Iniciação Científica **CESUMAR**
Jul./Dez. 2006, v. 08, n.02, p. 157-166

ENGENHARIA DE SOFTWARE PARA WEB

Douglas Eduardo Andreto[1]
Aline Maria Malachini Miotto Amaral[2]

RESUMO: Devido ao aumento de construções de *softwares* e aplicativos para WEB e da complexidade que tais construções vêm apresentando, surge a necessidade de utilização, para o projeto, de uma metodologia ou engenharia específica que venha a atender a todas as características que o processo abrange, visando dinamizar as etapas do processo de desenvolvimento de todas as áreas envolvidas no projeto, mantendo uma melhor qualidade e eficiência no seu desenvolvimento.

PALAVRAS-CHAVE: Engenharia de software para Web, engenharia de hiperdocumento, OOHDM.

SOFTWARE ENGINEERING FOR THE WEB

ABSTRACT: Due to the sharp increase of software and applicative constructions for the WEB and the complexity that such constructions have been presenting, a new need for a methodology or specific engineering that can attend all characteristics involved in the process emerges. This would bring more dynamism to the different stages of the process within all areas involved in the project and, therefore, keep higher quality and efficiency in its development.

KEYWORDS: Software engineering for the WEB; hyperdocument engineering; OOHDM.

[1] Acadêmico do Curso de Processamento de Dados do CESUMAR – Centro Universitário de Maringá, Programa de Iniciação Científica do CESUMAR (PICC). E-mail: douglasandreto@gmail.com

[2] Orientadora e docente do CESUMAR – Centro Universitário de Maringá. E-mail: amiotto@cesumar.br

1.INTRODUÇÃO

Observa-se nos dias atuais que o uso da tecnologia vem ganhando grande importância nos processos de tomada de decisão das organizações, que cada vez mais procuram agilidade e confiabilidade em seus processos. Diante desse contexto, os aplicativos computacionais na WEB se apresentam como uma excelente alternativa, pois permitem a disponibilização e o processamento eficiente de uma grande quantidade de informações, com um diferencial: a independência geográfica, em que um sistema gerencia informações de diferentes locais a um baixo custo.

Não obstante, com o aumento da necessidade de aplicativos para WEB e o conseqüente aumento da complexidade que esses aplicativos vêm apresentando, torna-se necessária a definição e utilização de métodos ou processos de engenharia específicos para o projeto de tais aplicações, uma vez que estas apresentam uma série de particularidades, como a definição da sua estrutura de navegação.

Neste contexto, o objetivo deste trabalho é estudar e identificar o suporte oferecido pelo método OOHDM da Engenharia de Software para as características dos aplicativos para Web. O OOHDM é uma metodologia de desenvolvimento para aplicações hipermídia em larga escala, podendo ser aplicações para CD-ROM, quiosque ou aplicações WEB. O OOHDM é independente da escolha de linguagens e ambientes de programação, tornando a escolha do projeto mais ampla, o que permite trabalhar tanto com o paradigma de orientação a objetos quanto com programação estruturada.

Para este trabalho foi realizado um estudo de caso para o desenvolvimento de um *software* para Web. O *software* em desenvolvimento deverá realizar o controle do horto de plantas medicinais do Cesumar, onde, utilizando-se a metodologia OOHDML, foram definidos a estrutura do sistema, seu projeto de interface e seu modelo de navegação.

2.OOHDM

Rossi (1996 apud LEITE, 2003) afirma que, com a utilização de abordagens oriundas da engenharia de software e, sobretudo, da metodologia de orientação ao objeto, o autor consegue chegar a um texto onde a expressão maior é traduzida na modelagem abstrata das aplicações, isto é, faz a modelagem de uma aplicação sem ter como foco a implementação da aplicação em questão. À modelagem abstrata soma-se a facilidade de ampliação do projeto, desde que se tenha em mão uma abstração da ampliação a ser executada em nível condizente com a modelagem já existente.

Segundo Rossi (1996 apud FELTRIN 1999), o OOHDM considera o processo de desenvolvimento de uma aplicação hipermídia como um processo de quatro atividades, desempenhadas combinando-se estilos iterativos e incrementais de desenvolvimento; onde em cada etapa um modelo é construído ou enriquecido.

A metodologia OOHDM provê primitivas de projeto de alto nível e mecanismos de abstração, baseados no paradigma da orientação a objetos, que permitem representar o projeto de aplicações hipermídia complexas que manipulam grande quantidade de informações estruturadas, tais como aplicações para a web, apresentações multimídia, quiosques, etc. (MEDEIROS, 2001). No entanto, o OOHDM é uma metodologia de desenvolvimento para aplicações hipermídia em larga escala, podendo ser aplicações para CD-ROM, quiosque ou aplicações WEB.

O OOHDM tem como precursor o modelo HDM. O OOHDM herdou do HDM as seguintes características:

- reconhecimento de que a modelagem é independente do ambiente e do modelo de referência;
- as estruturas hierárquicas são reforçadas, oferecendo a possibilidade da construção de agregados;
- inclui o conceito de perspectiva de atributo, onde cada classe implementa uma visão possível do atributo.

A metodologia OOHDM foi enriquecida com as seguintes características:

- não é apenas uma abordagem de modelagem, mas sim, uma metodologia, pois aborda muitas atividades;
- a modelagem conceitual é melhor, pois ela é orientada a objeto, suportando agregação, generalização e herança;
- permite o uso do esquema conceitual para definir possíveis perfis de usuários diferentes.

