

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

ALINE FERREIRA

PROJETO DE APLICAÇÃO *WEB* POR MEIO DA LINGUAGEM *WEBML*

TRABALHO DE DIPLOMAÇÃO

MEDIANEIRA

2013

ALINE FERREIRA

PROJETO DE APLICAÇÃO *WEB* POR MEIO DA LINGUAGEM *WEBML*

Trabalho de Diplomação apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas – COADS – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. *Msc.* Juliano Rodrigo Lamb.

MEDIANEIRA

2013



TERMO DE APROVAÇÃO

PROJETO DE APLICAÇÃO WEB POR MEIO DA LINGUAGEM WEBML

Por

ALINE FERREIRA

Este Trabalho de Diplomação (TD) foi apresentado às 14:40 h do dia 20 de Agosto de 2013 como requisito parcial para a obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, *Campus* Medianeira. O candidato foi argüido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Juliano Rodrigo Lamb
UTFPR – *Campus* Medianeira
(Orientador)

Prof. Fernando Schütz
UTFPR – *Campus* Medianeira
(Convidado)

Prof. Alessandra B.G. Hoffmann
UTFPR – *Campus* Medianeira
(Convidado)

Prof. Juliano Rodrigo Lamb
UTFPR – *Campus* Medianeira
(Responsável pelas atividades de TCC)

“Agir, eis a inteligência verdadeira. Serei o que quiser. Mas tenho que querer o que for. O êxito está em ter êxito, e não em ter condições de êxito. Condições de palácio tem qualquer terra larga, mas onde estará o palácio se não o fizerem ali?” - Fernando Pessoa.

RESUMO

FERREIRA, Aline. PROJETO DE APLICAÇÃO *WEB* POR MEIO DA LINGUAGEM *WEBML*. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas). Universidade Tecnológica Federal do Paraná. Medianeira 2013.

Com a expansão da *Web* nos últimos anos, a qual está sendo utilizada em diversas áreas, houve o surgimento de aplicações mais complexas. Devido a este avanço e complexidade torna-se fundamental a definição e utilização de métodos adequados para o projeto de aplicações *Web*. A Engenharia *Web* assume este papel, oferecendo modelagens diferenciadas do modelo tradicional, as quais analisam situações específicas da *Web*. Uma destas modelagens é a *Webml*, que é uma linguagem de modelagem visual que permite a representação de aspectos hipertextuais de um sistema *Web*. Este trabalho tem como objetivo realizar um estudo dos conceitos e características da Engenharia *Web* e da modelagem *Webml* bem como análise e projeção de *software* para *Web* por meio das mesmas, aplicados a um estudo experimental voltado a geração e controle de mapas.

Palavras-chaves: Engenharia *Web*, Aplicação *Web*, *Webml*

ABSTRACT

FERREIRA, Aline. DESIGN OF WEB APPLICATION THROUGH LANGUAGE WEBML. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas). Universidade Tecnológica Federal do Paraná. Medianeira 2013.

With the expansion of the Web in recent years, which is being used in different areas, there has been the emergence of more complex applications. Because of this advancement and complexity becomes is fundamental the definition and use of appropriate methods for designing Web applications. The Engineering Web assumes this function by offering differentiated modeling of the traditional model, which analyze specific situations Web. One of these modeling is Webml, which is a visual modeling language that allows hypertext representing aspects of a Web system. This paper intends to develop a study of the concepts and features of Web Engineering and Webml modeling and analysis and projection software for Web through the same applied to an experimental study aimed to generate and control maps.

Keywords: Web Engineering, Web Application, Webml

LISTA DE SIGLAS

CASE	Computer-Aided <i>Software</i> Engineering
ESRI	Environmental Systems Research Institute
HDM	Hypertext Design Model
HTML	HyperText Markup Language
OOHDM	Object Oriented Hypermedia Design Method
RMM	Relationship Management Methodology
UML	Unified Modeling Language
UWE	UML based Web Engineering
WEBML	Web Modeling Language
WWW	Word Wide Web
XML	Extensible Markup Language

LISTA DE FIGURAS

Figura 1 - Exemplo do esquema de estrutura.	25
Figura 2 - Unidade de dados (a) e unidade de dados baseada em HTML (b).	27
Figura 3 - Unidade de múltiplos dados (a) e unidade de dados múltiplos baseada em HTML (b).	28
Figura 4 - Unidade de índice (a) e unidade de índice baseada em HTML (b).	29
Figura 5 - Unidade de rolagem (a) e unidade de rolagem baseada em HTML (b).	29
Figura 6 - Unidade de filtro (a) e unidade de filtro baseada em HTML (b).	30
Figura 7 - Principais elementos da <i>Webml</i> , do modelo de composição.	31
Figura 8 - Notação Gráfica em <i>Webml</i> representando o modelo navegacional.	32
Figura 9 - Diagrama de Caso de Uso.	37
Figura 10 - Diagrama de Estrutura.	41
Figura 11 - Operação de acesso para administrador.	42
Figura 12 - Operação de acesso para usuário.	43
Figura 13 - Diagrama de navegação para cadastro de perfil.	43
Figura 14 - Diagrama de navegação para cadastro de usuário.	44
Figura 15 - Diagrama de navegação para cadastro de <i>shapefile</i>	45
Figura 16 - Diagrama de navegação para gerar mapas.	45

LISTA DE QUADROS

Quadro 1 - Representação de uma unidade de dados em XML.	26
Quadro 2 - Representação de uma unidade de múltiplos dados em XML.	27
Quadro 3 - Representação de uma unidade de índice em XML.	28
Quadro 4 - Representação de uma unidade de rolagem em XML.....	29
Quadro 5 - Notação XML para unidade de filtro.	30
Quadro 6 - Requisito cadastrar shapefile.....	35
Quadro 7 - Requisito importar shapefile	35
Quadro 8 - Requisito gerar mapas.	35
Quadro 9 - Requisito cadastrar perfil de acesso.	36
Quadro 10 - Requisito cadastrar usuário	36

LISTA DE TABELAS

Tabela 1 - Características de sistemas <i>Web</i> simples e avançados.	17
---	----

SUMÁRIO

1 INTRODUÇÃO	11
1.1 OBJETIVO GERAL.....	12
1.2 OBJETIVOS ESPECÍFICOS	12
1.3 JUSTIFICATIVA	12
1.4 ESTRUTURA DO TRABALHO	13
2 ENGENHARIA WEB.....	15
2.1 CARACTERÍSTICAS DE APLICAÇÕES WEB	16
2.1.1 Características de Aplicações Web	18
2.1.2 Características Relacionadas ao Uso	19
2.1.3 Características Relacionadas com o Desenvolvimento	20
2.1.4 Características Relacionadas à Evolução	20
2.2 METODOLOGIAS PARA ANÁLISE DE SISTEMAS WEB.....	21
3 WEBML	23
3.1 MODELO ESTRUTURAL	24
3.2 MODELO DE HIPERTEXTO	25
3.2.1 Modelo de Composição	25
3.2.2 Páginas (<i>Pages</i>)	30
3.2.3 Modelo de Navegação	31
3.3 MODELO DE APRESENTAÇÃO	32
3.4 MODELO DE PERSONALIZAÇÃO	33
4 MATERIAL E MÉTODOS	34
4.1 SISTEMA PARA GERAÇÃO DE MAPAS	34
4.1.1 Visão Geral Do Sistema	34
4.1.2 Requisitos Do Sistema.....	34
4.1.3 Caso De Uso	36

4.2 METODOLOGIA.....	37
4.3 FERRAMENTAS E TECNOLOGIAS	37
4.3.1 WebRatio.....	38
5 RESULTADOS E DISCUSSÕES	39
5.1 ARTEFATOS PARA ANÁLISE DE SISTEMAS WEB.....	39
5.2 MODELO ESTRUTURAL	40
5.3 MODELO DE HIPERTEXTO	41
5.3.1 Identificação de Usuários	42
5.3.2 Modelagem das funcionalidades	43
6 CONSIDERAÇÕES FINAIS.....	47
6.1 CONCLUSÃO.....	47
6.2 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO.....	48
REFERÊNCIAS BIBLIOGRÁFICAS	49

1 INTRODUÇÃO

De acordo com PRESSMAN e LOWE (2009), a *Web* tornou-se uma tecnologia indispensável para quase todos os setores da sociedade (negócios, comércio, comunicação, educação, engenharia, entretenimento, finanças, governos, indústria, mídia, medicina, política, ciência, transporte, dentre outras). Ela permite comprar produtos, encontrar pessoas, entender o mundo, ver notícias, expressar opiniões, divertimento e ainda permite aprendizado *on-line*.

A WWW (*Word Wide Web*) apresentava um conteúdo muito limitado, por meio de textos e gráficos. Com o passar do tempo e avanço das tecnologias envolvidas surgiram os sistemas e aplicações *Web*. “Hoje as *WebApps* se tornaram ferramentas de computação sofisticadas, que não apenas oferecem funcionalidade isolada ao usuário final, mas também foram integradas a banco de dados e aplicações corporativas e governamentais.” PRESSMAN e LOWE, (2009)

Com este avanço, os clientes desejam uma *WebApp* que atenda às suas necessidades, que seja extensível, confiável e funcional. Para isto, é essencial um conjunto de abordagens sistemáticas disciplinadas para o desenvolvimento *Web*. A engenharia *Web* é responsável por este papel.

A engenharia de *software* pode envolver princípios, conceitos e métodos voltados para aplicações *Web*, porém a engenharia *Web* tem uma colocação diferente, novos paradigmas como a multidisciplinaridade, produção de mídias, aspectos estéticos, tecnológicos e de desenvolvimento de *software*.

A necessidade de atender os atributos específicos encontrados nas aplicações *Web* tem proporcionado a criação de estratégias, táticas e métodos, habilitando o engenheiro de *software* a entender, documentar e validar requisitos de aplicações *Web*. (ZANIRO; FABBRI, 200-).

Um conceito pouco conhecido dentro da engenharia *Web* é a linguagem de modelagem *Webml* (*Web Modeling Language*), uma notação para especificação de *Web sites* complexos em um nível conceitual, independente de plataforma, linguagem de programação ou tecnologia. PRESSMAN e LOWE (2009), ressaltam que esta linguagem incorpora suporte robusto para vários aspectos da análise de *WebApps*, tem suporte de uma ferramenta CASE (*Computer-Aided Software Engineering*), gabaritos de modelagem e um conjunto de recursos técnicos.

De acordo com PRESSMAN e LOWE (2009) a *Webml* é uma notação para especificação de sites e permite a descrição alto nível de um *Web site* sobre distintas dimensões ortogonais: seu conteúdo de dados, as páginas que compõe o *site*, a topologia de links entre as páginas e a customização das características de conteúdo de acordo com o perfil do usuário.

O objetivo deste trabalho é realizar um estudo aprofundado da Engenharia *Web* e *Webml*, aplicados a um estudo experimental desenvolvido por CARMONA (2012).

1.1 OBJETIVO GERAL

Demonstrar o processo de Análise e Projeto de *software* para *Web* por meio da linguagem *Webml*, apresentando suas vantagens e características em um estudo experimental.

