



UNIVERSIDADE DO SUL DE SANTA CATARINA
DAIANE DORNER

DESENVOLVIMENTO WEB PARA FORMULÁRIOS DINÂMICOS NO
SISTEMA DE FISCALIZAÇÃO PARA ÁREA DA SAÚDE

Palhoça
2015

DAIANE DORNER

**DESENVOLVIMENTO WEB PARA FORMULÁRIOS DINÂMICOS NO
SISTEMA DE FISCALIZAÇÃO PARA ÁREA DA SAÚDE**

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
da Universidade do Sul de Santa Catarina, como
requisito parcial à obtenção do título de Bacharel
em Ciência da Computação.

Orientador: Prof. Maria Inés Castiñeira, Dra.

Palhoça

2015

DAIANE DORNER

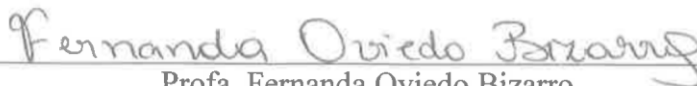
**DESENVOLVIMENTO WEB PARA FORMULÁRIOS DINÂMICOS NO
SISTEMA DE FISCALIZAÇÃO PARA ÁREA DA SAÚDE**

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo Curso de Graduação em Ciência da Computação da Universidade do Sul de Santa Catarina.

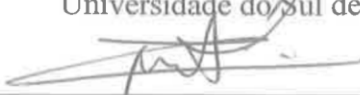
Florianópolis, 18 de junho de 2015.



Professora e orientadora Maria Inês Castiñeira, Dra.
Universidade do Sul de Santa Catarina



Profa. Fernanda Oviedo Bizarro.
Universidade do Sul de Santa Catarina



Prof. Ricardo Ribeiro Assink.
Universidade do Sul de Santa Catarina

AGRADECIMENTOS

Primeiramente a Deus, que permitiu que tudo isso acontecesse, ao longo de minha vida, e não somente nestes anos como universitária, mas que em todos os momentos é o maior Mestre e Pai que alguém pode conhecer.

À Universidade do Sul de Santa Catarina, pela oportunidade de fazer o curso.

Agradeço a todos os professores por me proporcionar o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender. A palavra mestre, nunca fará justiça aos professores dedicados aos quais sem nominar terão os meus eternos agradecimentos.

Às professoras Dra. Maria Inês Castiñeira (Orientadora) e Fernanda Oviedo (Co-orientadora), pelo empenho dedicado à elaboração deste trabalho e amizade conquistada e cultivada durante estes anos.

Aos meus pais, pelo amor, incentivo, dedicação e apoio incondicional.

Obrigada minhas irmãs e sobrinhos, que nos momentos de minha ausência dedicados ao estudo superior, sempre fizeram entender que o futuro é feito a partir da constante dedicação no presente!

Meus agradecimentos aos amigos, companheiros de trabalhos e irmãos na amizade que fizeram parte da minha formação e que vão continuar presentes em minha vida com certeza.

Agradeço a toda equipe da GBR Consultoria e Sistemas, pelo apoio recebido, grandes aprendizados e amizades.

A todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

“Que os vossos esforços desafiem as
impossibilidades, lembrai-vos de que as
grandes coisas do homem foram
conquistadas do que parecia impossível.”
(Charles Chaplin).

RESUMO

Este trabalho tem por finalidade o desenvolvimento de um sistema que permita atualizar de forma eficiente e eficaz informações importantes para a vigilância na área da saúde. Trata-se de um software que manipula arquivos em formato XML, chamados de formulários dinâmicos ou roteiros, que contém as normas e recomendações estipuladas pelo órgão competente (Conselho Federal de Medicina). Estes arquivos formam a base estrutural de um sistema de fiscalização utilizado em âmbito nacional para avaliar todas as entidades de saúde e garantir melhores condições e segurança para a população brasileira. As informações nestes arquivos estão dispostas em forma de questionários, seguindo leis e recomendações de Resoluções que são frequentemente modificadas. Estes arquivos são transformados em telas no sistema de fiscalização permitindo a interação entre os fiscais e o sistema, no momento de uma avaliação na área da saúde. Um trabalho realizado manualmente, lentamente e de forma desgastante por parte do profissional que desempenha esta função, transforma-se em um processo informatizado, ágil e organizado diminuindo consideravelmente o tempo e custo de sua realização. Este projeto é de grande importância para o bom desempenho da vigilância na área da saúde, pois proporciona atualizações nos dados de avaliação conforme as leis que regem a profissão e as instituições desta área. Após a definição da problemática e dos objetivos foi realizado um levantamento bibliográfico sobre o processo de fiscalização na área da saúde, a entidade responsável por administrar estas ações, sua importância para a sociedade, a fim de compreender a importância do desenvolvimento deste sistema. Também foi especificado o sistema aqui proposto, foi contextualizado o âmbito onde ele se aplica e explicado o objetivo de sua criação, para facilitar as constantes alterações de formulários dinâmicos utilizados no sistema de fiscalização. Também são apresentadas as melhorias que este sistema proporciona aos seus usuários em suas rotinas de trabalho. A modelagem do sistema é apresentada em UML através de diagramas de classe, requisitos funcionais e não funcionais, regras de negócios e demais diagramas que auxiliam na estruturação e entendimento do projeto. As tecnologias utilizadas, como por exemplo, a linguagem de programação Java, JQuery, Ajax, CSS, HTML, Bootstrap, XML, Javascript, forneceram ótimos recursos para o desenvolvimento e utilização do software, permitindo alta disponibilidade, usabilidade e eficácia. Após a validação da ferramenta verificou-se que esta atende às necessidades do cliente e seus usuários de acordo com os requisitos iniciais verificados e analisados.

Palavras-Chave: Formulário Dinâmico. Fiscalização na Saúde. Sistema Web de Apoio à Saúde.

LISTA DE ILUSTRAÇÕES

Figura 1: Tipos de Sistema de Informação	19
Figura 2: Fluxograma.	38
Figura 3: Arquitetura do Sistema.	39
Figura 4: Processo de Fiscalização	43
Figura 5: Modelagem de Processo.	44
Figura 6: Requisitos Funcionais Módulos Acessar e Legislação.	47
Figura 7: Requisitos Funcionais Módulos Item e Grupo.....	48
Figura 8: Requisitos Funcionais Módulos Item e Grupo.....	49
Figura 9: Requisitos Funcionais Módulos Irregularidade e Auto Avaliação.	51
Figura 10: Requisitos Não Funcionais.	52
Figura 11: Modelo de Domínio (parte1)	54
Figura 12: Modelo de Domínio (parte 2)	55
Figura 13: Diagrama de classes <i>Entity</i> 1.	57
Figura 14: Diagrama de classes <i>Entity</i> 2.	57
Figura 15: Diagrama de classes DAO.	58
Figura 16: Casos de Uso.....	60
Figura 17: Diagrama de Robustez para Tela <i>Login</i>	61
Figura 18: Diagrama de Sequência para Tela <i>Login</i>	61
Figura 19: Diagrama de Robustez para Tela Publicar Roteiros.	63
Figura 20: Diagrama de Sequência para Tela Publicar Roteiros.....	63
Figura 21: Diagrama de Robustez para Tela Gerenciar Roteiros.....	65
Figura 22: Diagrama de Robustez para Tela Cadastrar Formulário.....	65
Figura 23: Diagrama de Sequência para Tela Cadastrar Formulário.	66
Figura 24: Protótipo de Tela Inicial.....	67
Figura 25: Protótipo de Tela Gerenciar Roteiro.	67
Figura 26: Protótipo de Tela Publicar Roteiro.	68
Figura 27: Protótipo de Tela Publicar Roteiro.	68
Figura 28: Modelagem de Dados parte 1.	69
Figura 29: Modelagem de Dados parte 2.	70
Figura 30: Tecnologias utilizadas.....	71

Figura 31: Arquitetura de aplicação J2EE.....	74
Figura 32: Elementos de uma URL.	75
Figura 33: Definição de Filtro.	79
Figura 34: Sintaxe da regra CSS.	81
Figura 35: Hierarquia de objetos.	84
Figura 36: Principais propriedades do XMLHttpRequest.	86
Figura 37: Possíveis retornos da propriedade “readyState”.	87
Figura 38: Os métodos mais utilizados em Ajax.....	87
Figura 39: Exemplo de tabela de banco de dados.	90
Figura 40: Recursos Bootstrap.	94
Figura 41: Tela <i>Login</i>	96
Figura 42: Tela Inicial.	97
Figura 43: Tela Gerenciar Roteiros – Aba Roteiro.....	97
Figura 44: Tela Gerenciar Roteiros – Aba Grupo.	98
Figura 45: Tela Gerenciar Roteiros – Aba Item.	98
Figura 46: Tela Gerenciar Roteiros – Aba Legislação.	99
Figura 47: Tela Cadastro de Formulário (parte 1).....	100
Figura 48: Tela Cadastro de Formulário (parte 2).....	100
Figura 49: Tela Cadastro de Legislação.	101
Figura 50: Tela Publicar Roteiros.....	101

LISTA DE QUADROS

Quadro 1: Requisitos Funcionais Módulo Acessar.	46
Quadro 2: Requisitos Funcionais Módulo Legislação.....	46
Quadro 3: Requisitos Funcionais Módulo Item.....	47
Quadro 4: Requisitos Funcionais Módulo Grupo.....	48
Quadro 5: Requisitos Funcionais Módulo Formulário.....	49
Quadro 6: Requisitos Funcionais Módulo Publicar.....	49
Quadro 7: Requisitos Funcionais Módulo Irregularidade.	50
Quadro 8: Requisitos Funcionais Módulo Auto Avaliação.....	50
Quadro 9: Requisitos Não Funcionais.....	52
Quadro 10: Regras de Negócios.	53
Quadro 11: Casos de Uso 001 – Acessar Sistema.	60
Quadro 12: Casos de Uso 003 – Publicar Roteiros.	62
Quadro 13: Casos de Uso 004 – Cadastrar Formulário.....	64
Quadro 14: Conceitos em Servlets.	77

SUMÁRIO

AGRADECIMENTOS.....	4
RESUMO	6
1 INTRODUÇÃO	12
1.1 APRESENTAÇÃO DA PROBLEMÁTICA	13
1.2 OBJETIVOS	14
1.2.1 Objetivo Geral.....	14
1.2.2 Objetivos Específicos	15
1.3 JUSTIFICATIVAS	15
2 REVISÃO BIBLIOGRÁFICA	18
2.1 INFORMÁTICA NA SAÚDE.....	18
2.1.1 Classificação de Sistemas de Informação na área da saúde.....	20
2.1.2 Conselho Federal de Medicina.....	23
2.1.3 Fiscalização da Atenção Básica.....	25
2.1.4 Software de Fiscalização das Unidades de Saúde	27
2.2 ESPECIFICAÇÃO DO SISTEMA.....	28
2.2.1 Definição dos Elementos do Sistema	29
3 MÉTODO.....	33
3.1 CARACTERIZAÇÃO DO TIPO DE PESQUISA	33
3.2 ESTRUTURA DA MONOGRAFIA	34
3.2.1 Pesquisa Bibliográfica	35
3.2.2 Pesquisa exploratória.....	35
3.2.3 Pesquisa Qualitativa	36
3.2.4 Pesquisa Aplicada	36
3.3 ETAPAS METODOLÓGICAS	37
3.4 PROPOSTA DA SOLUÇÃO	38
3.5 DELIMITAÇÕES	40
4 MODELAGEM	41
4.1 MODELAGEM DE PROCESSO DE NEGÓCIOS.....	42
4.2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS	45
4.2.1 Requisitos Funcionais	46
4.2.2 Requisitos Não Funcionais	51

4.3	REGRAS DE NEGÓCIOS	53
4.4	MODELAGEM DE DOMÍNIO	54
4.5	MODELAGEM DE CLASSES	55
4.6	CASOS DE USO	58
4.6.1	Protótipos de Tela	66
4.7	MODELAGEM DE DADOS	69
4.8	CONSIDERAÇÕES FINAIS	70
5	DESENVOLVIMENTO.....	71
5.1	TECNOLOGIAS UTILIZADAS.....	71
5.1.1	Linguagem de Programação Java	72
5.1.2	Tecnologias do Java EE.....	73
5.1.3	Desenvolvimento de Aplicações Web	74
5.1.4	Servlets.....	76
5.1.5	Filtros.....	77
5.1.6	Java Server Pages	79
5.1.7	CSS - Folha de Estilo em Cascata.....	80
5.1.8	Javascript.....	82
5.1.9	JQuery.....	84
5.1.10	Ajax.....	86
5.1.11	Oracle Database 11G	89
5.1.12	Java Persistence API.....	91
5.1.13	JBoss.....	93
5.1.14	Bootstrap.....	93
5.2	HISTÓRICO E AVALIAÇÃO DO DESENVOLVIMENTO.....	95
5.3	APRESENTAÇÃO DO SISTEMA	96
6	CONCLUSÕES E TRABALHOS FUTUROS.....	102
6.1	CONCLUSÕES	102
6.2	TRABALHOS FUTUROS	103
	REFERÊNCIAS	105
	APÊNDICES.....	111
	APÊNDICE A – CRONOGRAMA.....	112
	APÊNDICE B – FORMULÁRIO DINÂMICO.....	113

1 INTRODUÇÃO

A cada dia a medicina vem contando mais com a ajuda da informática para seus propósitos rotineiros. Já em 1990 a área de Informática Médica era definida por Blois e Shortliffe (1990) apud SBSIS (2014)0020 como “um campo de rápido desenvolvimento científico que lida com armazenamento, recuperação e uso da informação, de dados e de conhecimentos biomédicos para a resolução de problemas e tomada de decisão”. Os conhecimentos de computação que antes eram utilizados apenas para criação de ferramentas de diagnóstico e tratamento de pacientes, agora também começam a ser aplicados para gerenciamento e controle das entidades de saúde.

A saúde já pode beneficiar-se de ferramentas bastante avançadas para auxiliar em suas tarefas administrativas. A necessidade de sistemas que facilitem a fiscalização das entidades de saúde e substituam os formulários em papel é um fator marcante, desta forma é nessa direção que o presente trabalho pode ser enquadrado.

O Conselho Federal de Medicina tem sua definição exposta no site portal.cfm.org.br (2010) como “um órgão que possui atribuições constitucionais de fiscalização e normatização da prática médica. Criado em 1951, sua competência inicial reduzia-se ao registro profissional do médico e à aplicação de sanções do Código de Ética Médica”. Este órgão luta em defesa da saúde da população e dos interesses da classe médica.

Ao defender os interesses corporativos dos médicos, o CFM empenha-se em defender a boa prática médica, o exercício profissional ético e uma boa formação técnica e humanista, convicto de que a melhor defesa da medicina consiste na garantia de serviços médicos de qualidade para a população. Além de zelar pelo desempenho ético da medicina e pelo bom conceito da profissão, o CFM organiza uma série de atividades e presta alguns serviços aos médicos e à sociedade brasileira. (CFM, 2014).

Dentre os serviços prestados pelo Conselho podemos destacar a “Normatização da Profissão” que disponibiliza toda a legislação referente à área médica. Entre as legislações disponíveis pode-se encontrar as resoluções, as quais são utilizadas pelo sistema de fiscalização da área da saúde, desenvolvido para auxiliar os Conselhos Regionais de Medicina (CRM) a realizar as fiscalizações necessárias para o controle de autorização de funcionamento dos serviços médicos de qualquer natureza.

A Saúde é uma das áreas onde há maior necessidade de informação para a tomada de decisões. A Informática Médica é o campo científico que lida com recursos, dispositivos e métodos para otimizar o armazenamento, recuperação e gerenciamento de informações biomédicas. O crescimento da Informática Médica como uma disciplina deve-se, em grande parte: aos avanços nas tecnologias de computação e comunicação, à crescente convicção de que o conhecimento médico e as informações sobre os pacientes são ingerenciáveis por métodos tradicionais baseados em papel, e devido à certeza de que os processos de acesso ao conhecimento e tomada de decisão desempenham papel central na Medicina moderna. (SBSIS, 2014).

O Conselho Federal de Medicina já tem a posse de um sistema aproveitado pelos Conselhos Regionais de Medicina que tem por função a fiscalização das entidades de saúde, sendo estas Clínicas, Postos de Saúde ou Hospitais, entidades públicas ou privadas. Dependendo do tipo de entidade a ser fiscalizada, várias regulamentações devem ser observadas. Estas regulamentações também devem ser atualizadas conforme a Legislação pertinente vai sendo alterada.

Neste trabalho será apresentado um sistema criado para configurar os formulários dinâmicos utilizados pela ferramenta de fiscalização para área da saúde. Será mostrado como ele foi desenvolvido, apontando quais técnicas foram utilizadas, as ferramentas, a linguagem de programação e o resultado final do desenvolvimento.

1.1 APRESENTAÇÃO DA PROBLEMÁTICA

O sistema de fiscalização utilizado em benefício do Conselho Federal de Medicina é composto por vários formulários dinâmicos que compõem sua estrutura funcional. Os dados que compõem estes formulários são armazenados em diversos arquivos em formato XML. A atualização destes dados é feita manualmente por profissionais responsáveis por este trabalho. A demanda pelas alterações depende da criação ou alteração de Resoluções criadas pelo Conselho Federal de Medicina (CFM) que disciplina os departamentos de fiscalização nos Conselhos Regionais de Medicina (CRM)'s estabelecendo critérios mínimos para o funcionamento de qualquer serviço médico, podendo vedar o funcionamento daqueles que não estiverem de acordo com os regulamentos.

Ao fornecer a manutenção ao sistema apresentado acima, pode-se constatar a dificuldade e lentidão no processo de modificação e validação dos dados que formam os formulários dinâmicos. A necessidade de um sistema para otimizar, tanto em velocidade quanto em qualidade, a atualização destes formulários é fato presente na vida dos responsáveis por este trabalho.

A proposta de criação de um sistema que permita a manutenção facilitada dos formulários dinâmicos que estruturam o sistema de fiscalização do Conselho Federal de Medicina, envolvendo ações como, por exemplo, cadastrar, alterar e exportar em formato XML preenche a necessidade dos responsáveis por alterar os formulários em questão. Alterações estas que são realizadas frequentemente devido ao alto índice de mudanças nas resoluções por parte dos responsáveis pelas diretrizes estabelecidas nestes documentos.

Nas sociedades atuais a informação tem grande importância no âmbito da saúde, de modo a ter uma preocupação crescente com o desenvolvimento de Sistemas de Informação que permitam a maximização da gestão e promovam a melhoria da qualidade dos cuidados de saúde. As dificuldades sentidas na gestão têm levado governos e entidades responsáveis a demonstrar um crescente envolvimento nos processos de desenvolvimento de sistemas de informação tendo em vista a melhoria da qualidade dos cuidados. (SOUZA, FRADE, MENDONCA, 2005).

1.2 OBJETIVOS

Neste trabalho os objetivos são apresentados divididos em: geral e específicos.

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um sistema web que auxilie e melhore o desempenho de configuração de formulários dinâmicos utilizados no sistema de fiscalização do Conselho Federal de Medicina, disponibilizando ferramentas de fácil acessibilidade e maior garantia de segurança dos dados, armazenando estes em um banco de

dados relacional e permitindo a atualização, criação, remoção e organização das informações contidas nos formulários através do sistema proposto.

1.2.2 Objetivos Específicos

Os objetivos específicos do trabalho são:

- ✓ Aplicar os conceitos da programação web e Java para criação de um sistema.
- ✓ Pesquisar sobre o Conselho Federal de Medicina, seu relacionamento com os Conselhos Regionais de Medicina e a fiscalização das entidades de saúde.
- ✓ Pesquisar a funcionalidade do sistema de fiscalização na área da saúde.
- ✓ Avaliar o trabalho realizado pelos profissionais que desempenham a função de manutenção dos formulários dinâmicos que estruturam o sistema de fiscalização.
- ✓ Realizar ações pertinentes na base de dados, como criação, transformação, carga e extração dos dados relevantes ao sistema.
- ✓ Agrupar os dados em formulários permitindo a extração dos mesmos em formato suportado pelo sistema de fiscalização.

1.3 JUSTIFICATIVAS

A tecnologia moderna, criada pelo homem a serviço do homem, tem contribuído em larga escala para a solução de problemas antes insolúveis e que pode reverter em melhores condições de vida e saúde para o paciente. (BARRA et. al., p. 423, 2006)

Para atender as demandas de atualização dos formulários eletrônicos de fiscalização utilizados pelos Conselhos Regionais de Medicina (CRM's), não basta apenas

criar um sistema que seja capaz de realizar as ações desejadas. Além disto, ele deve possuir uma interface amigável, de fácil entendimento e utilização por quem o adquire. Não basta apenas armazenar as informações, mas deve mantê-las armazenadas em local seguro. Deve ser seguro e poder ser acessado apenas por pessoas autorizadas. Também precisa ter garantia de bom desempenho no processamento para o cadastro ou edição de um formulário.

A autora, no momento do desenvolvimento desta monografia, trabalha em uma empresa que presta serviços de desenvolvimento para o Conselho Regional de Medicina de SC e também para o Conselho Federal de Medicina. A percepção das necessidades da entidade foi um dos fatores que motivou o desenvolvimento deste trabalho.

O desenvolvimento do sistema proposto tem fundamental importância para o Conselho Federal de Medicina, pois além de facilitar a formatação dos formulários que constituem a base estrutural do sistema de fiscalização da área da saúde, permite que os próprios utilizadores do sistema de fiscalização atualizem os dados da Resolução, transformando estas demandas de alterações independentes de profissionais qualificados a este requisito, ou seja, são profissionais que não necessitam de grande experiência em tecnologia para utilizar o sistema, devido à este ser desenvolvido com alta usabilidade. Ele também permite a inclusão de novos formulários de acordo com suas necessidades e privilégios.

O sistema proposto neste trabalho de conclusão de curso transforma um trabalho manual e dependente de profissionais qualificados ao serviço, em ferramentas acessíveis, ágeis e independentes de qualificação para sua utilização, garantindo a segurança dos dados e eficácia nas manutenções dos formulários.

Atualmente, o trabalho de atualização destes formulários dinâmicos é realizado por um profissional da empresa de tecnologia que presta serviços ao Conselho Federal de Medicina, e que desenvolveu o sistema de fiscalização. Este profissional compara os formulários individualmente com a nova Resolução estabelecida, observando linha por linha, comparando os dados e modificando diretamente nos arquivos XML.

O Sistema de Fiscalização é constituído de aproximadamente 50 formulários dinâmicos que são transformados em telas de questionários. Cada formulário contém um determinado número de questões, em média 20 questões cada. O trabalho de atualização de cada arquivo é lento, levando cerca de uma semana para sua realização.

A implantação do sistema aqui descrito terá como consequências imediatas a diminuição de custos, pois não será mais necessário que o trabalho seja realizado por

profissional especializado contratado e a agilidade na alteração destes formulários dinâmicos, pois o trabalho que atualmente é realizado em uma semana, com a utilização do sistema proposto deverá ser realizado em apenas um dia.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentados os diversos assuntos que dão o embasamento teórico para o desenvolvimento deste trabalho. Utilizando de fontes de diversos autores, são abordadas questões que ajudarão a descrever as teorias subjacentes ao trabalho.

2.1 INFORMÁTICA NA SAÚDE

A maioria das empresas depende do uso de tecnologias de informação para executar vários aspectos de seus negócios. A tecnologia da informação permite às empresas levar adiante uma variedade de tarefas de forma eficiente e eficaz. Isso acontece também na área da saúde.

A proposta de um sistema informatizado (seja em ambiente web ou não) transmitido ao usuário de forma clara e fácil é um desafio contínuo, sendo necessários constantes testes e pesquisas sobre o comportamento do usuário a fim de entender diferenças de perfis, limitações, formas de utilização e outros. São evidentes as expectativas do usuário quanto à necessidade de sistemas ágeis, fáceis e que consumam pouco tempo de operação. (PEREIRA e PAIVA, p. 123, 2011).

Sistemas informatizados são essenciais na área da saúde, mas sua eficácia é diretamente proporcional à sua usabilidade. A usabilidade reduz falhas na segurança, intencionais ou não, oriundas do usuário. E em saúde, a necessidade de segurança é ainda mais crítica do que em outras áreas, visto que envolve o bem estar e a vida humana. Como a segurança da informação abrange todos os colaboradores da organização, é necessário que todos os profissionais envolvidos na implantação de um sistema, se familiarizem com a engenharia da usabilidade e se conscientizem de que ela é uma poderosa ferramenta para reduzir riscos na segurança dos sistemas informatizados visto que não existe utilidade para um sistema inseguro, assim como não há necessidade de segurança em um sistema que não seja utilizado. (PEREIRA e PAIVA, p. 128, 2011).

De acordo com Denis Alcides Rezende (2011), na atualidade, a “informática” se transforma em “Tecnologia da Informação (TI)” integrando os seus emergentes e modernos

recursos. O autor ainda relata que “A TI pode ser conceituada como o conjunto dos recursos tecnológicos e computacionais para a guarda de dados, geração e uso da informação e de conhecimentos”.

O mundo em que habitamos é um mundo criado pela atividade da ciência. Ela nos ajuda a criar objetos físicos, sistemas de transportes, sistemas de comunicação, sistemas educacionais, sistemas de saúde e novas tecnologias, bem como a organização da sociedade; nos ajuda a entender como funciona o mundo natural, e é também responsável pelo fornecimento dos meios que permitem a destruição de nosso planeta. Assim, o que está sendo chamado de ciência moderna é o resultado da Revolução Científica que ocorreu no século XVII, mas desde o século VI, antes da era cristã, os gregos já manifestavam o desejo de conhecer o mundo como é. (AUDY et. al. 2005, p.14).

Todas as organizações possuem um SI com o propósito de auxiliar no cumprimento da sua missão. Esse sistema é normalmente composto de diversos subsistemas de natureza conceptual idêntica à daquele que integram, mas com características específicas quanto à sua finalidade e justificação, quanto ao tipo das tecnologias utilizadas e quanto ao nível dos processos ou natureza das pessoas que envolvem. (AMARAL p. 30, 1994).

A figura 01 apresenta um exemplo de classificação baseada numa mistura de critérios, são identificados os seis tipos de Sistemas da Informação expostos na imagem. (ALTER, 1992 apud AMARAL p. 31, 1994).

Figura 1: Tipos de Sistema de Informação

Tipo de sistema	Definição
Sistema Processamento de Transacções	Recolhe e mantém informação sobre transacções e controla pequenas decisões que fazem parte das transacções
Sistema de Informação de Gestão	Converte informação sobre transacções em informação para a gestão da organização
Sistema de Apoio à Decisão	Ajuda os utilizadores na tomada de decisões não estruturáveis fornecendo-lhes informação, modelos e ferramentas para analisar a informação
Sistema de Informação para Executivos	Fornece aos gestores, de modo muito interactivo e flexível, acesso a informação geral para a gestão da organização
Sistema Pericial	Suporta os profissionais do desenho, diagnóstico e avaliação de situações complexas que requerem conhecimento especializado em áreas bem definidas
Sistema de Automação de Escritório	Mantem as tarefas de comunicação e processamento de informação características de ambiente de escritório

Autor: ALTER, 1992 apud AMARAL p. 31, 1994.

Sistemas de apoio à decisão (SAD) – proporcionam suporte às decisões menos rotineiras e estruturadas, e não facilmente especificadas com antecipação. Os formulários criados e gerenciados pelo sistema deste trabalho fazem parte deste tipo de tecnologia da informação, pois são utilizados com a finalidade de gerar relatórios que controlam entidades empresariais. Exemplo deste tipo de sistema: Programa Risco Cirúrgico MED, avalia tanto o risco cardiológico e risco pulmonar como outros, facilitando a elaboração do laudo pré-operatório para o clínico ou cardiologista. O Risco Cirúrgico MED é um software que permite a padronização deste procedimento médico. Utilizando as tabelas de EMAPO, ACP (American College Physicians), oldman, ASA, Detsky, Larsen e Lee, possibilita uma rápida emissão do laudo de risco cirúrgico sendo este completamente customizado. (AMATO SOFTWARE, 2014).

Este trabalho enquadra-se na classificação de Sistema de Apoio à Decisão, contribuindo à Entidade ajuda na tomada de decisões importantes, expandindo suas capacidades, armazenando os dados necessários ao bom funcionamento de seu trabalho rotineiro de fiscalização permitindo alteração facilitada de informações pertinentes ao sistema. Com uma interface bastante simplificada, busca fornecer ferramentas que minimizem o trabalho manual realizado por profissionais que prestam suporte ao Sistema de Fiscalização do Conselho Federal de Medicina.

2.1.1 Classificação de Sistemas de Informação na área da saúde

Todas as atividades realizadas por um Sistema de Saúde geram dados que podem produzir informações. Todas estas atividades são realizadas em determinados tipos de unidades de produção ou unidades operacionais (laboratório, almoxarifado, unidade de saúde, setor de finanças etc.) que compõem o Sistema de Saúde e devem contar com informações que subsidiem o processo de planejamento, controle, avaliação e redirecionamento do que vem sendo produzido. (FERREIRA p. 6, 1999).

Existem dados e informações gerados nessas diferentes unidades operacionais que interessam não só a própria unidade, mas a todo Sistema de Saúde. São informações

consideradas estratégicas, voltadas para uma avaliação permanente das respostas que estão sendo produzidas e do impacto obtido sobre a situação de saúde. (FERREIRA p. 6, 1999).

Perez e Zwicker (2010, p.179) afirmam que a informática médica é o campo científico que lida com recursos, dispositivos e métodos para aperfeiçoar o armazenamento, a recuperação e o gerenciamento e informações biomédicas. Segundo Helms, Moore e Ahmadi (2008) citado por Perez e Zwicker (2010, p.179):

O uso de Sistemas de informação na saúde oferece importantes potenciais: incremento da segurança do paciente, maior eficiência operacional e infraestrutura de TI já existente na maioria das organizações. Mas o uso também é permeado de fraquezas relevantes: falta de integração de sistemas, lenta adoção de tecnologia da informação e resistência ao uso de novas tecnologias e redesenho de processos.

Um Sistema de Informação em benefício da Saúde, é um conjunto de componentes que atuam de forma integrada, através de mecanismos de coleta, processamento, análise e transmissão da informação necessária e oportunas para implementar processos de decisões no Sistema de Saúde. Seu propósito é selecionar dados pertinentes e transformá-los em informações para aqueles que planejam, financiam, provêm e avaliam os serviços de saúde. (FERREIRA p. 7, 1999).

Carvalho e Eduardo (1998) destacam que na gerência de serviços de saúde, é básica a necessidade de cadastro de pacientes, cadastro da população, cadastros de estabelecimentos, produção das atividades de saúde, conhecimento do perfil de doenças atendidas, da mortalidade, número de profissionais de saúde, número de consultórios, leitos, medicamentos utilizados, gastos efetuados e tantas outras informações. Com certeza, estas poderiam ser processadas manualmente, mas dificilmente de forma integrada e em tempo oportuno para a tomada de decisão.

A necessidade de fiscalizar Sistemas de Saúde no Brasil, assim como em qualquer outro país é tarefa primordial para que se obtenha um serviço prestado com qualidade e eficiência.

No processo de tomada de decisão torna-se essencial conhecer a origem das informações para garantir sua *fidedignidade*, bem como sua relevância, isto é, a importância delas no processo decisório. E, sobretudo devem estar oportunamente *disponíveis*, ou seja, facilmente acessíveis ou recuperáveis, para possibilitar uma resposta adequada, em tempo ideal, que permita subsidiar uma tomada de decisão. (CARVALHO; EDUARDO, 1998).

