FACULDADE DE COMPUTAÇÃO DE MONTES CLAROS
COORDENADORIA DE ENSINO
CURSO DE SISTEMAS DE INFORMAÇÃO

Alano Pereira David

DESENVOLVIMENTO DE UM SISTEMA DE INFORMAÇÃO PARA AUTOMATIZAÇÃO DOS PROCESSOS DE UMA CLÍNICA MÉDICA UTILIZANDO RECURSOS DE REUSO DE SOFTWARE

Montes Claros/MG
2010

Alano Pereira David

DESENVOLVIMENTO DE UM SISTEMA DE INFORMAÇÃO PARA AUTOMATIZAÇÃO DOS PROCESSOS DE UMA CLÍNICA MÉDICA UTILIZANDO RECURSOS DE REUSO DE SOFTWARE

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do Diploma de Bacharel em Sistemas de Informação, da Faculdade de Computação de Montes Claros.

ORIENTADOR: Luciano Soares De Souza

Montes Claros/MG

2010

Alano Pereira David

DESENVOLVIMENTO DE UM SISTEMA DE INFORMAÇÃO PARA AUTOMATIZAÇÃO DOS PROCESSOS DE UMA CLÍNICA MÉDICA UTILIZANDO RECURSOS DE REUSO DE SOFTWARE

Trabalho de Conclusão de Curso aprovado como requisito parcial para obtenção do Diploma de Graduação em Sistemas de Informação, da Faculdade de Computação de Montes Claros.

Aprovado em: 26/11/2010

BANCA EXAMINADORA:

Ass. Luciano Soares de Souza

_____

1º Exam.: Nome – Titulação - Instituição

Ass._____

2º Exam.: Douglas Andrey– Titulação - Facomp

Ass._____

3º Exam.: Christine Matos – Titulação – Facomp

Dedico este trabalho a minha família,
aos meus amigos e a minha namorada.

AGRADECIMENTOS

Nossas vidas são repletas de objetivos e sonhos que nunca conseguimos realizar sozinhos. Existem problemas e barreiras que, através deles surgem também amigos que nos incentivam em nossos sonhos e que de certa forma sempre nos ajudam direta e indiretamente em nossas vidas.

Agradeço ao grande Deus que me possibilitou a vida, a força e acima de tudo muita saúde e por me mostrar que sempre temos que ter força e paciência para superar os obstáculos que nos aparecem.

Agradeço ao meu orientador Luciano Soares, pelo incentivo e ajuda no desenvolvimento deste trabalho.

Agradeço minha família pelo incentivo na vida, em especial ao meu pai e minha mãe que sempre me incentivaram a estudar mais e mais, para conseguir mudar meu futuro e aos irmãos de república Célio, Sérgio e Luiz, além da minha cunhada Ludmilla.

Agradeço a Arlete Mesquita por ter me acolhido no emprego e dado força para continuar meus estudos, William Borges e Gabrielle Mourão pela amizade e auxílio na construção deste projeto, aos amigos da faculdade Daniel Pereira, Wanderson Souza, Letícia Kelly e todos os demais que de certa forma, contribuíram para meu crescimento.

Agradeço a minha namorada Verlane Gonçalves Santos pelo carinho e compreensão dos meus problemas e que se manteve forte comigo no momento de dificuldade, e aos amigos do Colégio Integral que proporcionaram conhecimento e responsabilidade para tratar dos problemas do dia-a-dia.

RESUMO

O reuso de *Software* é uma prática comum de desenvolvimento, além de se obter a redução dos custos do *Software* e aumento na qualidade em um curto espaço de tempo, o reuso de *Software* traz outros benefícios na hora de encontrar soluções mais rápidas para problemas comuns, antes já resolvidos. A construção de um *Software*, utilizando de técnicas de reuso de *Software*, é uma maneira prática de se obter resultados favoráveis para um cliente, no quesito tempo de entrega e custos do *Software*. Portanto, neste trabalho foi desenvolvido um sistema virtual de uma clínica, com o auxílio do *framework VirtualClass*, onde pôde-se construir o sistema da clínica fazendo reutilização (e geração), a partir do *VirtualClass,* observou-se que o reuso de software pode ser utilizado para o auxílio no desenvolvimento de novos software às quais se agregam funcionalidades cada vez mais importantes no seu diário.

Palavra-Chave: reuso, VirtualClass, *framework.*

ABSTRACT

Reuse of *Software* is a common practice of development, besides of the decreased costs and increased *Software* quality in a short time, the reuse of *Software* has otherbenefits in the time needed to find faster solutions to common problems, already solved before. The construction of a *Software*, using techniques of *Software* reuse is a practical way to obtain favorable results for a client, on item delivery time and *Software* costs. Therefore, this work done the development of a virtual system of a clinic with the help of the *framework VirtualClass* where we could build the system by making reuse the clinic (and generation) from the *VirtualClass*, where we could build the system by making reuse the clinic (and generation) from VirtualClass and reuse of software can be used to aid in the development of new software which adds features become increasingly important in his diary.

Keyword: reuse, *VirtualClass*, *framework*

LISTA DE FIGURAS

Figura 1: Estrutura do *Framework* VirtualClass 18
Figura 2: Interface Manter Clientes 24
Figura 3: Interface Manter Exames 25
Figura 4: Interface Gerar Relatórios 25
Figura 5: Interface Agendar Consultas 26
Figura 6: Interface Manter Profissional 26
Figura 7: Interface Solicitar Exames 26
Figura 8: Interface Manter Convênio 27
Figura 9: Interface Manter Especialidades 27
Figura 10: Interface Vínculo de Dependentes 27
Figura 11: Interface Vínculo de Perfil 27
Figura 12: Interface Manter Usuário 28
Figura 13: Diagrama de Classe do VirtualClass Clínica 29

LISTA DE QUADROS

Quadro 1: Benefícios do Reuso de *Software* 15
Quadro 2: Problemas com reuso 16
Quadro 3: Funções do Sistema 29
Quadro 4: Requisito Manter Clientes 50
Quadro 5: Requisito Manter Exames 51
Quadro 6: Requisito Gerar Relatórios 52
Quadro 7: Requisito Agendar Consultas 52
Quadro 8: Requisito Manter Profissional 53
Quadro 9: Requisito Solicitar exames 54
Quadro 10: Requisito Manter Convênio 55
Quadro 11: Requisito Manter Especialidades 56
Quadro 12: Requisito Manter Dependente 56
Quadro 13: Requisito Manter Usuários 57

SUMÁRIO

1- INTRODUÇÃO **12**
2- REUSO DE SOFTWARE **14**
2.1- *Framework* **14**
2.2- Vantagens **15**
2.3- Desvantagens **16**
3- *FRAMEWORK* VIRTUALCLASS **18**
3.1- PORTALMAKER **19**
3.2- TECNOLOGIAS ENVOLVIDAS **19**
3.2.1- AJAX 20
3.2.2- XML 20
3.2.3-JAVASCRIPT 21
3.2.4- HTML E XHTML 22
3.2.5- JAVA 23
4- DESENVOLVIMENTO DE UM SISTEMA DE CLÍNICA **24**
4.1- REQUISITOS DO SISTEMA **24**
4.2- PROTÓTIPOS DE INTERFACE **24**
4.2.1- Protótipo de Interface Manter Clientes 24
4.2.2- Protótipo de Interface Manter Exames 25
4.2.3- Protótipo de Interface Gerar Relatórios 25
4.2.4- Protótipo de Interface Agendar Consultas 25
4.2.5- Protótipo de Interface Manter Profissional 26
4.2.6- Protótipo de Interface Solicitar Exames 26
4.2.7- Protótipo de Interface Manter Convênio 26
4.2.8- Protótipo de Interface Manter Especialidades 27
4.2.9- Protótipo de Interface Vínculo de Dependentes 27
4.2.10- Protótipo de Interface Vínculo de Perfil 27
4.2.11- Protótipo de Interface Manter Usuário 27
4.3- Análise do Sistema **28**
4.3.1- Diagrama de Classes 28
4.3.2- Funções do Sistema 29
4.4- Fluxos dos Casos de Uso **30**
4.4.1- Caso de Uso Manter Clientes 30
4.4.1.1- Pré-Condições 30
4.4.1.2- Atores 30
4.4.1.3- Fluxo Principal 30

4.4.1.4- Fluxo Alternativo de Alteração 30
4.4.1.5- Fluxo Alternativo de Exclusão.....31
4.4.1.6- Fluxo Alternativo de Consulta 31
4.4.1.7- Fluxo Alternativo [A1] – Cliente não Encontrado.....31
4.4.2- Caso de Uso Manter Exames 31
4.4.2.1- Pré-Condições 31
4.4.2.2- Atores 31
4.4.2.3- Fluxo Principal 32
4.4.2.4- Fluxo Alternativo de Alteração 32
4.4.2.5- Fluxo Alternativo de Exclusão.....32
4.4.2.6- Fluxo Alternativo de Consulta 32
4.4.2.7- Fluxo Alternativo [A1] – Exame não Encontrado.....33
4.4.3- Caso de Uso Gerar Relatórios 33
4.4.3.1- Pré-Condições 33
4.4.3.2- Atores 33
4.4.3.3- Fluxo Principal 33
4.4.3.4- Fluxo Alternativo 33
4.4.4- Caso de Uso Agendar Consultas.....33
4.4.4.1- Pré-Condições 33
4.4.4.3- Fluxo Principal 34
4.4.4.4- Fluxo Alternativo A1 – Cliente Não Encontrado 34
4.4.5- Manter Profissional 34
4.4.5.1- Pré-Condições 34
4.4.5.2- Atores 34
4.4.5.3- Fluxo Principal 34
4.4.5.4- Fluxo Alternativo de Alteração 35
4.4.5.5- Fluxo Alternativo de Exclusão.....35
4.4.5.6- Fluxo Alternativo de Consulta 35
4.4.5.7- Fluxo Alternativo [A1] – Profissional não Encontrado 36
4.4.6- Solicitar Exames 36
4.4.6.1- Pré-Condições 36
4.4.6.2- Atores 36
4.4.6.3- Fluxo Principal 36
4.4.6.4- Fluxo Alternativo A1 - Cliente não Encontrado 36
4.4.7- Manter Convênios.....36
4.4.7.1- Pré-Condições 37
4.4.7.2- Atores 37

4.4.7.4- Fluxo Principal 37
4.4.7.5- Fluxo Alternativo de Alteração 37
4.4.7.6- Fluxo Alternativo de Exclusão.....37
4.4.7.7- Fluxo Alternativo de Consulta 38
4.4.7.8- Fluxo Alternativo [A1] – Convênio não Encontrado.....38
4.4.8- Manter Especialidades.....38
4.4.8.1- Pré-Condições 38
4.4.8.2- Atores 38
4.4.8.4- Fluxo Principal 38
4.4.8.5- Fluxo Alternativo de Alteração 39
4.4.8.6- Fluxo Alternativo de Exclusão.....39
4.4.8.7- Fluxo Alternativo de Consulta 39
4.3.8.8- Fluxo Alternativo [A1] – Especialidade não Encontrada 39
4.4.9- Vincular Dependentes.....40
4.4.9.1- Pré-Condições 40
4.4.9.2- Atores 40
4.4.9.3- Fluxo Principal 40
4.4.9.4- Fluxo Alternativo A1 – Cliente Não Encontrado.....40
4.4.10- Caso de Uso Manter Usuários.....40
4.4.10.1- Pré-Condições 41
4.4.10.2- Atores.....41
4.4.10.3- Fluxo Principal 41
4.4.10.4- Fluxo Alternativo de Alteração 41
4.4.10.5- Fluxo Alternativo de Exclusão.....41
4.4.10.6- Fluxo Alternativo de Consulta 42
4.4.10.7- Fluxo Alternativo Vincular Perfis 42
4.4.10.8- Fluxo Alternativo [A1] – Usuário não Encontrado 42
4.5- Criação do VirtualClass Clínica utilizando o *framework* VirtualClass 42
5- REUSO DE *SOFTWARE* E VIRTUALCLASS CLÍNICA 44
6- CONCLUSÃO.....46
REFERÊNCIAS 48
APÊNDICE 50

1- INTRODUÇÃO

A Internet como rede mundial de computadores, tem sido estimulada pelas novas tecnologias e pela facilidade de um acesso rápido a um grande número de informações que estão disponíveis para o mundo todo.