1.2 OBJETIVOS ESPECÍFICOS

São objetivos específicos deste trabalho:

- a. Apresentar os conceitos fundamentais da engenharia *Web*;
- b. Estudar a linguagem de modelagem *Webml*;
- c. Descrever a análise de um sistema para *Web* por meio da *Webml*, bem como levantamento de requisitos e elaboração dos diagramas adequados;
- d. Elaborar e apresentar os resultados da pesquisa desenvolvida.

1.3 JUSTIFICATIVA

Devido a popularidade da *World Wide Web*, surgiu a necessidade de organizar grandes quantidades de dados de maneira hipertextua, de forma não linear, em que se inserem links de navegação. Caso o *site* possua quantidades significativas de dados, seu *design* se torna um processo complexo que envolve pelo menos dois aspectos que o desenvolvedor

necessita tomar cuidado, de um lado o gerenciamento dos dados e de outro, estrutura de hipertextos (CONNALEN, 2002).

Para modelar aplicações *Desktop* existem linguagens próprias, porém, a maioria delas não dá suporte às características de uma aplicação *Web*. Visando resolver este problema houve o surgimento de linguagens como *Webml*, uma linguagem de modelagem de dados voltada para sites complexos, facilitando a implementação e suportando ferramentas CASE, que auxiliam atividades de engenharia de *software*, desde análise de requisitos, modelagem, até programação e testes. Aspectos importantes da linguagem de modelagem são: diferenciar as preocupações distintas de especificações do *site* (estrutura de informação da página, composição, navegação e apresentação), expressar a composição de páginas em termos de unidades de conteúdo, apoio de recursos avançados (dados derivados, modelagem e personalização de usuário, entrada de dados e operações de atualização), uso extensivo de XML e XSL nos aspectos de design e implantação do *site* (CERI et al., 2001).

A *Webml* é feita com a definição de modelos, são eles: **modelo estrutural**, que consiste no conteúdo de dados da aplicação *Web*, **modelo de composição**, que possibilita a definição de unidades de conteúdo e páginas, **modelo navegacional**, que permite a especificação da maneira pela qual as unidades de conteúdo e páginas estão relacionadas e **modelo de personalização**, que determina características individuais do conteúdo de cada usuário ou grupo de usuário (NETO, 200-).

Segundo GINIGE e MURUGESAN (2005), uma falha em algum ponto do sistema *Web* pode propagar problemas em outras funcionalidades. O desenvolvimento precário com fraco desempenho e conteúdo insatisfatório quando se tratando de aplicações *Web* é extremamente sério, sendo a quantidade de usuários maior comparado a sistemas convencionais, com consequências maiores.

A engenharia *Web* se aplica a estes aspectos, possibilitando uma melhor organização no desenvolvimento do sistema e por meio deste, sucesso aos envolvidos.

1.4 ESTRUTURA DO TRABALHO

O trabalho é apresentado em cinco capítulos. Primeiramente, o capítulo de Introdução, que apresenta os principais objetivos do trabalho, bem como a justificativa do mesmo. Sucessivamente é apresentado um estudo da engenharia *Web* e *Webml*. Por fim é

apresentado um estudo experimental aplicando a *Webml*, juntamente com os Resultados e Discussões que traz os resultados obtidos após a aplicação dos conceitos, e as Considerações finais, onde são apresentadas a conclusão e as sugestões de trabalhos que poderão ser feitos futuramente.

2 ENGENHARIA WEB

Sistemas *Web* criam uma complexa matriz de funcionalidades e conteúdo para seus usuários finais. Por meio do processo de engenharia *Web*, criam-se aplicações de alta qualidade. A engenharia *Web* não é idêntica à engenharia de *software*, mas herda muitas características, conceitos e princípios, focando nas atividades técnicas e gestão. As maneiras pelas quais essas atividades são conduzidas se diferem, mantendo, contudo, a essência para o desenvolvimento de um sistema. (PRESSMAN, 2002).

Segundo PRESSMAN (2011), a mais sucinta definição de engenharia de *software* foi proposta por Fritz Bauer, onde este cita que o uso de princípios da engenharia estabelece a possibilidade de obter economicamente um *software* confiável com funcionamento eficiente. PRESSMAN (2000) ainda salienta que “*software*, em todas as suas formas e em todos os seus campos de aplicação, deve passar pelos processos de engenharia”.

Para SOMMERVILE (2011):

“Engenharia de *software* é uma abordagem sistemática para produção de *software*; ela analisa questões práticas de custo, prazo e confiança, assim como as necessidades dos clientes e produtores do *software*. A forma como essa abordagem sistemática é realmente implementada varia dramaticamente de acordo com a organização que esteja desenvolvendo o *software*, o tipo de *software* e as pessoas envolvidas no processo de desenvolvimento.”

Com o aumento de aplicativos para *Web* e, conseqüentemente, o aumento da complexidade que esses aplicativos vêm apresentando, torna-se fundamental a definição e utilização de métodos ou processos de engenharia adequados para o projeto de tais aplicações, sobretudo por possuírem uma série de particularidades, como estética, navegação, segurança, conteúdo, escalabilidade, por exemplo, que diferem de sistemas convencionais. Devido a essas particularidades que a engenharia de *software* em si não atende completamente, surge a necessidade de utilização da engenharia *Web*.

A engenharia *Web* lida com os aspectos do desenvolvimento de sistemas *Web*, desde a sua concepção, desenvolvimento, até a implementação e evolução. O objetivo é gerenciar a diversidade e complexidade do desenvolvimento das aplicações *Web*, impedindo falhas e implicações no projeto (GINIGE; MURUGESAN, 2001).

De acordo com JACYNTHO (2008), as aplicações *Web* passaram de simples *Web sites*, cujo propósito era a navegação sobre a informação para verdadeiros e complexos sistemas de informações, repleto de dados e transações, definidos para a implementação de processos de negócio intra e inter-organização. Neste contexto, a necessidade e evolução de

um processo de *software* sistemático que gerencie o ciclo de vida das aplicações foram constantemente surgindo.

Ainda segundo JACYNTHO (2008):

“Poder-se-ia argumentar que sistemas *Web* não são diferentes, sob o ponto de vista de processo de *software*, de sistemas de *software* convencionais. Não obstante, existe um crescente reconhecimento que sistemas *Web* possuem características particulares que não são apropriadamente consideradas pelos processos de *software* tradicionais. O desenvolvimento de aplicações *Web* é realizado por equipes multidisciplinares, com diferentes habilidades, em prazos curtíssimos ditados pela voraz concorrência e em um contexto extremamente volúvel, marcado por incertezas.”

Para alcançar maior sucesso no desenvolvimento e na aplicação de sistemas complexos e evitar uma *Web* entrelaçada, utiliza-se abordagens de disciplinas de engenharia *Web*, novos métodos e ferramentas para o desenvolvimento, implantação e avaliação dos sistemas. Essas abordagens e técnicas devem levar em conta as características do meio, ambientes e cenários, perfis de usuários, situações estas que desafiam o desenvolvimento de aplicações *Web*. (KAPPEL, et al. 2006).

2.1 CARACTERÍSTICAS DE APLICAÇÕES WEB

É fundamental não confundir **aplicação na Web** e **aplicação Web**. A primeira é qualquer tipo de aplicação que utiliza a *Web* como ambiente de execução, como exemplo, um simples repositório de arquivos ou mesmo um sistema *Desktop*, tradicional, composto por buscas e formulários. Já a aplicação *Web* explora o paradigma hipermídia, possui uma estrutura navegacional bem definida. (JACYNTHO, 2008). O foco deste trabalho está em aplicações *Web*.

Para SUH (2005), as aplicações *Web* são revolucionárias. No início do desenvolvimento das mesmas não é possível especificar as necessidades reais dos sistemas e o que irão conter. Além do mais a estrutura e funcionalidade de um sistema *Web* mudam constantemente. Ainda para SUH, os principais desafios de desenvolvimento para a *Web* incluem o design de usabilidade, manutenção de conteúdos, integração com sistemas legados, implantação imediata das aplicações, escalabilidade e balanceamento de carga.

PRESSMAN, (2002) diz que a maioria das *WebApps* (aplicações *Web*) possuem as características relacionadas a seguir:

- a. **Redes concentradas:** é necessário que uma *WebApp* resida em uma rede, seja ela Internet, intranet ou extranet;
- b. **Impulsionadas pelo conteúdo:** a apresentação do conteúdo de texto, gráficos, áudio e vídeo ao usuário final é feita a partir do uso da hipermídia, o que, em vários casos, é a função principal de uma *WebApp*;
- c. **Evolução continuada:** as aplicações da *Web* possuem evolução contínua, ao contrário do *software* de aplicação convencional, que evolui de acordo com uma série de versões planejadas e cronologicamente espaçadas;
- d. **Imediatismo:** *WebApps* possuem um imediatismo que, geralmente, não são encontrados em outros tipos de *softwares*. Sendo assim, é necessário que os desenvolvedores utilizem métodos para planejamento, análise, projeto, implementação e teste, adaptados ao prazo requerido para o desenvolvimento da aplicação;
- e. **Segurança:** é importante implementar meios de segurança em toda infra estrutura, a fim de proteger o conteúdo das *WebApps*, fornecendo segurança na transmissão de dados;
- f. **Estética:** o sucesso de uma *WebApp* está diretamente relacionado com sua estética, tendo em vista que seu aspecto se torna parte da atração para os usuários.

GINIGE e MURUGESAN (2001) definem características de sistemas *Web* simples e avançados. A Tabela 1 apresenta essas características.

Tabela 1 - Características de sistemas *Web* simples e avançados.

Sistemas <i>Web</i> simples	Sistemas <i>Web</i> avançados
Páginas simples, apenas com informações textuais	Páginas complexas
Conteúdo estático (a informação de conteúdo é imutável)	Conteúdo dinâmico (a informação muda em função do tempo e da necessidade do usuário)
Navegação simples	Dificuldade para navegar e encontrar informações
Sistemas <i>stand alone</i>	Integrados com banco de dados e outros sistemas de programação, planejamento e monitoramento
Boa performance não é requisito principal	Requer boa performance e disponibilidade contínua
Desenvolvido por um único indivíduo ou por uma equipe pequena	Requer uma equipe bem desenvolvida e especialistas em especialistas em diversas áreas
Utilizada para a disseminação da informação que não se relaciona ao foco da aplicação	Implantado em aplicações de missões críticas

Fonte: GINIGE; MURUGESAN (2001).

Desta forma, considerando aplicações complexas, as quais estão cada vez mais presentes e necessárias nos dias de hoje, torna-se necessário um maior estudo e planejamento,

sendo este diferenciado da engenharia de *software* tradicional, a qual já não atende todos os quesitos demandados de uma aplicação *Web*, conforme citado pelos autores acima.

Segundo GINIGE e MURUGESAN (2001), uma pesquisa realizada pelo *Cutter Consortium* (relatado em seu Resumo de Pesquisa, 07 de novembro de 2000) sobre o desenvolvimento de projetos baseados na *Web*, identificou os seguintes problemas:

- a. As necessidades de negócios não foram atendidas por 84% dos sistemas entregues;
- b. 79% dos projetos não foram entregues no prazo determinado;
- c. 63% dos projetos extrapolaram o orçamento;
- d. 53% dos projetos apresentados não possuíam a funcionalidade requerida;

Com esse resultado, segundo GINIGE e MURUGESAN (2001), a preocupação de desenvolvedores, usuários e outros membros interessados, com relação ao desenvolvimento de complexas *WebApps*, aumentaram.