Segundo Carvalho e Eduardo (1998, p.27), conforme sua *Natureza*, os Sistemas de Informação em Saúde podem ser classificados como:

- *Sistemas de Informações Estatístico-epidemiológicas* - incluem o conhecimento da mortalidade e suas causas determinantes, do padrão de morbidade da população ou da demanda atendida pelos serviços, dos aspectos demográficos, sociais e econômicos e suas relações com a saúde da população, além da produção e utilização, da qualidade técnica dos procedimentos de saúde prestados e do grau de satisfação do usuário.

- *Sistemas de Informações Clínicas* – referem-se aos dados clínicos sobre o paciente, desde sua identificação, problemas de saúde relatados, diagnóstico médico, até exames clínicos, laboratoriais, radiológicos, gráficos, procedimentos cirúrgicos realizados ou medicamentos prescritos, entre outros.

- *Sistemas de Informações Administrativas* – refere-se ao controle de estoque, materiais, equipamentos, gestão financeira, entre outros.

Carvalho e Eduardo (1998, p.28) afirmam que “O conjunto das Informações *clínicas, estatístico-epidemiológicas e administrativas* compõem o Sistema de Apoio à Decisão em Saúde”.

As informações podem ainda ser classificadas segundo sua origem de produção, por tipo de instituição de saúde ou espaços coletivos, ou finalidade, tais como grupos populacionais de risco, ações programáticas, especialidades, etc. Alguns exemplos de classificação utilizando-se estes métodos são: Sistemas de Informação Ambulatorial; Sistemas de Informação Hospitalar; Sistemas de Informação de Mortalidade; Sistemas de Vigilância à Saúde, entre outros. (CARVALHO; EDUARDO, 1998).

De acordo com Sibeles (1999, p. 3), um Sistema de Saúde tem vários componentes (centros de saúde, ambulatorios especializados, farmácia, hospital etc.) que produzem vários tipos de ações (consultas médicas, vacinação, vigilância sanitária, etc.) segundo uma normatização estabelecida pela entidade responsável pelas unidades de saúde. Entretanto, para considerá-lo enquanto um sistema deve haver uma interligação e uma interação entre esses componentes que, se espera, resulte numa atenção organizada, produzindo respostas às necessidades de saúde de uma determinada população.

O processo de informatização deve considerar os objetivos e as necessidades de informação de interesse nacional, estadual, regional, municipal e local. Os atuais sistemas de abrangência nacional refletem o nível de desenvolvimento do modelo de atenção na maior parte do país. (FERREIRA p. 17, 1999).

Segundo esses conceitos, pode-se definir o sistema que será abordado neste trabalho como um sistema de informação administrativo de um órgão regulador da saúde. Esta entidade será descrita a seguir.

2.1.2 Conselho Federal de Medicina

Segundo o Conselho Federal de Medicina (CFM, 2014), CFM é um órgão que possui atribuições constitucionais de fiscalização e normatização da prática médica. Criado em 1951, sua competência inicial reduzia-se ao registro profissional do médico e à aplicação de sanções do Código de Ética Médica.

Nos últimos 50 anos, o Brasil e a categoria médica mudaram muito, e atualmente, as atribuições e o alcance das ações deste órgão estão mais amplas, extrapolando a aplicação do Código de Ética Médica e a normatização da prática profissional. Atualmente, o Conselho Federal de Medicina exerce um papel político muito importante na sociedade, atuando na defesa da saúde da população e dos interesses da classe médica. O órgão apresenta histórico de luta em prol dos interesses da saúde e do bem estar do povo brasileiro, sempre voltado para a adoção de políticas de saúde dignas e competentes, que alcancem a sociedade indiscriminadamente. (CFM, 2014)

Ao defender os interesses corporativos dos médicos, o CFM empenha-se em defender a boa prática médica, o exercício profissional ético e uma boa formação técnica e humanista, convicto de que a melhor defesa da medicina consiste na garantia de serviços médicos de qualidade para a população. (CFM, 2014)

O Conselho Federal de Medicina, com sede no Distrito Federal, tem jurisdição em todo território nacional e atua em conjunto com os Conselhos Regionais de Medicina, estes com jurisdição sobre as respectivas unidades federativas. Assim, os Conselhos formam uma autarquia dotada de personalidade jurídica de direito público, com autonomia administrativa e financeira. (CFM, 2014).

Para cumprir suas funções legais, os Conselhos Regionais funcionam como tribunais, apreciando denúncias contra médicos e instaurando processos ético-profissionais quando existem indícios de infração ética. As penas, na forma da lei, podem consistir em

advertência confidencial, censura confidencial, censura pública, suspensão do exercício profissional até 30 dias e cassação do exercício profissional. Das decisões dos Conselhos Regionais, cabe recurso ao Conselho Federal. (CFM, 2014).

De acordo com a entidade citada acima, além de defender o desempenho ético da medicina, o Conselho Federal de Medicina (CFM) organiza uma série de atividades e presta alguns serviços aos médicos e a sociedade brasileira, entre os quais pode-se destacar a Normatização da Profissão.

2.1.2.1 Normatização da Profissão

Segundo o CFM (2014), a Normatização da Profissão enquadra toda a Legislação referente à área médica, como Leis, Decretos, Pareceres, Resoluções. É dada maior importância a este último, pelo fato de ser utilizado como referência à criação dos formulários dinâmicos abordados neste Trabalho de Conclusão de Curso.

As Resoluções são atos normativos emanados dos plenários do Conselho Federal de Medicina (CFM) e de alguns dos Conselhos Regionais de Medicina que regulam temas de competência privativa dessas entidades em suas áreas de alcance. Elas resultam do esforço dos órgãos supervisores, normatizadores, disciplinadores, finalizadores e julgadores da atividade profissional médica em todo o território nacional. (CFM, 2014)

No foco das Resoluções, está o zelo pelo desempenho ético da Medicina, por adequadas condições de trabalho, pela valorização do profissional médico e pelo bom conceito da profissão e dos que a exercem legalmente e de acordo com os preceitos do Código de Ética Médica vigente. (CFM 2014)

Tais Resoluções aprovadas pela entidade e por 22 Conselhos Regionais de Medicina (CRMs) podem ser encontradas no portal do Conselho Regional de Medicina (CFM). Neste sítio são disponibilizados documentos produzidos desde 1957 até a presente data.

O usuário do Sistema proposto neste Trabalho de Conclusão de Curso utiliza as informações contidas nestas Resoluções citadas para atualizar os dados do Sistema de Fiscalização da Área da Saúde.

A missão do Conselho Federal de Medicina (CFM) envolve a promoção da ética médica e a defesa da qualidade da assistência em saúde no país. Para atingir esses objetivos a entidade tem, além de fortalecer os processos de fiscalização realizados pelos CRMs, buscado a articulação política com importantes setores da sociedade e denunciado publicamente os desmandos e abusos que comprometem a vida e o bem-estar de pacientes e profissionais. (CFM, 2015).

2.1.3 Fiscalização da Atenção Básica

A gestão da saúde pública no Brasil – em todas as esferas – tem deixado muito a desejar. As provas desse abandono, com consequências terríveis para pacientes, médicos e outros profissionais da área, estão quase diariamente no noticiário. O programa Fantástico, da Rede Globo, foi um dos últimos a colocar o tema em evidência ao apontar o grau de sucateamento das Unidades Básicas de Saúde (UBSs) e ambulatorios, revelando as condições degradantes às quais são submetidos médicos e pacientes. (TIESENHAUSEN p.2, 2015).

Segundo Tiesenhausen (p.2, 2015), “A transparência das informações da gestão é fundamental para que o cidadão possa apresentar suas cobranças a quem tem de fato responsabilidade”.

A sociedade conta, com um instrumento a mais para garantir a qualidade da assistência em saúde. Os Conselhos de Medicina possuem uma lista mínima, com a descrição de equipamentos e de infraestrutura necessários para o funcionamento de consultórios e ambulatorios médicos, como os postos de saúde (mais conhecidos como UBSs). Este *check list*, que será de conhecimento público, orienta as ações de fiscalização dos Conselhos Regionais de Medicina (CRMs) e consta de Resolução do Conselho Federal de Medicina (CFM). Existem também roteiros para hospitais, prontos socorros e outros tipos de estabelecimentos de saúde. (CFM, 2015).

O descumprimento dos itens elencados nas vistorias em consultórios e ambulatorios gerará cobrança de soluções junto aos gestores de saúde, a ser feito pelas entidades. Relatórios com as conclusões serão também encaminhadas a outras autoridades, como Tribunais de Contas, Ministério Público, Poder Legislativo, entre outros. Com isso,

espera-se estimular a tomada de decisões que leve à qualificação da assistência e a melhora das condições para o exercício da Medicina. Para conhecer o pacote mínimo que os consultórios e ambulatórios deverão cumprir, basta acessar o site do Conselho Federal de Medicina (CFM). (CFM, 2015).

Durante o ano de 2014, 952 Unidades Básicas de Saúde (UBSs) do Sistema Único de Saúde (SUS) foram vistoriadas pelo Sistema Nacional de Fiscalização do Conselho Federal de Medicina (CFM). Os números encontrados foram divulgados à imprensa, porém apontam uma situação alarmante. Dentre as unidades vistoriadas, 331 tinham mais de 50 itens em desconformidade com o estabelecido pelas normas sanitárias e 100 apresentavam mais de 80 itens fora dos padrões. A maioria delas está nas regiões Norte e Nordeste, o que comprova o abandono nas áreas mais pobres do país. (JORNAL DE MEDICINA p.6, 2015).

O CFM constatou que 15% dos consultórios não garantiam a confidencialidade das consultas, 22% das unidades não possuíam sala de espera e, no Pará, médicos atendiam debaixo de uma árvore. As fiscalizações foram realizadas pelos Conselhos Regionais de Medicina (CRMs) em ambulatórios (164), UBSs (118), centros de saúde (88) e postos do Programa de Saúde da Família. Foram avaliadas a estrutura física das unidades, os itens básicos necessários ao funcionamento de um consultório e as condições higiênicas. Em todos os aspectos, a situação encontrada é preocupante. (JORNAL DE MEDICINA p.6, 2015).

A falta de infraestrutura impede que o profissional da área da saúde forneça ao paciente um atendimento de qualidade, o que prejudica toda a população, pois o médico não consegue aplicar todo seu conhecimento a favor dos necessitados pela sua ajuda.

O trabalho fiscalização é efetuado nos serviços públicos, mas também pode ser utilizado em vistorias em unidades de atendimento vinculadas a planos de saúde ou empresas particulares. Os consultórios e ambulatórios foram divididos em três tipos. Eles vão dos que oferecem serviços mais simples, sem anestesia local e sedação, até àqueles que realizam procedimentos invasivos, com riscos de anafilaxias (reações alérgicas sistêmicas) ou paradas cardiorrespiratórias. (CFM, 2015).

2.1.4 Software de Fiscalização das Unidades de Saúde

Como parte do processo de modernização das suas atividades de fiscalização, os CRMs também contarão com instrumentos mais ágeis para fazer as visitas. A caneta e o papel foram aposentados e substituídos por *tablets*. Neles, estarão os formulários e *check lists* a serem preenchidos. Após as vistorias, os resultados serão remetidos para uma base de dados, centralizada no CFM. Desta forma, o Conselho Federal de Medicina tem acesso ao conteúdo das visitas de fiscalização de forma online e digitalizada. Essa rotina permitirá, entre outros pontos, elaboração de estudos e levantamentos sobre carências e necessidades comuns ao sistema. (CFM, 2015).

A implantação do Sistema Nacional de Fiscalização do Conselho Federal de Medicina (CFM) começou com a edição da Resolução nº 2.056, em novembro de 2013, que trouxe o Manual de Vistoria e Fiscalização da Medicina no Brasil. O documento apresenta uma lista com os itens que não podem faltar em ambulatórios, centros de saúde e consultórios médicos. Com base nessa listagem, foi desenvolvido um *software*, distribuído junto com *tablets*, máquinas fotográficas e *scanners* portáteis para todos os Conselhos Regionais de Medicina (CRMs), que receberam treinamentos. (JORNAL DE MEDICINA p.7, 2015).

Os profissionais responsáveis pela fiscalização nas Unidades de Saúde são denominados por Fiscais. Estes profissionais são treinados para vistoriar estas Entidades e, utilizando o Sistema de Fiscalização, coletar os dados referente aos questionários para posterior geração de relatório com regularidades ou irregularidades. Este relatório é gerado pelo próprio Sistema de Fiscalização e encaminhado ao Conselho Federal de Medicina para que este tome as devidas providências.

Os Conselhos anunciam ainda que intensificarão as fiscalizações nas unidades hospitalares, com divulgação dos resultados encontrados e envio de relatórios aos órgãos competentes, como Ministério Público e Tribunais de Contas. O objetivo é denunciar os problemas existentes e obrigar os gestores a apresentarem respostas urgentes à sociedade. (CFM, 2015).

Os recursos começaram a ser usados em 2011, com compra de equipamentos, organização de cursos e desenvolvimento de tecnologias. Até o momento, cerca de 100 *tablets* já foram entregues e, ao usá-los, os fiscais agregam agilidade e qualidade ao seu

trabalho nas vistorias realizadas. O presidente do CFM, Carlos Vital, reforça a importância dos CRMs na utilização dos novos instrumentos de fiscalização. “Com o que já conseguimos captar é possível comprovar a falta de estrutura da saúde no Brasil, mas temos de continuar aprimorando o sistema e a adesão de todos é fundamental”, afirma. (JORNAL DE MEDICINA p.7, 2015).

Os itens constantes no Manual de Vistoria obedecem ao que está estabelecido na RDC- 50/02, resolução da Agência Nacional de Vigilância Sanitária (Anvisa), que regulamenta os projetos físicos de estabelecimentos assistenciais. O documento também usa exigências do SomaSUS, um sistema de apoio à elaboração de projetos de investimentos em saúde, além de portarias do Ministério da Saúde e de Resoluções do CFM. (JORNAL DE MEDICINA p.7, 2015).

Com a informatização da fiscalização, o CFM conseguiu quantificar a falta de equipamentos básicos nas unidades de saúde. Também passou a ter um arquivo unificado de como está a estrutura de saúde no país. “Agora, conseguimos saber, com um clique, quais os locais onde estão faltando estetoscópios, por exemplo”, conta Emmanuel Fortes. Os números materializam o que os médicos fiscais constatarem diariamente nas vistorias realizadas pelos CRMs. (JORNAL DE MEDICINA p.7, 2015).

2.2 ESPECIFICAÇÃO DO SISTEMA

O Software de Fiscalização utilizado pelo Conselho Federal de Medicina em todo território Nacional permite uma avaliação das empresas da área da saúde, públicas ou privadas, quanto às suas faculdades em seguir as normas estabelecidas pela legislação vigente no período atual. Este sistema facilita o controle destas entidades em seguir as regras estipuladas e obrigatórias, que fornecem serviços à população, contribuindo para o melhoramento destes serviços e punindo aquelas que deixam de praticar os enunciados da Legislação.

Segundo Ian Junckes Filho (p.41, 2011), o Software de Fiscalização é um sistema que permite realizar os processos de extração, transformação e carga a partir de dados em formato XML provenientes do resultado do preenchimento de formulários para um DW (*Data*

Warehouse). Os dados contidos no DW servem como fonte de informação e conhecimento para os gestores da organização que virão a utilizá-lo, através da coleta de informações em diversas perspectivas por meio de uma ferramenta OLAP (*Online Analytical Processing*).

O sistema proposto nesta monografia permite aos usuários editar as informações contidas nos arquivos XML utilizadas pelo Sistema de Fiscalização. Estes arquivos, contêm informações relacionadas com a Legislação vigente atual, com todas as questões para avaliação das entidades da saúde. Eles formam os formulários a serem preenchidos na fiscalização e são denominados dinâmicos por estarem sempre sendo modificados de acordo com as mudanças nas Legislações.

2.2.1 Definição dos Elementos do Sistema

O trabalho de fiscalização é construído com um aplicativo e software para tornar ágeis as fiscalizações e permitir que dados estatísticos possam ser aferidos a cada ano, com o fito de garantir que os serviços médicos de qualquer natureza estejam sempre dentro de um padrão que garanta segurança na assistência ao povo Brasileiro. (CFM, 2015).

Os elementos avaliados pelo Sistema de Fiscalização seguem um padrão pré-estabelecido pelo Conselho Federal de Medicina (CFM), que está disposto nos documentos de Resolução. Segundo CFM (2014), as resoluções são atos normativos emanados dos plenários do Conselho Federal de Medicina que regulam temas de competência privativa destas entidades em suas áreas de alcance. Elas resultam do esforço dos órgãos supervisores, normatizadores, disciplinadores, fiscalizadores, julgadores da atividade profissional médica em todo o território nacional.

No foco das Resoluções, está o zelo pelo desempenho ético da Medicina, por adequadas condições de trabalho, pela valorização do profissional médico e pelo bom conceito da profissão e a dos que a exercem legalmente e de acordo com os preceitos do Código de Ética Médica vigente. (CFM, 2014).

As instituições de saúde são constantemente avaliadas pelo Conselho Federal de Medicina a fim de serem regulamentadas e punir as que estão agindo contra as normas estabelecidas pelo próprio CFM. Estas empresas podem ter vários setores diferentes, como

por exemplo, uma Clínica Médica pode ter sala de Oftalmologia, de Odontologia, de Cirurgia, entre outros. Cada departamento é avaliado de acordo com as suas necessidades individuais, como por exemplo, um ambulatório deve ter n “objetos” obrigatórios, m “objetos” que não são obrigatórios, porém auxiliares e importantes, deve ter um ambiente higienizado, profissionais qualificados e trabalhando de acordo com as leis, entre outros elementos.

O Sistema de Fiscalização é formado por múltiplos formulários contendo as avaliações a serem aplicadas nas entidades de saúde. Estes formulários, também chamados de Roteiros, são criados a partir do conteúdo das Resoluções. Cada vez que muda uma Resolução, deve-se consequentemente alterar os formulários, atividade esta que é realizada frequentemente. São arquivos em formato XML que formam a estrutura do Sistema de Fiscalização. O Apêndice B apresenta o conteúdo parcial de um arquivo XML utilizado pelo Software de Fiscalização. Cada Roteiro corresponde a um determinado tipo de consultório ou serviço. Em cada Roteiro existem vários Grupos (Grupo Um, Grupo Dois, Grupo Três, entre outros), cada um representando uma área, setor ou assunto para avaliação.

Dos consultórios e serviços do Grupo Um, em que são realizadas apenas consultas, são exigidos, por exemplo, equipamentos básicos como tensiômetro, estetoscópio, termômetro, maca, lençóis, pia, cadeiras para o médico e uma para o paciente e acompanhante entre outras exigências. “Vamos exigir além da infraestrutura física, todos os equipamentos para a propedêutica e a avaliação clínica, sem os quais o médico não vai poder atuar”, avisa Emmanuel Fortes. (CFM, 2015).

Já para os do Grupo Dois, onde se executam procedimentos sem anestesia local e sem sedação, como o consultório de um cardiologista que faz apenas eletrocardiogramas, é exigido, além do que está listado no consultório básico, os equipamentos próprios necessários para os exames específicos. (CFM, 2015).

Nos consultórios ou serviços com procedimentos invasivos ou que exponham os pacientes a risco de vida, do Grupo 3, que realize, por exemplo, teste ergométrico ou faça procedimento com anestesia local ou sedações leves, os fiscais devem averiguar se há os instrumentos que assegurem a aplicação de forma segura e, em havendo complicação tenha a sua mão equipamentos de socorro à vida. É claro que este médico precisa ser preparado para realizar os primeiros procedimentos de suporte à vida. (CFM, 2015).

Os Grupos, por sua vez, são compostos pela definição de uma lista de questões que são respondidas no momento da avaliação pelos profissionais competentes. São questões de única ou múltipla escolha contendo alguns itens como possíveis respostas. Estes itens

seguem as normas definidas pelas Legislações ao qual pertencem. No sistema proposto as referidas Legislações representam a lista de todas as Resoluções já criadas pelo Conselho Federal de Medicina. Assim sendo, os itens de uma questão podem pertencer a diferentes Legislações. Por exemplo, o item “a” da questão “1” pode pertencer à legislação “Resolução CFM nº 2056/13” e o item “b” pertencer à “Resolução CFM nº 2073/14”.

Existem três tipos de itens, quanto à sua obrigatoriedade. Tipo “obrigatório” que define o item como sendo obrigatório à Instituição possuir este e gerando uma irregularidade no sistema na sua falta. Tipo “opcional” que define o item como não obrigatório, podendo ter ou não ter na instituição, sem gerar uma irregularidade quando há falta deste. Tipo “recomendatório”, neste caso o item não é obrigatório, mas representa uma recomendação à instituição ter ele.

Quanto à sua funcionalidade no Sistema de Fiscalização o Item tem vários tipos. Como por exemplo, itens de seleção única, itens de seleção múltipla, itens dispostos em tabelas, itens com seleção múltipla limitada (quando, por exemplo, pode-se selecionar até 3 opções), entre outros.

As irregularidades são notificações geradas pelo próprio sistema de Fiscalização informando a falta de determinado item obrigatório. Neste caso a Empresa receberá a notificação para regularizar seu estabelecimento de acordo com as regras estabelecidas pelo CFM.

O software desenvolvido neste trabalho, além de possibilitar as alterações dos formulários dinâmicos utilizados no sistema de Fiscalização, terá ainda a função de poder realizar simulações de fiscalização com as Entidades de Saúde. Estas simulações são chamadas de Auto Avaliação. Um representante da Empresa de Saúde responde o formulário com as questões de avaliação, gerando ao final um relatório semelhante ao que é gerado pelo software de Fiscalização do CFM. Objetivando a empresa prevenir-se sobre as irregularidades encontradas, podendo regularizar a sua situação antes da avaliação legítima realizada pelo Conselho Federal de Medicina.

Inicialmente, o profissional que utiliza o Sistema proposto nesta monografia pertence à própria empresa que o desenvolveu, ou seja, a empresa de Tecnologia que presta serviços ao CFM continua sendo a responsável pela atualização dos formulários dinâmicos, porém de forma informatizada e não manual.

O profissional responsável pela função descrita acima não necessita de muita experiência com tecnologia, apenas precisa conhecer as funcionalidades do sistema e realizar

as alterações através deste, de acordo com a Resolução estabelecida no momento, alimentando os dados coletados da Resolução vigente no Sistema, que irá armazenar estes em um Banco de Dados Geral para posterior reutilização destes dados, e organizá-los conforme a estrutura dos formulários. Gerar os formulários ao final das alterações e enviar estes para o responsável por sua atualização no Sistema de Fiscalização (profissional este que pertence à mesma empresa de tecnologia).

Os critérios de usabilidade utilizados para desenvolvimento deste sistema estão descritos abaixo:

- **Presteza** – engloba os meios utilizados para levar o usuário a utilizar determinadas ações. Uma boa presteza guia o usuário e lhe poupa, por exemplo, o aprendizado de uma série de comandos. O sistema possui campos bem titulados, com demonstrações de como preencher em casos de datas ou documentos (RG, CPF, CNPJ).
- **Condução** – refere-se aos meios disponíveis para orientar, informar, conduzir o usuário na interação. O sistema possui mensagens indicando os próximos passos a seguir em uma determinada funcionalidade.
- **Agrupamento / Distinção de Itens** – o sistema é agrupado de acordo com os itens a que se refere. As funcionalidades relacionadas a Formulários ficam agrupados na mesma tela, as relacionadas a Grupos em outra tela, as relacionadas a Item em outra e assim por diante.
- **Feedback imediato** – diz respeito às respostas do sistema às ações do usuário. O sistema emite várias mensagens informativas de processo de suas ações, como ao finalizar um cadastro, apresenta mensagem informando que está ocorrendo uma ação, entre outros.
- **Legibilidade** – o sistema é desenvolvido com telas em cores básicas e fontes escuras para serem legíveis. Destaca fontes como títulos. Possui um bom espaçamento entre palavras e linhas.
- **Gestão de erros** – o sistema possui mecanismos que permite evitar ou diminuir a ocorrência de erros, bem como fornecer mensagens de erro legíveis e bem explicativas.

3 MÉTODO

A expressão “método” remonta à Grécia antiga; *methodos* (*methá* + *odon*), significa “o caminho para se chegar a um fim”, legando-nos o emprego que hoje fazemos, o sentido de eleger um caminho a ser percorrido para se atingir um fim. Em se tratando de pesquisa científica, são várias as possibilidades para se executar uma investigação e o fim almejado é o da comprovação ou refutação das hipóteses levantadas. (SANTOS E CANDELORO, p. 69, 2006).

Nesta etapa é abordado os métodos através dos quais é alcançado o objetivo proposto neste Trabalho de Conclusão de Curso, bem como o planejamento para realização deste. O cronograma definido para execução das atividades planejadas encontra-se no Apêndice A.

3.1 CARACTERIZAÇÃO DO TIPO DE PESQUISA

Quanto ao que se refere aos tipos de pesquisa existentes, segundo Pádua (p.37, 2004), muitos autores tem buscado organizar uma tipologia para as atividades de pesquisa existentes, a partir de diferentes critérios e enfoques, tais como: os campos da atividade humana (multidisciplinares/interdisciplinares), a utilização dos resultados (pura/aplicada), segundo as técnicas e instrumentos de observação (direta-participante/não participante ou indireta-questionários/entrevistas) ou ainda quanto ao “material” utilizado em sua elaboração (bibliográfica/documental).

Estas tipologias surgiram para auxiliar o desenvolvimento das atividades de pesquisa, entretanto para além do que uma tipologia de pesquisa requer, devemos reconhecer que o fundamental é compreender a realidade em seus múltiplos aspectos, e para tanto, esta compreensão vai requerer e talvez admitir, diferentes enfoques, diferentes níveis de aprofundamento, diferentes recursos, dependendo dos objetivos a serem alcançados e as possibilidades do próprio pesquisador para desenvolvê-los. (PÁDUA p.37, 2004).

Este Trabalho de Conclusão de curso é descrito como sendo uma pesquisa aplicada, qualitativa, bibliográfica e exploratória.

3.2 ESTRUTURA DA MONOGRAFIA

Este trabalho de conclusão de curso está estruturado em seis capítulos.

O primeiro capítulo é dividido em apresentação do tema, a problemática, objetivos (gerais e específicos), justificativa e estrutura da monografia.

O segundo capítulo apresenta a revisão da literatura contendo toda a teoria sobre o sistema de Conselhos de Medicina e sobre desenvolvimento de software para web.

No terceiro capítulo define-se o método utilizado para chegar ao resultado do objetivo especificado no início do trabalho e as etapas necessárias para isto.

No quarto capítulo é abordada a modelagem completa do sistema desenvolvido e no quinto capítulo a especificação do desenvolvimento e o seu resultado, ou seja, o sistema implementado.

O sexto e último capítulo contém as conclusões sobre esta monografia e relatório com ideias para trabalhos futuros.

3.2.1 Pesquisa Bibliográfica

De acordo com Rampazzo (p. 53, 2002), a pesquisa bibliográfica procura explicar um problema a partir de referências teóricas publicadas (em livros, revistas, etc). Pode ser realizada independentemente, ou como parte de outros tipos de pesquisa.

Qualquer espécie de pesquisa, em qualquer área, supõe e exige uma pesquisa bibliográfica prévia, quer para o levantamento da situação da questão, quer para fundamentação teórica, ou ainda para justificar os limites e contribuições da própria pesquisa. (RAMPAZZO p. 53, 2002).

Koche (1997, p. 122 apud LEONEL e HEERDT 2007, p.67) afirma que a pesquisa bibliográfica pode ser realizada com diferentes fins:

- Para ampliar o grau de conhecimento em uma determinada área, capacitando o investigador a compreender ou delimitar melhor um problema de pesquisa;
- Para dominar o conhecimento disponível e utiliza-lo como base ou fundamentação na construção de um modelo teórico explicativo de um problema, isto é, como instrumento auxiliar para a construção e fundamentação de hipóteses;
- Para descrever ou sistematizar o estado da arte, daquele momento, pertinente a um determinado tema ou problema.

3.2.2 Pesquisa exploratória

Segundo Prodanov e Freitas (p.51, 2013), a Pesquisa Exploratória ocorre quando a pesquisa se encontra na fase preliminar. Ela tem como finalidade proporcionar mais informações sobre o assunto que será investigado, possibilitando sua definição e seu delineamento, isto é, facilitar a delimitação do tema da pesquisa, orientar a fixação dos objetivos e a formulação das hipóteses ou descobrir um novo tipo de enfoque para o assunto. Assume em geral as formas de pesquisa bibliográficas e estudos de caso.

O autor citado acima ainda descreve que a Pesquisa Exploratória possui planejamento flexível, o que permite o estudo do tema sob diversos ângulos e aspectos. Em geral, esta pesquisa envolve:

- Levantamento bibliográfico.
- Entrevistas com pessoas que tiveram experiências práticas com o problema pesquisado.
- Análise de exemplos que estimulem a compreensão.

3.2.3 Pesquisa Qualitativa

A pesquisa de origem qualitativa é aquela que permite que o acadêmico levante dados subjetivos, bem como outros níveis de consciência da população estudada, a partir de depoimentos dos entrevistados, ou seja, informações pertinentes ao universo a ser investigado, que leve em conta a ideia de processo, de visão sistêmica, de significações e de contexto cultural. A pesquisa qualitativa não tem a pretensão de mensurar variáveis, mas de analisar, qualitativamente, de modo indutivo, todas as informações levantadas pelo acadêmico através da aplicação de um instrumento de coleta de dados adequado. Os mais usuais no plano qualitativo, são a entrevista semi-estruturada, o estudo de caso e os grupos focais. (SANTOS E CANDELORO, p. 71, 2006).

3.2.4 Pesquisa Aplicada

De acordo com Barros e Lehfeld (2000, p. 78 apud VILAÇA 2010, p.64), a pesquisa aplicada tem como motivação a necessidade de produzir conhecimento para aplicação de seus resultados, com o objetivo de “contribuir para fins práticos, visando à solução mais ou menos imediata do problema encontrado na realidade”. Appolinário (2004, p.

152, apud VILAÇA 2010, p.65) salienta que pesquisas aplicadas têm o objetivo de “resolver problemas ou necessidades concretas e imediatas”.

As pesquisas aplicadas dependem de dados que podem ser coletados de formas diferenciadas, tais como pesquisas em laboratórios, pesquisa de campo, entrevistas, gravação em áudio e / ou vídeo, diários, questionários, formulários, análises de documentos, etc (NUNAN, 1997; MICHEL, 2005; OLIVEIRA, 2007 apud VILAÇA 2010, p.65). Ao contrário da pesquisa teórica, investigações de natureza aplicada apresentam complexidades metodológicas e éticas muito mais complexas. Devido a estas questões, as práticas de pesquisas aplicadas estão mais frequentemente associadas ao ensino superior e à pós-graduação. (VILAÇA 2010, p.65).