O ambiente da Internet oferece suporte a diversos tipos de tecnologias, o que justifica o aumento das ferramentas desenvolvidas, que auxiliam os estudos, aumentando o número de pesquisas voltadas para a ela.

Os sistemas com plataforma *web* oferecem às pessoas a vantagem de poderem acessá-los de qualquer lugar. Existe uma gama de tecnologias que auxiliam na criação desses sistemas, que ajudam as empresas nos dias atuais a fazerem suas operações de qualquer lugar do mundo, tendo apenas a Internet como meio de acesso.

O sistema proposto neste trabalho usa dos princípios de reuso de *Software*, que é o reuso de partes de *Software* já prontas, informatizando uma clínica de atendimento médico e odontológico, para que assim seus processos possam se tornar mais ágeis.

O reuso de *Software* garante uma maior qualidade, tendo em vista que o sistema não é construído do início, sendo assim, permite a utilização de novos *Software*s gerados usando os novos artefatos e fragmentos do sistema.

O VirtualClass Clinica terá como base, o *framework* VirtualClass, gerenciado pelo *Portalmaker*[1], que fará o gerenciamento do desenvolvimento do sistema. Ele é uma aplicação que pode ser acessado via navegador, através da rede, sem necessidade de instalação em um microcomputador, sendo a instalação apenas feita em um servidor de *internet*.

A estrutura deste trabalho é feita da seguinte forma: no capítulo 2 são apresentados os conceitos de reuso de *Software*, definindo o tipo de técnica abordada para o desenvolvimento do sistema VirtualClass Clínica. Serão também observadas às vantagens do reuso de *Software* e as dificuldades na sua implementação. As tecnologias que foram usadas para o desenvolvimento do sistema, são descritas no capítulo 3. Já no capítulo 4, é demonstrado o desenvolvimento da aplicação. No capítulo 5, é feito um paralelo do uso das

[1] Sistema utilizado para o desenvolvimento do VirtualClass Clínica

tecnologias abordadas para o desenvolvimento do software. E por fim, as conclusões são apresentadas no capitulo 6, e em seguida, são expostas as referências.

2- REUSO DE SOFTWARE

O reuso de *Software* consiste na reutilização de partes de *Software* já prontas, o que ajuda na rapidez da entrega do *Software*, diminuindo gastos com o desenvolvimento entre outras vantagens.

Segundo Sommerville (2007), o reuso de *Software* é uma estratégia do processo de desenvolvimento de *Software* que é voltada para o reuso do *Software* existente e que é uma resposta às demandas por menores custos de produção e manutenção de *Software*, entregas mais rápidas de sistemas e aumento da qualidade do *Software*.

Melo (2004) afirma que, reuso é o uso de conceitos ou produtos previamente adquiridos ou construídos em uma nova situação. Isso envolve a representação desses produtos em vários níveis de abstração, o armazenamento dos mesmos para futuras referências, a identificação de similaridades entre situações novas e antigas, a recuperação de produtos já desenvolvidos (ou parte deles) e sua adaptação na nova situação. Reusabilidade é uma medida da facilidade em se utilizar os conceitos e produtos em novas situações.

Ainda segundo Melo (2004), a noção de reutilização de *Software* é antiga e consiste em utilizar *Software* existente para construir novos sistemas. Reutilização não é só aplicável a fragmentos de código fonte, mas a todo o trabalho gerado durante o processo de desenvolvimento de *Software*, como dados, arquitetura e projeto. Portanto, a informação susceptível à reutilização inclui a análise de requisitos, especificações do sistema, estruturas de desenho, e qualquer informação que seja necessária ao processo de desenvolvimento. Estes produtos do desenvolvimento são chamados artefatos de *Software*.

Dentre os diversos tipos de reuso de *Software*, este trabalho vem abordar o uso de *Frameworks*, por ser um método que pode trazer benefícios.

2.1- *Framework*

Para possibilitar o desenvolvimento das aplicações que constituem os Sistemas de Informação, existe o conceito de "*framework*". Um *framework* é um conjunto de componentes reutilizáveis como padrão de uso. Usando a tecnologia de

um *framework*, consegue-se então obter o desenvolvimento e aplicabilidade das técnicas de Reuso de *Software.* Essas técnicas correspondem a um conjunto de recursos e conceitos relacionados que criam uma plataforma de desenvolvimento contendo facilidades que exploram um modelo ou arquitetura de solução computacional. Tendo em vista esse conceito, este trabalho visa usar esse recurso para verificar o desempenho do desenvolvimento de um *Software*.

2.2- Vantagens

Segundo Sommerville (2007), uma vantagem óbvia do reuso do *Software* é que os custos totais de desenvolvimento são reduzidos. Poucos componentes de *Software* precisam ser especificados, projetados, implementados e validados. Mas ainda pode-se obter outras vantagens, como mostra o Quadro1:

Quadro 1: Benefícios do Reuso de *Software*

Benefício	Explicação
Confiança aumentada	Software reusado, experimentado e testado em sistemas de trabalho, deve ser mais confiável do que *Software* novo, pois seus defeitos de projeto e implementação já foram encontrados e corrigidos.
Risco de processo reduzido	O custo de software existente já é conhecido, enquanto os custos de desenvolvimento são sempre um problema de avaliação. Isso é um fator importante para o gerenciamento de projetos porque reduz a margem de erro de estimativa de custo de projeto. Tal fato é especialmente verdadeiro quando componentes de Softwares relativamente grandes, tais como subsistemas, são reusados.
Uso eficiente de especialistas	Em vez de fazer o mesmo trabalho repetidas vezes, esses especialistas podem desenvolver Software reusável englobando seus conhecimentos.
Conformidade com padrões	Alguns padrões, tais como padrões de interface com o usuário, podem ser implementados com um conjunto de componentes reusáveis, todas as aplicações apresentarão o mesmo formato de menu para o usuário. O uso de interfaces padronizadas com o usuário melhora a confiança porque os usuários provavelmente cometerão menos erros quando apresentados a uma interface familiar.
Desenvolvimento acelerado	A apresentação de um sistema para o mercado tão cedo quanto possível é muitas vezes mais importante do que os custos totais de desenvolvimento. O reuso de *Software* pode tornar rápida a produção do sistema porque tanto tempo de desenvolvimento quando de validação devem ser reduzidos.

Fonte: SOMMERVILLE, 2007 pag. 276

Segundo Tavares (2007):

> A reutilização otimiza as quatro variáveis que determinam o sucesso dos projetos de software: qualidade, custo, tempo e produtividade. Quando reutilizamos componentes de software, já analisados e testados, diminuímos os erros, e consequentemente aumentamos a qualidade e a segurança, reduzimos os prazos de entrega do projeto (tempo) e os custos e otimizamos recursos. A produtividade é aumentada porque se tem a possibilidade de realizar as tarefas com maior rapidez e o risco do processo é diminuído porque temos menos incertezas. A prática da reutilização resulta na retenção do conhecimento na empresa e coloca a empresa numa posição de maior competitividade na medida em que ela passa a conseguir desenvolver projetos de qualidade, no tempo acordado e com menores custos. Um software que incorpora componentes reutilizados apresenta uma estrutura mais flexível o que facilita a manutenção e a evolução. (TAVARES, 2010).

2.3- Desvantagens

A grande dificuldade que o reuso de *Software* enfrenta é a de conseguir ajustar os componentes que sejam adequados a cada tipo de aplicação, tendo em vista, as melhorias do sistema. Segundo Silva (2009), Uma grande dificuldade é manter a equipe de desenvolvimento informada sobre todos os recursos oferecidos pelos componentes disponíveis.

Em relação às desvantagens Tavares (2007) diz que:

> Quando se reutilizam componentes num contexto diferente daquele em que foram construídos, podem surgir alguns problemas: interações inesperadas entre os componentes reutilizados e os novos componentes; Embora os componentes reutilizados já tiverem sido analisados e testados, quando utilizadas no contexto de um novo projeto deve-se testar ou analisar a sua integração. Evitam-se assim problemas em fases mais avançadas do projeto. (TAVARES, 2007)

Ainda como desvantagens, Sommerville (2007), define os itens de acordo com o Quadro 2.

Quadro 2: Problemas com reuso

Benefício	Explicação
Custos de manutenção aumentados	Se o código fonte de um sistema ou componente de *Software* reusável não estiver disponível, então os custos de manutenção poderão ser aumentados, pois os elementos reusados do sistema podem tornar-se cada vez mais incompatíveis com as mudanças.
Falta de apoio de ferramenta	O conjunto de ferramentas CASE, podem apoiar o desenvolvimento com reuso. Pode ser difícil ou impossível integrar essas ferramentas a um sistema de biblioteca de componentes. O processo de *Software* suposto, por essas ferramentas pode não levar em conta o reuso.

Síndrome de não-inventado-aqui	Alguns engenheiros de *Software* preferem reescrever componentes porque acreditam que podem aprimorá-los. Isso tem, em parte, a ver com responsabilidade e, em outra, com o fato de quem a escrita do *Software* original é vista como mais desafiadora do que reusar *Software* de outras pessoas.
Criação e manutenção de uma biblioteca de componentes	Preencher uma biblioteca de componentes reusáveis e assegurar aos desenvolvedores de *Software* que podem usar essa biblioteca pode ser caro. Nossas técnicas atuais de classificação, catalogação e recuperação de componentes de *Software* são imaturas.
Procura, compreensão e adaptação de componentes reusáveis	Componentes de *Software* precisam ser descobertos numa biblioteca, compreendidos e, às vezes, adaptados para trabalhar num novo ambiente. Os engenheiros devem estar em razoavelmente confiantes de encontrar um componente na biblioteca antes de incluírem uma busca de componente como parte de seu processo normal de desenvolvimento.

Fonte: SOMMERVILLE, 2007 pag. 277

3- *FRAMEWORK* VIRTUALCLASS

O *framework* VirtualClass foi criado por Willian Borges Lisboa em 2007, para a sua dissertação de Mestrado, o qual se baseia no desenvolvimento de uma infraestrutura de suporte para a criação de grades computacionais através do compartilhamento de recursos, utilizando mecanismos que tornem transparente a existência dessa infraestrutura.

O VirtualClass é uma ferramenta de desenvolvimento de Sistemas *Web*, através da criação de portais, para gerenciamento de operações rotineiras de uma empresa, que pode utilizá-lo para adicionar funcionalidades referentes à sua área de atuação.

Ele é um *framework* capaz de gerar outros sistemas baseados na plataforma *web*, através do módulo Portalmaker, que provê o desenvolvimento de *Software* seguro, escalável, multiplataforma e em conformidade com ele.

Segundo Lisboa (2007), a definição e a adoção deste *framework,* objetiva favorecer o melhor entendimento e desenvolvimento, uma vez que todos os recursos visuais e de programação serão geridos pelo *framework*, o que resultará em facilidade ao se pensar nas manutenções de portais sejam elas: evolutiva, adaptativa e corretiva.

A manutenção evolutiva garante que os portais tenham seu desenvolvimento realizado de acordo com as novas necessidades e pretensões do administrador. Por sua vez, a manutenção adaptativa define a possibilidade dos portais atenderem a mudanças estruturais de maneira que novas interações e regras de negócios sejam atendidas. Por fim, a manutenção corretiva busca eliminar erros de operações do Sistema.

O gerenciamento das ferramentas do VirtualClass Clínica é feito através do Portalmaker, sendo o Portalmaker uma parte integrante do *framework* VirtualClass, como mostrado na figura 1, a seguir.