O processo de desenvolvimento da metodologia OOHDM possui 4 fases:

- modelagem conceitual;
- modelagem navegacional;
- projeto de interface abstrata;
- implementação.

As características das atividades no processo de desenvolvimento OOHDM são:

Figura 1 – Esboço da Metodologia OOHDM

Fonte: Rossi, 1996 apud Feltrin, 1999, p. 48

Uma das características do OOHDM é que os processos são executados iterativamente, sendo possível a realização de atividades em paralelo com as outras e podendo-se retornar para qualquer atividade quando necessário.

Com o OOHDM é possível desenvolver projetos modulares e de fácil manutenção, uma vez que as quatro etapas podem ser desenvolvidas separadamente, sendo possível retornar a uma etapa construída, quando necessário, o que permite nos concentrarmos em diferentes interesses do projeto. Sendo assim, obtemos uma estrutura lógica dos processos de todo o projeto, adquirindo uma experiência específica para cada atividade.

O OOHDM é independente da escolha de linguagens e ambientes de programação, o que torna a escolha do projeto mais ampla, podendo-se trabalhar tanto com o paradigma de orientação a objetos quanto com a programação estruturada, e ainda com ambos ao mesmo tempo.

2.1. MODELAGEM CONCEITUAL

De acordo com Feltrin (1999), o foco da modelagem conceitual está em modelar os objetos que constituem o domínio e os relacionamentos entre eles, sendo que nessa atividade observa-se uma maior preocupação com a estrutura dos objetos do que com seu comportamento.

Locatelli (2003) afirma que o ponto-chave da modelagem conceitual é analisar o domínio da aplicação, com o que se obtêm todas as informações importantes para o desenvolvimento do projeto. Isso não quer dizer que todas essas informações serão utilizadas na implementação, pois partes dela poderão ser descartadas.

A modelagem conceitual tem como objetivo a análise completa de todo o domínio da aplicação, para a obtenção de todas as informações úteis para o desenvolvimento da aplicação; mas nem todas as informações serão utilizadas, elas poderão ser descartadas a qualquer momento.

O objetivo do processo "modelagem conceitual" é a construção de um esquema contendo classes, objetos, relacionamentos e subsistemas existentes para o domínio da aplicação.

A modelagem conceitual contém os objetos do domínio da aplicação (classes, relacionamentos e subsistemas); e para a construção dos objetos da aplicação, é utilizada uma notação baseada em UML.

Exemplo de classe com relacionamentos e múltiplos atributos:

Figura 2 – Relacionamento de múltiplos atributos

Fonte: Locatelli, 2003, p. 24

As classes em OOHDM são iguais às classes utilizadas em modelagens orientadas a objetos, representando um conjunto de entidades com características semelhantes. No exemplo acima, temos a classe *equipamento* agregada com a classe *laboratório*.

Os relacionamentos fazem ligação entre os objetos e podem ser unários, binários e ternários. No exemplo acima, temos um relacionamento binário entre laboratório e equipamento.

A cardinalidade ou multiplicidade é representada por um subconjunto de inteiros não negativos, especificando quantas instâncias de uma classe relacionam-se a uma única instância de uma classe associada. No exemplo acima, temos que um laboratório contém apenas um ou infinitos equipamentos, e um equipamento está contido em apenas um laboratório.

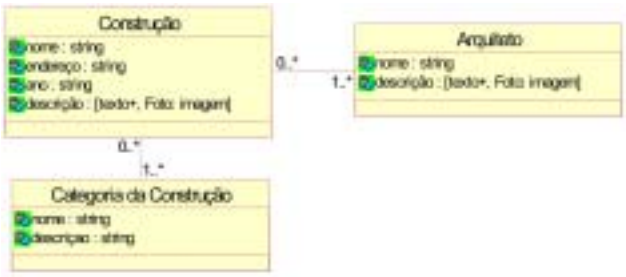
As classes podem ter seus relacionamentos representados por meio da agregação ou generalização/especialização, demonstrando como se relacionam e qual a sua hierarquia. No exemplo acima temos que o laboratório é composto por equipamento(s).

Atributos de classes representam as propriedades dos objetos. Em OOHDM é possível se ter mais de um valor diferente para o mesmo atributo. Para representar múltiplas perspectivas em um atributo, estas deverão estar entre colchetes "[]", tendo um identificador seguido do caracter dois pontos ":" e de um tipo. No exemplo acima temos o atributo descrição da classe equipamento tendo duas perspectivas: um texto para sua descrição e uma foto para sua representação.

O atributo com múltiplas perspectivas é apresentado, no mínimo, em uma perspectiva, sempre que o objeto é mostrado, sendo *uma delas default,* representada pelo caractere "+". A perspectiva *default* é aquela em que o atributo será apresentando em todas as instâncias, e é a única que não precisa apresentar um identificador, recebendo neste caso o identificador do atributo.

Para representar as classes, bem como seus relacionamentos, utiliza-se o esquema conceitual, não sendo necessário utilizar-se de qualquer método em particular parar construir o esquema conceitual de classes. O exemplo abaixo é de um esquema conceitual de um *site* sobre arquitetura.