3.3 ETAPAS METODOLÓGICAS

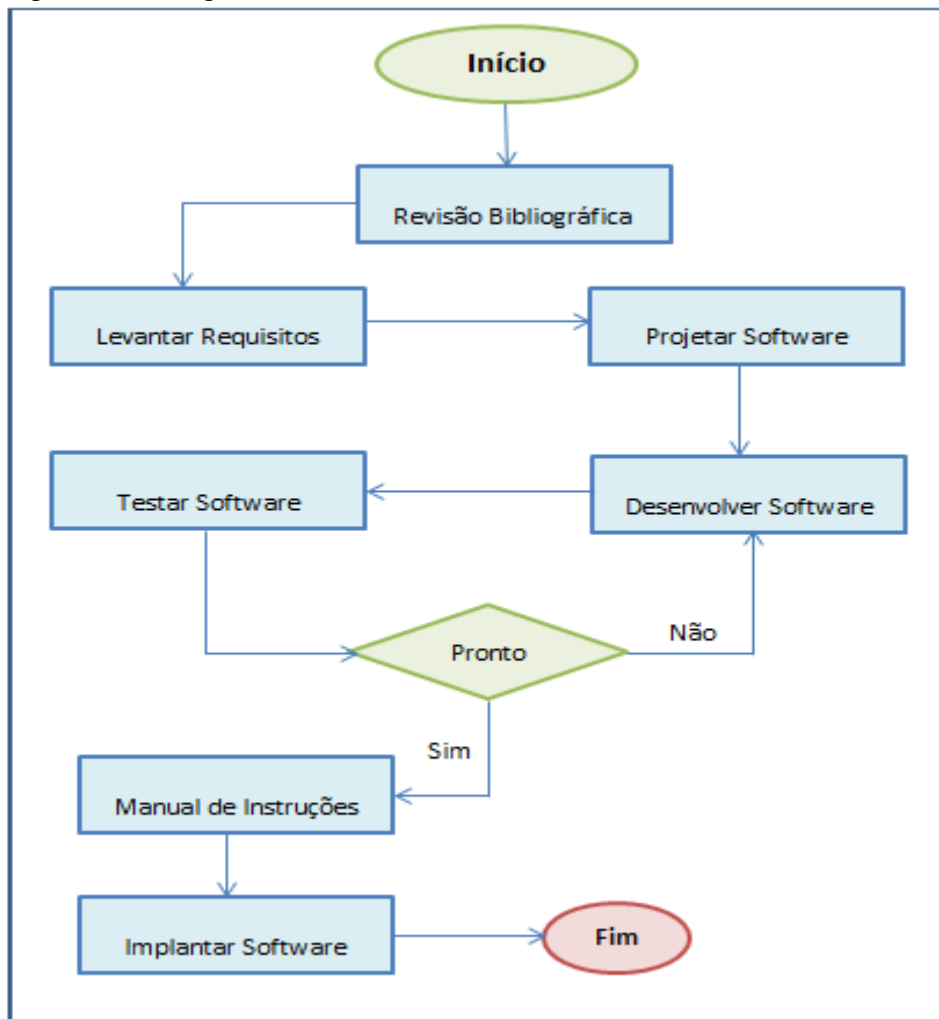
Para a realização deste Trabalho de Conclusão de Curso as seguintes etapas metodológicas estão definidas:

- Revisão bibliográfica de Programação Web e Sistemas de Informação para área da Saúde.
- Estudar sobre padrões de projeto Web.
- Levantamento de requisitos e necessidades do projeto em questão.
- Criação de um layout para a ferramenta apresentada.
- Escolha da plataforma de desenvolvimento e demais ferramentas necessárias para sua implementação.
- Modelagem do banco de dados relacional.
- Implementação do sistema Web proposto.
- Realização de testes sobre o sistema implementado.
- Criação do documento de Manual de Instrução do Sistema implementado e testado.
- Implantação da ferramenta para uso.

O presente Trabalho de Conclusão de Curso teve o intuito de seguir o fluxograma exposto na figura a seguir, buscando atingir os objetivos pretendidos para que se possa obter

ao final uma conclusão satisfatória. Se acaso o resultado não for satisfatório, deve-se voltar ao início do fluxograma e repassar as etapas definidas.

Figura 2: Fluxograma.



Fonte: Autora deste trabalho, 2015.

3.4 PROPOSTA DA SOLUÇÃO

A figura 03 mostra a arquitetura básica referente ao sistema proposto neste Trabalho de Conclusão de Curso. Constitui-se de uma arquitetura simples, onde no lado cliente se encontram as máquinas onde está implantado o sistema, podendo ser qualquer equipamento que contenha sistema operacional derivados de Windows.

A internet utilizada fica à escolha do cliente, podendo interferir apenas na velocidade de acesso às funcionalidades do sistema. O servidor ficará em outro local com segurança e *backup* de dados constante. O banco de dados utilizado é o Oracle 11G e o servidor o JBoss 7.1, definidos conforme necessidade e já adaptação da utilização por parte do cliente que utiliza este Banco para armazenar as informações referentes ao sistema de Fiscalização.

Nas Estações de trabalho do lado cliente, o usuário do sistema utilizando um *Browser* (como por exemplo, *Google Chrome*, *Mozilla Firefox*, *Internet Explorer*) acessa o Sistema, realiza os procedimentos necessários (cadastro, alteração, entre outros) até gerar os arquivos XML e exporta estes para a máquina local ao qual está trabalhando. Estes arquivos são substituídos no Sistema de Fiscalização pelos arquivos anteriores correspondentes. Com esta atualização o Fiscal responsável pelas vistorias nas Unidades de Saúde tem acesso às novas orientações dispostas na nova Resolução no seu *tablet* por onde faz as avaliações e armazena as respostas coletadas, permitindo uma atualização mais ágil de seu Sistema de Fiscalização conforme novas normas estipuladas pelo Conselho Federal de Medicina.

Figura 3: Arquitetura do Sistema.



Fonte: Autora deste trabalho, 2014.

3.5 DELIMITAÇÕES

O trabalho tem como foco principal a construção de uma ferramenta Web para manipular (cadastrar, editar, remover) formulários dinâmicos que são utilizados como base para outro sistema (fiscalização na área da saúde).

Não será implementado neste Trabalho de Conclusão de Curso este sistema em plataforma Desktop, apenas Web, dependendo então de um acesso à internet para utilização do mesmo.

A integração entre este Sistema e o Sistema de fiscalização ainda não foi desenvolvida, sendo que desta forma os formulários dinâmicos gerados pelo Sistema desta monografia necessitam ser substituídos manualmente no Sistema de Fiscalização.

4 MODELAGEM

Na criação de produtos são utilizados modelos, normalmente com o uso de símbolos e desenhos técnicos, que tem o objetivo de representar todas as suas características relevantes. Em cada engenharia existe uma forma técnica de utilizar símbolos e diagramas para criar modelos da peça, do produto ou do sistema a ser construído. Na modelagem de software não seria diferente. Através de diagramas são construídos modelos que tem a função de representar o software de forma objetiva. (LOBO p. 55, 2008).

A metodologia utilizada nesta monografia é ICONIX, por melhor se adaptar a este tipo de software. De acordo com George Silva et. all. (2007), o ICONIX é um processo de desenvolvimento de software desenvolvido pela ICONIX *Software Engineering*. Trata-se de uma metodologia prática e simples, mas também poderosa e com um componente de análise e representação de problemas sólido e eficaz.

É um processo não tão burocrático quanto o RUP (*Rational Unified Process*), ou seja, não gera tanta documentação, nem tão simples como o XP (*eXtreme Programing*), mas não deixa a desejar na análise e se destaca como um eficiente processo de análise de software. O ICONIX é um processo que está adaptado ao padrão da UML, possuindo uma característica exclusiva chamada *Traceability of Requiriments* (Rastreabilidade dos Requisitos), que através de seus mecanismos, permite checar em todas as fases se os requisitos estão sendo atendidos. (MAIA, 2005 apud SILVA; SILVA; GUIMARÃES; MEDEIROS; ROSSINI, 2007).

Segundo George Silva et. all. (2007), os principais diagramas do processo ICONIX são os seguintes:

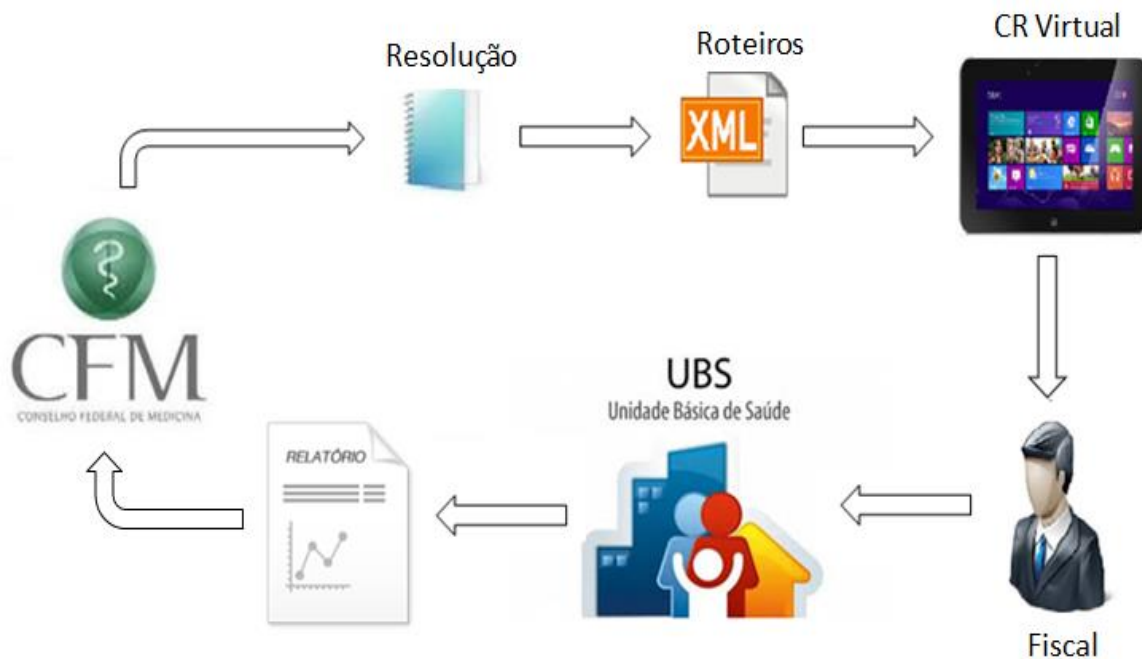
- Modelo de Domínio.
- Modelo de Caso de Uso.
- Diagrama de Robustez.
- Diagrama de Sequência.
- Diagrama de Classe.

4.1 MODELAGEM DE PROCESSO DE NEGÓCIOS

Modelos são representações que visam a descrever uma realidade ou uma concepção. A sua aplicação é variada, mas genericamente, modelos são construídos para servir a um propósito ancestral do ser humano inteligente: explicar e moldar o mundo à sua volta pela imposição de formas, relações, leis e limites concebidos ou interpretados segundo a vontade e entendimento do projetista e, não raro, de seu patrocinador. (FERREIRA, 2010).

O processo de fiscalização na área da saúde tem início pelo Conselho Federal de Medicina, que cria as Resoluções contendo as normas e demais especificações necessárias para que uma entidade possa fornecer um serviço de qualidade e eficiência à população, além de oferecer uma estrutura adequada para que o profissional da área da saúde possa exercer sua profissão de forma segura e ágil. As Resoluções são transformadas em arquivos XML contendo tais normativas. Estes arquivos formam a base estrutural do software de Fiscalização (chamado CR Virtual), sendo transformados em telas, permitindo ao Fiscal (responsável pelas vistorias) obter as informações atualizadas para desempenhar seu trabalho. Esse profissional através de um *tablet* contendo o software de fiscalização, realiza as vistorias nas unidades de saúde, gera os relatórios finais contendo o resultado da fiscalização e envia este relatório ao Conselho Federal de Medicina, para que possam ser tomadas as devidas providências em casos de irregularidade. Uma representação do processo de fiscalização nas entidades de saúde é apresentada através da figura abaixo.

Figura 4: Processo de Fiscalização

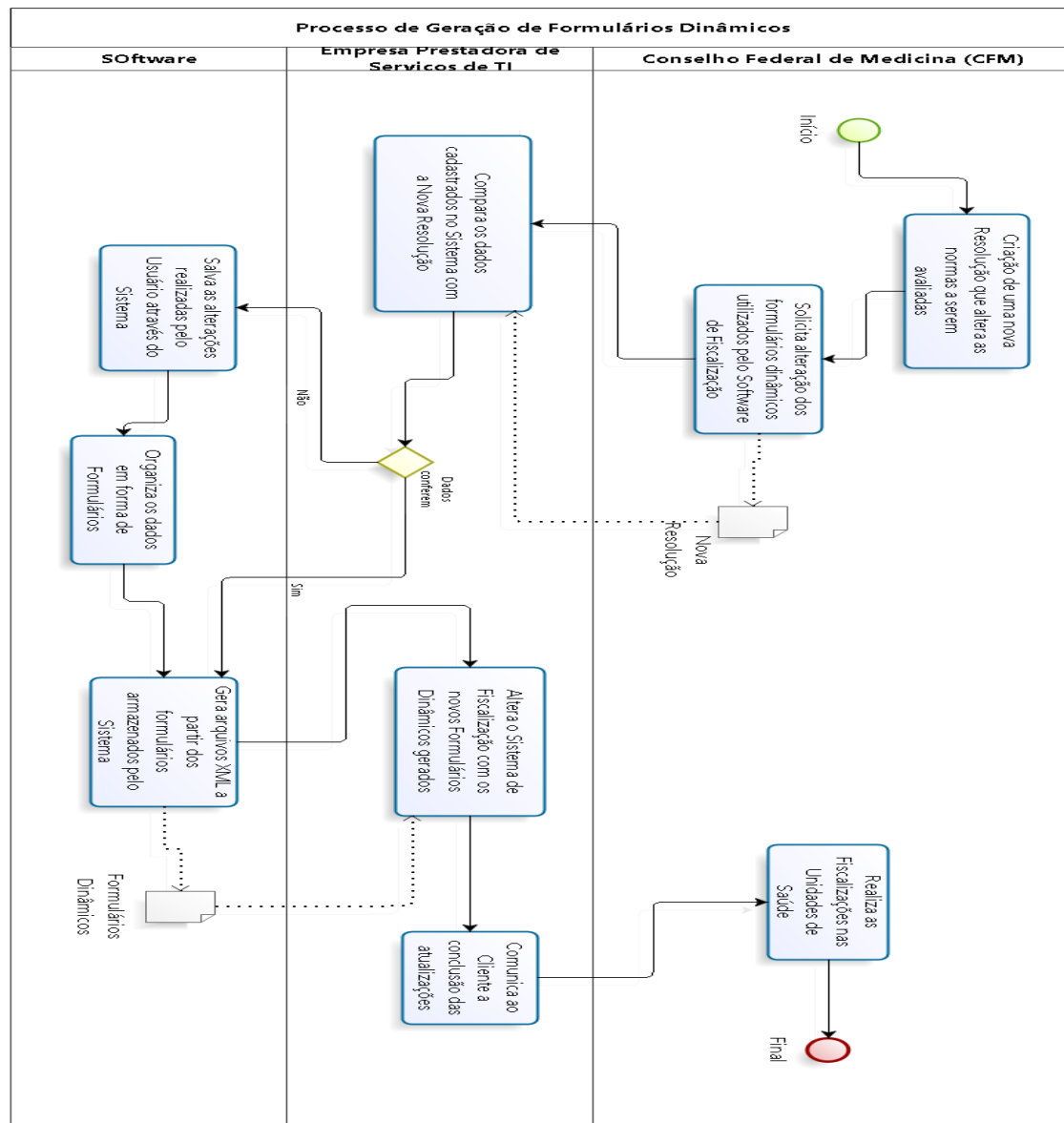


Fonte: Autora desta monografia, 2015.

Segundo Ferreira (2010), a Modelagem de Processo de Negócios é um modelo que descreve a forma pela qual ela realiza o seu negócio. No entanto não existem modelos absolutos, capazes de incorporar à representação de uma dada realidade de maneira definitiva e absoluta.

A figura a seguir representa de forma resumida o modelo de Processo de Negócios para Geração de Formulários Dinâmicos, conforme abordado na Especificação do Sistema, no Capítulo Dois desta Monografia.

Figura 5: Modelagem de Processo.



Fonte: Autora desta Monografia.

O processo de Geração de Formulários dinâmicos tem início no momento que o Conselho Federal de Medicina libera uma nova Resolução para normatizar Unidades de Saúde e Profissionais desta Área. Surge então a necessidade de alteração dos formulários dinâmicos que compõem a base estrutural do Sistema de Fiscalização, demanda esta que é enviada à Empresa de Tecnologia da Informação que presta Serviços ao CFM, a qual desenvolveu tanto o Sistema proposto nesta monografia, como o próprio Sistema de Fiscalização.

Um profissional da Empresa de TI mencionada acima, responsável pelas atualizações frequentes destes formulários, uma vez de posse da nova Resolução utiliza o

Sistema para realizar as devidas correções nos formulários, funcionalidade esta que anteriormente a este Sistema era realizada manualmente, formulário por formulário, utilizando um editor de textos qualquer.

O profissional acima descrito interage com o Sistema, cadastrando ou alterando os dados necessários, organizando os Itens nas respectivas Questões e estas em seus respectivos Grupos de Formulários. Estas alterações são armazenadas em um Banco de Dados Relacional para posterior reutilização e conferência.

Estes formulários podem ser visualizados no Software, da mesma forma que será apresentado no Sistema de Fiscalização, a fim de que sejam validadas suas informações e formatações anteriormente à publicação destes, evitando desta forma retrabalho e desperdício de tempo.

Depois de completada as alterações, o Usuário deste Sistema pode gerar os formulários dinâmicos criados ou modificados, sendo estes exportados em arquivo com formato XML, que é o tipo necessário para utilização no Sistema de Fiscalização. Estes arquivos são enviados para o responsável pela sua substituição destes no Software de Fiscalização (responsável este da mesma empresa de TI) e por fim notificado ao Conselho federal de Medicina o término das alterações.

Concluídas estas etapas o Software de Fiscalização estará atualizado com as orientações da nova Resolução. Um trabalho realizado lentamente e com pouca segurança passa a ser, através do Sistema apresentado nesta Monografia, uma tarefa fácil, ágil e segura.

4.2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

Na análise de requisitos, o analista vai procurar caracterizar certas propriedades dos requisitos já levantados. (WAZLAWICK, p. 29, 2011).

4.2.1 Requisitos Funcionais

Abaixo são identificados os requisitos funcionais do sistema, separados por módulos de acordo com suas funcionalidades.

Quadro 1: Requisitos Funcionais Módulo Acessar.

RF001 – Permitir acesso autenticado	O sistema deve permitir a autenticação dos usuários, com uma tela de login e senha.
--	---

Fonte: Autora do Trabalho, 2015.

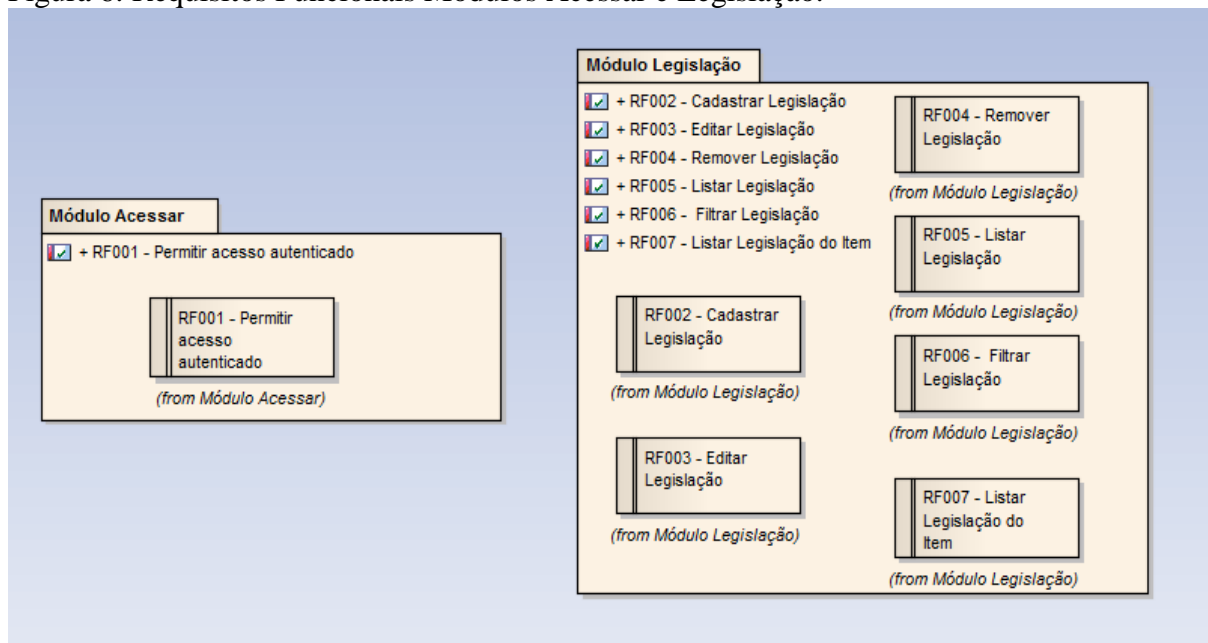
Quadro 2: Requisitos Funcionais Módulo Legislação.

RF002 – Cadastrar Legislação	O sistema deve permitir o cadastro de uma nova legislação.
RF003 – Editar Legislação	O sistema deve permitir a edição de uma legislação existente no banco.
RF004 – Remover Legislação	O sistema deve permitir a remoção de uma legislação existente no banco.
RF005 – Listar Legislação	O sistema deve permitir visualizar uma lista com todas as Legislações cadastradas no banco de dados.
RF006 – Filtrar Legislação	O sistema deve permitir que o usuário pesquise uma Legislação entre as existentes no banco de dados através de um texto digitado pelo usuário ou por seu tipo ou ainda por um Item vinculado.
RF007 – Listar Legislação do Item	O sistema deve permitir a visualização da Legislação pertencente ao Item selecionado.

Fonte: Autora do Trabalho, 2015.

A figura abaixo representa os requisitos funcionais para os módulos Acessar e Legislação.

Figura 6: Requisitos Funcionais Módulos Acessar e Legislação.



Fonte: Autor do Trabalho, 2015.

Quadro 3: Requisitos Funcionais Módulo Item.

RF008 – Cadastrar Item	O sistema deve permitir o cadastro de novos itens.
RF009 – Editar Item	O sistema deve permitir a edição de um Item existente no banco.
RF010 – Remover Item	O sistema deve permitir a remoção de um Item existente no banco.
RF011 – Listar Item	O sistema deve permitir visualizar uma lista com todos os Itens cadastrados no banco de dados.
RF012 – Filtrar Item	O sistema deve permitir que o usuário pesquise um Item entre os existentes no banco de dados a partir de um texto digitado pelo usuário ou por grupo ou categoria ou tipo ou ainda por legislação vinculada.
RF013 – Mover Item	O sistema deve permitir a alteração da posição de um Item no grupo a que ele pertence.
RF014 – Listar Item por Grupo	O sistema deve permitir a listagem dos Itens pertencentes a um Grupo escolhido pelo usuário.

Fonte: Autora do Trabalho, 2015.

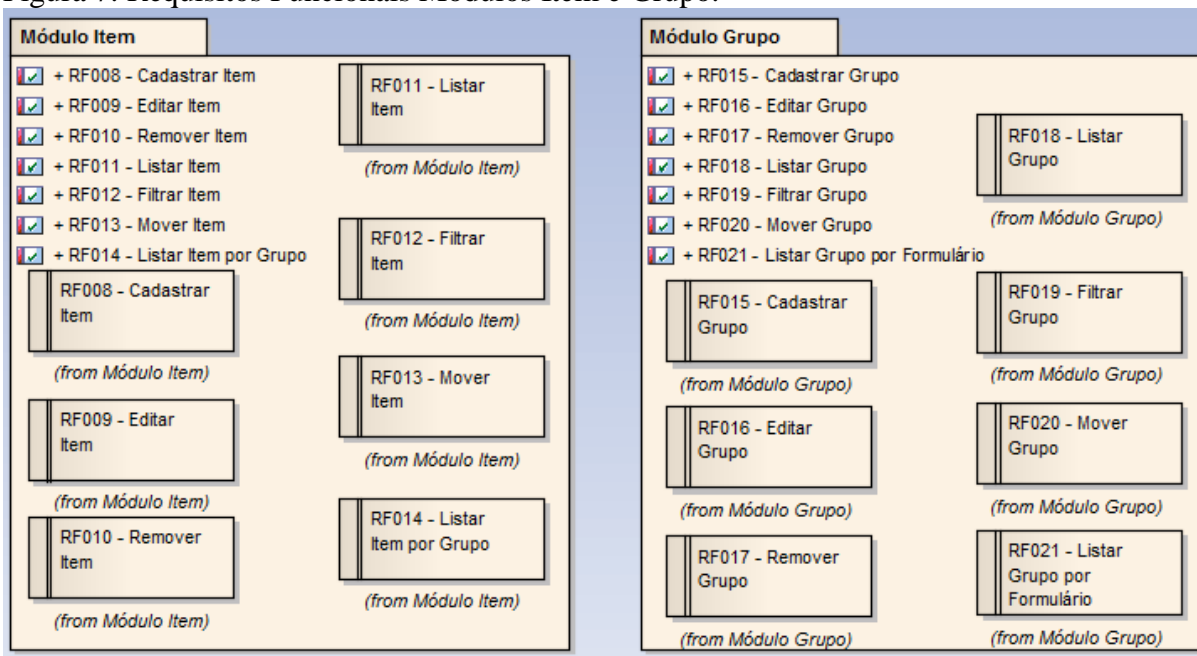
Quadro 4: Requisitos Funcionais Módulo Grupo.

RF015 – Cadastrar Grupo	O sistema deve permitir o cadastro de novo grupo.
RF016 – Editar Grupo	O sistema deve permitir a edição de um Grupo existente no banco.
RF017 – Remover Grupo	O sistema deve permitir a remoção de um Grupo existente no banco.
RF018 – Listar Grupo	O sistema deve permitir visualizar uma lista com todos os Grupos cadastrados no banco de dados.
RF019 – Filtrar Grupo	O sistema deve permitir que o usuário pesquise um Grupo entre os existentes no banco de dados a partir de texto digitado pelo usuário e também por roteiro/formulário.
RF020 – Mover Grupo	O sistema deve permitir a alteração de posição de um grupo dentro do formulário a que ele pertence.
RF021 – Listar Grupo por Formulário	O sistema deve permitir a listagem dos Grupos pertencentes a um Formulário escolhido pelo usuário.

Fonte: Autora do Trabalho, 2015.

A figura abaixo representa os requisitos funcionais para os módulos Item e Grupo.

Figura 7: Requisitos Funcionais Módulos Item e Grupo.



Fonte: Autor do Trabalho, 2015.

Quadro 5: Requisitos Funcionais Módulo Formulário.

RF022 – Cadastrar Formulário	O sistema deve permitir o cadastro de novo formulário.
RF023 – Editar Formulário	O sistema deve permitir a edição de um formulário existente no banco.
RF024 – Remover Formulário	O sistema deve permitir a remoção de um formulário existente no banco.
RF025 – Listar Formulário	O sistema deve permitir visualizar uma lista com todos os Formulários cadastrados no banco de dados.
RF026 – Listar dados de Formulário	O sistema deve permitir a listagem de todos os dados pertencentes aos formulários escolhidos pelo usuário.

Fonte: Autora do Trabalho, 2015.

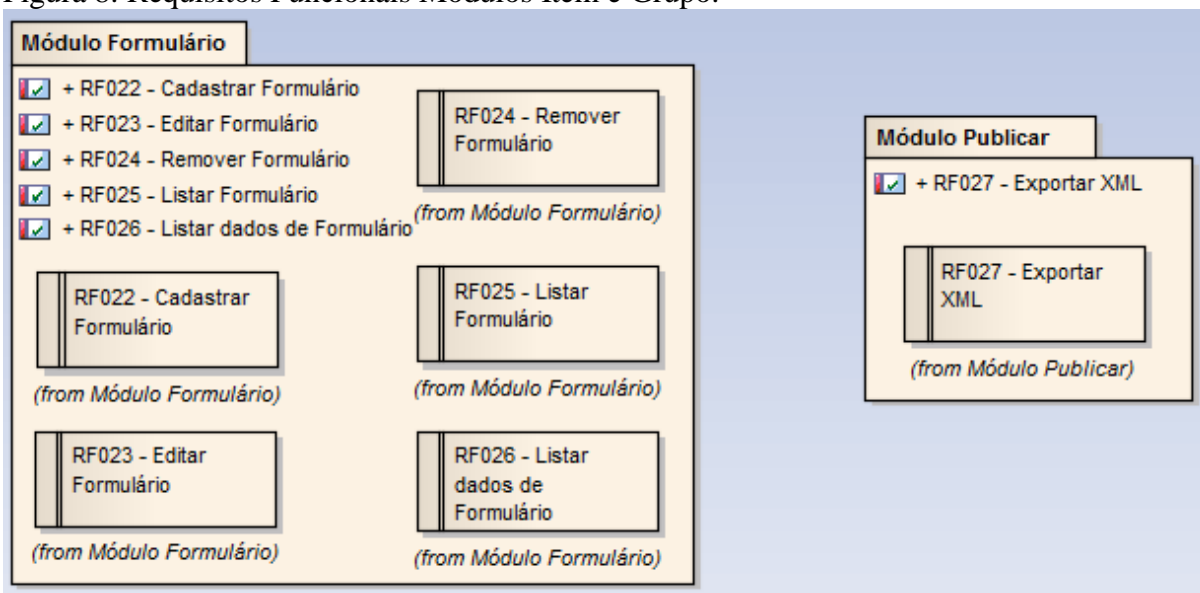
Quadro 6: Requisitos Funcionais Módulo Publicar.

RF027 – Exportar XML	O sistema deve permitir a exportação de um formulário escolhido pelo usuário.
-----------------------------	---

Fonte: Autora do Trabalho, 2015.

A figura abaixo representa os requisitos funcionais dos Módulos Formulário e Publicar.

Figura 8: Requisitos Funcionais Módulos Item e Grupo.



Fonte: Autora do Trabalho, 2015.

Quadro 7: Requisitos Funcionais Módulo Irregularidade.

RF028 – Listar Irregularidades	O sistema deve permitir a visualização das irregularidades geradas pela avaliação.
RF029 – Gerar relatório de irregularidade	O Sistema deve permitir que seja gerado um arquivo com todas as irregularidades encontradas pela auto avaliação.
RF030 – Enviar e-mail com irregularidade	O sistema deve permitir o envio do arquivo de irregularidades por e-mail.

Fonte: Autora do Trabalho, 2015.

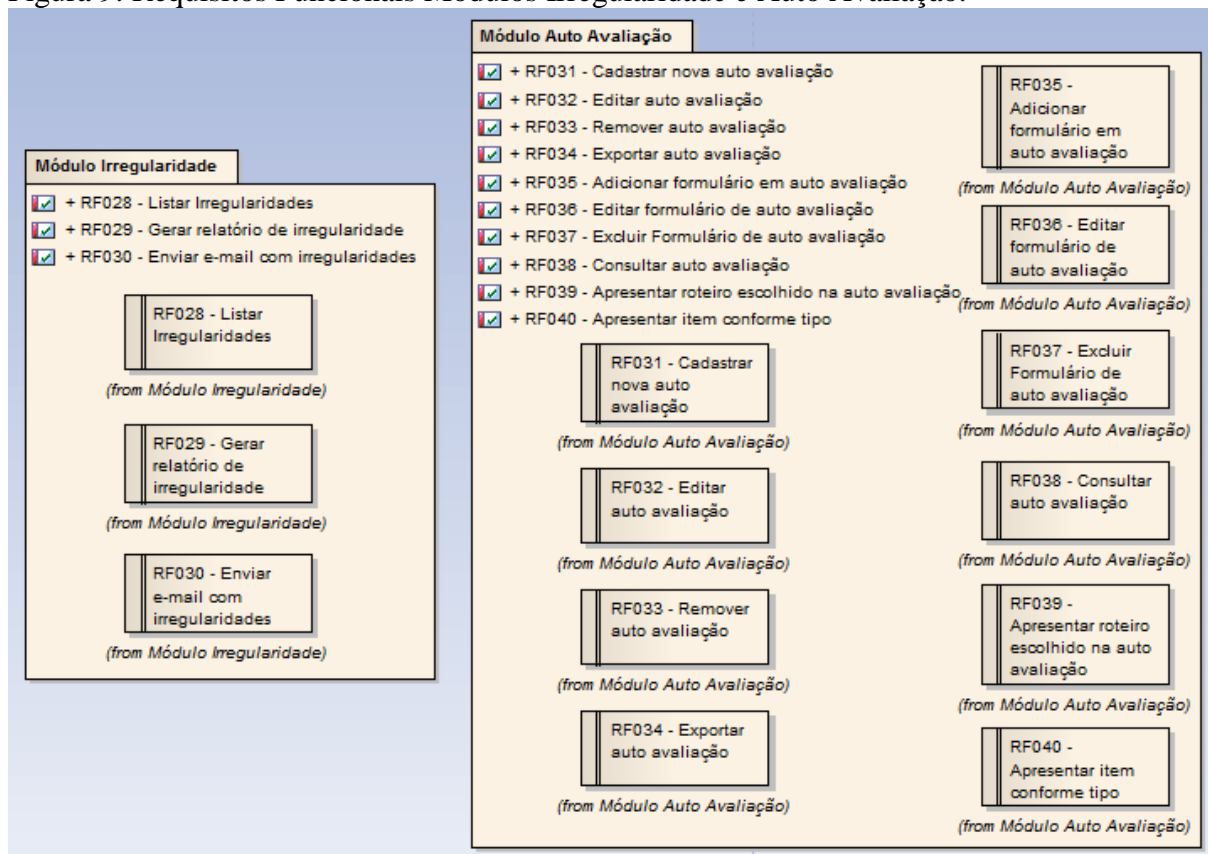
Quadro 8: Requisitos Funcionais Módulo Auto Avaliação.