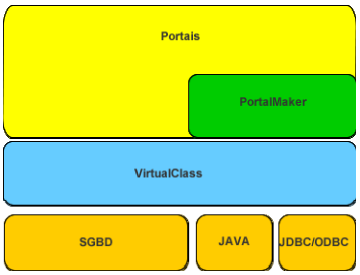

Figura 1: Estrutura do *Framework* VirtualClass
Fonte: Própria, 2010.

3.1- PORTALMAKER

O Portalmaker é o responsável pelo gerenciamento dos portais que são gerados pelo *framework* VirtualClass.

Segundo Lisboa (2007), “o Portalmaker é um sistema *web* que permite modelar portais, adicionando funcionalidades tanto visuais quanto funcionais sem a necessidade de implementação para o mesmo”.

É importante ressaltar que o Portalmaker é um *middleware*, que é uma camada de *Software* onde se concentra serviços como identificação, autenticação, autorização, diretórios, certificados digitais e outras ferramentas para segurança, entre outros serviços disponibilizados por ele.

O Portalmaker pode ser tratado como um meio para se chegar a outras aplicações, tratando-se de um portal meio e não de um portal fim. Ele pode ser considerado um gerador de portais onde sem muito conhecimento de programação, podem ser geradas e adicionadas funcionalidades e otimizações para todos os portais administrados.

O Portalmaker possui uma estrutura básica de serviços e recursos inerentes à criação e à facilitação de atividades como: criação de portais, *uploads*[2], suporte e comunicação entre usuários. Assim sendo, o Portalmaker possui um gama de recursos implementados com este propósito.

3.2- TECNOLOGIAS ENVOLVIDAS

O Portalmaker faz uso de diversas tecnologias para o desenvolvimento de seus portais e é responsável pela propagação dessas ferramentas, pois ele gerencia os complementos que são usados na implementação dos sistemas.

Cada ferramenta ou tecnologia utilizada pelo Portalmaker tem sua utilidade distinta, adicionando robusticidade para os sistemas gerados e que trabalham por trás do *framework*. O Ajax, o XML, o *JavaScript* e o HTML e XHTML.

---

[2] É o ato de enviar um arquivo para a Internet

3.2.1- AJAX

O Ajax é um apanhado de tecnologias que a *web* oferece para o desenvolvimento de novos sistemas mais interativos, dinâmicos e criativos oferecendo assim, novas funcionalidades.

De acordo com Guimarães e Rodrigues (2007, p. 38), AJAX é o uso de tecnologias como *JavaScript* e *XML* fazendo com que o navegador carregue conteúdo do servidor sem recarregar a página atual, o que torna o navegador mais interativo, transformando as páginas semelhantes às aplicações Desktop.

> O Ajax não é uma tecnologia. São realmente, várias tecnologias, cada uma atuando em seu próprio direito, juntando-se em poderosas formas novas. O Ajax incorpora: apresentação baseada em padrões, usando XHTML e CSS; exposição e interação dinâmica usando o DocumentObjectModel; intercâmbio de dados e manipulação de imagens usando XML e XSLT; recuperação assíncrona de dados usando o objeto XMLHttpRequest; e JavaScript unindo tudo. (GARRET, 2005).

O Portalmaker usa a tecnologia do Ajax em muitas das suas janelas, sendo que o desenvolvedor pode escolher se fará o uso dele em cada tela que adicionar no sistema. O Ajax faz todas as opções do Portalmaker e dos sistemas gerados, mais dinâmicos, sendo que é carregada apenas a opção que o usuário ou o desenvolvedor escolher.

3.2.2- XML

O *XML* é uma das ferramentas utilizadas para o desenvolvimento do *framework* VirtualClass. Com todas as tecnologias relacionadas, representa uma maneira distinta de compartilhar os dados com os quais se trabalha a todos os níveis, por todas as aplicações e suportes.

O servidor *web* pode retornar dados em qualquer formato baseado em texto. De qualquer forma, o XML é o mais utilizado. Isso ocorre porque, devido a sua grande evolução, o XML vem se tornando um formato padrão para troca de dados, não só em aplicativos *web*, mas também em aplicações *desktop*.

> O Extensible Markup Language (XML) surgiu em 1998 e é baseada no SGML, uma linguagem cujo propósito é definir a sintaxe de linguagens de marcação para representar dados usando tags. Ele conta que, Jacobs (2006) diz que o ponto mais importante sobre o XML é que não se trata de

> uma linguagem em si. Na verdade, trata-se de uma metalinguagem usada para construir outras linguagens e vocabulários (PINATTI, 2007, p. 60).

XML é uma linguagem que foi desenvolvida para superar as limitações do HTML, sendo definida como o formato universal para dados estruturados na *Web*, consistindo em tabelas, desenhos, parâmetros de configurações, entre outros, ou seja, a linguagem define as regras que permitem escrever documentos de forma que sejam adequadamente visíveis ao computador, não tratando da apresentação das informações, mas apenas do conteúdo a ser apresentado, tendo assim um papel importante na globalização e na compatibilidade entre os sistemas, pois o seu propósito principal é a facilidade de compartilhamento de informações com segurança, confiabilidade e facilidade através da Internet (GONÇALVES, 2006, p. 127).

Segundo Borges (2007), o Portalmaker faz uso de um arquivo em formato *XML*, que contém todas as configurações do Portalmaker. E ainda, dá exemplos de informações contidas nesse arquivo: parâmetros de conexão ao banco de dados remoto ou local; parâmetros de configuração de diretórios de *Upload* e *Download*; especificação de gerenciamento de *logs* de utilização do Portalmaker; definição de restrição e localização de recurso e módulos (alguns recursos do Portalmaker podem estar em servidores distintos).

3.2.3-JAVASCRIPT

A *JavaScript* é uma linguagem de programação simples e que se tornou padrão na *Internet*, que consiste em linguagem integrada e embutida no HTML e que permite um controle maior na apresentação de páginas, possibilitando recursos que não são disponíveis em HTML, deixando as páginas mais interativas.

Asleson e Schutta (2006, pag. 6) afirmam que: antes de se chamar *JavaScript*, a Netscape batizou primeiramente de Mocha, depois de *LimeWare* e *LiveScript* em seguida. Ela foi criada para ajudar no desenvolvimento e alteração dinamicamente das *tag's*[3] das páginas de internet, fornecendo uma experiência bem mais agradável para seus clientes.

---

[3] Estruturas de linguagem de marcação que consistem em breves instruções, tendo uma marca de início e outra de fim.

> A *JavaScript* pretende atender algumas necessidades que surgiram de acordo com a evolução da web que se tornava mais dinâmicas e tinha a necessidade de ter maior interação com o usuário como, por exemplo, a validação de formulários no lado cliente, tomar decisões e realizar outras validações simples que não eram possíveis de serem feitos utilizando a linguagem de marcação de texto simples como o HTML. As linguagens de script são mais simples por possuir uma sintaxe fácil de aprender, além de permitirem combinar script com HTML para criar páginas da Web interativas (GONÇALVES, 2006).

A *JavaScript* é voltada para executar operações no lado do usuário. Com ela, é possível fazer validações de campos, abertura de janelas, controle da utilização de botões, mensagens de alertas, confirmações e principalmente criar uma interatividade maior do usuário com a página utilizada. Com ela, também é possível alterar o estilo de uma página de forma dinâmica (LIMEIRA, 2006).

O Portalmaker utiliza *JavaScript* para fazer com que as páginas fiquem mais dinâmicas. Ele trabalha diretamente e está embutido dentro do AJAX, fazendo com que o usuário veja um *layout* mais agradável e obtenha mais interatividade. A *JavaScript* trabalha validando as entradas de dados, formatando as saídas, controlando e tratando a comunicação do AJAX dentro do Portalmaker.

3.2.4- HTML E XHTML

O *HTML* significa *HyperText Markup Language* é o formato da descrição dos documentos na *web*. Através de *tag's*, o HTML posiciona e coloca hipertexto no documento.

> HTML e XHTML definem a sintaxe e localização de instruções especiais que não são exibidas pelo browser, mas o dizem como exibir o conteúdo do documento, sendo este conteúdo texto, imagens, e outras mídias suportadas. Eles também tornam um documento interativo através de ligações de hipertexto especiais (hyperlinks), que conectam o documento a outros em qualquer lugar, assim como também a recursos da Internet PINATTI (2007).

Limeira (2006) assim diz:

> XHTML é a linguagem HTML baseada em XML. É a sigla em inglês para EXtensible HyperText Markup Language que em português resulta em Linguagem Extensível para Marcação de Hipertexto, uma aplicação XML, escrita para substituir o HTML. Com ela pode-se combinar marcações HTML com as regras XML. A ideia é ter um documento padronizado e acessível para ser apresentado em várias plataformas e dispositivos, seja computador desktop, PDA, Telefone Celular, Televisão, etc.

Em todas as páginas criadas e geradas pelo Portalmaker, existe algum código ou configuração HTML, pois esse é o primeiro conteúdo à qual se precisa descrever e entender para desenvolver um sistema *web*. O *HTML* e o XHTML são importantes no Portalmaker, para que as páginas de cada sistema tenham e dê acessibilidade para o usuário.

3.2.5- JAVA

Java é a linguagem de programação que foi usada para o desenvolvimento do *framework* VirtualClass.

O site da Oracle (2010), diz que:

> É a tecnologia que capacita muitos programas da mais alta qualidade, como utilitários, jogos e aplicativos corporativos, entre muitos outros, por exemplo. O Java é executado em mais de 850 milhões de computadores pessoais e em bilhões de dispositivos em todo o mundo, inclusive telefones celulares e dispositivos de televisão.

O desenvolvimento de aplicações para *web* obrigou os desenvolvedores a utilizar uma nova arquitetura de sistemas, abandonando a arquitetura monolítica para se empregar uma arquitetura em camadas. Com o advento da Internet, esta arquitetura monolítica teve que ser alterada, pois o tempo necessário para carregar todos os componentes da regra de negócio no cliente em um aplicativo *Web* é extremamente elevado (NETO, et al.).

Devido a grande aceitação da linguagem, ainda mais se tratando da crescente expansão dos sistemas operacionais de código abertos, a Linguagem Java tomou diversos rumos no que diz respeito a seus paradigmas de programação, onde alguns estão citados abaixo:

- *Java-Servlet*: Programas *Server-side* desenvolvidos para geração de conteúdo dinâmica de páginas *WEB*, o código HTML é colocado dentro do código Java.
- *JSP (Java Server Pages)*: Programas *server-side* desenvolvidos para geração de conteúdo dinâmico de páginas *Web*, o código Java é colocado dentro do código HTML (LISBOA, 2001).

4- DESENVOLVIMENTO DE UM SISTEMA DE CLÍNICA

Nesse capítulo será descrito tudo que envolveu o desenvolvimento do sistema, que usará como base o *framework* VirtualClass. Alguns requisitos, telas do sistema e a descrição de como cada função funciona, serão descritos. O *Software* pretende atender em parte, os afazeres operacionais de uma Clínica médica e odontológica, e sendo então levantados quesitos para atendê-la.

Neste item e no Apêndice deste documento, é demonstrada toda a documentação gerada através da análise dos requisitos levantados no decorrer do sistema. Nas clínicas os processos são manuais, mas contam com pessoas com uma carga de experiência na área, possibilitando um melhor estudo dos processos e como informatizá-los. Os diagramas apresentados são frutos de entrevistas, conversas com os responsáveis, ao longo de um período anterior ao início dos trabalhos e até mesmo durante o desenvolvimento desta monografia.

4.1- REQUISITOS DO SISTEMA

Os requisitos do sistema foram levantados de acordo com as necessidades de uma Clínica, com os padrões de requisitos e com configurações pertinentes a ela.

4.2- PROTÓTIPOS DE INTERFACE

Nessa seção será mostradas as telas geradas do sistema. O *layout* das telas é gerado automaticamente pelo *Framework* VirtualClass.