Figura 3 – Exemplo de um esquema conceitual



Fonte: Locatelli, 2003, p. 26

2.2. MODELAGEM NAVEGACIONAL

Segundo Leite (2003), os aplicativos do tipo hipermídia são projetados para realizar navegação através de um espaço de informação. Por isso, o projeto da estrutura de navegação de tais aplicativos é uma etapa crucial no empreendimento do desenvolvimento, onde o foco está nos objetos que compõem o aplicativo e os relacionamentos do domínio.

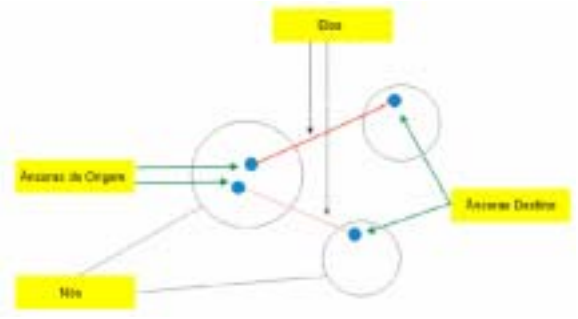
Uma das dificuldades encontradas em um desenvolvimento para Web é definir quais serão as informações que farão parte da aplicação, bem como a navegação da mesma. No OOHDM, a modelagem navegacional pode ser definida a partir da modelagem conceitual ou de acordo com os perfis de usuários para uma determinada aplicação.

Esquema de classes navegacionais

Classes navegacionais são um diagrama onde se define o conjunto de nós e elos que abrangerão o contexto navegacional da aplicação. As linhas que representam os elos são direcionadas representando uma possível navegação.

Exemplo de uma estrutura básica de um hiperdocumento:

Figura 4 – Estrutura básica de um hiperdocumento



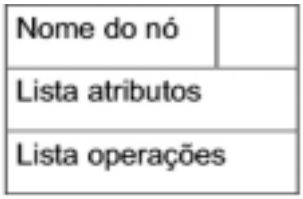
Fonte: Locatelli, 2003, p. 11

Nós

Nó é uma classe, e representa um conjunto de instâncias com características semelhantes (atributos, elos ou métodos). Os atributos dos nós são "imagens" de objetos de classes conceituais, e podem acessar diferentes classes relacionadas no esquema conceitual, sendo estas orientadas para uma certa classe de usuários e suas tarefas. Neste caso também é possível a utilização de generalização/especialização.

A diferença representacional entre uma classe e um nó é que o nó possui uma linha vertical logo após seu nome, como mostra o exemplo abaixo:

Figura 5 – Representação de um nó



Fonte: Schwabe; Vilain, 1999 apud Locatelli, 2003, p. 28

Âncoras e índices

Âncoras e índices da classe navegacional são apresentados como atributos, podendo referenciar outras informações.

A sintaxe de âncora que acessa elementos é: nome_âncora: âncora (índice (nome_contexto(parâmetros))). Exemplo: const_arq: âncora(índice (construção por arquiteto(self))).

Âncoras e índices também podem referenciar informações que não sejam necessariamente elementos de um contexto. Exemplo: nome_arquiteto: lista de projetos(âncora(projeto)). Este exemplo acessa uma lista de projeto(s) do(s) arquiteto(s).

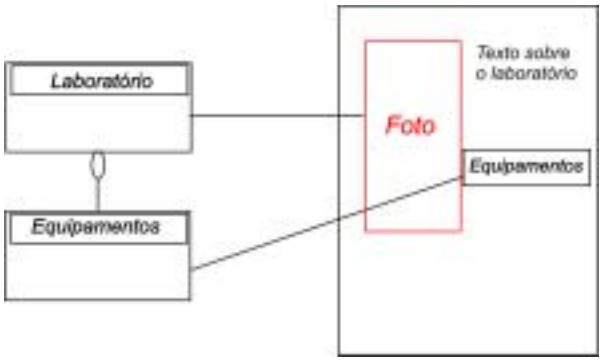
Ligações ou elos

As ligações ou eles têm como função ligar as páginas(nós), formando uma rede de informações. São os elos que definem ao usuário seu caminho de navegação através do hiperdocumento.

Classes navegacionais

É possível a utilização de um nó composto para objetos agregados do esquema conceitual, reduzindo-se assim a complexidade e aumentando a simplicidade para a definição da semântica da navegação entre os nós, como no exemplo abaixo:

Figura 6 – Semântica de navegação entre os nós

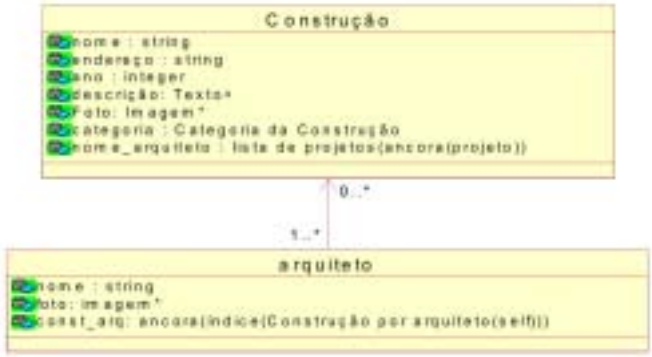


Fonte: Locatelli, 2003, p. 29

Para que a navegação seja intuitiva, esta deve permitir que o usuário navegue no interior dos nós de maneira coerente, no caso acima um nó composto.

A seguir observamos um exemplo de classe navegacional, de acordo com o esquema conceitual do arquiteto.