Para KAPPEL et al. (2003) algumas características devem ser levadas em consideração no desenvolvimento de aplicações *Web*, como características de aplicações *Web*, características relacionadas ao uso, características relacionadas ao desenvolvimento e características relacionadas à evolução.

2.1.1 Características de Aplicações *Web*

Segundo KAPPEL et al. (2003), a funcionalidade é importante no desenvolvimento de aplicações *Web*, mas devem ser considerados também o conteúdo, hipertextos e aspectos de apresentação:

- a. **Conteúdo:** envolve os dados estruturados que se encontram em sistemas de banco de dados, bem como os não estruturados e semiestruturados, como por exemplo, descrições textuais ou informações multimídia. Surge a complexidade pelo fato do conteúdo ser dinâmico e constantemente atualizado, e devido a exigência por parte dos usuários com relação à atualização, rigor, confiabilidade e consistência.
- b. **Hipertexto:** aplicações *Web* defendem o modelo de hipertexto como fundamental para a estruturação dos dados. A não linearidade é característica básica do modelo de hipertexto, podendo ser obtida Por meio de projetos específicos de navegação, como

mapas de *site*, palavra-chave e buscas, fatores estes que situarão os usuários no sistema, garantindo melhor acessibilidade.

- c. **Apresentação:** é um fator principal em relação à qualidade de aplicações *Web*. Sofre constantes mudanças devido aos novos recursos técnicos e tendências. A apresentação deve ser clara e intuitiva para os usuários, pois estes, diferentemente de usuários de sistema convencionais não terão acesso a manuais explicativos referentes ao uso do sistema.

2.1.2 Características Relacionadas ao Uso das Aplicações Web

KAPPEL et al., (2003), ressalta que diferentemente dos sistemas convencionais, os sistemas *Web* possuem geralmente um número maior de usuários, que têm a opção de escolha quanto a hora e forma de acesso as aplicações *Web*.

- a. **Contexto natural:** inclui aspectos de localização e tempo de acesso. Devido a computação móvel é necessário fornecer novos tipos de serviços. Outro fator fundamental de uma aplicação *Web* é a disponibilidade contínua para o usuário, que deve ter a opção de acesso a qualquer momento.
- b. **Infraestrutura técnica imprevisível:** dispositivos de usuário final variam em relação às capacidades de *hardware* e *software*. Tamanho da tela diferenciado, capacidade computacional, versão de navegadores, conexões de redes, largura da banda, são exemplos de características que devem ser levadas em consideração, pois implicam na qualidade do serviço. A complexidade em relação à infraestrutura aumenta de acordo com as configurações individuais por parte dos usuários, que podem deixar de lado algumas configurações essenciais para o bom funcionamento de uma aplicação *Web*.
- c. **Diversidade e magnitude de usuários:** os usuários de aplicações *Web* diferem em cultura, idade, objetivos, habilidades e capacidades. Desta forma, essas variadas características devem ser levadas em conta pelos desenvolvedores, para que a aplicação *Web* atenda a todos os tipos de usuários. Outra importante característica que deve ser levada em conta em um aplicativo *Web* é a escalabilidade, devido ao grande número de acessos.

2.1.3 Características Relacionadas com o Desenvolvimento das Aplicações Web

Para KAPPEL et al., (2003), os desenvolvedores de aplicações *Web* devem lidar com condições, riscos e incertezas, as quais não pertencem a *softwares* tradicionais.

- a. **Equipe de desenvolvimento:** o desenvolvimento de aplicações *Web* envolve uma mistura de editores de impressão, autores, desenvolvedores de *software*, especialistas em marketing, *designers* de arte. Geralmente membros mais jovens, que estão em busca de inovações, não apenas aderindo convenções.
- b. **Ambiente de desenvolvimento:** a infraestrutura deve ser volátil e heterogênea, pelo fato do desenvolvimento de aplicações *Web* contarem com variados componentes, como servidor *Web*, servidor de aplicação, banco de dados, editores de publicação, dentre outros.
- c. **Integração com sistemas legados:** geralmente aplicações *Web* necessitam se integrar com sistemas legados, porém muitas vezes estes sistemas são pouco documentados e modificados sem aviso prévio, interferindo na qualidade das aplicações *Web*.
- d. **Processo:** os processos para desenvolvimento de aplicações *Web* geralmente estão em constantes mudanças, em razão do crescente desenvolvimento tecnológico, novas tendências, exigências voláteis e cronogramas rígidos.

2.1.4 Características Relacionadas à Evolução das Aplicações Web

Diferentemente de aplicações convencionas, que se desenvolvem ao longo de lançamentos previstos, cronologicamente espaçados, as aplicações *Web* evoluem continuamente, estão sujeitas a mudanças devido a evolução tecnológica e a volatilidade dos usuários *Web*, surgindo a situação de competitividade (KAPPEL et al., 2003).

2.2 METODOLOGIAS PARA ANÁLISE DE SISTEMAS WEB

Conforme exposto anteriormente, devido às particularidades das aplicações *Web*, novas metodologias direcionadas para o desenvolvimento foram surgindo, tornando-se necessárias para um bom desenvolvimento *Web*.

A metodologia envolve as atividades que proporcionam o detalhamento para a construção de uma aplicação (PRESSMAN, 2000).

Para MACHADO et al, (2008):

“A *Unified Modeling Language* (UML) tornou-se a linguagem padrão para modelar aplicações orientadas a objetos. Tanto no mundo acadêmico quanto no mundo comercial, seus diagramas e definições são utilizadas em modelos de aplicações em geral. No entanto quando se trata de modelar aplicações *Web* novos desafios se impõem, pois a navegação, interação com o usuário e tipos de conteúdos não encontram suficiente suporte no núcleo da UML.”

Segundo LANGEGER, et al., (200-), com o crescimento de aplicações *Web*, as relevâncias de técnicas de modelagens que suportam essas aplicações foram surgindo. Afirmando ainda que, muitos conceitos científicos tentam combinar modelos já existentes para a análise de hipermídia, como exemplo modelos HDM ou RMM, com técnicas da engenharia de *software*. O modelo OOHDH tenta combinar a modelagem hipermídia com métodos de *software* orientado a objetos. A *WEBML* fornece um grande conjunto de padrões e modelos para aplicações *Web*.

São relacionadas algumas modelagens propostas, encontradas na literatura:

- a. *HDM (Hypertext Design Model)* é um modelo para aplicações hipermídia. É um dispositivo de modelagem que fornece formas concisas de descrever o sistema, ajudando o autor a conceituar uma aplicação sem relacionar com os detalhes de implementação, podendo ser usado como uma linguagem de comunicação entre projetistas, implementadores e usuários (GARZOTTO et al., 1993)
- b. *OOHDH (Object Oriented Hypermedia Design Method)* surgiu a partir da necessidade de abstrações, afim de facilitar a especificação de aplicações hipermídia. É composto por quatro atividades diferentes, sendo elas, projeto conceitual, projeto de navegação, interface de concepção abstrata e implementação (ROSSI, et al., 1998).
- c. *RMM (Relationship Management Methodology)* é utilizada para projeto e desenvolvimento de aplicações hipermídia, baseando-se no gerenciamento de relacionamentos entre objetos de informação (ISAKOWITZ et al., 1995).

- d. *UWE (UML based Web Engineering)*, um método para hipermídia adaptativa, tem como foco a definição de um processo sistemático. É uma modelagem orientada a objeto que utiliza técnicas, notações e extensões exclusivamente da UML (KOCH; WIRSING, 2001).
- e. *WEBML (Web Modeling Language)* é uma linguagem de modelagem para aplicações *Web*, que possibilita os projetistas modelarem em alto nível as funcionalidades de um sistema *Web*, sem se comprometerem com detalhes de arquitetura específica (CERI et al., 2001). Esta linguagem será o foco do trabalho, a qual será estudada e aplicada nos capítulos futuros.
- f. *IFML (Interaction Flow Modeling Language)* foi projetada para expressar o conteúdo, interação com o usuário e controle de comportamento *do front-end* de aplicações de *software*. Foi desenvolvida em 2012 e 2013 sob a liderança da WebRatio e inspirada pela notação *Webml*, bem como por algumas outras experiências no campo de modelagem na *Web*, se encontra na versão beta (IFML, 2013). A IFML é considerada ainda como uma evolução da *Webml* mas não substitui a linguagem.

3 WEBML

Segundo CERI et al. (2001), a *Webml* permite analisar as principais características de um sistema em nível elevado, não se comprometendo com os detalhes da arquitetura. *Webml* está associada a representação gráfica intuitiva, suportando ferramentas CASE e também sintaxe XML. A linguagem *Webml* possui quatro perspectivas ortogonais, sendo estas, modelo estrutural, modelo de hipertexto, modelo de composição e modelo de navegação. Todos estes modelos serão tratados posteriormente.

De acordo com WEBML (2013), *Webml* é uma linguagem de notação visual simples, que fornece gráficos e especificações, relacionados em um processo de projeto completo, que podem ser auxiliados por ferramentas de design visual. Os principais objetivos do processo de design *Webml* são:

- a. Expressar a estrutura de uma aplicação *Web*, com descrição de alto nível, utilizando-a para consultas, atualizações e manutenções;
- b. Prover diversas visões do mesmo conteúdo;
- c. Separar o conteúdo de informação, partindo da sua composição em páginas, navegação e apresentação;
- d. Armazenar informações obtidas no processo de projeto dentro de um repositório, podendo as mesmas ser utilizadas durante todo o tempo de vida de uma aplicação para gerar, de maneira dinâmica, as páginas da *Web*;
- e. Permitir a especificação de políticas de personalização, modelando os usuários e comunidades que devem estar explicitamente no repositório;
- f. Permitir a especificação de operações de manipulação de dados para atualização de conteúdos do *site*.

A *Webml* descreve aplicações *Web* em 4 dimensões: (CERI et al., 2001).

- a. **Modelo Estrutural (Structural Model):** demonstra o conteúdo dos dados da aplicação, em termos de entidades relevantes e relacionamentos. *Webml* não oferece ainda uma linguagem diferencial para modelagem de dados, mas é compatível com notações clássicas como por exemplo, entidade-relacionamentos e diagrama de classes da UML;
- b. **Modelo de Hipertexto (Hypertext Model):** descreve os hipertextos publicados no *site*. Cada hipertexto diferente define uma visão no sistema. Estas descrições de visualização consistem em dois submodelos:

- i. *Composição de Modelo*: especifica quais as páginas que compõem o hipertexto e quais unidades de conteúdo compõe uma página;
 - ii. *Modelo de Navegação*: especifica como as páginas e unidades de conteúdo estão ligadas formando um hipertexto.
- c. **Modelo de Apresentação (Presentation Model)**: expressa o layout e a aparência gráfica das páginas, independentemente da linguagem final que representará essas páginas;
 - d. **Modelo de Personalização (Personalization Model)**: modela os usuários e grupos de usuários para armazenamento de informações especificam dos mesmos.