4.2.1- Protótipo de Interface Manter Clientes

A figura 2 apresenta a tela de manutenção de clientes, onde o usuário pode fazer o cadastro dos clientes e operações pertinentes, sendo importante destacar os campos Estado e Cidade já estarão preenchidos.

Figura 2: Interface Manter Clientes
Fonte: Própria, 2010.

4.2.2- Protótipo de Interface Manter Exames

A figura 3 apresenta a tela de manutenção de exames, onde o usuário pode fazer o cadastro dos exames e operações pertinentes.

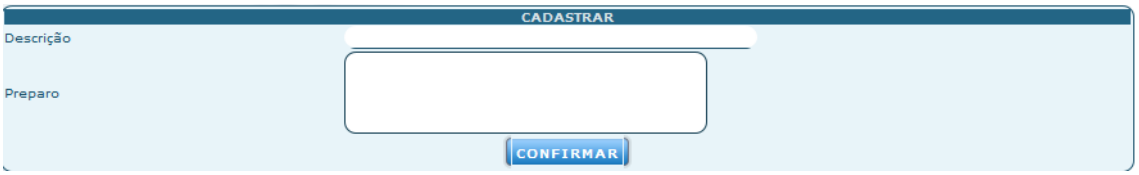

Figura 3: Interface Manter Exames
Fonte: Própria, 2010.

4.2.3- Protótipo de Interface Gerar Relatórios

A figura 4 apresenta a tela de relatórios, onde o usuário pode fazer a opção de consulta através de vários campos de opção, sendo importante destacar que essa tela, está subdividida de acordo com os principais itens de cadastro do sistema.

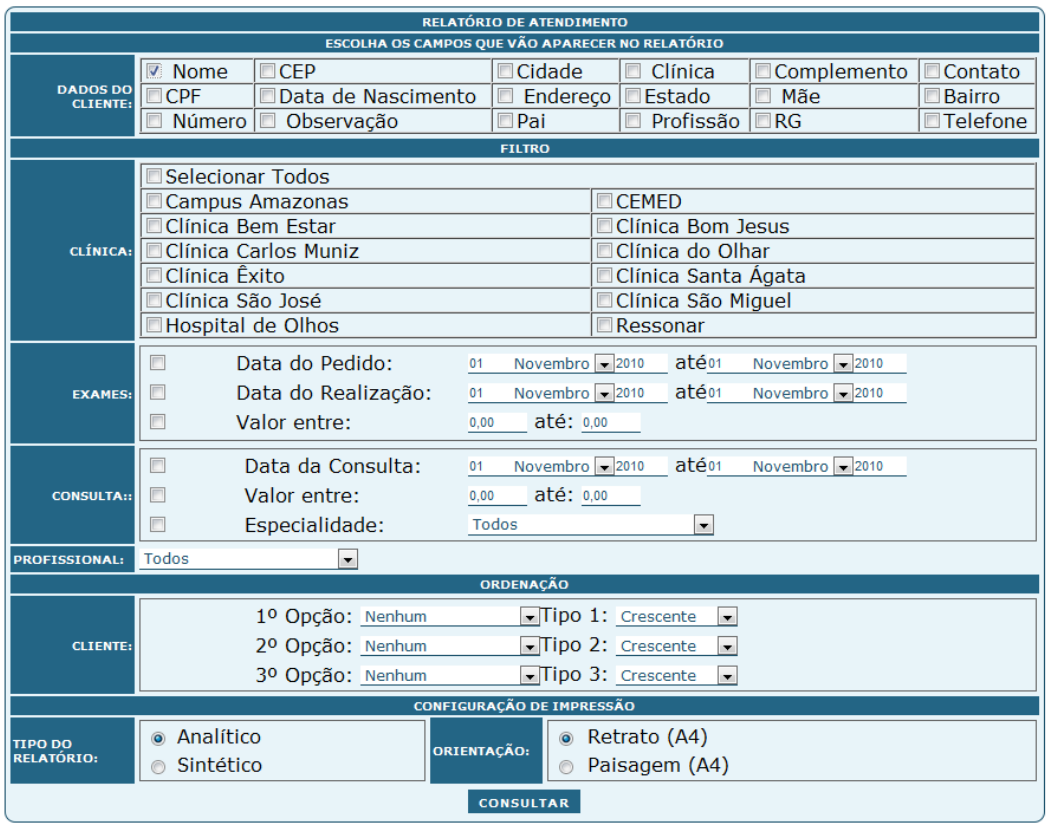

Figura 4: Interface Gerar Relatórios
Fonte: Própria, 2010.

4.2.4- Protótipo de Interface Agendar Consultas

A figura 5 apresenta a tela de agendamento de consultas, onde o usuário pode fazer o cadastro das consultas pertinentes aos clientes, sendo importante

destacar alguns campos já estão preenchidos e vinculados de acordo cadastro em outras telas, sendo: Convênio, Especialidade e Profissional.

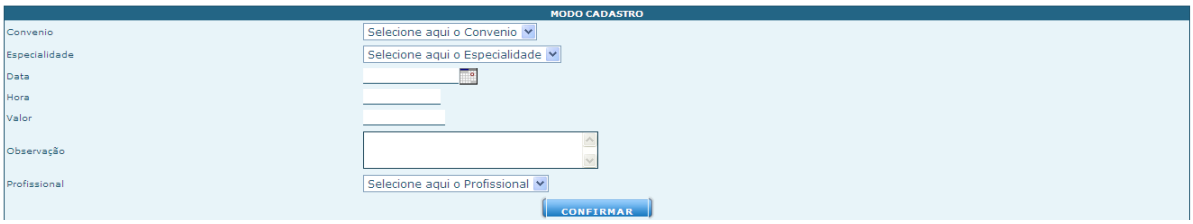

Figura 5: Interface Agendar Consultas
Fonte: Própria, 2010.

4.2.5- Protótipo de Interface Manter Profissional

A figura 6 apresenta a tela de manutenção de profissionais, onde o usuário pode fazer o cadastro dos profissionais e operações pertinentes.

Figura 6: Interface Manter Profissional
Fonte: Própria, 2010.

4.2.6- Protótipo de Interface Solicitar Exames

A figura 7 apresenta a tela de solicitação de exames, onde o usuário pode fazer o cadastro dos exames aos clientes, e importante destacar que alguns campos já estão preenchidos e, alguns campos já sejam previamente cadastrados, sendo: Convênio e Exame.

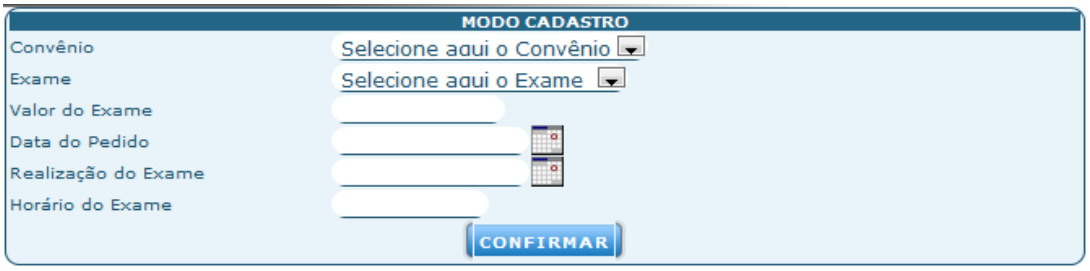

Figura 7: Interface Solicitar Exames
Fonte: Própria, 2010.

4.2.7- Protótipo de Interface Manter Convênio

A figura 8 apresenta a tela de manutenção de convênios, onde o usuário pode fazer o cadastro dos convênios e operações pertinentes, os convênios podem estar ou não associados à clínica que usa o sistema.

Figura 8: Interface Manter Convênio
Fonte: Própria, 2010.

4.2.8- Protótipo de Interface Manter Especialidades

A figura 9 apresenta a tela de manutenção de especialidade, onde o usuário pode fazer o cadastro das especialidades referentes aos serviços que a clínica ou seus associados disponibilizam.

Figura 9: Interface Manter Especialidades
Fonte: Própria, 2010.

4.2.9- Protótipo de Interface Vínculo de Dependentes

A figura 10 apresenta a tela vínculo de dependentes dos clientes, onde o usuário pode fazer o cadastro dos dependentes e operações pertinentes, sendo importante ressaltar que esse cadastro é alternativo.

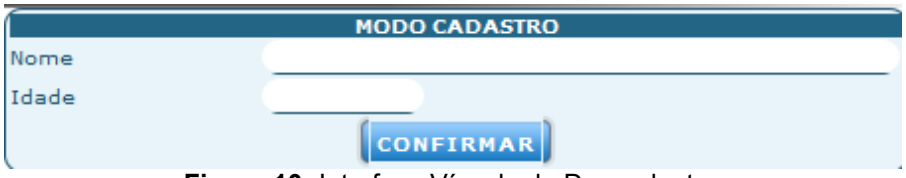

Figura 10: Interface Vínculo de Dependentes
Fonte: Própria, 2010.

4.2.10- Protótipo de Interface Vínculo de Perfil

A figura 11 apresenta a tela de vínculo de perfil dos usuários que tem acesso ao sistema, onde o administrador do Portalmaker pode fazer o vínculo dos perfis e operações pertinentes. E importante destacar que somente o administrador tem acesso a essa ferramenta.

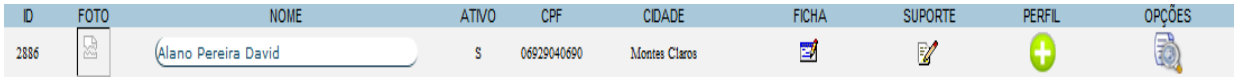

Figura 11: Interface Vínculo de Perfil
Fonte: Própria

4.2.11- Protótipo de Interface Manter Usuário

A figura 13 apresenta a tela de manutenção dos usuários do sistema, onde o administrador pode fazer o cadastro dos usuários e operações pertinentes, tendo somente o administrador com acesso a essa ferramenta.

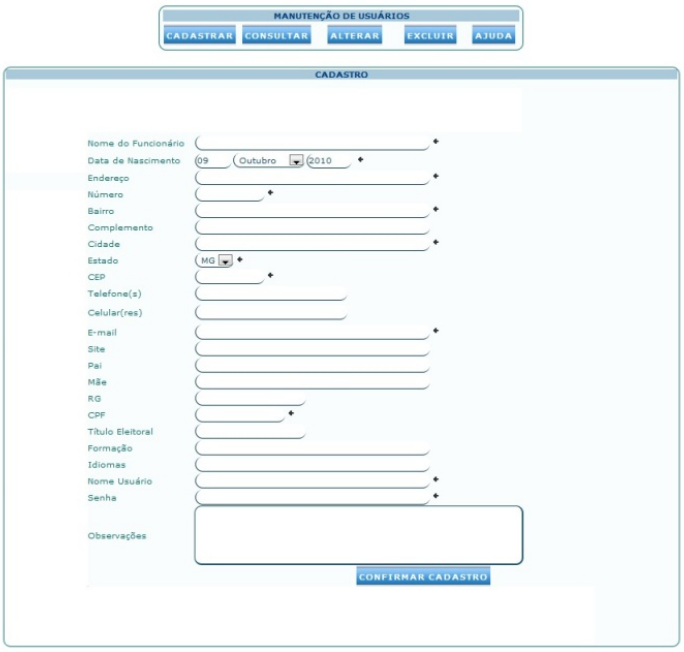

Figura 12: Interface Manter Usuário
Fonte: Própria, 2010.

4.3- Análise do Sistema

Com a análise, foram levantadas as funções e os processos que são exercidos por cada ator envolvido no processo cadastro e manutenção do sistema VirtualClass Clínica.

4.3.1- Diagrama de Classes

A figura 13 mostra o Diagrama de Classes para o Sistema VirtualClass Clínica. Esse diagrama informa as áreas tratadas pela execução do *Software*.