RF031 – Cadastrar nova auto avaliação	O sistema deve permitir que uma PF ou PJ inclua uma auto avaliação.
RF032 – Editar auto avaliação	O sistema deve permitir que uma Pessoa Física (PF) ou Pessoa Jurídica (PJ) altere uma auto avaliação existente.
RF033 – Remover auto avaliação	O sistema deve permitir a exclusão de uma auto avaliação.
RF034 – Exportar auto avaliação	O sistema deve permitir a exportação de uma auto avaliação realizada em formato pdf.
RF035 – Adicionar formulário em auto avaliação	O sistema deve permitir adicionar quantos formulários quiser na auto avaliação.
RF036 – Editar formulário de auto avaliação	O sistema deve permitir editar os formulários de uma auto avaliação, contendo em cada item a possibilidade de limpar, colocar observação e consultar legislação.
RF037 – Excluir formulário de auto avaliação	O sistema deve permitir exclusão de formulários de uma auto avaliação.
RF038 – Consultar auto avaliação	O sistema deve permitir que uma Pessoa Física (PF) ou Pessoa Jurídica (PJ) consulte uma auto avaliação existente.
RF039 – Apresentar roteiro escolhido na auto avaliação	O sistema deve permitir que se apresente na auto avaliação, o roteiro separado por grupos e itens.
RF040 – Apresentar Item conforme tipo	O sistema deve permitir que em uma auto avaliação os itens estejam de acordo com seu tipo cadastrado e com lógica de persistência própria.

Fonte: Autora do Trabalho, 2015.

A figura abaixo representa os requisitos funcionais dos Módulos Irregularidade e Auto Avaliação.

Figura 9: Requisitos Funcionais Módulos Irregularidade e Auto Avaliação.



Fonte: Autora do Trabalho, 2014.

4.2.2 Requisitos Não Funcionais

Requisitos não funcionais podem ser considerados permanentes ou transitórios de acordo com uma decisão tomada pelo analista em conjunto com o cliente. (WAZLAWICK, p. 29, 2011).

Segundo Wazlawick (p.25, 2011), os requisitos não funcionais aparecem sempre ligados a requisitos funcionais e podem ser basicamente de dois tipos: lógicos ou tecnológicos.

As *restrições lógicas* são as regras de negócio relacionadas à função em questão. Por exemplo, no registro de uma venda, uma série de restrições lógicas poderia ser considerada, como por exemplo: não efetuar a venda caso a operadora de cartão não autorize

o pagamento ou não efetuar a venda caso a venda anterior tenha sido cancelada devido a um endereço inválido que ainda não foi corrigido. (WAZLAWICK, p. 25, 2011).

As *restrições tecnológicas*, conforme afirma Wazlawick (p.25, 2011), dizem respeito à tecnologia para realização da função, como por exemplo, a interface (Web, por exemplo), o tipo de protocolo de comunicação, restrições de segurança ou tolerância a falhas, entre outros.

Figura 10: Requisitos Não Funcionais.



Fonte: Autora do Trabalho, 2014.

Quadro 9: Requisitos Não Funcionais.

RNF001 – Utilizar Banco de dados Oracle.	O sistema deve persistir seus dados em banco de dados oracle.
RNF002 – Utilizar Ajax.	Será utilizado para implementação do sistema além das tecnologias já apresentadas, a tecnologia Ajax que permite uma atualização mais dinâmica e eficiente.
RNF003 – Utilizar linguagem de Programação Java.	A linguagem de programação escolhida para implementação do sistema proposto foi Java, por motivos que serão explicados mais adiante.
RNF004 – Utilizar XML.	O sistema manipula arquivos XML pelo fato do sistema de Fiscalização estar estruturado com este tipo de arquivo.
RNF005 – Exibir mensagens de resposta.	A cada nova interação com o sistema, ao utilizar suas funções para algum propósito será apresentado mensagens informativas de acordo com o que está sendo realizado. Mensagens de erro, de ações realizadas com sucesso, entre outras.

Fonte: Autora do Trabalho, 2015.

4.3 REGRAS DE NEGÓCIOS

Segundo Peixoto (2015), Regras de Negócios são padrões que definem ou restringem algum aspecto do negócio. As regras de negócio podem começar como políticas comerciais, metas, estratégias e diretrizes, que são expressas como afirmações declarativas, às restrições ou ações que se baseiam. As regras de negócio são utilizadas para criar banco de dados, aplicação, ou a lógica de um processo.

O quadro abaixo apresenta as regras de negócio das funcionalidades mais significantes do sistema proposto nesta monografia.

Quadro 10: Regras de Negócios.

RN001 – Login.	O formulário de login deve conter os campos: nome de usuário, senha e botão Acessar.
RN002 – Tela Inicial.	O formulário da tela inicial do sistema deve conter os seguintes botões: Gerenciar Roteiros, Publicar Roteiros e Avaliar Roteiros.
RN003 – Legislação.	Os formulários de cadastro e edição de Legislação devem conter os seguintes campos: descrição, instruções, número, ano, capítulo, artigo, alínea, tipo, órgão, link, data inicial da vigência, data final da vigência e botão Salvar.
RN004 – Item.	Os formulários de cadastro e edição de Item devem conter os seguintes campos: título, descrição, descrição para relatório, tipo, categoria, lista com legislações, autodescrição, requerido, requerido fiscal e botão salvar.
RN005 – Grupo.	Os formulários de cadastro e edição de Grupo devem conter os seguintes campos: título, status, imagem, lista de Itens, requerido fiscal e botão salvar.
RN006 – Formulário	Os formulários de cadastro e edição de formulário devem conter os seguintes campos: tipo, área, data inicial da vigência, data final da vigência, dinâmico, lista de Grupos, lista de Estados (UF) e botão salvar.
RN007 – Publicar Roteiro	O formulário de publicar roteiro deve conter os seguintes campos: lista de formulários, nome do arquivo e botão publicar.

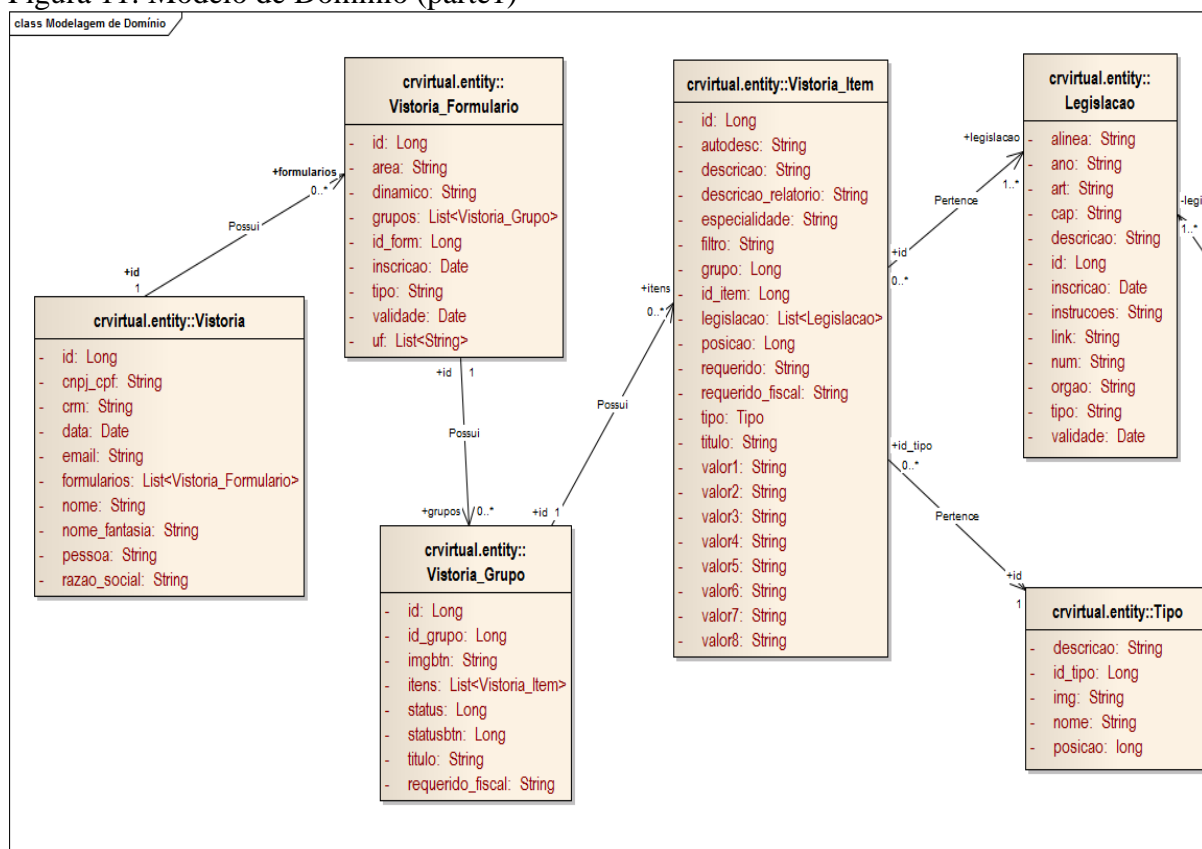
Fonte: Autora do Trabalho, 2015.

4.4 MODELAGEM DE DOMÍNIO

De acordo com Larman (p.160, 2005), um modelo de domínio é uma representação visual de classes conceituais, ou objetos do mundo real, em um domínio. Modelos de domínios também tem sido chamados de modelos conceituais, modelos de objetos de domínio e modelos de objeto da análise.

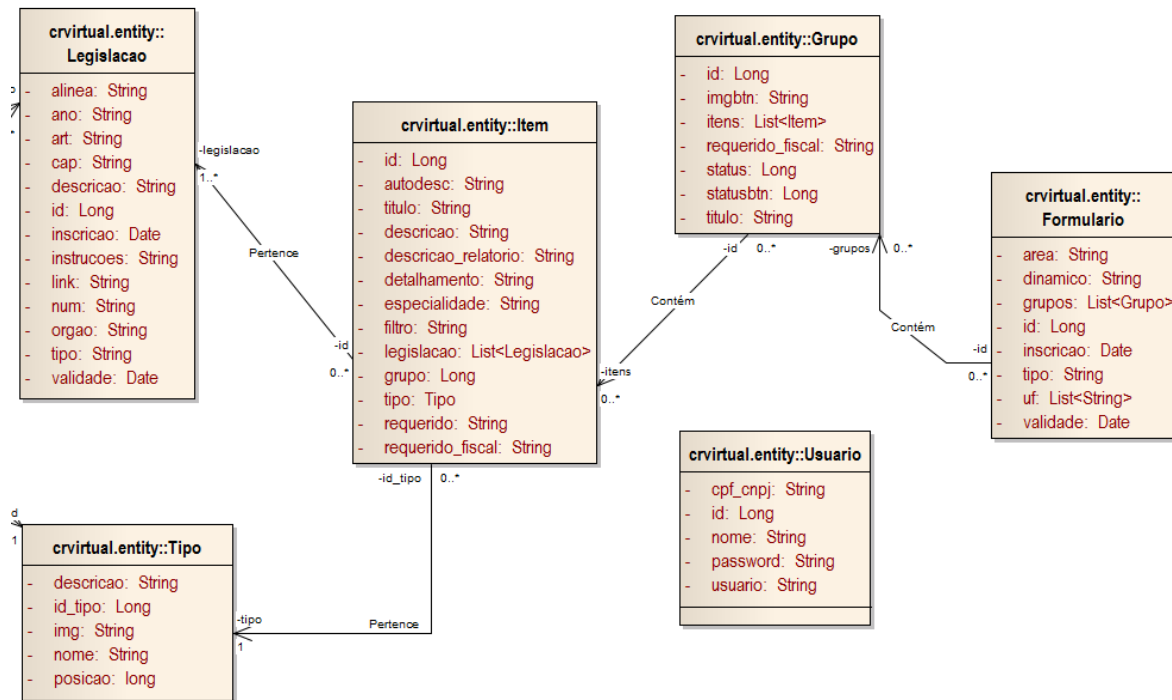
Aplicando a notação UML, um modelo de domínio é ilustrado com um conjunto de diagramas de classes em que nenhuma operação é definida. Ela fornece uma perspectiva conceitual. As figuras abaixo apresentam o modelo de domínio do sistema proposto nesta monografia.

Figura 11: Modelo de Domínio (parte1)



Fonte: Autora desta Monografia, 2015.

Figura 12: Modelo de Domínio (parte 2)



Fonte: Autora desta Monografia, 2015.

4.5 MODELAGEM DE CLASSES

O diagrama de classes é um dos principais diagramas da UML. Sem este diagrama não será possível definir o sistema, pois ele é essencial e serve de base para vários outros diagramas. As classes do sistema representam um modelo lógico das estruturas abstratas do software, ou seja, de tudo que será realizado no sistema. (LOBO p. 57, 2008)

As classes de entidade do sistema são:

(Para funções de manipulação dos arquivos XML)

- *Usuario* – classe contendo as informações para acesso do usuário ao sistema.
- *Tipo* - classe que representa o tipo do item (existem vários tipos de itens de acordo com seu fator de escolha na hora da avaliação).
- *Legislacao* – classe que representa todas as Resoluções existentes criadas pelo Conselho Federal de Medicina.

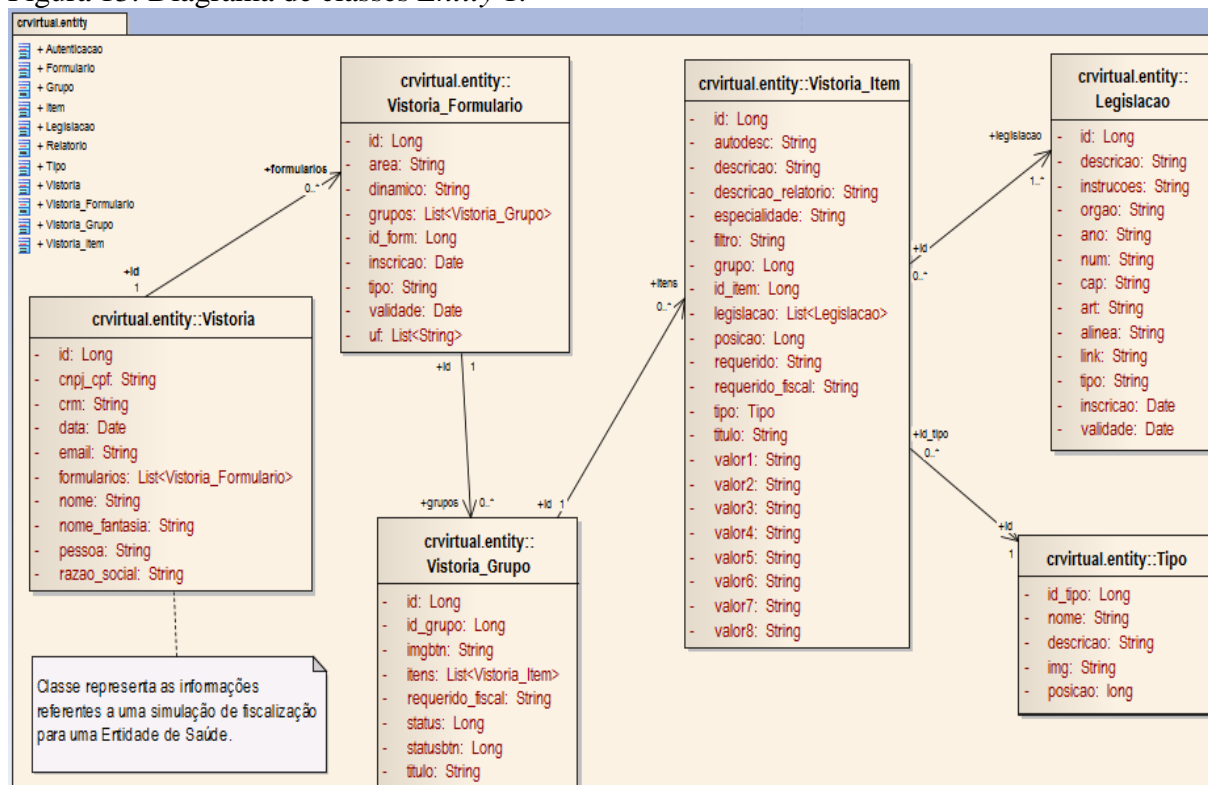
- *Item* – classe que representa os itens das questões. Contém uma relação de 0..* (Item - zero ou muitos) X 1..* (Legislacao - um ou muitos) com a classe *Legislacao* e uma relação de 0..* (Item - zero ou muitos) X 1 (Tipo - um) com a classe *Tipo*.
- *Grupo* – classe que representa as Questões do Formulário. Contém uma relação de 0..* (Grupo - zero ou muitos) X 0..* (Item - zero ou muitos) com a classe *Item*.
- *Formulario* - classe que representa os formulários dinâmicos que serão exportados em formato XML. Contém uma relação de 0..* (Formulario - zero ou muitos) X 0..*(Grupo - zero ou muitos) com a classe *Grupo*.

(Para simulação da fiscalização de uma Entidade de Saúde)

- *Vistoria_Item* – classe que representa as respostas de um Item na auto avaliação. Contém uma relação de 0..* (Item - zero ou muitos) X 1..* (Legislacao - um ou muitos) com a classe *Legislacao* e uma relação de 0..* (Item - zero ou muitos) X 1 (Tipo - um) com a classe *Tipo*. Além de uma referência ao *Item* que representa.
- *Vistoria_Grupo* – classe que representa as Questões respondidas na auto avaliação. Contém uma relação de 1 (Vistoria_Grupo - um) X 0..* (Vistoria_Item - zero ou muitos) com a classe *Vistoria_Item* e uma referência ao Grupo que está representando na auto avaliação.
- *Vistoria_Formulario* – classe que representa o formulário que está sendo respondido na auto avaliação. Contém uma relação de 1 (Vistoria_Formulario - um) X 0..* (Vistoria_Grupo - zero ou muitos) com a classe *Vistoria_Grupo* e uma referência ao Formulario que está representando na auto avaliação.
- *Vistoria* – classe contendo as informações referentes à entidade que está realizando a auto avaliação. Contém uma relação de 1 (Vistoria - um) X 0..* (Vistoria_Formulario - zero ou muitos) com a classe *Vistoria_Formulario*.

A figura abaixo apresenta o diagrama de classes do pacote *entity*, que representam as entidades do sistema para simulação da fiscalização de uma Entidade de Saúde.

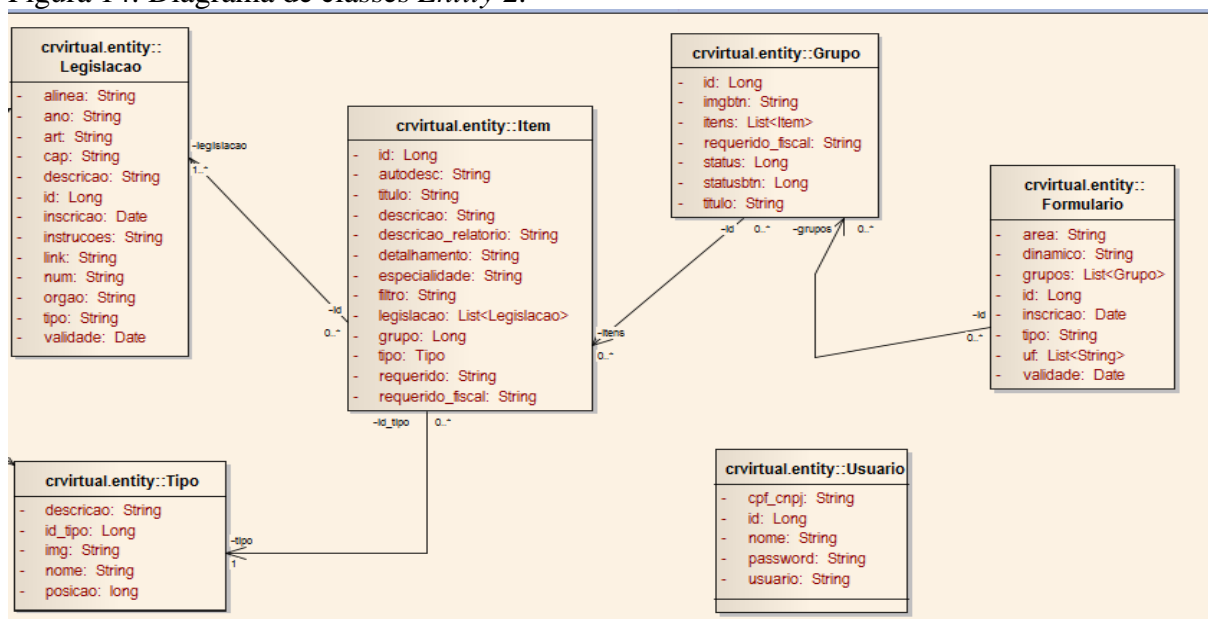
Figura 13: Diagrama de classes *Entity 1*.



Fonte: Autora do Trabalho, 2015.

A figura abaixo apresenta o diagrama de classes do pacote *entity*, que representam as entidades do sistema para funções de manipulação dos arquivos XML.

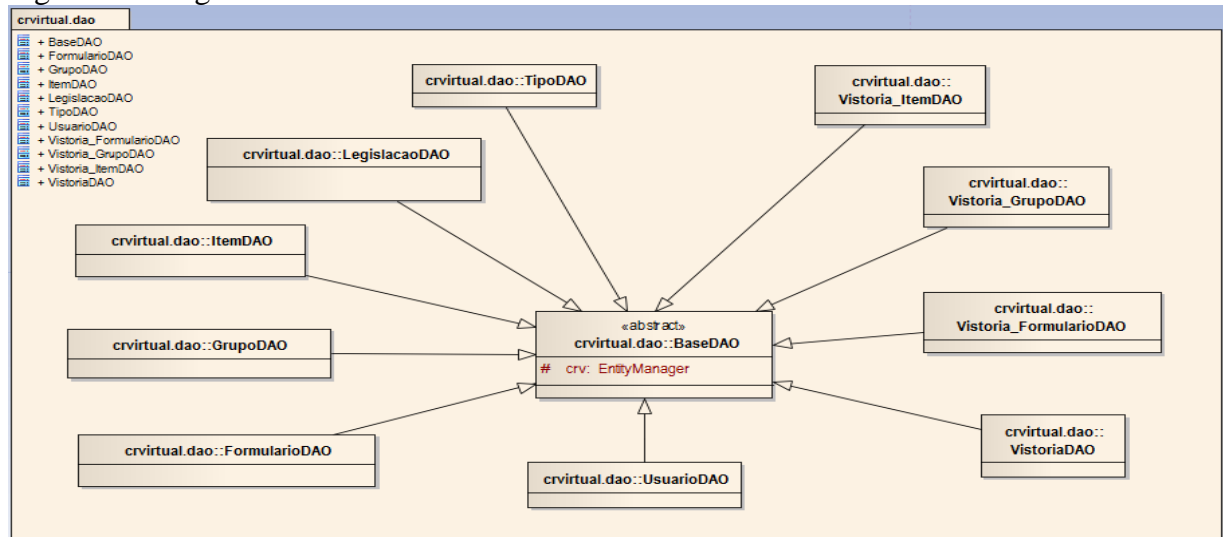
Figura 14: Diagrama de classes *Entity 2*.



Fonte: Autora do Trabalho, 2015.

A figura a seguir apresenta o diagrama de classes do pacote *dao*, que representam as relações das classes de entidades com o banco de dados. Do lado esquerdo as classes que contém os métodos para manipulação dos formulários dinâmicos e do lado direito as classes que contém os métodos para simulação da auto avaliação.

Figura 15: Diagrama de classes DAO.



Fonte: Autora do Trabalho, 2015.

4.6 CASOS DE USO

O diagrama de Casos de Uso representa os requisitos do Sistema. Após o levantamento de requisitos junto ao usuário, os analistas utilizam este diagrama para documentar as funcionalidades do sistema. (LOBO, p. 56, 2008).

De acordo com Lobo (p. 56, 2008), o diagrama de Casos de Uso pode ser criado para representar todas as funcionalidades do Sistema ou podem ser criados quantos diagramas forem necessários para representar estas funcionalidades. Normalmente, representamos o autor, ou seja, o usuário do sistema, trocando mensagens com “bolhas”, que representam as funções e recursos do sistema.

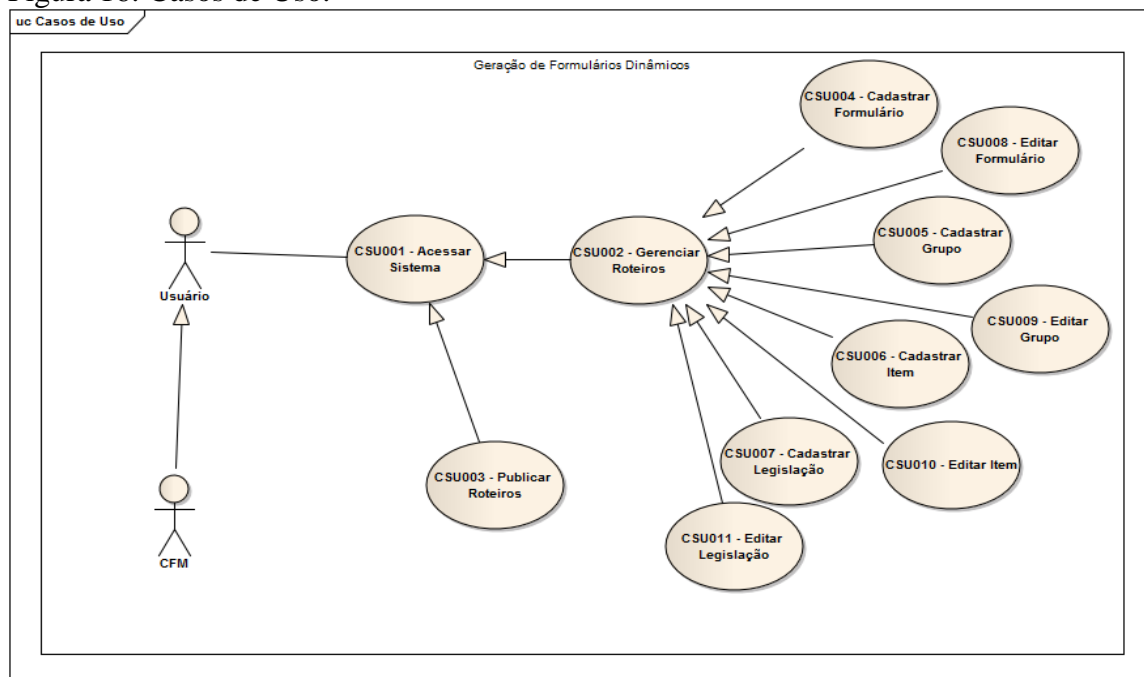
Segundo Ribeiro (2015), Diagramas de Casos de Uso são compostos basicamente por quatro partes:

- Cenário: Sequência de eventos que acontecem quando um usuário interage com o sistema.
- Ator: Usuário do sistema, ou melhor, um tipo de usuário.
- Use Case: É uma tarefa ou uma funcionalidade realizada pelo ator (usuário)
- Comunicação: É o que liga um ator com um caso de uso

Um dos Cenários mais importantes deste sistema aborda a funcionalidade de geração dos formulários dinâmicos e atualização dos dados deste, que é descrito da seguinte maneira: “O Conselho Federal de Medicina mediante necessidades reconhecidas pelos seus representantes da Lei, organizam e liberam nova Resolução de Legislações para o país, contatando a empresa que presta manutenção ao Sistema de Fiscalização da demanda de nova atualização nos dados deste sistema de acordo com a nova Resolução. Um funcionário desta empresa de TI responsável pela atualização dos formulários acessa o sistema aqui proposto com seus dados (usuário e senha), altera todos os formulários, os quais possuem seus dados armazenados em um Banco de Dados, se necessário criando novos formulários, questões, Itens, até que estes estejam de acordo com todas as normas descritas pela nova Resolução. Após as devidas modificações, o usuário do sistema exporta para sua máquina local estes formulários em formatos de arquivo XML para que possa ser substituído pelo seu respectivo no sistema de fiscalização”.

A figura a seguir representa o diagrama geral de Casos de Uso das funcionalidades do Sistema descritas acima.

Figura 16: Casos de Uso.



Fonte: Autora do Trabalho, 2015.

O quadro 11 apresenta a descrição do caso de uso: Acessar Sistema.

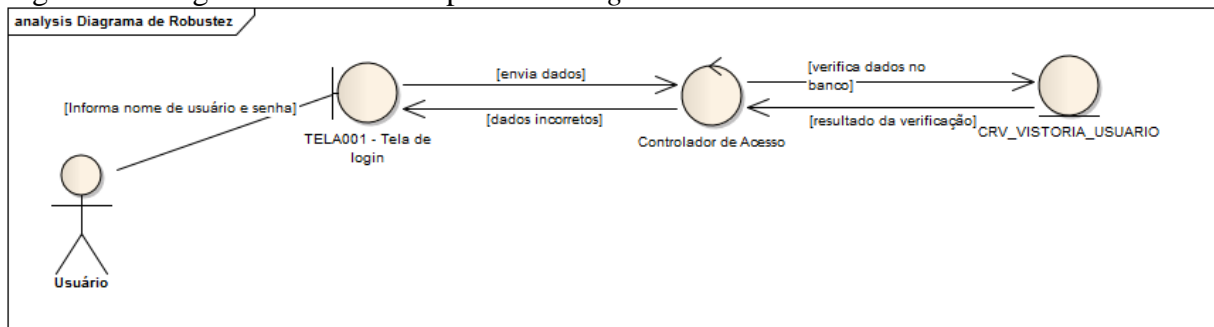
Quadro 11: Casos de Uso 001 – Acessar Sistema.

Nome do Caso de Uso	Acessar Sistema
Identificador	CSU001
Descrição	Acessar as funcionalidades do Sistema de forma segura e individual.
Ator	Usuário
Pré-Requisitos	- Deve-se acessar a tela de <i>Login</i> e possuir cadastro de Usuário no Sistema.
Cenário Principal	1. Usuário informa nome de usuário. 2. Usuário informa senha. 3. Usuário clica em botão “Acessar” 4. Sistema verifica autenticação dos dados informados pelo Usuário. 5. Sistema redireciona para tela de opções do Sistema.
Cenários Alternativos	<u>Dados incorretos:</u> 5. Sistema emite mensagem de erro no acesso na tela de <i>login</i>
Pós-Requisitos	- Usuário tem acesso às funcionalidades disponíveis no Sistema.

Fonte: Autora do Trabalho, 2015.

A figura abaixo apresenta o diagrama de Robustez para tela de acesso ao sistema.