Ainda, segundo o autor, além dos modelos apresentados há um processo de design em *Webml* que deve ser considerado. O desenvolvimento de aplicações *Web* envolve diferentes atores com diversas habilidades e objetivos. Sendo assim, a separação de interesses é um requisito fundamental para uma modelagem *Web*. *Webml* envolve esta questão, assumindo um processo de desenvolvimento, onde diferentes tipos de especialistas desempenham atividades distintas.

3.1 MODELO ESTRUTURAL

O modelo estrutural define o conteúdo dos dados da aplicação, em termos de entidades e relacionamentos. Para CERI et. al., (2000), o objetivo da modelagem de dados é permitir a especificação dos dados de uma maneira formal e intuitiva. A modelagem resulta em uma apresentação simples do conhecimento sobre a aplicação, útil tanto para o desenvolvedor de negócio, o qual irá operar os dados, quanto para o desenvolvedor da estrutura física de suporte ao armazenamento, atualização e recuperação dos dados.

Para NETO (200-), os elementos fundamentais para a definição do modelo estrutural são entidades e relacionamentos.

- a. **Entidades**: contêm os dados armazenados, são formadas por um nome e atributos associados a um tipo de dado; sendo que *Atributos*: representam as propriedades dos objetos do mundo real, as quais são relevantes à aplicação.
- b. **Relacionamentos**: permitem a ligação semântica entre as entidades, são definidos pelo nome e a cardinalidade das entidades que o compõem.

A Figura 1 representa um modelo estrutural, onde são especificadas as entidades que compõe o sistema e os relacionamentos entre estas.

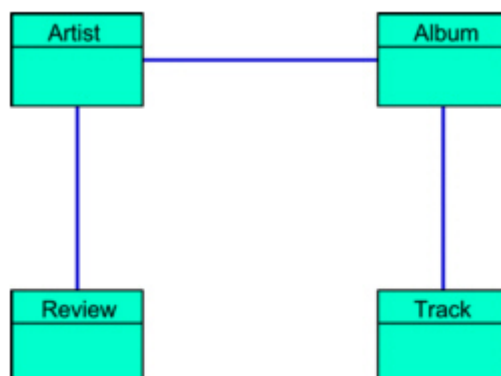


Figura 1 - Exemplo do esquema de estrutura.
Fonte: CERI et al., (2001).

3.2 MODELO DE HIPERTEXTO

O modelo de hipertexto descreve os hipertextos publicados no *site*. Cada hipertexto diferente define uma visão no sistema. É dividido em três submodelos, modelo de composição, dados e navegação.

3.2.1 Modelo de Composição

O objetivo do modelo de composição é especificar quais as páginas que compõem o hipertexto e quais as unidades de conteúdo compõem uma página. Para NETO (200-), as unidades de conteúdo definem a forma pelo quais os dados de uma determinada entidade serão exibidos, customizando os atributos desejados.

Segundo CERI et al., (2000), unidades são elementos atômicos que especificam o conteúdo de uma página *Web*. A *Webml* suporta seis tipos de unidades, unidade de dados, dados múltiplos, índice, rolagem, filtro e páginas.

3.2.1.1 Unidade de Dados (*Data unit*)

As unidades de dados mostram informações sobre um simples objeto, como por exemplo, instância de uma entidade ou de um componente. Essa unidade seleciona um mistura de informações, que fornece uma visão significativa de um determinado conceito do esquema de estrutura. Mais de uma unidade de dados pode ser definida para a mesma entidade, oferecendo alternativos pontos de vista. A unidade de dados é caracterizada Por meio do nome da unidade, da entidade que representa, dos atributos do objeto e de um seletor opcional que identifica um objeto.

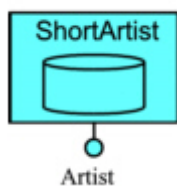
O Quadro 1 apresenta uma entidade representada pela notação XML. A tag DATAUNIT define a unidade de dados, onde são especificados o nome da unidade e da entidade que representa. DATAUNIT fornece tags e atributos para diversos aspectos da definição da unidade. A tag INCLUDE define o nome do atributo e a tag INCLUDEALL define todos os atributos que serão adicionados a essa unidade de dados.

```
<DATAUNIT id="ShortArtist" entity="Artist">
<INCLUDE attribute="firstName"/>
<INCLUDE attribute="lastName"/>
<INCLUDE attribute="photo"/>
</DATAUNIT>
<DATAUNIT id="BiographyUnit" entity="Artist">
<INCLUDEALL/>
</DATAUNIT>
```

Quadro 1 - Representação de uma unidade de dados em XML.

Fonte: CERI et al., (2001).

A Figura 2 (a) demonstra a representação gráfica segundo a *Webml*, para uma unidade de dados. Dentro do quadrado é apresentado o nome da unidade de dados e logo abaixo a entidade a qual se refere. A Figura 2 (b) apresenta os dados da entidade, de acordo com uma representação HTML, onde no primeiro quadrado é apresentado nome da entidade, seguindo com os seletores únicos de cada objeto. No exemplo, é apresentado uma instância da entidade, cujo primeiro nome (First Name) é Celine e o sobrenome (Last Name) Dion.



(a)



(b)

Figura 2 - Unidade de dados (a) e unidade de dados baseada em HTML (b).
Fonte: CERI et al., (2001).

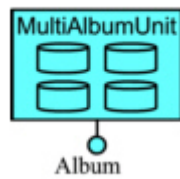
3.2.1.2 Unidades de Dados Múltiplos (*Multi-DataUnits*)

Conforme CERI et al., (2000), unidades de dados múltiplos apresentam múltiplas instâncias de uma entidade ou componentes agrupados, apresentados de maneira idêntica. A especificação da unidade de dados múltiplos possui, então, duas partes: o recipiente que inclui as instâncias a serem exibidas, podendo se referir à entidade, relacionamento ou componente; e a unidade de dados possuindo as informações que serão apresentadas. A unidade de dados múltiplos é representada pela tag `MULTIDATAUNIT`, a qual inclui um elemento `DATAUNIT` alojado. O Quadro 2 demonstra essa explicação.

```
<MULTIDATAUNIT id="MultiAlbumUnit" entity="Album">
<DATAUNIT id="AlbumUnit" entity="Album">
<INCLUDEALL/>
</DATAUNIT>
</MULTIDATAUNIT>
```

Quadro 2 - Representação de uma unidade de múltiplos dados em XML.
Fonte: CERI et al., (2001).

A Figura 3 (a) demonstra uma unidade de dados múltiplos, conforme a *Webml*. Dentro do quadrado é descrito o nome da unidade e logo abaixo a entidade a qual está representando. A Figura 3 (b) apresenta os dados da entidade, de acordo com uma representação HTML.



(a)

Diagram (b) shows a window titled 'All Albums' with a blue header bar. The window contains a table with four rows and two columns. Each row represents an album with its title and artist, and a checkbox to its right.

All Albums	
All The Way Celine Dion	Another Green World Brian Eno
Anthology Beatles	Beginnings Frank Sinatra

(b)

Figura 3 - Unidade de múltiplos dados (a) e unidade de dados múltiplos baseada em HTML (b).
Fonte: CERI et al., (2001).

3.2.1.3 Unidade de Índice (*Index Units*)

Unidades de Índice apresentam várias instâncias de uma entidade ou componente como uma lista, sem apresentar detalhes. A especificação da unidade de índice possui duas partes principais: o recipiente que inclui as instâncias a serem exibidas e os atributos utilizados como índices da entidade na lista. A unidade de índice é representada pela *tag* INDEXUNIT, a qual define o nome da unidade, entidade que representa e o atributo que será utilizado como índice, determinada pelo elemento DESCRIPTION alojado, por meio do atributo *Key*, entre as *tags* INDEXUNIT. O quadro 3 demonstra uma lista de álbuns, exibindo apenas o título de cada álbum em formato XML.

```
<INDEXUNIT id="AlbumIndex" entity="Album">
<DESCRIPTION Key="title"/>
</INDEXUNIT>
```

Quadro 3 - Representação de uma unidade de índice em XML.
Fonte: CERI et al., (2001).

A Figura 4 (a) demonstra uma unidade de índice, sendo o nome da unidade é apresentado dentro do quadrado e logo abaixo a entidade a qual apresenta no ícone que segue. A Figura 4 (b) apresenta os dados da entidade, de acordo com uma representação HTML.



Figura 4 - Unidade de índice (a) e unidade de índice baseada em HTML (b).
Fonte: CERI et al., (2001).

3.3.1.4 Unidade de Rolagem (*Scroller Units*)

Unidades de rolagem fornecem comandos para percorrer os objetos em um recipiente. Geralmente é usada em conjunto com uma unidade de dados, a qual representa o elemento visualizado do recipiente. A unidade de rolagem é definida pela *tag* SCROLLERUNIT. É necessário definir o nome da unidade, a entidade a qual representa e um seletor opcional, que ressalva o conjunto de objetos. O Quadro 4 determina a notação XML para unidade de rolagem.

```
<SCROLLERUNIT id="AlbumScroll" entity="Album" first="yes"
last="yes" previous="yes" next="yes"/>
```

Quadro 4 - Representação de uma unidade de rolagem em XML.
Fonte: CERI et al., (2001).

A Figura 5 (a) demonstra uma unidade de rolagem. A Figura 5 (b) apresenta os dados da entidade, de acordo com uma representação HTML.



Figura 5 - Unidade de rolagem (a) e unidade de rolagem baseada em HTML (b).
Fonte: CERI et al., (2001).

3.3.1.5 Unidade de Filtro (*Filter Units*)

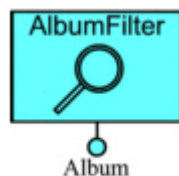
Unidades de filtro fornecem campos de entrada para procurar os objetos em um recipiente. A unidade de Filtro normalmente é usada em conjunto com a unidade de índice ou de múltiplos dados, que apresentam os valores correspondentes a pesquisa realizada. A tag `FILTERUNIT` é usada, para especificar o recipiente, forçando o conjunto de objetos a serem pesquisados. Dentro da tag `FILTERUNIT`, a tag `SEARCHATTRIBUTE` é utilizada para especificar um atributo específico, informando qual o atributo que deverá ser pesquisado. O quadro 5 ilustra a notação XML para unidade de filtro.

```
FILTERUNIT id="AlbumFilter" entity="Album"/>
</SEARCHATTRIBUTE name="title" predicate="like">
</SEARCHATTRIBUTE name="year" predicate="between">
</FILTERUNIT>
```

Quadro 5 - Notação XML para unidade de filtro.

Fonte: CERI et al., (2001).

A Figura 6 (a) demonstra uma unidade de filtro, sendo o nome da unidade apresentado dentro do quadrado e logo abaixo a entidade a qual representa. A Figura 6 (b) apresenta os dados da entidade, de acordo com uma representação HTML.



(a)

(b)

Figura 6 - Unidade de filtro (a) e unidade de filtro baseada em HTML (b).

Fonte: CERI et al., (2001).