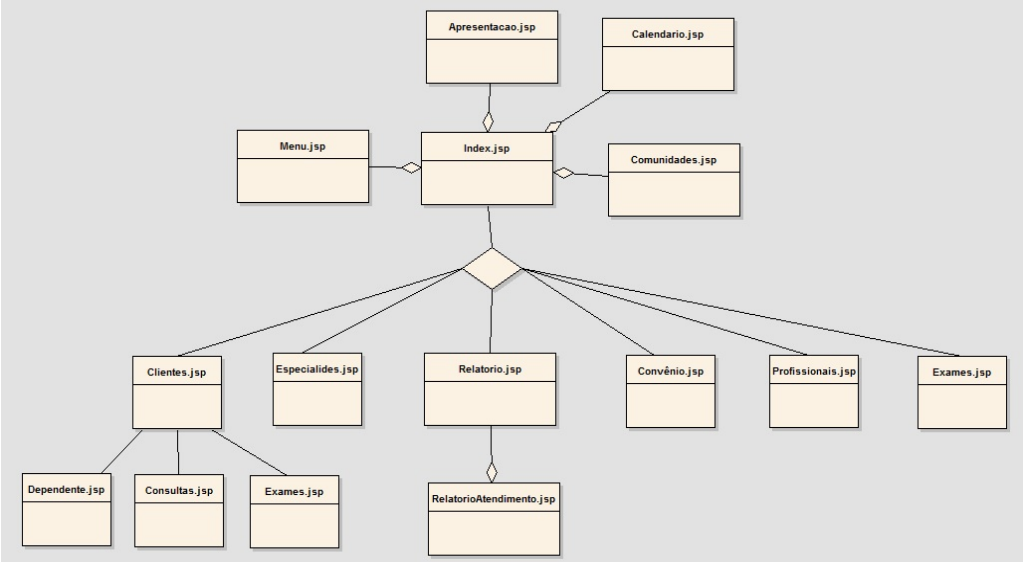

Figura 13: Diagrama de Classe do VirtualClass Clínica
Fonte: Própria, 2010.

4.3.2- Funções do Sistema

O quadro 13 demonstra um levantamento das funções que o sistema contará para distribuir os serviços disponíveis em uma clinica. O nome e a descrição de cada caso de uso, além de uma descrição detalhada sobre cada um, serão feitos no item 4.4 deste documento.

Quadro 3: Funções do Sistema

Número	Casos de uso	Descrição
01	Manter clientes	Atualizar a lista de clientes
02	Manter exames	Atualizar a lista de exames dos clientes
03	Gerar relatórios	Visualizar a situação dos processos que envolvem a clínica
04	Agendar consultas	Fazer o agendamento de consultas de acordo agenda do prossional
05	Manter profissional	Cadastro dos profissionais que atuarão na clínica
06	Solicitar exames	Atualizar a lista de exames pedidos
07	Manter convênios	Cadastro das clínicas conveniadas com a clínica cliente
08	Manter especialidades	Cadastro das especialidades médicas de atendimento e de convênios
08	Vincular dependentes	Cadastro dos dependentes dos clientes vinculados
09	Manter usuários	Cadastro de usuários para acesso ao sistema

Fonte: Própria, 2010.

4.4- Fluxos dos Casos de Uso

O fluxo de caso de uso é um detalhamento para o entendimento de como é o processo dentro do sistema. Segundo Medeiros (2004) “um caso de uso poder ser explicado como uma macroatividade que encerra diversas tarefas ou atividades. Essas tarefas visam a consecução dessamacroatividade”.

4.4.1- Caso de Uso Manter Clientes

Caso de uso utilizado quando clientes novos serão vinculados ao sistema. Os dados pessoais do cliente serão necessários para a atualização e inserção das informações dos mesmos.

4.4.1.1- Pré-Condições

- n/a

4.4.1.2- Atores

- Funcionário;

4.4.1.3- Fluxo Principal

1. O Ator seleciona a opção de Menu “Cliente”;
2. O Sistema oferece a interface de Cadastro de Cliente;
3. O Ator entra com os dados do cliente;
4. O sistema solicita confirmação;
5. O ator confirma os dados.
6. O sistema confirma o cadastro, mostrando o Cliente cadastrado.

4.4.1.4- Fluxo Alternativo de Alteração

1. O Ator seleciona a opção de Menu “Cliente”;
2. O Sistema oferece a interface de Consultas do Cliente;
3. O Ator entra com os dados do cliente;
4. O sistema lista o cliente [A1];
5. O ator seleciona a opção Alterar o cliente;
6. O ator faz a modificação dos dados;
7. O sistema solicita confirmação;
8. O ator confirma os dados.

9. O sistema confirma a alteração, mostrando o Cliente cadastrado.

4.4.1.5- Fluxo Alternativo de Exclusão

1. O Ator seleciona a opção de Menu “Cliente”;
2. O Sistema oferece a interface de Consultas do Cliente;
3. O Ator entra com os dados do cliente;
4. O sistema lista o cliente [A1];
5. O ator seleciona a opção Excluir o cliente;
6. O sistema solicita confirmação;
7. O ator confirma a exclusão.
8. O sistema mostra mensagem de “Conseguimos! Operação realizada com Sucesso”.

4.4.1.6- Fluxo Alternativo de Consulta

1. O Ator seleciona a opção de Menu “Cliente”;
2. O Sistema disponiliza os campos para Consulta;
3. O Ator entra com os dados do cliente;
4. O sistema lista o cliente [A1];

4.4.1.7- Fluxo Alternativo [A1] – Cliente não Encontrado

4. O sistema mostra mensagem de “Desculpe! Não foi encontrado nada”. Encerra o caso de uso.

4.4.2- Caso de Uso Manter Exames

Este caso de uso é executado quando um ou vários exames tem que ser vinculados no sistema, registrando o nome do exame e o preparo para o exame. A lista de exames pode ser inserida de acordo com os pedidos.

4.4.2.1- Pré-Condições

- n/a

4.4.2.2- Atores

- Funcionário;

4.4.2.3- Fluxo Principal

1. O Ator inicia o caso de uso Manter Exames;
2. O Sistema oferece a interface de Cadastro de Exames;
3. O Ator entra com os dados do exame;
4. O sistema solicita confirmação;
5. O ator confirma os dados.
6. O sistema confirma o cadastro, mostrando o exame cadastrado.

4.4.2.4- Fluxo Alternativo de Alteração

1. O Ator inicia o caso de uso Manter Exames;
2. O Sistema oferece a interface de Consultas do Exame;
3. O Ator entra com os dados do Exame;
4. O sistema lista o exame [A1];
5. O ator seleciona a opção Alterar o exame;
6. O ator faz a modificação dos dados;
7. O sistema solicita confirmação;
8. O ator confirma os dados.
9. O sistema confirma a alteração, mostrando o exame alterado.

4.4.2.5- Fluxo Alternativo de Exclusão

1. O Ator inicia o caso de uso Manter Exames;
2. O Sistema oferece a interface de Consultas do Exame;
3. O Ator entra com os dados do exame;
4. O sistema lista o exame [A1];
5. O ator seleciona a opção Excluir o exame;
6. O sistema solicita confirmação;
7. O ator confirma a exclusão.
8. O sistema mostra mensagem de “Conseguimos! Operação realizada com Sucesso”.

4.4.2.6- Fluxo Alternativo de Consulta

1. O Ator inicia o caso de uso Manter Exames;
2. O Sistema disponiliza os campos para Consulta;
3. O Ator entra com os dados do exame;
4. O sistema lista o exame [A1];

4.4.2.7- Fluxo Alternativo [A1] – Exame não Encontrado

4. O sistema mostra mensagem de “Desculpe! Não foi encontrado nada”. Encerra o caso de uso.

4.4.3- Caso de Uso Gerar Relatórios

Os relatórios gerados tem a finalidade de mostrar um apanhado de informações, pertinentes às operações realizadas no Sistema VirtualClass Clínica.

4.4.3.1- Pré-Condições

- As informações do sistema devem estar preenchidas para a geração dos relatórios.

4.4.3.2- Atores

- Profissional

4.4.3.3- Fluxo Principal

1. O Ator seleciona a opção Relatórios;
2. O Sistema oferece a interface com as opções do relatório;
3. O Ator entra seleciona os dados que ele quer que apareça no relatório;
4. O sistema lista o relatório;

4.4.3.4- Fluxo Alternativo

- n/a.

4.4.4- Caso de Uso Agendar Consultas

O agendamento das consultas é a marcação de horários específicos para os profissionais estarem diagnosticando, solicitando exames.

4.4.4.1- Pré-Condições

O Cliente a qual se deseja agendar a consulta deve estar previamente cadastrado no sistema.

4.4.4.2- Atores

- Funcionário;

4.4.4.3- Fluxo Principal

1. O Ator seleciona a opção consultar um cliente;
2. O Sistema oferece a interface de Consultas do Cliente;
3. O Ator entra com os dados do cliente;
4. O sistema lista o cliente [A1];
5. O ator seleciona a opção agendar consulta para o cliente;
6. O sistema oferece a interface de agendamento;
7. O Ator insere os dados do Agendamento;
8. O sistema solicita confirmação;
9. O ator confirma os dados.
10. O sistema mostra a interface de listagem de agendamentos.

4.4.4.4- Fluxo Alternativo A1 – Cliente Não Encontrado

4. O sistema mostra mensagem de “Desculpe! Não foi encontrado nada”. Encerra o caso de uso.

4.4.5- Manter Profissional

O cadastro de Profissionais é o cadastro dos médicos ou dentistas e profissionais que podem atuar ou não na Empresa Cliente. Ele estará diretamente ligado à questão que diz respeito às consultas realizadas pelas clínicas ou encaminhadas para outro convênio.

4.4.5.1- Pré-Condições

- n/a

4.4.5.2- Atores

- Funcionário;

4.4.5.3- Fluxo Principal

1. O Ator inicia o caso de uso Manter Profissional;
2. O Sistema oferece a interface de Cadastro de Profissionais;
3. O Ator entra com os dados do Profissional;

4. O sistema solicita confirmação;
5. O ator confirma os dados.
6. O sistema confirma o cadastro, mostrando o Profissional cadastrado.

4.4.5.4- Fluxo Alternativo de Alteração

1. O Ator seleciona a opção de Menu “Profissional”;
2. O Sistema oferece a interface de Consulta de Profissional;
3. O Ator entra com os dados do profissional;
4. O sistema lista o profissional [A1];
5. O ator seleciona a opção Alterar o profissional;
6. O ator faz a modificação dos dados;
7. O sistema solicita confirmação;
8. O ator confirma os dados.
9. O sistema confirma a alteração, mostrando o Profissional alterado.

4.4.5.5- Fluxo Alternativo de Exclusão

1. O Ator seleciona a opção de Menu “Profissional”;
2. O Sistema oferece a interface de Consultas do Profissional;
3. O Ator entra com os dados do profissional;
4. O sistema lista o profissional [A1];
5. O ator seleciona a opção Excluir o profissional;
6. O sistema solicita confirmação;
7. O ator confirma a exclusão.
8. O sistema mostra mensagem de “Conseguimos! Operação realizada com Sucesso”.

4.4.5.6- Fluxo Alternativo de Consulta

1. O Ator seleciona a opção de Menu “Profissional”;
2. O Sistema disponiliza os campos para Consulta;
3. O Ator entra com os dados do profissional;
4. O sistema lista o profissional [A1];

4.4.5.7- Fluxo Alternativo [A1] – Profissional não Encontrado

1. O sistema mostra mensagem de “Desculpe! Não foi encontrado nada”. Encerra o caso de uso.

4.4.6- Solicitar Exames

O Profissional vai solicitar exames de acordo com a necessidade do cliente. À medida que houver, ele irá inserir no sistema, através de cada cliente, o exame solicitado, podendo ser encaminhado para um convênio ou até mesmo na própria clinica.

