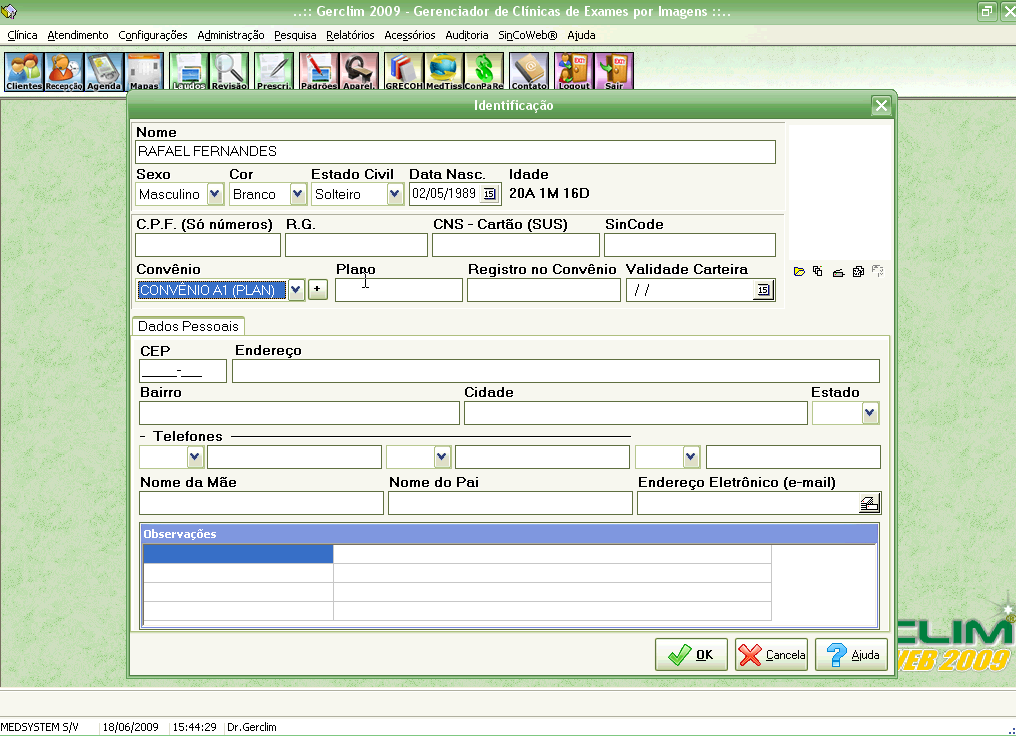


Figura 2.4 - Gerclim - Tela de Cadastro de Pacientes

Fonte: http://www.gerclim.com.br

A figura 2.5 demonstra a tela de agenda do sistema Gerclim. É possível identificar o mesmo problema que ocorre no sistema MDMed, a impossibilidade de visualizar a agenda de dois médicos de forma simultânea.

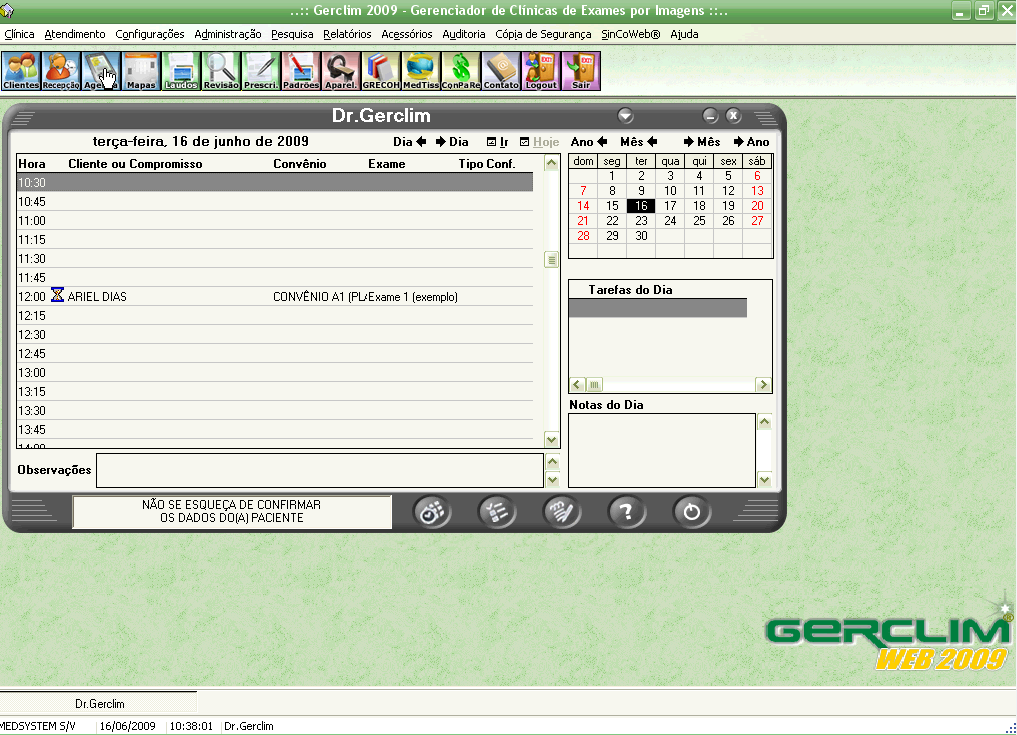


Figura 2.5 - Gerclim - Tela de Agenda

Fonte: http://www.gerclim.com.br

A figura 2.6 apresenta a ficha médica do paciente e parece mais completa e organizada que a do sistema MDMed. No entanto, ela acaba sendo complicada pelo grande número de janelas que são necessárias abrir.

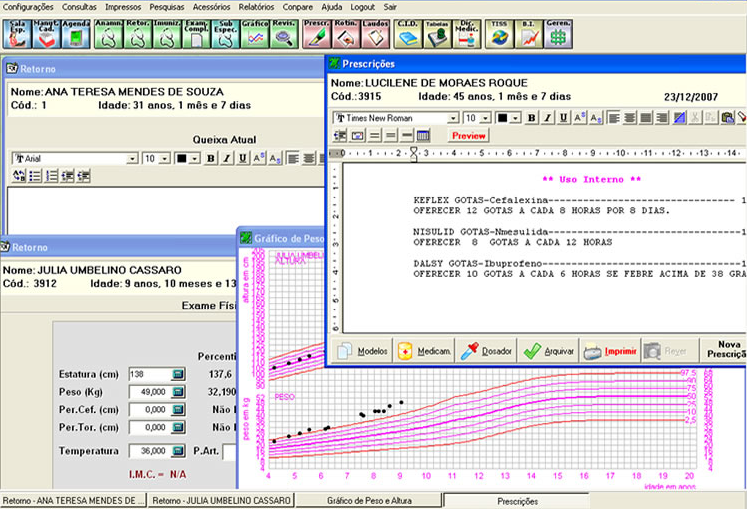


Figura 2.6 - Gerclim - Tela de Ficha Médica

Fonte: http://www.gerclim.com.br

Seguindo a mesma lógica do sistema MDMed, o sistema Gerclim também foi avaliado e comparado com os outros sistemas utilizando a técnica de análise de características proposta por Pfleeger (2004).

## O sistema ProDoctor

Finalmente, foi feita uma análise do terceiro software similar, o ProDoctor. Este software foi lançado no mercado em 1993 e possui hoje cerca de 40 mil usuários. De acordo com o Dr. Everett Langhans da clínica de Quiropraxia, o sistema possui uma interface amigável e bastante clara, porém peca em sua simplicidade, mesmo sendo desenvolvido na plataforma Microsoft.

O processo de vendas é realizado através do website ou por telefone. A implantação deve ser feita pelo próprio usuário e, para receber treinamentos e suporte, o cliente deve pagar um valor adicional. Além do valor das licenças. Os serviços são oferecidos de maneira remota o que, por um lado, é positivo para a empresa porque facilita seu alcance geográfico mas, por outro, priva os clientes de terem um atendimento presencial, dificultando a interatividade, a construção de relacionamento e impactando negativamente nos resultados do serviço.

Os módulos do sistema parecem estar bem integrados entre si. Não há opção de rodar este sistema através de uma plataforma web, é necessário que o usuário instale e rode o programa em cada um dos computadores.

O módulo financeiro do ProDoctor é organizado e atende às necessidades das clínicas médicas, porém, os inúmeros campos desnecessários para a maioria das clínicas de pequeno porte acabam confundindo os usuários na utilização de seus recursos.

Assim como os outros sistemas similares, o ProDoctor possui uma interface amigável e simples, mas acaba pecando da mesma forma que o MDMed: sua interface possui campos desnecessários que acabam não sendo utilizados pelos clientes. A tela de cadastro de pacientes ajuda a demonstrar esta situação (figura 2.7).

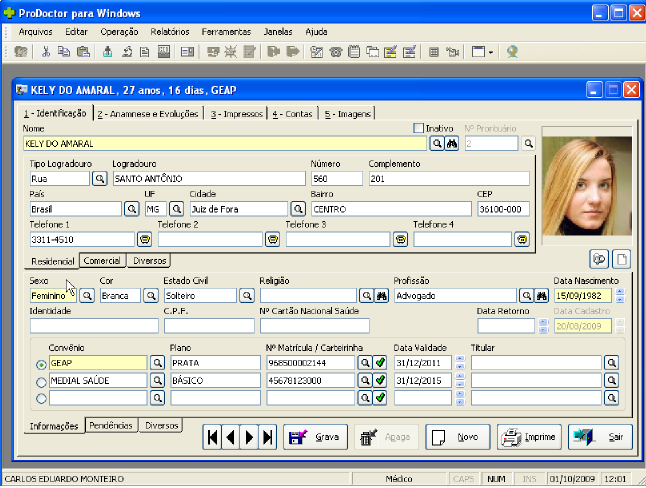


Figura 2.7 - ProDoctor - Tela de Cadastro de Pacientes

Fonte: http://www.prodoctor.net

A figura 2.8 demonstra a tela de agenda do sistema ProDoctor, e assim como acontece nas telas de agendas dos outros sistemas, também não é possível visualizar a agenda de dois médicos de forma simultânea.

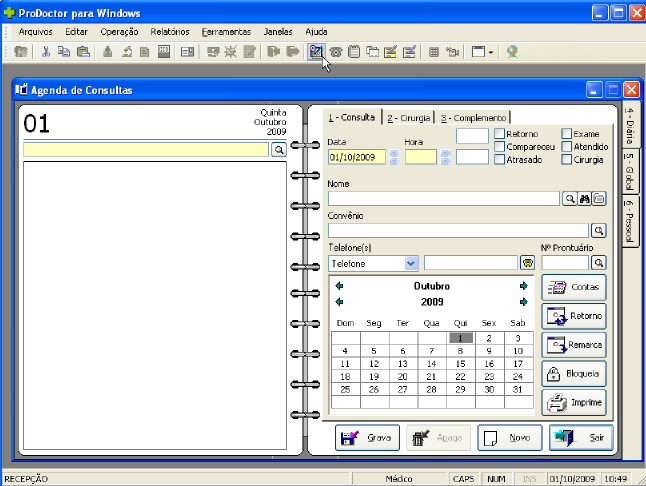


Figura 2.8 - ProDoctor - Tela de Agenda

Fonte: http://www.prodoctor.net

Diferente das telas de Ficha Médica dos outros sistemas analisados, a Ficha Médica do ProDoctor (figura 2.9) possui uma organização e eficiência superior, devido aos seus modelos de fichas que são parametrizáveis.

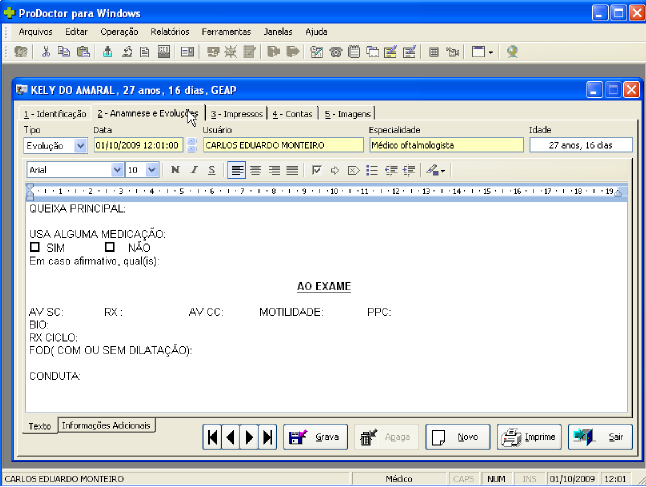


Figura 2.9 - ProDoctor - Tela de Ficha Médica

Fonte: http://www.prodoctor.net

Seguindo o exemplo dos outros sistemas avaliados, o ProDoctor também foi avaliado seguindo a técnica proposta por Pfleeger (2004).

## Análise de características dos sistemas similares

Com a análise dos sistemas similares é possível montar uma tabela comparativa entre estes sistemas, identificar os aspectos positivos e negativos de cada um e descrever sobre o porque de criar este novo sistema, tendo em vista que existem tantos no mercado.

A tabela 2.1 faz parte da técnica de avaliação de análise de características proposta por Pfleeger (2004). Ela demonstra os aspectos positivos e negativos dos sistemas avaliados segundo os principais atributos já definidos.

Tabela 2.1 – Análise de Características

| **Atributos** | **Sistema 1: MDMed** | **Sistema 2: Gerclim** | **Sistema 3: ProDoctor** | **Importância** |
| --- | --- | --- | --- | --- |
| Boa Interface com o usuário /  Simplicidade | 4 | 3 | 4 | 4 |
| Gestão Financeira | 2 | 2 | 3 | 5 |
| Serviços de implementação,  treinamento e suporte | 5 | 4 | 3 | 5 |
| Integração entre os módulos | 3 | 4 | 3 | 5 |
| Plataforma Web | 2 | 2 | 1 | 4 |
| **Pontuação** | 74 | 70 | 65 | — |

Fonte: PFLEEGER, 2004, p.416

Com base na análise das características demonstradas na tabela 2.1 é possível identificar a carência de um sistema que consiga atender as necessidades das clínicas médicas de pequeno porte.