3.2.2 Páginas (*Pages*)

Para que as informações de diversas unidades sejam entregues em conjuntos, é utilizada uma página. Páginas são os elementos da interface apresentados para os usuários. É representada na *Webml* por meio de um retângulo, onde na parte superior, é apresentado o

nome da página, dentro do retângulo são especificados os elementos que serão exibidos nessa página.

A Figura 7 apresenta os principais elementos da *Webml*, do modelo de composição.

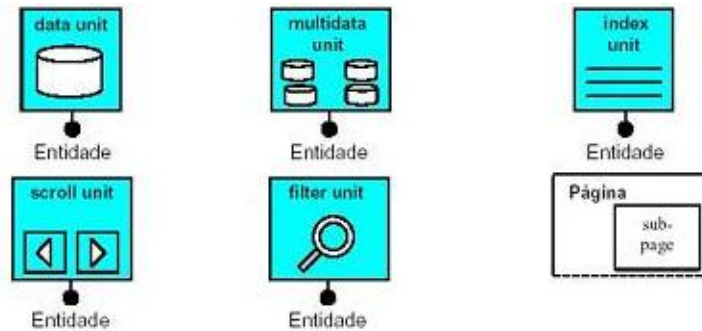


Figura 7 - Principais elementos da *Webml*, do modelo de composição.
Fonte: CERI et al., (2001).

3.2.3 Modelo de Navegação

As unidades e páginas não existem isoladamente, as mesmas devem estar interligadas para formar uma estrutura de hipertexto, por meio de links. A definição da modelagem de navegação é especificar o modo que as unidades e as páginas se interligam para formar um hipertexto. Links são representados por meio de setas direcionando a origem e o destino. A *Webml* fornece dois diferentes links, que fornecem essa ligação:

- a. *Links contextuais*: conectam as páginas de forma coerente para expressar uma relação semântica entre as entidades, modelados Por meio do modelo de estrutura;
- b. *Links não contextuais*: estes conectam as páginas de forma livre, sem nenhuma ligação semântica.

A Figura 8 demonstra a notação gráfica *Webml* para representar ligações contextuais, juntamente com uma implementação baseada em HTML. Neste exemplo, cada unidade é apresentada em uma página separada, interligadas entre si.

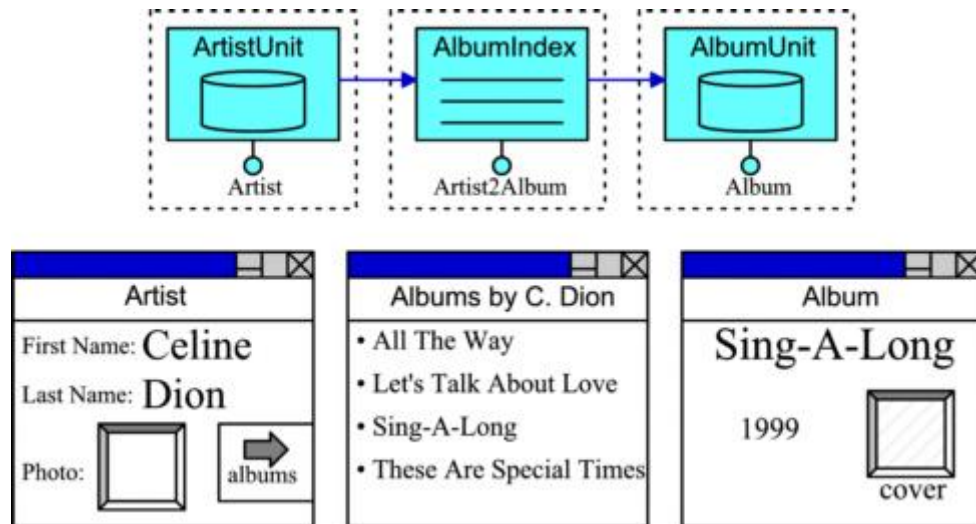


Figura 8 - Notação Gráfica em Webml representando o modelo navegacional.
Fonte: CERI et al., (2001).

Os links contextuais transmitem parâmetros entre as páginas. Segue as regras que definem as informações de contexto que fluem Por meio das unidades, são elas:

- a. *Unidades de dados*: o identificador do objeto mostrado na unidade;
- b. *Unidade de índice*: o valor chave selecionado na lista;
- c. *Unidade de rolagem*: o identificador do objeto selecionado Por meio dos comandos de rolagem;
- d. *Filtro de unidades*: os valores dos atributos apresentados na forma de entrada de dados;
- e. *Unidade de dados múltiplos*: a informação de contexto associada com a unidade de dados.

3.3 MODELO DE APRESENTAÇÃO

De acordo com CERI, et al., (2000), o modelo de apresentação, define a aparência e o layout das páginas. As páginas *Webml* são renderizadas de acordo com uma folha de estilos. As folhas de estilos definem os elementos do conteúdo a serem inseridos em tais páginas e o layout, independente da linguagem de desenvolvimento. A *Webml* define duas categorias de folhas de estilos: folhas de estilo não tipadas, que descrevem páginas de layout independentemente do conteúdo, e folhas de estilo tipadas, sendo estas específicas para páginas com determinado conteúdo.

3.4 MODELO DE PERSONALIZAÇÃO

O modelo de personalização define o conteúdo ou estilo de apresentação de acordo com o perfil do usuário. Em *Webml*, unidade, páginas, layout de apresentação podem ser definidos por usuários ou grupos de acordo com os dados específicos. O acesso aos sites podem ser feitos por usuários registrados, com preferências específicas, ou mesmo por visitantes eventuais, não identificados pelo sistema. A *Webml* trata como usuários registrados os que terão acesso apenas a visões protegidas e não registrados, os quais terão acesso a visões acessíveis por qualquer visitante. A *Webml* modela tais características de personalização conforme a definição de usuário, grupo, visões e o relacionamento entre estes elementos. Os sistemas que incluem restrições de acesso devem modelar um esquema, conforme modelo de estrutura, com criações entidades de usuários, grupos e visões, e relacionamentos entre estas.

4 MATERIAL E MÉTODOS

Visando aplicar a linguagem *Webml*, foi analisado como estudo experimental, um sistema *Web* que possibilita a criação de mapas de forma dinâmica, conforme proposto em CARMONA (2012).

4.1 SISTEMA PARA GERAÇÃO DE MAPAS

As próximas seções apresentam o sistema desenvolvido em CARMONA (2012) por meio dos elementos de análise da *Engenharia de Software*.

4.1.1 Visão Geral Do Sistema

O estudo experimental proposto compreende um sistema em ambiente *Web* que possibilita a criação de mapas de forma dinâmica por meio de shapefiles¹ disponibilizados pelo próprio usuário. Após o upload dos shapefiles as informações dos mesmos ficam disponíveis ao usuário para a geração de mapas de acordo com suas necessidades.

4.1.2 Requisitos Do Sistema

Os Requisitos de sistema são classificados como funcionais e não funcionais. Os funcionais especificam as funcionalidades do sistema e os requisitos não funcionais

¹ *Shapefile* é uma estrutura de dados não topológica, desenvolvido e mantido pelo *Environmental Systems Research Institute* (ESRI), com o objetivo de armazenar geometrias e atributos de dados espaciais em um conjunto de dados vetoriais (ESRI (1998), citado por CARMONA (2012)).

apresentam as restrições/formas pelas quais as funcionalidades são implementadas. Os Quadros 6 até o 10 apresentam os requisitos do sistema de geração de mapas.

F1 Cadastrar <i>shapefile</i>		Oculto()		
Descrição: o sistema deverá permitir o cadastro de <i>shapefiles</i> , indicando um nome e os arquivos que compõe o <i>shapefile</i> (shp, shx, dbf);				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF1.1 Controle de Acesso	A função só pode ser acessada por usuários com perfil que permita o acesso a tela de cadastro de <i>shapefile</i>	Segurança	()	(x)
NF1.1 Identificador do <i>shapefile</i>	O <i>shapefile</i> será identificado pelo nome	Interface	()	(x)

Quadro 6 - Requisito cadastrar shapefile.

Fonte: CARMONA (2012).

F2 Importar <i>shapefile</i>		Oculto()		
Descrição: o sistema deverá permitir a importação de <i>shapefiles</i> para o banco de dados por meio da criação de uma nova tabela.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF2.1 Controle de Acesso	A função só pode ser acessada por usuários com perfil que permita o acesso a tela de cadastro de <i>shapefile</i>	Segurança	()	(x)
NF2.1 Identificador da tabela	A tabela será identificada pelo nome cadastrado junto ao <i>shapefile</i>	Interface	()	(x)

Quadro 7 - Requisito importar shapefile

Fonte: CARMONA (2012).

F3 Gerar mapas		Oculto()		
Descrição: o sistema deverá permitir a geração de mapas com as informações geográficas contidas nas tabelas criadas por meio da importação dos <i>shapefiles</i> , permitindo a estilização, escolha e ordenação das camadas serão apresentadas				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1 Controle de Acesso	A função só pode ser acessada por usuários com perfil que permita o acesso a tela de geração de mapas	Segurança	()	(x)
NF3.1 Identificador do mapa	O mapa será identificado por um nome informado na hora de sua geração	Interface	()	(x)

Quadro 8 - Requisito gerar mapas.

Fonte: CARMONA (2012).

F3 Cadastrar perfil de acesso		Oculto()		
Descrição: o sistema deverá permitir o cadastro de perfis de acesso, com os campos de permissões cadastrar, alterar, visualizar, consultar e excluir que será relacionado com as telas do sistema.				

Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1 Controle de Acesso	A função só pode ser acessada por usuários com perfil que permita o acesso a tela de cadastro de perfil.	Segurança	()	(x)
NF3.1 Identificador do mapa	O perfil será identificado por um id	Interface	()	(x)

Quadro 9 - Requisito cadastrar perfil de acesso.

Fonte: CARMONA (2012).

F3 Cadastrar usuário	Oculto()			
Descrição: o sistema deverá permitir o cadastro de usuário com os campos usuário e senha, o usuário sera relacionado com um perfil de acesso já cadastrado.				
Requisitos Não-Funcionais				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1 Controle de Acesso	A função só pode ser acessada por usuários com perfil que permita o acesso a tela de cadastro de usuário.	Segurança	()	(x)
NF3.1 Identificador do mapa	O usuário será identificado por um id.	Interface	()	(x)

Quadro 10 - Requisito cadastrar usuário

Fonte: CARMONA (2012).

4.1.3 Caso De Uso

O diagrama de caso de uso apresentado na Figura 9 descreve as funcionalidades do sistema. O usuário terá acesso aos casos de uso *Manter Shapefile*, *Importar Shapefile* e *Manter Mapa*. Já o administrador, além destes acessos do usuário, possui acesso aos casos de uso *Manter Usuário* e *Manter Perfil de Acesso*.