Figura 7 – Exemplo de uma classe navegacional



Fonte: Locatelli, 2003, p. 29

Atributos multivalorados

Os atributos são os mesmos das classes conceituais, uma vez que as classes navegacionais são imagens das conceituais; porém atributos multivalorados têm que ser mapeados separadamente na classe navegacional, não podendo ser multivalorados. Exemplo, o atributo da classe conceitual "descrição: [Texto+, Foto:imagem]" aparece na classe navegacional separadamente, "descrição: Texto+" e "Foto:Imagem*". O atributo *default* é igual aos atributos comuns, já os opcionais são seguidos pelo símbolo "*".

Diferenças entre classe navegacional e conceitual

Em OOHDM, os objetos navegacionais são tratados diferentemente dos objetos conceituais.

Temos o exemplo do nó "construção"(classe navegacional), sendo uma imagem da classe conceitual "construção" e "categoria da construção", tendo o nó o atributo "categoria", que obtém o atributo "nome" da classe conceitual "categoria da construção".

Contexto de navegação

O contexto de navegação é formado por um grupo pré-definido de classes navegacionais, composto por objetos como: nós, *links*, âncoras e estruturas de acesso.

O contexto navegacional tem uma importância considerável em OOHDM, pois as ligações poderão ser usadas de maneiras diferentes pelos usuários finais, tendo diferentes aplicações para o mesmo contexto. Assim que as classes navegacionais forem concluídas, é muito importante que se estruture o espaço navegacional que será disponibilizado para os usuários. Ex.: o *site* sobre arquitetura poderia conter complexas fórmulas matemáticas por categoria de construção para arquitetos que o visitassem, e não possuir tal informação para usuários que não são arquitetos, que desejam apenas visualizar as obras realizadas.

O espaço navegacional é estruturado em grupos chamados de contextos, sendo composto por: elementos, especificação da estrutura navegacional, um ponto de entrada, acesso e restrições de classes de usuários e suas operações.

São definidos seis caminhos diferentes para a exibição de contextos.

Derivado de classe simples

Compõe-se de todos os objetos que satisfaçam alguma propriedade, como, por exemplo: "Arquitetos com nome = João".

Derivado de grupo de classes

Compõe-se um grupo de contexto derivado de classes simples, tendo parametrizado a propriedade de cada contexto, como, por exemplo: "arquitetos por região (a região é variável)".

Derivado de ligações simples

Compõe-se dos objetos relacionados com outro objeto, como, por exemplo: "Construções projetadas por João da Silva".

Derivado de ligações compostas

Compõe-se de um grupo de contextos derivados de ligações, onde o objeto obtido varia conforme o objeto fonte da ligação, como, por exemplo: "Construções projetadas por arquiteto" (neste exemplo o arquiteto pode variar).

Arbitrário

Os elementos do contexto podem ser escolhidos arbitrariamente, a partir de uma ou mais classes, por exemplo: "Um roteiro dos arquitetos que possui 2m projetos e são pertencentes ao conselho de arquitetos".

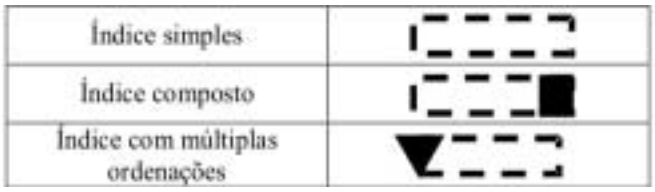
Dinâmico

Os elementos podem ser modificados conforme a navegação

do usuário, como, por exemplo: “Arquitetos, construções”.

Quando existir uma estrutura de acesso (índices) para os contextos acima, será utilizada uma notação correspondente para representação, e esta irá conter um pequeno quadro no canto esquerdo. As representações gráficas das estruturas de acesso associadas ao contexto podem ser:

Figura 8 – Representação gráfica das estruturas associadas ao contexto



Fonte: Schwabe et al., 1999 apud Locatelli, 2003, p.34

Podemos ilustrar o exemplo acima com a demonstração de um índice com múltiplas ordenações, onde a estrutura de acesso pode possuir vários critérios para sua ordenação e, conforme a situação, alternar entre esses critérios. Exemplo:

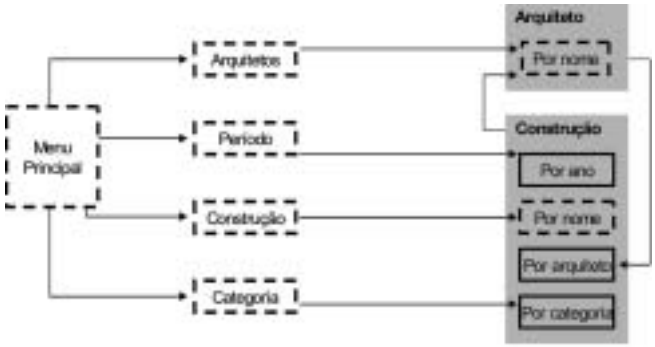
Figura 9 – Exemplo de um índice com múltiplas ordenações

Fonte: Locatelli, 2003, p. 34

Em OOHDM, a estrutura navegacional das aplicações é definida no esquema de contextos de navegação, que demonstra todas as estruturas de acesso, contextos definidos para a aplicação e a possível navegação entre eles.