A proposta deste trabalho é oferecer para estas empresas um sistema que atenda às suas necessidades específicas, não apenas de controle, mas também que gere informações com base nestes dados permitindo a tomada de decisões estratégicas por parte dos gestores, uma gestão financeira eficiente e que possua uma interface simples e ao mesmo tempo suficiente, conseguindo assim atender com maior qualidade os usuários do sistema.

Por fim, outro aspecto muito importante para o sistema é, atender os usuários da forma mais clara possível, sem a necessidade deles terem que atualizar o sistema ou ainda instalar este sistema.

1. TECNOLOGIAS

O objetivo deste capítulo é apresentar as tecnologias envolvidas neste projeto. Busca-se caracterizar a linguagem de programação Java, sendo que esta será a linguagem utilizada no desenvolvimento do Sistema de Gestão Médica (SGM), bem como buscar uma visão geral dos frameworks *Java Server Faces* (JSF), *Java Persistence API* (JPA) e *Enterprise Java Beans* (EJB). Ao fim, serão trazidas informações sobre padrões de projetos, dando-se um foco especial em *Model-View-Controller* (MVC) e *Data Access Object* (DAO).

## Linguagem Java

Java foi criada pela Sun Microsystems, sendo inicialmente destinada a dispositivos eletrônicos. No entanto, somente obteve sucesso quando passou a ser utilizada para a internet, já que sua característica multiplataforma possibilita a inclusão de aplicações diretamente na página de internet, independente do sistema utilizado pelo usuário. O processo utilizado na programação em Java consiste em digitar o código, compilar, executar e depurar quando necessário. (THOMPSON, 2005, p.21).

Thompson (2005) considera que as principais características desta linguagem são:

* **Abstração:** a orientação a objetos é tão forte que permite que os desenvolvedores possam trabalhar apenas com uma parte do projeto, sem conhecê-lo em sua plenitude, e obter os resultados esperados;

Abstração é a capacidade de olhar apenas uma parte de um todo, ou seja, retirar da realidade apenas as entidades e fenômenos considerados essenciais, excluindo todos os aspectos irrelevantes ou secundários. Pelo mecanismo de abstração, o ser humano pode entender formas complexas ao dividi-las em partes e analisar cada parte separadamente. (Thompson, 2005, p. 20)

* **Encapsulamento:** as classes e métodos em Java podem ser criados de maneira a não ficarem visíveis às classes com as quais não haja interesse de compartilhamento de dados;
* **Herança:** uma classe já criada irá criar outras classes com as mesmas características, porém com atributos distintos;
* **Polimorfismo:** é a capacidade de dois objetos, de classes distintas, implementarem o mesmo método. Isso pode acontecer por herança ou por interface;

Como programador, você desejará uma linguagem com uma sintaxe agradável, e semântica compreensível (ou seja, não C++). Java se encaixa neste perfil, como dúzias de outras boas linguagens. Algumas linguagens lhe fornecem portabilidade, coleta de lixo, e coisas do tipo, mas elas não possuem muitas opções de bibliotecas, forçando você a se virar para programar sozinho caso queira gráficos bonitos ou acesos a redes ou banco de dados. Bom, Java tem tudo, uma boa linguagem, um ambiente de execução de alta qualidade e uma vasta biblioteca. Essa combinação é o que torna Java uma proposta irresistível para tantos programadores. (HORSTMANN e COMELL, 2005, p. 01)

Horstmann e Cornell (2005), mencionam que Java obteve tanto sucesso por ser uma linguagem simples, orientada a objetos, distribuída, robusta, segura, neutra em relação à arquitetura, portável, interpretada, de alto desempenho e dinâmica.

## JSF – *Java Server Faces*

JSF (*Java Server Faces*) pode ser definido como um *framework* para a criação de componentes de interface de usuário para soluções baseadas na web. Ela pode ser utilizada sozinha ou em conjunto com outras tecnologias que tem objetivos similares e/ou complementares, como por exemplo, o Struts. Os recursos da JSF são a manipulação de eventos, a validação de *inputs* de usuários, especificação de navegação de páginas, tratamento de exceções, suporte à internacionalização e conjunto padrão de componentes gráficos (FRANKLINT, 2006, p.128).

Jacobi e Fallows (2006) definem o JSF como um *framework* de interface de usuário para aplicações web J2EE que, uma vez adotado, permite que as organizações migrem de tecnologias ultrapassadas para plataformas e tecnologias em um padrão mais atualizado. Os autores afirmam que, nos último 15 anos, a indústria de software tem visto muitas tecnologias e plataformas nascerem e morrerem.

Geralmente o uso de uma determinada tecnologia decai por diversas razões, sendo que uma razão muito comum é que as tecnologias são desenvolvidas e mantidas por uma única empresa, forçando os usuários a depender do suporte dos criadores daquela tecnologia. Sempre que o criador decide deixar de usar uma determinada tecnologia em prol de uma solução mais avançada, o usuário fica com uma plataforma sem suporte. O JSF permite que as organizações e consumidores passem para as tecnologias mais recentes assim que elas emergem, com um impacto mínimo nas aplicações JSF já existentes.

## EJB – *Enterprise Java Beans*

O EJB (*Enterprise Java Beans*) é um componente servidor que roda em um *container* EJB do servidor de aplicação. Seu objetivo é o desenvolvimento rápido e simplificado de aplicações Java, com base em componentes, distribuídas, transacionais, seguras e portáteis. Em definição, é um *framework* que permite que os desenvolvedores construam aplicações sem ter que reinventar serviços como transações e segurança, podendo assim focarem-se na lógica de negócio sem ter que investir tempo na construção de código de infra-estrutura. (PANDA, RAHMAN, LANE, 2007, p.5).

## JPA – *Java Persistence API*

O JPA (*Java Persistence API*) permite padronizar o mapeamento de objeto relacional na plataforma Java. O JPA foi definido na versão *Enterprise Java Beans, version 3.0*. Antes disso existia apenas uma forma bem mais complexa para mapear objetos em um banco de dados, que exigia um container EJB. (GONÇALVES, 2007, p.543).

Biswas e Ort (2006) afirmam que a adição do *Java Persistence* API é a maior melhoria do EJB 3.0 em relação à sua versão anterior, a EJB 2.1. A nova versão é mais fácil de se aprender e utilizar e deve resultar no desenvolvimento mais rápido de aplicações.

Os autores destacam como alguns dos principais benefícios do JPA o fato de que requer menor quantidade de classes e interfaces; traz as especificações mais típicas através de notas padrão; permite o mapeamento de objetos relacionados de forma mais clara, fácil e padronizada; elimina a necessidade de buscas por códigos e facilita o teste de entidades fora do container EJB.

## JDBC e SQL

Por causa da pluralidade de banco de dados e de seus provedores de acesso, a Sun desenvolveu a JDBC (*Java Data Base Connectivity*), uma API que tem por objetivo fazer a interface entre a camada do cliente, o *driver* do fabricante a fonte dos dados. Em outras palavras, uma ponte entre a linguagem Java e outra linguagem que todos os bancos de dados suportem. A linguagem escolhida foi o SQL, desta forma os fabricantes de bancos de dados fornecem os *drivers* para SQL e a JDBC cria a API de chamada genérica destes *drivers* (FRANKLINT, 2006, p.43).

O SQL (*Structured Query Language*), ou linguagem de consulta estruturada, é utilizada para estabelecer comunicação com o banco de dados. Thompson (2005) enfatiza que SQL se trata de uma linguagem embutida em outra linguagem, que dá suporte a SQL. Os servidores SQL como, por exemplo, o da Oracle, MySQL ou MS SQL Server, possibilitam o uso de linguagem SQL em toda sua plenitude.

A sua facilidade de uso e simplicidade fizeram da linguagem SQL um padrão de banco de dados. Ao contrário de outras linguagens de consulta a bancos de dados, o SQL determina a forma do resultado e não o caminho para chegar a ele. Adicionalmente, enquanto a maioria das demais linguagens são procedurais, a SQL é declarativa. (DATE, 2004, p.71).

## Padrões de projetos: MVC e DAO

Segundo Gonçalves (2007), as estruturas *Model View Controller* (MVC) e *Data Access Object* (DAO) são conceitos arquitetônicos muito importantes para o desenvolvimento de aplicações *web* escritas em Java. Além destes dois, existem ainda vários outros conceitos englobando o desenvolvimento em Java, conhecidos como *Design Patterns*.

O MVC é um paradigma que tem por objetivo separar uma aplicação em três partes distintas, sendo elas:

* ***Model*:** esta parte está relacionada ao trabalho atual que a aplicação administra, nela estão representados os dados do programa. O modelo não tem conhecimento específico dos controlados e das apresentações, pois são as classes que trabalham no armazenamento e busca de dados;
* ***View*:** a parte das apresentações está relacionada a exibir os dados e informações da aplicação, ou seja, é o responsável pela apresentação visual dos dados disponibilizados pelo *Model*;
* ***Controller*:** o controlador é responsável por coordenar os dois itens anteriores, exibindo a interface correta ou executando trabalhos que a aplicação precisa completar. Este é o objeto que responde às ordens executadas pelo usuário, atuando sobre os dados apresentados pelo modelo, decidindo qual apresentação deverá ser exibida.

A separação lógica da aplicação nestas partes assegura que a camada modelo não sabe nada praticamente do que é exibido; restringido por apresentar as partes de componentes do problema que é resolvido pela aplicação. Igualmente, a camada de apresentação só está relacionada a exibir os dados e não com implementar lógica de negócios que é controlada pela camada modelo. O controlador, como um gerenciador de tráfego, dirige as apresentações a serem exibidas e com as devidas mudanças de dados e recuperações vindas da camada modelo (GONÇALVES, 2007, p. 387)

O padrão DAO é o mais utilizado para acesso a dados contidos em um banco de dados. Gonçalves (2007) cita que colocar as instruções SQL em meio os *scriptlets* somados ao HTML em desenvolvimento de aplicações usando JDBC, embora funcione, é um método confuso e desordeiro. Foi pensando nisso que os desenvolvedores passaram a adotar o padrão DAO.

O padrão DAO fornece uma interface independente, que pode ser usada para persistir objetos de dados. “A idéia é colocar todas as funcionalidades encontradas no desenvolvimento de acesso e trabalho com dados em um só local, tornando simples sua manutenção” (GONÇALVES, 2007, p. 400).

Geralmente, o DAO abrange métodos para inserir, selecionar, atualizar e excluir objetos de um banco de dados. Dependendo de como o padrão DAO é implementado, é possível ter um DAO para cada classe de objetos ou ter um único DAO que é responsável por todos os objetos.

A utilização do Java como linguagem de programação, juntamente com seus *frameworks* e os padrões de projeto MVC e DAO, facilitarão e tornarão mais ágil o desenvolvimento do sistema. Com isso, encerra-se o capítulo sobre as tecnologias envolvidas e parte-se para a análise de requisitos.

1. ANÁLISE E PROJETO

Para a identificação das necessidades específicas das clínicas médicas de pequeno porte, foram feitas entrevistas com o gestor e médico da clínica de quiropraxia Dr. Everett Langhans, localizada na rua Almirante Barroso, 66, no centro de Novo Hamburgo - RS. Também foram analisados os sistemas similares objetivando buscar as melhores práticas efetuadas por cada um. A experiência do autor é outro fator importante no levantamento, análise dos requisitos e projeto, tendo em vista que o mesmo já trabalhou em diversos projetos na área da saúde e também com sistemas de gestão de empresas.

As entrevistas feitas com o gestor da clínica foram realizadas no decorrer do ano de 2010 e tiveram como objetivo levantar as necessidades de gestão da informação desta clínica, que representa os clientes em potencial para o software que este estudo se propõe a projetar.

As informações coletadas nestas entrevistas e na análise dos sistemas similares estão detalhadas neste capítulo e foram estruturadas nas seguintes partes:

1. Requisitos Funcionais (RF);
2. Requisitos Não Funcionais (RNF);
3. Casos de Uso;
4. Matriz de Rastreabilidade;
5. Diagramas de Casos de Uso;
6. Diagrama de Classes; e
7. Detalhamento dos Casos de Uso.

## Requisitos Funcionais

Segundo Pfleeger (2004), requisitos funcionais descrevem uma interação entre o sistema e o seu ambiente, como o sistema deve agir considerando uma determinada ação.

Esta seção apresenta os requisitos funcionais previstos para o sistema proposto.

Tabela 4.1 - Requisitos Funcionais

| ID | Requisitos Funcionais |
| --- | --- |
| RF001 | **Segurança e Controle de Acessos**  **Descrição:** O sistema deverá possuir um controle de acesso com níveis diferentes de permissões para cada tipo de usuário. |
| RF002 | **Cadastros de Usuários, Perfis e Acessos**  **Descrição:** O sistema deverá possuir cadastro de usuários, perfis de usuários e os diferentes acessos que estes possuem. |
| RF003 | **Cadastros de Pacientes e Responsáveis**  **Descrição:** O sistema deverá possuir cadastro de pacientes e de responsáveis. |
| RF004 | **Cadastros de Clínicas**  **Descrição:** O sistema deverá possuir cadastro de clínicas e seus horários de atendimento. |
| RF005 | **Cadastro de Funcionários e Médicos**  **Descrição:** O sistema deverá possuir cadastro de funcionários, médicos, as funções destes funcionários e as especialidades destes médicos. |
| RF006 | **Cadastros gerais**  **Descrição:** O sistema deverá possuir cadastros gerais tais como cidade, estado, país, sexo, profissão, estado civil, etc.. |
| RF007 | **Agendas Médicas**  **Descrição:** O sistema deverá possuir uma agenda por médico para cada clínica e deve ser possível definir os horários de atendimento dos médicos. |
| RF008 | **Atendimentos**  **Descrição:** O sistema deverá possuir apenas um atendimento por horário, cada atendimento deve possuir um tipo de consulta (Ex.: Reconsulta), estar ligado a um paciente e a um médico. |
| RF009 | **Tratamentos e Diagnósticos**  **Descrição:** O sistema deverá possuir cadastro de tratamentos e diagnósticos que serão associados a um atendimento quando este necessitar. |
| RF010 | **Procedimentos**  **Descrição:** O sistema deverá possuir cadastro procedimentos, seus tipos de procedimentos e preparos de procedimentos. |
| RF011 | **Anamnese de Pacientes**  **Descrição:** O sistema deverá possuir uma anamnese do paciente que demonstre o histórico médico (atendimentos) deste paciente. |
| RF012 | **Cadastros de Fornecedores e Insumos**  **Descrição:** O sistema deverá possuir cadastro de fornecedores, insumos e tipos de insumos, bem como o histórico dos insumos. |
| RF013 | **Controle de estoque**  **Descrição:** O sistema deverá possuir controle de entrada e saída de insumos e suas quantidades mínimas. |
| RF014 | **Cadastros Financeiros**  **Descrição:** O sistema deverá possuir cadastro das formas de pagamento, tipos de lançamento, tipos de documento e Centro de custos. |
| RF015 | **Fluxo de Caixa**  **Descrição:** O sistema deve permitir o cadastro de mais de um caixa para cada clínica, este caixa deve possuir registros de movimentações do dia com a forma de pagamento, o tipo de lançamento e o centro de custos deste movimento. |
| RF016 | **Contas a Pagar e Contas a Receber**  **Descrição:** O sistema deverá possuir um controle de contas a pagar e contas a receber, os tipos de documentos destas contas e o centro de custos para o qual esta conta se destina. |

## Requisitos não funcionais

Os requisitos não funcionais descrevem uma restrição que deve ser atendida, limitando as opções para que se possa criar uma solução para o problema (PFLEEGER, 2004, p.115). Esta seção apresenta os requisitos não funcionais previstos para o sistema proposto.