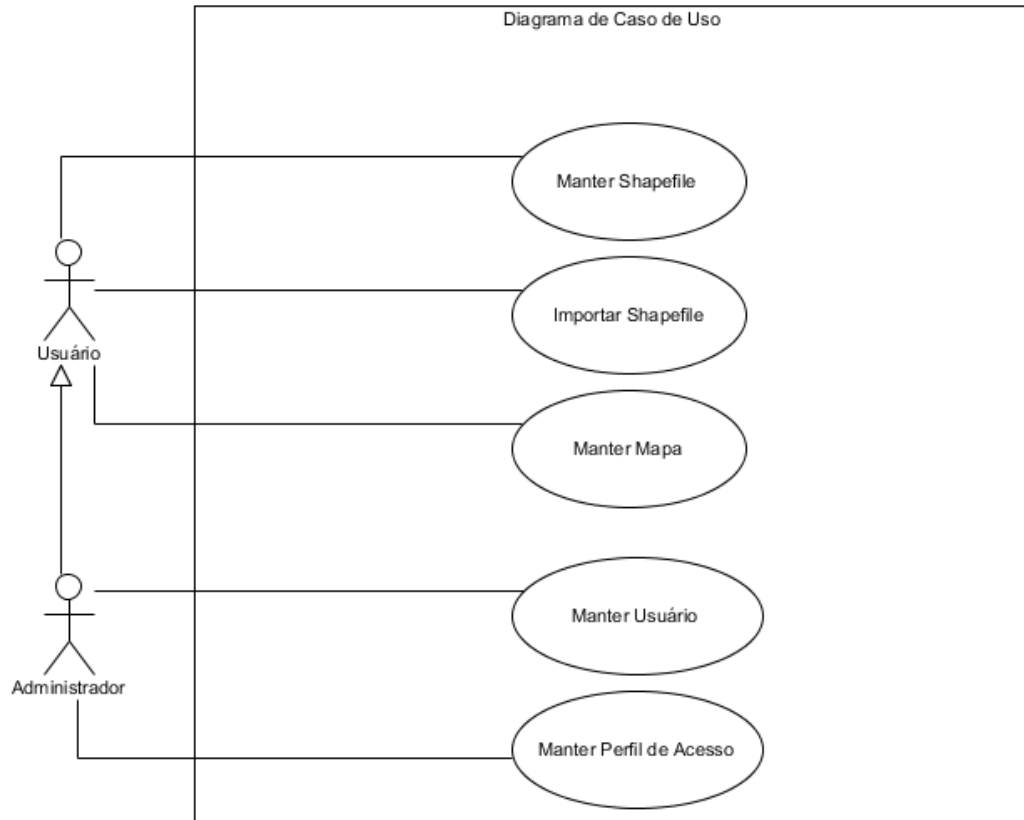


Figura 9 - Diagrama de Caso de Uso.
Fonte: Autoria própria.

4.2 METODOLOGIA

Será realizada a análise e projeto do sistema de geração de mapas desenvolvido por CARMONA (2012), por meio da *Webml*. Será feita a modelagem utilizando o *modelo estrutural*, o qual demonstra o conteúdo dos dados da aplicação, em termos de entidades relevantes e relacionamentos, e *modelo de hipertexto*, o qual descreve os hipertextos do sistema, através da *composição de modelo* e *modelo de navegação*.

4.3 FERRAMENTAS E TECNOLOGIAS

Para a modelagem do sistema foi utilizada a ferramenta WebRatio Personal 7.1.1, utilizada para criação dos diagramas Webml. Foi utilizada esta ferramenta por ser uma ferramenta CASE, recomendada na literatura.

4.3.1 WebRatio

WebRatio é um ambiente de desenvolvimento gratuito que apoia as empresas de *software* no planejamento, produção e manutenção de suas aplicações personalizadas baseadas na *Web*. Oferece a possibilidade de projetar uma aplicação a um nível maior de abstração, representando todos os requisitos funcionais em um modelo visual que é simples e independente da tecnologia.

Atualmente a WebRatio está consolidada na indústria, utilizada em diversas aplicações, além de ser utilizada em várias universidades e instituições para cursos de Engenharia Web. A versão WebRatio Personal está voltada para complexos aplicativos Web, com base no BPMN e linguagem Webml (WEBML, 2013).

5 RESULTADOS E DISCUSSÕES

Esta sessão apresenta os resultados e discussões relativos a modelagem de um sistema Web, a qual foi utilizada o modelo estrutural e o modelo de hipertexto, ambos definidos na *Webml*, modelagem pertencente à Engenharia Web, aplicada ao sistema de geração de mapas, proposto em CARMONA (2012). Estes artefatos são apresentados nas próximas subseções.

Mediante ainda, todo o embasamento teórico elaborado, é possível identificar os artefatos recomendados para o processo de Análise em um sistema Web. WAZLAWICK (2004) apresentou uma relação de artefatos para análise e projeto para sistema objeto orientados, no entanto essa relação não é declarada em sistemas Web. Por meio desta relação (de artefatos), espera-se que o analista possa realizar o processo mais seguro quanto a adoção ou não de tais artefatos

5.1 ARTEFATOS PARA ANÁLISE DE SISTEMAS WEB

Para realizar a análise de um sistema *Web* é necessário levar alguns artefatos em consideração, baseados no modelo tradicional da Engenharia de *Software*, utilizando a linguagem UML e baseado na Engenharia Web, conforme pesquisa realizada anteriormente, utilizando a linguagem *Webml*. São eles:

- a. **Visão Geral:** é necessário obter-se antes de mais nada uma visão geral do sistema, descrevendo qual será o objetivo do mesmo, de acordo com a solicitação do cliente. É de grande relevância na primeira fase, captar todas as perspectivas que o sistema irá abranger.
- b. **Requisitos:** deverá ser coletado e analisado todos os requisitos que englobará o sistema. É fundamental a identificação de todas as necessidades do sistema, de forma detalhada, de acordo com a UML. Nesta fase é conversado com o cliente para conhecer as funcionalidades que o sistema deverá possuir, e posteriormente detalhá-las.
- c. **Diagrama de Caso de Uso:** O diagrama de caso de uso, pertencente a UML, descreve os eventos, casos de uso para um determinado ator. Demonstra a interação entre o

usuário e o sistema. O diagrama descreve as funcionalidades do sistema do ponto de vista do usuário.

- d. **Diagrama de Estrutura:** o diagrama de estrutura da *Webml* demonstra o conteúdo dos dados do sistema. É semelhante ao diagrama entidade-relacionamento ou diagrama de classe, pertencente à UML. O diagrama descreve as entidades, com seus respectivos atributos e relacionamentos.
- e. **Identificação dos Usuários:** é necessário identificar as diferentes visões do sistema referente aos usuários, de acordo com a linguagem *Webml*. Geralmente um sistema possui usuários administradores e usuários não administradores, é necessário diferenciar estes acessos, podendo demonstrar a operação de acesso por um diagrama de hipertexto.
- f. **Diagrama de Navegação:** para aplicações *Web*, é de fundamental importância a análise das páginas que irá compor o sistema, e as ligações entre elas. O diagrama de navegação pertencente à *Webml* permite especificar as unidades e páginas e o modo que se interligam entre si.

5.2 MODELO ESTRUTURAL

O modelo estrutural demonstra o conteúdo dos dados do sistema, em termos de entidades e relacionamentos. A *Webml* utiliza notações clássicas como diagrama de classes da UML para representar este modelo.

A Figura 10 representa o modelo estrutural do sistema de geração de mapas.

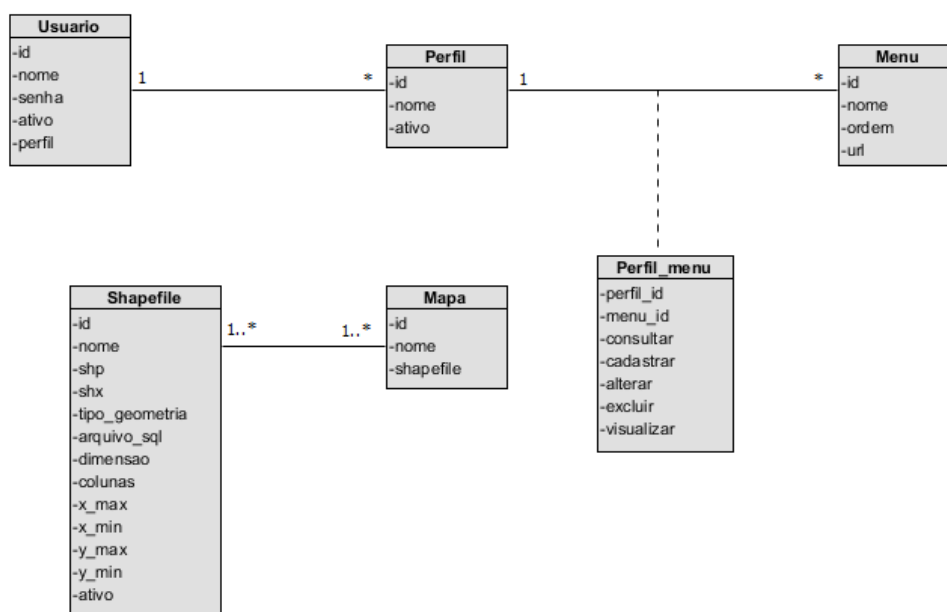


Figura 10 - Diagrama de Estrutura.

Fonte: Autoria própria.

Um usuário deverá pertencer a um perfil e um perfil poderá ter vários usuários. Um perfil ainda ter vários menus, para controle das permissões de acesso. O cadastro de *shapefiles* é realizado por um usuário, o qual deverá adicionar os arquivos *shp*, *shx* e *dpf*, conforme já descrito anteriormente. Para a geração de mapas, é necessário previamente o cadastro de um *shapefile*, o qual será utilizado, para então ser possível a funcionalidade para criação dos mapas. Um mapa poderá possuir um ou mais *shapefiles*.

5.3 MODELO DE HIPERTEXTO

O modelo de hipertexto descreve os hipertextos publicados no *site*. Cada hipertexto diferente define uma visão no sistema. Através deste modelo, características de sistemas *Web*, como fluxo de navegação de páginas e forma de exibição dos dados são representados, não sendo necessário especificar uma linguagem de programação a ser utilizada.

5.3.1 Identificação de Usuários

A primeira identificação a ser realizada, é referente às visões diferentes do sistema. Há dois tipos de usuários, o administrador, o qual terá acesso a todas as funcionalidades do sistema e o usuário, o qual terá acesso limitado no sistema, acessos estes, definidos pelo administrador. Todos os usuários, independente de administrador ou usuário, primeiramente ao acessar o sistema, irão ser direcionados a tela de *login*, onde fornecerão seus dados para logar no sistema. Será realizada uma comparação dos dados fornecidos com os dados contidos no banco de dados. Caso a operação for bem sucedida, havendo autenticação, o usuário será direcionado a tela de menu inicial, o qual terá uma sessão que irá armazenar os dados enquanto o usuário estiver ativo no sistema. Caso a operação não for realizada com sucesso, o usuário será direcionado novamente para a página de *login*.

A Figura 11 ilustra a operação de acesso do administrador. O administrador irá acessar a página *Index*, e nesta efetuar o acesso. Se os dados do acesso forem autenticados, o administrador será direcionado para a página *Menu Principal*, conforme representação da seta de **cor verde**, e posteriormente uma sessão será criada para o administrador. Caso contrário, caso o acesso não for realizado com sucesso, conforme representação da **seta vermelha**, o administrador será redirecionado para a página *Index* novamente.