A seguir temos um exemplo do esquema de contexto de navegação para o menu do *site* de arquitetos:

Figura 10 – Exemplo de um esquema de contexto de navegação



Fonte: Schwabe et al., 1999 apud Locatelli, 2003, p. 35

Analisando o menu acima temos:
Arquitetos: fornece acesso à lista de arquitetos pelo nome.
Período: fornece acesso às construções por ano de construção.
Construção: fornece acesso à lista de construções por nome.
Categorias: fornece acesso às construções por categoria (casa, edifícios, monumentos, etc...).

Podemos observar ainda que as construções podem ser acessadas de acordo com o(s) arquiteto(s) que os projetou(aram) (este contexto pode somente ser acessado por outros contextos, como, por exemplo, os “arquitetos”), e enquanto navegamos por uma construção em algum dos seus contextos, podemos escolher em qual contexto desejamos navegar. Exemplo: “enquanto olhamos uma construção de um certo arquiteto, podemos navegar para a próxima construção do mesmo ano ou da mesma categoria”.

2.3. PROJETO DE INTERFACE ABSTRATA

De acordo com Cunha (2002 apud LEITE, 2003), a construção de uma interface hipermídia é um aspecto crítico da criação de um programa aplicativo hipermídia. Para especificar um modelo abstrato de interface é necessário definir metáforas de interface e descrever suas propriedades estáticas e dinâmicas e seus relacionamentos com o modelo navegacional de uma forma independente de implementação.

Breve (2002, p.15) informa que, na Web:

> a interface tem um papel ainda mais importante que no *software* tradicional: ela é a primeira impressão. Uma interface mal-desenhada pode desapontar o usuário e este pode procurar outro *site*; e ainda pior: provavelmente ele não voltará mais ao *site* de que não gostou.

Segundo Locatelli (2003), no projeto de interface abstrata serão definidos os botões de controle, menus, barras de menu - enfim, todos os objetos que farão a interface com o usuário. Como o projeto de interface abstrata é executado antes da implementação, deve-se ficar atento aos tipos de objeto que serão colocados na interface, pois o ambiente de implementação utilizado poderá não suportar algumas características mapeadas.

É necessário especificar no projeto de interface abstrata a representação de cada objeto navegacional, ou seja, seus atributos e objetos de interface (menus, botões de controle, etc.), o relacionamento entre os objetos de interface e navegacionais, mostrando o resultado de um evento externo, por exemplo: a relação de um evento externo (clique do usuário) e o que acontecerá na navegação, as transformações de interface após o efeito da navegação e a sincronização e alguns objetos de interface, especialmente se houver meios dinâmicos, como áudio, vídeo, etc.

Um dos maiores problemas na construção de aplicações para web é a criação de interface para os usuários finais, pois nem todos os objetos usados na composição das interfaces são esboçados, não havendo assim uma especificação completa dos objetos que irão compor a interface com o usuário.

2.4. IMPLEMENTAÇÃO

Nesta etapa, os modelos anteriores serão traduzidos para um

ambiente de implementação específico, sendo esta a fase final da modelagem OOHDM. Há uma grande preocupação com a interface para o usuário, a qual deve garantir uma fácil navegação, aprendizado fácil, eficiência, memorabilidade, minimização dos erros e satisfação, garantindo a usabilidade do projeto.

O OOHDM possui um ambiente específico para implementação, denominado OOHDMweb. OOHDMweb oferece facilidades como: criação automática de índices em páginas html e representação dos contextos de acordo com as informações contidas nas tabelas.

2. RESULTADOS OBTIDOS

Inicialmente os requisitos do sistema proposto – Horto de Plantas Medicinais do Cesumar – foram levantados e as técnicas utilizadas foram coleta de documentos e entrevistas e questionários.

Na fase de concepção do projeto, a preocupação está em analisar todo o domínio da aplicação, para obter todas as informações que serão úteis durante as fases do projeto.

Na fase de concepção, ao invés de usarmos o diagrama de classes como esboço para depois ampliá-lo na classe navegacional, nós deixamos esse serviço para a equipe responsável pela classe navegacional, fazendo-o apenas uma vez e usando-o em apenas uma fase, e adotamos nesta fase o diagrama de caso de uso da UML para modelar o domínio da aplicação.