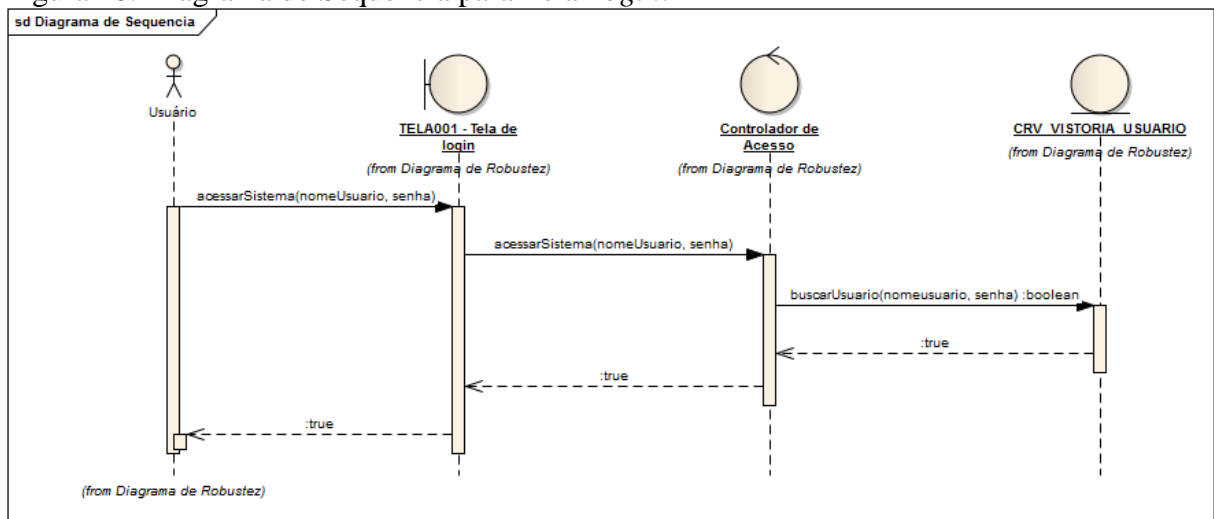
Figura 17: Diagrama de Robustez para Tela *Login*.



Fonte: Autora do Trabalho, 2015.

Mais detalhadamente é possível observar esta funcionalidade através do diagrama de Sequencia para o caso de uso Acessar Sistema.

Figura 18: Diagrama de Sequência para Tela *Login*.



Fonte: Autora do Trabalho, 2015.

O quadro 12 apresenta a descrição do caso de uso: Publicar Roteiros.

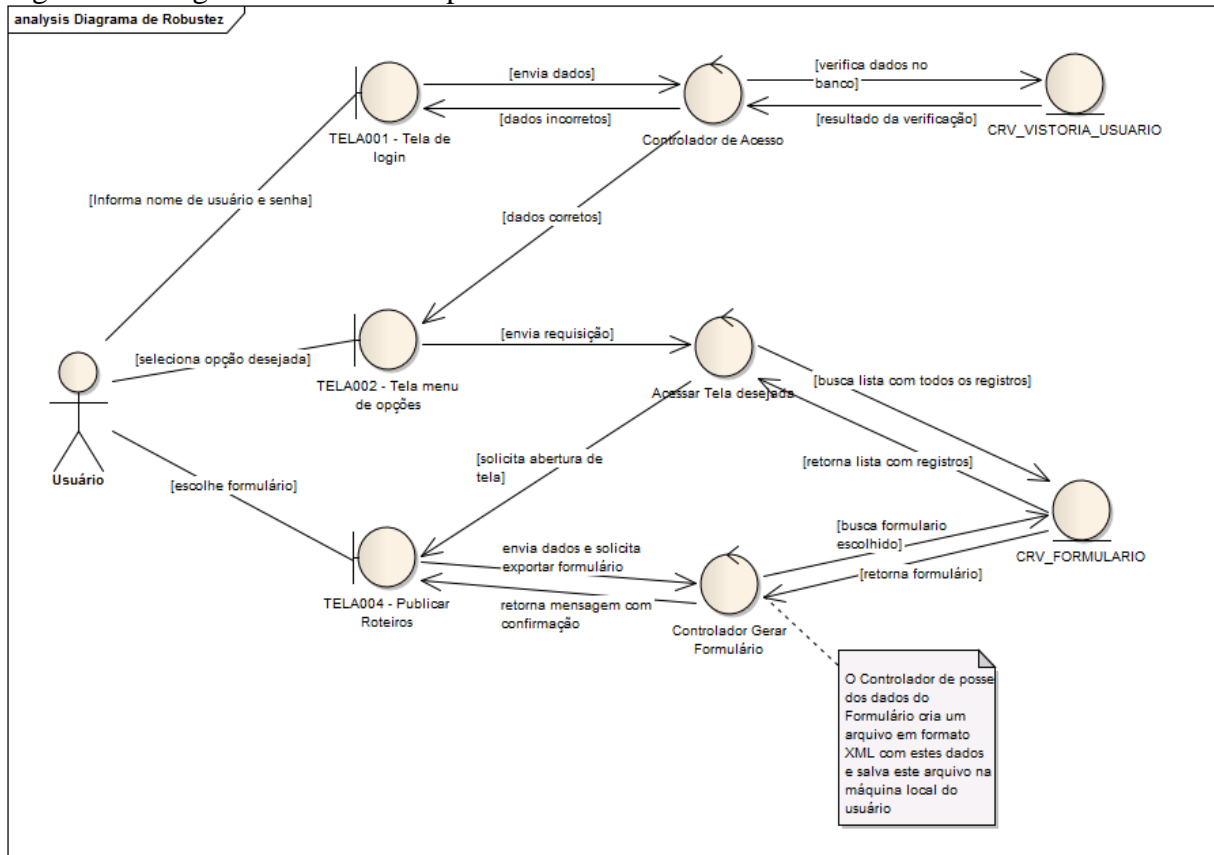
Quadro 12: Casos de Uso 003 – Publicar Roteiros.

Nome do Caso de Uso	Publicar Roteiros
Identificador	CSU003
Descrição	Gerar formulários dinâmicos como arquivos em formato XML. A Publicação de um Roteiro significa a extração de um Formulário Dinâmico que está gravado no Banco de Dados para a máquina local do usuário em formato compatível com o sistema de Fiscalização, no caso extensão XML.
Ator	Usuário
Pré-Requisitos	- Deve-se selecionar a opção “Publicar Roteiros” na tela de opções.
Cenário Principal	<ol style="list-style-type: none"> 1. Usuário escolhe o formulário que deseja exportar na lista que contém todos os formulários já cadastrados no sistema; 2. Sistema disponibiliza automaticamente o nome para o arquivo a ser exportado; 3. Usuário clica no botão “Publicar”; 4. Sistema gera um arquivo em formato XML a partir dos dados armazenados no banco e salva este arquivo na máquina local.
Cenários Alternativos	<u>Usuário informa o nome do arquivo:</u> Usuário digita o nome desejado para o arquivo a ser exportado; <u>Usuário desiste da operação:</u> <ul style="list-style-type: none"> • Usuário clica em voltar, retornando para a tela anterior.
Pós-Requisitos	- Arquivo XML gerado e salvo na máquina local.

Fonte: Autora do Trabalho, 2015.

A próxima figura apresenta o modelo de robustez para o caso de uso CSU003 (Publicar Roteiros), detalhando todos os passos desde o acesso ao sistema até a geração do arquivo XML.

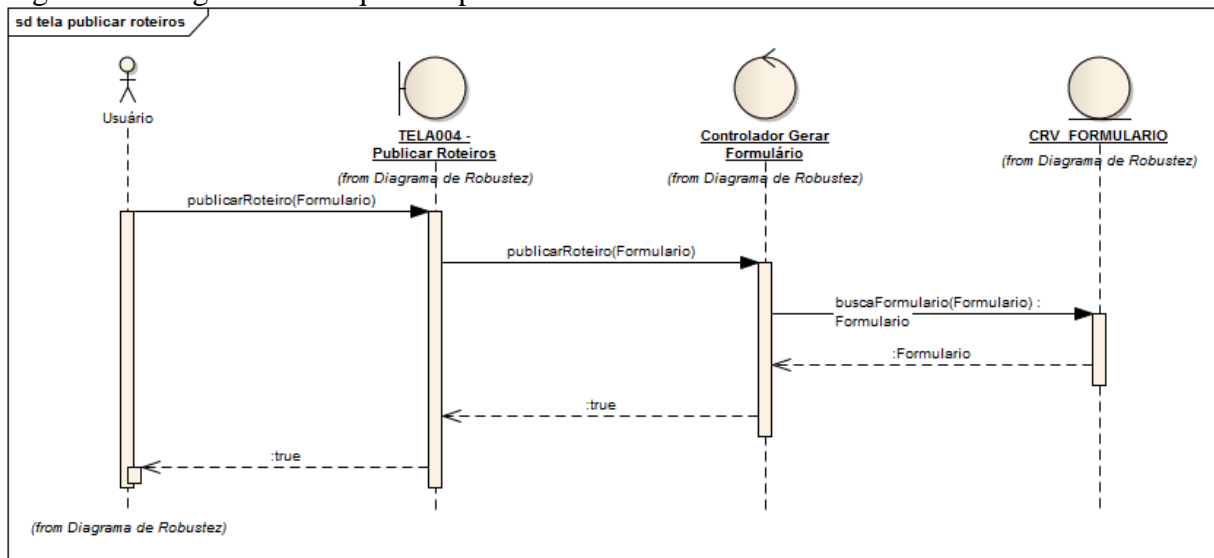
Figura 19: Diagrama de Robustez para Tela Publicar Roteiros.



Fonte: Autora do Trabalho, 2015.

Pode-se visualizar melhor este caso de uso através de seu diagrama de sequência mostrado abaixo.

Figura 20: Diagrama de Sequência para Tela Publicar Roteiros.



Fonte: Autora do Trabalho, 2015.

O quadro 13 apresenta a descrição do caso de uso: Cadastrar Formulário.

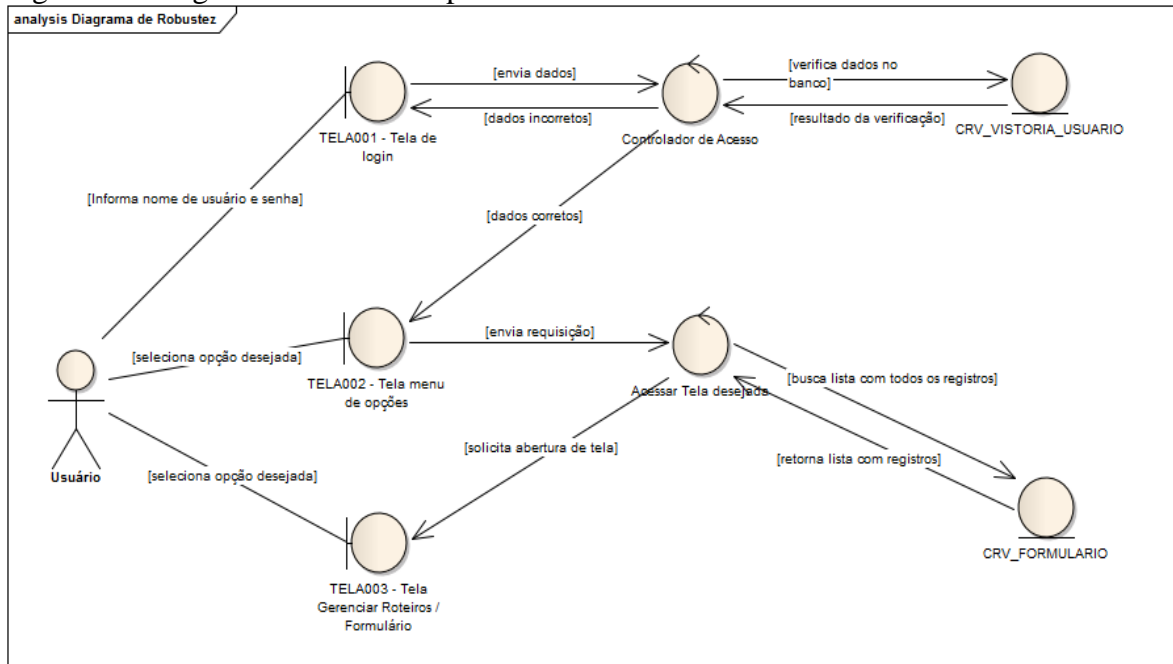
Quadro 13: Casos de Uso 004 – Cadastrar Formulário.

Nome do Caso de Uso	Cadastrar Formulário
Identificador	CSU004
Descrição	Cadastrar novo formulário dinâmico armazenando os dados no Banco de Dados. Os dados inseridos pelo Usuário serão armazenados em Banco de Dados para posterior reutilização, sendo possível a alteração do mesmo através de outra funcionalidade disponível no Sistema.
Ator	Usuário
Pré-Requisitos	- Deve-se selecionar a opção “Gerenciar Roteiros” na tela de opções e na tela de Aba Formulário clicar em botão “Incluir”.
Cenário Principal	1. Usuário digita e escolhe as informações disponibilizadas na tela e necessárias ao cadastro; 2. Usuário clica no botão “Salvar”; 3. Sistema armazena as informações inseridas pelo usuário no Banco de dados e retorna uma mensagem ao Usuário de confirmação do cadastro.
Cenários Alternativos	<u>Usuário desiste do cadastro:</u> • Usuário clica na opção de fechar a tela e volta à tela anterior. <u>Erro do Usuário:</u> 1. Usuário clica em “Salvar” antes de finalizar a inserção dos dados referentes ao formulário; 2. Sistema verifica se os campos obrigatórios foram preenchidos; a. Caso os campos estiverem devidamente preenchidos o Sistema armazena os dados no Banco de Dados e informa mensagem de confirmação do cadastro; b. Caso falem campos obrigatórios a serem preenchidos, o Sistema notifica o Usuário com mensagem de alerta sobre os campos que necessitam ser preenchidos antes de poder solicitar salvar o formulário.
Pós-Requisitos	- Formulário cadastrado no Banco de Dados.

Fonte: Autora do Trabalho, 2015.

A figura a seguir mostra o diagrama de robustez para o caso de uso CSU002 (Gerenciar Roteiros), detalhando como é realizado o acesso ao sistema pelo Usuário até chegar à tela de Gerenciar Roteiros.

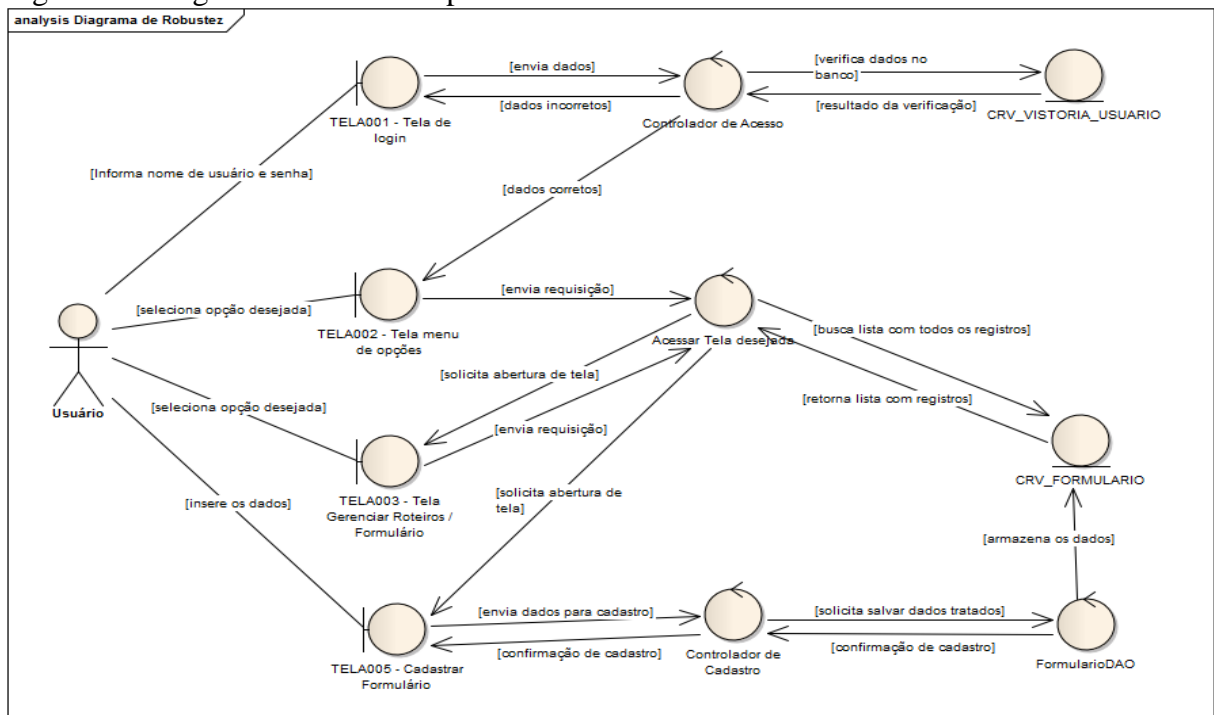
Figura 21: Diagrama de Robustez para Tela Gerenciar Roteiros.



Fonte: Autora do Trabalho, 2015.

O caso de uso CSU004 (Cadastrar Formulário) é representado no modelo de robustez a seguir, com início no acesso do Usuário ao sistema até a mensagem de confirmação de cadastro no banco de dados.

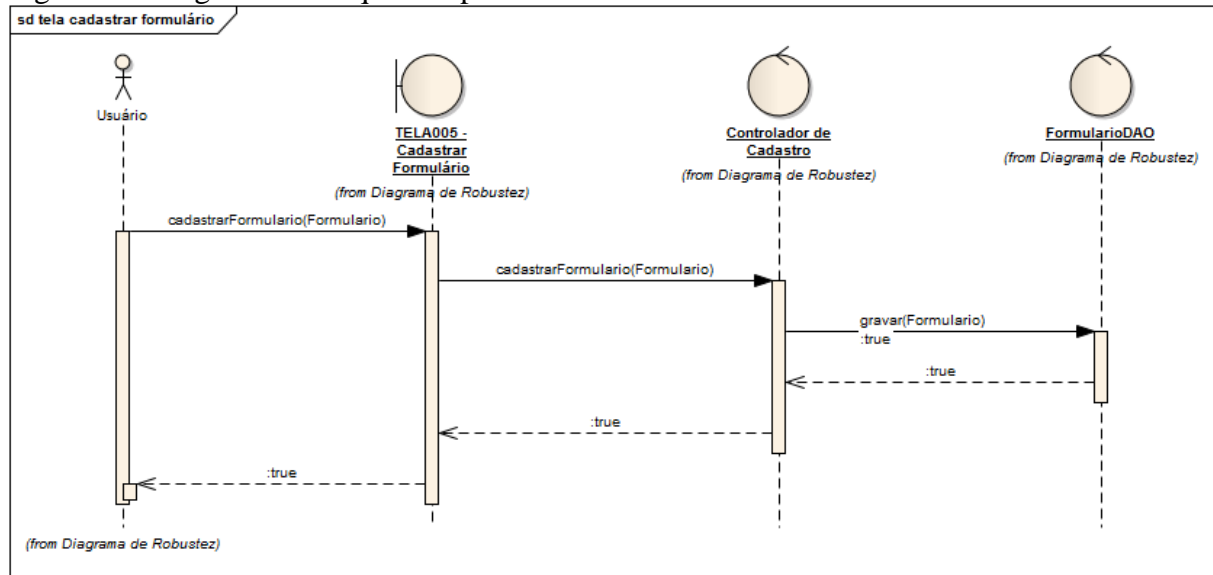
Figura 22: Diagrama de Robustez para Tela Cadastrar Formulário.



Fonte: Autora do Trabalho, 2015.

A figura a seguir apresenta o diagrama de sequencia para o caso de uso Cadastrar Formulário.

Figura 23: Diagrama de Sequência para Tela Cadastrar Formulário.



Fonte: Autora do Trabalho, 2015.

4.6.1 Protótipos de Tela

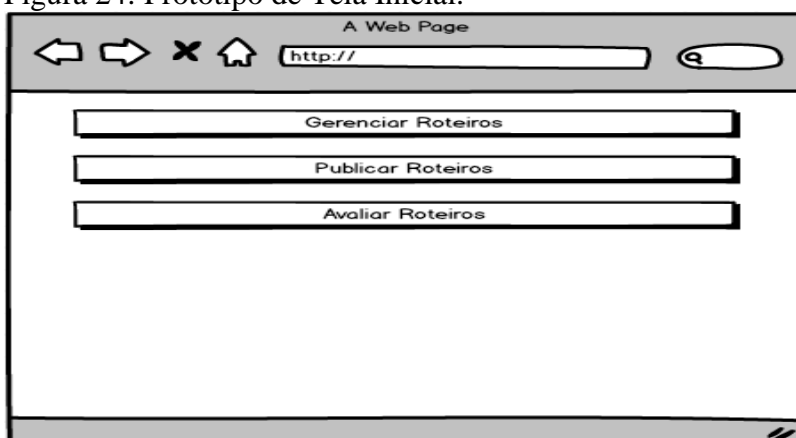
Sabemos que, por mais cuidadosos que sejamos no levantamento de requisitos, sempre falta alguns detalhes só descobertos na entrega do sistema: requisitos não lembrados, não revelados pelo usuário, mal compreendidos e/ou mal validados. (MELO, p. 72, 2010)

Por este motivo é fator importante que possa ser utilizado o máximo possível os recursos disponíveis para prevenir erros e facilitar e agilizar o processo de desenvolvimento. Um destes recursos é a prototipação das telas do sistema. Segundo Melo (p.72, 2010) o protótipo de tela é um tipo de rascunho das telas originais que, ao ser mostrado ao usuário, aumenta a segurança quanto à validação de requisitos. A orientação a objetos ajuda a dar vida a estes protótipos (sem correr o risco de codificar e depois remendar), por meio de utilização de classes prontas, já testadas.

Os protótipos de tela apresentados são referentes aos casos de uso CSU002 (Gerenciar Roteiros), CSU003 (Publicar Roteiros) e CSU004 (Cadastrar Formulário), sendo as duas funcionalidades mais relevantes do sistema proposto. Para criação destes protótipos é utilizado a ferramenta *Balsamiq Mockups*, disponível para trabalhar online, sem necessidade de instalação na máquina.

A figura abaixo apresenta a tela inicial do sistema após o acesso do usuário com seu nome de usuário e senha, informando as opções disponíveis separadas de acordo com tipo de função, representando o caso de uso CSU002 (Gerenciar Roteiros).

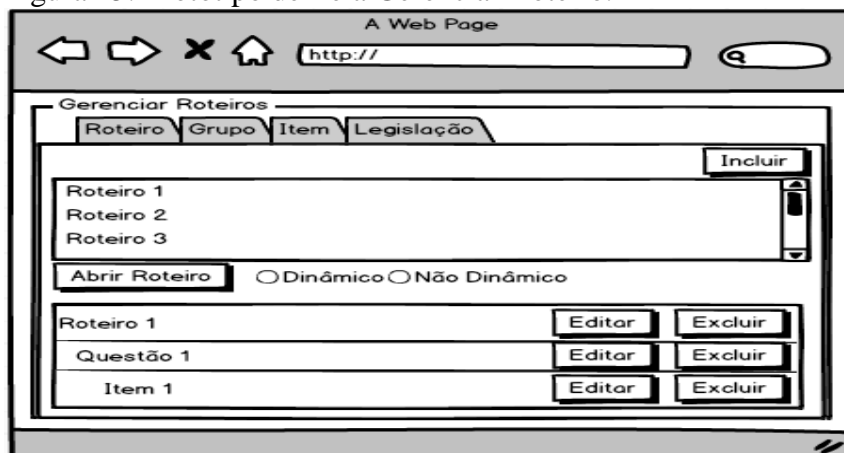
Figura 24: Protótipo de Tela Inicial.



Fonte: Autora do Trabalho, 2015.

A figura a seguir apresenta a tela de Gerenciamento de Roteiros aba Roteiro, abrangendo os casos de uso CSU004 (Cadastrar Formulário) e CSU008 (Editar Formulário). Cada aba contém as funcionalidades para atender aos requisitos do Item que a denomina.

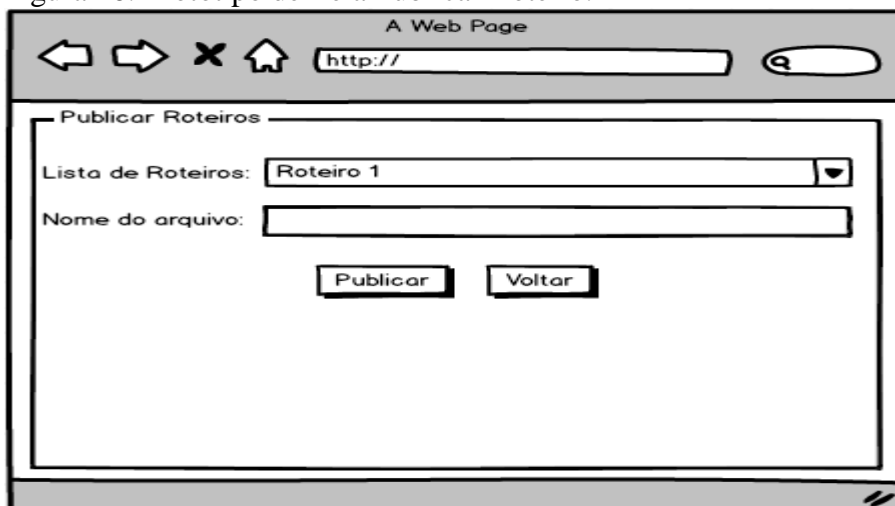
Figura 25: Protótipo de Tela Gerenciar Roteiro.



Fonte: Autora do Trabalho, 2015.

Abaixo é apresentado o protótipo para esta tela que define o caso de uso CSU003 (Publicar Roteiros).

Figura 26: Protótipo de Tela Publicar Roteiro.



A Web Page

Publicar Roteiros

Lista de Roteiros: Roteiro 1

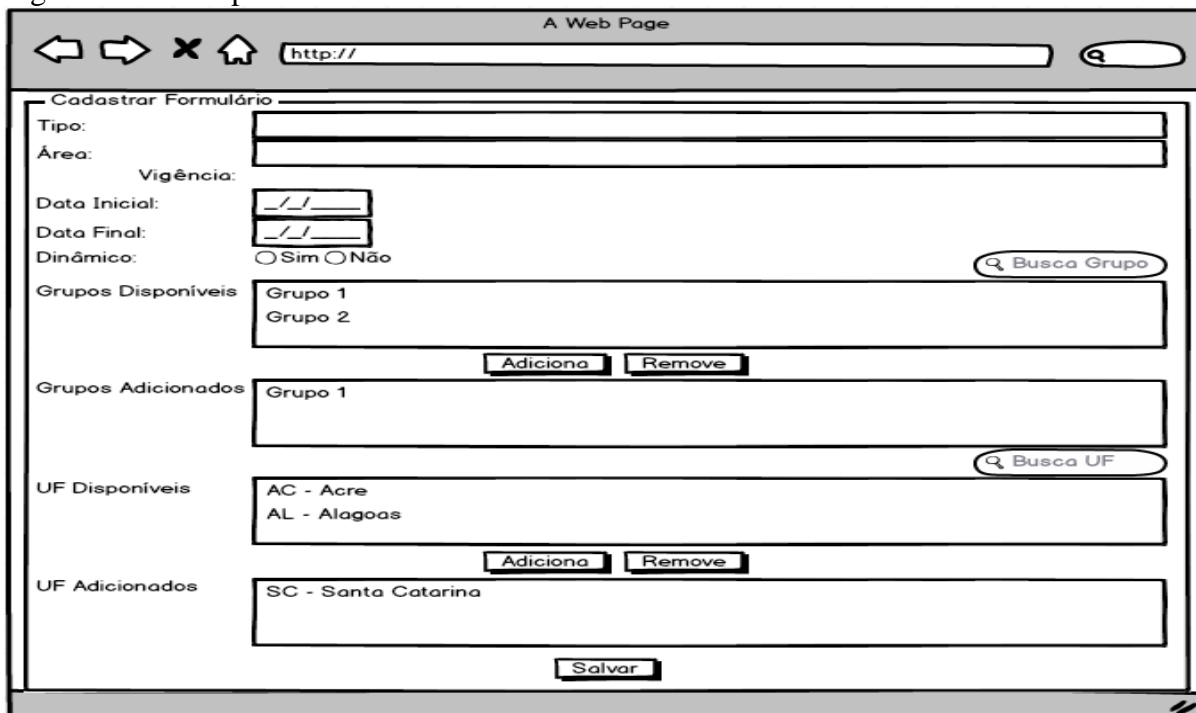
Nome do arquivo:

Publicar Voltar

Fonte: Autora do Trabalho, 2015.

O caso de uso CSU004 (Cadastrar Formulário) é apresentado na figura a seguir com seus componentes.

Figura 27: Protótipo de Tela Publicar Roteiro.



A Web Page

Cadastrar Formulário

Tipo:

Área:

Vigência:

Data Inicial: / /

Data Final: / /

Dinâmico: ☐ Sim ☐ Não

Busca Grupo

Grupos Disponíveis

Grupo 1

Grupo 2

Adiciona Remove

Grupos Adicionados

Grupo 1

Busca UF

UF Disponíveis

AC - Acre

AL - Alagoas

Adiciona Remove

UF Adicionados

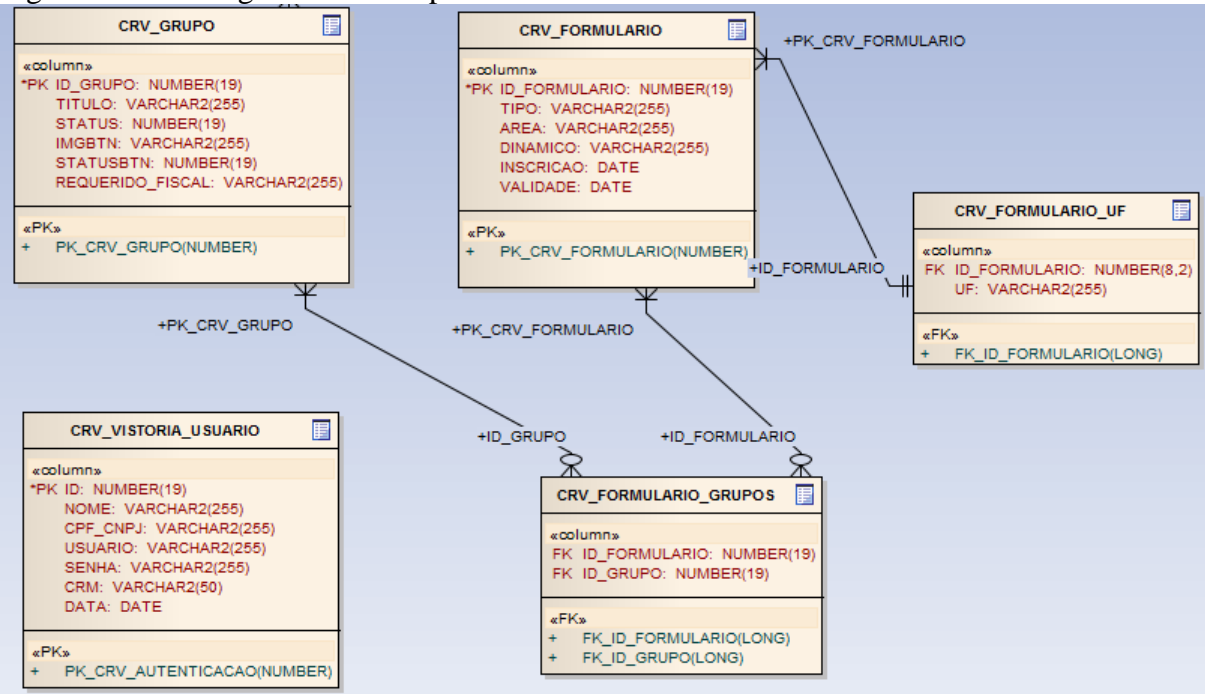
SC - Santa Catarina

Salvar

Fonte: Autora do Trabalho, 2015.