4.4.6.1- Pré-Condições

O Cliente a qual se deseja agendar o exame deve estar previamente cadastrado no sistema.

4.4.6.2- Atores

- Profissional.

4.4.6.3- Fluxo Principal

1. O Ator seleciona a opção consultar um cliente;
2. O Sistema oferece a interface de Consultas do Cliente;
3. O Ator entra com os dados do cliente;
4. O sistema lista o cliente [A1];
5. O ator seleciona a opção solicitar exames para o cliente;
6. O sistema oferece a interface de solicitação de exame;
7. O Ator insere os dados do Exame;
8. O sistema solicita confirmação;
9. O ator confirma os dados.
10. O sistema mostra a interface de listagem de exames.

4.4.6.4- Fluxo Alternativo A1 - Cliente não Encontrado

4. O sistema mostra mensagem de “Desculpe! Não foi encontrado nada”. Encerra o caso de uso.

4.4.7- Manter Convênios

O convênios são parceiras com as empresas que serão vinculadas às operações e consultas dos clientes, como exames e consultas com especialidades diferentes.

4.4.7.1- Pré-Condições

- n/a.

4.4.7.2- Atores

- Funcionário;

4.4.7.4- Fluxo Principal

1. O Ator inicia o caso de uso Manter Convênios;
2. O Sistema oferece a interface de Cadastro de Convênios;
3. O Ator entra com os dados do convênio;
4. O sistema solicita confirmação;
5. O ator confirma os dados.
6. O sistema confirma o cadastro, mostrando o convênio cadastrado.

4.4.7.5- Fluxo Alternativo de Alteração

1. O Ator inicia o caso de uso Manter Convênio;
2. O Sistema oferece a interface de Consultas do Convênio;
3. O Ator entra com os dados do Convênio;
4. O sistema lista os convênios [A1];
5. O ator seleciona a opção Alterar o convênio;
6. O ator faz a modificação dos dados;
7. O sistema solicita confirmação;
8. O ator confirma os dados.
9. O sistema confirma a alteração, mostrando o convênio alterado.

4.4.7.6- Fluxo Alternativo de Exclusão

1. O Ator inicia o caso de uso Manter Convênios;
2. O Sistema oferece a interface de Consultas dos Convênios;
3. O Ator entra com os dados do Convênio;
4. O sistema lista os Convênios [A1];

5. O ator seleciona a opção Excluir o Convênio;
6. O sistema solicita confirmação;
7. O ator confirma a exclusão.
8. O sistema mostra mensagem de “Conseguimos! Operação realizada com Sucesso”.

4.4.7.7- Fluxo Alternativo de Consulta

1. O Ator inicia o caso de uso Manter Convênios;
2. O Sistema disponiliza os campos para Consulta;
3. O Ator entra com os dados do Convênio;
4. O sistema lista os Convênios [A1];

4.4.7.8- Fluxo Alternativo [A1] – Convênio não Encontrado

4. O sistema mostra mensagem de “Desculpe! Não foi encontrado nada”. Encerra o caso de uso.

4.4.8- Manter Especialidades

Serão vinculadas especialidades médicas para controle de exames e consultas, nas diversas áreas de atuação da Clínica, ou de convênio, para o vínculo de clientes às suas referentes atividades.

4.4.8.1- Pré-Condições

- n/a

4.4.8.2- Atores

- Funcionário;

4.4.8.4- Fluxo Principal

1. O Ator inicia o caso de uso Manter Especialidades;
2. O Sistema oferece a interface de Cadastro de Especialidades;
3. O Ator entra com os dados da especialidade;
4. O sistema solicita confirmação;
5. O ator confirma os dados.
6. O sistema confirma o cadastro, mostrando a especialidade cadastrada.

4.4.8.5- Fluxo Alternativo de Alteração

1. O Ator inicia o caso de uso Manter Especialidades;
2. O Sistema oferece a interface de Consulta da Especialidade;
3. O Ator entra com os dados da Especialidade;
4. O sistema lista a Especialidade [A1];
5. O ator seleciona a opção Alterar a especialidade;
6. O ator faz a modificação dos dados;
7. O sistema solicita confirmação;
8. O ator confirma os dados.
9. O sistema confirma a alteração, mostrando a especialidade alterada.

4.4.8.6- Fluxo Alternativo de Exclusão

1. O Ator inicia o caso de uso Manter Especialidades;
2. O Sistema oferece a interface de Consultas da Especialidade;
3. O Ator entra com os dados da especialidade;
4. O sistema lista a Especialidade [A1];
5. O ator seleciona a opção Excluir a especialidade;
6. O sistema solicita confirmação;
7. O ator confirma a exclusão.
8. O sistema mostra mensagem de “Conseguimos! Operação realizada com Sucesso”.

4.4.8.7- Fluxo Alternativo de Consulta

1. O Ator inicia o caso de uso Manter Especialidades;
2. O Sistema disponiliza os campos para Consulta;
3. O Ator entra com os dados da especialidade;
4. O sistema lista a especialidade [A1];

4.3.8.8- Fluxo Alternativo [A1] – Especialidade não Encontrada

4. O sistema mostra mensagem de “Desculpe! Não foi encontrado nada”. Encerra o caso de uso.

4.4.9- Vincular Dependentes

Os dependentes dos clientes serão registrados, caso os convênios sejam vinculados para a família ou para dependentes financeiros. O limite de idade vai ser estipulado pela Clínica.

4.4.9.1- Pré-Condições

O cadastro dos dependentes será vinculado após cadastro do cliente referido.

4.4.9.2- Atores

- Funcionário;

4.4.9.3- Fluxo Principal

1. O Ator seleciona a opção consultar um cliente;
2. O Sistema oferece a interface de Consultas do Cliente;
3. O Ator entra com os dados do cliente;
4. O sistema lista o cliente [A1];
5. O ator seleciona a opção vincular dependentes para o cliente;
6. O sistema oferece a interface de dependentes;
7. O Ator insere os dados da Dependente;
8. O sistema solicita confirmação;
9. O ator confirma os dados.
10. O sistema mostra a interface de listagem de dependentes.

4.4.9.4- Fluxo Alternativo A1 – Cliente Não Encontrado

4. O sistema mostra mensagem de “Desculpe! Não foi encontrado nada”. Encerra o caso de uso.

4.4.10- Caso de Uso Manter Usuários

Neste caso de uso será feito o cadastro dos usuários que terão acesso ao sistema. Eles serão responsáveis por preencher os dados do sistema, e responsável pelas operações e acesso às funcionalidades.

4.4.10.1- Pré-Condições

O vínculo de Perfil será administrado pelo Portalmaker, sendo que o administrador é quem será responsável pelo cadastro dos usuários.

4.4.10.2- Atores

- Administrador;

4.4.10.3- Fluxo Principal

1. O Ator inicia o caso de uso Manter Usuários;
2. O Sistema oferece a interface de Cadastro de Usuários;
3. O Ator entra com os dados do usuário;
4. O sistema solicita confirmação;
5. O ator confirma os dados.
6. O sistema confirma o cadastro, mostrando o usuário cadastrado.

4.4.10.4- Fluxo Alternativo de Alteração

1. O Ator inicia o caso de uso Manter Usuários;
2. O Sistema oferece a interface de Consulta do Usuário;
3. O Ator entra com os dados do Usuário;
4. O sistema lista o Usuário [A1];
5. O ator seleciona a opção Alterar o usuário;
6. O ator faz a modificação dos dados;
7. O sistema solicita confirmação;
8. O ator confirma os dados.
9. O sistema confirma a alteração, mostrando o usuário alterado.

4.4.10.5- Fluxo Alternativo de Exclusão

1. O Ator inicia o caso de uso Manter Usuários;
2. O Sistema oferece a interface de Consultas do Usuário;
3. O Ator entra com os dados do usuário;
4. O sistema lista o usuário [A1];
5. O ator seleciona a opção Excluir o usuário;
6. O sistema solicita confirmação;
7. O ator confirma a exclusão.

8. O sistema mostra mensagem de “Conseguimos! Operação realizada com Sucesso”.

4.4.10.6- Fluxo Alternativo de Consulta

1. O Ator inicia o caso de uso Manter Usuários;
2. O Sistema disponiliza os campos para Consulta;
3. O Ator entra com os dados do usuário;
4. O sistema lista o usuário[A1];

4.4.10.7- Fluxo Alternativo Vincular Perfis

1. O Ator inicia o caso de uso Manter Usuários;
2. O Sistema oferece a interface de Consultas do Usuário;
3. O Ator entra com os dados do usuário;
4. O sistema lista o usuário [A1];
5. O Ator seleciona a opção perfil;
6. O Ator seleciona o perfil do usuário;
7. O Ator confirma o vínculo;
8. O sistema confirma o vínculo, mostrando a tela do usuário com o perfil vinculado.

4.4.10.8- Fluxo Alternativo [A1] – Usuário não Encontrado

4. O sistema mostra mensagem de “Desculpe! Não foi encontrado nada”. Encerra o caso de uso.

4.5- Criação do VirtualClass Clínica utilizando o *framework* VirtualClass

Através de instruções feitas por Lisboa (2007), o sistema pode ser gerado automaticamente sem muito conhecimento de programação, pois o *framework* é responsável pela geração de códigos e geração das templates utilizadas por ela, como forma de herança.

Ao iniciar, o desenvolvedor precisa apenas ter conhecimentos básicos, em forma de teorias dos processos que ocorre no sistema.

Segundo Lisboa (2007):

> Para que um portal seja gerado é preciso conhecer alguns conceitos abordados pelo sistema como: Configuração e Criação dos portais, os

> conceitos que envolvem a criação e utilização das templates e plugins, bem como a criação dos mesmos. Os portais gerados utilizam os recursos que o sistema contém e podem evoluir de acordo com a adição e manutenção dos recursos utilizados em sua geração. Tais recursos englobam o frameworkVirtualClass, as templates, os plugins e os serviços provenientes do Portalmaker como os Sistemas de Avisos, Suporte, Download e Uploads.

Em relação ao desenvolvimento e implementação das templates, Lisboa (2007), ainda diz que se devem levantar algumas questões referentes às funções e alguns tipos de telas:

- A qual Perfil pertence? Ex.: Administrador Geral;
- A qual Menu pertence? Ex.: Cadastro;
- Qual *Template* utilizar?

Dependendo da *template* escolhida poderá responder perguntas do tipo:

- A tabela possuirá tela de cadastro, Consulta e manutenção?
- Que campos serão de preenchimento do usuário?
- Quais possuem restrição de valores e modo de apresentação diferente?
- Que campos devem ser importados de outra tabela e que tabela é essa?

Enfim, definirá todas as informações pertinentes à geração de um formulário que irá gerir as informações das tabelas solicitadas.

Inicialmente, propôs-se o sistema para ser desenvolvido. Foi levantado os requisito e campos de cada tela, os campos de cadastro para serem inseridos. Para cada tela proposta e gerada pelo sistema, foi sendo criada uma página *jsp*[4] para cada classe com seus componentes e acessórios (*scripts*, imagens, vídeos, etc.). Depois os campos do banco de dados nas telas a serem geradas, foram vinculados.

[4] Formato de página gerada pela Java.

5- REUSO DE *SOFTWARE* E VIRTUALCLASS CLÍNICA

A Engenharia de *Software* baseada em reuso é uma abordagem de desenvolvimento que tenta maximizar o reuso de *Software* já existentes. Sendo assim o VirtualClass Clínica fez o reuso do *Software*, o qual esse herdou alguns atributos do *framework* VirtualClass.

Praticamente todo o desenvolvimento, que envolveu a criação dos artefatos e telas foram criados pelo VirtualClass Clínica. A linguagem de programação e os códigos do *framework* são herdados para o sistema, pois o padrão do desenvolvimento é todo baseado no *framework*. As telas geradas são obtidas das *templates* configuradas pelo desenvolvedor dos sistemas do Portalmaker, pois o reuso de *Software* auxilia para ganhar tempo na construção.