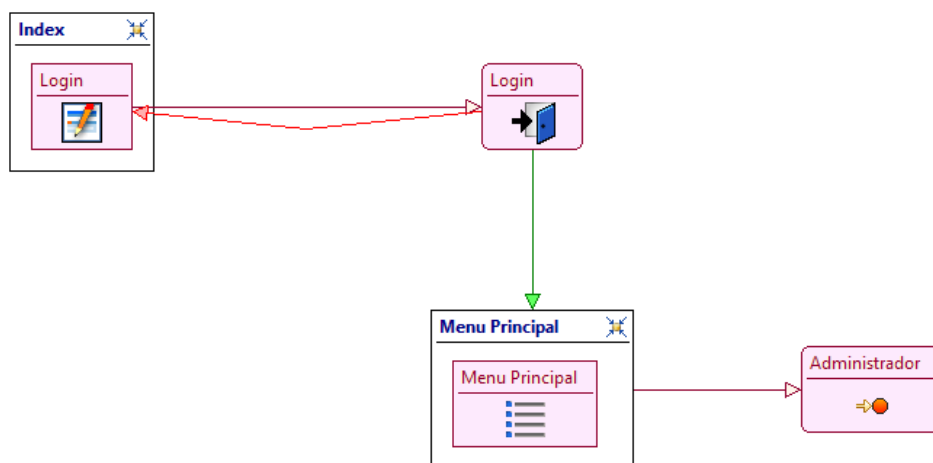


Figura 11 - Operação de acesso para administrador.
Fonte: Autoria própria.

A Figura 12 ilustra a operação de acesso do usuário. A operação é semelhante a do administrador.

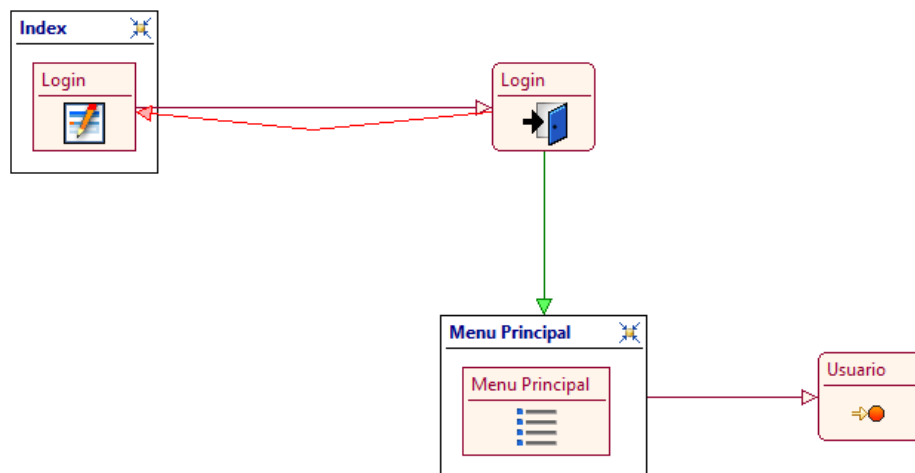


Figura 12 - Operação de acesso para usuário.
Fonte: Autoria própria.

5.3.2 Modelagem das funcionalidades

As funcionalidades do sistema foram modeladas com o diagrama de navegação. Primeiramente será visto as funcionalidades específicas do administrador e posteriormente do usuário. A Figura 13 ilustra o caso de uso de cadastrar perfil.

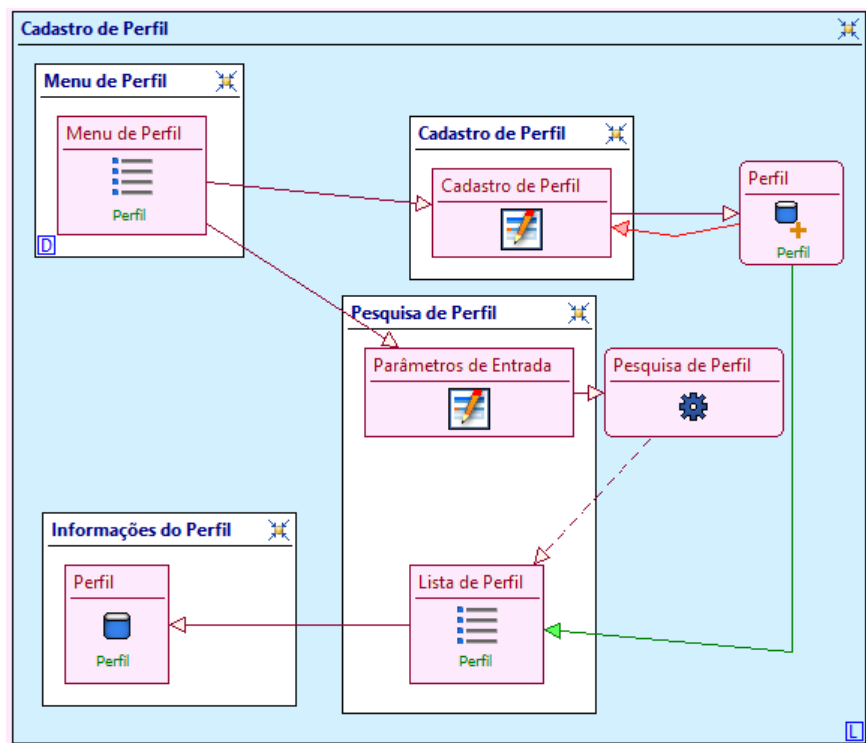


Figura 13 - Diagrama de navegação para cadastro de perfil.
Fonte: Autoria própria.

As ações relacionadas à entidade perfil foram englobadas em uma área chamada cadastro de perfil. Nesta área foram definidas quatro páginas, sendo *default* a página *Menu de Perfil*, sempre que o administrador desejar realizar alguma operação relacionado a entidade perfil, deverá acessar a página de menu e escolher a operação desejada por meio de uma lista de ações, sendo esta representada por uma unidade de índice.

A página *Cadastro de Perfil* está sendo representada por uma unidade de entrada, nesta página há um formulário que o administrador deverá preencher para o cadastro de um perfil. A página *Pesquisa de Perfil* apresenta as ações de busca de registros, é representada por uma unidade de entrada onde o administrador poderá fornecer parâmetros para a pesquisa, depois de informado os parâmetros, uma operação de busca é acionada, a qual irá redirecionar o usuário a mesma página, porém contendo a lista de perfis.

Essa listagem é representada pela unidade de índice, contendo objetos apenas da entidade perfil. Através da listagem, após a escolha de um dos objetos, o administrador será redirecionado para a página *Informações do Perfil*, a qual irá apresentar as informações do objeto selecionado. Na WebML as setas de **cor rosa** indicam o fluxo normal, a **cor verde** indica que o objetivo foi alcançado e a **cor vermelha** indica erro no processo.

A Figura 14 ilustra o caso de uso de cadastrar usuário.

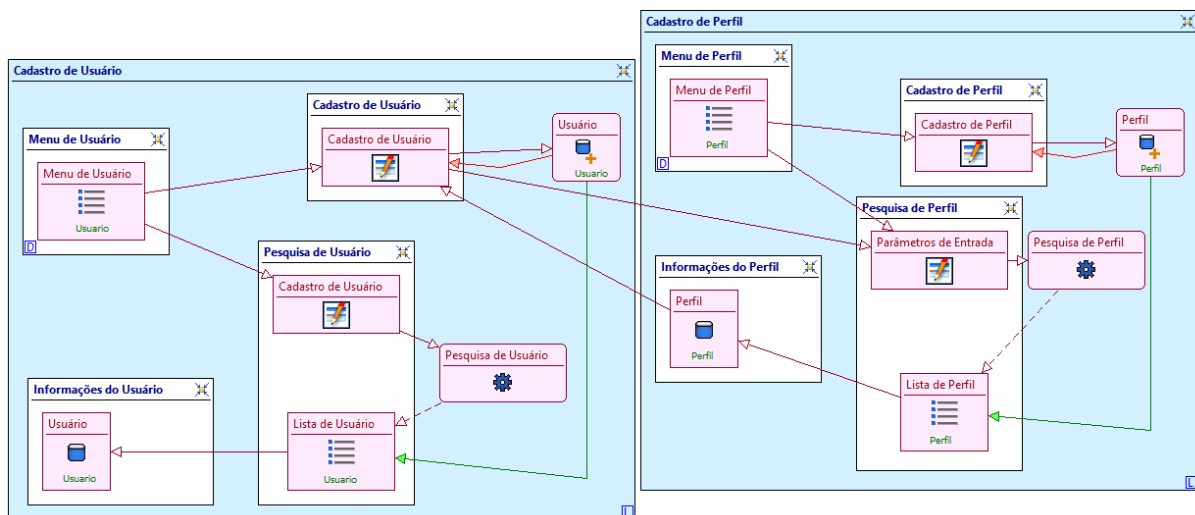


Figura 14 - Diagrama de navegação para cadastro de usuário.

Fonte: Autoria própria.

As operações referentes à entidade usuário são semelhantes com a descrita no perfil, diferenciando apenas na relação que o usuário deverá manter com um perfil. Ao cadastrar um usuário, deve-se indicar a qual perfil o mesmo irá pertencer. Para encontrar o perfil desejado, o administrador será direcionado à página *Pesquisa de Perfil*, onde poderá selecionar o objeto

desejado. As próximas funcionalidades pertencem tanto a administradores, mas também a usuários. A Figura 15 ilustra o caso de uso cadastrar *shapefile*.

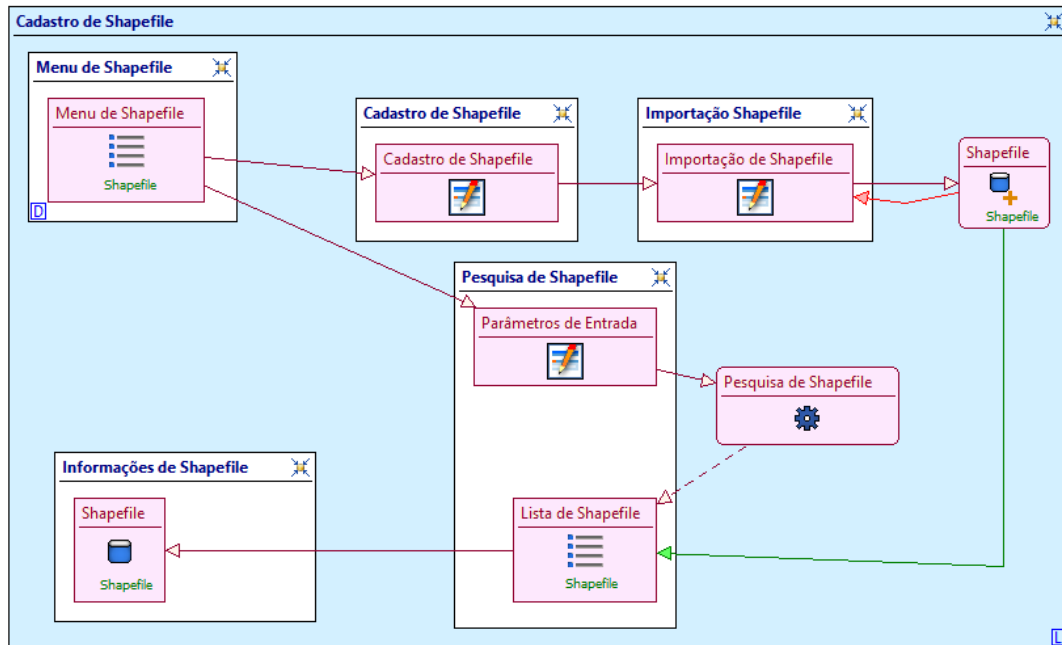


Figura 15 - Diagrama de navegação para cadastro de *shapefile*.