A segunda figura representa as seguintes tabelas: tabela que armazena os Formulários (CRV_FORMULARIO), o Estado correspondente ao Formulário (CRV_FORMULARIO_UF), os dados de acesso do Usuário ao sistema (CRV_VISTORIA_USUARIO) e a relação entre a tabela de Questões com a tabela de Formulário (CRV_FORMULARIO_GRUPOS).

Figura 29: Modelagem de Dados parte 2.



Fonte: Autora do Trabalho, 2015.

4.8 CONSIDERAÇÕES FINAIS

Este capítulo apresentou a modelagem do Sistema proposto. Foram apresentados os requisitos funcionais e não funcionais do sistema. O Diagrama de classes e modelagem UML bem como a descrição de alguns dos casos de Uso mais relevantes ao sistema proposto. Por se tratar de um sistema com várias funcionalidades, uma aplicação complexa, não será abordado todos os casos de uso e demais diagramas que competem às outras funcionalidades menos utilizadas pelo sistema, a fim de não prolongar excessivamente a leitura.

5 DESENVOLVIMENTO

Nesta etapa da monografia é apresentado as tecnologias utilizadas para o desenvolvimento do sistema proposto, a justificativa para cada escolha, as dificuldades encontradas para utilização das mesmas e onde cada uma se enquadra dentro do Sistema. Na continuação é apresentado o Sistema desenvolvido e sua avaliação.

5.1 TECNOLOGIAS UTILIZADAS

As tecnologias escolhidas para desenvolvimento do sistema proposto serão apresentadas e descritas resumidamente neste tópico, bem como o motivo de sua utilização, que é geralmente em função da demanda do cliente em já utilizar e estar habituado a estas ferramentas, ou pela facilidade e conhecimento destas tecnologias por parte do desenvolvedor.

A figura abaixo apresenta uma relação das ferramentas utilizadas para a implementação do sistema descrito nesta monografia.

Figura 30: Tecnologias utilizadas.



Fonte: Autora do Trabalho, 2015.

5.1.1 Linguagem de Programação Java

Segundo Augusto (p. 3, 2002), algumas das características da linguagem Java, a qual foi escolhida para realizar o desenvolvimento deste trabalho, são citadas abaixo:

- Simples – Java tem por base a linguagem C++, porém ela omite muitas características que são tidas como problemas em C++ como, por exemplo, os ponteiros, herança múltipla, sobrecarga de operadores, entre outros. Além de possuir a coleta de lixo automática facilitando a programação.

- Orientada a Objetos – Java possui portabilidade, tem herança, mas não herança múltipla, porém isto é compensado com interfaces. Esta linguagem também permite criação e classes abstratas.

- Distribuída – possui biblioteca extensa para lidar com protocolos TCP/IP.

- Robusta – Possui uma verificação antecipada para erros, verificação dinâmica posterior e eliminação de situações sujeitas a erros. A diferença entre esta para o C++ está em que o Java usa um modelo de ponteiros que elimina a possibilidade de sobrescrita de memória e consequente destruição de dados.

- Segura – A presença de coleta automática de lixo, evita erros comuns que os programadores cometem quando são obrigados a gerenciar diretamente a memória. Este detalhe traz benefícios à segurança, como o fato do programador não ter acesso à memória que não pertence a seu programa e não ter chances de cometer erros comuns, entre outros. Java possui mecanismos de tratamento de exceções, o que a torna mais robusta, não permitindo que o programa seja abortado, mesmo rodando sob condições anormais.

Dependendo da linguagem de programação que escolhermos para desenvolver um sistema, teremos maior ou menor facilidade para realizar as operações. A Linguagem de programação Java foi escolhida para desenvolvimento deste sistema pelos motivos descritos acima e também devido a ser a linguagem estudada durante todo o curso de Graduação da autora da Monografia. Portanto o conhecimento desta linguagem comparado com as demais existentes é maior, facilitando a programação do sistema, além de ser uma linguagem de fácil aprendizado.

5.1.2 Tecnologias do Java EE

De acordo com Cleuton Sampaio (p.21, 2011), a plataforma Java EE (*Java Enterprise Edition*) vem com vários serviços implementados pelas diversas tecnologias incorporadas ao framework. Entre elas pode-se citar:

- Java Servlet – Permite a criação de classes servidoras que podem ser instaladas em Containers Java EE.
- JavaServer Faces – Modelo de desenvolvimento de páginas RIA (*Rich Internet Application*).
- JavaServer Pages – Permite a criação de páginas dinâmicas de maneira simples e prática.
- Enterprise JavaBeans – Criação de componentes de negócio remotos.
- Java Persistence – Abstração objeto-relacional para acesso a banco de dados.
- Java API for XML-Based Web Services – Extensão do JAX-RPC para criação de Web Services.
- Java API for RESTful Web Services – Implementação da arquitetura REST em Web Services.

Essas tecnologias são as mais comuns, mais existem diversas outras que podem ser encontradas no website: <http://java.sun.com/javaee/technologies/>. Por se tratar de um software com interface Web, as tecnologias de Java EE utilizadas para desenvolvimento deste trabalho são JavaServer Pages, Java Persistence, Java Servlet.

Java Persistence (JPA) é utilizado nesta Monografia para o relacionamento do Sistema com o Banco de Dados Relacional. JavaServer Pages (JSP) é utilizado em substituição do HTML pelo fato de permitir inserir conteúdo dinâmico (dados oriundos do Banco de Dados) em suas páginas, facilitando a busca e apresentação dos dados já registrados. Java Servlet foi fundamental para o relacionamento das páginas com o Servidor da aplicação. Pode-se dizer que é o caminho que liga um (página) ao outro (servidor) e possibilita realizar as diversas ações necessárias ao bom desempenho das funcionalidades do sistema.

5.1.3 Desenvolvimento de Aplicações Web

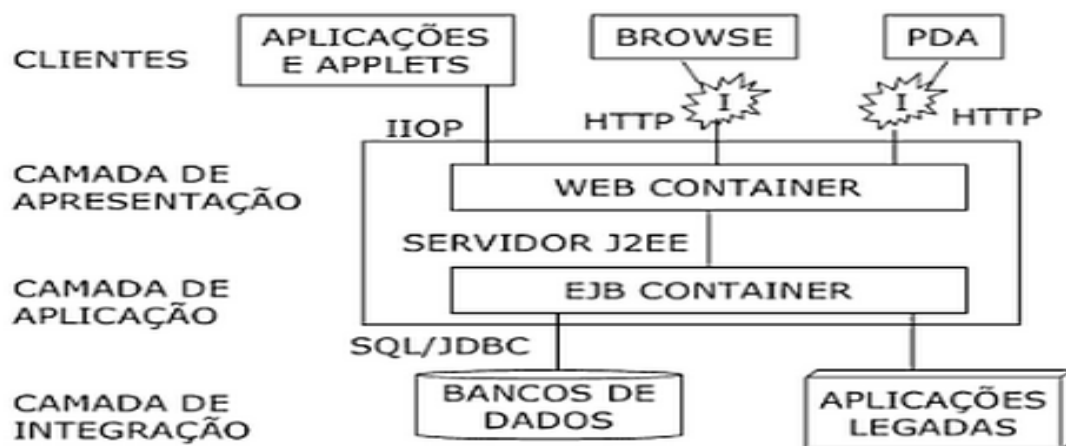
O Software descrito por esta Monografia pode ser denominado como sendo uma aplicação Web Java EE, conforme descrições a seguir apresentadas.

Uma aplicação Web Java EE é um conjunto de programas Java, páginas HTML, imagens, sons e outros arquivos disponibilizados através de um Web Container. Na terminologia Java EE, um contexto de aplicação (*application context*) é uma aplicação Web instalada dentro de um Web Container, incluindo todos os seus recursos. Um contexto é identificado por seu caminho de acesso, ou URL – *Uniform Resource Locator*. Como por exemplo: <http://localhost:8080/hello/index.html>. (SAMPAIO 2011, p.42).

A internet e seus padrões impuseram um modelo de programação no qual o cliente deve ser o navegador Web e o servidor deve resolver toda a complexidade da aplicação, além de responder o solicitado no mais simples HTML possível. Para atender a esta especificação, Java EE disponibiliza modelo de programação baseada na Web e em componentes de negócios gerenciados por um servidor de aplicação JEE. Esse servidor de aplicação constitui-se em um conjunto de API e serviços de nível básico disponível para os componentes. Como serviços de nível básico podemos citar: segurança, suporte a transação, conexões, gerenciamento de pooling, serviços de concorrência. Todas essas facilidades devem favorecer o desenvolvedor, que pode focar exclusivamente sobre a lógica de negócio a ser implementada. (ALBINADER NETO e LINS 2006, p.29).

Na arquitetura Java EE destaca-se a divisão em três camadas que pode ser visto na figura a seguir:

Figura 31: Arquitetura de aplicação J2EE.



Fonte: (ALBINADER NETO e LINS 2006, p.31).

Uma URL usada para acessar uma página possui os elementos destacados na figura a seguir:

Figura 32: Elementos de uma URL.

	Http://	Localhost	:8080	/hello	/index.html
Protocolo	HTTP				
Servidor		Localhost			
Porta			8080		
Contexto				Hello	
Path-info					/index.html

Fonte: SAMPAIO 2011, p.43.

Segundo Cleuton Sampaio (p.43, 2011), o protocolo será sempre HTTP para páginas e arquivos acessados via Web. Os outros elementos são:

- Servidor: nome do domínio, endereço IP ou nome NetBIOS do servidor Web. Caso a porta utilizada não seja a padrão (80), podemos informa-la após “:”.

- Porta: a porta TCP que o servidor está escutando. Se for a porta 80, não precisamos informar.

- Contexto: nome do contexto da aplicação Web. Pode-se ter vários Web Sites em um mesmo servidor e separá-los pelo contexto. Quando o contexto não é informado, assume-se que será no diretório RAIZ do servidor. O diretório raiz pode ser representado por uma barra “/”.

Apesar de que desenvolver aplicações para rodar em um navegador Web possa ser um pouco mais confuso do que um desenvolvimento de uma aplicação do tipo desktop, os esforços são justificados porque aplicações Web não necessitam de instalação, já que a distribuição e atualizações destas são realizadas no servidor, e as atualizações de novas versões são automáticas. Outro fator muito importante para definir a escolha do tipo de aplicação desta Monografia é o fato dos usuários poderem utilizar o sistema de qualquer computador com uma conexão com a internet, e normalmente é indiferente o sistema operacional que o computador está rodando. Como trata-se de um sistema em nível nacional, a manutenção a este seria difícil se fosse do tipo desktop. Aplicativos baseados na Web são também geralmente menos propensos a uma infecção de vírus do que um executável rodando na máquina.

Segundo Dourado (2015), se a aplicação estiver em um ponto centralizado, onde seu acesso for distribuído, onde as alterações são efetuadas em apenas uma única versão da

aplicação, e não exista a necessidade de instalação desta aplicação nas máquinas do usuário, então há uma melhor disponibilidade da aplicação, assim como um melhor controle sobre suas alterações. Este ponto centralizado veio com a utilização da plataforma web no desenvolvimento de aplicações. No desenvolvimento Web, o cliente usa um navegador (browser), necessitando apenas a instalação do navegador em sua máquina. Como a utilização dos navegadores foi difundida com a popularização da Internet, a plataforma web foi beneficiada neste ponto.

5.1.4 Servlets

Servlet é uma classe instalada em um servidor de aplicação Java EE, que processa pedidos HTTP enviados para um browser. Na verdade foi a primeira forma de criar aplicações Web Java. A especificação Servlet foi desenvolvida em 1997, antes mesmo de existir o pacote Java EE. (SAMPAIO 2011, p.47).

De acordo com Oziel Moreira Neto (p.30, 2006): “O nome Servlet significa pequena aplicação para um servidor, ou seja, podemos usar um servlet para criar elementos de aplicação que atendam requisições de um cliente remoto (em rede), formatando uma resposta”.

Embora não seja mais necessário programar Servlets para criar Websites em Java, ela será especificada neste trabalho devido ao seu uso no desenvolvimento do sistema proposto.

Os Servlets são classes Java descendentes de `javax.servlet.http.HttpServlet`, uma classe que fica no pacote do Container Java EE. Para criar uma Servlet, temos que estender a classe `HttpServlet` e fazer *override* dos métodos necessários. (SAMPAIO 2011, p.47).

Segundo Oziel Moreira Neto (p.30, 2006), os servlets são *multithread* e multiusuários por natureza, sendo assim, o programador não precisa se preocupar em desenvolver elementos para suportar essas características. Nesse caso é o web container Java EE que se encarrega de controlar as *threads* e os usuários.

Alguns conceitos importantes dentro dos Servlets retirados do livro de Oziel Moreira Neto (p.31, 2006):

Quadro 14: Conceitos em Servlets.

Conceito	Definição	Objeto/Métodos mais usados
Sessão	Área de memória dentro do servidor web usada para armazenar informações de um usuário durante seu contato com a aplicação.	javax.servlet.http.HttpSession.setAttribute() javax.servlet.http.HttpSession.getAttribute()
Contexto	Área de memória dentro do servidor web usada para armazenar informações que podem ser compartilhadas por todos os usuários.	javax.servlet.ServletContext.setAttribute() javax.servlet.ServletContext.getAttribute()
Requisição	Uma ação do usuário que foi convertida num comando, entendida pelo servidor web que executará um determinado processo, formatando e desenvolvendo uma resposta.	javax.servlet.http.HttpServletRequest.getParameter() javax.servlet.http.HttpServletRequest.getAttribute() javax.servlet.http.HttpServletRequest.setAttribute()
Resposta	Todo e qualquer resultado gerado dinamicamente dentro de um componente web.	javax.servlet.http.HttpServletResponse.getWriter() javax.servlet.http.HttpServletResponse.setStatus() javax.servlet.http.HttpServletResponse.sendRedirect()
Cabeçalho	São conteúdos de dados padronizados para facilitar o entendimento das requisições pelo servidor e do conteúdo desenvolvido para o cliente.	javax.servlet.http.HttpServletResponse.addHeader() javax.servlet.http.HttpServletResponse.getHeader()
Cookies	São pequenos conteúdos de dados, trocados entre o cliente e o servidor, para garantir a sessão do usuário e prover um mecanismo de armazenamento de informações do lado do cliente e que pode ser seguramente acessado pela aplicação.	javax.servlet.http.HttpServletResponse.addCookie() javax.servlet.http.HttpServletResponse.getCookie()

Fonte: Autora deste trabalho, 2014.

Servlets são classes Java para Web, uma classe normal como qualquer outra, porém com métodos diferentes, por exemplo, não tem o método *main* como nas classes padrões Java. Eles recebem uma requisição (*request*) e produzem uma resposta (*response*). É de fácil utilização, quem conhece as funcionalidades de uma classe java padrão não encontra muitas dificuldades em utilizar Servlet, apenas estudar o que cada método deste oferece e as formas de retorno disponíveis.

5.1.5 Filtros

Como as aplicações desenvolvidas para Web tem necessidades específicas com relação ao tratamento das requisições, bem como em relação à identificação do tipo de cliente

com o qual está se comunicando, um poderoso recurso está disponível na plataforma Java Enterprise com a característica de poder filtrar, baseado num padrão de URL, as requisições feitas aos componentes Web. (MOREIRA NETO 2006, p.62).

É possível criar componentes especiais para atuarem antes ou depois dos pedidos HTTP serem processados. Estes componentes atuam como aspectos da aplicação, executando funções adicionais não programadas nos Servlets. (SAMPAIO 2011, p.67).

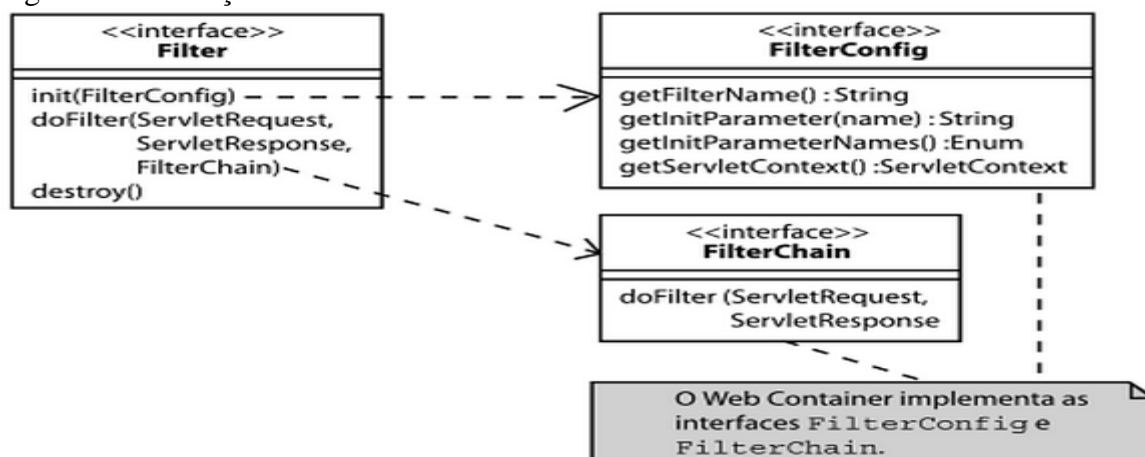
De acordo com autor citado acima, um filtro “é um componente que recebe o pedido (*request*) e a resposta (*response*) HTTP, podendo processá-los, alterá-los, etc”. Os filtros são processados em cadeia (*chain*), sendo que um filtro após seu processamento, deve invocar o próximo filtro (quando há mais de um).

Podem-se fazer várias coisas com um filtro, entre elas pode-se destacar: a implementação de um mecanismo de segurança; comprimir ou descomprimir o pedido e a resposta; redirecionar o pedido. Abaixo alguns métodos pertencentes ao Filtro: (SAMPAIO 2011, p.71).

- Init – é invocado quando o filtro está sendo adicionado ao contexto. Nele podemos pegar parâmetros de inicialização utilizando a classe “FilterConfig”;
- Destroy – é chamado quando o filtro está para ser desativado;
- doFilter – é chamado quando o filtro foi invocado na cadeia de filtros do contexto (*filter chain*). Neste caso podemos fazer o que quisermos com o pedido e com a resposta, mas temos que terminar invocando o método filtro na cadeia.

Segundo Oziel Moreira Neto (p.63, 2006) é possível definir mais de um filtro para um determinado componente, podendo tratar adequadamente a requisição, realizando operações de log, conversão de tipos de dados específicos para cada tipo de cliente. Para definir um filtro deve-se implementar a interface `javax.servlet.Filter`.

Figura 33: Definição de Filtro.



Fonte: MOREIRA NETO 2006, p.63.

O sistema desenvolvido nesta Monografia utiliza o recurso de Filtros para autenticação e autorização de acesso às suas páginas e funcionalidades, tornando as informações dispostas no Software mais seguras. Ele também é um excelente recurso para realizar determinada função (que deve ser repetitiva) sempre que acessar determinada página ou Servlet, como por exemplo, armazenar informações da máquina cliente (IP, navegador, etc) ao tentar realizar determinada função, facilitando a análise e correção em caso de falhas nestas ações, mantendo um rastro de todo o caminho que o usuário percorreu.

5.1.6 Java Server Pages

Java Server Pages – JSP – é uma tecnologia baseada em Java que simplifica o processo de desenvolvimento de sites da web dinâmicos. Com JSP, os designers da web e programadores podem rapidamente incorporar elementos dinâmicos em páginas web usando Java embutido em algumas *tags* de marcação simples. Essas *tags* fornecem ao designer de HTML um meio de acessar dados e lógica de negócios armazenados em objetos Java sem ter que dominar as complexidades do desenvolvimento de aplicações. (FIELDS E KOLB, 2000)

Considere a JSP como um tipo de linguagem de criação de scripts no servidor. Java Servers Pages são arquivos de texto, normalmente com a extensão .jsp, que substituem as páginas HTML tradicionais. Os arquivos JSP contém HTML tradicional junto com código

embutido que permitem que o designer de páginas acesse dados do código de Java rodando no servidor. Quando a página é solicitada por um usuário e processada pelo servidor de HyperText Transport Protocol (HTTP), a parte HTML da página é transmitida. No entanto as partes de código da página são executadas no momento em que a solicitação é recebida e o conteúdo dinâmico gerado por este código é unido na página antes de ser enviado para o usuário. Isto propicia uma separação dos aspectos de apresentação HTML da página, da lógica de programação contida no código, um benefício único. (FIELDS E KOLB, 2000).

Esta tecnologia foi escolhida para ser utilizada neste Software pela necessidade de colocar conteúdo dinâmico (oriundo do Banco de Dados) nas páginas Web. Apesar de ser uma linguagem mais antiga, JSP é mais conveniente para desenvolvimento utilizando Java pois sua sintaxe de código é baseado na Linguagem de programação Java.

5.1.7 CSS - Folha de Estilo em Cascata

De acordo com Maurício Samy Silva (p.49, 2008), CSS é a abreviação para os termos em inglês *Cascading Style Sheet*, traduzido para o português como folhas de estilo em cascata.

Segundo Flávia Jobstraibizer (p.6, 2009), CSS é uma linguagem de estilo (ou de formatação) muito apropriada para a construção do layout de páginas ou sites. Permite projetar páginas e até mesmo sistemas com uma técnica completamente diferente da convencional e possibilita uma considerável redução de tempo de trabalho.

O CSS é utilizado para definir a apresentação de documentos escritos em uma linguagem de marcação, como HTML ou XML. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento. (JOBSTRAIBIZER 2009, p.6).

O funcionamento é bem simples, ao invés de colocar a formatação dentro do documento original, formatando-o com as tags de marcação do HTML ou derivados, o desenvolvedor cria um link ou mesmo importa um arquivo de folha de estilos na página que contém informações a serem formatadas e então “chama” esses estilos, que uma vez aplicados, resultarão na página formatada conforme definido no arquivo CSS. (JOBSTRAIBIZER 2009, p.6).

Deste modo quando se tem a necessidade de alterar a formatação de uma página altera-se somente o conteúdo do arquivo CSS, o que possibilita ao desenvolvedor maior agilidade e facilidade nas alterações. Por estes motivos acima descritos, sua utilização no sistema proposto nesta monografia foi fundamental, organizando em arquivos distintos as formatações das páginas, facilitando a manutenção.

5.1.7.1 Regras do CSS

Segundo Maurício Samy Silva (p.51, 2008), a Regra CSS é a unidade básica (menor porção de código capaz de produzir um efeito de estilização) de uma folha de estilo. Uma Regra CSS é composta de duas partes: o *seletor* e a *declaração*, sendo que a declaração compreende uma propriedade e um valor.

Figura 34: Sintaxe da regra CSS.



Fonte: Maurício Samy Silva, p.51, 2008.

O *seletor* é o alvo da regra CSS. Genericamente é a tag do elemento da marcação ou uma entidade capaz de definir com precisão em qual lugar da marcação será aplicada a regra CSS. A *declaração* determina os parâmetros de estilização. Compreende a propriedade e o valor. A *propriedade* define qual será a característica do elemento, alvo do seletor, a ser utilizada. O *valor* é a quantificação ou a qualificação da propriedade. (SAMY SILVA, p.51, 2008).

De acordo com autor citado acima, uma regra CSS pode conter mais de uma declaração de estilo, o que permite que se escrevam todas as declarações de estilo em uma mesma regra. O Software aqui especificado possui vários seletores para formatar suas

páginas, que são organizados em grupos, definidos pelo nome dos componentes e separados em arquivos distintos dependendo da funcionalidade e generalidade.

5.1.8 Javascript

De acordo com José Antônio Ramalho (p.191, 2005), a linguagem HTML é especializada na formatação da página e a criação de hiperlinks. Ela é totalmente desprovida de recursos para validação de conteúdo e comandos estruturais que permitam a execução repetitiva de trechos do programa. Se for criado um formulário para receber dados do usuário, a linguagem HTML simplesmente mostra o campo e aceita a digitação. Se o usuário digitar dados incompatíveis com o campo, não se tem como testar e tomar uma atitude. Uma solução para isto seria utilizar uma linguagem de script como a JavaScript que complementam a linguagem HTML.

A linguagem JavaScript foi desenvolvida pela Netscape e passou a ser suportada a partir da versão 2 do Netscape Navigator. Esta linguagem não deve ser confundida com a linguagem Java. O Java permite a criação de uma aplicação independente e possui todos os recursos de uma linguagem destinada a criação de aplicações comerciais, assim como a linguagem C. Já o JavaScript é uma linguagem de script que se aloja dentro de um programa HTML. Não se pode criar um programa JavaScript e executá-lo sem ter um browser. (RAMALHO 2005, p.191).

Embora existam variações de JavaScript que podem ser utilizadas em outros ambientes, como o Jscript da Microsoft, o seu uso mais popular é dentro de páginas HTML. Existe um modelo de objetos, chamado de DOM (*Document Object Model*), que nos permite acessar e modificar os elementos de uma página HTML, além de invocar componentes externos. (MELO JUNIOR 2007, p. 20).

De acordo com autor acima, com o JavaScript e o DOM pode-se fazer diversas coisas interessantes em uma página web, como por exemplo:

- Validar dados de um formulário antes de enviá-lo ao servidor;
- Criar textos e elementos novos em uma página;
- Responder à ações do usuário, como cliques de botão e outros eventos;

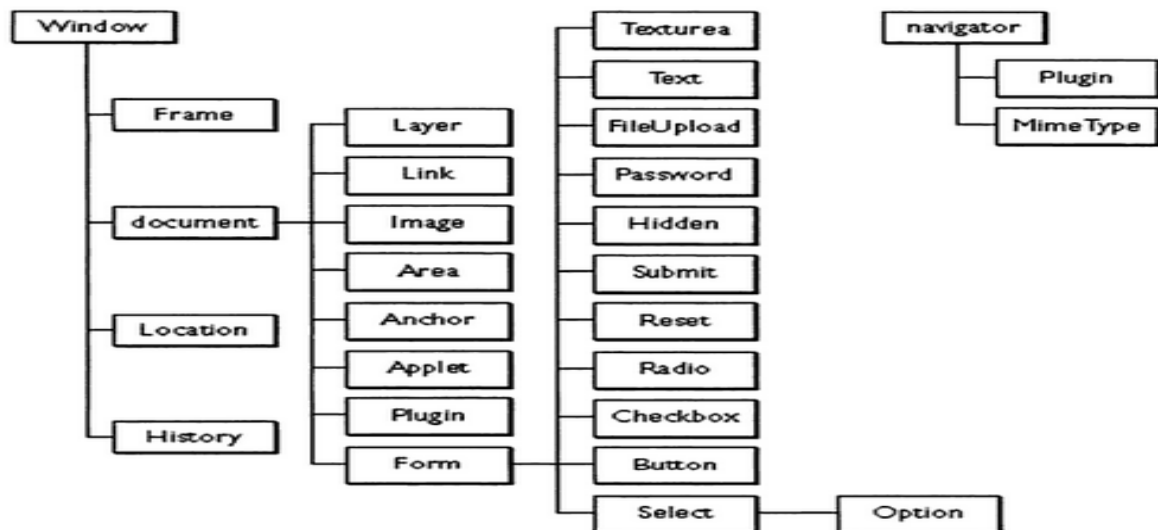
- Carregar informações do servidor utilizando Ajax;
- Exibir mapas do Google;
- Buscar e exibir notícias utilizando RSS.

JavaScript é uma linguagem interpretada, ou seja, o código fonte sempre é traduzido para uma linguagem de máquina que o computador entenda na hora em que for executado. É baseado em objetos, de forma que o programador pode utilizar-se de objetos predefinidos ou então criar novos objetos para satisfazer suas necessidades. JavaScript possui um núcleo com diversos objetos como *Array*, *Date*, *Math* e um conjunto de elementos de linguagem como operadores, estruturas de controle e funções. A linguagem pode ser dividida em duas, de acordo com sua aplicação. (RAMALHO 2005, p.192).

De acordo com José Antônio A. Ramalho (p.192, 2005), JavaScript é uma linguagem orientada a objetos, o que significa que ela trata todos os elementos de uma página web como um objeto. Objetos são agrupados de acordo com seu tipo ou finalidade. Javascript possui objetos que são automaticamente criados por ela e permitem criar novos objetos de acordo com a conveniência. Um objeto normalmente armazena uma série de informações que podem ser acessadas e utilizadas para processamento, ou alteradas pelo programador. Uma janela que exibe uma *home page* possui diversos objetos intrínsecos. Abaixo estão listados alguns deles:

- *navigator* – possui propriedades para o nome e versão do navegador que está sendo usado;
- *window* – é o objeto de mais alto nível. Possui propriedades que se aplicam à janela como um todo. Cada janela-filha possui também um objeto *window*. Em um documento composto por *frame* (divisão interna dentro de uma mesma janela do browser);
- *Document* – este objeto contém informações sobre a página ou documento HTML como um todo, incluindo dados sobre os elementos de formulários, links e âncoras, além de uma série de funções que permitem mudar as características da página;
- *Form* – este objeto guarda informações específicas sobre os formulários da página atual tais como o seu método, URL e dados sobre seus elementos e campos;
- *History* – este objeto mantém uma lista de todos os sites visitados na sessão de uso atual do browser;
- *Location* – este objeto possui informações sobre o local da página e informações relacionadas tais como o protocolo utilizado e seu domínio.

Figura 35: Hierarquia de objetos.



Fonte: RAMALHO 2005, p.194.

Javascript é uma das linguagens mais difundidas e utilizadas em todo o mundo. Utiliza-se esta linguagem para o Sistema proposto para validar campos obrigatórios, enviar os dados de formulários para os Servlets, atualizar páginas, entre outras funcionalidades que já foram descritas acima. Para o desenvolvimento de um aplicativo Web, a sua utilização é fundamental, por ser uma linguagem que a maioria dos navegadores populares suportam.

5.1.9 JQuery

Segundo Maurício Samy Silva (p.25, 2014), *JQuery* é uma biblioteca *JavaScript* criada por John Resig e disponibilizada como software livre e aberto. Pode-se utilizar esta biblioteca gratuitamente tanto para desenvolver projetos pessoais como comerciais. *JQuery* é uma maneira simples e fácil de escrever *JavaScript* colocada ao alcance não só de programadores experientes, mas também de designers e desenvolvedores com pouco conhecimento de programação.

Simplicidade é a palavra-chave que resume e norteia o desenvolvimento com jQuery. Linhas e mais linhas de programação JavaScript escritas para obter um simples efeito em um objeto são substituídas por apenas algumas, escrita com sintaxe jQuery. Intrincados e às vezes confusos códigos JavaScripts destinados a

selecionar um determinado elemento HTML, componente da árvore do documento, são substituídos por um simples método jQuery. (SAMY SILVA 2014, p.26).

jQuery destina-se a adicionar interatividade e dinamismo às páginas web, proporcionando ao desenvolvedor funcionalidades necessárias à criação de scripts que visem a incrementar, de forma progressiva e não obstrutiva, a usabilidade, a acessibilidade e o design, enriquecendo a experiência do usuário. (SAMY SILVA 2014, p.28).

O autor citado acima destaca algumas usabilidades de *jQuery*:

- Adicionar efeitos visuais e animações;
- Acessar e manipular o DOM;
- Buscar informações no servidor sem necessidade de recarregar a página;
- Prover interatividade;
- Alterar conteúdos;
- Modificar apresentação e estilização;
- Simplificar tarefas específicas de *JavaScript*;
- Realizar outras tarefas relacionadas às descritas.

jQuery foi criada com a preocupação de ser uma biblioteca em conformidade com os Padrões Web, ou seja, compatível com qualquer sistema operacional e navegador. *jQuery* é uma biblioteca *JavaScript* que possui as seguintes características: (SAMY SILVA 2014, p.29).