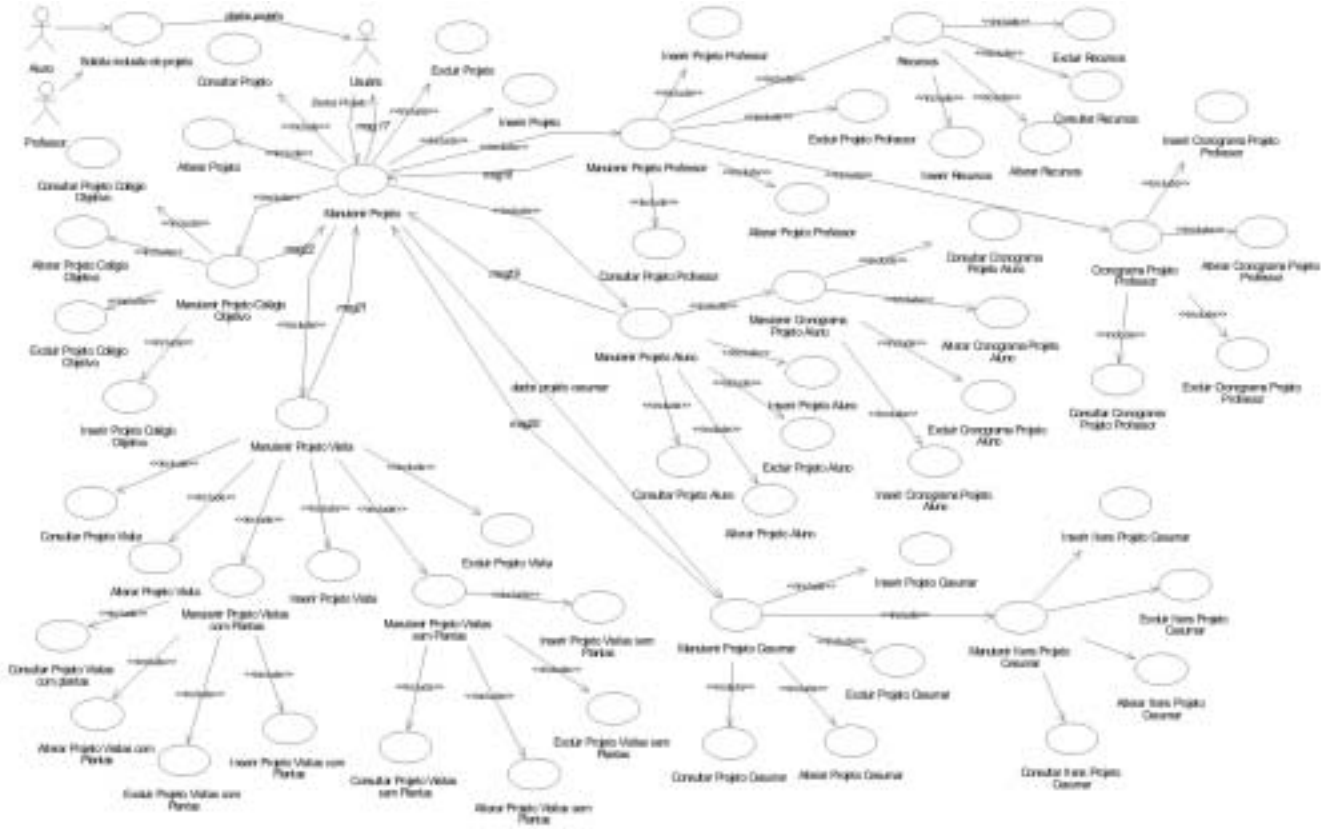
Segundo Medeiros (2004), o caso de uso é um instrumento que acompanha um *software* do seu início até a sua conclusão, e é uma ferramenta de consulta, acerto, discussão, reuniões, alterações em requisitos e alterações em desenho, sendo uma análise intrínseca de um negócio dentro de um processo de desenvolvimento de *software*.

De acordo com Conallen (2003), os casos de uso são uma técnica poderosa para capturar e expressar o comportamento detalhado do sistema, e os casos de uso são uma maneira formal de capturar e expressar a interação e o diálogo entre usuários do sistema, denominados atores, e o próprio sistema.

As figuras a seguir representam alguns dos casos de uso desenvolvidos para a solução de desenvolvimento do sistema proposto – Horto de Plantas Medicinais do Cesumar:

Figura 11 – Casos de uso do sistema proposto

Figura 12 – Casos de uso do Sistema Proposto



Na fase de modelagem navegacional, foram definidos o menu principal e as possíveis interações entre as páginas do sistema. O diagrama abaixo esboça o menu principal do sistema:

Figura 13 – Modelo navegacional – menu principal

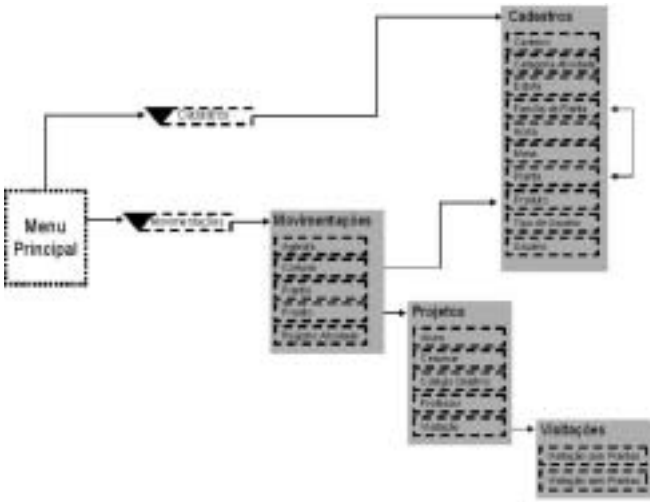
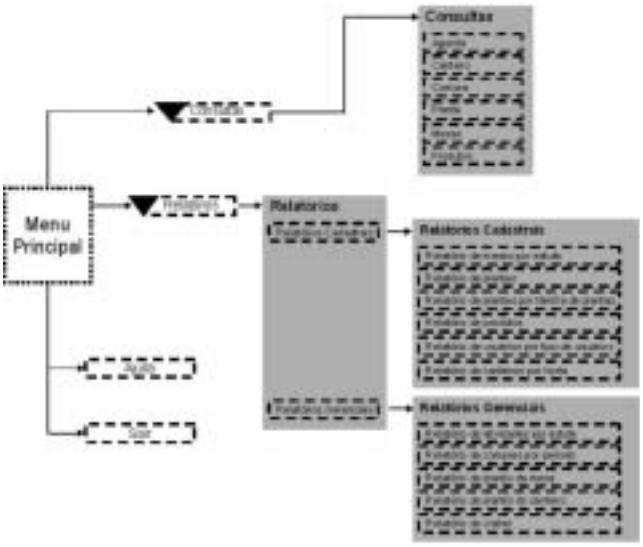
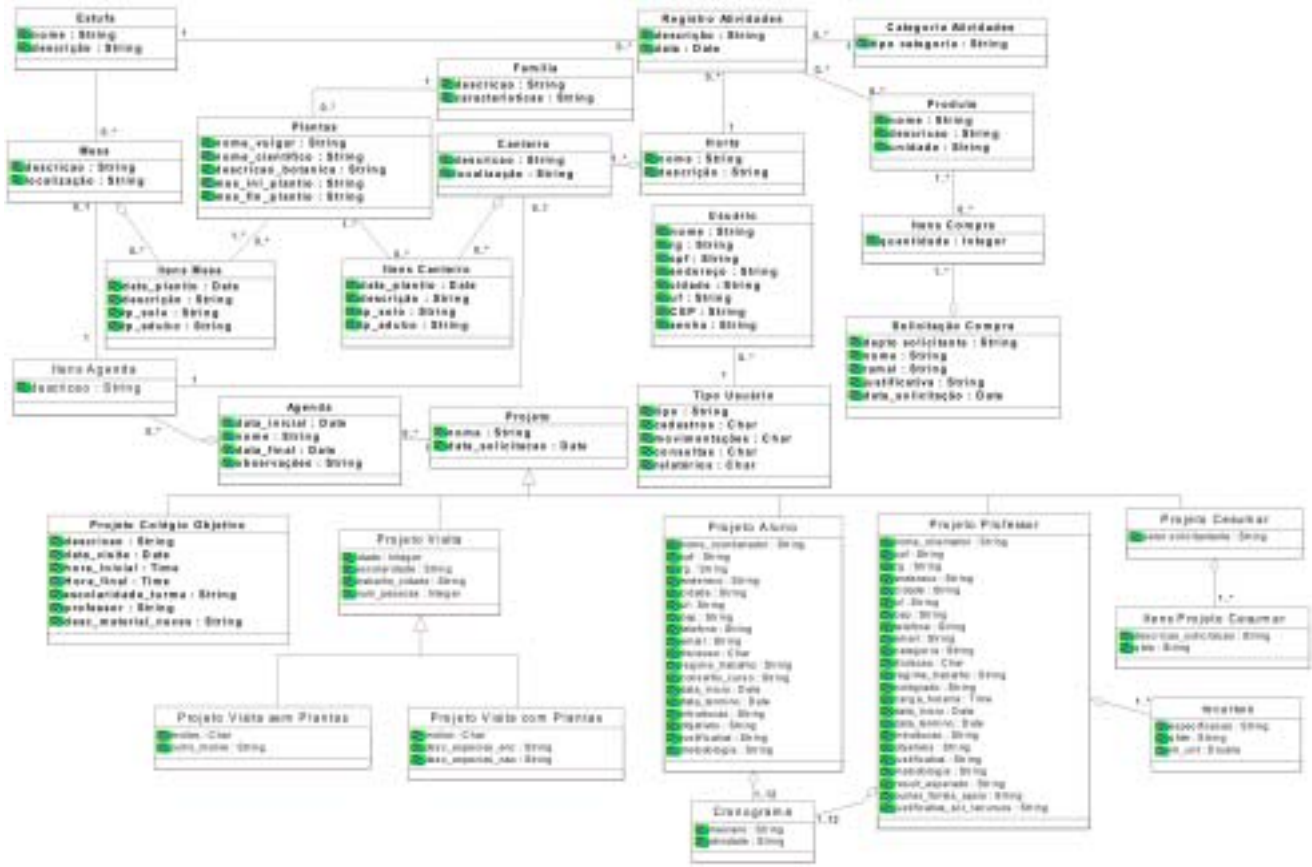


Figura 14 – Modelo Navegacional – Menu Principal



O diagrama abaixo esboça a classe navegacional do sistema proposto:

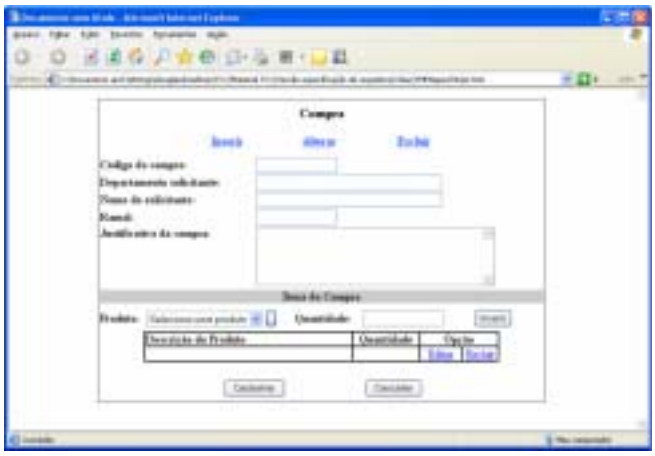
Figura 15 – Classe Navegacional do Sistema



Na fase *projeto de interface abstrata*, as definições do sistema foram esboçadas em forma de protótipo, lembrando-se que nesta fase as representações têm que manter uma abstração suficiente, que a deixe independente da linguagem de implementação.