O desenvolvimento do sistema como um todo, foi consideravelmente rápido, pois alguns processos nesse desenvolvimento foram acelerados pelo fato de que os códigos e as telas do sistema serem gerados automaticamente. As funções do VirtualClass Clínica já foram implementadas e testadas anteriormente, deixando o trabalho de desenvolvimento mais confiável que um *Software* novo, ainda no início de sua evolução. Em vez de fazer o trabalho, e testar as funções, como seria se fosse um novo *Software* ele foi reusado, sendo então incorporadas as funções do VirtualClass *framework*.

A padronização do *Software*, com suas interfaces, foram reusadas como um conjunto de atributos herdados do *framework*. Através do reuso, pode-se ter a confiança dos usuários, pois eles podem não cometer os mesmos erros no uso de *Software*s futuros, a serem implementados usando o VirtualClass.

Os serviços oferecidos pelo *framework* são disponibilizados para todo sistema criado e gerado pelo *Portalmaker* como *Downloads* e *Uploads* de imagens. Os campos das telas, cadastros e criação de perfis, foram todos herdados pelo *framework*.

Conforme a estrutura sugere, um *framework* é uma estrutura genérica, que pode ser ampliada para criar uma aplicação, mais específica, como é o caso do VirtualClass, que sendo gerado por um *framework*, pode se tornar modelo para o incentivo a Enfim, observa-se que o reuso de *Software* é uma técnica interessante no ponto de vista em que o desenvolvimento pode ser aplicado, pois ela permite obter realmente resultados expressivos e que auxiliam o desenvolvimento de novos *Softwares*. Estudar

melhor essa forma pode significar que a empresa de desenvolvimento tem o seu foco em oferecer aos clientes soluções de baixo custo, em um curto espaço de tempo e com boa qualidade. criação de mais sistemas de informação.

6- CONCLUSÃO

Este trabalho introduziu alguns aspectos teóricos e práticos relativos ao estudo do reuso de *Software*, enfatizando as técnicas de utilização de um *framework* como meio de conseguir atingir as metas de desenvolvimento.

Estudar as tecnologias criadas para implementar essa solução e suas limitações é essencial para a implantação de um processo de reuso de *Software*, pois requer um estudo detalhado das atividades e funções que um cliente necessita.

O desenvolvimento desse projeto e o estudo proposto de reuso de *Software* possibilitaram a obtenção de um sistema que se adapta as necessidades de uma clínica, demonstradas nas quatro áreas: custo, tempo e qualidade e produtividade. No quesito custo, pois são necessários pouco desenvolvedores, e esses sem muito conhecimento em programação, para desenvolver os sistemas. O tempo no desenvolvimento foi economizado, pois as funções já estavam prontas e puderam ser aproveitadas e já que os códigos e funções aproveitadas já tinham sido testados antes, o sistema conseguiu ter uma boa qualidade. Além disso, pode-se obter uma alta produtividade proporcionando dados para a criação de outros sistemas no futuro. Dentro de pouco tempo, com um baixo custo e com uma qualidade e produtividade consideráveis, pode-se obter uma ferramenta de auxílio a uma clínica médica e odontológica.

A ferramenta VirtualClass Clínica se torna uma ferramenta para uso no dia-a-dia de uma empresa, para tratar de problemas simples da rotina operacional da clínica, sendo que suas funções e operações são comuns e idênticas às operações comuns a qualquer outro sistema.

Além do reaproveitamento e padronização do *Software* e na redução da complexidade em tarefas semelhantes, um sistema que já tenha funções tratadas e resolvidas, tem também menores possibilidades de problemas na fase de implementação, fator que depende do nível de tratamentos do *framework.*

Como projetos futuros para o *Software* VirtualClass Clínica, pode-se adicionar funcionalidades financeiras, para um completo e mais amplo auxílio aos serviços oferecidos pela clínica. Além de que, pelo fato de ser online, pode-se implementar uma integração do cliente com o sistema, onde o mesmo poderá marcar e agendar consultas diretamente de casa.

Enfim, observa-se que o reuso de *Software* é uma técnica interessante no ponto de vista em que o desenvolvimento pode ser aplicado, pois ela permite obter realmente resultados expressivos e que auxiliam o desenvolvimento de novos *Softwares*. Estudar melhor essa forma pode significar que a empresa de desenvolvimento tem o seu foco em oferecer aos clientes soluções de baixo custo, em um curto espaço de tempo e com boa qualidade.

REFERÊNCIAS

ASLESON, R.; SCHUTTA, N. Fundamentos do AJAX. Rio de Janeiro: Alta Books, 2006.

GARRET, J. J. (2005) “Ajax: A New Approach to *Web* Applications”. Disponível em <http://www.adaptivepath.com/publications/essays/archives/000385.php>, Fev. 2005. Acesso em: 12 mai. 2010.

GIMENES et al., 2005. Desenvolvimento Baseado em Componentes: Conceitos e Técnicas. Rio de Janeiro: Editora Ciência Moderna Ltda., 2005.

GONÇALVES, Edson. Dominando o AJAX. Rio de Janeiro: Ciência Moderna, 2006.

GUIMARÃES, Nelson; RODRIGUES, Raphael Zanon (2007) “Aplicações *Web* com Ajax”, Disponível em: <http://www.univem.edu.br/cursos/tc_admin/nelson_raphael.pdf>. Acesso em: 03 mai. 2010.

LIMEIRA, José Luiz Silveira (2006). “Utilização de AJAX no desenvolvimento de sistemas *Web*”, <http://www.limeira.eti.br/monografia_ajax.pdf>. Acesso em 06 de mai. de 2010.

LISBOA, Willian Borges. Especificação e Implementação de uma Arquitetura Distribuída Multiplataforma Utilizando: Orientação a Objeto, *XML* e Consciência de Contexto. 2001. 49 f. Monografia (Graduação em Ciência da Computação) – Faculdade de Ciência de Computação, Universidade José do Rosário Vellano - UNIFENAS, Alfenas – MG, 2001.

LISBOA, Willian Borges. Um Portal para o P2PGrid**.** 2007. 121f. Dissertação (Mestrado em Ciências da Computação) – DEPARTAMENTO DE COMPUTAÇÃO, Universidade de São Carlos, São Carlos – SP, Julho/2007.

MEDEIROS, Ernani. Desenvolvendo *Software* com UML – Definitivo 2.0**.**São Paulo: Pearson Makron Books, 2004.

MELO, Cláudia de Oliveira (2004). “Reutilização de *Software*: Classificação e Seleção de Artefatos Reutilizáveis”, <http://www.ime.usp.br/~yw/2004/mac5701i/monografias/claudia-acvm.pdf>. Acesso em 10 de Out. de 2010.

NETO, KanjiHaraet al. “Desenvolvimento de Sistema *Web* utilizando arquitetura em Três Camadas e Applets”, <http://inf.unisul.br/~ines/workcomp/cd/pdfs/2905.pdf>. Acesso em: 17 de maio de 2010.

ORACLE. O Que é a tecnologia Java e por que é necessária? Disponível em: <http://www.java.com/pt_BR/download/faq/whatis_java.xml>. Acesso em 08 de Novembro de 2010.

PINATTI, P. R. (2007) “Desenvolvimento De Aplicações *Web* Utilizando a abordagem Ajax”, <http://bibdig.poliseducacional.com.br/document/?view=68, 2007>. Acesso em: 12 mai. 2010.

TAVARES, Joelma. Reutilização de Software. Disponível em: <http://twiki.fe.up.pt/bin/view/ERSS0607/Reutiliza%e7%e3oDe*Software*#VantagensAnchor>. Acesso em 13 de Out. de 2010.

SANTOS, D. T. (2007). “Estudo comparativo do desempenho quanto a adoção do ajax em desenvolvimento *web*”, <http://www.ccet.unimontes.br/arquivos/monografias/250.pdf>. Acesso em: 14 de mai. de 2010.

SILVA, Jackson Alexandre. Reuso de *Software* aplicado ao desenvolvimento de aplicações *web* – gestão de projetos. 2009. 22f**.** Monografia(Curso de Especialização em Análise, Projeto e Gerência de Sistemas com Ênfase em Inteligência em Negócios). Curso de Ciência da Computação da Universidade Estadual de Londrina, Londrina - SC, 2009.

SOMMERVILLE, Ian. Engenharia de *Software*, 8ª Edição; tradução: Selma Shin Shimizu Melnikoff, Reginaldo Arakaki, Edílson de Andrade Barbosa; revisão técnica: KeichiKirama. 8ª Ed. São Paulo: Pearson Addison – Wesley, 2007.

APÊNDICE

Levantamento dos Requisitos do Sistema

Quadro 4: Requisito Manter Clientes

F1 Manter	Oculto ( )			
Descrição: O sistema deve registrar os dados pessoais dos clientes que fizeram convênio ou contrato com a clínica, bem como os seus dependentes.				
Requisitos Não Funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF 1.1 Campos do cadastro	Os campos disponíveis para o cadastro são: Nome, CPF, RG, Endereço, Número, Estado, Cidade, Bairro, Pai, Mãe, Complemento, Telefone, Contato, Tipo do Contato.	Implementação	()	(X)
NF 1.2 Campo Nome do Cliente	Campo de texto com no máximo 100 caracteres alfabéticos	Implementação	()	(X)
NF 1.3 Campo CPF do Cliente	Campo de texto com no máximo 11 caracteres alfanuméricos	Implementação	(X)	()
NF 1.4 Campo RG do Cliente	Campo de texto com no máximo 15 caracteres alfanuméricos	Implementação	(X)	()
NF 1.5 Campo Rua do Cliente	Campo de texto com no máximo 100 caracteres alfabéticos	Implementação	(X)	()
NF 1.2 Campo Número do Cliente	Campo de texto com no máximo 5 caracteres alfanuméricos	Implementação	(X)	()
NF 1.6 Campo Estado do Cliente	Campo de texto com no máximo 30 inseridos através de uma lista.	Implementação	()	(X)
NF 1.7 Campo Cidade do Cliente	Campo de texto com no máximo 30 caracteres alfabéticos	Implementação	()	(X)

Nome	Restrição	Categoria	Desejável	Permanente
NF 1.8 Campo Bairro do Cliente	Campo de texto com no máximo 50 caracteres alfabéticos	Implementação	(X)	()
NF 1.9 Campo Pai do Cliente	Campo de texto com no máximo 100 caracteres alfabéticos	Implementação	(X)	()
NF 1.10 Campo Mãe do Cliente	Campo de texto com no máximo 100 caracteres alfabéticos	Implementação	(X)	()
NF 1.11 Campo Complemento do Cliente	Campo de texto com no máximo 5 caracteres alfabéticos	Implementação	(X)	()
NF 1.12 Campo Telefone do Cliente	Campo de texto com no máximo 14 caracteres alfabéticos	Implementação	(X)	()
NF 1.2 Campo Contato do Cliente	Campo de texto com no máximo 50 caracteres alfabéticos	Implementação	(X)	()
NF 1.2 Campo Observações do Cliente	Campo de texto com no máximo 100 caracteres alfabéticos	Implementação	(X)	()
NF 1.3 Controle de Acesso	A função poderá ser acessada por um Funcionário e pelo Administrador	Segurança	()	(X)
NF 1.5 Tempo de Registro	O tempo previsto para cadastro de cada cliente é de no máximo 2 minutos	Desempenho	(X)	()

Fonte: Própria, 2010.