Tabela 4.2 - Requisitos Não Funcionais

| ID | Requisitos Não Funcionais |
| --- | --- |
| RNF001 | **Navegadores**  **Descrição:** Os navegadores utilizados podem ser o *Mozilla Firefox 3.0* ou superior, *Google Chrome 6.0* ou superior e o *Safari 5.0* ou superior, visto que atendem ao padrão estabelecido pela W3C. |
| RNF002 | **Desenvolvimento**  **Descrição:** O sistema deverá ser desenvolvido utilizando a tecnologia *Java Enterprise Edition 6.0 (Java EE)* ou superior, bem como os *frameworks Java Server Faces (JSF)*, *Enterprise Java Beans (EJB)* e *Java Persistence API (JPA)*. |
| RNF003 | **Servidor de Aplicação**  **Descrição:** O servidor de aplicação que deve ser utilizado é o *Glassfish 3* ou superior. |
| RNF004 | **Banco de Dados**  **Descrição:** O banco de dados utilizado deverá ser o *MySQL 5.0* ou superior. |
| RNF005 | **Agilidade**  **Descrição:** O sistema deverá possuir um tempo de resposta (para consultas) inferior a 4 segundos. |
| RNF006 | **Auditoria**  **Descrição:** Todo o sistema deverá possuir controle de auditoria, bem como um espelho dos registros alterados e excluídos nos últimos 30 dias. |

## Casos de Uso

Pfleeger (2004) diz que casos de uso descrevem as funcionalidades específicas que um sistema deve desempenhar. Cada caso de uso descreve um possível cenário de interação que um sistema externo ou outra entidade tem com o sistema a ser desenvolvido. Os casos de uso que foram identificados como necessários para o sistema estão descritos nas seções a seguir, divididos de acordo com os módulos correspondentes.

### Módulo de Segurança

O módulo de segurança engloba os casos de uso responsáveis pela segurança, níveis de acessos e permissões do sistema. A tabela 4.3 apresenta os casos de uso deste módulo.

Tabela 4.3 - Casos de Uso - Módulo de Segurança

| ID | Casos de Uso - Módulo de Segurança |
| --- | --- |
| UC001 | **Acesso**  **Descrição:** Têm a responsabilidade de liberar o acesso às telas do sistema para um determinado perfil de usuários. Cada acesso possui as ações permitidas para uma tela. |
| UC002 | **Perfil**  **Descrição:** Sua responsabilidade é separar em grupos os diferentes tipos de usuários do sistema. |
| UC003 | **Tela**  **Descrição:** É responsável pelas telas do sistema ao qual um determinado usuário pode possuir acesso. |
| UC004 | **Usuário**  **Descrição:** É o método de identificação individual das pessoas que utilizarão o sistema. |
| UC005 | **Alterar Senha**  **Descrição:** É o responsável por alterar as senhas dos usuários do sistema. Cada usuário poderá alterar somente a sua senha, independente do seu perfil. |

### Módulo de Cadastros

O módulo de cadastros engloba os casos de uso responsáveis pelos cadastros gerais do sistema. A tabela 4.4 apresenta os casos de uso deste módulo.

Tabela 4.4 - Casos de Uso - Módulo de Cadastros

| ID | Casos de Uso - Módulo de Cadastros |
| --- | --- |
| UC006 | **Cidade**  **Descrição:** Mantém o cadastro de cidades do sistema. |
| UC007 | **Clínica**  **Descrição:** Mantém o cadastro de clínicas do sistema. |
| UC008 | **Especialidade Médica**  **Descrição:** Mantém o cadastro de especialidades médicas do sistema. |
| UC009 | **Estado Civil**  **Descrição:** Mantém o cadastro de estado civil do sistema. |
| UC010 | **Função de Funcionário**  **Descrição:** Mantém o cadastro de funções de funcionários do sistema. |
| UC011 | **Funcionário**  **Descrição:** Mantém o cadastro de funcionários do sistema. |
| UC046 | **Médico**  **Descrição:** Mantém o cadastro de médicos do sistema. |
| UC012 | **Horários de Clínica**  **Descrição:** Mantém o cadastro de horários de atendimento das clínicas do sistema. |
| UC013 | **Paciente**  **Descrição:** Mantém o cadastro de pacientes do sistema. |
| UC014 | **País**  **Descrição:** Mantém o cadastro de países do sistema. |
| UC015 | **Profissão**  **Descrição:** Mantém o cadastro de profissões do sistema. |
| UC016 | **Responsável**  **Descrição:** Mantém o cadastro de responsáveis do sistema. |
| UC017 | **Sexo**  **Descrição:** Mantém o cadastro de sexos do sistema. |
| UC018 | **Tipo Conselho Regional**  **Descrição:** Mantém o cadastro de tipos de conselho regional do sistema. |
| UC019 | **Tipo Cor**  **Descrição:** Mantém o cadastro de cores, para identificação de médicos do sistema e de identificação de consultas na agenda. |
| UC020 | **Tipo Logradouro**  **Descrição:** Mantém o cadastro de tipos de logradouros do sistema. |
| UC021 | **Estado (UF)**  **Descrição:** Mantém o cadastro de estados do sistema. |

### Módulo de Agenda

O módulo de agenda engloba os casos de uso responsáveis pelas agendas de médicos do sistema. A tabela 4.5 apresenta os casos de uso deste módulo.

Tabela 4.5 - Casos de Uso - Módulo de Agenda

| ID | Casos de Uso - Módulo de Agenda |
| --- | --- |
| UC022 | **Agenda**  **Descrição:** Tem como responsabilidade armazenar e marcar as consultas de pacientes para cada médico de uma clínica. |
| UC023 | **Horários de Agenda**  **Descrição:** São definidos aqui os horários das agendas dos médicos para cada clínica que este médico atenda. |

### Módulo de Prontuário

Segundo Freire, o prontuário eletrônico está cada vez mais reconhecido perante o Conselho Federal de Medicina.

O Conselho Federal de Medicina tem adotado diversas resoluções, que regulamentam, entre outras questões, o relacionamento médico-paciente, o relacionamento dos médicos com operadoras de planos de saúde, participação de pacientes em pesquisas, divulgação de estudos científicos e o prontuário médico do paciente. (FREIRE, p.13)

O módulo de prontuário engloba os casos de uso responsáveis pelas informações médicas dos pacientes do sistema. A tabela 4.6 apresenta os casos de uso deste módulo.

Tabela 4.6 - Casos de Uso - Módulo de Prontuário

| ID | Casos de Uso - Módulo de Prontuário |
| --- | --- |
| UC024 | **Anamnese**  **Descrição:** Tem como responsabilidade armazenar o histórico dos atendimentos dos pacientes, bem como outras informações médicas de caráter específico. |
| UC025 | **Atendimento**  **Descrição:** Tem como responsabilidade armazenar as informações da consulta que o paciente realizou em um determinado dia. Aqui também são armazenadas as informações dos procedimentos realizados e dos diagnósticos e tratamentos identificados pelo médico. |
| UC026 | **Diagnóstico**  **Descrição:** Mantém o cadastro de diagnósticos do sistema. |
| UC027 | **Preparos de Procedimentos**  **Descrição:** Mantém o cadastro de preparos de procedimentos do sistema. |
| UC028 | **Procedimento**  **Descrição:** Mantém o cadastro de procedimentos do sistema. |
| UC029 | **Tipo de Consulta**  **Descrição:** Mantém o cadastro de tipos de consultas do sistema. |
| UC030 | **Tipo de Procedimento**  **Descrição:** Mantém o cadastro de tipos de procedimentos do sistema. |
| UC031 | **Tratamento**  **Descrição:** Mantém o cadastro de tratamentos do sistema. |

### Módulo de Estoque

O módulo de estoque engloba os casos de uso responsáveis pelas informações de estoque de insumos do sistema. A tabela 4.7 apresenta os casos de uso deste módulo.

Tabela 4.7 - Casos de Uso - Módulo de Estoque

| ID | Casos de Uso - Módulo de Estoque |
| --- | --- |
| UC032 | **Fornecedor**  **Descrição:** Mantém o cadastro de fornecedores do sistema. |
| UC033 | **Histórico de Insumos**  **Descrição:** Mantém as informações de todas as alterações em um determinado insumo. |
| UC034 | **Insumo**  **Descrição:** Mantém o cadastro de insumos do sistema. |
| UC035 | **Lançamento de Insumo**  **Descrição:** Mantém o cadastro de lançamentos de insumos no estoque do sistema. |
| UC036 | **Tipos de Insumo**  **Descrição:** Mantém o cadastro de tipos de insumo do sistema. |
| UC037 | **Tipos de Lançamento de Insumo**  **Descrição:** Mantém o cadastro de tipos de lançamentos de insumos do sistema. |
| UC038 | **Unidades de Insumos**  **Descrição:** Mantém o cadastro de unidades de insumos do sistema. |

### Módulo Financeiro

O módulo financeiro engloba os casos de uso responsáveis pelas informações financeiras do sistema. A tabela 4.8 apresenta os casos de uso deste módulo.

Tabela 4.8 - Casos de Uso - Módulo Financeiro

| ID | Casos de Uso - Módulo Financeiro |
| --- | --- |
| UC039 | **Fluxo de Caixa**  **Descrição:** Tela responsável por todos os movimentos financeiros de um caixa de uma clínica. |
| UC040 | **Contas a Pagar e Contas a Receber**  **Descrição:** Responsável por todos os movimentos financeiros futuros de uma clínica, independente se for um lançamento de receita ou despesa. |
| UC041 | **Pagamento de Consultas**  **Descrição:** Responsável por realizar o pagamento de uma consulta e lançar no fluxo de caixa ou contas a receber. Deve ser possível efetuar o pagamento parcelado das consultas através desta tela. |
| UC042 | **Tipo de Documento**  **Descrição:** Mantém o cadastro de tipos de documentos do sistema. |
| UC043 | **Tipo de Lançamento**  **Descrição:** Mantém o cadastro de tipos de lançamentos financeiros do sistema. |
| UC044 | **Forma de Pagamento**  **Descrição:** Mantém o cadastro de formas de pagamentos do sistema. |
| UC045 | **Centro de Custo**  **Descrição:** Mantém o cadastro de centro de custos do sistema. |

## Matriz de Rastreabilidade

A matriz de rastreabilidade tem o intuito de demonstrar a relação entre os requisitos funcionais do sistema e os casos de uso que irão atender estes requisitos. A tabela 4.9 apresenta o mapeamento dos requisitos funcionais em relação aos casos de uso.

Tabela 4.9 - Matriz de Rastreabilidade

|  | RF  001 | RF  002 | RF  003 | RF  004 | RF  005 | RF  006 | RF  007 | RF  008 | RF  009 | RF  010 | RF  011 | RF  012 | RF  013 | RF  014 | RF  015 | RF  016 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| UC001 | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC002 | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC003 | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC004 | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC005 | X | X |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC006 |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |
| UC007 |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |
| UC008 |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |
| UC009 |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |
| UC010 |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |
| UC011 |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |
| UC012 |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |
| UC013 |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC014 |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |
| UC015 |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |
| UC016 |  |  | X |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC017 |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |
| UC018 |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |
| UC019 |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |
| UC020 |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |
| UC021 |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |
| UC022 |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |
| UC023 |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |
| UC024 |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |
| UC025 |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |
| UC026 |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |
| UC027 |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |
| UC028 |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |
| UC029 |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |
| UC030 |  |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |
| UC031 |  |  |  |  |  |  |  |  | X |  |  |  |  |  |  |  |
| UC032 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |
| UC033 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |
| UC034 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |
| UC035 |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| UC036 |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |  |
| UC037 |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| UC038 |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |  |
| UC039 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |
| UC040 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |
| UC041 |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X | X |
| UC042 |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |
| UC043 |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |
| UC044 |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |
| UC045 |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |  |
| UC046 |  |  |  |  | X |  |  |  |  |  |  |  |  |  |  |  |

## Diagrama de Casos de Uso

De acordo com Booch, Rumbaugh e Jacobson (2000), diagramas de casos de uso fazem parte dos diagramas disponíveis na *UML*, e possuem papel central na modelagem de um sistema. Os diagramas de casos ajudam a visualizar, especificar e documentar o comportamento de elementos do sistema.

Booch, Rumbaugh e Jacobson (2000) comentam que os diagramas de casos de uso costumam possuir os seguintes conteúdos:

* Casos de uso;
* Atores; e
* Relacionamentos de dependência, generalização e associação.