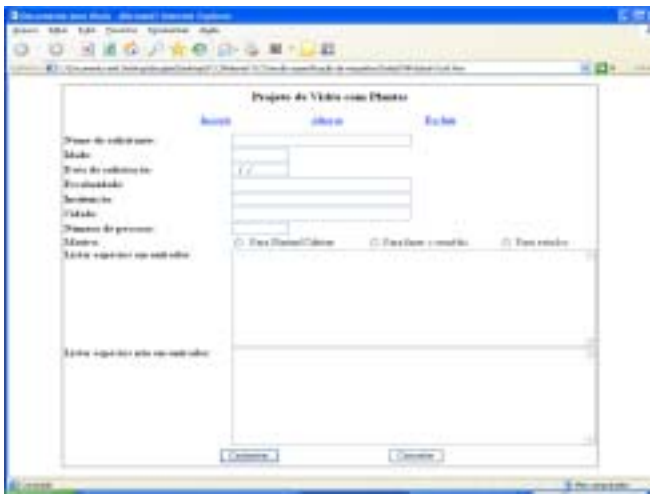
A figura a seguir representa uma interface do sistema proposto, neste caso, a tela de solicitação de compras:

Figura 16 – Representação da Movimentação Compra



A figura a seguir representa a interface da tela de Projeto de Visitas com Plantas:

Figura 17 – Representação do Projeto de Visitação com Plantas



Concluímos com este processo na fase implementação, onde o sistema foi implementado utilizando-se a linguagem de programação PHP com o banco de dados relacional MySQl.

4. CONCLUSÕES

Este trabalho apresentou uma discussão sobre o uso da metodologia OOHDM e suas fases para o desenvolvimento de aplicativos WEB.

Destacamos a necessidade e as vantagens de se utilizar uma metodologia específica durante o desenvolvimento de projetos WEB.

No desenvolvimento, observamos que as fases não são complexas e são desenvolvidas independentemente das demais, o que torna todo o processo de desenvolvimento mais dinâmico e rápido.

Na fase conceitual, optamos por usar diagramas de casos de uso para modelar o domínio da aplicação ao invés de utilizar o diagrama de classe que seria uma abstração para a classe navegacional (fase posterior), fazendo com que essas duas fases sejam realmente independentes entre si e evitando que um trabalho tenha que ser refeito. Lembramos que os casos de uso são uma ótima ferramenta não só para modelar o domínio do negócio, mas também para demonstrar o relacionamento do sistema com os usuários do mesmo.

O projeto de interface abstrata auxiliou muito no processo de refinamento dos requisitos. O usuário teve mais facilidade em opinar sobre o sistema, analisando as interfaces abstratas, e esse projeto se mostrou realmente independente da escolha do ambiente de implementação.

Concluímos o desenvolvimento deste projeto com a escolha da linguagem de programação e do banco de dados utilizados para a implementação da solução proposta. O sistema foi desenvolvido e o manual de análise proposto pela metodologia OOHDM foi completado.

REFERÊNCIAS

BREVE, Fabrício Aparecido. **Engenharia para WEB**. 2002. 25 Fls. Trabalho de Curso - Universidade Federal de São Carlos, São Carlos, 2002. Disponível em: <http://www.dc.ufscar.br/~fabricio/trabalhos/engenharia_web.pdf>. Acesso em: 15 set. 2006.

CONALLEN, Jim. **Desenvolvimento de aplicações Web com UML.** Rio de Janeiro: Campus, 2003.

FELTRIN, Valéria Delisandra. **Apoio À Documentação de Engenharia Reversa de Software por meio de Hipertextos**. 1999. 120p. Tese (Mestrado em Ciências de Computação e Matemática Computacional) - Universidade de São Carlos, São Carlos, 1999. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-01022001-172954/>. Acesso em: 15 ago. 2005.

LEITE, Rodrigo Nazaré da Silva. **Modelagem OOHDM do Portal Corporativo da 6ª Região da PMMG**. 2003. 68 fls. Tese (Bacharelado em Ciência da Computação). Universidade Federal de Lavras, Lavras, 2003. Disponível em:<http://www.comp.ufla.br/monografias/ano2003/Modelagem_OOHDM_do_portal_corporativo_da_6_regiao_da_PM_MG.pdf>. Acesso em: 07 ago. 2005.

LOCATELLI, Márcio Henrique. **Engenharia de Software para o desenvolvimento de WEBAPPS e as metodologias OOHDM e**

WEBML. 2003. 57p. Tese (Especialização em Ciência da Computação) - Universidade Federal de Santa Catarina, Santa Catarina, 2003. Disponível em: <http://www.inf.ufsc.br/~leandro/ensino/esp/monografiaMarcioHenriqueLocatelli.pdf>. Acesso em: 15 set. 2006.

MEDEIROS, Adriana Pereira de. **Especificação Declarativa e Implementação de Aplicações Hipermídia na WEB**. 2001. 231p. Dissertação (Mestrado em Ciência da Computação) - Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2001. Disponível em: <http://www-di.inf.puc-rio.br/schwabe/papers/Adriana%20SBMidia%202001.pdf>. Acesso em: 15 ago. 2005.

MEDEIROS, Ernani. **Desenvolvendo software com UML 2.0:** definitivo**.** São Paulo: Pearson Makron Books, 2004.