- Utiliza seletores CSS para localizar elementos componentes da estrutura de marcação HTML da página;
- Possui arquitetura compatível com instalação de plug-ins e extensões em geral;
- É indiferente às inconsistências de renderização entre navegadores;
- É capaz de interação implícita, isto é, não há necessidade de construção de loops para localização de elementos no documento;
- Admite programação encadeada, ou seja, cada método retorna um objeto.
- É extensível, pois admite criação e inserção de novas funcionalidades na biblioteca existente.

A sua utilização no desenvolvimento do Sistema proposto nesta Monografia deve-se principalmente, além dos detalhes definidos acima, ao seu uso simplificado e sem restrições com AJAX, tecnologia explicada a seguir. jQuery possui inúmeros tutoriais pela Internet com exemplos, facilitando seu entendimento e utilização.

5.1.10 Ajax

De acordo com W3SCHOOLS (2014), Ajax refere-se à *Asynchronous JavaScript and XML*, sendo uma técnica para criação de páginas web de forma rápida e dinâmica. Ajax permite que as páginas da web sejam atualizadas de forma assíncrona através da troca de pequenas quantidades de dados com o servidor nos bastidores. Isso significa que é possível atualizar partes de uma página web, sem recarregar a página inteira. Esta foi uma necessidade encontrada ao desenvolver o Sistema descrito nesta Monografia, um dos principais motivos pela utilização do Ajax, além de ser uma ferramenta que dispõe de muitas funcionalidades como é descrito a seguir.

Ajax é uma releitura das técnicas de script dinâmico com o uso de um objeto chamado XMLHttpRequest, capaz de enviar requisições http de dentro de scripts JavaScript. Seu uso por grandes provedores de soluções, como o Google, popularizou a técnica. (MELO JUNIOR, p.71, 2007). A figura a seguir apresenta as principais propriedades do objeto XMLHttpRequest.

Figura 36: Principais propriedades do XMLHttpRequest.

Propriedade	Descrição
onreadystatechange	Esta propriedade recebe um objeto do tipo "function()" (ou o nome de uma função) para ser chamada quando a propriedade "readyState" sofre alguma mudança.
responseText	Retorna a resposta da requisição como um string.
responseXML	Retorna a resposta da requisição como um objeto DOM document.
status	Status http da requisição. O valor 200 significa "ok".

Fonte: MELO JUNIOR, p.75, 2007.

Na figura 9 é ilustrado os possíveis valores que pode retornar a propriedade *readyState*.

Figura 37: Possíveis retornos da propriedade “readyState”.

Estado	Descrição
0	Não inicializado. O método “open” ainda não foi invocado.
1	O método “send” ainda não foi invocado.
3	Interativo. Algum resultado já está disponível, mas o documento ainda não foi totalmente carregado.
4	Completo. O documento terminou de ser carregado. Podemos testar o status.

Fonte: MELO JUNIOR, p.75, 2007.

Ajax possui vários métodos, os mais utilizados estão descritos na figura 10, a seguir.

Figura 38: Os métodos mais utilizados em Ajax.

Método	Descrição
abort	Cancela a requisição em andamento.
getAllResponseHeaders	Retorna um string contendo todos os Headers http da resposta.
getResponseHeader	Retorna o valor de um Header cujo nome foi passado como parâmetro.
open	Cria e inicializa uma requisição http. Seus parâmetros são: <ul style="list-style-type: none"> ⊖ Método http (“GET” ou “POST”); ⊖ URL; ⊖ Assíncrono ou não (true/false); ⊖ Usuário (se necessário); ⊖ Senha (se necessário).
send	Envia a requisição. Opcionalmente, podemos informar o corpo da requisição, que deve ser um objeto DOM document. No MS IE podemos simplesmente ignorar o parâmetro, mas no Firefox temos que passar null .

Fonte: MELO JUNIOR, p.75, 2007.

De acordo com Cleuton Sampaio de Melo Junior (p. 76, 2007), existem duas maneiras de utilizar *XMLHttpRequest*: síncrona e assíncrona. Na maneira síncrona o *browser* ficará parado aguardando o termino da requisição, o que contradiz todo o propósito do Ajax. Na maneira assíncrona, a requisição será enviada e, quando terminar, uma função no script será invocada (isto se chama “*call-back*”).

O autor citado acima ainda declara que para utilizar a maneira assíncrona, deve-se utilizar as seguintes propriedades e métodos:

- “*onreadystatechange*” deve receber um objeto do tipo “function()” ou o nome de uma função;
- “*readyState*” utilizado para verificar se o documento já terminou de ser carregado (*readyState* == 4);
- “*status*” utilizado para verificar se a requisição foi bem-sucedida (*status* == 200);
- “*open*” sempre informar o parâmetro booleano “true”.

Segundo Crane e Pascarello, 2006 (apud MARAFON, 2006), a tecnologia AJAX segue três princípios:

- Os browsers apenas acessam a aplicação:

Os browsers são apenas terminais que possuem acesso ao sistema, portanto, cada vez que o usuário necessita requisitar uma informação nova, deve solicitar tal informação ao servidor. Para diminuir esse tráfego de requisições, o conjunto de tecnologias AJAX transfere uma parte da lógica de aplicação para o browser do cliente.

- Servidores enviam dados:

A cada nova requisição o servidor deve enviar ao seu cliente uma resposta contendo, normalmente, um conjunto de dados solicitados.

- Interação com usuário pode ser contínua:

No desenvolvimento de sistemas sem a utilização de AJAX, a cada ação o usuário deveria esperar o processamento do sistema e a renderização da página para continuar a utilizar o programa.

A utilização do AJAX pelo sistema aqui proposto foi fundamental para atualizar “partes” da página, sem a necessidade de atualizá-la inteira, fato este que aumenta significativamente a velocidade do site e permite também que alguns dados desta página não se percam, enquanto outros são atualizados. O AJAX permite uma certa liberdade na escolha da linguagem de programação utilizada em concordância com ele, desta forma não houve dificuldades em conciliar o AJAX com Java para manipulação do Banco de Dados, bem como JQuery e JavaScript.

Uma característica extremamente importante e útil do *Ajax* é que as solicitações são assíncronas. Isso quer dizer que o *Javascript* de um aplicativo *Ajax* se comunica

assincronamente com o servidor, ou seja, apesar do código *Javascript* fazer uma solicitação ao servidor, o usuário da página pode continuar navegando na página clicando em botões ou até mesmo preenchendo um formulário. (MEDEIROS, 2015).

De acordo com Medeiros (2015), como vantagens de se usar *Ajax* temos o aumento de velocidade de um site, melhor uso por parte do usuário e flexibilidade para escolha da linguagem do lado servidor. Entre as desvantagens estão a utilização de frameworks que podem impossibilitar a utilização de outros frameworks, botões de avançar e voltar do navegador que não voltará ao conteúdo anterior, dificuldade na manutenção do site, pois ao utilizar muitos Ajax por toda a página pode acabar poluindo o código com diversas solicitações e tratamentos diferentes.

5.1.11 Oracle Database 11G

Segundo Bob Bryla e Kevin Loney (p.28, 2007), o banco de dados Oracle 11g é uma evolução da versão anterior, o Oracle 10g. O banco Oracle 11g continua a tradição de aprimoramento de recursos, tornando o gerenciamento de memória mais automatizado.

Um banco de dados é uma coleção de dados em um ou mais arquivos no disco de um servidor que coleta e mantém informações relacionadas. O banco Oracle 11g consiste em várias estruturas físicas e lógicas, sendo a tabela a sua estrutura lógica mais importante. Uma tabela é composta de linhas e colunas que contem dados relacionados. No mínimo um banco de dados deve conter ao menos tabelas para armazenar informações uteis. (BRYLA; LONEY, p.28, 2007).

Um banco de dados fornece níveis de segurança para impedir o acesso não autorizado aos dados. O Oracle Database 11g fornece muitos mecanismos para manter seguros os dados confidenciais. (BRYLA; LONEY, p.29, 2007).

Os arquivos que compõem um banco de dados se encaixam em duas categorias abrangentes: arquivos de banco de dados e arquivos não banco de dados. A distinção encontra-se no tipo de dado que está armazenado em cada um deles. O arquivo de banco de dados contém dados e metadados; os arquivos não banco de dados contém parâmetros de inicialização, informações de log e assim por diante. Os arquivos de banco de dados são

críticos para a operação contínua do banco de dados momento a momento. (BRYLA; LONEY, p.29, 2007).

Figura 39: Exemplo de tabela de banco de dados.

O diagrama mostra uma tabela com o cabeçalho 'TABLE: HR_EMPLOYEE' e o rótulo 'nome da tabela' apontando para ele. A tabela possui três colunas: 'EMPLOYEE_NUMBER', 'LAST_NAME' e 'FIRST_NAME', com o rótulo 'nomes das colunas' apontando para os cabeçalhos. O corpo da tabela contém cinco linhas de dados. O rótulo 'colunas linhas de dados' aponta para a primeira coluna, e o rótulo 'colunas' aponta para a base da tabela.

EMPLOYEE_NUMBER	LAST_NAME	FIRST_NAME
1	KRAUSE	JULIE
2	PYEATT	JOHN
3	TYLER	LIV
5	MUNN	OLIVIA

Fonte: BRYLA e LONEY, p.29, 2007.

De acordo com PROJEDATA (2014), Oracle é líder mundial em Bancos de Dados Corporativos, possuindo grande variedade de distribuições (Macintosh, Windows, Linux, FreeBSD, Unix). Segue o padrão SQL com uma linguagem própria para desenvolvimento de aplicações (PL/SQL). As principais vantagens do Banco de Dados Oracle podem ser vistas abaixo:

- *Escalabilidade.* O Oracle permite gerenciar desde pequenas bases de dados, até grandes massas de dados. O Banco Oracle gerencia os maiores bancos de dados existentes.
- *Performance.* Além de oferecer estratégias de manutenção de performance para bancos com grandes volumes de registros, conta com otimizadores que tanto podem dar autonomia para o desenvolvedor avançado decidir pela melhor estratégia de acesso, quanto decidem a melhor estratégia de acesso pela análise do comportamento real dos dados.
- *Confiabilidade.* O Oracle é um dos bancos de dados mais sólidos e robustos que existem, na medida que oferece total preservação dos dados, mesmo diante de falhas catastróficas no ambiente.
- *Recursos.* O Oracle, mesmo nas versões mais básicas, tem inúmeras funcionalidades que outros bancos só têm em versões mais avançadas.
- *Múltiplas plataformas.* Oracle roda em vários sistemas operacionais, de forma transparente para o desenvolvedor. De Windows até Linux, Unix, Mac OS, Z-Os, entre outros.

- *Suporte*. Por ser um dos bancos de dados mais usados no mundo, o Oracle conta uma expressiva base de profissionais, comunidades e grupos de discussão para dar suporte ao produto.

Os fatores que motivaram a escolha do Banco de Dados da aplicação proposta foram principalmente o fato da cultura da empresa Cliente já trabalhar com este Banco e estar habituada com sua manipulação, além de ter um alto índice de confiabilidade, o que é imprescindível quando se trabalha com dados de empresas (públicas ou privadas) e suas informações que devem permanecer confidenciais. Este Banco de Dados possui uma extensa variedade de funções que facilitam sua utilização e integração com a linguagem de programação Java.

Através da ferramenta PL/SQL é possível realizar as operações no Banco de Dados de forma mais ágil e simplificada. Outro fator de grande destaque é o suporte que o Banco oferece e seu tempo de existência no mercado.

A Oracle está no mercado a mais de 20 anos. Ela é responsável pelo primeiro SGBD a rodar em mainframes e minicomputadores em 1981. A Oracle fornece altos níveis de disponibilidade e proteção para os dados disponíveis no mercado atualmente. Como exemplos de recursos pode-se destacar: rápida recuperação de falhas no sistema, recuperação transparente de erro humano, proteção contra corrupção de dados e reduções drásticas na necessidade de *downtime* planejado. Tudo isto para garantir que o sistema esteja sempre disponível. (CORRÊA; MARCONDES, p.7, 2015).

5.1.12 Java Persistence API

De acordo com a ORACLE (2014), o Java Persistence API fornece um modelo de persistência POJO para mapeamento objeto-relacional. Ele também pode ser usado diretamente por aplicações web e clientes de aplicativos e até mesmo fora da plataforma Java EE, por exemplo, em aplicações Java SE.

O Sistema proposto utiliza JPA para o relacionamento entre a aplicação e o Banco de Dados Oracle. Uma ferramenta muito eficiente que disponibiliza fácil utilização para relacionar as Classes Java (Entidades) com as Tabelas do Banco. Cada tabela possui uma

entidade que a representa, seus campos são especificados como os atributos da classe, mapeando de forma eficaz e segura.

Segundo Erick Jendrock et. al. (2014), o Java Persistence API fornece uma facilidade de objeto/relacional mapeamento para desenvolvedores Java para o gerenciamento de dados relacionais em aplicativos Java. Ele consiste em três áreas: o Java Persistence API, a linguagem de consulta e objeto/metadados relacional.

Entre as principais inclusões na JPA pode-se destacar:

- **POJOS Persistentes:** Talvez o aspecto mais importante da JPA seja o fato que os objetos são POJOs (*Plain Old Java Object* ou Velho e Simples Objeto Java), significando que os objetos possuem design simples que não dependem da herança de interfaces ou classes de frameworks externos. Qualquer objeto com um construtor default pode ser feito persistente sem nenhuma alteração numa linha de código. Mapeamento Objeto-Relacional com JPA é inteiramente dirigido a metadados. Isto pode ser feito através de anotações no código ou através de um XML definido externamente. (MEDEIROS, 2014).

- **Consultas em Objetos:** As consultas podem ser realizadas através da Java Persistence Query Language (JPQL), uma linguagem de consulta que é derivada do EJB QL e transformada depois para SQL. As consultas usam um esquema abstraído que é baseado no modelo de entidade como oposto às colunas na qual a entidade é armazenada. (MEDEIROS, 2014).

- **Configurações simples:** Existe um grande número de características de persistência que a especificação oferece, todas são configuráveis através de anotações, XML ou uma combinação das duas. Anotações são simples de usar, convenientes para escrever e fácil de ler. Além disso, JPA oferece diversos valores defaults, portanto para já sair usando JPA é simples, bastando algumas anotações. (MEDEIROS, 2014).

- **Integração e Teste:** Atualmente as aplicações normalmente rodam em um Servidor de aplicação, sendo um padrão do mercado hoje. Testes em servidores de aplicação são um grande desafio e normalmente impraticáveis. Efetuar teste de unidade e teste caixa branca em servidores de aplicação não é uma tarefa tão trivial. Porém, isto é resolvido com uma API que trabalha fora do servidor de aplicação. Isto permite que a JPA possa ser utilizada sem a existência de um servidor de aplicação. Dessa forma, testes unitários podem ser executados mais facilmente. (MEDEIROS, 2014).

5.1.13 JBoss

O *JBoss* é um servidor de aplicações baseado em Java. Um servidor de aplicações é um software que provê um ambiente completo para que outras aplicações sejam executadas dentro dele usando uma gama de serviços provida pelo servidor de aplicações (4LINUX, 2015). O servidor utilizado para o software desenvolvido nesta monografia é o *JBoss* 7.1.

A grande vantagem de um servidor de aplicações é que os desenvolvedores podem se concentrar nas necessidades de negócio. Aspectos como conexões a bancos de dados, autenticação e gerenciamento de recursos são gerenciados pelo servidor de aplicações. Além disso, o padrão Java EE define padrões abertos que aceleram o desenvolvimento com uso de API padronizada e pensada para computação distribuída. (4LINUX, 2015).

O *JBoss* é líder neste segmento, oferecendo na sua versão da comunidade (*open source*) recursos avançados de *clustering*, segurança e integração. Esses recursos são ausentes em produtos concorrentes ou fornecidos apenas como complementos cobrados a parte. (4LINUX, 2015).

5.1.14 Bootstrap

Uma tendência em alta no mundo front-end é o uso de frameworks CSS com estilos base para nossas páginas. Ao invés de começar todo o projeto do zero, criando todo estilo na mão, existem frameworks que já trazem toda uma base construída de onde partiremos a aplicação. (CAELUM, p. 178, 2015).

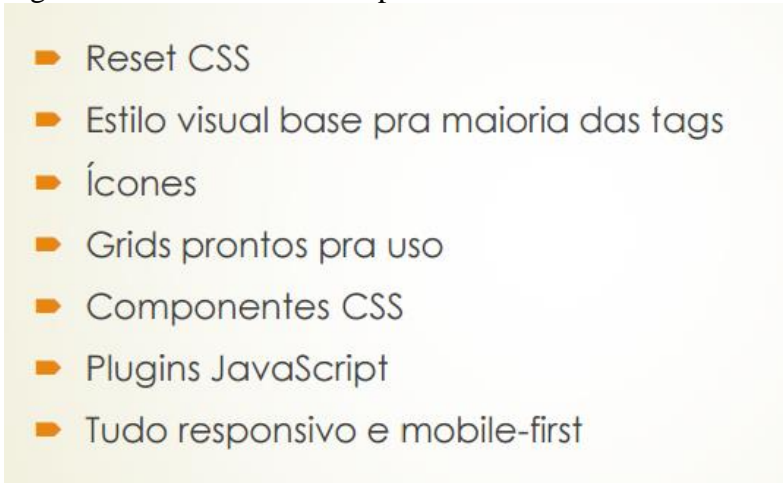
O Twitter Bootstrap é uma destas tecnologias que atualmente está sendo muito utilizada pelos desenvolvedores em busca de design responsivo de alta qualidade. Segundo Caelum (p.178, 2015) Twitter Bootstrap foi criado pelo Twitter, a partir de códigos que eles já utilizavam internamente. O projeto cresceu bastante em maturidade e importância no mercado a ponto de se desvincular do Twitter e se tornar apenas Bootstrap.

O termo web design responsivo refere-se a uma técnica de estruturação HTML e CSS, que visa adaptar uma página web aos diferentes dispositivos e resoluções onde é

exibida. A ideia que surge quando se fala em design responsivo, é de identificar o dispositivo que está requisitando a página e com base nessa informação, fornecer uma página que seja melhor exibida. (COSTA, 2015).

O *Bootstrap* é um framework *opensource* compatível com HTML5 e CSS3 que foi criado para auxiliar no desenvolvimento de web sites responsivos. Alguns recursos disponíveis são mostrados na figura abaixo. (COSTA, 2015).

Figura 40: Recursos Bootstrap.

- 
- A list of Bootstrap resources presented in a light yellow rectangular box. Each item is preceded by a small orange square icon. The items are: Reset CSS, Estilo visual base pra maioria das tags, Ícones, Grids prontos pra uso, Componentes CSS, Plugins JavaScript, and Tudo responsivo e mobile-first.
- Reset CSS
 - Estilo visual base pra maioria das tags
 - Ícones
 - Grids prontos pra uso
 - Componentes CSS
 - Plugins JavaScript
 - Tudo responsivo e mobile-first

Fonte: COSTA, 2015.

A utilização de Bootstrap para o desenvolvimento do sistema proposto nesta monografia foi de fundamental importância facilitando a criação das telas de forma responsiva para posteriormente ser utilizada também em Tablets. Algumas vantagens em sua utilização são destacadas por Costa (2015):

- Documentação simples e ampla.
- Feito para trabalhar com layouts responsivos.
- Inúmeros componentes a disposição.
- Mantém padrões.
- Funciona em todos os navegadores atuais (Chrome, Safari, Firefox, IE, Opera).

5.2 HISTÓRICO E AVALIAÇÃO DO DESENVOLVIMENTO

A dificuldade e demora em atualizar os formulários dinâmicos utilizados como estrutura para o sistema de Fiscalização da área da saúde fez com que surgisse a ideia de criação de um sistema que facilitasse essa atividade, assim surgiu o projeto pelo qual se desenvolveu o sistema apresentado nesta monografia.

Inicialmente foi apresentada a ideia de criação deste sistema ao cliente principal, o Conselho Federal de Medicina, o qual verificou a necessidade e aceitou que o mesmo fosse desenvolvido. Posteriormente foram levantados todos os requisitos funcionais e não funcionais necessários para atender a várias necessidades quanto à atualização destes formulários eletrônicos.

Após colher os requisitos iniciou a etapa de criação dos protótipos das telas a serem implementadas para o sistema, atendendo de forma clara e responsiva as necessidades do usuário. Estabelecidos os modelos das telas, iniciou-se a criação das tabelas do banco de dados a serem utilizadas pelo sistema e a implementação do sistema por parte da autora desta monografia. Algumas ferramentas eram pouco conhecidas pelo desenvolvedor, permitindo através da implementação deste sistema um conhecimento mais aprofundado nas tecnologias JQuery, Ajax e Oracle que são ferramentas bem complexas com muitas funcionalidades disponíveis.

Concluída a etapa de desenvolvimento, os testes foram sendo realizados pela equipe de tecnologia da informação da qual a programadora faz parte. Foram realizados testes por usuários que nunca conheceram nada sobre a ferramenta, nem sobre o sistema de fiscalização, para avaliar a facilidade de sua utilização por pessoas leigas. Também foram realizados testes por usuários que atuam atualizando os formulários dinâmicos manualmente a fim de avaliar a real diferença desta atividade com e sem o sistema.

Atualmente o sistema é utilizado pela empresa de tecnologia, encontrando-se em fase de testes para posteriormente ser implantado no cliente, fato este que deverá ser realizado o mais rápido possível devido à demanda frequente de atualizações.

Todos os usuários que até o presente momento utilizaram o sistema avaliaram este como sendo fácil de utilizar, atendendo às demandas que necessitavam para o trabalho a ser realizado, cumprindo com os requisitos impostos inicialmente no projeto.

5.3 APRESENTAÇÃO DO SISTEMA

A seguir são apresentadas algumas telas do sistema implementado, bem como uma breve descrição de suas funcionalidades.

A figura abaixo mostra a tela de *Login* do sistema, é o primeiro contato do usuário com o sistema, onde este deve informar o nome de usuário e a senha ao qual é registrado para ter acesso às funcionalidades deste sistema.

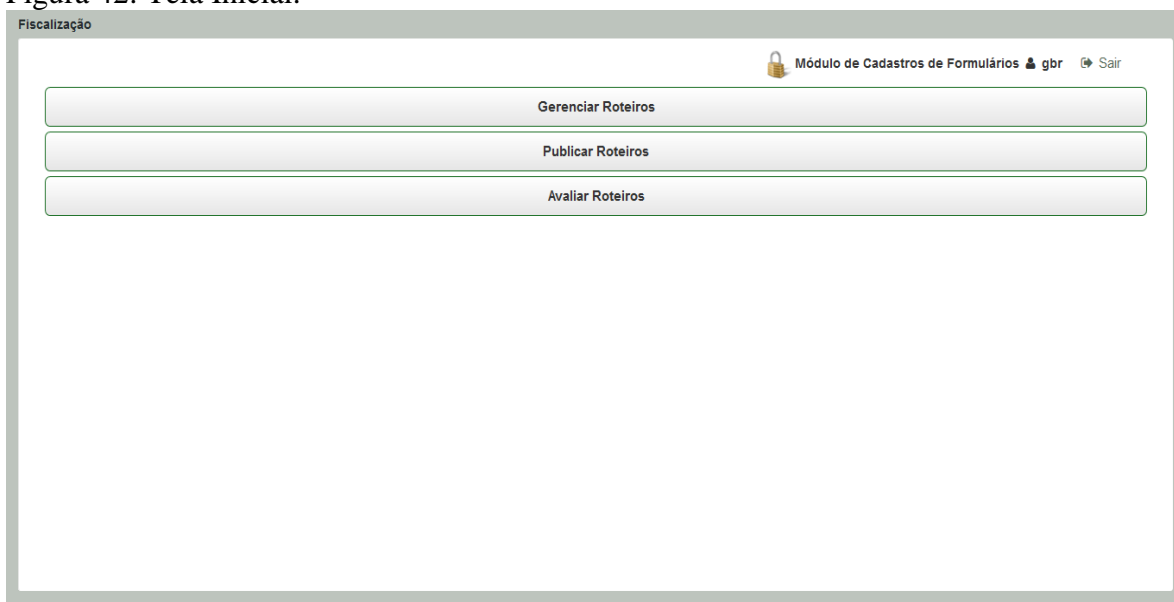
Figura 41: Tela *Login*.

A imagem mostra a tela de login do sistema. O layout é dividido em duas partes principais. À esquerda, sobre um fundo branco, está o logotipo do CFM (Conselho Federal de Medicina) com o símbolo de um caduceu verde. Abaixo dele, há o texto 'Módulo de Cadastros de Formulários' e 'CR VIRTUAL'. À direita, há um formulário de login com dois campos de entrada: 'Usuário' e 'Senha'. Abaixo desses campos, há um botão 'Acessar'. O fundo geral da tela é verde escuro com formas abstratas.

Fonte: Autora da Monografia, 2015.

Abaixo é apresentada a tela inicial contendo as opções do sistema separadas de acordo com as funcionalidades mais generalizadas. A primeira opção (Gerenciar Roteiros) refere-se à manutenção dos formulários dinâmicos, a segunda opção (Publicar Roteiros) refere-se à exportar os formulários dinâmicos para a máquina local e a terceira e menos comentada nesta monografia são avaliações destes formulários.

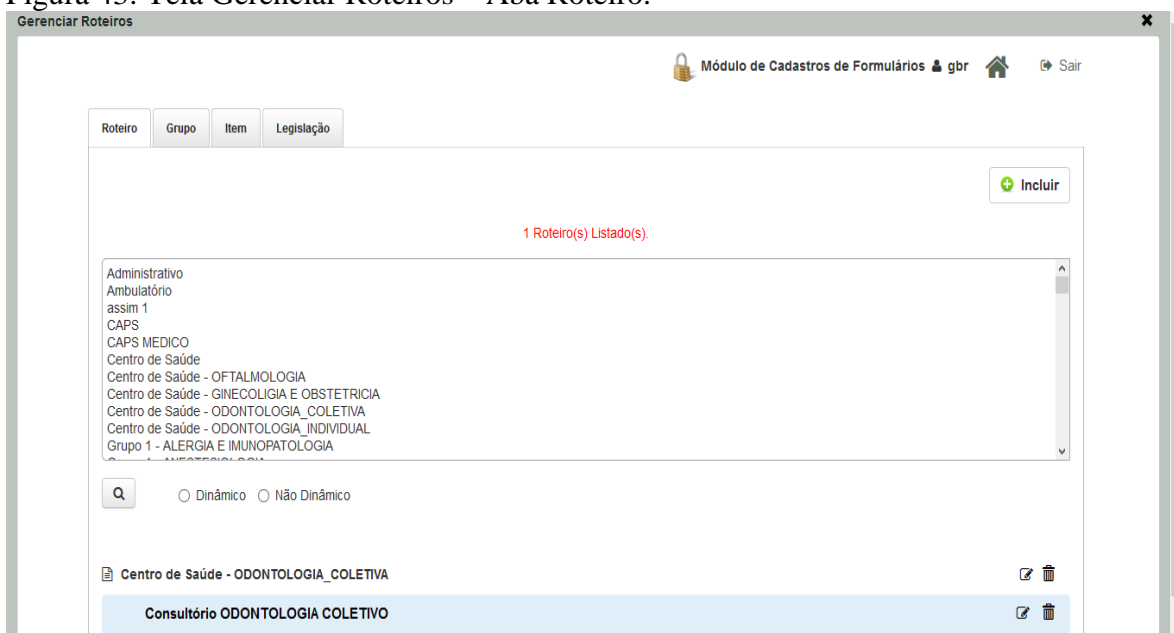
Figura 42: Tela Inicial.



Fonte: Autora da Monografia, 2015.

A figura a seguir apresenta a tela inicial de Gerenciar Roteiros, onde é acessada primeiramente a aba de Roteiros (Formulários), sendo a mais utilizada em manutenções. As demais abas complementam os formulários atuando de forma igualmente significativa.

Figura 43: Tela Gerenciar Roteiros – Aba Roteiro.

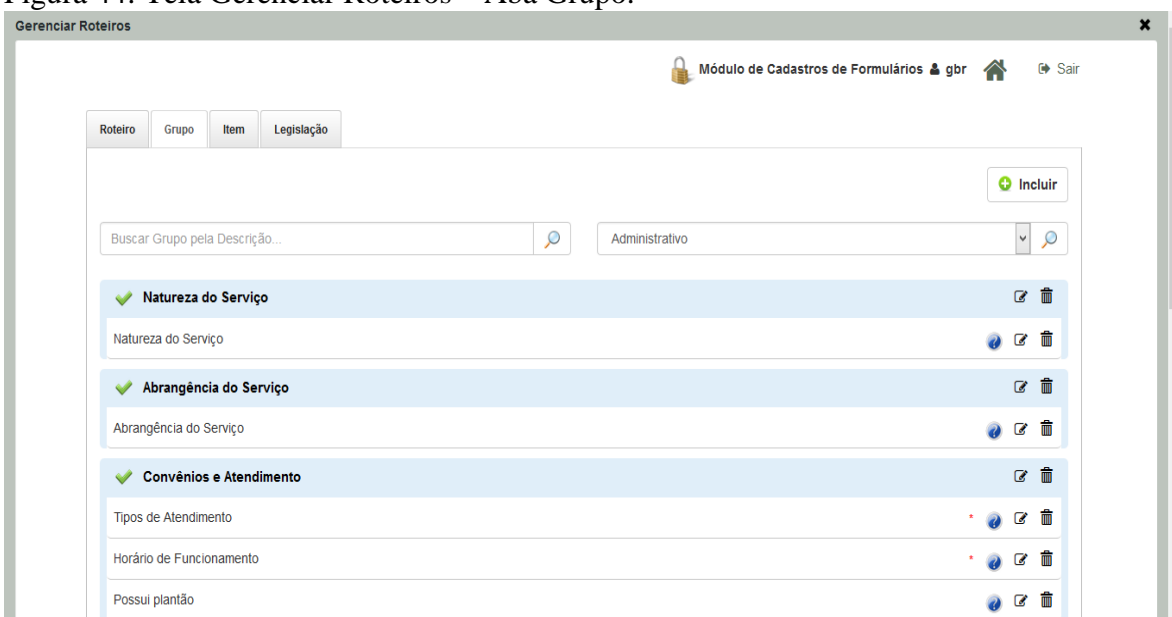


Fonte: Autora da Monografia, 2015.

As demais abas possuem o mesmo layout da aba de Roteiro, modificando apenas com a adição de um filtro a mais para pesquisa, como é mostrado na figura abaixo na tela de

Aba Grupo, que possui um filtro para selecionar os grupos de determinado Formulário escolhido.

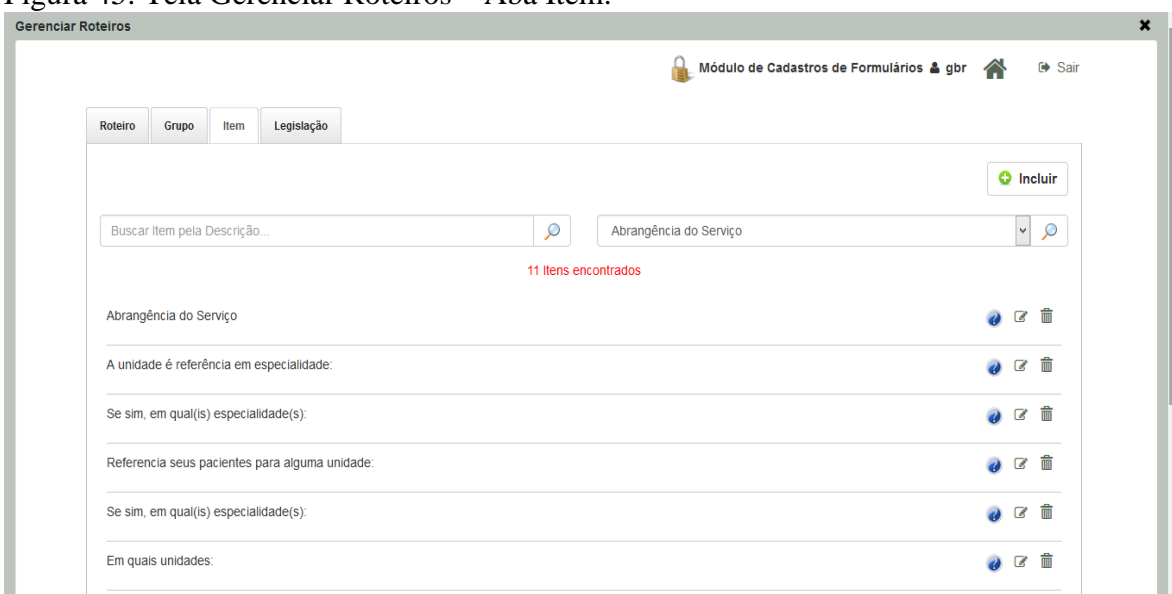
Figura 44: Tela Gerenciar Roteiros – Aba Grupo.