Na seção 4.3 foi apresentada a relação de casos de uso do sistema e na seção anterior (seção 4.4) foi realizado um mapeamento de rastreabilidade entre os requisitos funcionais e os casos de uso. Neste momento serão demonstrados os atores que afetam estes casos de uso. Todos os atores foram definidos seguindo um perfil de acessos e permissões ao sistema. Eles foram divididos em:

* **Administrador:** é o ator com o perfil mais completo. Possui acesso a todos os recursos do sistema;
* **Administrativo:** é o ator com perfil específico para acessar recursos com informações administrativas e financeiras mais completas;
* **Médico:** é o ator com perfil para acessar os recursos com informações técnicas sobre os pacientes;
* **Secretária:** é o ator com o perfil para controlar as agendas dos médicos, cadastros gerais do sistema e algumas informações financeiras mais superficiais.

Para evitar a poluição visual, o ator “Administrador” só aparece nos casos de uso onde só ele possua perfil de acesso.

Nas seções seguintes são demonstrados os diagramas de casos de uso divididos em módulos, seguindo a mesma divisão já utilizada na seção 4.3.

### Segurança

A figura 4.1 apresenta o diagrama de caso de uso do módulo de segurança.



Figura 4.1 - Diagrama de Caso de Uso do Módulo de Segurança

### Cadastros

A figura 4.2 apresenta o diagrama de casos de uso do módulo de cadastros na visão do ator “secretária”.



Figura 4.2 - Diagrama de Caso de Uso do Módulo de Cadastro – Visão da Secretária

A figura 4.3 apresenta o diagrama de casos de uso do módulo de cadastros na visão do ator “médico”.



Figura 4.3 - Diagrama de Caso de Uso do Módulo de Cadastro – Visão do Médico

A figura 4.4 apresenta o diagrama de casos de uso do módulo de cadastros na visão do ator “administrativo”.



Figura 4.4 - Diagrama de Caso de Uso do Módulo de Cadastro – Visão do Administrativo

### Agenda

A figura 4.5 apresenta o diagrama de casos de uso do módulo de agenda.



Figura 4.5 - Diagrama de Caso de Uso do Módulo de Agenda

### Prontuário

A figura 4.6 apresenta o diagrama de casos de uso do módulo de prontuário.



Figura 4.6 - Diagrama de Caso de Uso de Prontuário

### Estoque

A figura 4.7 apresenta o diagrama de casos de uso do módulo de estoque.



Figura 4.7 - Diagrama de Caso de Uso de Estoque

### Financeiro

A figura 4.8 apresenta o diagrama de casos de uso do módulo financeiro.



Figura 4.8 - Diagrama de Caso de Uso Financeiro

## Diagrama de Classes

Booch, Rumbaugh e Jacobson (2000), dizem que os diagramas de classes são os diagramas mais usados nas modelagens de sistemas orientados a objetos, eles demonstram a visão estática do projeto do sistema.

Os diagramas de classes são importantes não só para a visualização, a especificação e a documentação de modelos estruturais, mas também para a construção de sistemas executáveis por intermédio de engenharia de produção e reversa. (BOOCH, RUMBAUGH, JACOBSON, 2000, p. 104)

Todos os diagramas de classes exemplificados no decorrer da seção 4.6 possuem herança da classe Bean. Esta classe é a representação de um conjunto de atributos que todas as classes do sistema devem possuir. Além disso, a classe Bean possui uma interface chamada IBean, que garante que a classe Bean terá que implementar os métodos necessários para as outras classes dos sistema.

Na figura 4.9 é possível visualizar esta classe e esta interface que são referência para todas as outras classes do sistema.



Figura 4.9 - Diagrama das Classes Comuns

Nas seções seguintes são demonstrados os diagramas de classes divididos em módulos, seguindo a mesma divisão já utilizada na seção 4.5.

### Segurança

A figura 4.10 apresenta o diagrama de classes do módulo de segurança.



Figura 4.10 - Diagrama de Classes do Módulo de Segurança

### Cadastros

A figura 4.11 apresenta o diagrama de classes do módulo de cadastros.



Figura 4.11 - Diagrama de Classes do Módulo de Cadastros

### Agenda

A figura 4.12 apresenta o diagrama de classes do módulo de agenda.



Figura 4.12 - Diagrama de Classes do Módulo de Agenda

### Prontuário

A figura 4.13 apresenta o diagrama de classes do módulo de prontuário.



Figura 4.13 - Diagrama de Classes do Módulo de Prontuário

### Estoque

A figura 4.14 apresenta o diagrama de classes do módulo de estoque.



Figura 4.14 - Diagrama de Classes do Módulo de Estoque

### Financeiro

A figura 4.15 apresenta o diagrama de classes do módulo financeiro.



Figura 4.15 - Diagrama de Classes do Módulo de Financeiro

## Detalhamento dos casos de uso

Casos de uso são muito populares, pois contam como um sistema irá se comportar. Os usuários do sistema conseguem enxergar o que será o novo sistema e conseguem reagir mais rápido para aceitar ou rejeitar o caso de uso. Esta é apenas uma das maneiras pelas quais os casos de uso contribuem para o desenvolvimento do sistema (COCKBURN, 2005).

Cockburn (2005) cita que existem inúmeros modelos de desenvolvimento de casos de uso, o modelo inteiramente completo é o utilizado nos casos de uso descritos nas seções a seguir.

Embora todos os casos de uso descritos na seção 4.3 tenham sido detalhados, esta seção apresenta somente o detalhamento dos principais casos de uso do sistema.

### Cadastros

A seguir é apresentado o detalhamento do caso de uso **UC007 - Cadastro de Clínicas.** Este caso de uso tem por objetivo demonstrar a maneira pela qual o usuário irá interagir com o cadastro de clínicas.

|  |
| --- |
| 1. **Nome do Caso de Uso: UC007 - Cadastro de Clínicas**    1. **Descrição Resumida**   Este caso de uso descreve e controla operações para Criar, Alterar, Consultar e Excluir Clínicas.   * 1. **Atores** * Administrativo; * Administrador; * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário seleciona a pesquisa de clínicas.   * + 1. **Pesquisar uma clínica**        1. Usuário informa os dados do filtro e seleciona a pesquisa;        2. O sistema exibe uma lista com as clinicas que são atendidas por este filtro;        3. Usuário seleciona a clínica desejada;        4. O sistema exibe as informações da clínica.   1. **Fluxo Alternativo**      1. **Criar uma clínica**         1. Estando na tela de pesquisa, o usuário seleciona a opção de nova clínica;         2. O sistema exibe a tela de cadastro de clínicas para o usuário;         3. O usuário informa os dados da clínica e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.1.6.;         5. O usuário informa os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados da clínica e informa ao usuário.      2. **Alterar uma clínica**         1. Usuário seleciona uma clínica na pesquisa e coloca para edição;         2. Sistema exibe a tela de cadastros para o usuário;         3. Usuário altera os dados da clínica e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.2.5.;         5. O sistema salva os dados da clínica e informa ao usuário.      3. **Consultar uma clínica (Ocorre somente quando o usuário não possui permissão para editar uma clínica)**         1. Usuário seleciona uma clínica na pesquisa e coloca para edição;         2. Sistema exibe a tela de consulta da clínica para o usuário;         3. Usuário visualiza os dados da clínica e seleciona sair.      4. **Excluir uma clínica**         1. Usuário seleciona uma clínica da pesquisa e marca para exclusão;         2. Sistema mostra alerta de confirmação de exclusão para o usuário;         3. Se o usuário cancelar a exclusão, o sistema volta para a tela de pesquisa, caso contrário, segue o fluxo;         4. Sistema informa para o usuário, que a clínica foi excluída ou que não é possível excluir uma clínica que possua ligações com outros cadastros;         5. Usuário volta para tela de pesquisa de clínicas.  1. **Requisitos Especiais**   Ao incluir ou alterar uma clínica, é obrigatório o cadastro de horários de funcionamento desta clínica. Este cadastro é feito através UC012 – Cadastro de Horários de Clínicas.   1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**   Nenhum. |

O caso de uso **UC011 - Cadastro de Funcionário** tem por objetivo demonstrar a maneira pela qual o usuário irá interagir com o cadastro de funcionários.

|  |
| --- |
| 1. **Nome do Caso de Uso: UC011 - Cadastro de Funcionário**    1. **Descrição Resumida**   Este caso de uso descreve e controla operações para Criar, Alterar, Consultar e Excluir Funcionários.   * 1. **Atores** * Secretária; * Profissional Especializado (Ex.: Médico); * Administrativo; * Administrador; * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário seleciona a pesquisa de funcionários.   * + 1. **Pesquisar um funcionário**        1. Usuário informa os dados do filtro e seleciona a pesquisa;        2. O sistema exibe uma lista com os funcionários que são atendidos por este filtro;        3. Usuário seleciona o funcionário desejado;        4. O sistema exibe as informações do funcionário.   1. **Fluxo Alternativo**      1. **Criar um funcionário**         1. Estando na tela de pesquisa, o usuário seleciona a opção de um novo funcionário;         2. O sistema exibe a tela de cadastro de funcionários para o usuário;         3. O usuário informa os dados do funcionário e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.1.6.;         5. O usuário informa os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados do funcionário e informa ao usuário.      2. **Alterar um funcionário**         1. Usuário seleciona um funcionário da pesquisa e coloca para edição;         2. Sistema exibe a tela de cadastros para o usuário;         3. Usuário altera os dados do funcionário e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.2.5.;         5. O sistema salva os dados do funcionário e informa ao usuário.      3. **Consultar um funcionário (Ocorre somente quando o usuário não possui permissão para editar um funcionário)**         1. Usuário seleciona um funcionário da pesquisa e coloca para edição;         2. Sistema exibe a tela de consulta do funcionário para o usuário;         3. Usuário visualiza os dados do funcionário e seleciona sair.      4. **Excluir um funcionário**         1. Usuário seleciona um funcionário da pesquisa e marca para exclusão;         2. Sistema mostra alerta de confirmação de exclusão para o usuário;         3. Se o usuário cancelar a exclusão, o sistema volta para a tela de pesquisa, caso contrário, segue o fluxo;         4. Sistema informa para o usuário, que o funcionário foi excluído ou que não é possível excluir um funcionário que possua ligações com outros cadastros;         5. Usuário volta para tela de pesquisa de funcionários.  1. **Requisitos Especiais**   É neste cadastro que é feito o lançamento da informação do tempo padrão de atendimento deste funcionário quando o mesmo for um profissional especializado.   1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**   Nenhum. |

O caso de uso **UC012 - Cadastro de Horários de Clínicas** tem por objetivo demonstrar a maneira pela qual o usuário irá interagir com o cadastro de horários de clínicas.

|  |
| --- |
| 1. **Nome do Caso de Uso: UC012 - Cadastro de Horários de Clínicas**    1. **Descrição Resumida**   Este caso de uso descreve e controla operações para Criar, Alterar, Consultar e Excluir Horários de Clínicas.   * 1. **Atores** * Administrativo; * Administrador; * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário está no cadastro de uma clínica.   * + 1. **Criar um horário**        1. Estando na tela de cadastro de clínicas, o usuário seleciona a opção de um novo horário para a clínica;        2. O sistema exibe a tela de cadastro de horários para o usuário;        3. O usuário informa os dados do horário e seleciona salvar;        4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.1.1.6.;        5. O usuário informa os dados obrigatórios e seleciona salvar;        6. O sistema salva os dados do horário, informa ao usuário e volta para o cadastro da clínica.   1. **Fluxo Alternativo**      1. **Alterar um horário**         1. Usuário seleciona um horário na lista de horários no cadastro de clínicas e coloca para edição;         2. Sistema exibe a tela de cadastro para o usuário;         3. Usuário altera os dados do horário e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.1.5.;         5. O sistema salva os dados do horário, informa ao usuário e volta para a tela de cadastro de clínicas.      2. **Consultar um horário**         1. O Usuário pode visualizar os horários de atendimento da clínica através da tela de edição ou consulta de uma clínica.      3. **Excluir um horário**         1. Usuário seleciona um horário na tela de cadastro de clínicas e marca para exclusão;         2. Sistema mostra alerta de confirmação de exclusão para o usuário;         3. Se o usuário cancelar a exclusão, o sistema volta para a tela de cadastro de clínicas, caso contrário, segue o fluxo;         4. Sistema informa para o usuário, que o horário foi excluído e volta para a tela de cadastro de clínicas.  1. **Requisitos Especiais**   Todos os horários de clínicas são salvos no momento em que a clínica for salva, somente na tela de inclusão de clínicas.   1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**   Nenhum. |

O caso de uso **UC013 - Cadastro de Paciente** tem por objetivo demonstrar a maneira pela qual o usuário irá interagir com o cadastro paciente.

|  |
| --- |
| 1. **Nome do Caso de Uso: UC013 - Cadastro de Paciente**    1. **Descrição Resumida**   Este caso de uso descreve e controla operações para Criar, Alterar, Consultar e Excluir Pacientes.   * 1. **Atores** * Secretária; * Profissional Especializado (Ex.: Médico); * Administrativo; * Administrador; * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário seleciona a pesquisa de pacientes.   * + 1. **Pesquisar um paciente**        1. Usuário informa os dados do filtro e seleciona a pesquisa;        2. O sistema exibe uma lista com os pacientes que são atendidos por este filtro;        3. Usuário seleciona o paciente desejado;        4. O sistema exibe as informações do paciente.   1. **Fluxo Alternativo**      1. **Criar um paciente**         1. Estando na tela de pesquisa, o usuário seleciona a opção de um novo paciente;         2. O sistema exibe a tela de cadastro de pacientes para o usuário;         3. O usuário informa os dados do paciente e seleciona salvar paciente;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.1.6.;         5. O usuário informa os dados obrigatórios e seleciona salvar paciente;         6. O sistema salva os dados do paciente e informa ao usuário.      2. **Alterar um paciente**         1. Usuário seleciona um paciente da pesquisa e coloca para edição;         2. Sistema exibe a tela de cadastros para o usuário;         3. Usuário altera os dados do paciente e seleciona salvar paciente;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.2.5.;         5. O sistema salva os dados do paciente e informa ao usuário.      3. **Consultar um paciente (Ocorre somente quando o usuário não possui permissão para editar um paciente)**         1. Usuário seleciona um paciente da pesquisa e coloca para edição;         2. Sistema exibe a tela de consulta do paciente para o usuário;         3. Usuário visualiza os dados do paciente e seleciona sair.      4. **Excluir um paciente**         1. Usuário seleciona um paciente da pesquisa e marca para exclusão;         2. Sistema mostra alerta de confirmação de exclusão para o usuário;         3. Se o usuário cancelar a exclusão, o sistema volta para a tela de pesquisa, caso contrário, segue o fluxo;         4. Sistema informa para o usuário, que o paciente foi excluído ou que não é possível excluir um paciente que possua ligações com outros cadastros;         5. Usuário volta para tela de pesquisa de pacientes.  1. **Requisitos Especiais**   Nenhum requisito especial foi especificado para este caso de uso.   1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**   Nenhum. |