Quadro 5: Requisito Manter Exames

F2 Manter Exames	Oculto ( )			
Descrição: O sistema deve registrar os exames e que serão vinculados para eles, de acordo com o Caso de Uso, Vínculo de Exames ao cliente desejado. O exame que será solicitado pelo Médico, será cadastrado através do sistema.				
Requisitos Não Funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**

NF 2.1 Campos do Cadastro	Os campos disponíveis para o cadastro são: Descrição, Preparo.	Implementação	()	(X)
NF 2.2Campo Descrição do Exame	Campo de texto com no máximo 100 caracteres alfabéticos, com o nome do exame.	Implementação	(X)	( )
NF 2.3Campo Preparo do Exame	Campo de texto com no máximo 100 caracteres alfabéticos, onde será descrito o preparo para a realização do exame.	Implementação	(X)	( )
NF 2.4Controle de Acesso	A função poderá ser acessada por um usuário com nível de Funcionário	Segurança	( )	(X)

Fonte: Própria, 2010.

Quadro 6: Requisito Gerar Relatórios

F3 Gerar Relatórios	Oculto ( )			
Descrição: O sistema deverá gerar relatórios como critério de especificação da Clínica. Os relatórios serão gerados com informações diversas para facilitar o atendimento e resolução de problemas, além do levantamento de informações pertinentes aos serviços fornecidos pela clínica.				
Requisitos Não Funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF 3.1Campos do Relatório	O Funcionário deverá selecionar os campos do relatório: Dados do Cliente, Clínica, Exames, Consulta.	Configurabilidade	(X)	( )

Fonte: Própria, 2010.

Quadro 7: Requisito Agendar Consultas

F4 Agendar Consultas	Oculto ( )			
Descrição: Os funcionários deverão agendar as consultas dos profissionais, para que os profissionais possam acompanhar a agenda de atendimentos.				
Requisitos Não Funcionais				

Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**

NF 4.1 Campos do Cadastro do Agendamento de Consultas	Os campos disponíveis para o cadastro são: Convênio, Especialidade, Data, Hora, Valor, Profissional.	Implementação	( )	(X)
NF 4.2 Campo Convênio do Agendamento de Consultas	O campo será gerado da tabela de Cadastro de Convênios	Implementação	( )	(X)
NF 4.3 Campo Especialidade do Agendamento de Consultas	O campo será gerado da tabela de Cadastro de Especialidades	Implementação	( )	(X)
NF 4.4 Campo Data do Agendamento de Consultas	O Campo vai ser inserido de acordo a data de disponibilidade do Profissional, que será vinculada pelo mesmo através do Sistema.	Implementação	( )	(X)
NF 4.5 Campo Hora do Agendamento de Consultas	O Campo vai ser inserido de acordo com o horário de disponibilidade do Profissional.	Implementação	( )	(X)
NF 4.6 Campo Valor do Agendamento de Consultas	O campo vai ser inserido de acordo preço da consulta, respeitando a tabela de preços dos convênios fornecidos pelos Convênios.	Implementação	( )	(X)
NF 4.7 Campo Profissional do Agendamento de Consultas	O Campo vai ser inserido de acordo com a tabela de cadastro dos profissionais, vinculando o nome dos mesmos através de uma caixa de seleção.	Implementação	( )	(X)

Fonte: Própria, 2010.

Quadro 8: Requisito Manter Profissional

F5 Manter Profissional	Oculto ( )

Descrição: O Ator deverá fazer o cadastro do Profissional, para vincular as consultas e clientes relativos aos atendimentos feitos pelo médico ou dentista ao qual está vinculado a clinica ou não.				
Requisitos Não Funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF 5.1Campos do Cadastro do Profissional	O campo disponível para o cadastro é: Nome.	Implementação	(X)	( )
NF 5.2 Campo Nome da Manutenção de Profissionais	Campo de texto com no máximo 60 caracteres alfabéticos.			

Fonte: Própria, 2010.

Quadro 9: Requisito Solicitar exames

F6 Solicitar Exames	Oculto ( )			
Descrição: O Profissional poderá solicitar exames, tendo o controle das solicitações e acompanhamento dos pedidos.				
Requisitos Não Funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF 6.1 Campos do Cadastro da Solicitação de Exames	Os campos disponíveis para o cadastro são: Convênio, Exame, Valor do Exame, Data de Pedido, Realização do Exame, Horário do	Implementação	(X)	( )
NF 6.2 Campo Convênio da Solicitação de Exames	O campo será gerado da tabela de Cadastro de Convênios.	Implementação	()	(X)
NF 6.2 Campo Exame da Solicitação de Exames	O campo será gerado da tabela de cadastro de Exames.	Implementação	()	(X)

Nome	**Restrição**	**Categoria**	**Desejáve**	**Permanente**

NF 6.3 Campo Valor do Exame da Solicitação de Exames	Campo de texto com no máximo 6 caracteres alfanuméricos, e poderá depender ou não dos preços dos convênios.	Implementação	(X)	( )
NF 6.4 Campo Data do Pedido da Solicitação de Exame	O Campo vai ser gerado de acordo a data de vinculação do Exame.	Implementação	(X)	( )
NF 6.5 Campo Realização do Exame da Solicitação de Exames	A data vai ser definida pelo profissional de acordo com as datas disponíveis pelos convênios vinculados à Clínica.	Implementação	(X)	( )
NF 6.6 Campo Horário do exame da Solicitação de Exames	O Horário vai ser definido pelo profissional de acordo com as datas disponíveis pelos convênios.	Implementação	(X)	( )

Fonte: Própria, 2010.

Quadro 10: Requisito Manter Convênio

F7 Manter Convênio	Oculto ( )			
Descrição: O Ator deverá fazer o cadastro do Profissional, para vincular as consultas e clientes relativos aos atendimentos feitos pelo médico ou dentistas ao qual está vinculado a clinica ou não.				
Requisitos Não Funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF 7.1 Campos do Cadastro do Convênio	Os campos disponíveis para o cadastro são: Nome, Responsável.	Implementação	(X)	( )
NF 7.2 Campo Nome da Manutenção de Convênios	Campo de texto com no máximo 50 caracteres alfabéticos.	Implementação	(X)	( )

Nome	Restrição	Categoria	Desejável	Permanente
NF 7.3 Campo Responsável da Manutenção de Convênios	Campo de texto com no máximo 05 caracteres alfabéticos.	Implementação	(X)	( )

Fonte: Própria, 2010.

Quadro 11: Requisito Manter Especialidades

F8 Manter Especialidades	Oculto ( )			
Descrição: O Ator deverá fazer o cadastro das especialidades oferecidas pela clínica ou oferecidas pelos seus convênios. Onde, depois serão vinculadas para os exames e profissionais pertinentes às áreas de atuação.				
Requisitos Não Funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF 8.1 Campos do Cadastro Especialidades	Os campos disponíveis para o cadastro são: Nome.	Implementação	(X)	( )
NF 8.2 Campo Nome da Manutenção de Especialidades	Campo de texto com no máximo 50 caracteres alfabéticos.	Implementação	(X)	( )

Fonte: Própria

Quadro 12: Requisito Manter Dependente

F9 Manter Dependente	Oculto ( )			
Descrição: O Ator deverá fazer o vínculo do Dependente, caso o cliente tenha algum, relacionado ao cliente que dependa ou que seja menor de idade para ter acesso aos serviços da clínica.				
Requisitos Não Funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF 9.1 Campos do Cadastro Dependente	Os campos disponíveis para o cadastro são: Nome e Idade.	Implementação	(X)	( )
NF 9.2 Campo Nome da Manutenção do Dependente	Campo de texto com no máximo 50 caracteres alfabéticos.	Implementação	(X)	( )
NF 9.3 Campo Idade da Manutenção do Dependente	Campo de texto com no máximo 2 caracteres alfanuméricos.	Implementação	(X)	( )

Fonte: Própria, 2010.

Quadro 13: Requisito Manter Usuários

F10 Manter Usuários	Oculto ( )			
Descrição: O Sistema disponibilizará o cadastro de Usuários para acesso ao sistema vinculado às permissões para o acesso.				
Requisitos Não Funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF 10.1 Campos do Cadastro de Usuário	Nome Do Funcionário, Data de Nascimento, Endereço, Número, Bairro, Complemento, Cidade, Estado, CEP, Telefone, Celular, E-mail, Site, Pai, Mãe, RG, CPF, Título Eleitoral, Formação, Idiomas, Nome Usuário, Senha, Observações.	Implementação	( )	(X)
NF 10.2 Campo Nome do Funcionário do Cadastro de Usuário	Campo de texto com no máximo 100 caracteres alfabéticos	Implementação	( )	(X)
NF 10.3 Campo Data de Nascimento do Cadastro de Usuário	Campo de texto com dia contendo dois caracteres, mês vinculado de uma caixa de seleção e ano com 4 caracteres.	Implementação	( )	(X)
NF 10.4 Campo Endereço do Cadastro de Usuário	Campo de texto com no máximo 50 caracteres alfabéticos	Implementação	()	(X)
NF 10.5 Campo Número do Cadastro de Usuário	Campo de texto com no máximo 5 caracteres numéricos	Implementação	( )	(X)

Nome	Restrição	Categoria	Desejável	Permanente
NF 10.6 Campo Bairro do Cadastro de Usuário	Campo de texto com no máximo 50 caracteres alfabéticos	Implementação	( )	(X)
NF 10.7 Campo Complemento do Cadastro de Usuário	Campo de texto com no máximo 10 caracteres alfabéticos	Implementação	( )	(X)
NF 10.8 Campo Cidade do Cadastro de Usuário	Campo de texto com no máximo 100 caracteres alfabéticos	Implementação	( )	(X)
NF 10.9 Campo Estado do Cadastro de Usuário	Campo de texto com no máximo 2 caracteres alfabéticos, vinculado de uma caixa de seleção.	Implementação	( )	(X)
NF 10.10 Campo CEP do Cadastro de Usuário	Campo de texto com no máximo 9 caracteres alfanuméricos.	Implementação	( )	(X)
NF 10.11 Campo Telefone do Cadastro de Usuário	Campo de texto com no máximo 30 caracteres alfanuméricos.	Implementação	( )	(X)
NF 10.12 Campo Celular do Cadastro de Usuário	Campo de texto com no máximo 30 caracteres alfanuméricos.	Implementação	( )	(X)
NF 10.13 Campo E-mail do Cadastro de Usuário	Campo de texto com no máximo 100 caracteres alfabéticos.	Implementação	( )	(X)
NF 10.14 Campo Site do Cadastro de Usuário	Campo de texto com no máximo 100 caracteres alfabéticos.	Implementação	( )	(X)

Nome	Restrição	Categoria	Desejável	Permanente
NF 10.15 Campo Pai do Cadastro de Usuário	Campo de texto com no máximo 50 caracteres alfabéticos	Implementação	( )	(X)
NF 10.16 Campo Mãe do Cadastro de Usuário	Campo de texto com no máximo 50 caracteres alfabéticos	Implementação	( )	(X)
NF 10.17 Campo RG do Cadastro de Usuário	Campo de texto com no máximo 15 caracteres alfanuméricos	Implementação	( )	(X)
NF 10.18 Campo CPF do Cadastro de Usuário	Campo de texto com no máximo 11 caracteres numéricos	Implementação	( )	(X)
NF 10.19 Campo Título Eleitoral do Cadastro de Usuário	Campo de texto com no máximo 20 caracteres numéricos	Implementação	( )	(X)
NF 10.20 Campo Formação do Cadastro de Usuário	Campo de texto com no máximo 30 caracteres alfabéticos	Implementação	( )	(X)
NF 10.21 Campo Idiomas do Cadastro de Usuário	Campo de texto com no máximo 30 caracteres alfabéticos	Implementação	( )	(X)
NF 10.22 Campo Nome Usuário do Cadastro de Usuário	Campo de texto com no máximo 11 caracteres alfabéticos	Implementação	( )	(X)
NF 10.23 Campo Senha do Cadastro de Usuário	Campo de texto com no máximo 10 caracteres alfanuméricos	Implementação	( )	(X)

Nome	Restrição	Categoria	Desejável	Permanente
NF 10.24 Campo Observações do Cadastro de Usuário	Campo de texto com no máximo 100 caracteres alfanuméricos	Implementação	( )	(X)

Fonte: Própria, 2010.