Fonte: Autoria própria.

As operações referentes à entidade *shapefile* são semelhantes com a descrita no perfil. Na área *Cadastro de Shapefile*, também foram definidas páginas de menu, cadastro, pesquisa e informações, diferenciando apenas com a presença de uma nova página *Importação Shapefile*, responsável pela importação do *shapefile* para o banco de dados. A Figura 16 ilustra o caso de uso gerar mapas.

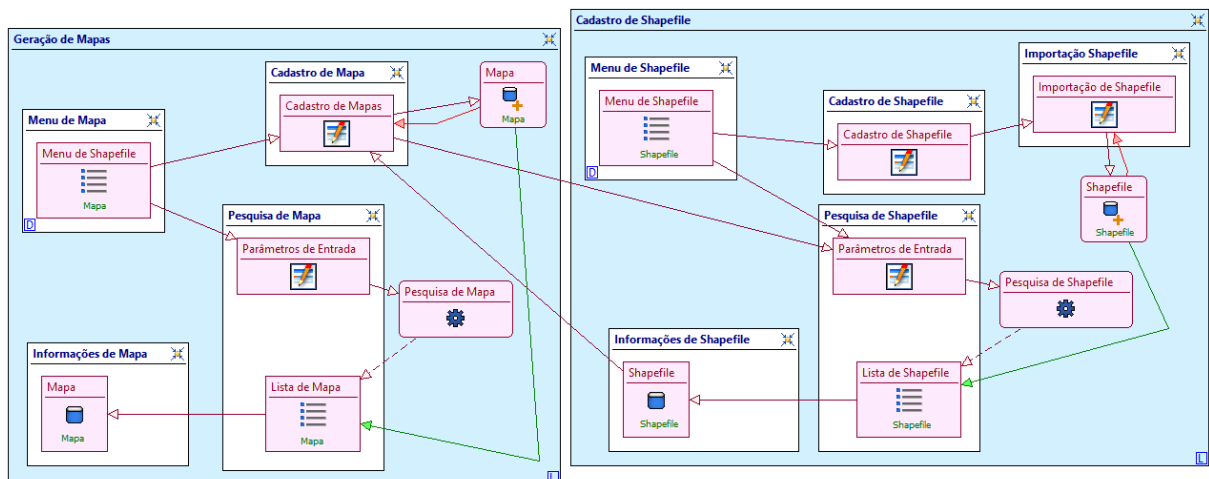


Figura 16 - Diagrama de navegação para gerar mapas.

Fonte: Autoria própria.

As operações referentes à entidade mapa são semelhantes com a descrita no usuário, diferenciando apenas a entidade relacionada, neste caso a entidade mapa se relaciona com a entidade *shapefile*, onde ao cadastrar um mapa é necessário relacioná-lo a um ou mais *shapefiles*.

6 CONSIDERAÇÕES FINAIS

Com o avanço dos sistemas Web, devido ao crescimento da WWW, utilidade em diversas áreas, e a complexidade que estes sistemas vêm apresentando, surgiu à necessidade de modelagem específica para aplicações *Web*, já que estas se diferem de aplicações tradicionais. Questões de conteúdo, estética e navegação possuem grande relevância em uma aplicação Web, juntamente com fatores de segurança, escalabilidade e disponibilidade.

A Engenharia *Web* oferece modelagens adequadas para o projeto de aplicações *Web*, tratando as particularidades que tais aplicações possuem. Uma das modelagens é a *Webml*, que permite analisar as principais características de um sistema em nível elevado, não se comprometendo com os detalhes da arquitetura.

O modelo estrutural, definido pela *Webml*, utiliza as definições da UML para representar entidade-relacionamento, já conhecido no meio. Um dos diferenciais da *Webml* é o modelo de hipertexto (composição e navegação). O modelo de composição utiliza notação gráfica intuitiva, a qual utiliza diferentes componentes presentes em aplicações *Web*, de fácil entendimento por parte dos membros do projeto. O modelo de navegação é um ponto forte da linguagem, expressando por meio de links os fluxos entre as páginas, representadas com suas unidades específicas.

O modelo de apresentação e personalização da *Webml* não define nenhuma metodologia nova. Por exemplo, o modelo de personalização o qual define a separação de usuários e grupos de usuários pode ser realizado e especificado pela metodologia tradicional.

O suporte a ferramentas CASE, como *WebRatio* é um ponto positivo da *Webml*. A ferramenta foi desenvolvida para oferecer suporte à modelagem utilizando a linguagem *Webml*.

6.1 CONCLUSÃO

Com o estudo de caso proposto para o sistema de geração de mapas, conclui-se que:

- a. A *Webml* é uma linguagem de modelagem competente para desenvolvimento de aplicações *Web*.

- b. Em conjunto com metodologias tradicionais pode-se tornar ainda mais eficiente, como por exemplo, a UML, a qual foi utilizada para o desenvolvimento do diagrama de caso de uso, o qual define os usuários e funcionalidades de sistema pertence a cada um, onde a *Webml* não atende a esta necessidade.
- c. *Webml* oferece modelagens desconhecidas no meio da engenharia tradicional, mas de forma simples e intuitiva, podendo eventualmente diferenciar se aplicada em sistemas diferentes e mais complexos.

6.2 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO

Como a Engenharia *Web* fornece outras metodologias e linguagens como HDM, OOHDM e IFML, propõem-se outros estudos experimentais abordando estas metodologias voltadas para aplicações *Web*, possibilitando uma comparação por meio dos novos resultados.

REFERÊNCIAS BIBLIOGRÁFICAS

CARMONA, Lubian. **Aplicação de Sistemas de Informação Geográfica em Ambiente Web – Um Estudo Experimental para Geração de Mapas.** 2012

CERI, Stefano, et al. **Web Modeling Language (Webml): a modeling language for designing Web sites.** 2000. Disponível em: <<http://www.conference.org/proceedings/www9/177/177.html>> Acesso 08 de julho de 2013.

CERI, S.; et al. **Designing Multi-Role Collaborative Web Sites with Webml: a Conference Management System Case Study.** Valencia. 2001. Disponível em: <<http://www.Webml.org/Webml/page92.do?dau39.oid=11&UserCtxParam=0&GroupCtxParam=0&ctx1=EN>> Acesso em 09 de julho de 2013.

CONALLEN J. **Building Web Applications with UML.** 2002. Boston: Addison-Wesley, 2002.

GARZOTTO, Franca, et al. **HDM - A Model-Based Approach to Hypertext Application Design.** 1993. Disponível em: <<http://www.ice.unifei.edu.br/ramos/download/Hipermedia/p1-garzottoHDM.pdf>> Acesso 08 de Julho de 2013.

GINIGE; MURUGESAN, S. **Web Engineering: Introduction and Perspectives.** In: SUH, W. (eds) *Web Engineering: Principles and Techniques.* London: IDEA Group, London, 2005.

GINIGE, MURUGESAN, S. **Web Engineering: An Introduction.** Australian: University of Western Sydney, 2001. Disponível em <<http://www-itec.uni-klu.ac.at/~harald/proseminar/Web1.pdf>> Acesso 01 de julho de 2013.

IFML. **The Interaction Flow Modeling Language.** Disponível em: <<http://www.ifml.org/>> Acesso 26 de agosto de 2013.

ISAKOWITZ, Tomás, et al. **RMM: A Methodology for Structured Hypermedia Design.** 1995. Disponível em:

<<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.63.1769&rep=rep1&type=pdf>>

Acesso 08 de Julho de 2013.

JACYNTO, Mark Douglas de Azevedo. **Processos para Desenvolvimento de Aplicações Web**. Pontifica Universidade Católica do Rio de Janeiro, 2008. Disponível em <ftp://ftp.inf.puc-rio.br/pub/docs/techreports/09_23_jacyntho.pdf>. Acesso 02 de julho de 2013.

KAPPEL, Gerti, et al., **Web Engineering: The Discipline of Systematic Development of Web Applications**. Alemanha: John Wiley & Sons Ltd. All, 2003.

KAPPEL, Gerti, et al., **Web Engineering**. 2006. Disponível em: <<http://big.tuwien.ac.at/research/publications/2004/0404.pdf>> Acesso 05 de julho de 2013.

KOCH, Nora; WIRSING, Martin. **Software Engineering for Adaptive Hypermedia Applications?** 2001. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.24.4017&rep=rep1&type=pdf>> Acesso 08 de Julho de 2013.

LANGEGGER, A., et al., **Simplifying a Web Application's Architecture – the DaVinci Framework**. Disponível em: <http://www.langegger.at/papers/IIWAS05_DaVinci.pdf> Acesso 08 de julho de 2013.

MACHADO, C. Leonardo, et al. **UWE-R: uma extensão de metodologia em Engenharia Web para Rich Internet Applications**. Rio de Janeiro: Universidade do Estado do Rio de Janeiro. 2008. Disponível em: <<http://revistaseletronicas.pucrs.br/ojs/index.php/hifen/article/viewFile/4600/3487>> Acesso 06 de Julho de 2013.

NETO, Arilo Claudio Dias. **Metodologias de desenvolvimento de aplicações Web**. Disponível em: <<http://www.devmedia.com.br/metodologias-de-desenvolvimento-de-aplicacoes-Web-parte-04/9819>> Acesso em 20 de outubro de 2012.

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron. 2000.

PRESSMAN, Roger S. **Engenharia de Software**. Rio de Janeiro: McGraw-Hill, 2002.

PRESSMAN, R. S.; LOWE, D. **Engenharia Web**. Rio de Janeiro: LTC, 2009.

PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. Porto Alegre: AMGH, 2011.

ROSSI, Gustavo; et al. **Systematic Hypermedia Application Design with OOHDM**. 2008.
Disponível em:
<<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.38.7873&rep=rep1&type=pdf>>
Acesso em 10 de Julho de 2013.

SOMMERVILLE, Ian. **Engenharia de Software**. São Paulo: Pearson Prentice Hall, 2011.

SUH, Woojong. **WEB ENGINEERING: Principles and Techniques**. Londres: Idea Group Inc., 2005.

WEBRATIO. **Webratio**. Disponível em: <<http://www.Webratio.com>> Acesso em 25 de julho de 2013.

WEBML. **Webml**. Disponível em: <<http://www.Webml.org>> Acesso 09 de julho de 2013.

ZANIRO, D.; FABBRI, S.. **Um Processo Guiado para Levantamento e Modelagem de Requisitos de Aplicações Web Baseado em Objetivos e Casos de Uso**. Disponível em:
<http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER08/zaniero.pdf> Acesso em 21 de outubro de 2012.