### Agenda

A seguir é apresentado o detalhamento do caso de uso **UC022 – Agenda.** Este caso de uso tem por objetivo demonstrar a maneira pela qual o usuário irá interagir com a agenda de um ou mais médicos.

|  |
| --- |
| 1. **Nome do Caso de Uso: UC022 – Agenda**    1. **Descrição Resumida**   Este caso de uso tem como responsabilidade armazenar e marcar agendamentos de pacientes para cada médico de uma clínica.   * 1. **Atores** * Secretária; * Profissional Especializado (Ex.: Médico); * Administrador; * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário seleciona a agenda para um ou mais médicos.   * + 1. **Pesquisar um agendamento**        1. Usuário seleciona uma clínica na lista que aparece em tela (Caso seja um usuário com acesso a mais de uma clínica, caso contrário aparece apenas à clínica deste usuário pré-selecionada);        2. Usuário seleciona um ou mais médicos correspondentes à clínica (Aparecem apenas os médicos vinculados aquela clínica);        3. O sistema exibe as agendas dos médicos selecionados identificando cada um com a cor definida para este no seu cadastro.   1. **Fluxo Alternativo**      1. **Criar um agendamento**         1. Estando na tela de agenda, o usuário seleciona o dia e horário para agendamento;         2. O sistema exibe a tela de cadastro de agendamentos para o usuário;         3. O usuário informa os dados do paciente (solicitado apenas o nome e telefone para pacientes ainda não cadastrados) e do médico escolhido, confirma o horário e tempo do agendamento, informa o tipo de agendamento (tipo de consulta) e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.1.6.;         5. O usuário informa os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados do agendamento, informa ao usuário, fecha a tela de cadastro e atualiza a agenda com o novo agendamento.      2. **Alterar um agendamento**         1. Estando na tela de agenda, o usuário seleciona o agendamento que deseja alterar;         2. O sistema exibe a tela de cadastro do agendamento para o usuário;         3. O usuário informa ou altera os dados desejados e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.2.6.;         5. O usuário informa os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados do agendamento, informa ao usuário, fecha a tela de cadastro e atualiza a agenda com as alterações do agendamento.      3. **Consultar um agendamento (Ocorre somente quando o usuário não possui permissão para editar um agendamento)**         1. Usuário seleciona um agendamento da agenda e coloca para edição;         2. Sistema exibe a tela de consulta do agendamento para o usuário;         3. Usuário visualiza os dados do agendamento e seleciona sair.      4. **Excluir um agendamento**         1. Estando na tela de agenda, o usuário seleciona o agendamento que deseja excluir;         2. O sistema exibe a tela de cadastro do agendamento para o usuário;         3. O usuário seleciona excluir;         4. Sistema mostra alerta de confirmação de exclusão para o usuário;         5. Se o usuário cancelar a exclusão, o sistema volta para a tela de cadastro, caso contrário, segue o fluxo;         6. Sistema informa para o usuário, que o agendamento foi excluído ou que não é possível excluir um agendamento que possua um atendimento;         7. O sistema volta para tela de agenda e a atualiza.  1. **Requisitos Especiais**   Nenhum requisito especial foi especificado para este caso de uso.   1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**   Estando em edição na tela de cadastro de um agendamento, o usuário pode selecionar a qualquer momento a opção de realizar um atendimento **(UC 0025)**. |

O caso de uso **UC023 - Horários de Agenda** tem por objetivo demonstrar a maneira pela qual o usuário irá interagir com o cadastro de horários da agenda do médico.

|  |
| --- |
| 1. **Nome do Caso de Uso: UC023 – Horários de Agendas**    1. **Descrição Resumida**   Este caso de uso tem como responsabilidade definir os horários de atendimento de cada médico para uma clínica.   * 1. **Atores** * Secretária; * Profissional Especializado (Ex.: Médico); * Administrador; * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário seleciona a pesquisa de horários de agenda.   * + 1. **Pesquisar um horário de agenda**        1. Usuário seleciona uma clínica na lista que aparece em tela (Caso seja um usuário com acesso a mais de uma clínica, caso contrário aparece apenas à clínica deste usuário pré-selecionada);        2. Usuário seleciona um médico correspondente à clínica (Aparecem apenas os médicos vinculados aquela clínica);        3. O sistema exibe uma lista com os horários da agenda do médico.   1. **Fluxo Alternativo**      1. **Criar um horário de agenda**         1. Estando na tela de pesquisa de horários de agenda, o usuário seleciona novo;         2. O sistema exibe a tela de cadastro de horários de agenda para o usuário;         3. O usuário informa os dados do novo horário;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.1.6.;         5. O usuário informa os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados do horário, informa ao usuário, fecha a tela de cadastro e atualiza a lista com o novo horário.      2. **Alterar um agendamento**         1. Estando na tela de pesquisa de horários de agenda, o usuário seleciona o horário que deseja alterar;         2. O sistema exibe a tela de cadastro de horários de agenda para o usuário;         3. O usuário informa ou altera os dados desejados e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.2.6.;         5. O usuário informa os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados do horário, informa ao usuário, fecha a tela de cadastro e atualiza a lista com as alterações do horário.      3. **Consultar um horário de agenda (Ocorre somente quando o usuário não possui permissão para editar um horário de agenda)**         1. Usuário seleciona um horário de agenda e coloca para edição;         2. Sistema exibe a tela de consulta do horário de agenda para o usuário;         3. Usuário visualiza os dados do horário e seleciona sair.      4. **Excluir um horário de agenda**         1. Estando na tela de pesquisa de horário de agenda, o usuário seleciona o horário que deseja excluir;         2. O sistema exibe a tela de cadastro do horário para o usuário;         3. O usuário seleciona excluir;         4. Sistema mostra alerta de confirmação de exclusão para o usuário;         5. Se o usuário cancelar a exclusão, o sistema volta para a tela de pesquisa, caso contrário, segue o fluxo;         6. Sistema informa para o usuário, que o horário foi excluído;         7. O sistema volta para tela de pesquisa e a atualiza.  1. **Requisitos Especiais**   Nenhum requisito especial foi especificado para este caso de uso.   1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**   Não há pontos de extensão associados a este caso de uso. |

### Prontuário

A seguir é apresentado o detalhamento do caso de uso **UC024 – Anamnese.** Este caso de uso tem por objetivo demonstrar a maneira pela qual o usuário irá interagir com o histórico médico de um paciente.

|  |
| --- |
| 1. **Nome do Caso de Uso: UC024 - Anamnese**    1. **Descrição Resumida**   Este caso de uso tem como responsabilidade armazenar as informações do histórico de atendimentos que o paciente já realizou, além de informações históricas sobre a saúde do paciente e de seus familiares.   * 1. **Atores** * Profissional Especializado (Ex.: Médico); * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário seleciona a pesquisa de anamnese.   * + 1. **Pesquisar uma anamnese**        1. Usuário informa os dados do filtro e seleciona a pesquisa;        2. O sistema exibe uma lista com os pacientes que são atendidos por este filtro;        3. Usuário seleciona o paciente desejado;        4. O sistema exibe as informações da anamnese do paciente selecionado.   1. **Fluxo Alternativo**      1. **Criar uma anamnese**         1. Estando na tela de pesquisa, o usuário seleciona a opção de nova anamnese;         2. O sistema exibe a tela de cadastro de anamnese para o usuário;         3. O usuário informa o paciente;         4. O sistema carrega todos os atendimentos deste paciente para a anamnese;         5. O usuário informa os dados da anamnese e seleciona salvar;         6. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.1.8.;         7. O usuário informa os dados obrigatórios e seleciona salvar;         8. O sistema salva os dados da anamnese e informa ao usuário.      2. **Alterar uma anamnese**         1. Usuário seleciona uma anamnese na pesquisa e coloca para edição;         2. O sistema exibe a tela de cadastro de anamnese para o usuário;         3. O usuário informa ou altera os dados da anamnese e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.2.6.;         5. O usuário informa os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados da anamnese e informa ao usuário.      3. **Consultar uma anamnese (Ocorre somente quando o usuário não possui permissão para editar uma anamnese)**         1. Usuário seleciona uma anamnese da pesquisa e coloca para edição;         2. Sistema exibe a tela de consulta da anamnese para o usuário;         3. Usuário visualiza os dados da anamnese e seleciona sair.      4. **Excluir uma anamnese**         1. Usuário seleciona uma anamnese da pesquisa e coloca para edição;         2. O sistema exibe a tela de cadastro da anamnese para o usuário;         3. O usuário seleciona excluir;         4. Sistema mostra alerta de confirmação de exclusão para o usuário;         5. Se o usuário cancelar a exclusão, o sistema volta para a tela de cadastro, caso contrário, segue o fluxo;         6. Sistema informa para o usuário, que a anamnese foi excluída;         7. O sistema volta para tela de pesquisa e a atualiza.  1. **Requisitos Especiais**   Não há requisitos especiais associados a este caso de uso.   1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**   Não há pontos de extensão associados a este caso de uso. |

O caso de uso **UC025 - Atendimento** tem por objetivo demonstrar a maneira pela qual o usuário irá interagir com o cadastro de atendimento de um paciente.

|  |
| --- |
| 1. **Nome do Caso de Uso: UC025 - Atendimento**    1. **Descrição Resumida**   Este caso de uso tem como responsabilidade armazenar as informações da consulta que o paciente realizou em um determinado dia, dos procedimentos realizados e dos diagnósticos e tratamentos identificados pelo médico.   * 1. **Atores** * Profissional Especializado (Ex.: Médico); * Administrador; * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário seleciona a pesquisa de atendimentos.   * + 1. **Pesquisar um agendamento**        1. Usuário seleciona uma clínica na lista que aparece em tela (Caso seja um usuário com acesso a mais de uma clínica, caso contrário aparece apenas à clínica deste usuário pré-selecionada);        2. Usuário pode selecionar um médico correspondente à clínica (Aparecem apenas os médicos vinculados aquela clínica);        3. Usuário pode selecionar um paciente;        4. Usuário pode selecionar um dia;        5. O sistema exibe os atendimentos realizados de acordo com o filtro informado.   1. **Fluxo Alternativo**   O usuário seleciona a opção realizar atendimento a partir de um agendamento **(UC022)**.   * + 1. **Criar um atendimento**        1. Estando na tela de agenda, o usuário seleciona a opção realizar atendimento;        2. O sistema exibe a tela de cadastro de atendimento para o usuário;        3. O usuário informa os procedimentos realizados no paciente, além dos dados de diagnósticos e tratamentos e seleciona salvar;        4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.1.6.;        5. O usuário informa os dados obrigatórios e seleciona salvar;        6. O sistema salva os dados do atendimento, informa ao usuário.     2. **Alterar um atendimento**        1. Estando na tela de agenda, o usuário seleciona a opção realizar atendimento;        2. O sistema exibe a tela de cadastro de atendimento para o usuário;        3. O usuário informa ou altera os procedimentos realizados no paciente, além dos dados de diagnósticos e tratamentos e seleciona salvar;        4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.2.6.;        5. O usuário informa os dados obrigatórios e seleciona salvar;        6. O sistema salva os dados do atendimento, informa ao usuário.     3. **Consultar um atendimento (Ocorre somente quando o usuário não possui permissão para editar um atendimento)**        1. Usuário seleciona um atendimento da pesquisa de atendimentos e coloca para edição;        2. Sistema exibe a tela de consulta do atendimento para o usuário;        3. Usuário visualiza os dados do atendimento e seleciona sair.     4. **Excluir um atendimento**        1. Usuário seleciona um atendimento da pesquisa de atendimentos e coloca para edição;        2. O sistema exibe a tela de cadastro do atendimento para o usuário;        3. O usuário seleciona excluir;        4. Sistema mostra alerta de confirmação de exclusão para o usuário;        5. Se o usuário cancelar a exclusão, o sistema volta para a tela de cadastro, caso contrário, segue o fluxo;        6. Sistema informa para o usuário, que o atendimento foi excluído ou que não é possível excluir um atendimento que possua um ou mais procedimentos associados;        7. O sistema volta para tela de pesquisa e a atualiza.  1. **Requisitos Especiais**   O usuário pode editar o atendimento a partir de dois pontos específicos:   * Tela de Pesquisa de Atendimentos; e * Tela de Cadastro de Agendamentos (Somente em edição).  1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**   Não há pontos de extensão associados a este caso de uso. |