Fonte: Autora da Monografia, 2015.

A figura abaixo apresenta a tela da aba Item.

Figura 45: Tela Gerenciar Roteiros – Aba Item.



Fonte: Autora da Monografia, 2015.

Abaixo é mostrada a tela de aba Legislação, com algumas legislações listadas.

Figura 46: Tela Gerenciar Roteiros – Aba Legislação.

The screenshot displays the 'Gerenciar Roteiros' application window. At the top, there's a header bar with the title 'Gerenciar Roteiros' and a close button. Below the header, a navigation bar contains tabs for 'Roteiro', 'Grupo', 'Item', and 'Legislação', with 'Legislação' being the active tab. The main content area features a search bar labeled 'Buscar Legislação pela Descrição...' with a magnifying glass icon. To the right of the search bar are buttons for 'Todos' (with a list icon) and 'Incluir' (with a plus icon). Below these, a list of legislative items is shown, each with a text description and two icons (edit and delete) on the right. The items listed are: 'ABNT NBR Nº 9050/2004', 'Lei Nº 6839/1980, art. 1º (04/08/2014 - 28/08/2014)', 'No caso de sedação profunda ou anestesia, são necessários dois médicos: uma para o procedimento endoscópico e o outro para sedação ou anestesia', 'INSTRUÇÕES: teste - Os serviços classificados como tipo III devem possuir uma maca adicional por sala de recuperação, quais sejam, aqueles com sedação que requeiram um segundo médico', 'Portaria Nº 2226/2009 do MS (12/08/2014 - 28/08/2014)', 'Portaria da atenção básica do MS', and 'Portaria GM Nº 2095/2013'.

Legislação	Ações
ABNT NBR Nº 9050/2004	[Edit] [Delete]
Lei Nº 6839/1980, art. 1º (04/08/2014 - 28/08/2014)	[Lock] [Edit] [Delete]
No caso de sedação profunda ou anestesia, são necessários dois médicos: uma para o procedimento endoscópico e o outro para sedação ou anestesia	[Edit] [Delete]
INSTRUÇÕES: teste - Os serviços classificados como tipo III devem possuir uma maca adicional por sala de recuperação, quais sejam, aqueles com sedação que requeiram um segundo médico	[Edit] [Delete]
Portaria Nº 2226/2009 do MS (12/08/2014 - 28/08/2014)	[Edit] [Delete]
Portaria da atenção básica do MS	[Edit] [Delete]
Portaria GM Nº 2095/2013	[Edit] [Delete]

Fonte: Autora da Monografia, 2015.

O cadastro e edição possuem o mesmo layout, divergindo apenas que no cadastro os campos se apresentam vazios e na edição os campos já possuem os dados de determinado item (Formulário, Grupo, Item ou Legislação) que se deseja alterar. Abaixo é apresentado a tela de cadastro de Formulário, em duas partes devido à sua dimensão.

Figura 47: Tela Cadastro de Formulário (parte 1).

Fonte: Autora da Monografia, 2015.

Figura 48: Tela Cadastro de Formulário (parte 2).

Fonte: Autora da Monografia, 2015.

A figura abaixo mostra a tela de cadastro de Legislação que é semelhante à edição de Legislação diferenciando apenas por estarem dispostos os dados da legislação na edição.

Figura 49: Tela Cadastro de Legislação.

Cadastrar Legislação

Descrição:
Digite a descrição da legislação

Instruções:
Digite as instruções

Número:
Digite o número da legislação

Ano:
Digite o ano da legislação

Artigo:
Digite o Artigo da legislação

Alínea:
Digite Alínea da legislação

Tipo:
Digite o tipo da legislação

Data Inicial (Vigência)
//____

Link:
Digite o link da legislação

« Maio 2015 »

Do	Se	Te	Qu	Qu	Se	Sa
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Limpar

Salvar

Fonte: Autora da Monografia, 2015.

A figura abaixo apresenta a tela de publicação de Formulários, onde estes são exportados em formato XML para a máquina local do usuário.

Figura 50: Tela Publicar Roteiros.

Publicar Roteiros

Módulo de Cadastros de Formulários gbr Sair

Lista de Roteiros:
CAPS

Nome do arquivo:
xmlCAPS

Publicar

Publicação realizada com sucesso na pasta: C:\XMLS\xmlCAPS.xml

Fonte: Autora da Monografia, 2015.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo são apresentadas as conclusões, que abrange toda a monografia, e posteriormente indicações para realizações de trabalhos futuros.

6.1 CONCLUSÕES

Esta monografia iniciou apresentando a problemática, justificativa e objetivos do trabalho de conclusão de curso. Assim, apresentou toda a especificação do sistema desenvolvido, qual a sua utilidade, quem a utiliza e o objetivo de sua criação. É descrito detalhadamente todo o processo que origina os formulários dinâmicos e percorrido todo o caminho até este ser utilizado pelo sistema de fiscalização.

O sistema de fiscalização o qual utiliza-se do principal componente do sistema aqui desenvolvido, que são os formulários eletrônicos, é fundamental para a organização das entidades de saúde. Esse sistema de fiscalização, controla todas as entidades de saúde fornecendo maior segurança para toda a população Brasileira, gerando portanto, um impacto nacional a agilidade com que as informações são renovadas neste sistema.

Tais informações são dispostas em forma de questionários nos formulários dinâmicos que são manipulados pelo sistema aqui proposto. Concluindo desta forma que o sistema permitiu acelerar as atualizações de normas e legislações para os sistema de fiscalização. Os fiscais podem prestar seus serviços, avaliando as entidades de saúde sempre com os questionários atualizados segundo as normas recebidas do próprio Conselho Federal de Medicina. Cabe destacar, que o CFM modifica suas Resoluções, onde se encontram essas normas, com muita frequência.

O estudo das ferramentas a serem utilizadas foi realizado durante todo o desenvolvimento do projeto, sendo escolhidas as que fornecem melhores resultados de acordo com o entendimento da autora, as disposições estabelecidas pelo cliente e as qualidades esperadas por seus usuários, com boa usabilidade, aparência e responsividade.

A monografia apresentou uma pequena descrição das ferramentas e tecnologias utilizadas para o desenvolvimento do sistema, bem como o porquê de cada utilização. Todo o desenvolvimento do sistema forneceu maior conhecimento para a autora sobre estas tecnologias, algumas já estudadas durante a graduação e outras mais atuais estudadas exclusivamente para este desenvolvimento.

A informatização está cada dia progredindo mais, e os trabalhos que eram feitos manualmente tornam-se informatizados, o que aumenta a facilidade e agilidade em sua realização.

O sistema proposto nesta monografia conseguiu reduzir o tempo de realização de um trabalho anteriormente feito manualmente em mais de cinquenta por cento, sendo conseqüentemente diminuído também o custo pago por este trabalho.

6.2 TRABALHOS FUTUROS

Com a finalização do desenvolvimento deste sistema foram surgindo novas ideias para acrescentar às funcionalidades já existentes, melhorias e novos módulos que poderá fazer o sistema expandir em número e qualidade, sempre buscando melhorar a vida do cliente e dos usuários, diretos e indiretos.

Uma necessidade reconhecida desde o início do desenvolvimento, porém ainda não implementada, devido ao software de fiscalização não possuir ainda suporte para tal funcionalidade, é a integração entre o sistema aqui proposto nesta monografia e o sistema de fiscalização, de tal forma que os formulários dinâmicos sejam passados de um para outro sem a necessidade de salvar em máquina local.

Outra necessidade definida e que já esta sendo implementada é a utilização deste sistema em *tablets*. A utilização da ferramenta Bootstrap nos permite obter sistemas responsivos e que se enquadrem em várias resoluções de tela. As telas estão sendo alteradas com esta nova tecnologia, este processo não está totalmente finalizado e o sistema não está completamente atualizado com esta tecnologia, mas está em construção e em breve período de tempo poderá ser utilizado por *tablets* e *smartphones*.

Um módulo para disposição das entidades de saúde se auto avaliarem de forma online está sendo estudado para implementação. O sistema já possui um módulo de auto avaliação de entidades, porém interno e com objetivo de testar os formulários manipulados, tem-se a necessidade de exteriorizar este módulo para disposição de qualquer entidade de saúde que deseje verificar suas condições e precaver de irregularidades.

Desta forma conclui-se este trabalho com a satisfação de haver podido contribuir nos processos de apoio ao sistema de saúde brasileiro.

REFERÊNCIAS

4LINUX, Open Software Specialists. **O que é JBoss**. Disponível em: <http://www.4linux.com.br/o-que-e-jboss>. Acesso em: 29 jun. 2015.

ALBINADER NETO, Jorge Abílio; LINS, Rafael Dueire. **Web Services em Java**. Rio de Janeiro: Brasport, 2006.

AMARAL, Luiz Alfredo Martins. PRAXIS, Um Referencial para o Planejamento de Sistemas de Informação. Universidade do Minho, 1994. Disponível em <http://hdl.handle.net/1822/49>. Acesso em 17 fev. 2015.

AMATO SOFTWARE. **Risco Cirúrgico MED**. Disponível em: <http://www.riscocirurgico.com.br/risco-cirurgico-pre-operatorio>. Acesso em: 21 out. 2014.

AUDY, Jorge Luis Nicolas; ANDRADE, Gilberto Keller de; CIDRAL, Alexandre. **Fundamentos de Sistemas de Informação**. São Paulo: Artmed Editora S.A., 2005.

BARRA, Daniela Couto Carvalho; NASCIMENTO, Eliane Regina Pereira do; MARTINS, Josiane de Jesus; ALBUQUERQUE, Gelson Luiz; ERDMANN, Alacoque Lorenzini. Evolução Histórica e Impacto da Tecnologia na Área da Saúde e da Enfermagem. **Revista Eletrônica de Enfermagem**, v. 08, n. 03, p. 422-430, 2006. Disponível em: <http://h200137217135.ufg.br/index.php/fen/article/view/7081/5012> . Acesso em: 03 fev. 2015.

BRYLA, Bob; LONEY, Kevin. **Oracle Database 11G: Manual do DBA**. São Paulo: ARTMED Editora S.A., 2007.

CAELUM, Ensino e Inovação. **Desenvolvimento Web com HTML, CSS e Javascript**. Disponível em: <https://books.google.com.br/books?id=SwPIAgAAQBAJ&pg=PT187&dq=bootstrap+design+responsivo&hl=pt-BR&sa=X&ei=e75YVeHsNomkgwTU04DYBg&ved=0CC8Q6AEwAQ#v=onepage&q=bootstrap&f=false>. Acesso em: 17 mai. 2015.

CARINGI, Augusto Mecking. **Curso de Java – Semana Acadêmica**. 2002.

CARVALHO, André de Oliveira; EDUARDO, Maria Bernadete de Paula. **Sistemas de Informação em Saúde para Municípios**. São Paulo, 1998. Ed. Fundação Peirópolis Ltda. Disponível em: <http://www.bvs-sp.fsp.usp.br/tecom/docs/1998/car001.pdf>. Acesso em: 20 set. 2014.

CFM, Conselho Federal de Medicina. **A instituição**. Disponível em: http://portal.cfm.org.br/index.php?option=com_content&view=article&id=20671&Itemid=23. Acesso em 04 set. 2014.

CFM, Conselho Federal de Medicina. **RESOLUÇÃO CFM Nº 2.073/2014**. Disponível em: http://www.portalmedico.org.br/resolucoes/CFM/2014/2073_2014.pdf. Acesso em 28 mar. 2015.

___. **CFM cobra transparência nas listas de espera do SUS e anuncia reforço nas fiscalizações**. Disponível em: http://portal.cfm.org.br/index.php?option=com_content&view=article&id=20671&Itemid=23. Acesso em 10 abr. 2015.

___. **Fiscalização: CFM estabelece regras mínimas para serviços de assistência médica**. Disponível em: http://portal.cfm.org.br/index.php?option=com_content&view=article&id=20671&Itemid=23. Acesso em 10 abr. 2015.

CORREIA, Juliano Soares; MARCONDES, Marcos Roberto. **Vantagens e Desvantagens da Utilização do Banco de Dados Oracle**. Disponível em: http://www.esfcex.ensino.eb.br/revista/producaocientifica/arquivo/218_Artigo.pdf. Acesso em: 21 abr. 2015.

COSTA, Fernando Freitas. **Design Responsivo para Web com Bootstrap**. Disponível em: <http://blog.fimes.edu.br/fernando/files/2014/05/Design-responsivo-para-WEB-com-Bootstrap.pdf>. Acesso em: 17 mai. 2015.

DOURADO, Lorena Souza. **Introdução ao desenvolvimento de aplicações Web**. Disponível em: <http://www.devmedia.com.br/introducao-ao-desenvolvimento-de-aplicacoes-web/29798>. Acesso em: 19 abr. 2014.

EMEASOFT. **EMeAsOft Sistemas Inteligentes**. Disponível em: http://www.emeasoft.com.br/site/index.php?option=com_content&view=article&id=10&Itemid=8. Acesso em: 21 out. 2014.

FERREIRA, Ayrton Sérgio Rochedo. **Modelagem Organizacional por Processos: um sistema óbvio de gestão, um passo além da hierarquia**. Rio de Janeiro: Mauad, 2010. Disponível em: https://books.google.com.br/books?id=4zPvBwAAQBAJ&pg=PT93&dq=Modelagem+Processo+de+N%C3%A9g%C3%B3cios&hl=pt-BR&sa=X&ei=AEUwVfPKLIu2sAXg_YGACw&ved=0CDEQ6AEwAg#v=onepage&q=Modelagem%20Processo%20de%20N%C3%A9g%C3%B3cios&f=false. Acesso em: 16 abr. 2015.

FERREIRA, Sibele Maria Gonçalves. **Sistemas de Informação em Saúde**. 1999. Disponível em: <https://www.nescon.medicina.ufmg.br/biblioteca/imagem/2249.pdf>. Acesso em: 21 set. 2014.

FIELDS, Duane K.; KOLB, Mark A. **Desenvolvendo na Web com Java Server Pages**. Rio de Janeiro: Editora Ciência Moderna Ltda, 2000.

JENDROCK, Erick; BALL, Jennifer; CARSON, Debbie; EVANS, Ian; FORDIN, Scott; HAASE, Kim. **The Java EE 5 Tutorial**. Disponível em:

<http://docs.oracle.com/javaee/5/tutorial/doc/?wp406143&PersistenceIntro.html#wp78460>. Acesso em: 19 out. 2014.

JOBSTRAIBIZER, Flávia. **Criação de sites com CSS**. São Paulo: Digerati Books, 2009.

JUNCKES FILHO, Ian. **Business Intelligence em um Formulário Dinâmico**. Monografia (Sistemas de Informação) – Universidade do Sul de Santa Catarina, Palhoça, 2011.

LEONEL, Vilson; HEERDT, Mauri Luiz. **Metodologia Científica e da Pesquisa**. Palhoça: UnisulVirtual, 5ª edição, 2007.

LARMAN, Craig. **Utilizando UML e Padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo**. Porto Alegre: Artmed, 2005.

Disponível em:

<https://books.google.com.br/books?id=hzi2tmT8QkUC&pg=PA160&dq=modelo+de+dom%C3%ADnio&hl=pt-BR&sa=X&ei=AeORVf3AM8uq-AHB4azYDw&ved=0CCsQ6AEwAA#v=onepage&q=modelo%20de%20dom%C3%ADnio&f=false>. Acesso em: 29 jun. 2015.

LOBO, Edson J. R. **Curso de Engenharia de Software: métodos e processos para garantir a qualidade no desenvolvimento de softwares**. São Paulo: Digerati Books, 2008. Disponível em:

<https://books.google.com.br/books?id=ZJznA9UrtVAC&pg=PA56&dq=casos+de+uso+engenharia&hl=pt-BR&sa=X&ei=XVUwVZOxHMi5sAXGjID4Cg&ved=0CEQQ6AEwBw#v=onepage&q=casos%20de%20uso%20engenharia&f=false>. Acesso em: 16 abr. 2015.

MARAFON, Diego Luiz. **Integração JavaServer Faces e Ajax: estudo da integração entre as tecnologias JSF e AJAX**. Florianópolis: 2006.

MEDEIROS, Igor. **Introdução à JPA – Java Persistence API**. Disponível em:

<http://www.devmedia.com.br/introducao-a-jpa-java-persistence-api/28173>. Acesso em: 19 out. 2014.

—. **Ajax Básico: Introdução**. Disponível em:

<http://www.linhadecodigo.com.br/artigo/3585/ajax-basico-introducao.aspx>. Acesso em: 20 abr. 2015.

MELO, Ana Cristina. **Desenvolvendo Aplicações com UML 2.2: do conceitual à implementação**. 3ª edição. Rio de Janeiro: Brasport, 2010. Disponível em:

<https://books.google.com.br/books?id=BPVHsG17bAYC&pg=PA72&dq=Prot%C3%B3tipos+de+tela&hl=pt-BR&sa=X&ei=d4NDVbuWF4yWNt7RgKgC&ved=0CE0Q6AEwBQ#v=onepage&q=Prot%C3%B3tipos%20de%20tela&f=false>. Acesso em: 01 mai 2015.

MELO JUNIOR, Cleuton Sampaio de. **Web 2.0 e Mashups: reinventando a internet**. Rio de Janeiro: Brasport, 2007.

MUKHI, Vijay; MUKHI, Sonal; KOTECHA, Neha. **Java Servlets JSP**. São Paulo: Makron Books, 2002.

MÜLBERT, Ana Luiza; AYRES, Nilce Miranda. **Fundamentos para Sistemas de Informação**. 2005.

MORAES, Lissandro Schuelter de. **Black Bean: Utilizando Ajax em uma Aplicação Web**. 83 f. Monografia (Ciência da Computação) – Universidade do Sul de Santa Catarina, Palhoça, 2009.

MOREIRA NETO, Oziel. **Entendendo e dominando o Java para a Internet**. São Paulo: Digerati Books, 2006.

NEVES DA SILVA, Cristiano. **Conhecendo as diretivas JSP**. Disponível em: <http://www.devmedia.com.br/conhecendo-as-diretivas-jsp/5397>. Acesso em: 18 out. 2014.

ORACLE. **Java Persistence API**. Disponível em: <http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html>. Acesso em: 19 out. 2014.

PÁDUA, Elisabete Matallo Marchesini de. **Metodologia da pesquisa: Abordagem teórico-prática**. São Paulo: Papirus, 2004.

PEIXOTO, Mario. **GRC Dicionário Especial**. Disponível em: <https://books.google.com.br/books?id=omVMBQAAQBAJ&pg=PT36&dq=o+que+%C3%A9+regras+de+neg%C3%B3cios+engenharia&hl=pt-BR&sa=X&ei=UhhpVZr6E67IsQT564GICA&ved=0CDcQ6AEwBQ#v=onepage&q&f=false>. Acesso em: 29 mai, 2015.

PEREIRA, Samaris Ramiro; PAIVA, Paulo Bandiera. A importância da Engenharia da Usabilidade para a Segurança de Sistemas Informatizados em Saúde. **J. Health Inform**, pag. 123-129, jul-set, 2011.

PEREZ, Gilberto; ZWICKER, Ronaldo. Fatores determinantes da adoção de Sistemas de Informação na área da saúde: um estudo sobre o prontuário médico eletrônico. **RAM – Revista de Administração Mackenzie**, v. 11, n. 1. São Paulo, 2010. Disponível em: <http://www.scielo.br/pdf/ram/v11n1/08.pdf>. Acesso em: 20 set. 2014.

PRESSMAN, Roger S. **Engenharia de Software – Uma Abordagem Profissional**. 7ª edição. Porto Alegre: Artmed, 2011. Disponível em: https://books.google.com.br/books?id=y0rH9wuXe68C&pg=PA138&dq=atores+de+um+software&hl=pt-BR&sa=X&ei=jJP_VKeVEMGYgwSXh4GYCw&ved=0CB0Q6AEwAA#v=onepage&q=atores%20de%20um%20software&f=false. Acesso em: 10 mar. 2015.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. **Metodologia do Trabalho Científico: métodos e técnicas da pesquisa e do trabalho acadêmico**. 2ª edição. Nova Hamburgo: Feevale, 2013.

PROJEDATA, Software para Gestão Integrada. **Tecnologia: Banco de Dados Oracle**. Disponível em: <http://www.projedata.com.br/tecnologia/banco-de-dados-oracle>. Acesso em: 19 out. 2014.

RAMALHO, José Antônio A. **Curso Completo para Desenvolvedores Web**. Rio de Janeiro: Elsevier, 2005. Disponível em: <https://books.google.com.br/books?id=p4tFp-b92H8C&printsec=frontcover&hl=pt-BR#v=onepage&q&f=false>. Acesso em: 09 mar 2015.

RAMARKRISHNAN, Johannes R. **Sistemas de Gerenciamento de Banco de Dados**

RAMPAZZO, Lino. **Metodologia Científica**. São Paulo: Loyola, 2002.

REZENDE, Denis Alcides. **A evolução da tecnologia da informação nos últimos 45 anos**. 2011.

RIBEIRO, Leandro. O que é UML e Diagrama de Casos de Uso: Introdução prática à UML. Disponível em: <http://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>. Acesso em: 01 mai 2015.

ROMAN, Ed; AMBLER, Scott W.; JEWELL, Tyler. **Dominando Enterprise JavaBean**. Tradução: Edson Furmankiewicz. 4ª edição. Porto Alegre: Bookman, 2004.

SAMPAIO, Cleuton. **Java Enterprise Edition 6: desenvolvendo aplicações corporativas**. Rio de Janeiro: Brasport, 2011.

SANCHEZ, Otávio Henrique Vieira. **Apresentando EJB Entity Bean – CMP**. Disponível em: <http://www.devmedia.com.br/apresentando-ejb-entity-bean-cmp/7049>. Acesso em: 16 mar. 2015.

SANTOS, Vanice dos; CANDELORO, Rosana J. **Trabalhos acadêmicos: uma orientação para pesquisa e normas técnicas**. Porto Alegre: AGE, 2006.

SBIS, Sociedade Brasileira de Informática em Saúde. **O que é Informática em Saúde**. Disponível em: <http://www.sbis.org.br/>. Acesso em: 04 set. 2014.

SILVA, George; SILVA, Gilbert; GUIMARÃES, Gabriel; MEDEIROS, Rodrigo; ROSSINI, Tiago. **Utilizando ICONIX no desenvolvimento de aplicações Delphi**. II Congresso de Pesquisa e Inovação da Rede Norte Nordeste de Educação Tecnológica. João Pessoa, 2007.

SILVA, Maurício Samy. **Construindo Sites com CSS e (X)HTML: sites controlados por folhas de estilo em cascata**. São Paulo: Novatec Editora, 2008.

SILVA, Maurício Samy. **JQuery: a Biblioteca do Programador JavaScript**. São Paulo: Novatec, 2014.

SOUSA, Paulino Artur Ferreira de; FRADE, Marta Hansen Lima Basto Correia; MENDONCA, Denisa Maria de Melo Vasques de. Um modelo de organização e partilha de informação de enfermagem entre hospital e centro de saúde: estudo delphi. In: **Acta paul. enferm.**, São Paulo, v. 18, n. 4, Dec. 2005.

TODD, Nick; SZOLKOWSKI, Mark. **JavaServer Pages: o guia do desenvolvedor**. Rio de Janeiro: Elsevier, 2003.

VAREJÃO, Flávio. **Linguagens de Programação: conceitos e técnicas**. Rio de Janeiro: Elsevier, 2004.

VILAÇA, Márcio Luiz Corrêa. Pesquisa e Ensino: considerações e reflexões. **Revista E-scrita**. Nilópolis: v. 1, num 2, mai. – ago. 2010.

W3SCHOOLS. **AJAX Introdução**. Disponível em:
http://www.w3schools.com/ajax/ajax_intro.asp. Acesso em: 19 out. 2014.

WAZLAWICK, Raul Sidnei. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. 2ª edição. Rio de Janeiro: Elsevier, 2011. Disponível em:
<https://books.google.com.br/books?id=Cf6tE2zISf0C&printsec=frontcover&dq=An%C3%A1lise+e+Projeto+de+Sistemas+de+Informa%C3%A7%C3%A3o+Orientados+a+Objetos&hl=pt-BR&sa=X&ei=uVAZVbLsHovFgwTlhYLABw&ved=0CCoQ6AEwAA#v=onepage&q=An%C3%A1lise%20e%20Projeto%20de%20Sistemas%20de%20Informa%C3%A7%C3%A3o%20Orientados%20a%20Objetos&f=false>. Acesso em 30 mar. 2015.

WK SISTEMAS. **Sistema de Controle de estoque**. Disponível em:
<http://www.wk.com.br/Produtos/RadarEstoque.aspx>. Acesso em: 21 out. 2014.

APÊNDICES

APÊNDICE A – CRONOGRAMA

ETAPAS DO LEVANTAMENTO	2014				
	AGOSTO	SETEMBRO	OUTUBRO	NOVEMBRO	DEZEMBRO
1. Especificação dos objetivos do projeto.	X				
2. Estudo sobre os conceitos necessários à criação do projeto.	X	X	X	X	X
3. Levantamento de requisitos.	X	X			
4. Escrever a revisão bibliográfica	X	X	X	X	X
5. Entrega especificação do trabalho	X				
6. Entrega do 1º Capítulo		X			
7. Entrega do 2º Capítulo			X		
8. Entrega do 3º Capítulo			X		

ETAPAS DO LEVANTAMENTO	2015						
	Janeiro	Fevereiro	Março	Abril	Maio	Junho	Julho
9. Escrita do 4º Capítulo	x	x	x				
10. Revisão do Trabalho escrito.				x			
11. Implementação do código fonte do Trabalho proposto.	x	x	x	x			
12. Testes sobre o Sistema implementado.	x	x	x	x			
13. Entrega final					x		
14. Apresentação do TCC						x	

APÊNDICE B – FORMULÁRIO DINÂMICO

```
<list>
  <xml>
    <formulario>
      <tipo>Administrativo</tipo>
      <item>
        <id>521</id>
        <titulo />
        <descricao>Natureza do Serviço</descricao>
        <grupo>22</grupo>
        <legislacao>
          <id />
          <descricao />
        </legislacao>
        <tipo>CmpNaturezaServicoCFM</tipo>
        <requerido>>false</requerido>
        <autodesc>Item obrigatório conforme Resolução CFM Nº 2056/2013,
Resolução CFM Nº 2073/2014</autodesc>
      </item>
      <item>
        <id>522</id>
        <titulo />
        <descricao>Abrangência do Serviço</descricao>
        <grupo>21</grupo>
        <legislacao>
          <id />
          <descricao />
        </legislacao>
        <tipo>CmpAbrangenciaServicoNovo</tipo>
        <requerido>>false</requerido>
```

<autodesc>Item obrigatório conforme Resolução CFM Nº 2056/2013,
Resolução CFM Nº 2073/2014</autodesc>

</item>

<item>

<id>467</id>

<titulo />

<descricao>Tipos de Atendimento</descricao>

<grupo>23</grupo>

<legislacao>

<id />

<descricao />

</legislacao>

<tipo>CmpTiposAtendimentoCFM</tipo>

<requerido>true</requerido>

<autodesc>Item obrigatório conforme Resol. CFM Nº 1716/2004, Cap. III,
Resolução CFM Nº 2073/2014</autodesc>

</item>

<item>

<id>523</id>

<titulo />

<descricao>Horário de Funcionamento</descricao>

<grupo>23</grupo>

<legislacao>

<id />

<descricao />

</legislacao>

<tipo>CmpFuncionamentov2</tipo>

<requerido>true</requerido>

<autodesc>Item obrigatório conforme Resolução CFM Nº 2056/2013,
Resolução CFM Nº 2073/2014</autodesc>

</item>

<item>

<id>524</id>

<titulo />
 <descricao>Possui plantão</descricao>
 <grupo>23</grupo>
 <legislacao>
 <id />
 <descricao />
 </legislacao>
 <tipo>Tipo_3</tipo>
 <requerido>>false</requerido>
 <autodesc>Item obrigatório conforme Resolução CFM Nº 2056/2013,
 Resolução CFM Nº 2073/2014</autodesc>
 </item>
 <item>
 <id>525</id>
 <titulo />
 <descricao>Possui sobreaviso</descricao>
 <grupo>23</grupo>
 <legislacao>
 <id />
 <descricao />
 </legislacao>
 <tipo>CmpRemunerado</tipo>
 <requerido>>false</requerido>
 <autodesc>Item obrigatório conforme Resolução CFM Nº 2056/2013,
 Resolução CFM Nº 2073/2014</autodesc>
 </item>
 <item>
 <id>526</id>
 <titulo />
 <descricao>Consultas agendadas antecipadamente</descricao>
 <grupo>24</grupo>
 <legislacao>
 <id />

<descricao />
 </legislacao>
 <tipo>Tipo_2_CFM</tipo>
 <requerido>>false</requerido>
 <autodesc>Item obrigatório conforme Resolução CFM Nº 2056/2013,
 Resolução CFM Nº 2073/2014</autodesc>
 </item>
 <item>
 <id>527</id>
 <titulo />
 <descricao>Há termo de consentimento esclarecido</descricao>
 <grupo>24</grupo>
 <legislacao>
 <id />
 <descricao />
 </legislacao>
 <tipo>Tipo_2_CFM</tipo>
 <requerido>>false</requerido>
 <autodesc>Item obrigatório conforme Resolução CFM Nº 2056/2013,
 Resolução CFM Nº 2073/2014</autodesc>
 </item>
 <item>
 <id>528</id>
 <titulo />
 <descricao>Os exames físicos são acompanhados por auxiliar de
 sala</descricao>
 <grupo>24</grupo>
 <legislacao>
 <id />
 <descricao />
 </legislacao>
 <tipo>Tipo_2_CFM</tipo>
 <requerido>>false</requerido>

<autodesc>Item obrigatório conforme Resolução CFM Nº 2056/2013,
Resolução CFM Nº 2073/2014</autodesc>

</item>

<item>

<id>529</id>

<titulo />

<descricao>Há material e recursos humanos suficientes para as atividades
desempenhadas</descricao>

<grupo>24</grupo>

<legislacao>

<id />

<descricao />

</legislacao>

<tipo>Tipo_2_CFM</tipo>

<requerido>>false</requerido>

<autodesc>Item obrigatório conforme Resolução CFM Nº 2056/2013,
Resolução CFM Nº 2073/2014</autodesc>

</item>

<item>

<id>530</id>

<titulo />

<descricao>Nº Consultas</descricao>

<grupo>25</grupo>

<legislacao>

<id />

<descricao />

</legislacao>

<tipo>CmpProducaoMedica</tipo>

<requerido>>false</requerido>

<autodesc>Item obrigatório conforme Resolução CFM Nº 2056/2013,
Resolução CFM Nº 2073/2014</autodesc>

</item>

*Continua...