### Estoque

A seguir é apresentado o detalhamento do caso de uso **UC034 – Insumos.** Este caso de uso tem por objetivo demonstrar a maneira pela qual o usuário irá interagir com o cadastro de insumos;

|  |
| --- |
| 1. **Nome do Caso de Uso: UC034 - Cadastro de Insumos**    1. **Descrição Resumida**   Este caso de uso descreve e controla operações para Criar, Alterar, Consultar e Excluir Insumos.   * 1. **Atores** * Secretária; * Administrativo; * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário seleciona a pesquisa de insumos.   * + 1. **Pesquisar um insumo**        1. Usuário informa os dados do filtro e seleciona a pesquisa;        2. O sistema exibe uma lista com os insumos que são atendidos por este filtro;        3. Usuário seleciona o insumo desejado;        4. O sistema exibe as informações do insumo.   1. **Fluxo Alternativo**      1. **Criar um insumo**         1. Estando na tela de pesquisa, o usuário seleciona a opção de novo insumo;         2. O sistema exibe a tela de cadastro de insumos para o usuário;         3. O usuário informa os dados do insumo e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.1.6.;         5. O usuário informa os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados do insumo e informa ao usuário.      2. **Alterar um insumo**         1. Usuário seleciona um insumo na pesquisa e coloca para edição;         2. Sistema exibe a tela de cadastros para o usuário;         3. Usuário altera os dados do insumo e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.2.5.;         5. O sistema salva os dados do insumo e informa ao usuário.      3. **Consultar um insumo (Ocorre somente quando o usuário não possui permissão para editar um insumo)**         1. Usuário seleciona um insumo na pesquisa e coloca para edição;         2. Sistema exibe a tela de consulta do insumo para o usuário;         3. Usuário visualiza os dados do insumo e seleciona sair.      4. **Excluir um insumo**         1. Usuário seleciona um insumo da pesquisa e marca para exclusão;         2. Sistema mostra alerta de confirmação de exclusão para o usuário;         3. Se o usuário cancelar a exclusão, o sistema volta para a tela de pesquisa, caso contrário, segue o fluxo;         4. Sistema informa para o usuário, que o insumo foi excluído ou que não é possível excluir um insumo que possua ligações com outros cadastros;         5. Usuário volta para tela de pesquisa de insumos.  1. **Requisitos Especiais**   Não há requisitos especiais para este caso de uso.   1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**   Nenhum. |

O caso de uso **UC035 – Lançamentos de Insumos** tem por objetivo demonstrar a maneira pela qual o usuário irá interagir com o lançamento de insumos no sistema.

|  |
| --- |
| 1. **Nome do Caso de Uso: UC035 - Lançamento de Insumos**    1. **Descrição Resumida**   Este caso de uso descreve e controla operações para Criar e Consultar lançamentos de insumos.   * 1. **Atores** * Secretária; * Administrativo; * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário seleciona o cadastro de lançamento de insumos.   * + 1. **Pesquisar lançamentos de insumo**        1. Usuário informa os dados do filtro e seleciona a pesquisa;        2. O sistema exibe uma lista com os lançamentos de insumos que são atendidos por este filtro;        3. Usuário seleciona o lançamento desejado;        4. O sistema exibe as informações do lançamento.   1. **Fluxo Alternativo**      1. **Criar um lançamento de insumo**         1. Estando na tela de pesquisa, o usuário seleciona a opção de novo lançamento;         2. Usuário informa o insumo desejado para lançamento, o tipo de lançamento e a quantidade;         3. Usuário seleciona salvar;         4. Se algum item obrigatório não for informado o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.1.6;         5. O usuário informa os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados do lançamento do insumo e informa ao usuário.      2. **Consultar um lançamento de insumo**         1. Usuário seleciona um lançamento na pesquisa e coloca para consulta;         2. Sistema exibe a tela de consulta do lançamento para o usuário;         3. Usuário visualiza os dados do lançamento e seleciona sair.  1. **Requisitos Especiais**  * Não existe alteração de lançamentos, o usuário deve estornar o lançamento, gerando assim outro lançamento automaticamente; * Todos os lançamentos alteram a quantidade total daquele insumo.  1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**   Um lançamento automaticamente altera a quantidade total de um insumo. |

### Financeiro

A seguir é apresentado o detalhamento do caso de uso **UC039 – Fluxo de Caixa.** Este caso de uso tem por objetivo demonstrar a maneira pela qual o usuário irá movimentar o fluxo de caixa da clínica.

|  |
| --- |
| 1. **Nome do Caso de Uso: UC039 – Fluxo de Caixa**    1. **Descrição Resumida**   Este caso de uso descreve e controla operações para movimentação (fluxo) de um caixa.   * 1. **Atores** * Secretária; * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário seleciona o caixa.   * + 1. **Pesquisar um fluxo de caixa**        1. Usuário informa a clínica, o caixa e o dia e seleciona a pesquisa;        2. O sistema exibe uma lista com os movimentos que são atendidos por estes filtros;        3. Usuário pode efetuar quaisquer ações de movimentação no caixa.   1. **Fluxo Alternativo**      1. **Efetuar abertura de caixa**         1. Estando na tela de fluxo de caixa, o usuário seleciona o caixa desejado e a opção de abertura de caixa;         2. O sistema exibe a tela de abertura de caixa para o usuário;         3. O usuário informa o valor que o caixa irá iniciar e seleciona salvar;         4. Se o valor não for informado, o sistema automaticamente abrirá o caixa com o valor zero;         5. O sistema salva os dados de abertura e informa ao usuário.      2. **Efetuar retirada (sangria) no caixa**         1. Estando na tela de fluxo de caixa, o usuário seleciona o caixa desejado e a opção de retirada;         2. O sistema exibe a tela de retirada para o usuário;         3. O usuário informa o valor que irá retirar do caixa, o motivo e seleciona salvar;         4. Se o valor ou o motivo não forem informados, o sistema avisa o usuário e não permite o movimento.         5. O usuário informa todos os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados de retirada e informa ao usuário.      3. **Efetuar depósito no caixa**         1. Estando na tela de fluxo de caixa, o usuário seleciona o caixa desejado e a opção de depósito;         2. O sistema exibe a tela de depósito para o usuário;         3. O usuário informa o valor que irá depositar no caixa, o motivo e seleciona salvar;         4. Se o valor ou o motivo não forem informados, o sistema avisa o usuário e não permite o movimento.         5. O usuário informa todos os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados de depósito e informa ao usuário.      4. **Efetuar estorno no caixa**         1. Estando na tela de fluxo de caixa, o usuário seleciona o caixa desejado, o registro e a opção de estorno;         2. O sistema pede a confirmação de estorno daquele registro para o usuário;         3. O usuário confirma o estorno;         4. O sistema efetua o estorno daquele registro lançando um novo registro e informa ao usuário.      5. **Efetuar fechamento de caixa**         1. Estando na tela de fluxo de caixa, o usuário seleciona o caixa desejado e a opção de fechamento de caixa;         2. O sistema pede a confirmação de fechamento de caixa para o usuário;         3. O usuário confirma o fechamento;         4. O sistema efetua o fechamento do caixa e informa ao usuário.  1. **Requisitos Especiais**  * O caixa só pode ser aberto se estiver fechado; * O caixa só pode ser fechado se estiver aberto; * Não é permitido movimentações com o caixa fechado; * O caixa pode ser aberto e fechado quantas vezes forem necessárias no decorrer de um dia; * Por padrão é permitido o caixa ficar negativo, mas o sistema deve permitir parametrizar esta opção no cadastro de caixas; * Ao efetuar a abertura de um caixa com o valor maior que zero, é automaticamente efetuado um lançamento de depósito com o motivo de abertura de caixa; * Ao efetuar o fechamento de um caixa com valor superior a zero, é automaticamente efetuado um lançamento de retirada com o motivo de fechamento de caixa.  1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**  * O fluxo de caixa possui ligação direta com as contas a pagar e contas a receber; * Todas as contas a pagar ou a receber que estiverem vinculadas a um caixa e forem confirmadas, são automaticamente lançadas neste caixa; * O pagamento de uma consulta gera um lançamento no fluxo de caixa, desde que seja um pagamento no ato. |

O caso de uso **UC040 – Contas a Pagar e Contas a Receber** tem por objetivo demonstrar a maneira pela qual o usuário irá pesquisar e criar contas a pagar e a receber para uma clínica.

|  |
| --- |
| 1. **Nome do Caso de Uso: UC040 - Contas a pagar e a receber**    1. **Descrição Resumida**   Este caso de uso descreve e controla operações para Criar, Alterar, Consultar e Excluir Contas a pagar e a receber.   * 1. **Atores** * Administrativo; * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário seleciona a pesquisa de contas a pagar e a receber.   * + 1. **Pesquisar uma conta a pagar ou a receber**        1. Usuário informa os dados do filtro e seleciona a pesquisa;        2. O sistema exibe uma lista com as contas que são atendidas por este filtro;        3. Usuário seleciona a conta desejada;        4. O sistema exibe as informações da conta.   1. **Fluxo Alternativo**      1. **Criar uma conta a pagar ou a receber**         1. Estando na tela de pesquisa, o usuário seleciona a opção de nova conta;         2. O sistema exibe a tela de cadastro de contas para o usuário;         3. O usuário informa os dados da conta, o tipo da conta (Pagar ou Receber) e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.1.6.;         5. O usuário informa os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados da conta e informa ao usuário.      2. **Alterar uma conta a pagar ou a receber**         1. Usuário seleciona uma conta na pesquisa e coloca para edição;         2. Sistema exibe a tela de cadastro para o usuário;         3. Usuário altera os dados da conta e seleciona salvar;         4. Se algum item obrigatório do cadastro não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cadastro, caso contrário segue o fluxo a partir de 2.2.2.6.;         5. O usuário informa os dados obrigatórios e seleciona salvar;         6. O sistema salva os dados da conta e informa ao usuário.      3. **Consultar uma conta a pagar ou a receber (Ocorre somente quando o usuário não possui permissão para editar)**         1. Usuário seleciona uma conta na pesquisa e coloca para edição;         2. Sistema exibe a tela de consulta da conta para o usuário;         3. Usuário visualiza os dados da conta e seleciona sair.      4. **Excluir uma conta a pagar ou a receber**         1. Usuário seleciona uma conta da pesquisa e marca para exclusão;         2. Sistema mostra alerta de confirmação de exclusão para o usuário;         3. Se o usuário cancelar a exclusão, o sistema volta para a tela de pesquisa, caso contrário, segue o fluxo;         4. Sistema informa para o usuário, que a conta foi excluída ou que não é possível excluir uma conta já paga ou recebida;         5. Usuário volta para tela de pesquisa de contas.      5. **Lançamento de uma conta a pagar ou a receber**         1. Usuário seleciona uma conta da pesquisa e marca para lançamento;         2. Sistema mostra alerta de confirmação de lançamento para o usuário;         3. Se o usuário cancelar o lançamento, o sistema volta para a tela de pesquisa, caso contrário, segue o fluxo;         4. Sistema informa para o usuário, que a conta foi lançada e que gerou um movimento de caixa;         5. Usuário volta para tela de pesquisa de contas.  1. **Requisitos Especiais**  * Só é permitida a alteração de uma conta enquanto ela não for paga ou recebida, após isso o sistema permite apenas o estorno gerando uma nova conta com as mesmas informações da conta estornada; * Só é permitida a exclusão de uma conta que não tenha sido paga ou recebida;  1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**  * Todo lançamento de contas a pagar ou a receber, automaticamente gera um movimento no fluxo do caixa daquela conta. |

O caso de uso **UC041 – Pagamento de Consultas** tem por objetivo demonstrar a maneira pela qual o usuário conseguirá efetivar o pagamento de um atendimento/consulta de uma clínica.

|  |
| --- |
| 1. **Nome do Caso de Uso: UC041 – Pagamento de Consultas**    1. **Descrição Resumida**   Este caso de uso descreve e controla operações para efetuar e cancelar o pagamento de consultas, independente se for pagamento à vista ou parcelado.   * 1. **Atores** * Secretária; * Todos os atores são chamados de usuários no decorrer do caso de uso.   1. **Acionadores**   Usuário seleciona operações explicitamente usando a interface da tela.   1. **Fluxo de Eventos**    1. **Fluxo Básico**   O usuário seleciona a agenda para um ou mais médicos.   * + 1. **Pesquisar uma consulta**        1. Usuário seleciona uma clínica na lista que aparece em tela (Caso seja um usuário com acesso a mais de uma clínica, caso contrário aparece apenas à clínica deste usuário pré-selecionada);        2. Usuário seleciona um ou mais médicos correspondentes à clínica (Aparecem apenas os médicos vinculados aquela clínica);        3. O sistema exibe as agendas dos médicos selecionados identificando cada um com a cor definida para este no seu cadastro.   1. **Fluxo Alternativo**      1. **Efetuar Pagamento de Consulta**         1. Estando na tela de agenda, o usuário seleciona a consulta que deseja efetuar o pagamento;         2. O sistema exibe a tela de agendamento da consulta para o usuário;         3. O usuário seleciona a opção de efetuar pagamento (esta opção só aparece quando existe um paciente cadastrado vinculado a consulta);         4. O sistema abre uma janela de pagamento de consultas;         5. O usuário informa os dados de pagamento e seleciona salvar;         6. Se algum item obrigatório do pagamento não for informado, o sistema informa ao usuário estes itens e retornará para a tela de pagamento, caso contrário segue o fluxo a partir de 2.2.1.8.;         7. O usuário informa os dados obrigatórios e seleciona salvar;         8. O sistema salva os dados do pagamento, efetua o lançamento no fluxo de caixa e informa ao usuário.      2. **Cancelar Pagamento de Consulta**         1. Estando na tela de agenda, o usuário seleciona a consulta que deseja cancelar o pagamento;         2. O sistema exibe a tela de agendamento da consulta para o usuário;         3. O usuário seleciona a opção de cancelar pagamento (esta opção só aparece quando existe um pagamento vinculado a consulta);         4. O sistema abre uma janela de cancelamento do pagamento de consultas;         5. O usuário informa o motivo do cancelamento e seleciona salvar;         6. Se algum item obrigatório do cancelamento não for informado, o sistema informa ao usuário estes itens e retornará para a tela de cancelamento, caso contrário segue o fluxo a partir de 2.2.2.8.;         7. O usuário informa os dados obrigatórios e seleciona salvar;         8. O sistema salva os dados do cancelamento, efetua o lançamento de estorno no fluxo de caixa e informa ao usuário.      3. **Consultar um pagamento de consulta**         1. A qualquer momento o usuário pode consultar o pagamento de uma consulta através da pesquisa de atendimentos.      4. **Excluir um pagamento de consulta**         1. Não é permitida a exclusão de um pagamento.         2. A opção mais similar a exclusão é o cancelamento do pagamento.  1. **Requisitos Especiais**   Nenhum.   1. **Pré-condições**    1. **Entrar no sistema**  * Antes do caso de uso começar o usuário entrou no sistema; * Ter permissão para realizar o caso de uso.  1. **Pós-condições**   Não há pós-condições associadas a este caso de uso.   1. **Pontos de extensão**  * Todo pagamento gera lançamentos no fluxo de caixa e/ou contas a receber; * Todo cancelamento gera lançamentos de estorno no fluxo de caixa e/ou exclusão do registro de contas a receber. |

Através da análise de requisitos realizada neste capítulo, juntamente com a análise dos sistemas similares, foi possível elaborar o projeto deste sistema de gestão para clínicas médicas de pequeno porte. Para completar o projeto proposto, foram elaborados protótipos de telas do sistema, o que pode ser visto no capítulo 5.

1. PROTÓTIPO DO SISTEMA

Este capítulo tem como objetivo abordar as principais características do protótipo do sistema proposto no presente trabalho. Houve uma atenção especial com o layout das telas, a fim de torná-las amigáveis e de uso intuitivo, facilitando o entrosamento do usuário com o sistema.

Uma das características presente nas telas e que merecem destaque são as caixas de campos, ou seja, as informações são disponibilizadas em conjuntos de campos relacionados. Estas caixas podem ser individualmente minimizadas ou maximizadas segundo a necessidade do usuário, facilitando o acesso e visualização às informações.

Alguns campos possuem máscaras, que evitam que o usuário ingresse informações fora do padrão esperado. Um exemplo são os campos de telefone e CPF, que permitem apenas números, não sendo possível informar letras ou caracteres. Finalmente, algumas informações podem ser escolhidas em caixas de seleção com opções de respostas pré determinadas, evitando que o usuário tenha que digitar a mesma informação diversas vezes.

As seções a seguir apresentam o protótipo de seis telas do sistema e, adicionalmente, descrevem seus principais objetivos e funcionalidades.

## Cadastro de Pacientes

O objetivo da tela cadastro de pacientes é permitir o registro das informações pessoais de cada paciente da clínica como, por exemplo, dados de identificação, contato e endereço. A figura 5.1 apresenta o protótipo desta tela.

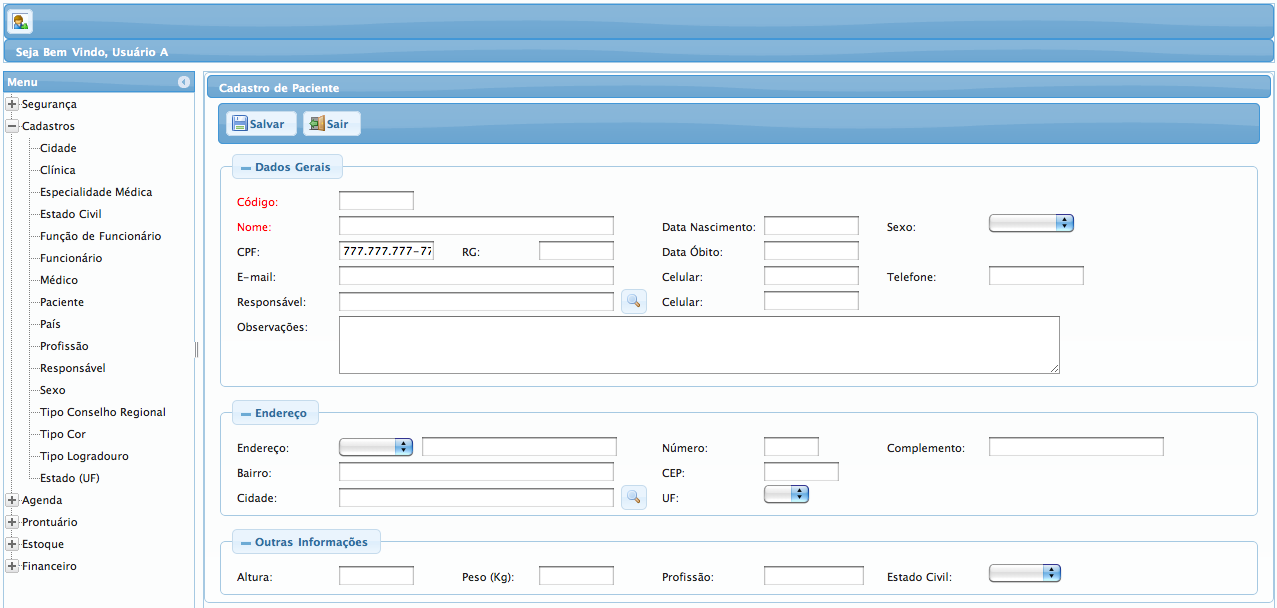


Figura 5.1 - Protótipo da Tela de Cadastro de Pacientes

Caso o paciente em questão seja menor de idade, há um campo que solicita o nome do responsável. Se este responsável já estiver registrado no sistema, o campo permite que seja feita uma busca e que as informações sejam vinculadas.

## Cadastro de Clínicas

Caso um mesmo cliente do sistema possua mais de uma sede física, é possível conectar os dados destas clínicas para que os administradores tenham uma visão holística das informações. A tela de cadastro de clínicas (figura 5.2) permite o registro destas diferentes sedes para que cada uma delas tenha operações individuais dentro do sistema.

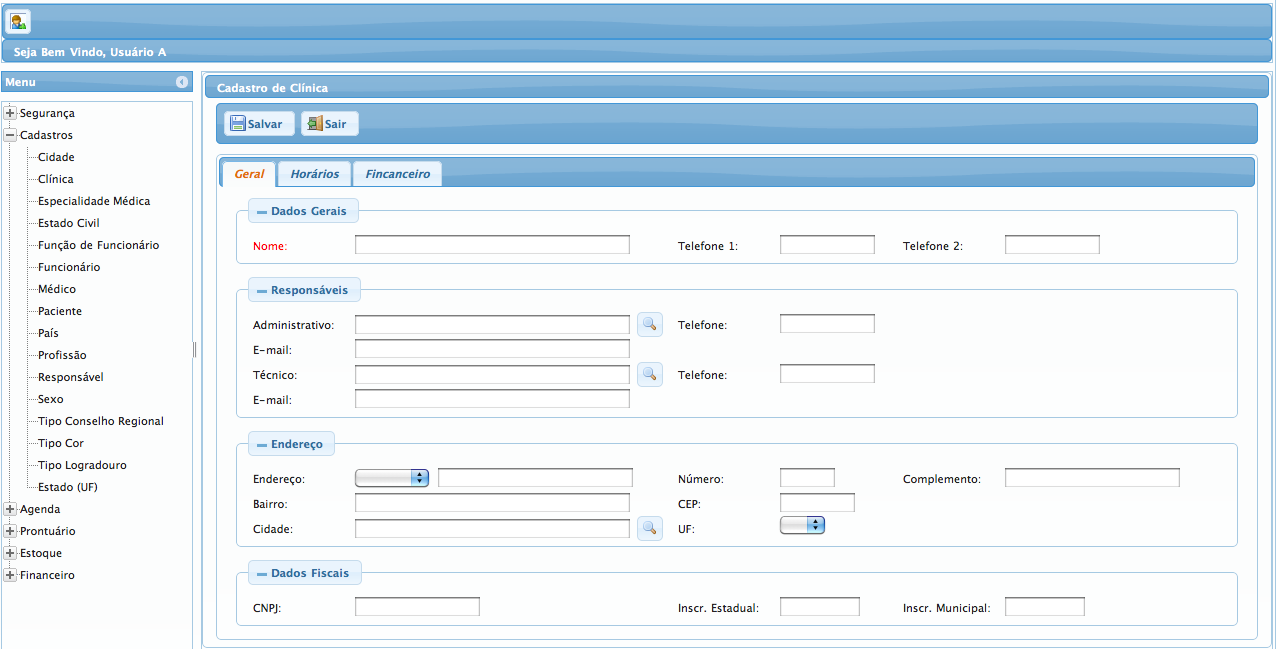


Figura 5.2 - Protótipo da Tela de Cadastro de Clínicas

Esta tela possui as abas Geral, Horários e Financeiro. Esta divisão de tópicos por aba foi aplicada para permitir que o usuário não tenha que utilizar a barra de rolagem para encontrar as informações que deseja e tenha, assim, uma visualização mais clara e dinâmica da tela.

Na aba geral estarão os dados de identificação, contato, responsáveis e endereço da clínica. Na aba horários será possível definir os dias e turnos de funcionamento da clínica em questão, sendo que esta informação será refletida nos horários disponíveis para agendamento de consultas. Por fim, na aba Financeiro são cadastrados os caixas da empresa, como conta corrente e caixa físico da recepção.

## Agenda

O objetivo da tela de agenda é permitir que os usuários registrem, consultem e editem as consultas marcadas para os médicos da clínica. A figura 5.3 apresenta o protótipo desta tela.

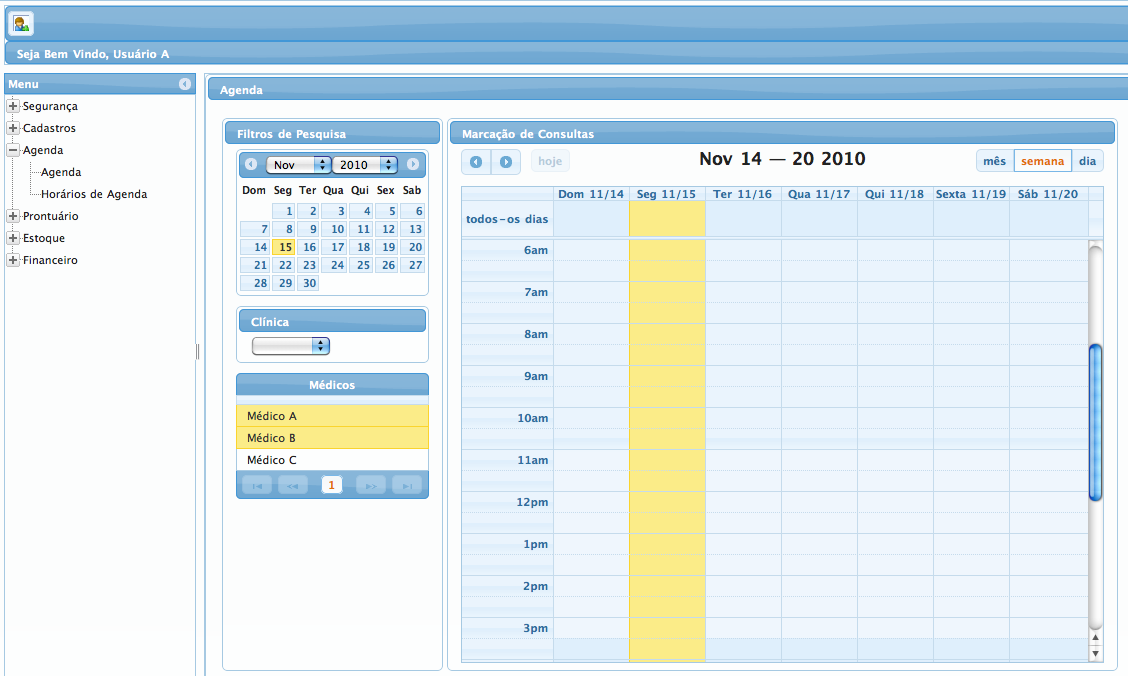


Figura 5.3 - Protótipo da Tela de Agenda

O *layout* desta tela possui o mesmo estilo do calendário do Google e da Agenda do Outlook. Assim, os usuários que já estão familiarizados com o formato destas outras ferramentas rapidamente entenderão o conceito de uso desta tela. É possível visualizar a agenda de um médico específico ou de diversos médicos simultaneamente, para determinar o modelo de visualização basta que o usuário selecione os nomes dos médicos na caixa à esquerda da agenda.

## Cadastro de Atendimento

Na tela de cadastro de atendimento (figura 5.4), os médicos poderão fazer o registro de cada procedimento, diagnóstico e tratamento relacionados a um atendimento. O objetivo é que os médicos formem um histórico de cada um de seus pacientes.

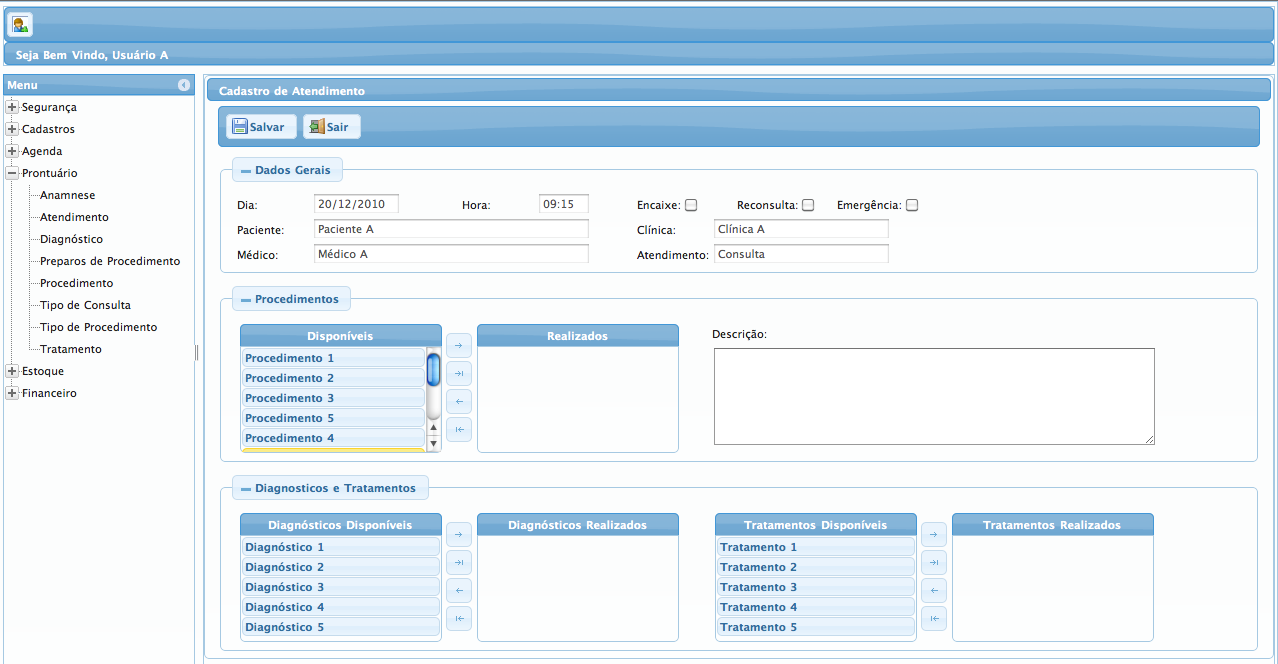


Figura 5.4 - Protótipo da Tela de Cadastro de Atendimentos

Os campos de procedimento, diagnóstico e tratamento trarão dados pré registrados que poderão ser selecionados para o atendimento em questão. A transferência dos dados das caixas de “disponíveis” para as caixas de “realizados” pode ser feita arrastando os componentes de uma caixa para a outra ou utilizando os botões em forma de setas. Há ainda um campo livre de descrição para que o médico faça anotações adicionais sobre o atendimento.

## Fluxo de Caixa

A tela de fluxo de caixa permite que os usuários visualizem e registrem as transações financeiras da clínica. Estes registros são salvos por períodos de um dia. A figura 5.5 apresenta o protótipo desta tela.

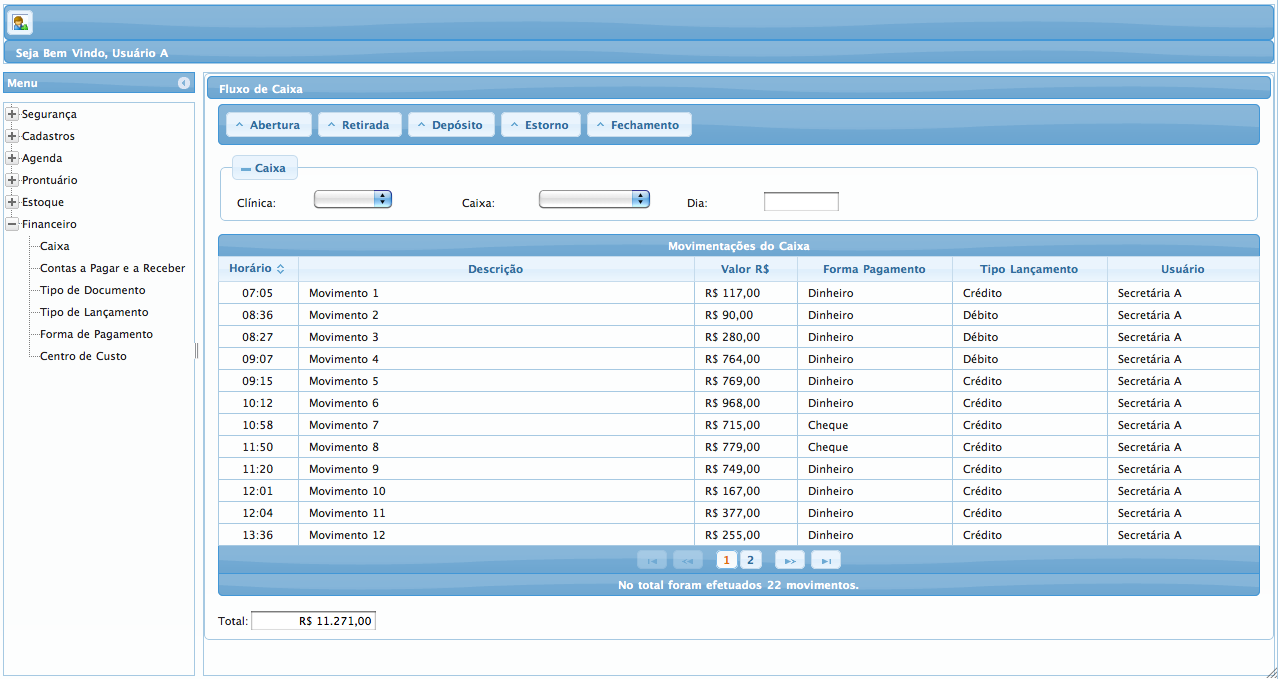


Figura 5.5 - Protótipo da Tela de Fluxo de Caixa

As movimentações financeiras serão exibidas em ordem cronológica, sendo que as informações mais recentes estarão no topo. Depois que uma entrada é salva, não pode ser excluída. Junto com a descrição da movimentação também serão registrados o horário da movimentação, o valor, a forma de pagamento, se a operação se trata de um débito ou crédito e o usuário responsável. Na parte inferior da tela haverá um campo informando o saldo do caixa.

## Contas a Pagar e Contas a Receber

Esta tela permitirá que os usuários façam o controle das contas a pagar e a receber da clínica, podendo consultar, incluir, editar e excluir entradas. A figura 5.6 apresenta o protótipo desta tela.

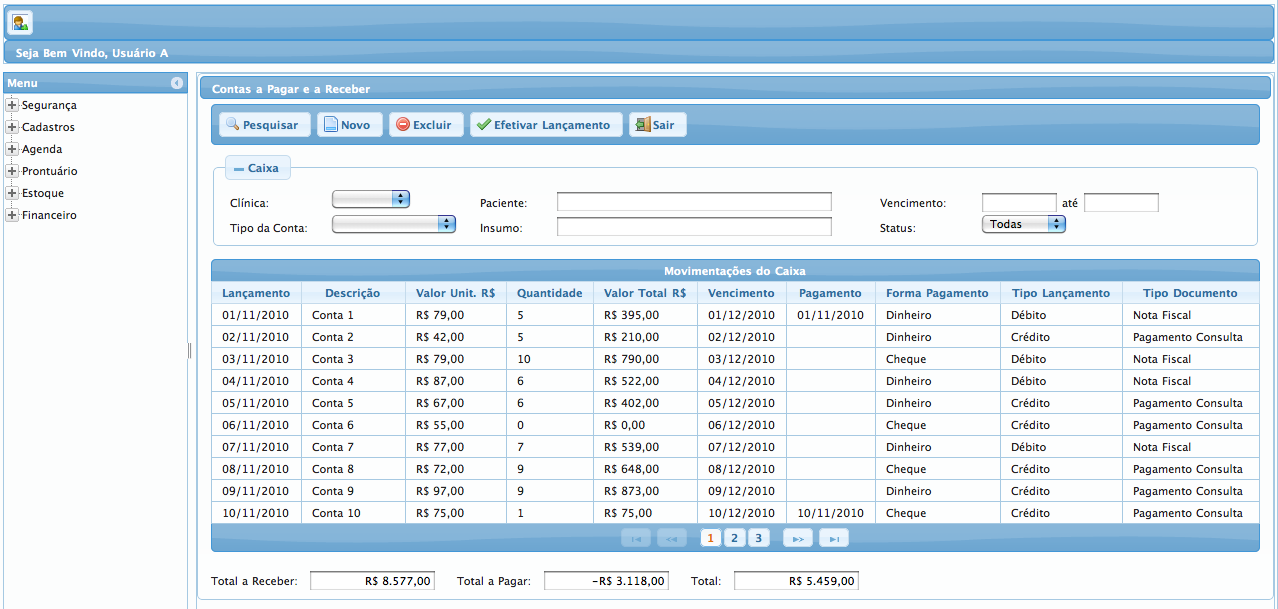


Figura 5.6 - Protótipo da Tela de Contas a Pagar e Contas a Receber

Na parte superior da tela existe uma série de filtros que o usuário poderá utilizar para buscar pelos registros de contas a pagar, contas a receber ou ambos. Algumas opções de filtro são o status das contas (em dia ou com atraso), data de vencimento e paciente.

Na parte central da tela são exibidas informações como descrição, valor, forma de pagamento e data de vencimento das transações que se enquadram na busca realizada pelo usuário. As informações são disponibilizadas em ordem cronológica e na parte inferior da tela o usuário poderá consultar os totais.

Através do protótipo das telas é possível confirmar o *layout* simples e amigável que o sistema possui. As facilidades que as telas propõem, auxiliam na agilidade das tarefas diárias dos funcionários de uma clínica de pequeno porte.

Fechando este capítulo, pode-se destacar a importância de prototipar um sistema, pois é possível visualizar de forma mais clara as opções e ferramentas que o mesmo terá.

CONCLUSÃO

É possível dizer que os ERP’s são os principais componentes dos sistemas de informação de empresas. Esta constatação reforça a importância de projetar e desenvolver sistemas de gestão para empresas não atendidas por ERPs tradicionais, como é o caso das clínicas médias de pequeno porte.

Para aumentar a produtividade e organizar o processo de desenvolvimento de ERPs , uma metodologia amplamente utilizada é o processo unificado. O processo unificado ajuda a guiar todo o projeto de desenvolvimento do sistema, suas fases e os ciclos que devem ser realizados no decorrer de cada etapa, contribuindo muito para o bom andamento das atividades.

Na análise de sistemas com recursos similares foi possível constatar que alguns dos mais populares softwares para clínicas médicas não são tão simples de usar quanto os usuários desejariam. Além disso, os softwares concorrentes não dão a atenção necessária às informações de extrema importância para a gestão da clínica como, por exemplo, um fluxo de caixa simples e completo. Desta forma, concluiu-se que o desenvolvimento de um software de gestão com interface simples e funcionalidades adequadas seriam um diferencial em um novo software para gestão de clínicas médicas.

A arquitetura Java, comparada a outras linguagens de programação amplamente utilizadas, possui como principais benefícios sua robustez, sua grande quantidade de *frameworks* e sua orientação a objetos. Java foi a linguagem escolhida para o desenvolvimento do sistema em questão, pois tal tecnologia traz mais agilidade e produtividade, além de ajudar a aumentar a confiabilidade da arquitetura.

Com base em todos os estudos e análises realizados e de um detalhado levantamento de requisitos, foi realizada toda a análise, projeto e prototipação do sistema proposto. Os resultados obtidos até o momento permitem afirmar que o sistema de gestão de clínicas médicas de pequeno e médio porte irá contribuir em muito para o cotidiano destas clínicas. Irá facilitar o cotidiano destas clínicas em tarefas já sistematizadas atualmente, como o agendamento de consultas, registro dos atendimentos, anamnese, entre outras tarefas. Mas, como maior contribuição, irá sistematizar, facilitar e organizar processos hoje não informatizados, como controle de insumos, fluxo de caixa e contas a pagar e receber.

Como trabalhos futuros, destacam-se a conclusão do desenvolvimento do sistema proposto e a implantação em uma primeira clínica médica, visando concluir a validação do sistema para futura comercialização

REFERÊNCIAS

AUDY, Jorge; PRIKLADNICKI, Rafael. **Desenvolvimento distribuído de software**. São Paulo: Campus, 2007.

BISWAS, Rahul; ORT, Ed. The **Java Persistence API - A Simpler Programming Model for Entity Persistence.** Disponível em: <<http://www.oracle.com/technetwork/articles/javaee/jpa-137156.html>>. Acesso em 07/11/2010.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: Guia do Usuário**. São Paulo: Campus, 2000.

COCKBURN, Alistair. **Escrevendo Casos de Uso Eficazes: Um guia prático para desenvolvedores de software**. Porto Alegre: Bookman, 2005.

DATE, C.J., **Introdução a Sistemas de Bancos de Dados.** Tradução da 8 edição Americana. Rio de Janeiro: Campus, 2004.

DAVENPORT, Thomas H. **Missão Critica: Obtendo vantagem competitiva com os sistemas de gestão empresarial.** Porto Alegre: Bookman, 2002.

FRANKLINT, Kleitor. **Java EE 5, Java Platform Enterprise Edition 5, Guia Prático.** São Paulo: Editora Érica, 2006.

FREIRE, Sergio Miranda. **Sigilo das Informações.** Disponível em: <<http://www.ans.gov.br/data/files/8A958865266CAFE201267FB9CDED2DB2/TT_AS_19_SMirandaFreire_SigiloInformacoes.pdf>>. Acesso em 27/12/2010.

GERCLIM. **Sistema de gestão médica.** Disponível em: <<http://www.gerclim.com.br/>>. Acesso em: 12 maio 2010.

GONÇALVES, Edson. **Desenvolvendo Aplicações Web com JSP, Servlets, Java Server Faces, HIbernate, EJB 3 Persistence e Ajax.** Rio de Janeiro: Editora Ciência Moderna, 2007.

HORSTMANN, Cay S.; CORNELL, Gary. **Core Java, Volume I – Fundamentos.** Rio de Janeiro: Ed. Alta Books, 2005.

JACOBI, Jonas; FALLOWS, John R. **Pro JSF and Ajax – Building Rich Internet Components.** Apress, 2006.

MDMED. **Sistema de gestão médica.** Disponível em: <<http://www.mdmed.com.br/>>. Acesso em: 13 maio 2010.

PADILHA, Thais Cássia Cabral; MARINS, Fernando Augusto Silva. Sistemas **ERP: características, custos e tendências.** Prod. [online]. 2005, vol.15, n.1, pp. 102-113. ISSN 0103-6513. Disponível em <http://www.scielo.br/scielo.php?pid=S0103-65132005000100009&script=sci\_arttext&tlng=es>. Acesso em 13/04/2010

PANDA, Debu; RAHMAN, Reza; LANE; Derek**. EJB 3 in Action**. Manning, 2007.

PFLEEGER, Shari Lawrence. **Engenharia de Software – Teoria e Prática.** 2ª ed. São Paulo: Prentice Hall, 2004.

PRESSMAN, Roger S. Software Engineering: a practitioner’s approach, 6 edição, Mc Graw Hill, 2004.

PRODOCTOR. **Sistema de gestão médica.** Disponível em: <<http://www.prodoctor.net/>>. Acesso em: 14 maio 2010.

SCOTT, Kendall. **O Processo Unificado Explicado.** 1ª ed. Porto Alegre: Bookman, 2003.

SOUZA, Cesar Alexandre de; SACCOL, Amarolinda Zanela. **Sistemas ERP no Brasil.** São Paulo: Atlas, 2008.

THOMPSON, Marco Aurélio. **Java 2 & Banco de Dados.** 3 edição. São Paulo: Ed. Érica Ltda; 2005

VALARIE A. Zeithaml; Mary Jo BITNER. **Marketing de Serviços: a empresa com foco no cliente.** Tradução Martin Albert Haag e Carlos Alberto Silveira Netto Soares – 2ª edição – Porto Alegre: Bookman, 2008.